Statistical Methods - Part 1

Philip Dahlqvist-Sjöberg

18^{th} of May 2020

1 Gibbs sampler

1.1 Reproduce figure 1,3 and 5

Casella & George (1992) gives a brief illustrative walk through of how the Gibbs sampler works. In particular, there are three graphs, that provide an idea of how powerful the method is, Figure 1,3 and 5 in their article. The use of a Gibbs sampler is when calculating, an arbitrary function $f(x)$, density function directly from integration, which can be problematic for curtain functional forms. The idea is then, to use knowledge of conditional distributions, to generate a sample of size m, that is approximately $f(x)$. Theoretically, if $m \to \infty$, we can obtain any characteristic from the function $f(x)$.

To create Figure 1, which is Figure 1 in Casella & George (1992) article, we have a two variable case with the pair, (X, Y), of random variables. To find $f(x)$, we can sample iteratively the conditional distributions of $f(x|y)$ and $f(y|x)$, thus creating the Gibbs sequence,

$$Y_0', X_0', Y_1', X_1', Y_2', X_2', \dots, Y_k', X_k', \tag{1}$$

where the only manual task is to set $Y_0' = y_0'$. The iteratively process uses the conditional distributions, to update the sequence with,

$$X_j' \sim f(x|Y_j' = y_j')$$
$$Y_{j+1}' \sim f(y|X_j' = x_j').$$

For this specific example, the joint distribution of X and Y, is a Beta-Binomial distribution. The respective conditional distributions are,

$$f(x|y) \sim Binomial(n, y)$$
$$f(y|x) \sim Beta(x + \alpha, n - x + \beta).$$

With sequences of k = 10, m = 500 sequences of observations are generated, where the last k^{th} observation is sampled as an observation in the Gibbs sample. Further, Casella & George (1992) uses the parameters n = 16, $\alpha = 2$ and $\beta = 4$ to generate this figure. The first y_0' value is randomly generated from a Beta(2,4) distribution. See Figure 1 for a approximately similar figure, due to randomness.

To estimate the density of $f(x)$, one can use a Gibbs sample, in order to estimate the density using,

$$\hat{f}(x) = \frac{1}{m} \sum_{i=1}^{m} f(x|y). \tag{2}$$

In this example, the sampled y observations from the Gibbs sequence serves as the conditional values in the equation (2), to generate Figure 2. In this figure, a valuable aspect of the Gibbs method is illustrated. By using the information generated for the Y variable, one can attain a better estimation of the marginal density of $f(x)$. This feature, Casella & George (1992) refers to as a consequence of the Rao-Blackwell theorem.

For the final Figure 5, Casella & George (1992) have introduced more than 2 variables. This is to show

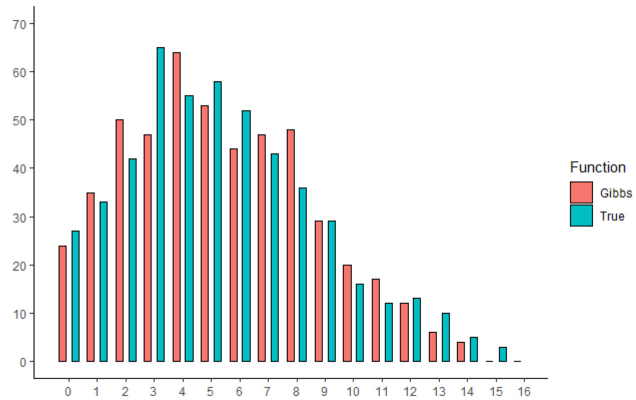

Figure 1: Comparison of two histograms of sample size 500. 'True' histogram, represents the Beta-Binomial distribution with n = 16, $\alpha = 2$ and $\beta = 4$. The 'Gibbs' histogram, represents a sample, using the Gibbs algorithm with k = 10.

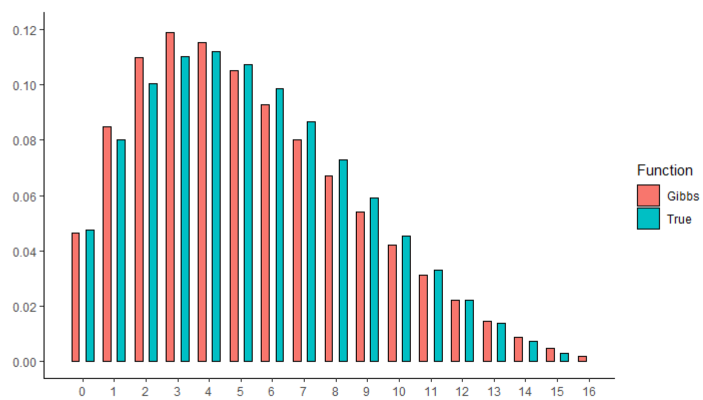

Figure 2: Comparison of two probability histograms for the Beta-Binomial density distribution. 'True' represents the actuall Beta-Binomial distribution with n = 16, $\alpha = 2$ and $\beta = 4$. 'Gibbs' represents an estimated distribution, based on equation (2). The Gibbs sequence had length k = 10. The histogram is dependent om randomness, but for this specific figure, the mean absolute error between the true and Gibbs sample is 0.0037.

how one can use *substitution sampling* to simplify more complex joint distributions, which occurs often when the dimensions increase. Following analogues that of Figure 2, we can sample from the conditional distributions where now,

$$\begin{aligned} f(x|y,n) &\sim Binomial(n,y) \\ f(y|x,n) &\sim Beta(x+\alpha, n-x+\beta) \\ f(n|x,y) &\propto x + Poisson((1-y)\lambda). \end{aligned}$$

With the iteratively sequence from equation (1), but with three variables, Figure 3 is created. The initial y_0' value is randomly generate from a Beta(2,4) distribution, and the initial n_0' value is generated from a Poisson(16) distribution. This once again estimates probabilities of the marginal distribution of X. Here Casella & George (1992) uses $\lambda = 16$, all else equal to Figure 2.

