Bayesian linear Regression

Remy

1/9/2020

#Introduction Consider the following linear regression:
$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon$, which can be written as:

$$y_i = \boldsymbol{x_i^T \beta} + \epsilon_i$$

or

$$\boldsymbol{y = X\beta + \epsilon}$$

$\epsilon_i \sim N(0, \sigma^2) \longleftrightarrow \boldsymbol{\epsilon} \sim N(0, \sigma^2 I_n)$

#Normal likelihood $y_i \sim N(x_i^T \beta, \sigma^2)$
$\boldsymbol{y} \sim N(\boldsymbol{X\beta, \sigma^2 I_n})$
Therefore, the distribution of the date is a multivariate normal with density

$$[y_i|\beta, \sigma^2] \propto (\sigma^2)^{-1/2} \exp\left\{-\frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - x_i^T\beta)\right\}$$

Which is equivalent to

$$[\boldsymbol{y}|\boldsymbol{\beta, \sigma^2}] \propto (\sigma^2)^{-n/2} \exp\left\{\boldsymbol{(y - X\beta)^T \Sigma^{-1} (y - X\beta)}\right\}$$

with $\Sigma = \sigma^2 I_n$

#Prior and posterior distributions Given a normal likelihood, if we consider a prior for the normal variance $\sigma^2 \sim IG(a, b)$ to be an inverse gamma, and place a normal or a noninformative prior on the linear regression coefficients $\boldsymbol{\beta} = \beta_0, \beta_1, \ldots, \beta_p$, then the we obtain full conditional posteriors

- $\boldsymbol{\beta} \sim N(\boldsymbol{\mu_0, \Sigma_0})$
- Given the data: $\boldsymbol{y} \sim N(\boldsymbol{X\beta, \sigma^2 I_n})$
- $\boldsymbol{\beta|y, \sigma^2} \sim N(\boldsymbol{\mu_n, \Sigma_n})$
- $\sigma|\alpha, \gamma \sim IG(a, b = rate)$

Considering an noninformative prior on $\boldsymbol{\beta}$, we can use the ordinary least square results:

- $\boldsymbol{\hat{\beta} = (X^T X)^{-1} X^T y}$,
- $var(\boldsymbol{\hat{\beta}_{ols}}) = \sigma^2 (X^T X)^{-1}$.

It can be seen that $\boldsymbol{\hat{\beta}} = \underbrace{\boldsymbol{(X^T X)^{-1} X^T}}_{\boldsymbol{constant}} \boldsymbol{y}$ is a linear combination of multivariate normal random variables, which means that $\boldsymbol{\hat{\beta}}$ will also have a normal distribution. Therefore

$$\boldsymbol{\hat{\beta} \sim N(\beta, \sigma^2 (X^T X)^{-1})}$$

or

$$\boldsymbol{\beta \sim N(\hat{\beta}, \sigma^2 (X^T X)^{-1})}$$

Thus, $\boldsymbol{\hat{\beta}}$ is randomly distributed around its least square estimate with $\sigma^2 (X^T X)^{-1}$ determining its randomness.

Posterior for σ^2

Given $\boldsymbol{\beta}$, the posterior for σ^2 can be calculated as follows:
$\sigma^2|\boldsymbol{y},\boldsymbol{\beta} \sim N(\boldsymbol{X\beta},\boldsymbol{\sigma^2 I_n})IG(a,b)$, where

- $a_n = \frac{n}{2} + a$ and
- $b_n = \frac{1}{2\sigma^2}\boldsymbol{(y - X\beta)^T(y - X\beta)} + b$

Posterior for $\boldsymbol{\beta}$ with a normal prior

$\beta|\boldsymbol{y},\boldsymbol{\sigma^2} \sim N(\mu_0,\Sigma_0)N(\boldsymbol{X\beta},\boldsymbol{\sigma^2 I_n}) \sim N(\boldsymbol{\mu_n},\boldsymbol{\Sigma_n})$, with

- $\boldsymbol{\Sigma_n} = \left(\boldsymbol{\Sigma_0^{-1} + (\sigma^2)^{-1}(X^T X)}\right)^{-1}$
- $\boldsymbol{\mu_n = \Sigma_n \left(\Sigma_0^{-1}\mu_0 + (\sigma^2)^{-1}(X^T X)\hat{\beta}\right)^{-1}}$

