

presentation

Kaiden Liu

28/07/2021

Bayesian Idea

- 1) prior: $p(\text{parameter values}) - p(\theta)$
- 2) likelihood: $p(\text{data values} \mid \text{parameter values}) - p(D \mid \theta)$
- 3) posterior: $p(\text{parameter values} \mid \text{data values}) - p(\theta \mid D)$

Bayesian Idea Cont.

$$\begin{aligned}p(D, \theta) &= p(\theta) \cdot p(D|\theta) \\ &= p(D) \cdot p(\theta|D)\end{aligned}$$

$$p(D) \cdot p(\theta|D) = p(\theta) \cdot p(D|\theta)$$

We can conclude that:

$$p(\theta|D) = \frac{p(\theta) \cdot p(D|\theta)}{p(D)}$$

$$p(\theta|D) \propto p(\theta) \cdot p(D|\theta)$$

(Shape of) posterior = prior · likelihood

1) Grid approximation

- 1) We want to propose an interval that we use to evaluate the posterior with even-spaced points.
- 2) Compute prior and likelihood at each point
- 3) Compute the posterior

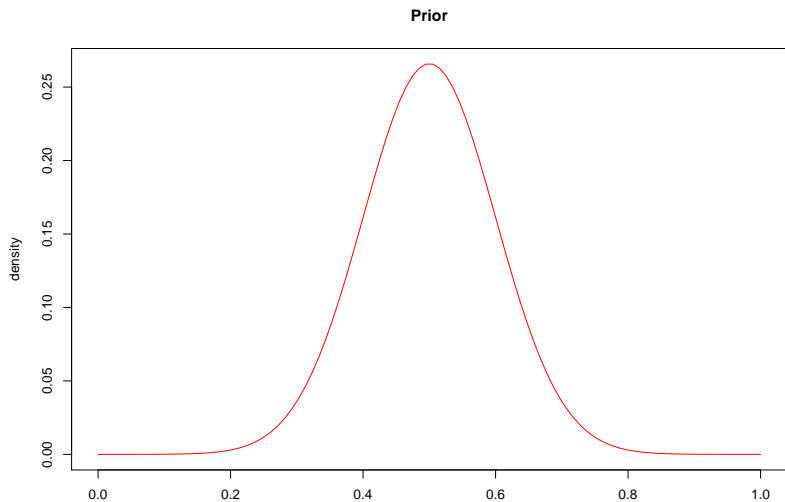
Example 1)

Prior: $P(T) \sim \text{Normal}(0.5, 0.1)$

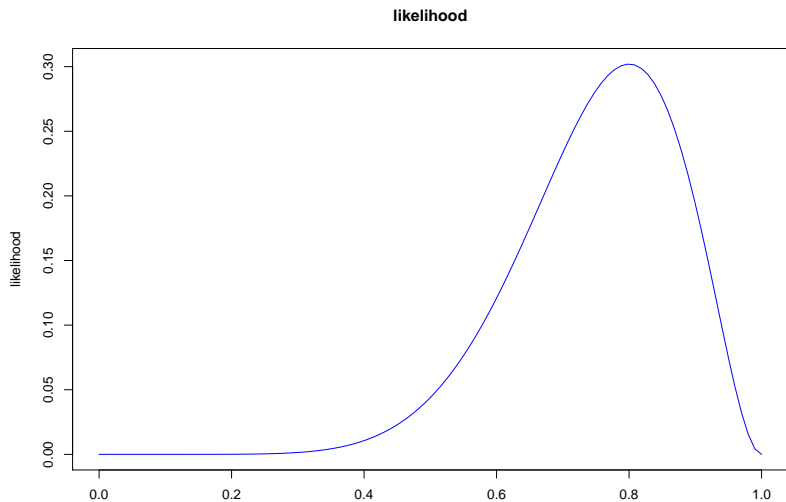
likelihood: $P(X \sim \text{Binomial}(10, P(T)) = 8)$

```
rangeP <- seq(0, 1, length.out = 100)
prior<- dnorm(x = rangeP, mean = .5, sd = .1)
likelihood <- dbinom(x = 8, prob = rangeP, size = 10)
posterior <- likelihood * prior
```