What is interesting between Figure 2 and 3, which in estimating the same marginal distribution, is the length of the right tail of the two histogram. Casella & George (1992) bring it up in a theoretical sense of the Poisson variability, which is true, but can be explained a bit more. In Figure 2, the parameter

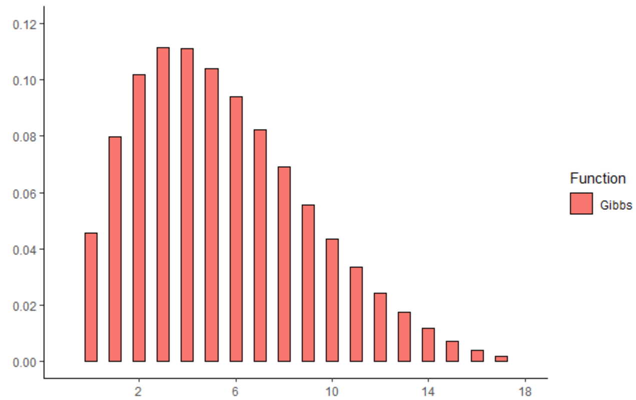

Figure 3: Histogram representing the estimated probabilities of the marginal distribution of x, using equation (2). This was generated using three conditional distributions with $\lambda = 16$, $\alpha = 2$ and $\beta = 4$.

n is fixed at 16. This creates a boundary for X in the Beta-Binomial joint distribution. X can not attain a higher value than 16 is this case. However, for Figure 3, the parameter n is a random variable which can attain any value from zero to infinity, theoretically. Hence, the marginal distribution of X with unknown parameter n, will for a large k and m value have a longer right tail.

1.2 More advanced version of Gibbs sampler

For the Gibbs sampler method, there is many ways to optimize the algorithm. *Coda* (Plummer et al. 2012) is a R-package that have many build in functions to analyze Markov Chain Monte Carlo (MCMC) objects. In order to bench how much better the algorithm will be, I have chosen to redo Figure 2 since it has a theoretical marginal density function, Beta-Binomial. Hence, comparing the Gibbs sample, which has some randomness to it, with the true density is trivial when comparing.

The process of analyzing convergence etcetera, is conducted by generating 10 Gibbs sequences of length 4000, as input in *coda*. Firstly, checking autocorrelation for time associated with sequence in the MCMC object. The MCMC suffer from high autocorrelation when looking at Table 1, thinning at Lag 10 seems reasonable. Hence, thinning the sequence by selecting only every 10^{th} observation, will increase the efficiency in the sample (Robert & Casella 2010, p. 256).

Secondly, looking in Figure 4, we can observe the convergence of the two random variables used to create Figure 3. Both of them looks to be converging to the shrink factor of 1, at around 1500 iterations. Hence removing first 1500 observations in the Gibbs sequence, will provide a more stable and accurate sample to represent the distribution (Robert & Casella 2010, p. 254).

With the knowledge of convergence and autocorrelation, a more optimized model can be estimated.

Table 1: Autocorrelation for the MCMC over lag of sequence iterations. For high autocorrelation, thinning the sequence is preferred, to increase efficiency in sample. Hence, looking at the X variable from Figure 2 model, which is of interest, Lag 10 seems reasonable to increase efficient sample.

	X	Y
Lag 0	1.0000	1.0000
Lag 1	0.7264	0.7282
Lag 5	0.1986	0.203
Lag 10	**0.0377**	0.0418
Lag 50	0.0006	-0.0006

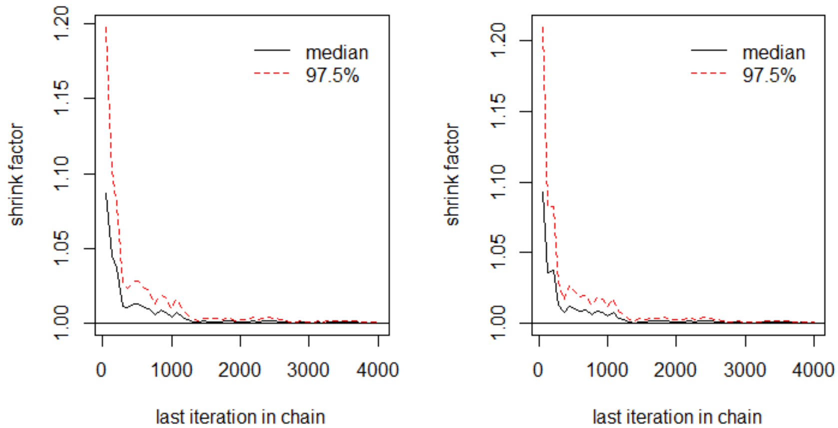

Figure 4: Evaluation of convergence for the two random variables in Figure 2. Left is X and right is Y. Noth variables are converging to the shrink factor of 1 at around 1500 iterations.

