1D Theta Simulation

Sakul Mahat

October 29, 2023

1 Data Generation

$$(\boldsymbol{M_i}, \boldsymbol{R_i}) = (\{\boldsymbol{M}(\boldsymbol{p_i}, \boldsymbol{\theta}), \boldsymbol{R}(\boldsymbol{p_i}, \boldsymbol{\theta})\}) + \varepsilon_i$$

Where,

$$\varepsilon_i \sim \boldsymbol{N_2}(\boldsymbol{0}, \boldsymbol{\Sigma})$$

, independently for i = 1, . . . , n , where n is the number of NICER measurements.

For $\{\boldsymbol{M}(\boldsymbol{p_i}, \boldsymbol{\theta}), \boldsymbol{R}(\boldsymbol{p_i}, \boldsymbol{\theta})\}$, consider the parametric relation: $\boldsymbol{M^2} + \boldsymbol{R^2} = \boldsymbol{\theta}$, where $\sqrt{\boldsymbol{\theta}}$ could be thought of as the radius of the circle, and $\boldsymbol{p_i}$ as the angles (latent variables). Assuming a true value for $\boldsymbol{\theta}$ and $\boldsymbol{p_i}$, the data is generated as:

```
sigma= 0.1; theta0=1.5; n= 100; p_true = runif(n,0,pi/2)
eps_m = rnorm(n,mean=0,sd=sigma)
eps_r = rnorm(n,mean=0,sd=sigma)
m = sqrt(theta0)*sin(p_true) + eps_m
r = sqrt(theta0)*cos(p_true) + eps_r
plot(m,r)
```

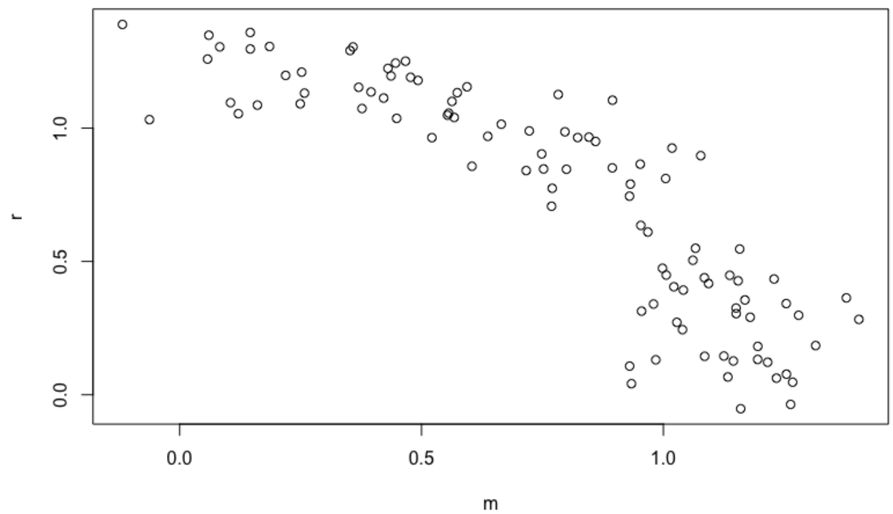

Figure 1: Data Generation

2 Bayesian Model Setup

Given the parametric equation, consider the Bayesian setup as:

$$\begin{aligned}
(\boldsymbol{M_i}, \boldsymbol{R_i}) \mid \boldsymbol{p_i}, \boldsymbol{\theta}, \boldsymbol{\Sigma} &\sim \boldsymbol{\mathcal{N}_2}\left(\{\sqrt{\boldsymbol{\theta}}\,\mathbf{sin}(\boldsymbol{p_i}), \sqrt{\boldsymbol{\theta}}\,\mathbf{cos}(\boldsymbol{p_i})\}, \boldsymbol{\Sigma}\right), \\
\boldsymbol{p_i} &\sim \text{Discrete_uniform}(\mathbf{0}, \frac{\boldsymbol{\pi}}{\mathbf{2}}), \\
\boldsymbol{\theta} &\sim \text{Discrete_uniform}(\mathbf{0}, \mathbf{5}), \\
\boldsymbol{\Sigma} &\sim \text{Inverse_Wishart}\left(\boldsymbol{\nu} = \mathbf{2}, \boldsymbol{\Psi} = \text{diag}(\mathbf{2})\right).
\end{aligned}$$

