

# Notes on Bayesian Linear Regression

Max Leung

April 19, 2024

## 1 Joint Posterior Distribution

$$\begin{aligned} p(\beta, \sigma^2 | \mathbf{y}, \mathbf{X}) &= p(\beta | \sigma^2, \mathbf{y}, \mathbf{X}) p(\sigma^2 | \mathbf{y}, \mathbf{X}) \\ &= p(\sigma^2 | \beta, \mathbf{y}, \mathbf{X}) \underbrace{p(\beta | \mathbf{y}, \mathbf{X})}_{\int p(\beta, \sigma^2 | \mathbf{y}, \mathbf{X}) d\sigma^2} \end{aligned}$$

### 1.1 Normally Distributed and Homoscedastic y with Non-informative Prior

#### 1.1.1 Marginal Posterior Distribution

$$\begin{aligned} p(\beta, \sigma^2 | \mathbf{y}, \mathbf{X}) &\propto L(\mathbf{y} | \beta, \sigma^2, \mathbf{X}) \pi(\beta, \sigma^2 | \mathbf{X}) \\ &= N(\mathbb{E}(\mathbf{y} | \mathbf{X}), \text{Var}(\mathbf{y} | \mathbf{X})) \cdot C \\ &\propto N(\mathbf{X}\beta, \sigma^2 \mathbf{I}) \\ &= \frac{1}{(2\pi)^{n/2} \det(\sigma^2 \mathbf{I})^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\beta)'(\sigma^2 \mathbf{I})^{-1}(\mathbf{y} - \mathbf{X}\beta)\right) \\ &= \frac{1}{(2\pi)^{n/2} (\sigma^2)^{n/2} \det(\mathbf{I})^{1/2}} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)\right) \\ &= \frac{1}{(2\pi)^{n/2} (\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\hat{\beta} + \mathbf{X}\hat{\beta} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\hat{\beta} + \mathbf{X}\hat{\beta} - \mathbf{X}\beta)\right) \\ &= \frac{1}{(2\pi)^{n/2} (\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}((\mathbf{y} - \mathbf{X}\hat{\beta}) - \mathbf{X}(\beta - \hat{\beta}))'((\mathbf{y} - \mathbf{X}\hat{\beta}) - \mathbf{X}(\beta - \hat{\beta}))\right) \\ &= \frac{1}{(2\pi)^{n/2} (\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(\hat{\epsilon}' - (\beta - \hat{\beta})' \mathbf{X}')(\hat{\epsilon} - \mathbf{X}(\beta - \hat{\beta}))\right) \\ &= \frac{1}{(2\pi)^{n/2} (\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(\hat{\epsilon}'\hat{\epsilon} - \hat{\epsilon}' \mathbf{X}(\beta - \hat{\beta}) - (\beta - \hat{\beta})' \mathbf{X}' \hat{\epsilon} + (\beta - \hat{\beta})' \mathbf{X}' \mathbf{X}(\beta - \hat{\beta}))\right) \\ &= \frac{1}{(2\pi)^{n/2} (\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(\hat{\epsilon}'\hat{\epsilon} + (\beta - \hat{\beta})' \mathbf{X}' \mathbf{X}(\beta - \hat{\beta}))\right) \quad \text{since } \mathbf{X}' \hat{\epsilon} = \mathbf{0} \\ &= \frac{1}{(2\pi)^{n/2} (\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2} \hat{\epsilon}' \hat{\epsilon}\right) \exp\left(-\frac{1}{2\sigma^2} (\beta - \hat{\beta})' \mathbf{X}' \mathbf{X}(\beta - \hat{\beta})\right) \\ &= \frac{1}{(2\pi)^{n/2} (\sigma^2)^{n/2}} \exp\left(-\frac{n-k}{2\sigma^2} \hat{\sigma}^2\right) \exp\left(-\frac{1}{2\sigma^2} (\beta - \hat{\beta})' \mathbf{X}' \mathbf{X}(\beta - \hat{\beta})\right) \\ &= \frac{1}{(2\pi)^{n/2} (\sigma^2)^{n/2}} \exp\left(-\frac{n-k}{2\sigma^2} \hat{\sigma}^2\right) \frac{(2\pi)^{k/2} \det(\sigma^2 (\mathbf{X}' \mathbf{X})^{-1})^{1/2}}{(2\pi)^{k/2} \det(\sigma^2 (\mathbf{X}' \mathbf{X})^{-1})^{1/2}} \exp\left(-\frac{1}{2\sigma^2} (\beta - \hat{\beta})' \mathbf{X}' \mathbf{X}(\beta - \hat{\beta})\right) \\ &= \frac{1}{(2\pi)^{n/2} (\sigma^2)^{n/2}} \exp\left(-\frac{n-k}{2\sigma^2} \hat{\sigma}^2\right) (2\pi)^{k/2} (\sigma^2)^{k/2} \det((\mathbf{X}' \mathbf{X})^{-1})^{1/2} N(\beta | \hat{\beta}, \sigma^2 (\mathbf{X}' \mathbf{X})^{-1}) \\ &= \frac{\det((\mathbf{X}' \mathbf{X})^{-1})^{1/2}}{(2\pi)^{(n-k)/2}} (\sigma^2)^{-(n-k)/2} \exp\left(-\frac{n-k}{2\sigma^2} \hat{\sigma}^2\right) N(\beta | \hat{\beta}, \sigma^2 (\mathbf{X}' \mathbf{X})^{-1}) \\ &= \frac{\det((\mathbf{X}' \mathbf{X})^{-1})^{1/2}}{(2\pi)^{(n-k)/2} C_1} C_1 (\sigma^{-2})^{(n-k)/2+1} \exp\left(-\frac{(n-k)\hat{\sigma}^2}{2}/\sigma^2\right) N(\beta | \hat{\beta}, \sigma^2 (\mathbf{X}' \mathbf{X})^{-1}) \\ &\propto IG(\sigma^2 | \frac{n-k}{2}, \frac{(n-k)\hat{\sigma}^2}{2}) N(\beta | \hat{\beta}, \sigma^2 (\mathbf{X}' \mathbf{X})^{-1}) \quad \text{due to } IG(z | a, b) = \frac{b^a}{\Gamma(a)} (1/z)^{a+1} \exp(-b/z) \\ &= N(\beta | \hat{\beta}, \sigma^2 (\mathbf{X}' \mathbf{X})^{-1}) \chi^{-2}(\sigma^2 | n-k, \hat{\sigma}^2) \quad \text{due to } IG(z | \nu/2, \nu\tau^2/2) = \chi^{-2}(z | \nu, \tau^2) \end{aligned}$$