Since the sequences converge after 1500 iterations, a *burn in* of 1500 iterations will reduce variability in the sequences that represent the Gibbs sample. For more efficient sampling, every 10^{th} iteration will be sampled. Hence a *systematic sampling* method (S.S) can provide a random approach to the sampling. The S.S method picks r randomly from $1, 2, \ldots, k$ as a starting value, then select every k^{th} element, starting with r, from the Gibbs sequence up to the length of the sequence. To sample a fixed number of observations, $m = 500$ similarly as in Figure 2, the number of iterations must then be $m \times k+$burn in.

Looking at Figure 5, the same estimated marginal density has been generated as in Figure 2, but with the optimized Gibbs sampler. Visually the optimized algorithm has a closer and more smoother shape as the true distribution, than the un-optimized algorithm. Since the histograms shape are generated with randomness, it is hard to compare the two different algorithms, but the mean absolute difference between the true marginal and the estimated probability of the marginal from the Gibbs sample is smaller for the optimized (0.0024) than the un-optimized (0.0037). The un-optimized algorithm is more skewed towards smaller x values, while the optimized is almost identical to the true Beta-Binomial density distribution.

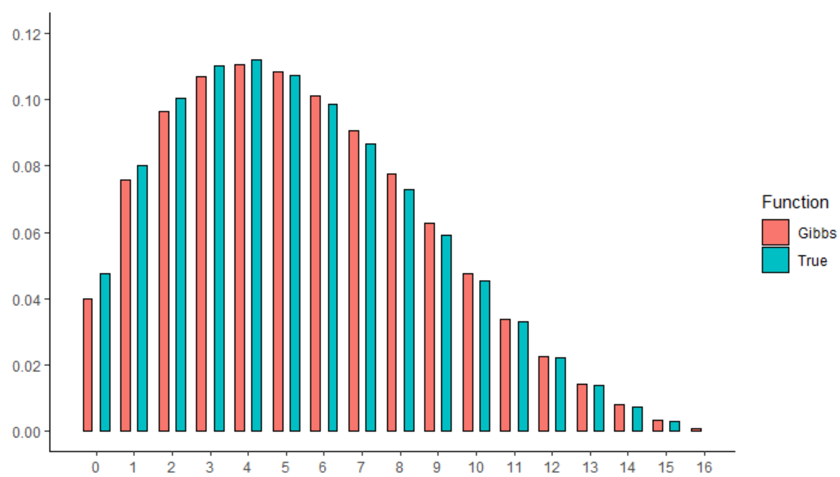

Figure 5: Histogram representing the estimated probability of the marginal distribution of x, using a optimized Gibbs sampler to recreate Figure 2. The mean absolute error between true and Gibbs sampler is 0.0024. This error is lower than in Figure 2, which uses an un-optimized algorithm.

2 Poisson Example

2.1 Description and modeling

The data set used, see Table 2, is number of bomb hits in different area district of London during a war. These bomb hits are of discrete form where each hit is independent of each other, and the variable Y is number of occurrences of these hits, distributed over the duration of the war. The total number of districts are $n = 576$ and the sum of y indicates the total number of bomb hits,

$$\sum_{i=1}^{n} y_i = 535.$$

Hence the likelihood of the data has a Poisson likelihood, with an unknown mean parameter θ which is proportional to,

$$L(y|\theta) \propto \theta^{\sum_{i=1}^{n} y_i} e^{-\theta n}.$$

In order to estimate the unknown parameter, a Bayesian posterior is to be calculated. However, calculating a posterior can be very tricky. To simplify the calculations, conjugate families can be used, in order to straight forward derive the posterior given the prior. A natural prior for θ in a Poisson likelihood is a Gamma distribution,

$$\pi(\theta) \propto \theta^{\alpha-1} e^{-\beta\theta},$$

where $\alpha - 1$ is the bomb hits count ant β is the number of observations. Given this likelihood and prior, the posterior will also be a Gamma distribution,

$$\pi(\theta|y) \propto \theta^{\alpha+\sum_{i=1}^{n} y_i - 1} e^{-(\beta+n)\theta}. \tag{3}$$

Looking at Figure 6, nine plots are displayed, showing how the prior and posterior change based on different α and β inputs, note that 0 as been changed to 0.5 since Gamma(0,0) is not defined. Visually, we can see that, increasing β will result in a narrower posterior. This would in return give a shorter credible interval, which is more precise when estimating a parameter. The α value shifts the mode of the posterior left for small values, and right for larger values.

Table 2: Data set of number of districts being hit by different number of bombs, during a war in London.