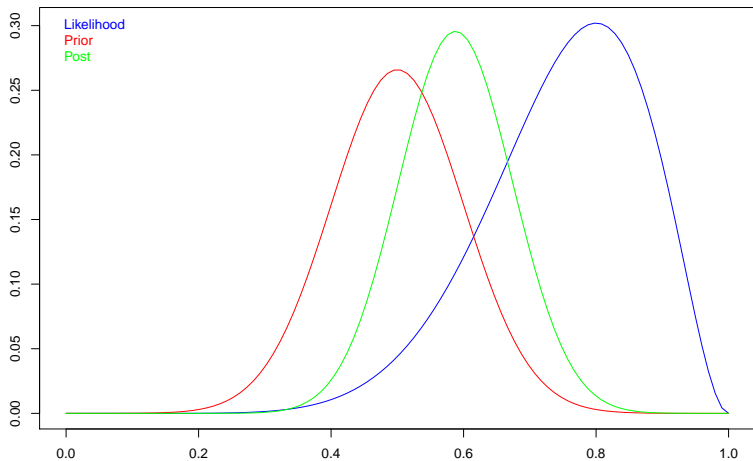
Example 1) - Prior



Example 1) - Likelihood

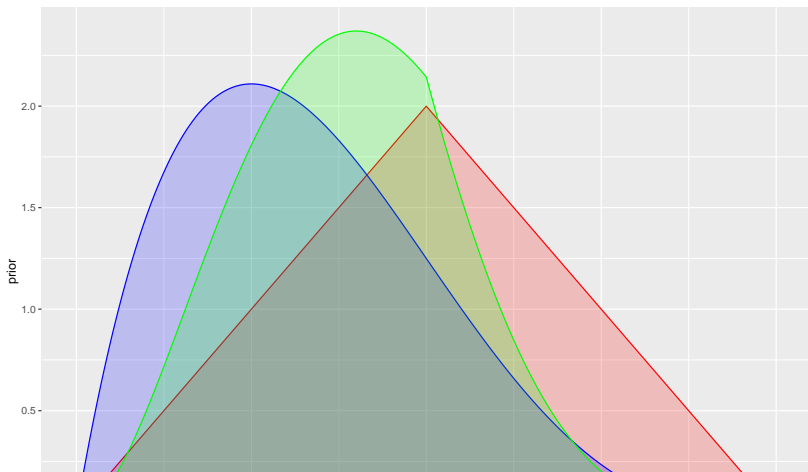


Posterior



Graphs

```
ggplot(example) +  
  geom_area(data=example,aes(x=theta,y=prior),alpha=0.2, fill="blue") +  
  geom_area(data=example,aes(x=theta,y=likelihood1),alpha=0.2, fill="green") +  
  geom_area(data=example,aes(x=theta,y=posterior),alpha=0.2, fill="red")
```



Grid Approximation - Limitation

Take a sample of n observations:

Prior:

$$\mu \sim \text{Normal}(0, 5)$$

$$\sigma \sim \text{Exp}(1)$$

Likelihood:

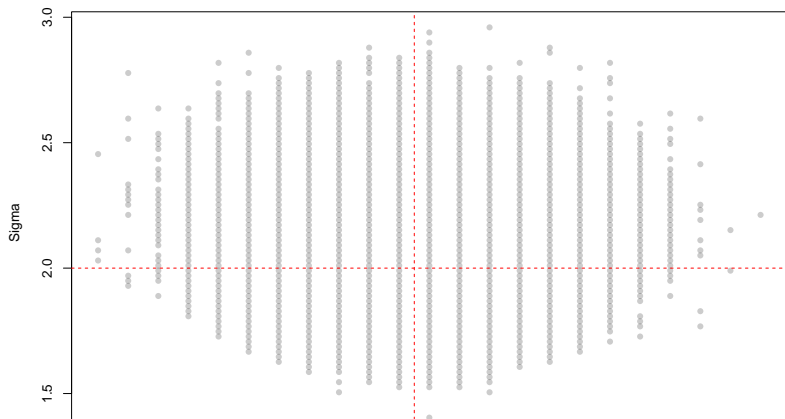
$$X \sim \text{Normal}(\mu, \sigma)$$

by computing the likelihood for every $n \times n$ combination (grid approximation)