Thus,  $p(\beta|\sigma^2, \mathbf{y}, \mathbf{X}) = N(\hat{\beta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$  and  $p(\sigma^2|\mathbf{y}, \mathbf{X}) = \chi^{-2}(\sigma^2|n-k, \hat{\sigma}^2)$ .

$$\begin{aligned} p(\beta|\mathbf{y}, \mathbf{X}) &= \int p(\beta, \sigma^2|\mathbf{y}, \mathbf{X}) d\sigma^2 \\ &= \int p(\beta|\sigma^2, \mathbf{y}, \mathbf{X}) p(\sigma^2|\mathbf{y}, \mathbf{X}) d\sigma^2 \\ &= \int N(\beta|\hat{\beta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1}) \chi^{-2}(\sigma^2|n-k, \hat{\sigma}^2) d\sigma^2 \\ &= t(\beta|\hat{\beta}, (\mathbf{X}'\mathbf{X})^{-1}, n-k) \end{aligned}$$

### 1.1.2 Algorithm for sampling from joint posterior distribution

Given the closed form solutions of the marginal posterior distribution, we can sample from joint posterior distribution without using Metropolis-Hastings algorithm,

Step 1 - compute  $(\mathbf{X}'\mathbf{X})^{-1}$  and  $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$  and  $\hat{\sigma}^2 = \hat{\varepsilon}'\hat{\varepsilon}/(n-k) = (\mathbf{y} - \mathbf{X}\hat{\beta})'(\mathbf{y} - \mathbf{X}\hat{\beta})/(n-k)$

Step 2 - draw  $\sigma^2$  from  $\chi^{-2}(\sigma^2|n-k, \hat{\sigma}^2)$

Step 3 - draw  $\beta$  from  $N(\beta|\hat{\beta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$

In order to speed up the computation, we can apply QR decomposition of  $\mathbf{X}$  to compute both  $(\mathbf{X}'\mathbf{X})^{-1} = \mathbf{R}^{-1}\mathbf{R}^{-1'}$  and  $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$  which is the numerical solution of  $\mathbf{R}\hat{\beta} = \mathbf{Q}'\mathbf{y}$ . Read Gelman, et al. (2013, p.356) for the details.

The following R code performs the above algorithm.

```
library(LearnBayes)
set.seed(15)

sim_beta <- list()
sim_sigma_square <- list()
for (i in seq_len(2000)) {
  # lm() performs QR decomposition under the hood
  ols <- lm(y ~ x1 + x2 + x3, data = d, method = "qr")
  n_minus_k <- ols$df.residual
  beta_hat <- ols$coefficients
  sigma_square_hat <- sum(ols$residuals^2, na.rm = TRUE) / n_minus_k
  sim_sigma_square[[i]] <- LearnBayes::rigamma(1, n_minus_k / 2, n_minus_k * sigma_square_hat / 2)
  XX_inverse <- vcov(ols) / sigma_square_hat
  sim_beta[[i]] <- rmnorm(1, beta_hat, sim_sigma_square[[i]] * XX_inverse)
}

sim_beta_m <- do.call(rbind, sim_beta)
sim_sigma_square_m <- do.call(rbind, sim_sigma_square)

colMeans(sim_beta_m, na.rm = TRUE)
apply(sim_beta_m, 2, quantile, c(0.25, 0.5, 0.75), na.rm = TRUE)

mean(sim_sigma_square_m, na.rm = TRUE)
quantile(sim_sigma_square_m, c(0.25, 0.5, 0.75), na.rm = TRUE)
```

