STA 360/602L: Module 6.2

Bayesian linear regression (illustration)

Dr. Olanrewaju Michael Akande

Swimming data

- Back to the swimming example. The data is from Exercise 9.1 in Hoff.
- The data set we consider contains times (in seconds) of four high school swimmers swimming 50 yards.

```
Y <- read.table("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/swim.dat")
Y
```

```
##     V1   V2   V3   V4   V5   V6
## 1 23.1 23.2 22.9 22.9 22.8 22.7
## 2 23.2 23.1 23.4 23.5 23.5 23.4
## 3 22.7 22.6 22.8 22.8 22.9 22.8
## 4 23.7 23.6 23.7 23.5 23.5 23.4
```

- There are 6 times for each student, taken every two weeks. That is, each swimmer has six measurements at $t = 2, 4, 6, 8, 10, 12$ weeks.
- Each row corresponds to a swimmer and a higher column index indicates a later date.

Swimming data

- Given that we don't have enough data, we can explore hierarchical models. That way, we can borrow information across swimmers.

- For now, however, we will fit a separate linear regression model for each swimmer, with swimming time as the response and week as the explanatory variable (which we will mean center).

- For setting priors, we have one piece of information: times for this age group tend to be between 22 and 24 seconds.

- Based on that, we can set uninformative parameters for the prior on σ^2 and for the prior on $\boldsymbol{\beta}$, we can set

$$\pi(\boldsymbol{\beta}) = \mathcal{N}_2\left(\boldsymbol{\beta}_0 = \begin{pmatrix} 23 \\ 0 \end{pmatrix}, \Sigma_0 = \begin{pmatrix} 5 & 0 \\ 0 & 2 \end{pmatrix}\right).$$

- This centers the intercept at 23 (the middle of the given range) and the slope at 0 (so we are assuming no increase) but we choose the variance to be a bit large to err on the side of being less informative.

Posterior computation

```
#Create X matrix, transpose Y for easy computayion
Y <- t(Y)
n_swimmers <- ncol(Y)
n <- nrow(Y)
W <- seq(2,12,length.out=n)
X <- cbind(rep(1,n),(W-mean(W)))
p <- ncol(X)

#Hyperparameters for the priors
beta_0 <- matrix(c(23,0),ncol=1)
Sigma_0 <- matrix(c(5,0,0,2),nrow=2,ncol=2)
nu_0 <- 1
sigma_0_sq <- 1/10

#Initial values for Gibbs sampler
#No need to set initial value for sigma^2, we can simply sample it first
beta <- matrix(c(23,0),nrow=p,ncol=n_swimmers)
sigma_sq <- rep(1,n_swimmers)

#first set number of iterations and burn-in, then set seed
n_iter <- 10000; burn_in <- 0.3*n_iter
set.seed(1234)

#Set null matrices to save samples
BETA <- array(0,c(n_swimmers,n_iter,p))
SIGMA_SQ <- matrix(0,n_swimmers,n_iter)
```

Posterior computation

```
#Now, to the Gibbs sampler
#library(mvtnorm) for multivariate normal

#first set number of iterations and burn-in, then set seed
n_iter <- 10000; burn_in <- 0.3*n_iter
set.seed(1234)

for(s in 1:(n_iter+burn_in)){
  for(j in 1:n_swimmers){

    #update the sigma_sq
    nu_n <- nu_0 + n
    SSR <- t(Y[,j] - X%*%beta[,j])%*%(Y[,j] - X%*%beta[,j])
    nu_n_sigma_n_sq <- nu_0*sigma_0_sq + SSR
    sigma_sq[j] <- 1/rgamma(1,(nu_n/2),(nu_n_sigma_n_sq/2))

    #update beta
    Sigma_n <- solve(solve(Sigma_0) + (t(X)%*%X)/sigma_sq[j])
    mu_n <- Sigma_n %*% (solve(Sigma_0)%*%beta_0 + (t(X)%*%Y[,j])/sigma_sq[j])
    beta[,j] <- rmvnorm(1,mu_n,Sigma_n)

    #save results only past burn-in
    if(s > burn_in){
      BETA[j,(s-burn_in),] <- beta[,j]
      SIGMA_SQ[j,(s-burn_in)] <- sigma_sq[j]
    }
  }
}
```