Generally, the prior for $\boldsymbol{\beta}$ is $N(\boldsymbol{0},\boldsymbol{\Sigma_0})$, which results in

$\boldsymbol{\mu_n = \Sigma_n} \left[(\sigma^2)^{-1}\boldsymbol{(X^T X)(X^T X)^{-1}X^T y}\right]^{-1}$.
Hence,

$$\boldsymbol{\mu_n = \Sigma_n}\left(\frac{1}{\sigma^2}\boldsymbol{X^T y}\right)^{-1}$$

Gibbs sampler in r

```
library(MASS)
library(invgamma)
library(ggplot2)
```

#create the data

```
set.seed(500)
k = 5
n = 100
mu_0 <- rep(5,k)
a<- 1
b<- 1
sigma_sq <- rinvgamma(a,b)
X <- cbind(1,mvrnorm(n, mu_0, diag(sqrt(sigma_sq), nrow = k, ncol = k)))
p = ncol(X)
beta_0 <- mvrnorm(1, rep(0,p), diag(1, nrow = p, ncol = p))
beta_0
```

```
## [1]  0.93982465  0.12632078  1.30880767  1.33386294  1.83918410 -0.08710633
```

```
y <- X%*%beta_0 + rnorm(n)
y
```

```
##             [,1]
##   [1,] 26.10795
##   [2,] 24.84746
##   [3,] 24.40064
##   [4,] 21.22677
```

```
##   [5,] 25.17422
##   [6,] 22.61218
##   [7,] 19.58396
##   [8,] 23.00568
##   [9,] 23.14592
##  [10,] 23.09130
##  [11,] 20.95627
##  [12,] 26.00864
##  [13,] 22.04884
##  [14,] 25.61237
##  [15,] 23.27876
##  [16,] 25.40521
##  [17,] 24.37630
##  [18,] 27.27199
##  [19,] 22.70006
##  [20,] 24.95977
##  [21,] 21.11948
##  [22,] 24.47853
##  [23,] 21.07187
##  [24,] 25.40718
##  [25,] 29.94345
##  [26,] 22.74661
##  [27,] 25.86486
##  [28,] 24.33287
##  [29,] 23.12953
##  [30,] 22.27680
##  [31,] 24.60884
##  [32,] 18.90845
##  [33,] 23.55489
##  [34,] 23.56026
##  [35,] 21.56586
##  [36,] 21.60303
##  [37,] 23.04420
##  [38,] 25.20702
##  [39,] 21.58299
##  [40,] 26.08561
##  [41,] 23.95030
##  [42,] 25.02623
##  [43,] 19.77077
##  [44,] 22.59349
##  [45,] 25.49661
##  [46,] 24.07498
##  [47,] 22.67581
##  [48,] 25.22633
##  [49,] 29.41261
##  [50,] 20.14673
##  [51,] 25.07768
##  [52,] 21.64436
##  [53,] 21.53927
##  [54,] 24.94877
##  [55,] 19.67023
##  [56,] 23.83284
##  [57,] 27.93730
##  [58,] 20.23228
```

```
##  [59,] 24.00155
##  [60,] 26.24619
##  [61,] 21.04811
##  [62,] 26.51349
##  [63,] 25.90731
##  [64,] 25.33707
##  [65,] 20.15522
##  [66,] 21.97057
##  [67,] 22.47257
##  [68,] 21.58136
##  [69,] 26.71615
##  [70,] 17.11643
##  [71,] 21.33717
##  [72,] 23.92145
##  [73,] 21.86949
##  [74,] 18.41003
##  [75,] 22.53258
##  [76,] 29.01101
##  [77,] 24.59012
##  [78,] 24.47152
##  [79,] 19.92747
##  [80,] 22.06411
##  [81,] 22.46087
##  [82,] 22.84764
##  [83,] 19.40497
##  [84,] 22.31050
##  [85,] 24.83159
##  [86,] 22.98860
##  [87,] 23.92760
##  [88,] 17.87301
##  [89,] 26.08815
##  [90,] 25.95767
##  [91,] 20.72363
##  [92,] 19.26372
##  [93,] 21.36900
##  [94,] 20.75411
##  [95,] 24.19505
##  [96,] 22.58767
##  [97,] 26.21773
##  [98,] 20.01469
##  [99,] 21.73043
## [100,] 21.65034
```

