IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Using Markov Chains to Predict Outcomes in Baseball

JESSE GAILBREATH[1], JANSEN LONG[2], RYAN MORSE[3], DILAFRUZ SHAMSIEVA[4], ELIJAH WHITE[5]

Corresponding author: First A. Author (e-mail: author@ boulder.nist.gov).

ABSTRACT

MARKOV Chains have been shown to be an effective way in which baseball teams can statistically select batter/pitcher match-ups that are favorable to them. These calculations could take some time using normal computing methods due to the amount of data that may be useful. Therefore, we will implement an algorithm that solves Markov chains with Major League Baseball statistics and will use parallel processing concepts and technologies in order to create an efficient system for predicting what matchups will be favorable for a given player

I. INTRODUCTION

BASEBALL has a well-defined and clean structure, which allows us to predict the probability of certain outcomes. Markov chain models have been used in advanced baseball analysis since the 1960s. Research has shown that Markov chain models can be used to evaluate runs created for the evaluation of individual and team performances.Using the Markov chain model in our study, we will get a reasonable approximation of the probability of events occurring, from which we will develop our analysis. [1]

In order to get as accurate a prediction as possible, there will need to be a large amount of data taken into account. Many calculations can go into just the prediction of one plate appearance. This is where using parallel processing methods will help in this model. Using parallel processing, we will be able to speed up the prediction of events for use in real time scenarios. This allows us to predict a plate appearance as it occurs.

The data that will be used in this project is from "baseball-savant.com" [3]. This is a website that tracks and stores data from Major League games for public use.

II. LITERATURE REVIEW

FOR real-time baseball prediction, many methods have been proposed and implemented. In baseball, consistently winning the matchup between pitcher and batter is integral to winning games. Markov chains models have been used in sports analysis for years, with applications being implemented for predicting the NCAA Men's Basketball Tournament (Paul and Sokol, 2006), making the decision of when to pull a goalie in a hockey game (Zaman, 2001), and ranking college football teams (Kolbush and Sokol, 2017), among many other examples [3]. The predictive power of Markov Chains can give us a ton of insight into pitcher batter matchup and can even help to predict the outcomes in those matchups. There are numerous scientific papers that impressively demonstrate the effectiveness of Markov Chain.

One of the papers explains a statistical analysis of the stochastic event-transition matrices. The central consideration is a balance between accuracy and (possible) changes in baserunner advancement from season to season. This analysis provides a lower bound to the probabilities that transitions from any baseball state to all others are simultaneously within specified distances. It also highlights additional considerations that must be made for event-based matrices, compared to the (total) transition matrix of the (standard) Markov model [5].

An article done by Beaudoin, David from the journal "Journal of Quantitative Analysis in Sports" develops a simulator for matches in Major League Baseball. Aspects of the approach that are studied include the introduction of baserunning probabilities which were obtained through a large data set, and the simulation of nine possible outcomes for each at-bat [6].

III. SYSTEM DESIGN

THE Markov Chain approach to predicting baseball outcomes has several variations. Our team believes that we have found an approach to the problem that would greatly benefit from the use of parallel processing. We will start with

a 19x19 transition matrix representing probabilities of a given player to cause certain outcomes.

Once the data is loaded into the matrix, we then multiply that matrix by itself until the steady state matrix is reached. This is where we see the first possible implementation of parallel processing in this approach. By splitting the 2 matrices into chunks we can parallelize their multiplication. We predict that the computation time will benefit from this process. The steady state will be reached when all the elements to the left of the 1B (Single) column are zero. You are then left with the individual players general probabilities of hitting a single, double, triple, or home run respectively, as well as their batting average, ball, and strikeout percentages. The result is a nicely compressed 12x7 stochastic probabilities matrix which can be calculated for any player with the requisite statistics.

	1B	2B	3B	HR	BIP	BB	K
0-0	**0.110**	**0.047**	**0.003**	**0.079**	**0.379**	**0.171**	**0.210**
0-1	0.094	0.034	0.004	0.063	0.391	0.121	0.293
1-0	0.106	0.053	0.000	0.085	0.344	0.248	0.163
0-2	0.080	0.024	0.000	0.042	0.303	0.096	0.454
1-1	0.103	0.053	0.000	0.067	0.337	0.179	0.262
2-0	0.101	0.053	0.000	0.070	0.254	0.408	0.113
1-2	0.067	0.033	0.000	0.057	0.269	0.130	0.445
2-1	0.084	0.066	0.000	0.086	0.269	0.307	0.188
3-0	0.022	0.037	0.000	0.033	0.106	0.732	0.070
2-2	0.070	0.052	0.000	0.073	0.265	0.207	0.333
3-1	0.037	0.061	0.000	0.054	0.175	0.557	0.115
3-2	0.056	0.056	0.000	0.040	0.177	0.403	0.266

FIGURE 1. Example of Final Steady State Matrix.

From here there are several directions we could go in terms of application for this model. The one that we felt would best benefit from parallel processing is batter-pitcher matchups. To do so is actually quite simple. Once the steady-state matrices are both calculated, to simulate the matchup one only needs to take the average of the 2 matrices. To do so for an entire batter-pitcher lineup outlook would be very computationally taxing, and it will be necessary to apply parallel processing to the algorithm.