Results

- Before looking at the posterior samples, what are the OLS estimates for all the parameters?

```
beta_ols <- matrix(0,nrow=p,ncol=n_swimmers)
for(j in 1:n_swimmers){
beta_ols[,j] <- solve(t(X)%*%X)%*%t(X)%*%Y[,j]
}
colnames(beta_ols) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
rownames(beta_ols) <- c("beta_0","beta_1")
beta_ols
```

```
##          Swimmer 1   Swimmer 2 Swimmer 3   Swimmer 4
## beta_0 22.93333333 23.35000000  22.76667 23.56666667
## beta_1 -0.04571429  0.03285714   0.02000 -0.02857143
```

- Give an interpretation for the parameters.
- Any thoughts on who the coach should recommend based on this alone?
- Is this how we should be answering the question?

Posterior inference

- Posterior means are almost identical to OLS estimates.

```
beta_postmean <- t(apply(BETA,c(1,3),mean))
colnames(beta_postmean) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
rownames(beta_postmean) <- c("beta_0","beta_1")
beta_postmean
```

```
##         Swimmer 1   Swimmer 2   Swimmer 3   Swimmer 4
## beta_0 22.9339174 23.34963191 22.76617785 23.56614309
## beta_1 -0.0453998  0.03251415  0.01991469 -0.02854268
```

- How about credible intervals?

```
beta_postCI <- apply(BETA,c(1,3),function(x) quantile(x,probs=c(0.025,0.975)))
colnames(beta_postCI) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
beta_postCI[,,1]; beta_postCI[,,2]
```

```
##       Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
## 2.5%   22.76901  23.15949  22.60097  23.40619
## 97.5%  23.09937  23.53718  22.93082  23.73382
```

```
##          Swimmer 1   Swimmer 2   Swimmer 3   Swimmer 4
## 2.5%  -0.093131856 -0.02128792 -0.02960257 -0.07704344
## 97.5%  0.002288246  0.08956464  0.06789081  0.01940960
```

- Is there any evidence that the times matter?

Posterior inference

- Is there any evidence that the times matter?

```
beta_pr_great_0 <- t(apply(BETA,c(1,3),function(x) mean(x > 0)))
colnames(beta_pr_great_0) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
beta_pr_great_0
```

```
##      Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
## [1,]    1.0000    1.0000    1.0000    1.0000
## [2,]    0.0287    0.9044    0.8335    0.0957
```

```
#or alternatively,
beta_pr_less_0 <- t(apply(BETA,c(1,3),function(x) mean(x < 0)))
colnames(beta_pr_less_0) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
beta_pr_less_0
```

```
##      Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
## [1,]    0.0000    0.0000    0.0000    0.0000
## [2,]    0.9713    0.0956    0.1665    0.9043
```