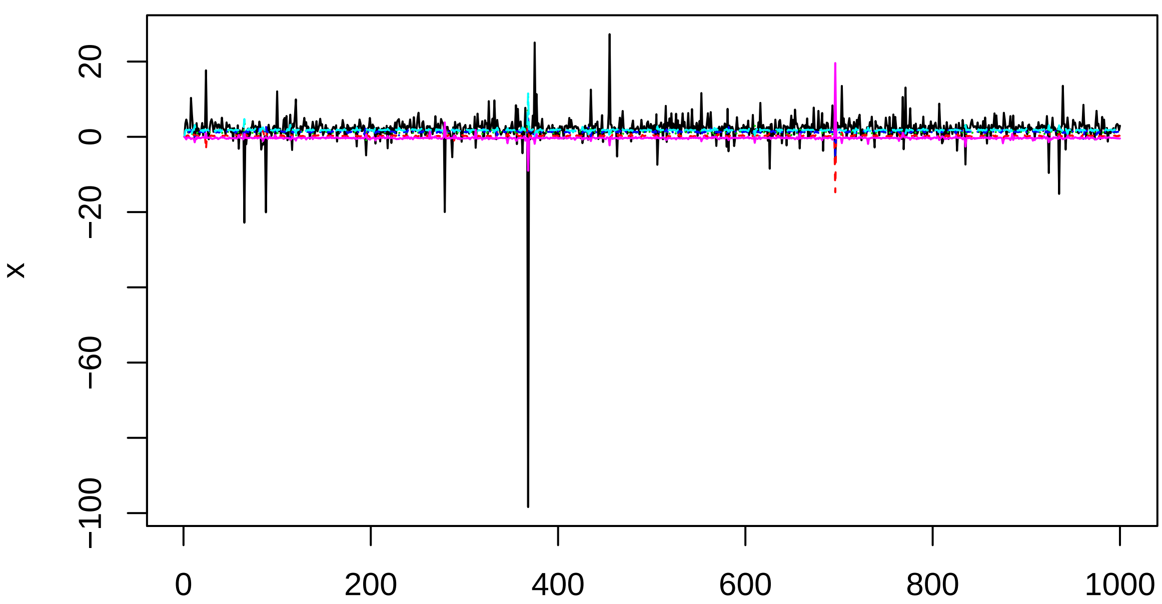
Gibbs sampler

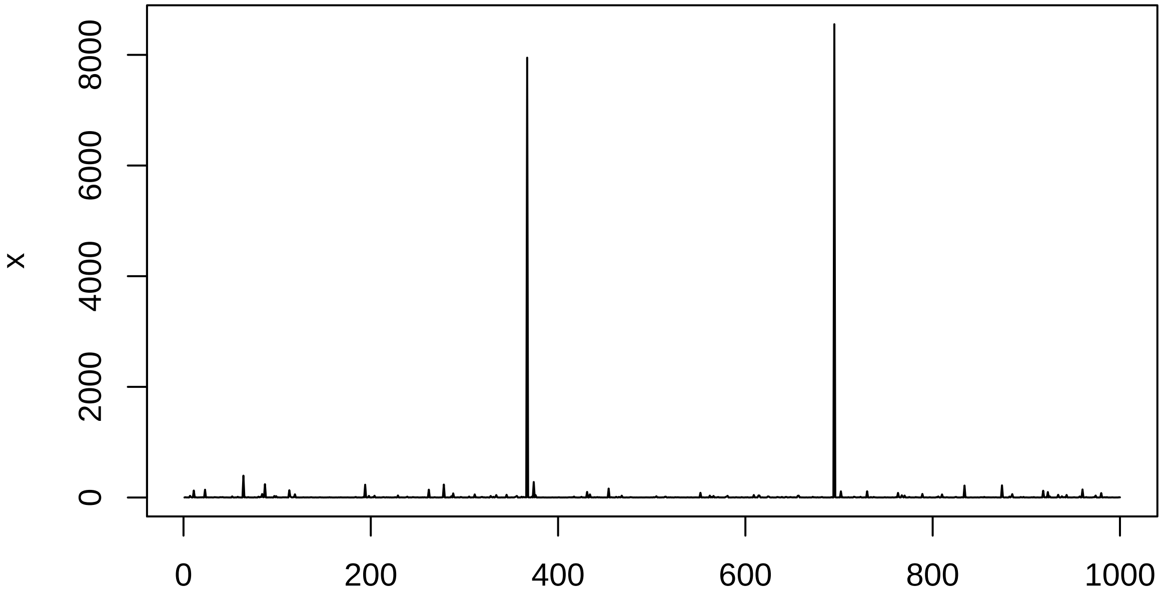
```
MCMC <- function(niter){
  #Itial values
  beta <- matrix(0,niter, p)
  sigma <- rep(0,niter)
  sigma[1] <- 1
  for(i in 2:niter){
```

```r
    # sample beta

    mu <- solve(t(X) %*% (X)) %*% (t(X) %*% (y))
    Dispersion <- solve(t(X) %*% (X)) * sigma_sq[i-1]
    beta[i,] <- mvrnorm(1, mu, Dispersion)

    # sample sigma_sq

    b_n <- 0.5 * t(y - X %*% beta[i,])%*%(y - X %*% beta[i,]) + 1
    a_n <- (n/2) + 1
    sigma_sq[i] <- rinvgamma(1, a, rate = b)

  }
  return(
    list(
       beta = beta,
    sigma_sq = sigma_sq
    )

  )



}
```

