Using Statcast Data to Build a Bayesian Prior for MLB Pitcher Performance

Andy Price

Harvard Gov 1005 — Fall 2019

Research Goal

The goal of this project was to develop a Bayesian prior for predictions of Major League Baseball pitcher performance that rely entirely on pitch velocity and movement. Current attempts to forecast pitcher performance primarily rely on outcomes-based data — how many batters does a pitcher strike out? How many home runs does he give up? Major League Baseball now has Trackman technology installed in all 30 ballparks that allows it to catalog the motion of every pitch thrown, giving us another method for evaluating pitcher performance.

For established pitchers, this type of analysis will be less useful; their prior projections will be updated and outweighed by years of data about the effectiveness of their pitches as measured by the results recorded with those pitches. If we want to know the effectiveness of Clayton Kershaw's slider, we would look at how much success he's had with it over the past decade. This would implicitly include all sorts of factors that a pitch velocity/motion analysis cannot: how well does he command his slider, meaning can he locate it in the part of the strike zone where the batter he's facing is worst at hitting? How well does he mix his pitches, since effecting sequencing (say, following a fastball with an off-speed pitch or a pitch in one area of the zone with one in another) can be as important as the raw movement and velocity of a pitch. But for newer pitchers who are called up from the minor leagues and may not have a very long track record of facing quality hitters, it's useful to compare the shape and speed of their pitches to the hundreds of thousands of pitches thrown each year in order to find a baseline projection that can then be updated with new information as it arrives.

Methods

I start by scraping individual pitch-level data directly from baseball.savant.com. To do this I use Bill Petti's BaseballR package. Baseball Savant is the official archive of this Trackman data (known more commonly as "Statcast" data); Statcast records the velocity and movement in both the x and y axes of every pitch, among many other measurements. I built a Statcast database for 2018 and 2019, starting here because Statcast has been most accurate the past two years. Before 2018, the data has a large "park factor", meaning that because of improper calibration or something related, it was hard to make comparisons between pitches recorded by Statcast at two different stadiums. This phenomenon was discovered by Jared Cross, founder of Steamer Projections, who helped provide background information for this project. Many thanks to Cross and Petti.

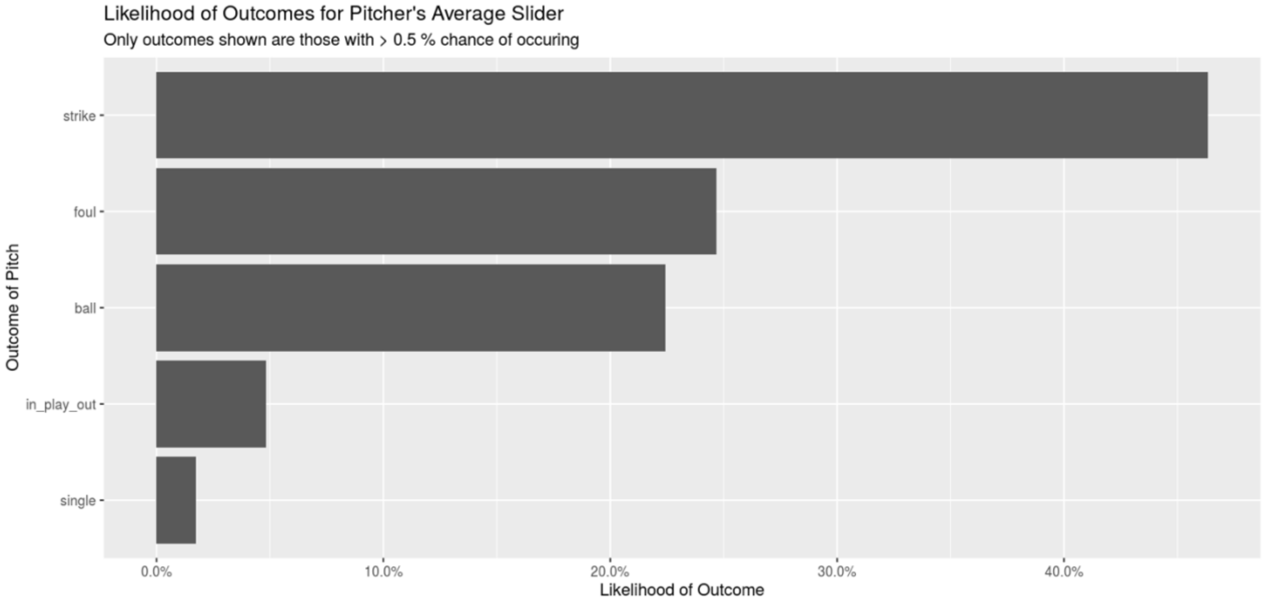
Then I convert the events in the pitch data (single, sacrifice fly, stolen base, etc.) into a series of less granular categories, since there's hardly any measurable difference in pitcher skill between a sacrifice fly and a fly ball or between a foul and a foul tip. Additionally, I change walks and strikeouts into balls and strikes, since there's no meaningful difference between them, other than game situation. After eliminating the noisy categories, I'm left with:

Ball, Strike, Foul, In-play out, Single, Double, Triple, Home Run, and Hit By Pitch.

Then I group the cleaned data by pitch type so I can build a model for each type of pitch. (I only examined pitch types that are thrown more than 10,000 times per year: Four-Seam Fastball, Two-Seam Fastball, Changeup, Curveball, Cutter, Knuckle Curve, Slider, Split-Finger, and Sinker.) I run a random forest regression on each pitch type, creating probability trees (with 100 trees per model). Unlike the typical tree often used in a random forest analysis that either predicts or classifies, these models, pioneered by Malley et al. (2012), produce a probability distribution that takes a pitch's velocity, vertical movement, and horizontal movement and compares it to similar pitches, showing the likelihood that a given pitch will result in any given outcome (say, a called strike or a double). There's no need for cross validation when using random forest regression, since the ranger package automatically withholds parts of the dataset in order to calculate OOB (out-of-bag) error, which is measured using the brier score metric since these predictions are probabilistic.

Now that I have a model for each pitch type, I calculate each pitcher's mean velocity and movement and use these to forecast what range of outcomes we'd expect on their average fastball, for example, if all we knew was its speed and motion (and nothing of the pitcher's command or sequencing).

Here is a sample output, which shows Jacob DeGrom's average slider.



The most likely outcome is that his average slider will be a strike, with the next likeliest outcomes being a foul or a ball.

My most generalizable finding (the true purpose of the project was not to make broad discoveries but rather to create a model that can be used in the future as a prior for Bayesian pitcher predictions) is that off-speed pitches with more movement are far less likely to result in balls in play than fastballs and pitches with less movement.