CheeseData

Rohit Arora

1. Setup

The data consists of quantities sold and the prices for 88 stores selling cheese. In addition we have a store specific display variable indicating if the cheese was on display. We start of with the following model

$$y_i = X_i\beta + Z_i\gamma_i + \epsilon_i$$

where the coefficients β correspond to the fixed effect across stores and γ_i correspond to the store specific random effects. To setup a fully bayesian model we assume the following set of priors on the parameters in the model

$$\begin{gathered}
y_i|\beta_i, \gamma_i, \sigma^2, \Sigma \sim \mathcal{N}\left(X\beta + Z_i\gamma_i, \sigma^2 I\right) \\
\beta \propto 1 \\
\gamma_i|\Sigma \sim \mathcal{N}\left(0, \Sigma\right) \\
\Sigma \sim \mathcal{IW}\left(d_0, C_0\right) \\
\sigma^2 \propto \frac{1}{\sigma^2}
\end{gathered}$$

β is assumed to have a flat prior. γ_i are considered to have a Jeffrey's prior. Other prior's are standard.

2. Gibbs Updates

Below we specify the posterior updates for the parameters in our model. n_s is the number of stores, n_t is the number of observations per store and n is the total number of stores

- posterior for for fixed effect β

$$\begin{aligned}
p\left(\beta|\mathbf{y}, \gamma, \sigma^2, \Sigma; \Theta\right) &\propto p\left(\beta\right)\prod_{i=1}^{n} p\left(\mathbf{y}_i|\beta, \gamma_i, \sigma^2\right) \\
&\propto 1 \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n_s}\sum_{t=1}^{n_t}\left(y_{it} - x_{it}^\top\beta - z_{it}^\top\gamma_i\right)^2\right) \\
&= \exp\left(-\frac{1}{2}\sum_{i=1}^{n_s}\left(\mathbf{y}_i - X_i\beta - Z_i\gamma_i\right)^\top\left(\sigma^2 I\right)^{-1}\left(\mathbf{y}_i - X_i\beta - Z_i\gamma_i\right)\right) \\
&= \exp\left(-\frac{1}{2}\left(\beta^\top\left(\sigma^{-2}\sum_{i=1}^{n_s}X_i^\top X_i\right)\beta - 2\left(\sum_{i=1}^{n_s}\sigma^{-2}(\mathbf{y}_i - Z_i\gamma_i)^\top X_i\right)\beta\right)\right) \\
&\sim \mathcal{N}\left(\left(\sum_{i=1}^{n_s}X_i^\top X_i\right)^{-1}\left(\sum_{i=1}^{n_s}\left(\mathbf{y}_i - Z_i\gamma_i\right)^\top X_i\right), \left(\sigma^{-2}\sum_{i=1}^{n_s}X_i^\top X_i\right)^{-1}\right)
\end{aligned}$$

- posterior update for random effect γ_i

$$
\begin{aligned}
p\left(\gamma_i|\mathbf{y}_i, \sigma^2, \Sigma; \Theta\right) &\propto p\left(\mathbf{y}_i|\gamma_i, \sigma^2\right) p\left(\gamma_i|\Sigma\right) \\
&\propto \exp\left(-\frac{1}{2\sigma^2}\sum_{t=1}^{n_i}\left(y_{it} - x_{it}^\top\beta - z_{it}^\top\gamma_i\right)^\top\left(y_{it} - x_{it}^\top\beta - z_{it}^\top\gamma_i\right)\right)\exp\left(-\frac{1}{2}\gamma_i^\top\Sigma^{-1}\gamma_i\right) \\
&= \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}_i - X_i\beta - Z_i\gamma_i)^\top\left(\mathbf{y}_i - X_i\beta - Z_i\gamma_i\right)\right)\exp\left(-\frac{1}{2}\gamma_i^\top\Sigma^{-1}\gamma_i\right) \\
&= \exp\left(-\frac{1}{2}\left((\mathbf{y}_i - X_i\beta - Z_i\gamma_i)^\top\left(\sigma^2 I\right)^{-1}(\mathbf{y}_i - X_i\beta - Z_i\gamma_i) + \gamma_i^\top\Sigma^{-1}\gamma_i\right)\right) \\
&\propto \exp\left(-\frac{1}{2}\left(-2\sigma^{-2}(\mathbf{y}_i - X_i\beta)^\top Z_i\gamma_i + \sigma^{-2}\gamma_i^\top Z_i^\top Z_i\gamma_i + \gamma_i^\top\Sigma^{-1}\gamma_i\right)\right) \\
&= \exp\left(-\frac{1}{2}\left(\gamma_i^\top\left(\sigma^{-2}Z_i^\top Z_i + \Sigma^{-1}\right)\gamma_i - 2\sigma^{-2}(\mathbf{y}_i - X_i\beta)^\top Z_i\gamma_i\right)\right) \\
&\sim \mathcal{N}\left(\left(\sigma^{-2}Z_i^\top Z_i + \Sigma^{-1}\right)^{-1}\sigma^{-2}(\mathbf{y}_i - X_i\beta)^\top Z_i, \left(\sigma^{-2}Z_i^\top Z_i + \Sigma^{-1}\right)^{-1}\right)
\end{aligned}
$$