#MCMC results

```r
out <- MCMC (1000)
lapply(out, function(x) matplot(x, type = "l"))
```



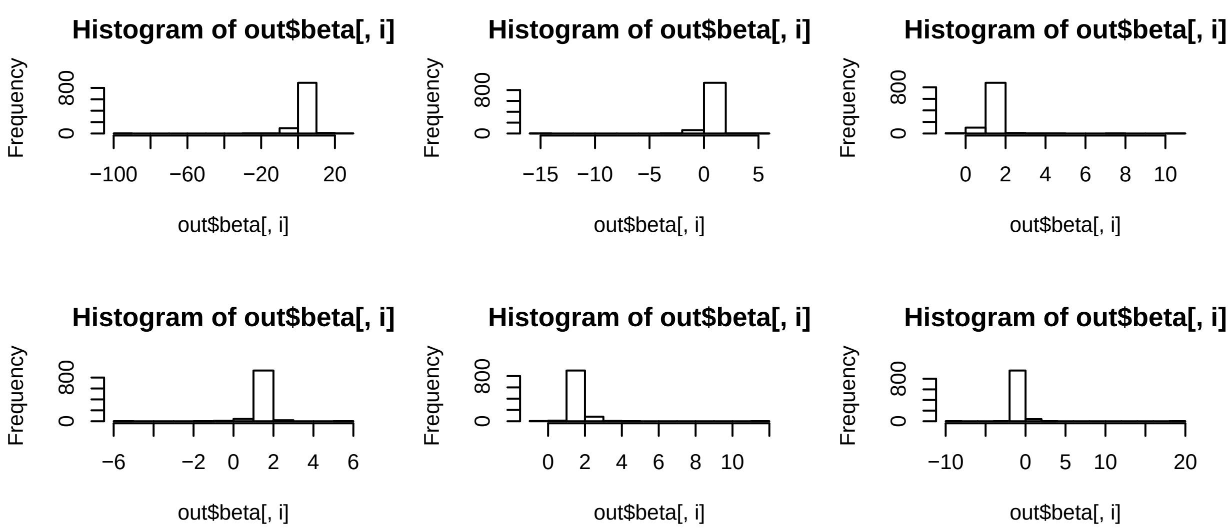



```
## $beta
## NULL
##
## $sigma_sq
## NULL
```

```r
# apply(out$beta, 2, hist)
```

plot the coeficients

```r
par(mfrow=c(3,3))
for (i in 1:6) {
  hist(out$beta[, i])
}
```



#posterior means

```r
#post_beta

beta_hat <- apply(out$beta, 2, mean)

# post_sigma_sq
mean(out$sigma_sq)
```

```
## [1] 23.76186
```

Digonises

```r
y_pred <- X %*% beta_hat
residuals <- y - y_pred

# residual plots
 par(mfrow = c(2,2))
hist(residuals, main = "residual plot")
qqnorm(residuals, main ="res_qqplot")
qqline(residuals, col = "blue")

# observed vs predicted values

plot(y, y_pred, main = "y vs y_pred")
abline(c(0, 1), col ="blue") ## a line of 45 degree angle
```