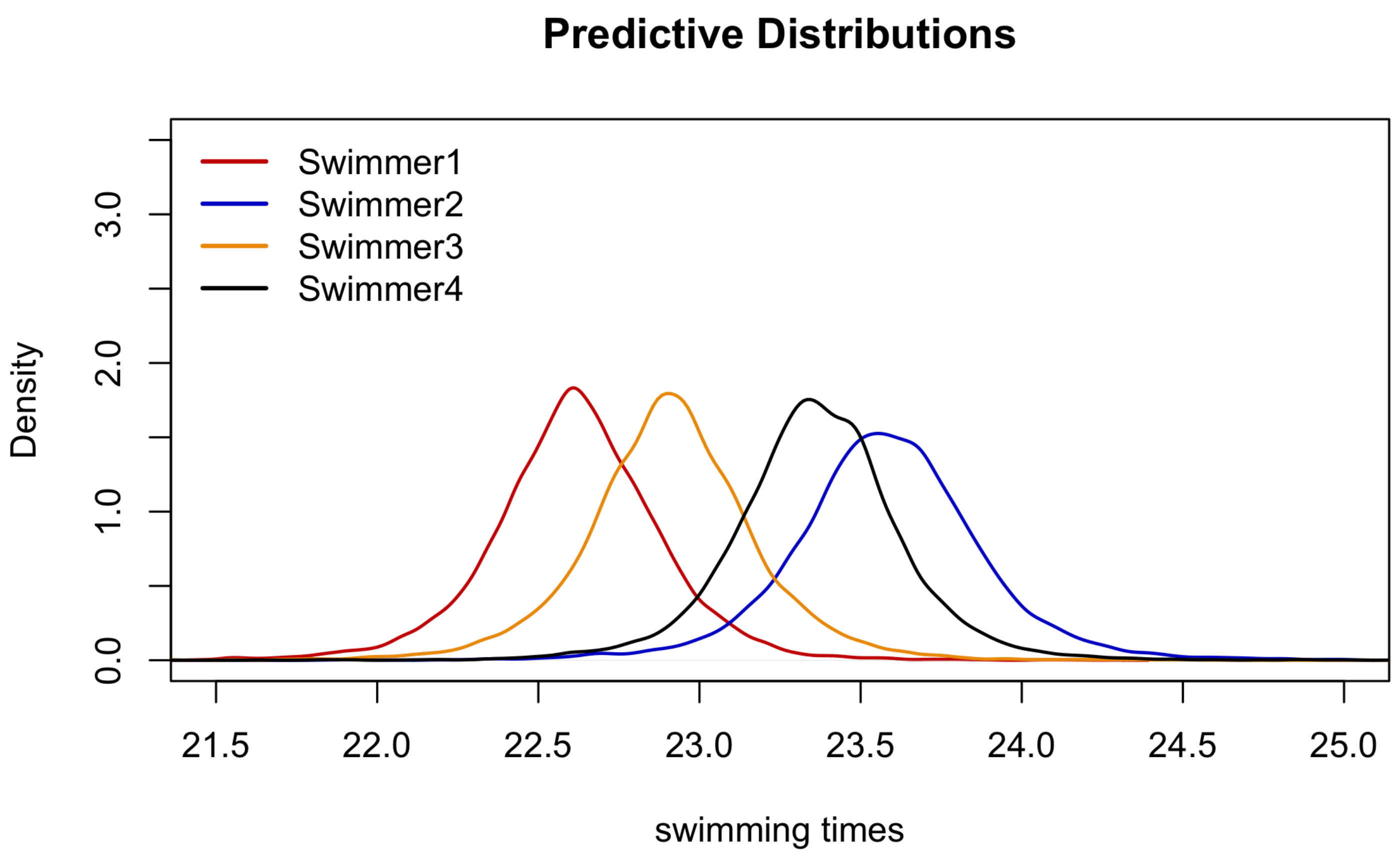
Posterior predictive inference

- How about the posterior predictive distributions for a future time two weeks after the last recorded observation?

```
x_new <- matrix(c(1,(14-mean(W))),ncol=1)
post_pred <- matrix(0,nrow=n_iter,ncol=n_swimmers)
for(j in 1:n_swimmers){
post_pred[,j] <- rnorm(n_iter,BETA[j,,]%*%x_new,sqrt(SIGMA_SQ[j,]))
}
colnames(post_pred) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")

plot(density(post_pred[,"Swimmer 1"]),col="red3",xlim=c(21.5,25),ylim=c(0,3.5),lwd=1.5
 main="Predictive Distributions",xlab="swimming times")
legend("topleft",2,c("Swimmer1","Swimmer2","Swimmer3","Swimmer4"),col=c("red3","blue3"
lines(density(post_pred[,"Swimmer 2"]),col="blue3",lwd=1.5)
lines(density(post_pred[,"Swimmer 3"]),col="orange2",lwd=1.5)
lines(density(post_pred[,"Swimmer 4"]),lwd=1.5)
```

Posterior predictive inference



Posterior predictive inference

- How else can we answer the question on who the coach should recommend for the swim meet in two weeks time? Few different ways.

- Let $Y_j^\star$ be the predicted swimming time for each swimmer j. We can do the following: using draws from the predictive distributions, compute the posterior probability that $P(Y_j^\star = \min(Y_1^\star, Y_2^\star, Y_3^\star, Y_4^\star))$ for each swimmer j, and based on this make a recommendation to the coach.

- That is,

```
post_pred_min <- as.data.frame(apply(post_pred,1,function(x) which(x==min(x))))
colnames(post_pred_min) <- "Swimmers"
post_pred_min$Swimmers <- as.factor(post_pred_min$Swimmers)
levels(post_pred_min$Swimmers) <- c("Swimmer 1","Swimmer 2","Swimmer 3","Swimmer 4")
table(post_pred_min$Swimmers)/n_iter
```

```
##
## Swimmer 1 Swimmer 2 Swimmer 3 Swimmer 4
##    0.7790    0.0078    0.1994    0.0138
```

- Which swimmer would you recommend?

What's next?

Move on to the readings for the next module!