- posterior update for σ^2

$$
\begin{aligned}
p\left(\sigma^2|\mathbf{y}, \gamma, \Sigma; \Theta\right) &\propto p\left(\mathbf{y}|\gamma, \sigma^2\right) p\left(\sigma^2\right) = p\left(\sigma^2\right)\prod_{i=1}^{n_s} p\left(\mathbf{y}_i|\gamma_i, \sigma^2\right) \\
&\propto \sigma^{-2\frac{(n+1)}{2}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n_s}\sum_{t=1}^{n_t}\left(y_{it} - z_{it}^\top\gamma_i\right)^\top\left(y_{it} - z_{it}^\top\gamma_i\right)\right) \\
&\sim \mathrm{I\Gamma}\left(\frac{n}{2}, \frac{1}{2}\sum_{i=1}^{n_s}\sum_{t=1}^{n_t}\left(y_{it} - x_{it}^\top\beta - z_{it}^\top\gamma_i\right)^\top\left(y_{it} - x_{it}^\top\beta - z_{it}^\top\gamma_i\right)\right)
\end{aligned}
$$

- posterior update for Σ

$$
p\left(\Sigma|\mathbf{y}, \gamma, \sigma^2; \Theta\right) \propto \mathcal{IW}\left(d_0 + n_s, C_0 + \sum_{i=1}^{n_s}\gamma_i\gamma_i^\top\right)
$$

3. Gibbs Sampler

Below is the main function for the Gibbs Sampler. We start with β and then sequentially update in the following order γ, σ^2 and Σ.

```
gibbsCheese <- function(y, X, Z, allStores, d, C, iter= 10, burn=2, thin=2) {

  n <- length(y); p <- ncol(Z); nstores <- nlevels(allStores);

  #setup the structure for the chain
  beta <- array(0, dim=c(ncol(X),iter)); gamma <- array(0,dim=c(nstores, ncol(Z), iter));
  Sigma <- array(0, dim=c(p, p, iter)); sig.sq <- rep(0,iter)

  #initialize the chain
```

```r
gamma[,,1] <- rep(0, nstores*p);  Sigma[,,1] <- diag(p)
beta[,1] <- sig.sq[1] <- 1

for (iter in 2:iter) {

  # Update fixed effect for beta
  beta.post.var <- solve(
    sig.sq[iter-1]^-1 * sum(
      sapply(1:nstores, function(ns) {
        bStore <- which(ns==as.integer(allStores));
        Xi <- X[bStore,,drop=FALSE]; crossprod(Xi)
      })
    ))

  beta.post.mean <- beta.post.var/sig.sq[iter-1] %*% sum(
    sapply(1:nstores, function(ns) {
      bStore <- which(ns==as.integer(allStores));
      Zi <-  Z[bStore,]; yi <- y[bStore]; Xi <- X[bStore,,drop=FALSE]
      t(yi - Zi %*% gamma[ns,,iter-1]) %*% Xi
    })
  )

  beta[,iter] <- rmvnorm(1, beta.post.mean, beta.post.var)

  # Update random effect gamma for each store
  Sig.inv <- solve(Sigma[,,iter-1])

  gamma[,,iter] <-
    t(sapply(1:nstores, function(ns) {

      bStore <- which(ns==as.integer(allStores))
      Xi <- X[bStore,,drop=FALSE]; Zi <-  Z[bStore,]; yi <- y[bStore]

      gamma.post.var <- solve(Sigma[,,iter-1] + (sig.sq[iter-1])^-1 * crossprod(Zi))
      gamma.post.mean <-
        gamma.post.var %*%
        (sig.sq[iter-1]^-1 * t(Zi) %*% (yi - Xi %*% beta[,iter,drop=FALSE]))

      rmvnorm(1, gamma.post.mean, gamma.post.var)
    }))


  # Update sig.sq

  SS <- sum(
    sapply(1:nstores, function(ns) {
      bStore <- which(ns==as.integer(allStores));
      Zi <-  Z[bStore,]; yi <- y[bStore]; Xi <- X[bStore,,drop=FALSE]
      crossprod(yi - Zi %*% gamma[ns,,iter] - Xi %*% beta[,iter,drop=FALSE])
    })
  )

  sig.sq[iter] <- 1/rgamma(1, n/2, SS/2)
```

```r
    #Update Sigma values for IW
    S <- 0; for (s in 1:nstores) S <- S + gamma[s,,iter] %*% t(gamma[s,,iter])
    Cn = C + S; dn = d + nstores
    Sigma[,,iter] <- riwish(dn, Cn)
  }

  thinseq <- seq(1,iter - 1, by=thin)
  gamma = gamma[,,-burn]; Sigma = Sigma[,,-burn];sig.sq = sig.sq[-burn]
  gamma = gamma[,,thinseq]; Sigma = Sigma[,,thinseq]; sig.sq = sig.sq[thinseq]

  #posterior medians
  gamma.post.median    <- apply(gamma, 2, median)
  sig.sq.post.median  <- median(sig.sq)
  Sigma.post.median   <- apply(Sigma, c(1,2), mean)

  list(gamma.post.median = gamma.post.median,
              sig.sq.post.median = sig.sq.post.median,
              Sigma.post.mean = Sigma.post.median)
}
```