STT 465
Bayesian Multiple Linear Regression:

- Mixed Effects Models

- Gibbs Sampler with blocked or scalar updates of effects.

Bayesian Multiple Linear Regression

- **Gaussian Linear Regression Model**

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i$$

- **Matrix representation**

Stack equations 1-n to get $y = X\beta + \varepsilon$

- **Likelihood (assuming iid normal errors)** $\varepsilon \sim MVN\left(0, \mathrm{I}\sigma_\varepsilon^2\right)$

$$p\left(y \mid X, \beta, \sigma_\varepsilon^2\right) = N\left(y \middle| X\beta, I\sigma_\varepsilon^2\right)$$

$$= \left(2\pi\right)^{-n/2} \left\| I\sigma_\varepsilon^2 \right\|^{-1/2} Exp\left\{ -\frac{1}{2\sigma_\varepsilon^2} \left(y - X\beta\right)' \left(y - X\beta\right) \right\}$$

Prior Distribution

⇒ So far we have assumed that effects come all from the same prior.

⇒ However, in practice we may need to assign different priors to different sets of effects.

⇒ For instance: (i) we may want to estimate some effects (e.g., age, etc.) without shrinkage (i.e., using a flat prior) and (ii) we may want to estimate different variances for different sets of predictors.

⇒ Suppose we define K groups of effects, according to the following partition of the columns of X

$$X = \left(X_1, X_2, \ldots, X_K\right) \qquad \beta = \left(\beta_1, \beta_2, \ldots, \beta_K\right)'$$

$$X\beta = X_1\beta_1 + X_2\beta_2 + \ldots + X_K\beta_K$$

Bayesian Multiple Linear Regression

- If we group predictors in k sets we can write the regression as follows

$$y = \sum_{k=1}^{K} X_k \beta_k + \varepsilon$$

- And the likelihood can be expressed as

$$p\left(y \mid X, \beta, \sigma_\varepsilon^2\right) = N\left(y \middle| \sum_{k=1}^{K} X_k \beta_k, I\sigma_\varepsilon^2\right)$$

$$= \left(2\pi\right)^{-n/2} \left\| I\sigma_\varepsilon^2 \right\|^{-1/2} Exp\left\{ -\frac{1}{2\sigma_\varepsilon^2} \left(y - \sum_{k=1}^{K} X_k \beta_k\right)' \left(y - \sum_{k=1}^{K} X_k \beta_k\right) \right\}$$

Prior Distribution

=> Assume that effects are independent, each following a normal distribution with mean zero and group-specific variance, that is

$$\beta_{kj} \sim N\left(0, \sigma^2_{\beta_k}\right) \quad \text{[group-specific variances]}$$

=> If we assign scaled-inverse chi-squared priors to each of these variances the joint prior becomes

$$p\left(\beta, \sigma^2_\varepsilon, \sigma^2_{\beta_1}, \ldots, \sigma^2_{\beta_K}\right) = \prod_{k=1}^{K} N\left(\beta_k \middle| 0, I\sigma^2_{\beta_k}\right) \chi^{-2}\left(\sigma^2_{\beta_k} \middle| df_k, S_k\right)$$
$$\times \chi^{-2}\left(\sigma^2_\varepsilon \middle| df_\varepsilon, S_\varepsilon\right)$$

Posterior Density

Joint Posterior Density

$$
\begin{aligned}
p\left(\beta,\sigma_{\varepsilon}^{2},\sigma_{\beta_1}^{2},\ldots,\sigma_{\beta_K}^{2} \mid y\right) \propto\; & N\left(y \middle| \sum_{k=1}^{K} X_k\beta_k, I\sigma_{\varepsilon}^{2}\right) \\
& \times \prod_{k=1}^{K} N\left(\beta_k \middle| 0, I\sigma_{\beta_k}^{2}\right) \chi^{-2}\left(\sigma_{\beta_k}^{2} \middle| df_k, S_k\right) \\
& \times \chi^{-2}\left(\sigma_{\varepsilon}^{2} \middle| df_{\varepsilon}, S_{\varepsilon}\right)
\end{aligned}
$$

Fully Conditionals

Marker Effects

$$p\left(\beta_k \mid ELSE\right) \propto N\left(y \middle| \sum_{l=1}^{K} X_l \beta_l, I\sigma_\varepsilon^2\right) \times N\left(\beta_k \middle| 0, I\sigma_{\beta_k}^2\right)$$

$$\propto N\left(y - \sum_{l \neq k} X_l \beta \middle| X_k \beta_k, I\sigma_\varepsilon^2\right) \times N\left(\beta_k \middle| 0, I\sigma_{\beta_k}^2\right)$$

$$\propto N\left(\tilde{y}_{(k)} \middle| X_k \beta_k, I\sigma_\varepsilon^2\right) \times N\left(\beta_k \middle| 0, I\sigma_{\beta_k}^2\right) \quad \text{where: } \tilde{y}_{(k)} = y - \sum_{l \neq k} X_l \beta_l$$

