Hierarchical Binomial Model of Bicycle Traffic

Greg Johnson

A city is trying to estimate the proportion of bicycle traffic it has. It collects the following counts of vehicles from 10 locations - bicycles are differentiated from other vehicles.

```
dat = data.frame(
  nbike = c(16,9,10,13,19,20,18,17,35,55),
  nother = c(58,90,48,57,103,57,86,112,273,64)
)
y = dat$nbike
n = with(dat,nbike+nother)
```

We will model the data as separate binomial variables where the number of bicycles is y_j, the total number of variables is n_j, and the underlying (true) proportion of bicycles is θ_j for $j = 1, ..., 10$.

$$y_j|\theta_j \sim \text{Bin}(\theta_j, n_j)$$

Further, we model a beta prior for θ_j with a noninformative hyperprior. We can now analytically derive the joint posterior of the full hierarchical model.

Since (y_j, θ_j, n_j) are exchangeable, we can write the likelihood of $\boldsymbol{\theta}$ as the product of binomial densities:

$$L(\boldsymbol{\theta}) = p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{j=1}^{10} p(y_j|\theta_j) = \prod_{j=1}^{10} \text{Bin}(\theta_j, n_j) \propto \prod_{j=1}^{10} \theta_j^{y_j}(1-\theta_j)^{n_j-y_j}$$

Conceptually we think of the θ_j's as i.i.d. samples from a superpopulation parameterized by hyperpriors α, β.

$$\theta_j|\alpha, \beta \sim \text{Beta}(\alpha, \beta)$$

$$p(\boldsymbol{\theta}|\alpha, \beta) = \prod_{j=1}^{10} p(\theta_j|\alpha, \beta) = \prod_{j=1}^{10} \text{Beta}(\alpha, \beta) \propto \prod_{j=1}^{10} \theta_j^{\alpha-1}(1-\theta_j)^{\beta-1}$$

Finally, we assign a noninformative hyperprior:

$$p(\alpha, \beta \propto (\alpha+\beta)^{-5/2}$$

Now we can derive the full posterior (up to a constant) as the product of the likelihood, the prior, and the hyperprior.

$$p(\alpha, \beta, \boldsymbol{\theta}|\mathbf{y}) \propto L(\alpha, \beta, \boldsymbol{\theta})p(\alpha, \beta, \boldsymbol{\theta}) = L(\alpha, \beta, \boldsymbol{\theta})p(\boldsymbol{\theta}|\alpha, \beta)p(\alpha, \beta)$$

$$p(\alpha, \beta, \boldsymbol{\theta}|\mathbf{y}) \propto \prod_{j=1}^{10} \theta_j^{\alpha+y_j-1}(1-\theta_j)^{\beta+n_j-y_j-1}$$

Let's derive the marginal posterior of hyperparameters and draw simulations from the joint posterior. Since this is a conjugate model, derivation of the marginal posterior of hyperparameters is pretty easy. First we want the conditional posterior of $\boldsymbol{\theta}$ which, again thanks to conjugacy, is also a product of β densities:

$$p(\boldsymbol{\theta}|\alpha, \beta, \mathbf{y}) = \prod_{j=1}^{10} \text{Beta}(\alpha + y_j, \beta + n_j - y_j)$$

Now we can use the conditional probability formula:

$$p(\alpha, \beta|\mathbf{y}) = \frac{p(\boldsymbol{\theta}, \alpha, \beta|\mathbf{y})}{p(\boldsymbol{\theta}|\alpha, \beta, \mathbf{y})}$$

The terms to do with $\boldsymbol{\theta}$ cancel out neatly, leaving the hyperprior and a ratio of the normalizing constant of the Beta densities for the $\boldsymbol{\theta}$ prior and the $\boldsymbol{\theta}$ posterior:

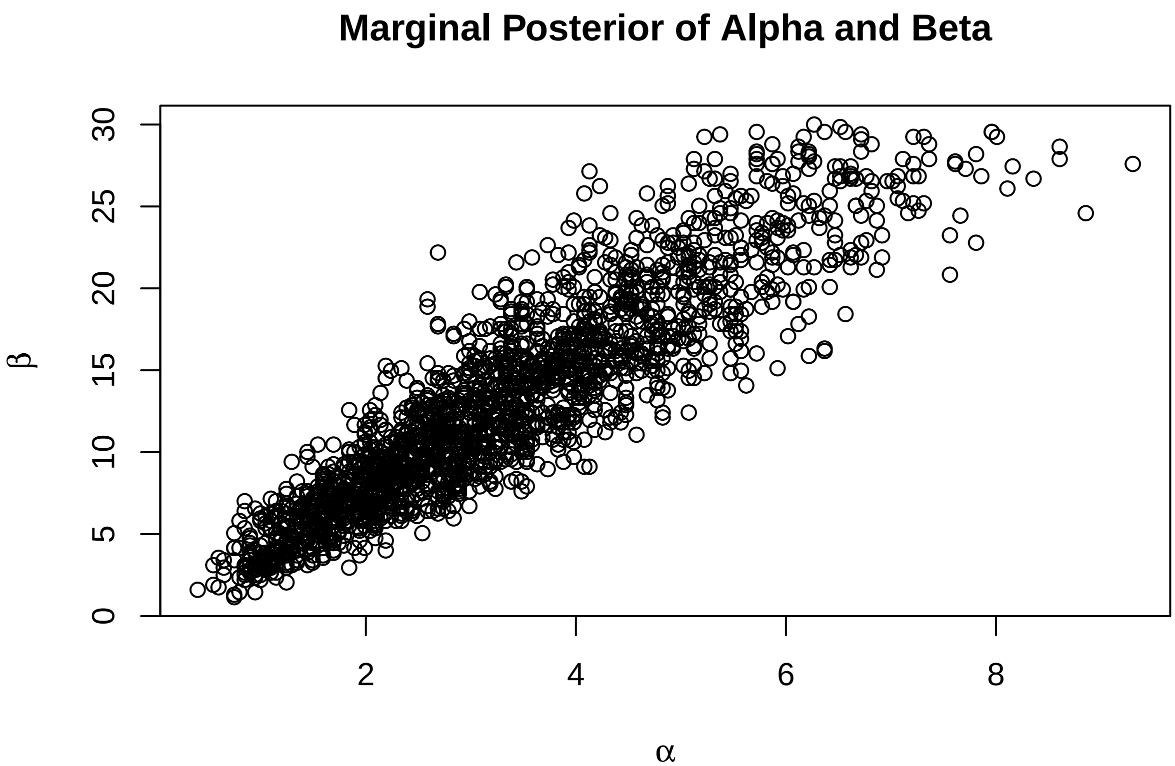
$$p(\alpha, \beta|\mathbf{y}) \propto (\alpha + \beta)^{-5/2} \prod_{j-1}^{10} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(\alpha + y_j)\Gamma(\beta + n_j - y_j)}{\Gamma(\alpha + \beta + n_j)}$$