Number of bomb hits	0	1	2	3	4	5
Number of districts	229	211	93	35	7	1

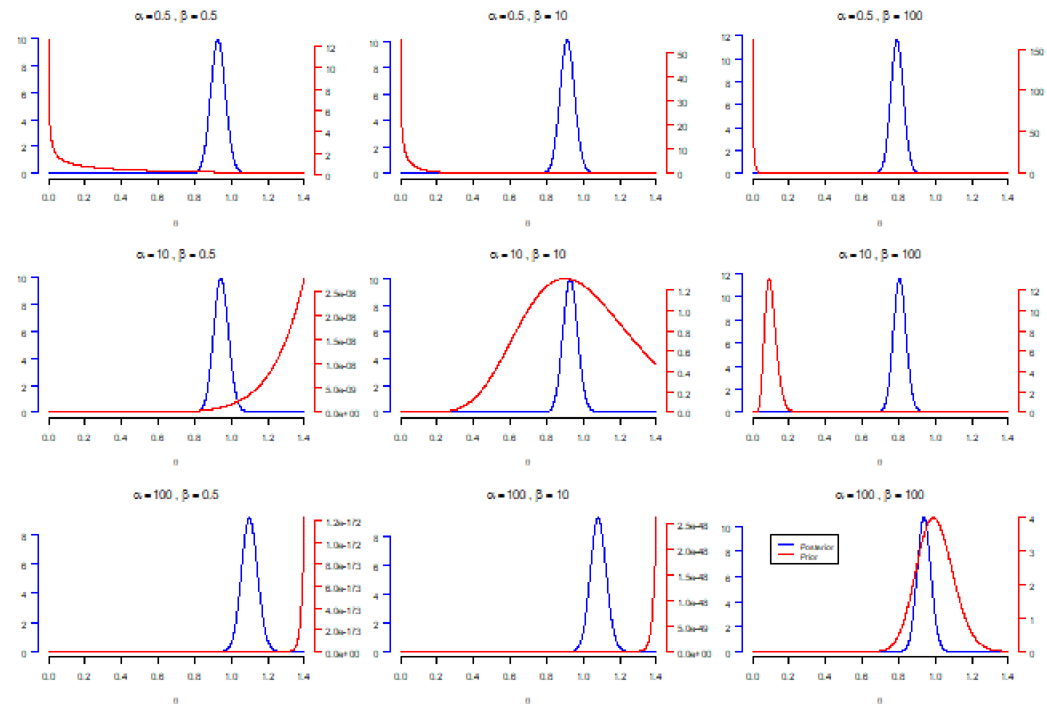

Figure 6: Comparison of different α and β values for the prior ($\theta \propto \text{Gamma}(\alpha, \beta)$) and posterior ($\theta | y_1, \ldots, y_n \propto \text{Gamma}(\alpha + \sum_{i=1}^{n} y_i, \beta + n)$) with respective scaled axis color coded; *Red* for prior and *blue* for posterior.

In this example, the assumption of independent bomb hits are made due to the restrictions of the Poisson model. However, one could argue about that assumption truly being fulfilled. Even with little knowledge of warfare, one could assume that during this war, successful bombing should encourage continuous bombing, making the bomb hits dependent on previous results within the time period, violating the assumptions of the Poisson model. Yet, to compute this example the independence is assumed to be true.

2.2 Credible intervals

For the classical (frequentist) approach, results about an estimated parameter can be expressed through a *confidence interval*. The interval is designed around the estimated parameter, to include the true parameter value, with a given certainty usually at 95%. Hence, the true parameter is said to be unknown. The interpretation is that for multiple experiments, the true parameter will be included in 95% of these experiment's confidence interval. I.e., we express that with 95% certainty, the true parameter is within a given interval.

For the Bayesian approach, in contrary to the frequentist approach, the true parameter is said to be fixed, and chosen from a probability distribution. This distribution is a combination of the a likelihood (data) and a prior belief about the parameter, which together generate a posterior distribution. The parameter of interest is treated as a random variable, which can be assigned probability for given values. This implies that a parameter can have a *credible interval*, which usually has 95% probability

Table 3: Approximate- Exact equal tail and Highest Probability Density credible intervals of the posterior equation (3), for α and β equal to zero. The values are rounded at the 4^{th} decimal.

Interval type	α	β	Lower limit	Upper limit	Length of interval
Approximate	0	0	0.8501	1.0075	0.1574
Exact	0	0	0.8518	1.0092	0.1574
HPD	0	0	0.8506	1.008	0.1573

that it include the true parameter.

In order to make any assumptions about the parameter θ, described in section 2.1, credible intervals are a good way of comparing methods which estimates parameters. Firstly, there is the exact equal tail interval. This will assign equal probability on both sides of the interval. This is calculated for α at 5% as,

$$P(a \leq \theta \leq b|y) = \int_a^b \pi(\theta|y) = 1 - \alpha = 95\% \text{ where,} \tag{4}$$

$$\int_0^a \pi(\theta|y) = 0.025\% \text{ and,} \tag{5}$$

$$\int_0^b \pi(\theta|y) = 0.975\% \tag{6}$$

meaning that there is exactly 2.5% mass probability on both sides of the interval. The approximate equal tail credible interval is similar in theory, but is calculated with the expected value and standard deviation as,

$$E(\theta|y) = \frac{\alpha + \sum_{i=1}^n y_i}{\beta + n},$$
$$SD(\theta|y) = \frac{(\alpha + \sum_{i=1}^n y_i)^{\frac{1}{2}}}{\beta + n},$$
$$CI_{95\%} = E(\theta|y) \pm 1.96 \times SD(\theta|y).$$

The advantage of the approximate credible interval is when deriving the integral in equation (4) is to complex. Lastly, there is the Highest Probability Density (HPD) credible interval. This type of interval, finds the section(s) of the posterior, where it can cover the highest density, for the shortest interval. It is a numeric integration, to find this shortest interval. Worth mentioning, is that this method can split the interval in to multiple sections if the density has multiple peaks. Hence this is a very powerful method when trying to find a parameter from a non-symmetric or multiple peak function.