The likelihood and the full conditionals take the form:

$$\begin{aligned}
\boldsymbol{P}(\mathrm{M}, \mathrm{R} | \mathrm{p}, \boldsymbol{\theta}, \Sigma) &= \prod_{i=1}^{n} \boldsymbol{\mathcal{N}_2}\left(\{\boldsymbol{M}(\boldsymbol{p_i}, \boldsymbol{\theta}), \boldsymbol{R}(\boldsymbol{p_i}, \boldsymbol{\theta})\}, \Sigma\right), \\
\boldsymbol{P}(\boldsymbol{\theta} \mid \mathrm{M}, \mathrm{R}, \Sigma, \mathrm{p}) &\propto \boldsymbol{P}(\mathrm{M}, \mathrm{R} \mid \boldsymbol{\theta}, \Sigma, \mathrm{p}) \cdot \boldsymbol{P}(\boldsymbol{\theta}), \\
\boldsymbol{P}(\Sigma \mid \mathrm{M}, \mathrm{R}, \boldsymbol{\theta}, \mathrm{p}) &\propto \boldsymbol{P}(\mathrm{M}, \mathrm{R} \mid \boldsymbol{\theta}, \Sigma, \mathrm{p}) \cdot \boldsymbol{P}(\Sigma), \\
\boldsymbol{P}(\boldsymbol{p_i} \mid \mathrm{M}, \mathrm{R}, \boldsymbol{\theta}, \Sigma) &\propto \boldsymbol{\mathcal{N}_2}\left(\{\boldsymbol{M}(\boldsymbol{p_i}, \boldsymbol{\theta}), \boldsymbol{R}(\boldsymbol{p_i}, \boldsymbol{\theta})\}, \Sigma\right) \boldsymbol{P}(\boldsymbol{p_i}).
\end{aligned}$$

3 Gibbs Sampling Algorithm

Step 1: Define the global variables and create a discrete grid for theta & p.

```
#Global Variables:
    nreps= 500; nu=2; Psi = diag(2); N=200;
    theta_L =0; theta_U=5
    grid_theta = seq(theta_L, theta_U, length.out=N)
    grid_p=seq(0, pi/2, length.out=N)
```

Step 2: Intialize p and sigma (Draw them from the prior distribution)

```
# Storage variables:
    theta_store = numeric(nreps);
    p_store = matrix(0,nrow=nreps,ncol=n);
    Sigma_store = array(0,dim = c(2,2,nreps))
# Initialization:
    p=sample(seq(0,pi/2, length.out=N), n, replace=T);
    Sigma = rinvwishart(nu, Psi)
```

Step 3: Start the Gibbs Sampling loop and update theta first

```
 for (iter in 1:nreps)
    {
      iSigma = solve(Sigma)
      # Update theta
        theta_prob_log= mapply(function(theta){res = rbind(m - sqrt(theta)*sin(p),r - sqrt(theta)*cos(p));
          -0.5*sum(diag(t(res)%*%iSigma%*%res))},grid_theta)
       unnorm_prob = exp( theta_prob_log - max(theta_prob_log))
       norm_prob = unnorm_prob / sum(unnorm_prob)
       theta =  sample(grid_theta, 1, prob = norm_prob)
       theta_store[iter] = theta
```

Step 4: Update $\boldsymbol{p_i}$ values

```
 for (iter in 1:nreps)
 # Update p
       for (i in 1:n)
       {
         p_prob_log= mapply(function(x){res = c(m[i] - sqrt(theta)*sin(x),r[i] - sqrt(theta)*cos(x));
         -0.5*res%*%iSigma%*%res},grid_p)
         unnorm_prob = exp( p_prob_log - max(p_prob_log))
         norm_prob = unnorm_prob / sum(unnorm_prob)
         p_store[iter,i] = sample(grid_p, 1, prob = norm_prob)
```

```
        }
        p = p_store[iter,]
```