Sample from Posterior

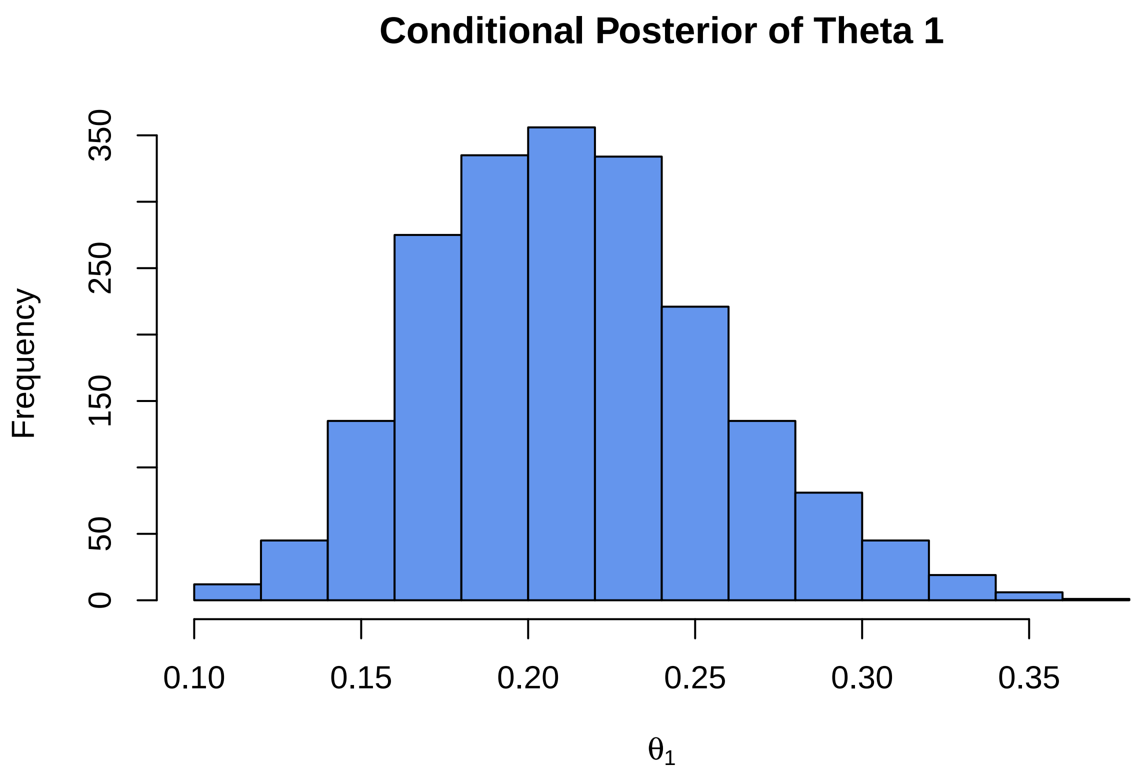
Our sampling strategy will employ a basic grid-sampling technique for the marginal posterior for the hyperparameters and sampling of θ_j's conditional on the hyperparameter draw.

```
grid_size=200
agrid = seq(.1,10,length.out=grid_size)
bgrid = (seq(.1,30,length.out=grid_size))
post_grid = matrix(0,grid_size,grid_size)
post_density = function(alpha,beta){
  exp(
      -5/2*log(alpha+beta) +
      10*lgamma(alpha+beta) -
      10*lgamma(alpha) -
      10*lgamma(beta) +
      sum(lgamma(alpha+y)) +
      sum(lgamma(beta+n-y)) -
      sum(lgamma(alpha+beta+n))
  )
}
for(i in 1:grid_size){
  for(j in 1:grid_size){
    alpha = agrid[j]
    beta = rev(bgrid)[i]
    post_grid[i,j] = post_density(alpha,beta)
  }
}
post_grid_norm = post_grid/sum(post_grid)


##sample 2000 hyperparameters from marginal posterior
ndraws = 2000
post_marg = matrix(NA,ndraws,2,dimnames=list(NULL,c("alpha","beta")))
for(j in 1:ndraws){
  #sample alpha
  alpha_draw = sample(agrid,1,prob=(apply(post_grid_norm,2,sum)))
  post_marg[j,1] = alpha_draw
```

```
  #sample beta/alpha
  beta_cond = post_grid_norm[,which(alpha_draw==agrid)]
  beta_draw = sample(rev(bgrid),1,prob=beta_cond/sum(beta_cond))
  post_marg[j,2] = beta_draw
}

plot(post_marg[,1],post_marg[,2],xlab = expression(alpha),ylab=expression(beta),main="Marginal Posterio
```




```
#sample 2000 theta vectors from conditional posterior
post_cond = matrix(NA,ndraws,10,dimnames=list(NULL,paste("theta",1:10)))
for(j in 1:ndraws){
  for(k in 1:10){
    alpha = post_marg[j,1]
    beta = post_marg[j,2]
    post_cond[j,k] = rbeta(1,alpha+y[k],beta+n[k]-y[k])
  }
}

hist(post_cond[,1],xlab = expression(theta[1]),
     main="Conditional Posterior of Theta 1",col="cornflowerblue")
```