The credible intervals for all types of intervals described, can be seen in Table 3. As expected, the HPD interval is shortest, however, since the posterior is a symmetrically distributed around the mode, the difference between the three methods are non substantial. Further, a visualisation of the three credible intervals are shown in Figure 7. Vertical lines indicate the lower and upper bound of the credible intervals, color coded. It is quite hard to distinguish the different intervals in the graph due to the non substantial differences between them.

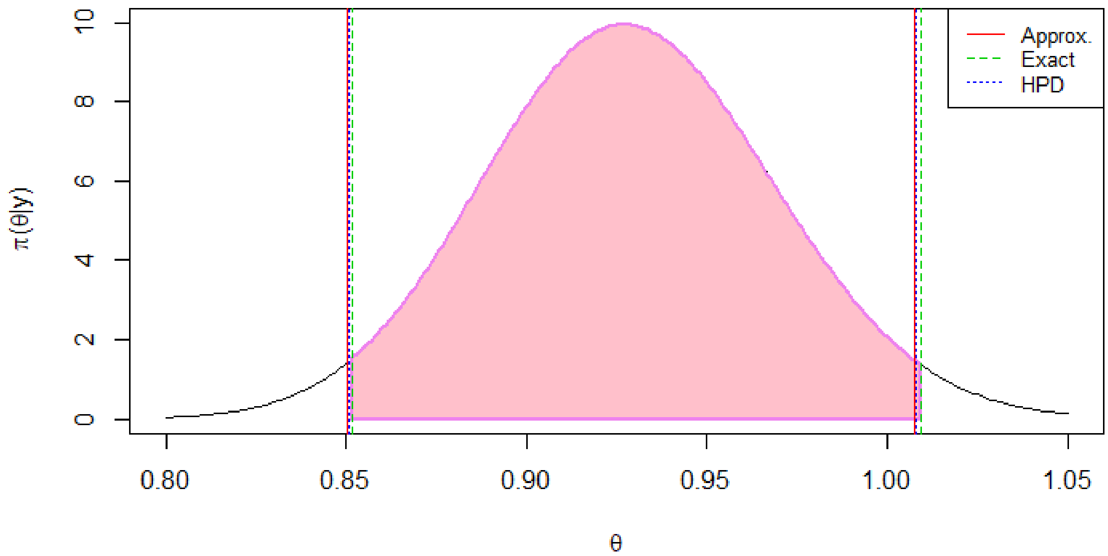

Figure 7: Illustration of three types of credible intervals; approximate- exact equal tail and highest probability density (HPD) credible interval. All intervals are very similar due to the symmetric posterior. The colored area represent all interval probability density aggregated, while the vertical lines, indicated the upper- and lower bound of the respective intervals.

3 Metropolis Hasting sampler

3.1 Reproduce figure 1

When performing Bayesian statistics, much like in section 1, problem can occur when trying to calculating complex integrals to find the posterior probability distribution. Another method, other than the Gibbs sampler, is Metropolis-Hasting algorithm which uses complimentary distributions to estimate the posterior. In fact, the Gibbs sampler is a special case of a Metropolis-Hasting sampling. Yildirim (2012) provides a brief illustration of this method, along with an example for estimating the covariance of two normally distributed variables, which have been reproduced in this paper.

The algorithm uses conditional probability, iteratively for each *proposed* observation. This proposed observation, is drawn randomly from a proposal distribution, which for best result, should have a similar distribution as the target distribution. The calculation then looks like,

$$\alpha(x^{(i)}|x^{(i-1)}) = min\{1, \frac{q(x^{(i-1)}|x^{(i)})\pi(x^i)}{q(x^{(i)}|x^{(i-1)})\pi(x^{i-1})}\}, \tag{7}$$

where q is the proposal distribution, x^i is the sampled *candidate* observation, x^{i-1} is the latest sampled observation, and π is the full joint distribution. α represents a ratio for acceptance of the observation candidate. This acceptance function is then compared with a randomly drawn Uniform(0,1), u, observation. If α is larger than u, then the candidate observation is kept as a new observation x^i, otherwise the new observation is set equal to x^{i-1}. This way, the algorithm will always have as many observations as it iterates, however, what is important is the total acceptance ratio of new observations. For one and two dimensional models, the target acceptance rate is 50%, while for more dimensions the target is 25% (Robert & Casella 2010, p. 195).

Important note about the α function, is that for a symmetric proposal distribution,

$$q(x^{(i-1)}|x^{(i)})\pi(x^i) = q(x^{(i)}|x^{(i-1)})\pi(x^{i-1}),$$

hence, the function is reduced to,

$$\alpha(x^{(i)}|x^{(i-1)}) = min\{1, \frac{\pi(x^i)}{\pi(x^{i-1})}\}. \tag{8}$$

In order to reproduce the figure in Yildirim (2012) article, a streamlined Metropolis-Hasting algorithm is created, see attached R-code. The joint distribution is defined as $x_i, y_i|\rho \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where we are interested in the parameter ρ. From Yildirim (2012) article, the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ matrix is defined as,

$$\boldsymbol{\mu} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.4 \\ 0.4 & 1 \end{pmatrix}.$$