$$\text{True}\mu = 5 \quad \text{True}\Sigma = 2$$

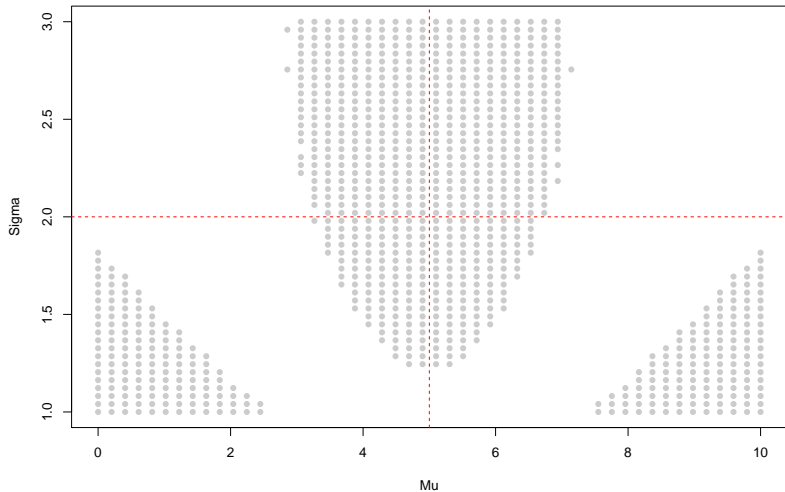
Limitation

```
trueMu <- 5  
trueSig <- 2  
  
grid_limit(100)
```



Limitation

```
grid_limit(50)
```



Example

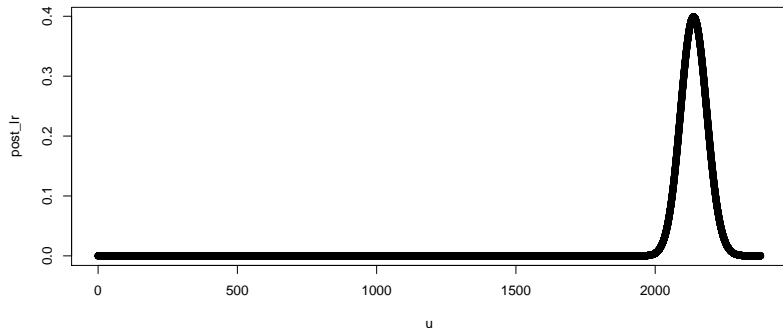
In a real scenario, we would compute some initial point estimates to choose an interval which should contain almost all of the probability mass of the posterior distribution.

Example

```
grid <- rgamma(n = 1000, shape = al_c, rate = be_c)
maxu <- max(grid)
minu <- min(grid)
u <- seq(minu, maxu, by = .1)

post_lr <- posterior(u, k, al_c, be_c)
#post_lr <- lapply(as.list(u), function(j){posterior(j, k, al_c, be_c)}) %>% unlist()

plot(u, post_lr)
```



2) Rejection Sampling

Sample data from a complicated distribution

- ▶ Target (distribution) function $f(x)$ — The “difficult to sample from” distribution. Our distribution of interest!
- ▶ Proposal (distribution) function $g(x)$ — The proxy distribution from which we can sample.

1)

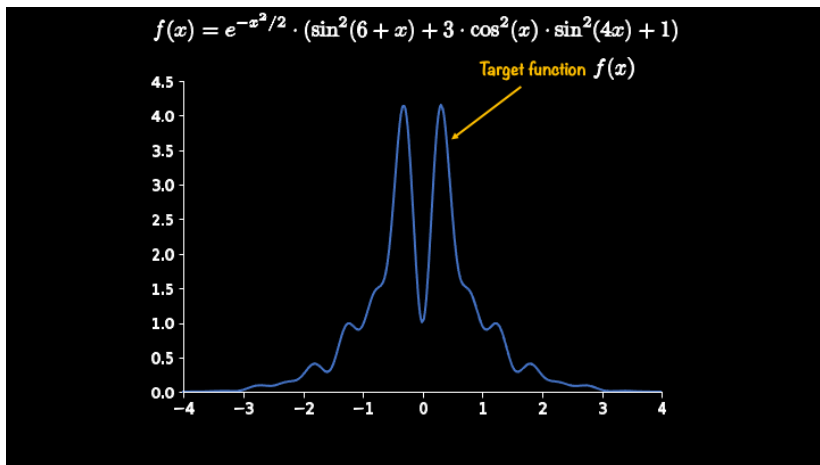


Figure 1: Target Function

2)

