Baseball Pitch Detection

Zach Potthoff and Alex Rimerman

November 12, 2023

When you attend Major League Baseball and even some Minor League and college games, it is common to see a pitch tracker. Whenever a pitch is thrown, you can instantly turn to the scoreboard and see information about what pitch was thrown. Features such as velocity, movement, and spin rate are all captured with radar, high-speed imaging, and other object tracking technology. However, the pitch pitch type is not as easy to measure, as not every slider is the same from pitcher to pitcher just as the idea that no two snowflakes are the same. The idea is that they have similar properties and therefore can be referred to as the same thing, not that they are exactly the same. Additionally, the pitch detection at games happens seemingly instantaneously, clearly too fast for an intern to be clicking a button. So, it must be that given the data picked up from the stadium, an algorithm can quickly determine what pitch was thrown with close to perfect accuracy. For our project, we want to build a machine learning algorithm that, given the data similar to what is picked up at the stadium, can determine with high accuracy what pitch was likely thrown when given an arbitrary pitch from an MLB game.

The problem that we are looking to solve in this project is if we can make a similar type of algorithm to that of the MLB, in order to determine pitch types in an efficient and accurate manner. A solution to this problem would be one that when given a selected pitch, our algorithm can properly output what type of pitch was thrown in conjunction with the handedness of the pitcher. This needs to happen together because it is often the case that, for example, a left handed pitcher’s sinker might have the very similar movement patterns as a right handed pitcher’s slider or cutter. However, we hope to negate this issue by using other statistics like spin rate and true

spin. We could seek to benefit from a good solution to this problem in order for us to accurately describe what type of pitch we are throwing. In some cases, it can be very easy for pitchers to not properly identify their own pitch types. For example, it is easy to call your curveball a slider and vice versa. In terms of a developmental perspective, having proper knowledge of our own pitch arsenal will allow for us to more accurately develop the pitches into their best possible shape. We would deem our project successful if we can accurately classify 90% of pitches in the test set in under 1 second per pitch. This would be comparable to the MLB pitch tracker, which classifies pitches quickly and accurately, but not perfectly as that is unrealistic.

For our implementation, our initial thought is to look into using a random forest to train and classify pitches. We are considering using velocity, horizontal movement, vertical movement, effective speed, and spin rate as our factors contributing towards pitch type. This is based on our intuition as baseball players as the main factors affecting what a pitch is considered. A random forest seems like a good starting point for an algorithm because we know these features are measured on different scales and may have different weights on the output. For example, it's reasonable to assume that velocity has a bigger impact on whether a pitch is a fastball or not rather than its spin rate, but the opposite can be said for some breaking pitches. Since random forest can account for this, it seems like a good choice to start with as an implementation algorithm. Some other things we may want to consider would be using clustering to see what pitches get clustered together as an analysis aspect of the project. Our intuition says that this part of the project will really struggle as there are huge differences in the same pitch from pitcher to pitcher. However, we think that we could get some interesting results to look at as we see the similarity in certain pitches from one pitcher as compared to a different pitch from another.

The data we are using will be a collection of all pitches thrown throughout the 2023 MLB season. We were able to extract this data from https://baseballsavant.mlb.com through a web scraping tool known as pybaseball. This can be found on github as an open source tool for the baseball community: https://github.com/jldbc/pybaseball. We are using two data sets; one is all pitches thrown in the 2023 World Series, so it is a smaller set of around 1400 pitches. This is what we will use initially to make training and testing as well as error identification easier. The other set that we hope to use as we build the algorithm consists of all pitches thrown in the 2023 season, containing around 740,000 pitches. As previously mentioned, we are using velocity, horizontal movement, vertical movement, effective speed, and spin rate as our factors contributing towards pitch type.

For our experimentation, we will train our algorithm and then test it on arbitrary MLB pitches. At a high-level, we are looking to classify pitches with an accuracy of 90% or greater in an efficient manner. To validate our results, one method we could use would be to run hypothesis tests on accuracy vs. if our algorithm was just flipping a coin with probability 1/(number of unique pitches). We recognize that it is unrealistic for our results to compare in accuracy to the MLB's, as they use deep neural networks to identify pitches. However, we still think it is interesting to see how close we can get with a much simpler model in a 4-week project timeframe. There are minimal impacts with this project, as there are similar technologies as this used at a professional level. Additionally, there are no true negative impacts of this algorithm as it cannot be applied beyond baseball. And, the worst that can come out of baseball applications is a mis-labeled pitch.

References: https://github.com/jldbc/pybaseball, https://baseballsavant.mlb.com/