Using the $Jeffreys$ prior (Yildirim 2012, equation 5), the likelihood of a bivariate normal, and taking the logarithm of the function to avoid underflow, our π function is,

$$-\frac{3}{2}\log(1-\rho^2) - n\log((1-\rho^2)^{\frac{1}{2}}) - \sum_{i=1}^{n} \frac{1}{2(1-\rho^2)}(x^2 - 2\rho xy + y^2)).$$

In alignment with the article, the proposal distribution is a symmetric,

$$\rho^{cand} \sim Uniform(\rho^{(i-1)} - 0.07, \rho^{(i-1)} + 0.07). \tag{9}$$

In Figure 8, the results are shown for the replication of the article. With the proposal (9) and 10000 iterations, the acceptance rate is 0.5227, which is very close to the target rate. In the middle graph of Figure 8, one can see that the algorithm quickly converged to the true parameter, and the histogram also indicates that the frequency is centered around 0.4.

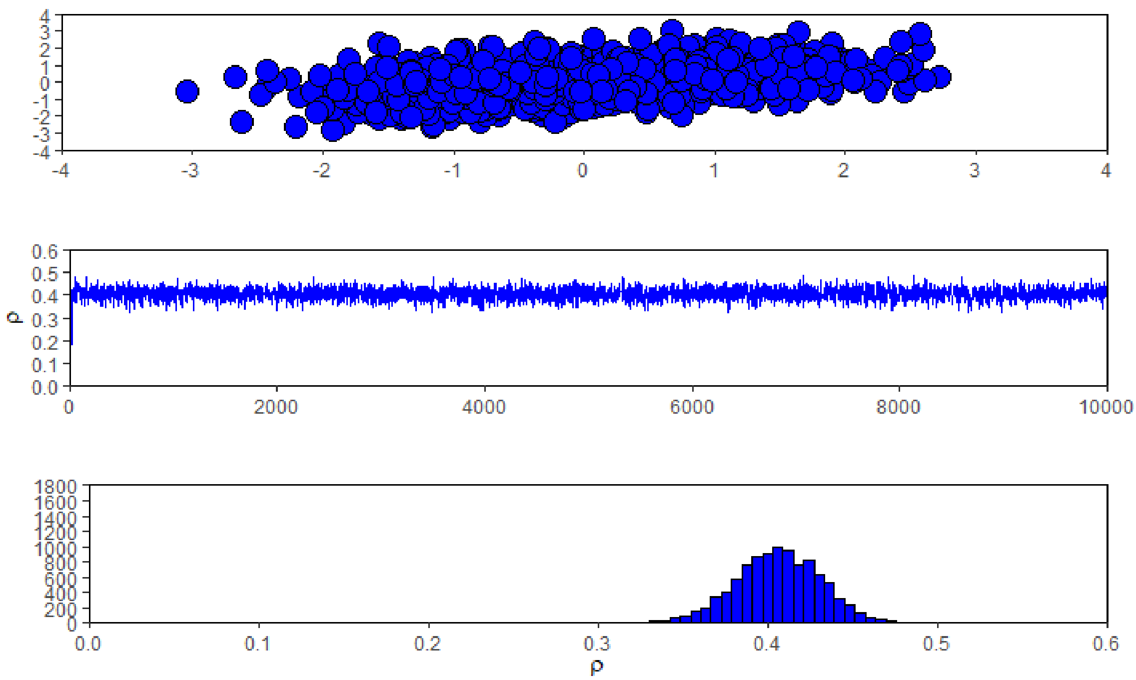

Figure 8: Top row, randomly generated populations, using R-package (mvtnorm), from Multivariate Normal Density, given in Yildirim (2012) equation (3). Middle row, a traceplot of ρ. Bottom row, histogram for the posterior of ρ, based on the Metropolis-Hasting chain. The mean ρ in the histogram is 0.4054, while ρ from the generated data is 0.3924.

3.2 Uniform, normal and non-symmetric proposal

As stated in Yildirim (2012) article, the choice of proposal distribution is not unique. However, choosing a proposal that performs well in the task of finding the true parameter, needs a preliminary thought process and exploration.

Firstly, the use of symmetrical proposals is simplifying the search of the parameter greatly, due to the reduced function form (8). Two symmetrical distributions are Uniform and Normal, which for both have the conditional aspect entered for the center of the distribution. In the Uniform case, as seen in equation (9), the candidate observation is sampled from a interval over the conditional input. Likewise, a Normal proposal would sample the candidate over the conditional input, as the mean in the distribution.

The adjustable parameter in both choices of proposal, is the width of the distribution. In the Uniform, a closed interval is derived while the Normal distribution is open, but changes shape based on the standard deviation. Hence,

$$\rho^{cand} \sim Uniform(\rho^{(i-1)} - a, \rho^{(i-1)} + a),$$
$$\rho^{cand} \sim Normal(\rho^{(i-1)}, \sigma),$$

where a and σ are fixed values that will give different outcomes of the algorithm.

A non-symmetric distribution, implies that the acceptance ratio is calculated with the full equation (7). This can become complicated, due to the assumption of a MCMC, where going from state x^{i-1} to x^i must include the possibility to go from x^i to x^{i-1}. Hence the proposal has to be defined for the same values in both positive and negative direction of the conditional value.

A skewed Normal density, is a type of non-symmetric distribution that fulfills the assumptions of a MCMC. The function works similarly to the Normal distribution, however, it also has a skewness

Table 4: Values in equation (7) that resulted in an acceptance ratio between 40% and 60% for the Uniform proposal distribution.