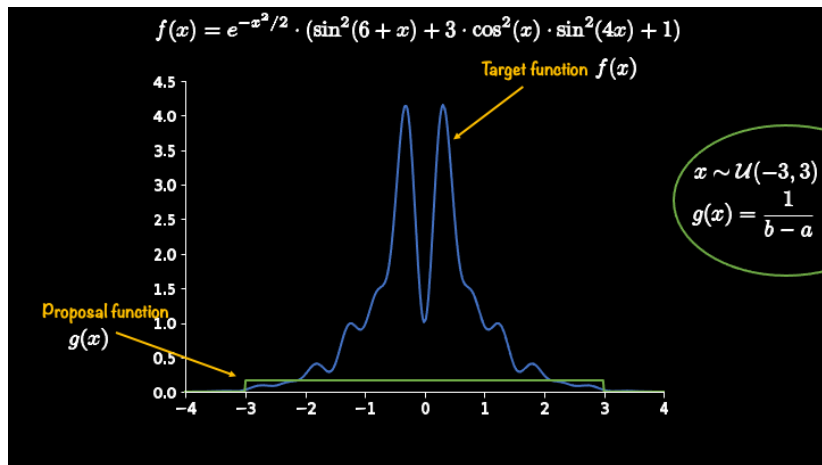


Figure 2: Proposal Function

3)

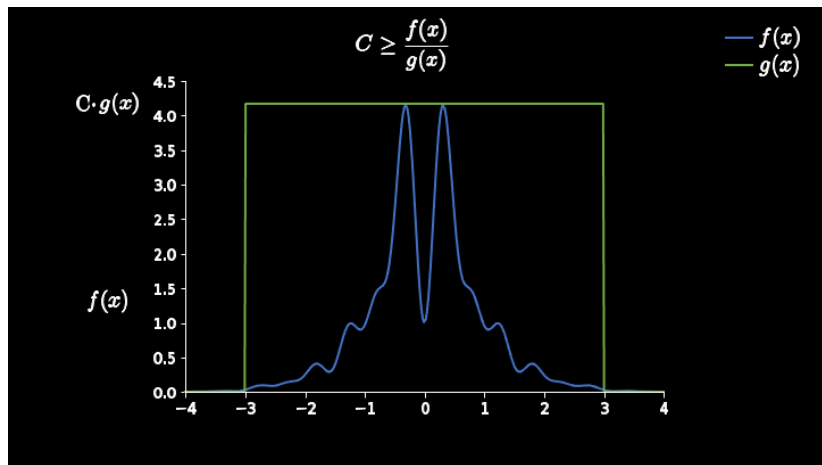
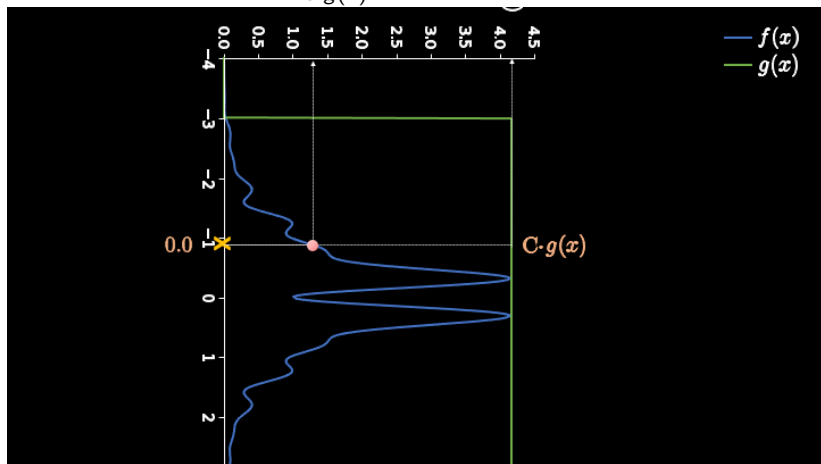


Figure 3: Constant

4)