Step 5: Update $\boldsymbol{\Sigma}$ values

```
#Update Sigma
        res = rbind(m - sqrt(theta)*sin(p),r - sqrt(theta)*
    cos(p));  S <- res%*%t(res)
        # Updated parameters for inverse Wishart
        nu_star =nu + n; Psi_star <- Psi + S
        Sigma = rinvwishart(nu_star, Psi_star)
        Sigma_store[,,iter] = Sigma
```

4 Simulation Diagnosis / Plots

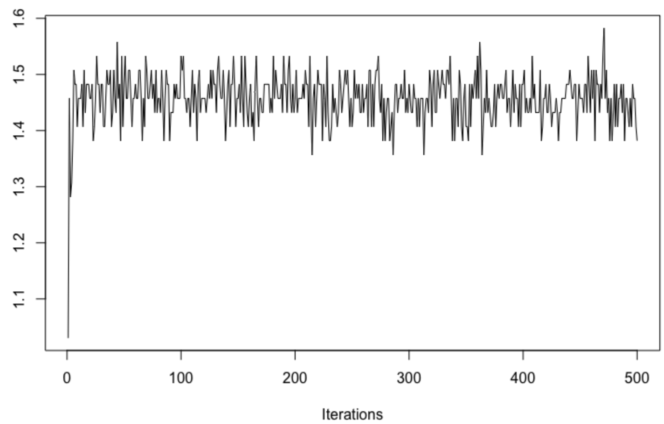

Figure 2: Trace for $\boldsymbol{\theta}$

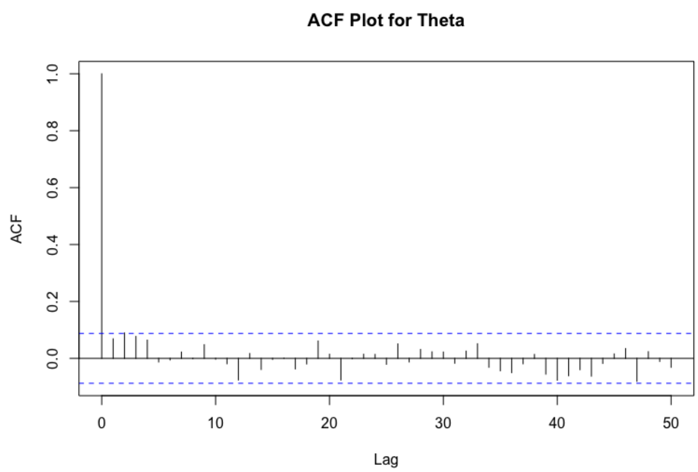

Figure 3: ACF for $\boldsymbol{\theta}$

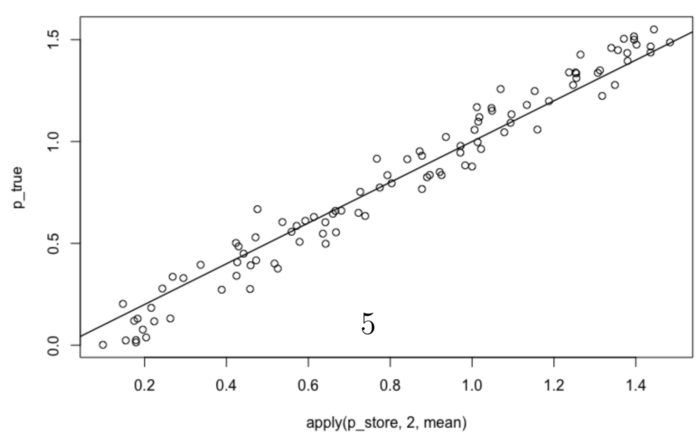

Figure 4: $\boldsymbol{p_i}$ estimates