Acceptance ratio	Mean ρ	Sd ρ	Absolute error	Values
0.5579	0.406	0.0271	0.01360097	0.06
0.5119	0.4056	0.0273	0.01320415	**0.07**
0.4603	0.4058	0.0279	0.01339006	0.08
0.4307	0.4053	0.027	0.01293631	0.09

Table 5: Value of standard deviation that resulted in an acceptance ratio between 40% and 60% for the Normal proposal distribution.

Acceptance ratio	Mean ρ	Sd ρ	Absolute error	Values
0.5665	0.4061	0.0261	0.01365818	0.04
0.4976	0.406	0.0265	0.0135762	**0.05**
0.4346	0.4054	0.0266	0.01301498	0.06

parameter α,

$$\rho^{cand} \sim SN(\rho^{(i-1)}, \sigma, \alpha).$$

In order too choose a good prior, Uniform, Normal and the non-symmetric Skewed Normal proposal distribution have been tested for different input values. When trying different values in both the Uniform, see Table 4, and the Normal model, see Table 5, it is very obvious that the Uniform and Normal model is similar, and the value of 0.07, which is used in Yildirim (2012) article, is superior for the Uniform. Both the Uniform and Normal input parameter have been tested for values over the interval $[0, 0.5]$ by every 0.01 step. The skewness parameter, see Table 6, have been tested similarly to the other input values for Uniform and Normal proposal, but over the interval $[0, 2]$ and the standard deviation fixed at the optimum from testing the Normal proposal at 0.05.

From the Tables 4, 5 and 6, there are no substantial differences between the three proposal distributions. Based on the acceptance rate target of 50%, and the absolute error, the proposal that performed best is the Skew Normal, with standard deviation of 0.05 and skewness of 1.44. This distribution is illustrated in Figure 9, where it is visible that it is skewed towards smaller values.

Table 6: Value of skewness that resulted in an acceptance ratio between 49.95% and 50.05% for the Skew Normal proposal distribution. The interval of acceptance ration is smaller than Table 4 and 5 since almost all values gave an acceptance rate close to 50%.

Acceptance ratio	Mean ρ	Sd ρ	Absolute error	Values
0.4996	0.407	0.0261	0.01464218	0.95
0.5002	0.4063	0.0261	0.0139075	0.98
0.5001	0.406	0.0276	0.0136035	1.01
0.4999	0.4029	0.0279	0.01048055	1.13
0.5004	0.3998	0.0268	0.00743454	1.24
0.5001	0.3973	0.0284	0.00487208	**1.44**
0.4997	0.3959	0.0291	0.00348895	1.64
0.4997	0.3947	0.0312	0.00227404	1.69

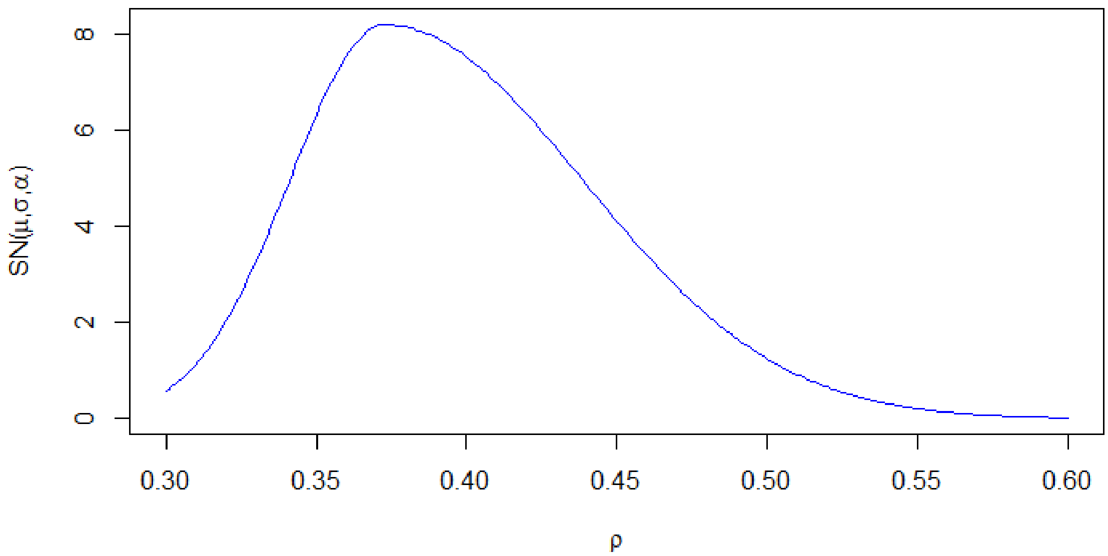

Figure 9: Function of a Skew Normal with the optimal parameters; $\mu = 0.4$, $\sigma = 0.05$ and $\alpha = 1.44$.

