

# Illustration

Wenhao Gou | wg2364

4/17/2021

## Re-state the question:

Climate researchers are interested in modeling the hurricane trajectories to forecast the wind speed. Let  $t$  be time (in hours) since a hurricane began, and For each hurricane  $i$ , we denote  $Y_i(t)$  be the wind speed of the  $i$ th hurricane at time  $t$ . The following Bayesian model was suggested.

$$Y_i(t+6) = \beta_{0,i} + x_{i,1}\beta_{1,i} + x_{i,2}\beta_{2,i} + x_{i,3}\beta_{3,i} + \beta_{4,i,j}Y_{i,j}(t) + \beta_{5,i,j}\Delta_{i,1}(t) + \beta_{6,i,j}\Delta_{i,2}(t) + \beta_{7,i}\Delta_{i,3} + \epsilon_i(t)$$

where  $x_{i,1}$  is the month of year when the hurricane started,  $x_{i,2}$  is the calendar year of the hurricane, and  $x_{i,3}$  is the type of hurricane,  $\Delta_{i,1}(t)$ ,  $\Delta_{i,2}(t)$  and  $\Delta_{i,3}(t)$  is the change of latitude longitude, and wind speed between  $t-6$  and  $t$ , and  $\epsilon_{i,t}$  follows a normal distributions with mean zero and variance  $\sigma^2$ , independent across  $t$ .

In the model,  $\beta_i = (\beta_{0,i}, \beta_{1,i}, \dots, \beta_{7,i})$  are the random coefficients associated the  $i$ th hurricane, we assume that

$$\beta_i \sim N(\beta, \Sigma)$$

follows a multivariate normal distributions with mean  $\beta$  and covariance matrix  $\Sigma$ .

**Prior distributions** We assume the following non-informative or weak prior distributions for  $\sigma^2$ ,  $\beta$  and  $\Sigma$ .

$$P(\sigma^2) \propto \frac{1}{\sigma^2}; \quad P(\beta) \propto 1; \quad P(\Sigma^{-1}) \propto |\Sigma|^{-(d+1)} \exp(-\frac{1}{2}\Sigma^{-1})$$

$d$  is dimension of  $\beta$ .

## Find out the posterial distribution:

It is easy to see that:

$$f(Y_i(t+6)|\beta, \sigma) \sim N(\mu, \sigma^2)$$

Where  $\mu$  is the linear combination of all coefficients.

Our objective is to derive  $f(\beta, \sigma|Y_i(t+6))$

$$f(\beta, \beta_i, \Sigma^{-1}, \sigma^2 | Y_i(t+6)) \propto f(Y_i(t+6) | \beta_i, \sigma^2) f(\beta_i | \beta, \Sigma^{-1}) f(\beta) f(\Sigma^{-1}) f(\sigma^2)$$

$$f(Y_i(t+6) | \beta_i, \sigma) \propto \prod_{i=1}^N \frac{1}{\sigma} \exp\left\{-\frac{(x_i - \mu_i)^2}{2\sigma^2}\right\}, \text{ where } \mu_i = X_i^T \beta_i$$

$$f(\beta_i | \beta, \Sigma^{-1}) \propto \frac{1}{\sqrt{|\Sigma^{-1}|}} \exp\left\{-\frac{1}{2}(\beta_i - \beta)^T \Sigma (\beta_i - \beta)\right\}$$

$$f(\sigma^2) \propto \frac{1}{\sigma^2}$$

$$f(\beta) \propto 1$$

$$f(\Sigma^{-1}) \propto |\Sigma|^{-8} \exp\left(-\frac{1}{2}\Sigma^{-1}\right)$$

```
data = read.csv("hurricane356.csv")
df = data %>%
  mutate(Month = factor(Month, levels = c("January", "April", "May", "June", "July", "August",
                                           "September", "October", "November", "December")),
         Month = as.numeric(Month),
         Nature = as.numeric(as.factor(Nature)))
id = df %>% group_by(Season, ID) %>% summarize(n = n())
```

## `summarise()` has grouped output by 'Season'. You can override using the `.groups` argument.

```
upd.bi = function(df, epis, beta, sigma, n){
  sd = solve(epis + diag(n*sigma, 8, 8)) # posterior distribution for beta.i
  mu = sd %*% (epis %*% beta + sigma*sum(df %*% beta)) # posterior distribution for beta.i
  return(list(sd = sd, mu = mu))}

mc = function(df, a, b, ini.sigma, niter = 2){
  delta1 = c(0, sapply(2:nrow(df), function(i){df[i,7] - df[i-1, 7]}))
  delta2 = c(0, sapply(2:nrow(df), function(i){df[i,8] - df[i-1, 8]}))
  delta3 = c(0, sapply(2:nrow(df), function(i){df[i,9] - df[i-1, 9]}))
  dt = cbind(df[,1], df[,4], df[,3], df[,5], df[,9], delta1, delta2, delta3) # final dataset for each i
  n = nrow(dt)
  epis = vector("list", niter)
  beta = matrix(NA, nrow = 8, ncol = niter)
  beta.i = matrix(NA, nrow = 8, ncol = niter)
  sigma = rep(NA, niter)
  epis[[1]] = cor(matrix(1/extraDistr::rinvgamma(64, alpha = a, beta = b), 8, 8)) # initial epsilon^-1
  beta[,1] = rep(1, 8) # initial beta (8x1)
  sigma[1] = ini.sigma # initial sigma
  beta.i[,1] = mvrnorm(1, mu = upd.bi(dt, epis[[1]], beta[,1], sigma[1], n)$mu,
                      Sigma = upd.bi(dt, epis[[1]], beta[,1], sigma[1], n)$sd)
  # mvrnorm generates the beta_i for each subgroup(8x1) - beta.i ~ N(beta, epsilon)
  for (i in 2:niter){
    epis[[i]] = rinvwishart(n + a, diag(sum((beta.i[,i-1] - beta[,i-1])^2), 8, 8) + 0.001)
    # use beta.i and beta update epsilon, pi(epsilon) ~ Inviwishart(n + a, beta_variance + b)!!!!!!!!!!!!
    beta[,i] = mvrnorm(1, mu = n*beta.i[,i-1], Sigma = n*epis[[i]]) # beta ~ N(n*beta_i, n*epsilon)
    sigma[i] = extraDistr::rinvgamma(1, alpha = n + 1, sum((dt[,5] - dt %*% beta.i[,i-1])^2))
    # sigma ~ inverse.gamma(n + 1, sample residual of Yi)
    beta.i[,i] = mvrnorm(1, mu = upd.bi(dt, epis[[i]], beta[,i], sigma[i], n)$mu,
                          Sigma = upd.bi(dt, epis[[i]], beta[,i], sigma[i], n)$sd)}
  return(list(beta.i = beta.i))}
```

```

whole.mc = function(df, id){
  sub.grp = vector("list", nrow(id))
  id.n = c(0, id$n)
  for (i in 1:length(sub.grp)){
    sub.grp[[i]] = df[(id.n[i]+1):id.n[i+1],]
    ith.mc = vector("list", length(sub.grp))
    for (ii in 1:length(ith.mc)){
      ith.mc[[ii]] = sapply(1: length(sub.grp),
                            function(iii){mc(sub.grp[[iii]],a = 0.001, b = 0.001, ini.sigma = 1)})})
    }
  return(ith.mc)}

```