Accept with probability $\frac{f(x)}{C \cdot g(x)}$



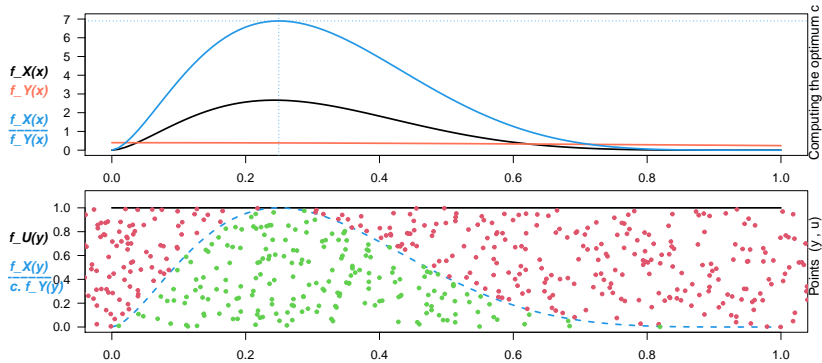
Example

```
blue is ratio
library(AR)

simulation = AR.Sim( n = 200,
  f_X = function(y){dbeta(y,2.7,6.3)},
  Y.dist = "norm", Y.dist.par = c(0,1),
  Rej.Num = TRUE,
  Rej.Rate = TRUE,
  Acc.Rate = FALSE
)
```

```
## Optimal c = 6.898
```

Graphical Presentation to Acceptance-Rejection Method



Output

n

The number/length of data which must be generated/simulated from $(f_X)(\text{TARGET})$ density.

Optimal $c = 6.898$

The numbers of Rejections = 1295

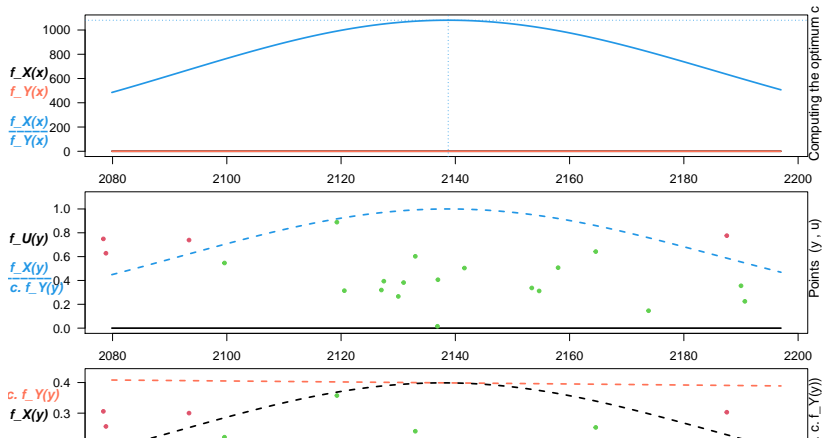
Ratio of Rejections = 0.866

Example_DNA

```
lr_rs = AR.Sim( n = 20,
  f_X = function(y){(posterior(y,as.numeric(gammaPrior_Cont[[sam]]$kij[taxa]), al_c,be_c))},
  Y.dist = "gamma", Y.dist.par = c((k/mean(lc))*al_c,be_c), xlim=c(k-2*mean(lc),k),
  Rej.Num = TRUE,
  Rej.Rate = TRUE,
  Acc.Rate = FALSE
)
```

Optimal $c = 1080.855$

Graphical Presentation to Acceptance-Rejection Method



Limitation

- ▶ Selecting the appropriate proposal function & finding its scaling constant
- ▶ Requires that the PDF of the target function is known
- ▶ Generally inefficient especially in higher dimensions

2.5) Adaptive Rejection Sampling

Define our proposal distribution in log space

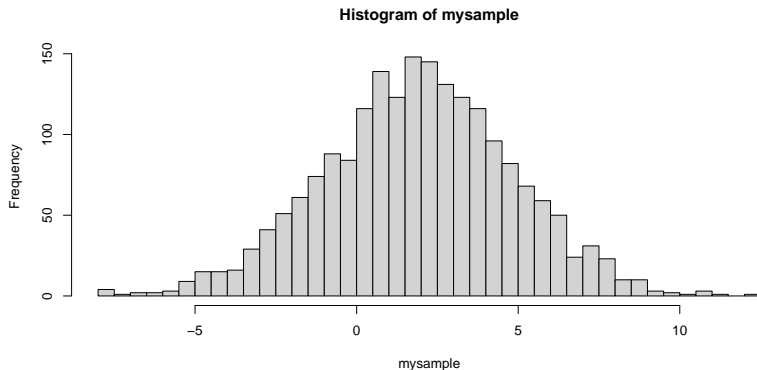
1

¹Gilks, W. R., & Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 41(2), 337-348.

