Statistical Inference

Lecture 12b

ANU - RSFAS

Last Updated: Tue May 22 12:40:52 2018

Sampling & Bayesian Inference

We are interested in modeling data where:

$= \frac{1}{\sigma^2}$

$$X_1, \ldots, X_2 \overset{\text{iid}}{\sim} \text{normal}(\theta, \xi)$$

precision

$$f_X(x \middle| \theta, \xi) = \left(\frac{\xi}{2\pi}\right)^{1/2} \exp\left(-\frac{1}{2}\xi(x-\theta)^2\right)$$

Where $\xi = \frac{1}{\sigma^2}$. As we are considering Bayesian inference, we need to have priors on both parameters (which are considered random in this framework). Here we will model the priors as being independent.

$$p(\theta, \xi) = p(\theta)p(\xi)$$

The prior for θ is:

$$\theta \sim \text{normal}(\theta_0, \tau_0)$$

and the prior for ξ is:

$$\xi \sim \text{gamma}(\alpha_0, \lambda_0)$$

Metropolis-Hastings

- Here we have the following:

$$p(\theta, \xi | \boldsymbol{x}) \propto p(\boldsymbol{y} | \theta, \xi) p(\theta) p(\xi)$$

- In a Metropolis-Hastings sampling scheme:
  - we propose a new value of θ, say θ^*, and decide to accept or reject.
  - we propose a new value of ξ, say ξ^*, and decide to accept or reject.

Metropolis-Hastings - Update θ

- $\theta^* \sim norm(\theta, \delta_1)$ - Symmetric proposal.

$$\rho = \frac{p(\boldsymbol{y}|\theta^*, \xi)p(\theta^*)p(\xi)}{p(\boldsymbol{y}|\theta, \xi)p(\theta)p(\xi)} = \frac{p(\boldsymbol{y}|\theta^*, \xi)p(\theta^*)}{p(\boldsymbol{y}|\theta, \xi)p(\theta)}$$

Metropolis-Hastings - Update ~~θ~~ ξ

$\Rightarrow \xi'' = |\xi^*|$

abs

- $\xi^* \sim unif(\xi - \delta_2, \xi + \delta_2)$. If $\xi^* < 0$ then reflect the value to the positive line.

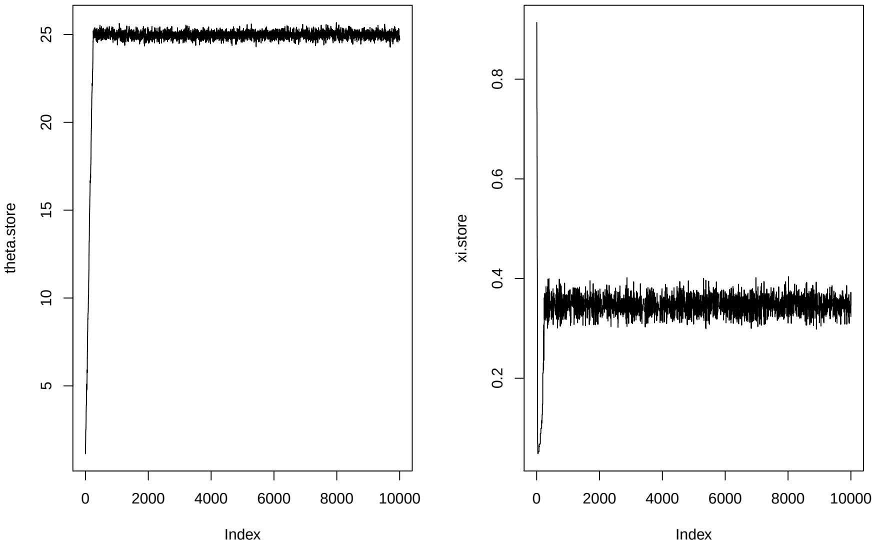
$$\rho = \frac{p(\boldsymbol{y}|\theta, \xi^*)p(\theta)p(\xi^*)}{p(\boldsymbol{y}|\theta, \xi)p(\theta)p(\xi)} = \frac{p(\boldsymbol{y}|\theta, \xi^*)p(\xi^*)}{p(\boldsymbol{y}|\theta, \xi)p(\xi)}$$

Metropolis-Hastings - GDP Data

```
x <- read.csv("gdp2013.csv")
x <- log(na.omit(x$X2013))
n <- length(x)

## Prior values
theta.0 <- 0
tau.0 <- 0.001
alpha.0 <- lambda.0 <- 1

## Storage
theta.store <- NULL
xi.store <- NULL


## Starting values
theta <- 1
xi <- 1
```

```
## Start the chain
S <- 10000
for(s in 1:S){

    ##                                    = δ1
    theta.star <- rnorm(1, theta, 0.25)

    log.r <- sum(dnorm(x, theta.star, 1/xi, log=TRUE)) +
      dnorm(theta.star, theta.0, 1/tau.0, log=TRUE) -
      sum(dnorm(x, theta, 1/xi, log=TRUE)) -
      dnorm(theta, theta.0, 1/tau.0, log=TRUE)


    if(runif(1) < exp(log.r)){
     theta <- theta.star
    }
```

```
##
    xi.star <- runif(1, xi-0.1, xi+0.1)
    if(xi.star<0){xi.star <- abs(xi.star)}

    log.r <- sum(dnorm(x, theta, 1/xi.star, log=TRUE)) +
      dgamma(xi.star, alpha.0, lambda.0, log=TRUE) -
      sum(dnorm(x, theta, 1/xi, log=TRUE)) -
      dgamma(xi, alpha.0, lambda.0, log=TRUE)


    if(runif(1) < exp(log.r)){
     xi <- xi.star
    }

##
    theta.store <- c(theta.store, theta)
    xi.store <- c(xi.store, xi)
    }
```


theta.store
25
20
15
10
5
0
2000
4000
6000
8000
10000
Index
xi.store
0.8
0.6
0.4
0.2
0
2000
4000
6000
8000
10000
Index

- Remove the first 999 for burn-in.

```
par(mfrow=c(1,2))

hist(theta.store[-c(1:999)], col="cyan4")
abline(v=quantile(theta.store[-c(1:999)], c(0.025, 0.5, 0.975)), lwd=3)

hist(xi.store[-c(1:999)], col="orange")
abline(v=quantile(theta.store[-c(1:999)], c(0.025, 0.5, 0.975)), lwd=3)
```

- Remove the first 999 for burn-in.