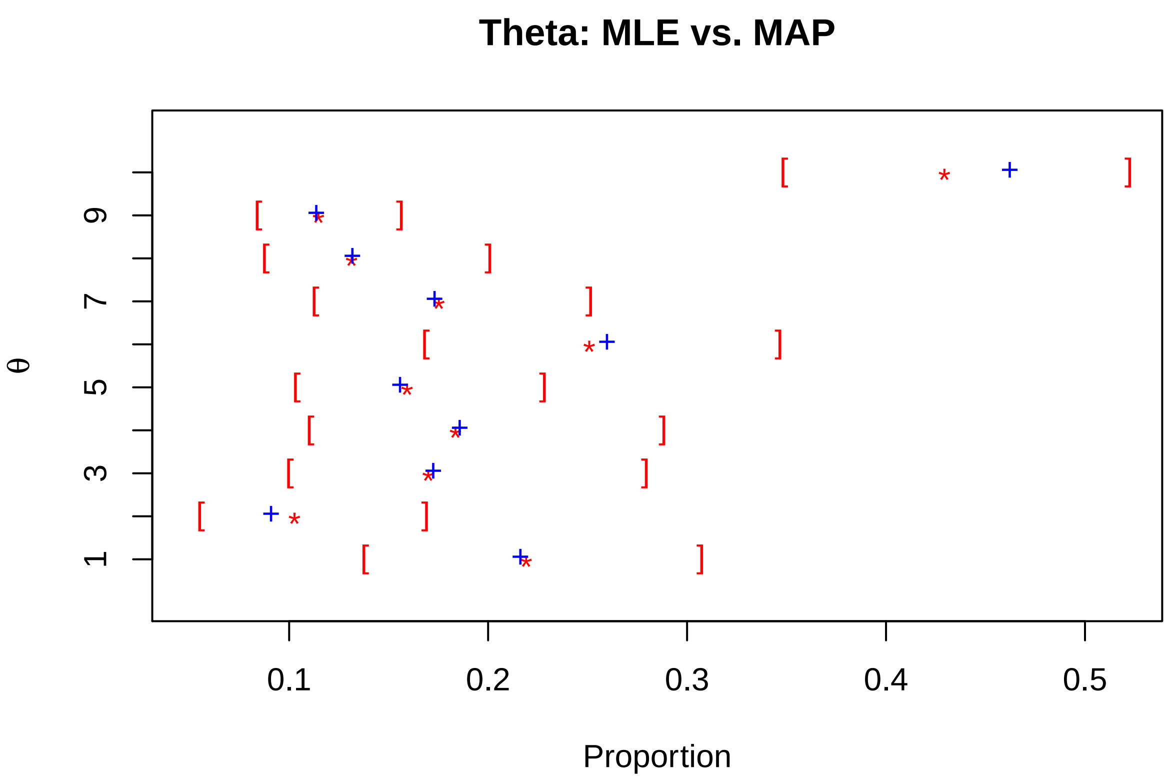
Let's compare the posterior distribution of $\boldsymbol{\theta}$ to the raw proportions $\hat{\boldsymbol{\theta}} = \boldsymbol{y}/\boldsymbol{n}$?

```
CredInt95 = matrix(NA,12,3,dimnames=list(c("alpha","beta",paste("theta",1:10)),c("LB","UB","MAP")))

#alpha
CredInt95[1,1:2] = quantile(post_marg[,1],c(.025,.975))
  d = density(post_marg[,1])
  i = which.max(d$y)
CredInt95[1,3] = d$x[i]

#beta
CredInt95[2,1:2] = quantile(post_marg[,2],c(.025,.975))
  d = density(post_marg[,2])
  i = which.max(d$y)
CredInt95[2,3] = d$x[i]



for(j in 1:10){
  CredInt95[j+2,1:2] = quantile(post_cond[,j],c(.025,.975))
    d = density(post_cond[,j])
    i = which.max(d$y)
  CredInt95[j+2,3] = d$x[i]
}

round(CredInt95)
```

```
##          LB UB MAP
## alpha     1  7   3
## beta      3 27  10
```

```
## theta 1   0  0   0
## theta 2   0  0   0
## theta 3   0  0   0
## theta 4   0  0   0
## theta 5   0  0   0
## theta 6   0  0   0
## theta 7   0  0   0
## theta 8   0  0   0
## theta 9   0  0   0
## theta 10  0  1   0
```

```
prop_mle = y/n

plot(CredInt95[3:12,3], 1:10, pch = "*", col = "red",
     ylim = c(0,11), xlim = c(.05,.52), yaxt = "n",
     xlab = "Proportion", ylab = expression(theta),
     main = "Theta: MLE vs. MAP")
points(CredInt95[3:12,1],1:10,pch="[",col="red")
points(CredInt95[3:12,2],1:10,pch="]",col="red")
points(prop_mle,1:10,pch="+",col="blue")
axis(2, at=1:10,labels=1:10)
```



Most of the MLE's are quite close to the MAP estimates (and never stray from the 95% credible interval) - of course the discrepancies are due to the beta prior that we fit to the thetas.