Example

sample 2000 values from the normal distribution $N(2,3)$

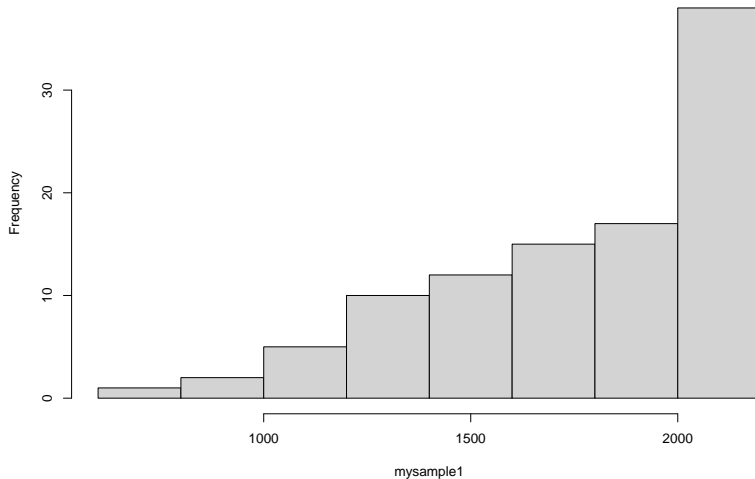
```
library("ars")  
  
f<-function(x,mu=0,sigma=1){-1/(2*sigma^2)*(x-mu)^2}  
fprima<-function(x,mu=0,sigma=1){-1/sigma^2*(x-mu)}  
mysample<-ars(2000,f,fprima,mu=2,sigma=3)  
hist(mysample, breaks=30)
```



Example

```
f1<-function(x,shape=al_c,scale=(1/be_c)){-1*((shape-1)*log(x)-x/scale)}  
f1prima<-function(x,shape=al_c,scale=(1/be_c)){-1*((shape-1)/x-1/scale)}  
mysample1<-ars(100,f1,f1prima,x=1,m=1,lb=TRUE,xlb=0,shape=al_c,scale=1/be_c, ub=TRUE, xub =max(k) )  
hist(mysample1)
```

Histogram of mysample1

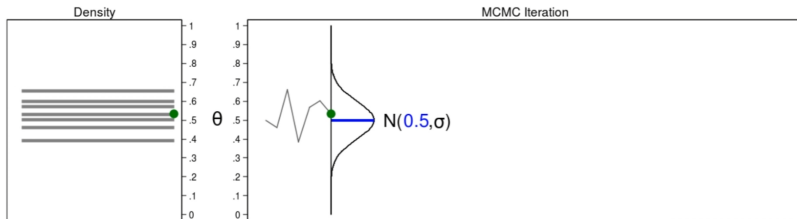


Monte Carlo

Relies on repeated random sampling to obtain numerical result

Ex) $\theta_t \sim \text{Normal}(0.5, \sigma)$

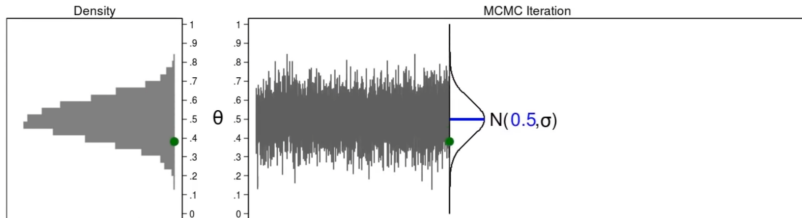
Monte Carlo Trace Plot



Draw $\theta_t \sim \text{Normal}(0.5, \sigma) = 0.534$

Figure 4: Trace Plot

50000 iterations



Draw $\theta_t \sim \text{Normal}(0.5, \sigma) = 0.380$

Figure 5: 50000

Markov property

Given the present, the future does not depend on the past.

$$P(X_n = x_n \mid X_{n-1} = x_{n-1}, \dots, X_0 = x_0) = P(X_n = x_n \mid X_{n-1} = x_{n-1}).$$