Using previous results we can show that

$$p\left(\beta_k \mid ELSE\right) \propto N\left(\beta_k \middle| C_k^{-1} rhs_k, C_k^{-1}\right)$$

$$C_k = \left[X_k' X_k \sigma_\varepsilon^{-2} + I\sigma_{\beta_k}^{-2}\right]$$

$$rhs_k = X_k' \tilde{y}_{(k)} \sigma_\varepsilon^{-2}$$

Fully Conditionals

Error Variances

$$p\left(\sigma_\varepsilon^2 \mid ELSE\right) \propto \left(\sigma_\varepsilon^2\right)^{-n/2} Exp\left\{-\frac{\varepsilon'\varepsilon}{2\sigma_\varepsilon^2}\right\}\left[\left(\sigma_\varepsilon^2\right)^{-(1+df_\varepsilon/2)} e^{-\frac{S_\varepsilon}{2\sigma_\varepsilon^2}}\right]$$

$$\propto \left(\sigma_\varepsilon^2\right)^{-[1+(n+df_\varepsilon)/2]} Exp\left\{-\frac{\varepsilon'\varepsilon + S_\varepsilon}{2\sigma_\varepsilon^2}\right\}$$

$$= \chi^{-2}\left(\sigma_\varepsilon^2 \middle| S = \varepsilon'\varepsilon + S_\varepsilon, df = n + df_\varepsilon\right) \qquad [2]$$

Gibbs Sampler

Variances of effects

$$p\left(\sigma^2_{\beta_k} \mid ELSE\right) \propto N\left(\beta_k \middle| 0, I\sigma^2_{\beta_k}\right)\chi^{-2}\left(\sigma^2_{\beta_k} \middle| df_{\beta_k}, S_{\beta_k}\right)$$

Using previous results we can show that

$$p\left(\sigma^2_{\beta_k} \mid ELSE\right) \propto \chi^{-2}\left(\sigma^2_{\beta_k} \middle| df_{\beta_k} + p_k \ , \ S_{\beta_k} + \beta_k'\beta_k\right)$$

Gibbs Sampler

Gibbs sampler with scalar updates (sampling one effect at a time)

- Among the computations we need to perform, inverting the the matrix of coefficients (C_k) is the most demanding.

- This inversion needs to be performed at every iteration of the sampler.

- We can avoid doing this by sampling effects one at a time.

- Suppose that the kth group contains only one predictor, then

$p(\beta_k \mid ELSE) \propto N\left(\beta_k \middle| C_k^{-1} rhs_k, C_k^{-1}\right)$ the fully conditional is a normal density, not a multivariate normal.

- And $C_k = \left[X_k' X_k \sigma_\varepsilon^{-2} + I\sigma_{\beta_k}^{-2}\right]$ $rhs_k = X_k' \tilde{y}_{(k)} \sigma_\varepsilon^{-2}$ are scalar.

- Therefore $C_k^{-1} = 1/C_k$.

Sample code

```
z<-rnorm(ncol(X))
for(j in 1:ncol(X)){
    xj=X[,j]
    error<-error+xj*beta[j]
      C=sumSqX[j]/varE[i]+1/varB[i,groups[j]]
      rhs<-sum(xj*error)/varE[i]
      sol<-rhs/C
      beta[j]<-sol+z[j]/sqrt(C)
    error<-error-xj*beta[j]
}
```

Dealing with missing values

Types of missing values

- Non-informative (e.g., completely at random)

- Informative (e.g., censoring)

Non-informative missings can be simply removed, e.g.,

```
isNA=is.na(y)

y_no_NA=y[!isNA]
X_no_NA=X[!isNA,]
## now regress y_no_NA on X_no_NA
```

But we can also deal with NAs in different manner: that is by sampling the unobserved values from fully conditionals.

Sampling Non-Informative Missing Values

Types of missing values

- Non-informative (e.g., completely at random)

- Informative (e.g., censoring)

Non-informative missings can be simply removed, e.g.,

```
isNA=is.na(y)

y_no_NA=y[!isNA]
X_no_NA=X[!isNA,]
## now regress y_no_NA on X_no_NA
```

But we can also deal with NAs in different manner: that is by sampling the unobserved values from fully conditionals.