Predicting OBP Using Bayesian Analysis

Paulina Leperi

2023-05-13

The data used in this analysis was downloaded from https://www.seanlahman.com/baseball-archive/statistics/update (https://www.seanlahman.com/baseball-archive/statistics/update).

On-base percentage (OBP) for MLB players regresses to the mean, so players who have an exceptionally good season are likely to be closer to the league average the following year, and the same is true of players who have exceptionally bad years. Below, I will attempt to quantify this effect using Bayesian analysis.

The beta distribution is the conjugate prior for binomial distributions, so using it will allow for straightforward inference. I fit a beta-binomial distribution to give more weight to players with more plate appearances.

I'll try out three different priors, using players who were active in 2022 and batted in more games than they pitched as a test dataset. I'll predict their 2022 OBP using their career stats from previous years and determine the best-performing prior using RMSE.

The first chunk of code gets the 2022 OBP and previous career OBP for the players in the test data.

```
test_data_players <- Batting %>%
  filter(yearID == 2022 & (AB+BB+HBP+SF) > 0) %>%
  left_join(Pitching %>%
              filter(yearID == 2022) %>%
              dplyr::select(playerID, stint, pitcher_games = G),
            by = c("playerID", "stint")) %>%
  mutate(pitcher_games = replace_na(pitcher_games, 0)) %>%
  filter(G > pitcher_games) %>%
  group_by(playerID) %>%
  summarize(gets_on_base = sum(H+BB+HBP),
            PA = sum(AB+BB+HBP+SF),
            OBP = gets_on_base/PA)

test_data <- Batting %>%
  filter(yearID < 2022 & (AB+BB+HBP+SF) > 0) %>%
  group_by(playerID) %>%
  summarize(prior_gets_on_base = sum(H+BB+HBP),
            prior_PA = sum(AB+BB+HBP+SF)) %>%
  right_join(test_data_players, by = "playerID") %>%
  mutate(prior_gets_on_base = replace_na(prior_gets_on_base, 0),
         prior_PA = replace_na(prior_PA, 0))
```

I want to double-check that I haven't filtered out Shohei Ohtani, who was both a pitcher and a hitter in 2022.

```
subset(People, nameFirst=='Shohei' & nameLast=='Ohtani')
```

```
## # A tibble: 1 × 24
##   playerID  birthYear birthMonth birthDay birthCountry birthState birthCity
##   <chr>         <dbl>      <dbl>    <dbl> <chr>        <chr>      <chr>
## 1 ohtansh01      1994          7        5 Japan        Iwate      Oshu
## # ℹ 17 more variables: deathYear <dbl>, deathMonth <dbl>, deathDay <dbl>,
## #   deathCountry <chr>, deathState <chr>, deathCity <chr>, nameFirst <chr>,
## #   nameLast <chr>, nameGiven <chr>, weight <dbl>, height <dbl>, bats <chr>,
## #   throws <chr>, debut <date>, finalGame <date>, retroID <chr>, bbrefID <chr>
```

```
shohei <- "ohtansh01"
subset(test_data, playerID == shohei)
```

```
## # A tibble: 1 × 6
##   playerID  prior_gets_on_base prior_PA gets_on_base    PA   OBP
##   <chr>                  <dbl>    <dbl>        <dbl> <dbl> <dbl>
## 1 ohtansh01                566     1603          237   666 0.356
```

Training version 1

For this prior, I will include career batting through 2021 for all hitters since 1901. I replace missing values for HBP and SF with 0; AB, H, and BB are never missing.

```
training1 <- Batting %>%
  filter(yearID >= 1901 & yearID < 2022 & (AB+BB) > 0) %>%
  left_join(Pitching %>%
              dplyr::select(playerID, yearID, stint, pitcher_games = G),
            by = c("playerID", "yearID", "stint")) %>%
  mutate(pitcher_games = replace_na(pitcher_games, 0),
         HBP = replace_na(HBP, 0),
         SF = replace_na(SF, 0)) %>%
  filter(G > pitcher_games) %>%
  group_by(playerID) %>%
  summarize(gets_on_base = sum(H+BB+HBP),
            PA = sum(AB+BB+HBP+SF),
            OBP = gets_on_base/PA)

minusLogLike1 <- function(alpha, beta) {
  -sum(dbetabinom.ab(training1$gets_on_base, training1$PA, alpha, beta, log = TRUE))
}
m1 <- mle(minusLogLike1, start = list(alpha = 1, beta = 1), method = "L-BFGS-B")

test_data %>%
  mutate(pOBP = (prior_gets_on_base + m1@coef[1]) / (prior_PA + m1@coef[1] + m1@coef[2]),
         diff = (pOBP - OBP)**2) %>%
  summarize(MSE = mean(diff),
            RMSE = MSE**.5)
```

```
## # A tibble: 1 × 2
##       MSE   RMSE
##     <dbl>  <dbl>
## 1 0.00690 0.0831
```

The RMSE of this version is .0831.