*blinreg* function from *LearnBayes* package does the same thing.

```
library(LearnBayes)
set.seed(15)

linear_reg_u_prior <-
  blinreg(
    y = d[["y"]],
    X = as.matrix(cbind(1, d[c("x1", "x2", "x3")])),
    m = 1000,
    prior = NULL
  )

colMeans(linear_reg_u_prior$beta, na.rm = TRUE)
apply(linear_reg_u_prior$beta, 2, quantile, c(0.25, 0.5, 0.75), na.rm = TRUE)

mean(linear_reg_u_prior$sigma, na.rm = TRUE)
quantile(linear_reg_u_prior$sigma, c(0.25, 0.5, 0.75), na.rm = TRUE)
```

The package *rstanarm* in R offers another easy-to-use function to draw samples from joint posterior distribution with Hamiltonian Monte Carlo (HMC) algorithm and QR decomposition.

```
library(rstanarm)

stan_glm(
  y ~ x1 + x2 + x3,
  family = gaussian(),
  prior_intercept = NULL,
  prior = NULL,
  prior_aux = NULL,
  algorithm = "sampling",
  QR = TRUE,
  data = d
)
```

## 1.2 t Distributed y

It is the bayesian estimation of robust student t regression.

### 1.2.1 Joint Posterior Distribution

$$\begin{aligned}
 p(\beta, \Omega, \nu | \mathbf{y}, \mathbf{X}) &\propto L(\mathbf{y} | \beta, \Omega, \nu, \mathbf{X}) \pi(\beta, \Omega, \nu | \mathbf{X}) \\
 &= t(\mathbf{y} | \beta, \Omega, \nu, \mathbf{X}) \cdot \pi(\beta, \Omega, \nu | \mathbf{X}) \\
 &= \int N(\mathbf{y} | \beta, z \cdot \Omega, \mathbf{X}) \chi^{-2}(z | \nu, 1) dz \cdot \pi(\beta, \Omega, \nu | \mathbf{X}) \quad \text{where } \Omega = \text{diag}(\sigma^2)
 \end{aligned}$$

The joint posterior distribution can be sampled by using Metropolis-Hastings algorithm (including Gibbs Sampler). The following R code demonstrates an example where  $\pi(\beta, \Omega, \nu | \mathbf{X}) = \prod_j N(\beta_j | 0, 10^2) \cdot \text{Gamma}(\sigma^2 | 2, 0.1) \cdot \text{Gamma}(\nu | 2, 0.1)$ .

```
library(runjags)
set.seed(15)

model_string =
"
  model {
    for (i in 1:N) {
      # variance of y_i = z_i sigma^2
      # precision of y_i = 1 / (z_i sigma^2) = (1 / z_i) / sigma^2 = phi_i / sigma^2
      y[i] ~ dnorm(mu[i], phi[i] / sigma_square)
      mu[i] <- b[1] + b[2]*x1[i] + b[3]*x2[i] + b[4]*x3[i]
      phi[i] ~ dgamma(nu / 2, nu / 2)
    }
    # prior
    for (j in 1:4) {
      b[j] ~ dnorm(0, 1 / 100)
    }
    sigma_square ~ dgamma(2, 0.1)
    nu ~ dgamma(2, 0.1)
  }
"

model_data <-
list(
  "y" = d[["y"]],
  "N" = length(d[["y"]]),
  "x1" = d[["x1"]],
  "x2" = d[["x2"]],
  "x3" = d[["x3"]]
)

t_reg_jags <-
run.jags(
  model = model_string,
  n.chains = 1,
  data = model_data,
  monitor = c("b", "sigma_square", "nu"),
  adapt = 1000,
  burnin = 5000,
  sample = 5000
)

print(t_reg_jags)
plot(t_reg_jags, var = "b")
```

```
plot(t_reg_jags, var = "sigma_square")
plot(t_reg_jags, var = "nu")
```

The *brms* package in R offers simpler code.

```
library(brms)

brm(
  data = d,
  family = student,
  y ~ 1 + x1 + x2 + x3,
  prior = c(
    prior(normal(0, 10), class = Intercept),
    prior(normal(0, 10), class = b),
    prior(gamma(2, 0.1), class = nu),
    prior(gamma(2, 0.1), class = sigma)
  ),
  seed = 15
)
```

## 2 Reference

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). Bayesian Data Analysis (3rd ed.). Chapman and Hall/CRC. <https://doi.org/10.1201/b16018>

Gelman, A., Hill, J., & Vehtari, A. (2020). Regression and Other Stories. Cambridge: Cambridge University Press.