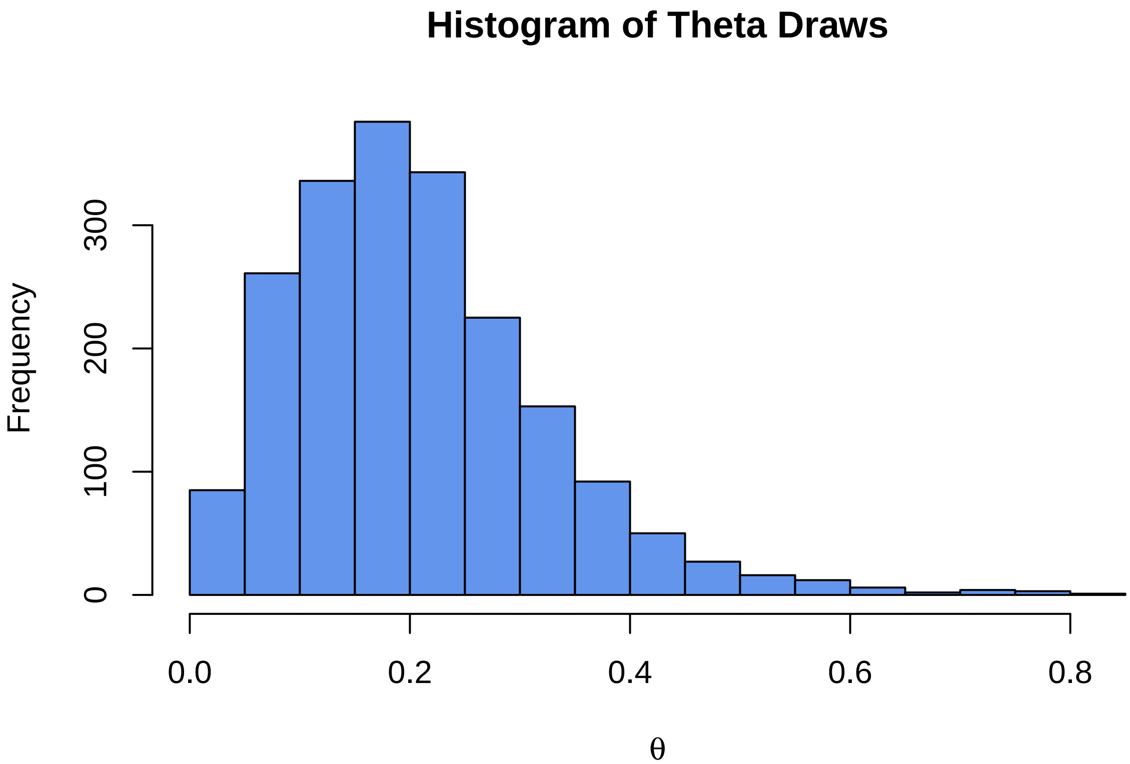
Let's get a 95% posterior interval for the average underlying proportion of traffic that is bicycles. We will sample 2000 θ's from the (α, β) posterior draws we took.

```
theta_draws = rbeta(ndraws,post_marg[,1],post_marg[,2])
quantile(theta_draws,c(.025,.975))
```

```
##       2.5%      97.5%
```

```
## 0.03823229 0.48040323
```

```r
hist(theta_draws,xlab=expression(theta),
     main="Histogram of Theta Draws",col="cornflowerblue")
```



Say a new city block is sampled with 100 vehicles in an hour. Let's get a 95% posterior interval for the number of those vehicles that are bicycles. To simulate numbers of bicycles we can just take the simulated θ's, which are bicycle proportions, and multiply them by $n = 100$.

```r
quantile(theta_draws*100,c(.025,.975))
```

```
##      2.5%     97.5%
##  3.823229 48.040323
```

I trust this interval as much as its probabilistic interpretation allows: there is a 95% chance that the number of bicycles for this new city block will fall in the above interval. This confidence, though, is conditional on the accuracy of our full probability model - any departure of the actual data from our likelihood, prior, or hyperprior will reduce my trust.

The beta distribution has proven to be excellent for the θ_j's since its support perfectly matches the parameter space that θ is restricted to. Further, once the data have been observed, the hyperparameters adjust their posterior distribution to the observed data and the posterior beta distribution for the θ_j's adapts. Really, the only restriction necessary is to make sure the Beta distribution doesn't assign zero-probability to any portion of $[0, 1]$.