Training version 2

The second version includes career stats for all hitters active in 2021 in the estimation of the prior.

```
training2 <- Batting %>%
  filter(yearID < 2022) %>%
  group_by(playerID) %>%
  mutate(active = if_else(max(yearID) == 2021, 1, 0)) %>%
  filter((AB+BB+HBP+SF) > 0 & active == 1) %>%
  summarize(G = sum(G),
            gets_on_base = sum(H+BB+HBP),
            PA = sum(AB+BB+HBP+SF),
            OBP = gets_on_base/PA) %>%
  ungroup() %>%
  left_join(Pitching %>%
              group_by(playerID) %>%
              summarize(pitcher_games = sum(G)),
            by = c("playerID")) %>%
  mutate(pitcher_games = replace_na(pitcher_games, 0)) %>%
  filter(G > pitcher_games)

minusLogLike2 <- function(alpha, beta) {
  -sum(dbetabinom.ab(training2$gets_on_base, training2$PA, alpha, beta, log = TRUE))
}
m2 <- mle(minusLogLike2, start = list(alpha = 1, beta = 1), method = "L-BFGS-B")

test_data %>%
  mutate(pOBP = (prior_gets_on_base + m2@coef[1]) / (prior_PA + m2@coef[1] + m2@coef[2]),
         diff = (pOBP - OBP)**2) %>%
  summarize(MSE = mean(diff),
            RMSE = MSE**.5)
```

```
## # A tibble: 1 × 2
##       MSE   RMSE
##     <dbl>  <dbl>
## 1 0.00730 0.0854
```

This version performs slightly worse, with RMSE of .0854.

Training version 3

Finally, the third version is the same as the second, except that it weighs the 2021 season twice as much as prior seasons.

```
training3 <- Batting %>%
  filter(yearID < 2022) %>%
  mutate(gets_on_base = H+BB+HBP,
         PA = AB+BB+HBP+SF,
         w_gets_on_base = if_else(yearID == 2021, gets_on_base * 2, gets_on_base),
         wPA = if_else(yearID == 2021, PA * 2, PA)) %>%
  group_by(playerID) %>%
  mutate(active = if_else(max(yearID) == 2021, 1, 0)) %>%
  filter(PA > 0 & active == 1) %>%
  summarize(G = sum(G),
            gets_on_base = sum(w_gets_on_base),
            PA = sum(wPA),
            OBP = gets_on_base/PA) %>%
  ungroup() %>%
  left_join(Pitching %>%
              group_by(playerID) %>%
              summarize(pitcher_games = sum(G)),
            by = c("playerID")) %>%
  mutate(pitcher_games = replace_na(pitcher_games, 0)) %>%
  filter(G > pitcher_games)

minusLogLike3 <- function(alpha, beta) {
  -sum(dbetabinom.ab(training3$gets_on_base, training3$PA, alpha, beta, log = TRUE))
}
m3 <- mle(minusLogLike3, start = list(alpha = 1, beta = 1), method = "L-BFGS-B")

test_data %>%
  mutate(pOBP = (prior_gets_on_base + m3@coef[1]) / (prior_PA + m3@coef[1] + m3@coef[2]),
         diff = (pOBP - OBP)**2) %>%
  summarize(MSE = mean(diff),
            RMSE = MSE**.5)
```

```
## # A tibble: 1 × 2
##       MSE   RMSE
##     <dbl>  <dbl>
## 1 0.00718 0.0847
```

This version performed better than the previous one, with RMSE of .0847, but not as well as the first one.

Predictions with the winning prior

So which players were predicted to have the highest OBP in 2022 with the first prior?

