Binomial with Beta Prior Sampling Ranking Simulation

Cora Allen-Coleman

9/20/2017

Introduction

If we have a list of baseball players, what is the best way to rank them by batting average? We can estimate the probability that each player is ranked at each position: "What is player i's probability of being ranked nth?"

If n is the number of items to be ranked (baseball players here) then:

$P_i = \text{Player}_i$ where $i = 1, ..., n$
$r_j = \text{Rank}_j$ where $j = 1, ..., n$
Observed Successes: $\hat{x}_i$
Observed Batting Attempts: $\hat{a}_i$
Observed Batting Averages: $\hat{b}_i \sim \text{Binomial}(b_i, a_i)$ where b_i is the player's true batting average parameter
$\theta_{i,j} = P(P_i \text{ is ranked at position j } |\tilde{x},\ \text{prior})$

To estimate $\theta_{i,n}$, we'll sample from the distribution of possible batting averages for each player, using a Beta(α, β) prior.

data/likelihood: $\hat{b}_i \sim \text{Binomial}(b_i, \hat{a}_i)$
prior: $\text{Beta}(\alpha, \beta) = \text{Beta}(0.5, 0.5)$
posterior for sampled batting averages: $\tilde{b}_i \sim \text{Beta}(\alpha + \hat{x}_i, \beta + \hat{a}_i - \hat{x}_i)$

Procedure:

1. Sample a vector of sampled batting averages, one for each player: $(\tilde{b}_1, \tilde{b}_2, ..., \tilde{b}_n)$
2. Record the of the players using their sampled $\tilde{b}_i$: $(\tilde{r}_1, \tilde{r}_1, ..., \tilde{r}_n)$
3. Repeat steps 1 and 2 10,000 times. CHECK
4. Estimate the probability that each player is in each position $\theta_{i,j}$.

Create a matrix of probabilities:

$$\tilde{\theta}_{i,j} = \frac{\text{Count of times player i is at spot j}}{\text{total simulations}}$$

Choose a position for each player.
How do you choose the position for each player? Do we go spot by spot (over j), so that the player with the higher $\tilde{\theta}_{i,j}$ gets spot j from rank 1 to rank n? Or in the other direction? How do we deal with ties?
For this initial attempt, I will use this technique:
Start at rank 1 (j = 1). Choose the player with high probability of being at rank 1 (largest $\theta_{i,1}$). If there are ties, choose the player who comes first in the list. Once a player has been chosen for a rank, they can't be chosen for a following rank. Continue for ranks 2 through n.

Does this technique prioritize the correctness of early ranks (1-3) over later ranks?

```
## Loading required package: plyr
```

```
## Loading required package: ggplot2
```

Computer Model Implementation

Step 1: Sample batting averages for each player 10,000 times.

a. Create an nxn matrix of sampled rank counts full of zeros for all players

```
samples <- 20000
#stores ranks with nrow = #samples and col = players
ranks <- as.data.frame(matrix(rep(0, n*samples), nrow = samples, ncol = n)) #rows = samples, columns=ra
names(ranks) <- c("Player1", "Player2", "Player3", "Player4", "Player5", "Player6", "Player7", "Player8
rankCounts <- as.data.frame(matrix(rep(0, n*n), nrow = n, ncol = n)) #rows = samples, columns=rank posi
names(rankCounts) <- c("Rank1", "Rank2", "Rank3", "Rank4", "Rank5", "Rank6", "Rank7", "Rank8", "Rank9",
```

b. Sample a $\tilde{b}_i$ vector for each player.

```
set.seed(1)
alpha <- .5
beta <- .5
```

c. Store the rank of the batting averages for each of the samples.

```
for (sample in 1:samples){
  batAvg <- c()
  for (i in 1:n){
    batAvg[i] <- rbeta(1, alpha + batData$X[i], beta + batData$n[i] - batData$X[i])
  }
  #ranks of the each of the players for this vector
  ranks[sample,] <- rank(batAvg)
}
```

d. Count the number of times each player ranked at each position in the rankCount matrix.

```
for (player in 1:n){
    df <- as.data.frame(count(ranks[,player],))
    for (rank in 1:nrow(df)){
      rankCounts[player, df$x[rank]] <- df$freq[rank]
    }
}
```

Step 2: Calculate the probability that each player is in each position.

a. Create a matrix of where:

$$\tilde{\theta}_{i,j} = \frac{\text{Count of times player i sampled at position j}}{\text{total simulations}}$$

```
rankProbs <- rankCounts/samples
```

b. Starting at rank 1, choose the player with the higher probability of being in that position. Store them at that rank and their probability of being in that position. Set their probabilities to zero so that they can't be chosen for later ranks.

```
estimatedRanks <- as.data.frame(matrix(rep(0, 2*n), nrow = 2, ncol = n)) #rows = samples, columns=rank
names(estimatedRanks) <- c("Rank1", "Rank2", "Rank3", "Rank4", "Rank5", "Rank6", "Rank7", "Rank8", "Ran

for (position in 1:n){
  estimatedRanks[1,position] <- which.max(rankProbs[,position])
  estimatedRanks[2,position] <- rankProbs[which.max(rankProbs[,position]), position] #store prob
```

```
  #replace this player's row with 00s
  rankProbs[which.max(rankProbs[,position]), ] <- rep(0, n)
}
rankProbs <- rankCounts/samples #fixes rankProb
estimatedRanks #rows = players
```

```
##     Rank1   Rank2  Rank3   Rank4   Rank5   Rank6 Rank7   Rank8  Rank9
## 1 12.0000 10.0000 9.0000 8.00000 7.00000 6.00000 5.000 2.00000 3.0000
## 2  0.3124  0.1659 0.1294 0.13705 0.14545 0.12225 0.103 0.10315 0.1369
##   Rank10  Rank11  Rank12
## 1 4.0000 1.00000 11.0000
## 2 0.1399 0.18355  0.0163
```

Step 3 Calculate Variance

```
varianceEstimates <- c(rep(0, n))
for (player in 1:n){
  meanRank <- which(estimatedRanks[1,] == player)
  for (sampledRank in 1:n){
    varianceEstimates[player] <- varianceEstimates[player] + rankCounts[player, sampledRank]*((sampledRa
  }
}
varianceEstimates <- varianceEstimates/samples; varianceEstimates #by player
```

```
##  [1]  7.39990  6.12330  5.45550  9.90580  9.88070  7.18475  5.98910
##  [8]  7.49200 10.81905 10.46170 71.78280  6.44750
```

Step 4: Sampling Size

How many times to sample? Decide by comparing variance estimates by sampling sizes.

Notes for Discussion:
- What are reasonable choices for alpha and beta? Do I want my expectation to be 0.5? or 1/n? E = alpha/(alpha + beta)
- How do you choose the position for each player? Do we go spot by spot (over j), so that the player with the higher $\theta_{i,j}$ gets spot j from rank 1 to rank n? Or in the other direction? How do we deal with ties?
- Do these variance estimates make sense?

To Do:

- Graph variance over sampling sizes