Introduction to Multilevel Modeling

Jim Albert

March 18, 2018

Efron and Morris Baseball Data

Efron and Morris, in a famous 1975 JASA paper, introduced the problem of estimating the true batting averages for 18 players during the 1971 baseball season. In the table, we observe the number of hits for each player in the first 35 batting opportunities in the season.

```
> d <- data.frame(Name=c("Clemente", "Robinson", "Howard", "Johnstone",
+       "Berry",  "Spencer",  "Kessinger", "Alvarado", "Santo",
+       "Swaboda", "Petrocelli",  "Rodriguez", "Scott",  "Unser",
+       "Williams",  "Campaneris",  "Munson",   "Alvis"),
+       Hits=c(18, 17, 16, 15, 14, 14, 13, 12, 11,
+            11, 10, 10, 10, 10, 10,  9,  8,  7),
+       At.Bats=45)
```

The Multilevel Model

One can simultaneously estimate the true batting averages by the following multilevel model. We assume the hits for the jth player y_j has a binomial distribution with sample size n_j and probability of success p_j, $j = 1, ..., 18$. The true batting averages $p_1, .., p_{18}$ are assumed to be a random sample from a beta(a, b) distribution. It is convenient to reparameterize a and b into the mean $\eta = a/(a + b)$ and precision $K = a + b$. We assign (η, K) the noninformative prior

$$g(\eta, K) \propto \frac{1}{\eta(1-\eta)} \frac{1}{(1+K)^2}$$

After data y is observed, the posterior distribution of the parameters $(\{p_j\}, \eta, K)$ has the convenient representation

$$g(\{p_j\}, \eta, K|y) = g(\eta, K|y) \times g(\{p_j\}|\eta, K, y).$$

Conditional on η and K, the posterior distributions of $p_1, ..., p_{18}$ are independent, where

$$p_j \sim Beta(y_j + K\eta, n_j - y_j + K(1 - \eta)).$$

The posterior density of (η, K) is given by

$$g(\eta, K|y) \propto \prod_{j=1}^{18} \left(\frac{B(y_j + K\eta, n_j - y_j + K(1-\eta))}{B(K\eta, n_j - y_j + K(1-\eta))} \right) \frac{1}{\eta(1-\eta)} \frac{1}{(1+K)^2}.$$

Simulation of the Posterior of (η, K)

For computational purposes, it is convenient to reparameterize η and K to the real-valued parameters

$$\theta_1 = \log \frac{\eta}{1-\eta}, \theta_2 = \log K.$$

The log posterior of the vector $\theta = (\theta_1, \theta_2)$ is programmed in the function `betaabinexch`.

We initially use the `laplace` function to find the posterior mode and associated variance-covariance matrix. The inputs are the log posterior function, an initial guess at the mode, and the data.

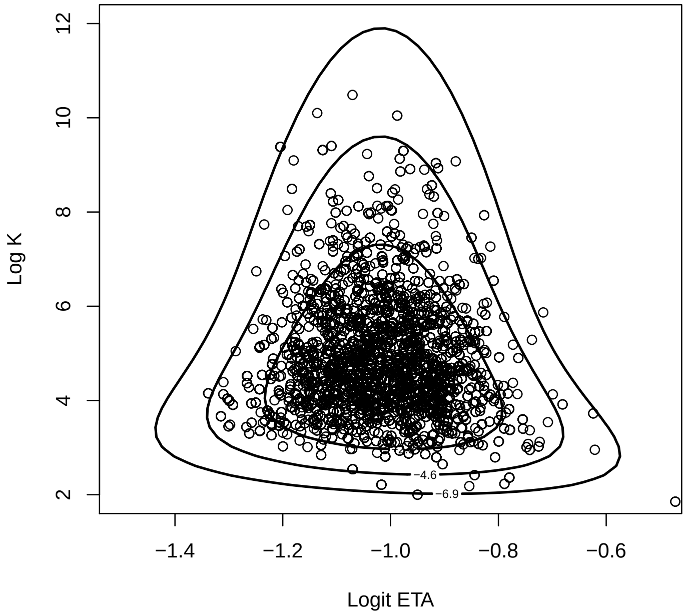
```
> library(LearnBayes)
> laplace.fit <- laplace(betabinexch,
+                         c(0, 0),
+                         d[, c("Hits", "At.Bats")])
> laplace.fit

$mode
[1] -1.013462  4.436169

$var
             [,1]         [,2]
[1,]  0.009556755 -0.005509578
[2,] -0.005509578  0.702918191

$int
[1] -474.3837

$converge
[1] TRUE
```

The outputs from `laplace` are used to inform the inputs of a random walk Metropolis algorithm in the function `rwmetrop`. The inputs are the function defining the log posterior, the estimate of the variance-covarance matrix and scale for the proposal density, the starting value in the Markov Chain, and the data.

```
> mcmc.fit <- rwmetrop(betabinexch,
+                       list(var=laplace.fit$var, scale=2),
+                       c(0, 0),
+                       5000,
+                       d[, c("Hits", "At.Bats")])
```