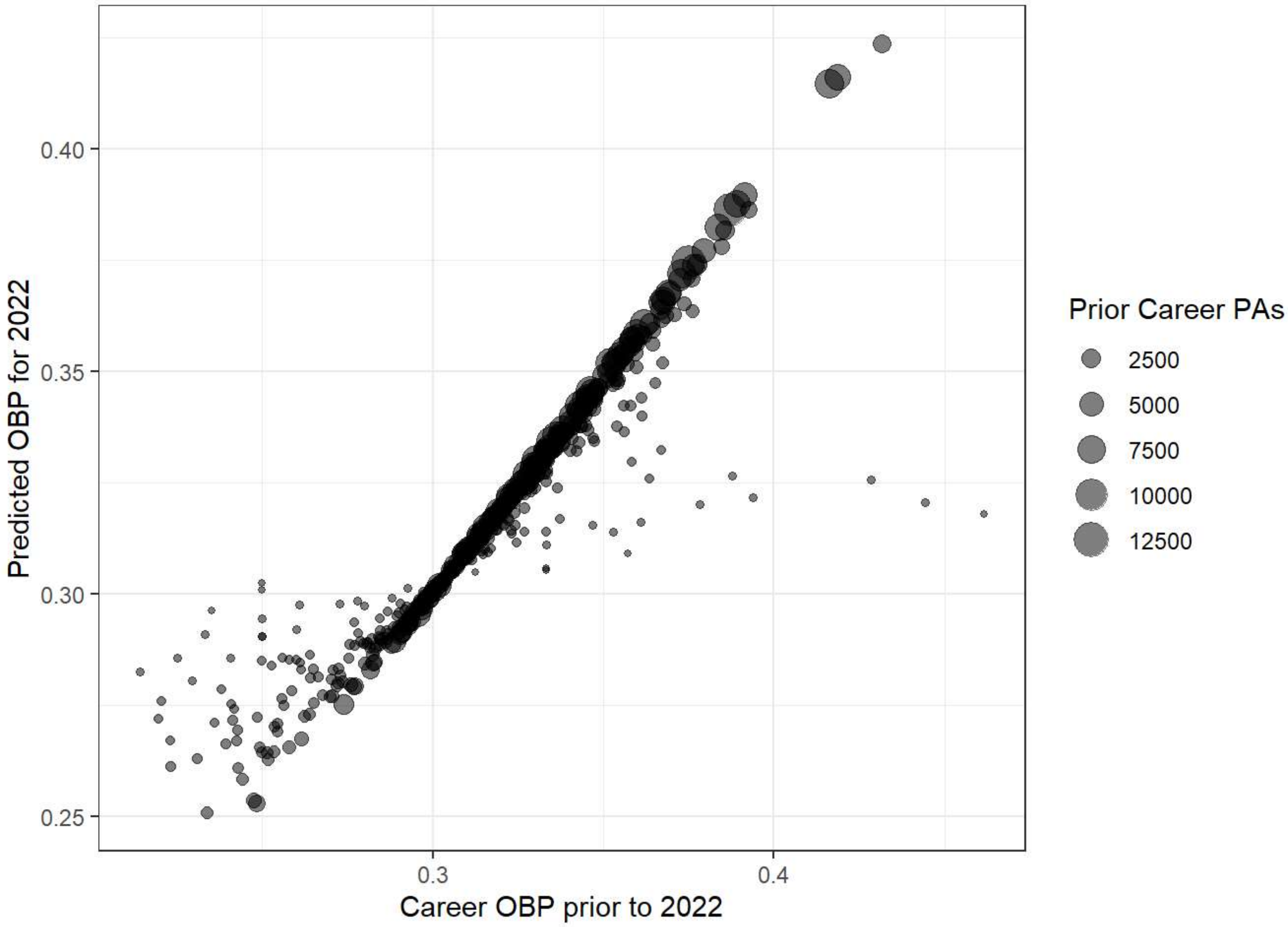
```
test_data %>%
  mutate(pOBP = (prior_gets_on_base + m1@coef[1]) / (prior_PA + m1@coef[1] + m1@coef[2])) %>%
  dplyr::select(playerID, pOBP) %>%
  arrange(desc(pOBP)) %>%
  slice_head(n = 10) %>%
  left_join(People %>% dplyr::select(playerID, nameFirst, nameLast))
```

```
## Joining with `by = join_by(playerID)`
```

```
## # A tibble: 10 × 4
##    playerID   pOBP nameFirst nameLast
##    <chr>     <dbl> <chr>     <chr>
##  1 sotoju01  0.424 Juan      Soto
##  2 troutmi01 0.416 Mike      Trout
##  3 vottojo01 0.415 Joey      Votto
##  4 harpebr03 0.390 Bryce     Harper
##  5 goldspa01 0.388 Paul      Goldschmidt
##  6 cabremi01 0.386 Miguel    Cabrera
##  7 nimmobr01 0.386 Brandon   Nimmo
##  8 freemfr01 0.382 Freddie   Freeman
##  9 judgeaa01 0.382 Aaron     Judge
## 10 winkeje01 0.378 Jesse     Winker
```

It's clearly visible in the following plot that regression to the mean impacts players with both high and low OBPs, but those with more career plate appearances have less expected regression because their performance has more weight than the effective sample size of the prior. For hitters who are early in their careers, on the other hand, the prior is more influential in predicting future performance.

```
test_data %>%
  mutate(pOBP = (prior_gets_on_base + m1@coef[1]) / (prior_PA + m1@coef[1] + m1@coef[2])) %>%
  dplyr::filter((prior_gets_on_base/prior_PA) < .5 &
                (prior_gets_on_base/prior_PA) > .2 &
                !is.na(prior_gets_on_base) &
                !is.na(prior_PA) &
                !is.na(pOBP)) %>%
  ggplot(aes((prior_gets_on_base/prior_PA), pOBP, size = prior_PA)) +
  geom_point(alpha = 0.5) +
  labs(x = "Career OBP prior to 2022",
       y = "Predicted OBP for 2022",
       size = "Prior Career PAs")
```



Predicted OBP for 2022
0.40
0.35
0.30
0.25
0.3
0.4
Career OBP prior to 2022
Prior Career PAs
2500
5000
7500
10000
12500