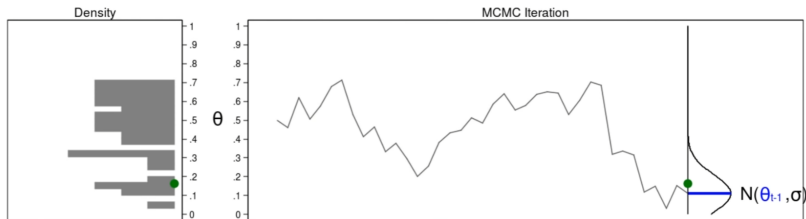
Figure 6: markovproperty

Example

Ex) $\theta_t \sim \text{Normal}(\theta_{t-1}, \sigma)$

Depends on the previous number on a sequence

Trace Plot

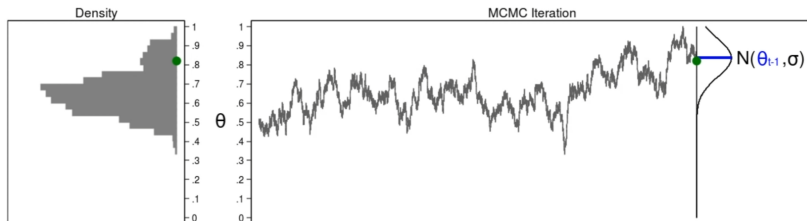


Draw $\theta_t \sim \text{Normal}(\theta_{t-1}, \sigma)$

$\text{Normal}(0.111, \sigma) = 0.164$

Figure 7: Trace Plot

50000 entries



Draw $\theta_t \sim \text{Normal}(\theta_{t-1}, \sigma)$

$\text{Normal}(0.836, \sigma) = 0.820$

3) Metropolis Hasting

The Metropolis–Hastings algorithm can draw samples from any probability distribution $f(x)$, provided that we know a function $q(x)$ proportional to the density of f and the values of $q(x)$ can be calculated. The requirement that $q(x)$ must only be proportional to the density

1. Compute $\rho(x, y) = \min \left\{ \frac{f(y)}{f(x)} \times \frac{q(x|y)}{q(y|x)}, 1 \right\}$
2. If $\rho(x, y) < 1$ then generate $U_t \sim \text{Uniform}(0, 1)$.
3. Set $X^{(t+1)} = \begin{cases} y & \text{if } \rho(x, y) = 1 \text{ or } U_t < \rho(x, y) \\ x & \text{if } U_t \geq \rho(x, Y_t) \end{cases}$

Figure 8: mh

Intuition

$$\alpha = \min\left\{\frac{f(b)}{f(a)}, 1\right\}$$

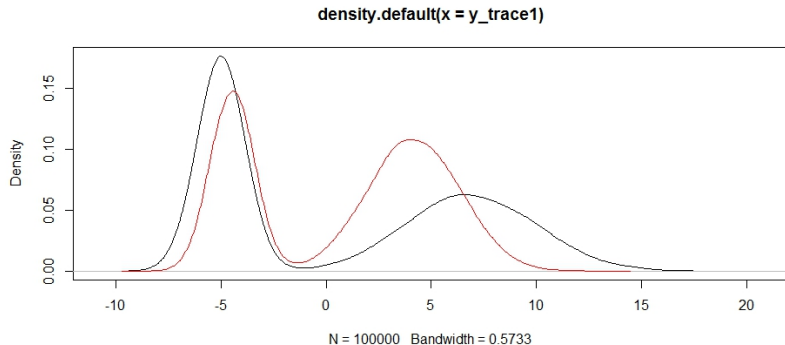


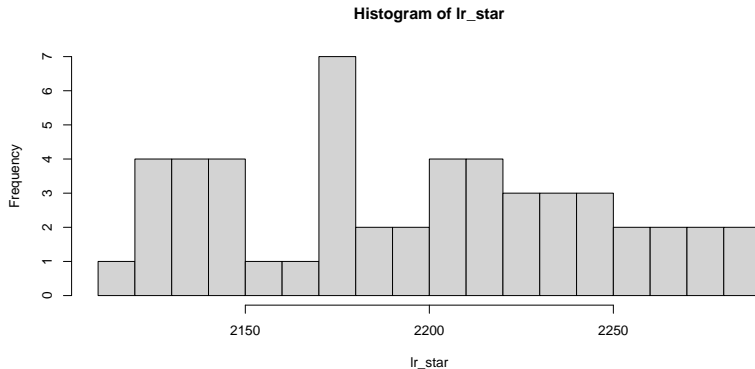
Figure 9: intuition

Limitation

- ▶ Dependence on starting value
 - ▶ Burn-in period
- ▶ Autocorrelation due to the Markov Chain properties