To demonstrate that this MCMC algorithm produces a reasonable sample from the posterior, the `mycontour` function displays a contour graph of the exact posterior density and the `points` function is used to overlay 5000 draws from the MCMC algorithm.

```
> mycontour(betabinexch, c(-1.5, -0.5, 2, 12),
+           d[, c("Hits", "At.Bats")],
+           xlab="Logit ETA", ylab="Log K")
> with(mcmc.fit, points(par))
```



Simulation of the Posterior of the Probabilities

One can simulate from the joint posterior of $(\{p_j\}, \eta, K)$, by (1) simulating (η, K) from its marginal posterior, and (2) simulating $p_1, ..., p_{18}$ from the conditional distribution $[\{p_j\}|\eta, K]$. In the R script, I store the simulated draws from the posterior of K and η in the vectors `K` and `eta`. Then the function `p.estimate` simulates draws from the posterior of the jth probability and computes a 90% probability interval by extracting the 5th and 95th percentiles. I repeat this process for all 18 players by the `sapply` function and display the 90% intervals for all players.

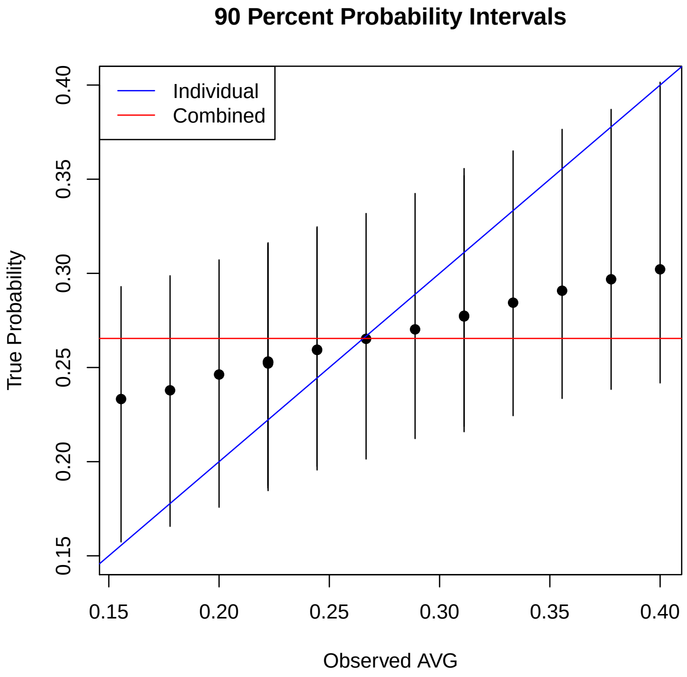
```
> eta <- with(mcmc.fit, exp(par[, 1]) / (1 + exp(par[, 1])))
> K <- exp(mcmc.fit$par[, 2])
> p.estimate <- function(j, eta, K){
+   yj <- d[j, "Hits"]
+   nj <- d[j, "At.Bats"]
+   p.sim <- rbeta(5000, yj + K * eta, nj - yj + K * (1 - eta))
+   quantile(p.sim, c(0.05, 0.50, 0.95))
+ }
> E <- t(sapply(1:18, p.estimate, eta, K))
> rownames(E) <- d[, "Name"]
> round(E, 3)
```

```
              5%   50%   95%
Clemente   0.242 0.302 0.401
Robinson   0.238 0.297 0.387
Howard     0.234 0.291 0.376
Johnstone  0.224 0.284 0.365
Berry      0.219 0.277 0.356
Spencer    0.216 0.278 0.352
Kessinger  0.212 0.270 0.342
Alvarado   0.201 0.265 0.332
Santo      0.196 0.259 0.325
Swaboda    0.197 0.259 0.324
Petrocelli 0.187 0.252 0.316
Rodriguez  0.186 0.253 0.316
Scott      0.186 0.253 0.316
Unser      0.185 0.253 0.316
Williams   0.187 0.253 0.314
Campaneris 0.176 0.246 0.307
Munson     0.166 0.238 0.299
Alvis      0.157 0.233 0.293
```

The following graph displays the 90 percent probability intervals for the players' true batting averages. The blue line represents *individual estimates* where each batting probability is estimated by the observed batting average. The red line represents the *combined estimate* where one combines all of the data. The multilevel estimate represented by the dot is a compromise between the individual estimate and the combined estimate.

```
> plot(d$Hits / 45, E[, 2], pch=19,
+     ylim=c(.15, .40),
+     xlab="Observed AVG", ylab="True Probability",
+     main="90 Percent Probability Intervals")
> for (j in 1:18)
```

```
+   lines(d$Hits[j] / 45 * c(1, 1), E[j, c(1, 3)])
> abline(a=0, b=1, col="blue")
> abline(h=mean(d$Hits) / 45, col="red")
> legend("topleft", legend=c("Individual", "Combined"),
+         lty=1, col=c("blue", "red"))
```


90 Percent Probability Intervals
Individual
Combined
True Probability
Observed AVG