# Using Machine Learning to Forecast Outcomes of Baseball At-Bats

Lukasz Zbroszczyk and Peter Bigica

Western Connecticut State University

## Abstract

Over the past few decades, there has been a dramatic rise in the use of statistics and data science to inform tactical and strategic decisions in the sport of baseball. More recently, advanced systems such as MLB Statcast have allowed the tracking of physical measures of player performance. We develop an algorithm that employs Markov chains and logistic regression to project the outcomes of at-bats between arbitrary pitchers and hitters, even ones who have never faced one another.

## Modeling At-Bats

An at-bat is an encounter between a pitcher and a batter, consisting of individual **pitches**, which we may model using a Markov process.

During an at-bat, the pitcher throws pitches into the strike zone. A batter is charged with a strike ($K$) when he fails to swing at a pitch in the strike zone, swings and misses at any pitch, or hits the pitch out of play (a foul ball). On the other hand, if the pitcher fails to deliver the ball into the strike zone, he is charged with a ball ($B$). If a batter accrues three strikes, he is said to have struck out. If he manages to accrue four balls, however, the at-bat ends and he is allowed to take a base for free.

This results in twelve **non-terminating states** corresponding to the twelve possible combinations of balls and strikes, as well as six **terminating states** corresponding to the events where an at-bat ends:

- Out (*O*) (field out or strikeout)
- Walk (*BB*)
- Hit-by-pitch (*HBP*)
- Single (*1B*)
- Double (*2B*)
- Home Run (*HR*)

A walk may only occur in a three-ball count, while the other outcomes may occur in any count. In a two-strike count, an out occurs when the batter swings and misses at a pitch (strikeout) or hits the ball into play and makes a field out.

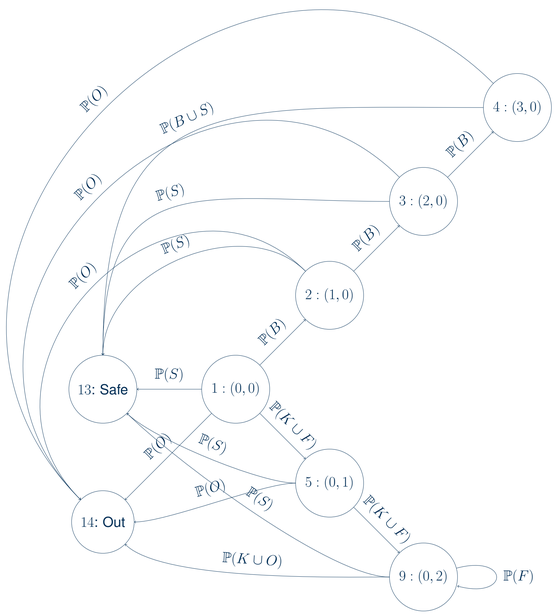



Fig. 1: A simplified graph showing some states of an At-Bat

## The Stochastic Matrix

In general, we may use conditional probability to construct a stochastic matrix **A** representing a Markov process. Such a matrix will be $n \times n$ dimensional, where $n$ is the total number of states the system may find itself in. The $i$th column vector of **A**, $\mathbf{a}_i$, is a stochastic vector where

$$\mathbf{a}_{i_j} = \mathbb{P}\left(E_j | E_i\right)$$

and $E_j$ is the event that the system moves to state $j$ while $E_j$ is the event that the system is in state $i$. In a terminating state $k$, i.e. one from which it is impossible to move to any other state, all entries of the corresponding vector are zero except the $k$th entry, which is equal to one.

In the baseball at-bat model with 18 states, the first twelve states correspond to the twelve possible non-terminating counts, with six additional terminating states corresponding to the final outcomes of the at-bat. We construct a new Markov matrix for every pitcher and batter combination we wish to simulate. For example, the column corresponding to state 1 (no balls and no strikes) will be

$$\mathbf{a}_1 = \begin{bmatrix} 0 & \mathbb{P}_1(B) & 0 & 0 & \mathbb{P}_1(K \cup F) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbb{P}_1(HBP) & \mathbb{P}_1(1B) & \mathbb{P}_1(2B) & \mathbb{P}_1(HR) & \mathbb{P}_1(O) \end{bmatrix}^T$$

