## Introduction

This project was motivated by a desire for a better public expected batted ball metric. The existing public xwOBA metrics do not consider spray angle. This is problematic because power to the pull side (or to the opposite field) is more valuable than power to straight away center. Additionally, if there is a skill to hitting balls into the gaps, or down the line, or anywhere fielders typically aren’t, then public xwOBA is missing on that too. However, considering spray angle does not solve all our problems. It is conceivable that a player might randomly hit more than their fair share of balls right where fielders typically stand (note that it doesn’t matter where they were actually standing). This would seem to unfairly penalize that player for his bad luck. The current xwOBA metrics regress this problem away by not considering spray angle at all, implicitly assuming the league average spray angle distribution for a given exit velocity/launch angle combination. However, this has its own problems as discussed above.

## Proposal

Instead, I propose using Kernel Density Estimation to create a probability distribution function in {exit velocity, launch angle, spray angle} space based on a player’s actual batted balls. Then draw a large sample of batted balls (e.g. 5000) from that distribution and calculate their average xwOBA. This allows players to get proper credit for having pull-side power while the large sample size eliminates the problem of too many (or too few) line drives going close to where fielders typically stand. This should get closer to measuring a player’s true talent level.

## xwOBA Model

To complete this, first I had to build my own xwOBA model. I downloaded all the batted balls in 2018 from the Statcast Search tool on BaseballSavant.mlb.com. Spray angle is not explicitly listed as a feature, however the coordinates of where the ball is first fielded is included. This allows spray angle to be implicitly calculated. Then, I trained a Support Vector Machine to predict wOBA using the batted ball characteristics. I used an RBF kernel and 5-fold cross-validation to select the RBF exponent and the regularization penalty. Due to the large number of batted balls (over 125,000), the O(n2) complexity of training SVMs, and the limited computing power available to me, I could not train the SVM on any more than 5% of the data when performing cross-validation. Once parameters were selected, I was able to do one training run on 50% of the data.

## Kernel Density Estimation

After the xwOBA model was built, I turned my focus to the kernel density estimation (KDE) of batted ball profiles. I selected a gaussian kernel and used 5-fold cross-validation to choose the bandwith of the KDE. The data for each batter was split into 5 groups, and then the estimator was trained on 4 of the folds and used to calculate the log-likelihood of the held-out fold. This is done 5 times for each potential bandwidth value, with each fold serving as a held-out test set, and the bandwidth that produces the highest log-likelihood across all 5 folds is selected. This CV is done separately for each batter, since the ideal bandwidth varies by sample size and distribution.

## Results

Once the KDE is done, I calculated the xwOBA of a large sample (5000) of batted balls from the resulting pdf for each player. I did this for all batters with at least 100 batted balls in 2018, and the results are attached in a spreadsheet. The top 10 players are below. **Note that this is only the xwOBA of batted balls and does not include strikeouts or walks. It is analogous to xwOBAcon on Baseball Savant.**

|  |  |  |
| --- | --- | --- |
| Rank | Player | KDE xwOBA |
| 1 | Joey Gallo | 0.507 |
| 2 | J.D. Martinez | 0.505 |
| 3 | Max Muncy | 0.491 |
| 4 | Aaron Judge | 0.483 |
| 5 | Shohei Ohtani | 0.480 |
| 6 | Mookie Betts | 0.479 |
| 7 | Nicholas Castellanos | 0.472 |
| 8 | Matt Carpenter | 0.472 |
| 9 | Khris Davis | 0.468 |
| 10 | Mike Trout | 0.466 |

## Future Research

* This metric can be combined with walk rates, strikeout rates, and HBP rates, as well as the batter’s speed to create an overall xwOBA.
* It could also be redone with various splits, such as by pitcher handedness.
* Using some additional preprocessing to turn launch angle, spray angle, and exit velocity into distributions with a mean of 0 and a standard deviation of 1 may improve the performance.
* Kernels other than gaussian can be explored for use in the kernel density estimation.
* Players like Joey Gallo who often face extreme shifts are likely to underperform their   
  KDE xwOBA because fielders’ actual locations against a specific hitter are not considered. Trying to account for this in some way would be an interesting project.