IV. DATA COLLECTION AND PREPARATION

THE dataset is collected from the baseball savant website from regular seasons 2017 through September 2022 years. Excel was used to collect, pre-process and store the data. Python is used to retrieve statistics from pybaseball library and leading it into a CSV to do the calculation in C/C++. For transition matrix we are using the data of a baseball pitcher Madison Bumgarner and baseball second baseman Jose Altuve changing it into the matrices format.

V. PROJECT BREAKDOWN

MARKOV chain has advantages in terms of speed and precision.From successional data, a Markov model is relatively simple to construct. The primary parameters of dynamic change (in context of the sport's dynamics as it generates massive amounts of data) are summarized in the transition matrix.It provides a comprehensive view of the sports system's evolution over time. It can easily model additional data and new parameters as data collection is improved.

The Markov-Chain approach is generally applied to baseball by breaking a half-inning of play up into 24 different states and attempts to predict the probability of the corresponding state transitions. [7].

While this method is tried and true, our team happened across another use of this method that we found interesting and decided it could benefit from parallelization. In this method we will use a 19x19 Absorbing Markov-Chain to generate stochastic transition matrices for individual players. These probabilities will be calculated based on the Count (Balls-Strikes), and the outcomes of At Bats [3].

At plate appearance batter finds themselves in 1 of 25 different situations (or states). If relevant stats are available, then a transition matrix can be generated for any such player.

$$P = \begin{bmatrix} A_0 & B_0 & C_0 & D_0 \\ 0 & A_1 & B_1 & E_1 \\ 0 & 0 & A_2 & F_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

25x25

{A, B, C /8x8 Matrices}
{D, E, F/ 8x1 Column Vectors}
{0's in middle 2 rows are 8x8
Last row contains 24 0's and a 1
Subscripts are number of outs of plate appearance

FIGURE 2. Example of Final Steady State Matrix.

The 8x8 blocks represent transitions from the current state to runner on first, runner on second, runner on third, runners on first and second, runners on first and third, runners on second and third, and bases loaded respectively. A blocks represent events that do not increase the number of outs. B blocks represent events that increase the number of outs by one, without reaching three. C blocks events that increase the number of outs from 0 to 2. Column D, E and F represent events that increase the number of outs to 3.

1) Breakdown of block A_0

Each of the rows and columns represent a different configuration of runners on bases. The block A0 specifically represents the probabilities of events that start with no outs and do not incur outs.

The matrix in Figure 3 tracks probabilities of runner movements based on the action of the current hitter. Example: The cell in the 6th row and 8th column is accross from R13 and down from RL. That means it represents the probability of going from a runner on first and third to bases loaded without

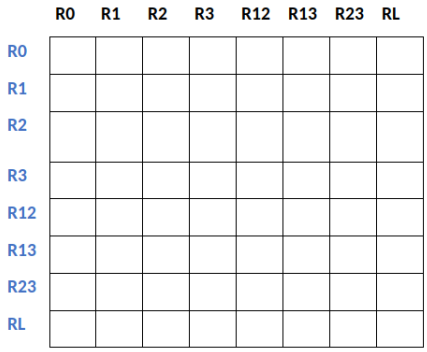

FIGURE 3. .Probability of runner movements

incurring an out. This would most likely be caused by hitting a single.

These stats can be calculated from relevant stats. Our next job is to find the discrete rules governing the matrix formation. For instance using A_0 as another example. A batter stepping to plate with no runners on base will have no way to go from their current state to a runner on first and third. Thus certain cells can be immediately initialized to 0.

To use the matrix a row vector u_0 will contain 25 values and will represent a simulated current state. To place in linear linear algebra terms P is the transition matrix, u_0 is a state matrix, and a simulated base appearance or "At Bat" can be represented by $u_0 x P_{n+1}$ where P_{n+1} is the transition matrix for the next batter.

REFERENCES

[1] S. Deb, "Explore Markov Chains With Examples — Markov Chains With Python," Edureka, May 12, 2020. https://medium.com/edureka/introduction-to-markov-chains-c6cb4bcd5723 (accessed Sep. 02, 2022).

[2] "Baseball Savant: Trending MLB Players, Statcast and Visualizations," baseballsavant.com. https://baseballsavant.mlb.com/ (accessed Sep. 02, 2022).

[3] Turner, C. (2020, July 20). "The pinch-hitter problem" The Diamond. from https://readthediamond.com/research/the-pinch-hitter-problem

[4] Pathak, S. (2021, July 9). Markov chain algorithm in sports. Medium. from https://medium.com/analytics-vidhya/markov-chain-algorithm-in-sports-a54d086c155e

[5] Statshacker. (2018, August 14). Statistical analysis of the stochastic Markov matrices. statshacker. from http://statshacker.com/blog/2018/06/26/statistical-analysis-of-the-stochastic-markov-matrices/

[6] Beaudoin, David. "Various applications to a more realistic baseball simulator" Journal of Quantitative Analysis in Sports, vol. 9, no. 3, 2013, pp. 271-283

[7] Bukiet, Bruce, et al. "A Markov Chain Approach to Baseball." Operations Research, vol. 45, no. 1, 1997, pp. 14–23. JSTOR.