while one corresponding to state 12 (three balls and two strikes) will be

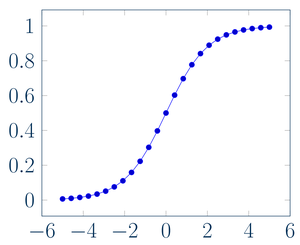
$$\mathbf{a}_{12} = \begin{bmatrix} 0 & \dots & 0 & \mathbb{P}_{12}(F) & \mathbb{P}_{12}(B) & \mathbb{P}_{12}(HBP) & \mathbb{P}_{12}(1B) & \mathbb{P}_{12}(2B) & \mathbb{P}_{12}(HR) & \mathbb{P}_{12}(O \cup K) \end{bmatrix}^T$$

## Calculating Probabilities

We use a logistic regression model to construct the probability of each event at each stage of the Markov chain. For any event $E$, we compute the conditional probability

$$\mathbb{P}\left(E | \text{count}\right)$$

And use it to construct the stochastic Markov matrix. The method used to calculate the individual properties here is multinomial logistic regression, which uses a linear function to classify observations in a sample space as belonging to one of several classes.




## Model Training

We first select one pitcher and one batter with data in the MLB Statcast system, and collect a sample of pitches. The sample is from every at bat by the given pitcher and hitter, against any other batter or pitcher. On the pitcher side, the pitches are grouped by type (fastball, changeup, slider, etc) and average features such as velocity, movement, and location are recorded for each type. On the hitter side, a logistic model is trained using every pitch the batter has seen during the sample period, with the same feature set as the pitcher data, and then classified depending on outcome:

- Ball
- Strike
- Foul Ball
- Hit By Pitch
- Single
- Double
- Home Run
- Field Out

For every state in the at-bat, we then compute the probability that any pitch type is thrown, and use the batter's logistic model to calculate expected performance. The resulting outcomes form the entries of the stochastic matrix for this particular pitcher-batter matchup.

## Simulation

If **A** is the stochastic matrix for a given pitcher-batter matchup and

$$\mathbf{x}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

is a seed vector representing the first pitch of the at-bat, then every iteration of the matrix equation

$$\mathbf{x}_{j+1} = \mathbf{A}\mathbf{x}_j$$

is equivalent to simulating one pitch of the at-bat. Since it is highly unlikely that an at-bat will last indefinitely, in the limit, $\mathbf{x}_j$ is an outcome vector with the first twelve entries all zero, and the last six corresponding to the probability of reaching any terminating state of the at-bat. In practice, we may take **A** and raise it to sufficiently high power, since at-bats rarely last more than ten pitches.

$$\mathbf{x} = \mathbf{A}^{20}\mathbf{x}_0$$

A sample output vector is

$$\begin{bmatrix} 0 \dots 0 & 0.191 & 0 & 0.103 & 0.039 & 0.066 & 0.599 \end{bmatrix}^T$$

## Acknowledgments

We extend our special thanks to:

- Dr. Xiaodi Wang for guiding us along on this research project and encouraging us to present at a conference.
- The Mathematics Department for making this research possible.
- The developers of the `numpy`, `pandas`, `pybaseball`, and `scikit-learn` Python packages for making it easy to access, transform and analyze the MLB Stacast Data.
- Major League Baseball for providing free and open access to their Statcast database and API.

## References

[1] S. Christian Albright. "A Statistical Analysis of Hitting Streaks in Baseball". In: *Journal of the American Statistical Association* 88.424 (1993), pp. 1175–1183. ISSN: 0162-1459.

[2] Bruce Bukiet, Elliotte Rusty Harold, and José Luis Palacios. "A Markov Chain Approach to Baseball". In: *Operations Research* 45.1 (1997), pp. 14–23. ISSN: 0030-364X.

[3] Stephen Marsland. *Machine Learning: An Algorithmic Perspective, Second Edition*. 2nd edition. Boca Raton: Chapman and Hall/CRC, Oct. 8, 2014. 457 pp. ISBN: 978-1-4665-8328-3.

[4] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. ISSN: 1533-7928.

## GitHub Repository

`https://github.com/lwzbr/mlb_at_bats`