Example

```
chain <- MH_MCMC(itera = 100,  
  k = as.numeric(gammaPrior_Cont[[sam]]$kij[taxa]),  
  al_c = gammaPrior_Cont[[sam]]$alpha_ij_c[taxa],  
  be_c = gammaPrior_Cont[[sam]]$beta_ij_c[taxa],  
  startvalue_lamda_r = 0)  
  
lr_star <- chain[50:100]  
hist(lr_star, breaks=18)
```



Gibbs Sampling

1. Initialize the chain at $X^{(0)}$.
2. At iteration $t = 1, 2, \dots$
 - (a) Generate $Y^{(t)}$ from $f_{Y|X} (y | X^{(t-1)})$
 - (b) Generate $X^{(t)}$ from $f_{X|Y} (x | Y^{(t)})$

Limitation

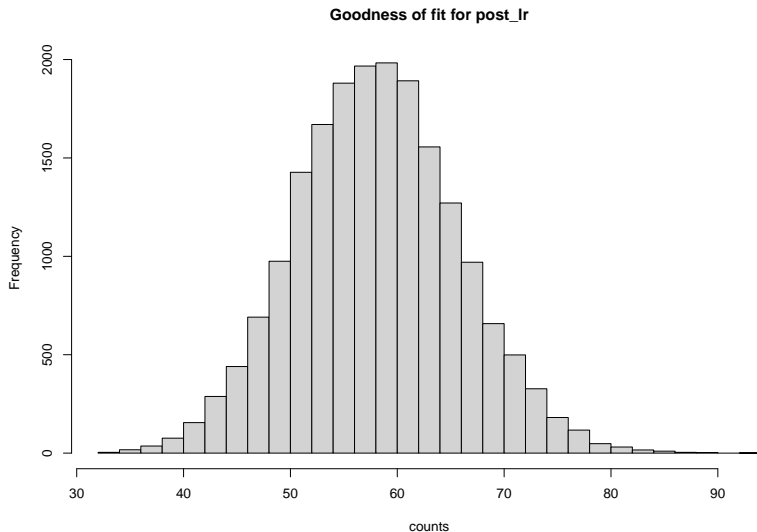
False Convergence

Goodness of fit

```
goodness_of_fit <- function(lc,lr){  
  kij_c_star <- rpois(length(lc),mean(lc))  
  kij_r_star <- rpois(length(lc),mean(lr))  
  kij_star <- kij_r_star+kij_c_star  
  hist(kij_star, main=paste("Goodness of fit for",deparse(substitute(lr)  
    abline(v=as.numeric(gammaPrior_Cont[[sam]]$kij[taxa])))  
}
```

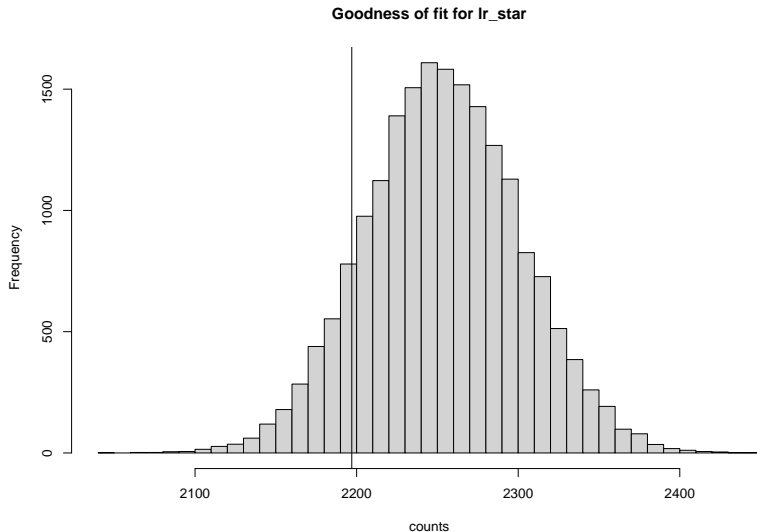
Grid Approximation

```
goodness_of_fit(lc,post_lr)
```



Metropolis-Hasting

```
goodness_of_fit(lc,lr_star)
```



Rejection Sampling

```
goodness_of_fit(lc,lr_rs)
```