3.3 Three dimensional analysis

The Metropolis-Hasting algorithm can be extended to multiple dimensions. To illustrate this, an algorithm for three variable from Normal distribution, generated from

$$\boldsymbol{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \text{ where}$$
$$\boldsymbol{\mu} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$
$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.4 & 0.4 \\ 0.4 & 1 & 0.4 \\ 0.4 & 0.4 & 1 \end{pmatrix}$$

has been created, see R-code. The extensions to a multidimensional problem, is very straight forward in the Metropolis-Hasting method. Apart from the π function being different,

$$\pi(\boldsymbol{\rho}|\boldsymbol{x}) \propto \log \frac{1}{\det(\boldsymbol{\Sigma})^{\frac{4}{2}}} - \frac{n}{2} \log \det(\boldsymbol{\Sigma}) - \frac{1}{2} \sum_{i=1}^{n} \boldsymbol{x_i}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x_i},$$

the algorithm works much the same. Since there are three ρ parameters in a three dimensional case, each have individual observation candidates. For this example, once again the proposal in equation (9) is used, but for each parameter. For the acceptance calculation, all three parameters are seen as one object, i.e., either all candidates are accepted, or non of them. This is why, as stated before, the target acceptance ratio is smaller for multiple dimensions.

Looking in Figure 10, each iteration have been plotted for all three parameters. Not many iterations was necessary for the algorithm to converge to the blue line, which represents the true parameter from the data. The acceptance rate for this model is 0.201, very close to the target ratio. Since this was mainly an illustration of how the algorithm can be extended for more dimensions, only 1000 iterations was done.

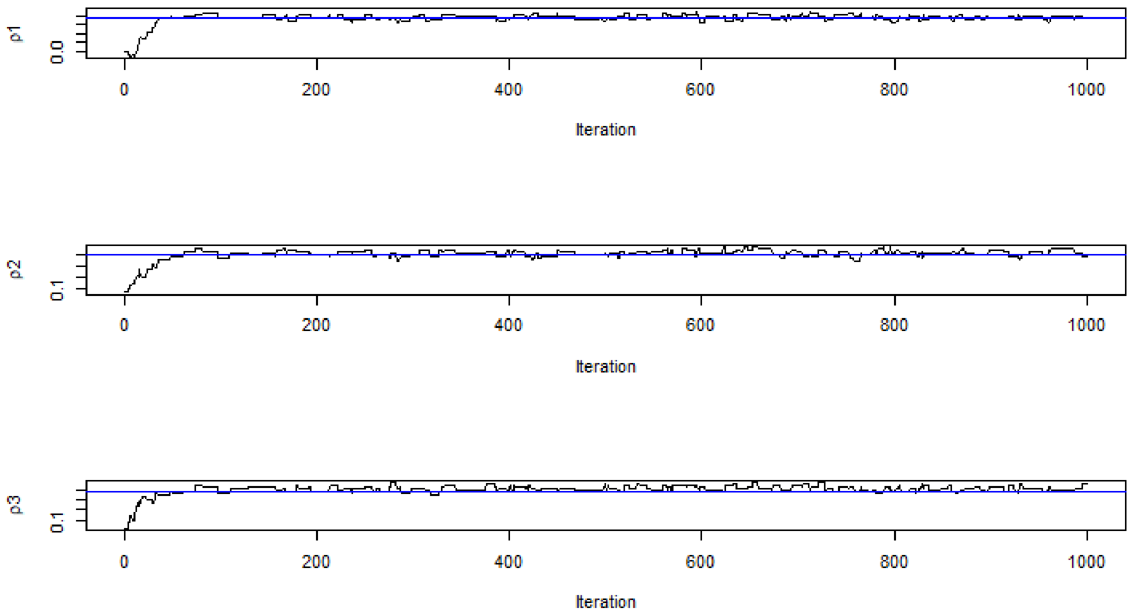

Figure 10: Trace plot of the convergence for three parameter ρ_1, ρ_2 and ρ_3 after 1000 iterations with Metropolis-Hasting algorithm. Blue line indicate true parameter value, 0.4. Proposal is Uniform$(\rho_j - 0.07, \rho_j + 0.07)$. The acceptance ratio is 0.201, close to the target of 25%.

Table 7: Values that resulted in an acceptance ratio between 15% and 30% for the Uniform proposal distribution in three dimensions, from three normal random variables.

Acceptance ratio	Mean ρ_1	True ρ_1	Mean ρ_2	True ρ_2	Mean ρ_3	True ρ_3	Value
0.236	0.3708	0.3769	0.4038	0.4038	0.3926	0.3841	**0.06**
0.174	0.3749	0.3769	0.4145	0.4038	0.3927	0.3841	0.07
0.158	0.3773	0.3769	0.408	0.4038	0.3967	0.3841	0.08

Similarly to the two variable case in section 3.2, different input values for the proposal in equation (9) have been iterated for the three variable case. Looking at Table 7, the value 0.06 has achieved a acceptance rate of 0.236, which is very close to the target. Further, this algorithm have converged very close for all parameters true value from the data, proving it's power even in multiple dimensions.

References

Casella, G. & George, E. I. (1992), 'Explaining the gibbs sampler', *The American Statistician* **46**(3), 167–174.

Plummer, M., Best, N., Cowles, K., Vines, K., Sarkar, D. & Almond, R. (2012), *Output analysis and diagnostics for MCMC*. R package version 0.19-3.
URL: *https://CRAN.R-project.org/package=coda*

Robert, C. P. & Casella, G. (2010), *Introducing Monte Carlo methods with R*, Springer, New York.

Yildirim, I. (2012), 'Bayesian inference: Metropolis-hastings sampling', *Dept. of Brain and Cognitive Sciences, Univ. of Rochester, Rochester, NY* .