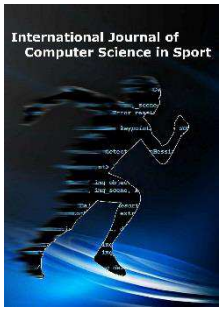
International Journal of Computer Science in Sport

Volume 22, Issue 1, 2023

Journal homepage: http://iacss.org/index.php?id=30

sciendo

DOI: 10.2478/ijcss-2023-0001



Estimating the effect of hitting strategies in baseball using counterfactual virtual simulation with deep learning

Hiroshi Nakahara[a], Kazuya Takeda[a] and Keisuke Fujii[a,b,c]

[a]Graduate School of Informatics, Nagoya University, Nagoya, Aichi, Japan; [b]RIKEN Center for Advanced Intelligence Project, Fukuoka, Fukuoka, Japan; [c]PRESTO, Japan Science and Technology Agency, Kawaguchi, Saitama, Japan

Abstract

In baseball, every play on the field is quantitatively evaluated and the statistics have an effect on individual and team strategies. The weighted on base average (wOBA) is well known as a measure of a batter's hitting contribution. However, this measure ignores the game situation, such as the runners on base, which coaches and batters are known to consider when employing multiple hitting strategies, yet, the effectiveness of these strategies is unknown. This is probably because (1) we cannot obtain the batter's strategy and (2) it is difficult to estimate the effect of the strategies. Here, we propose a new method for estimating the effect using counterfactual batting simulation. The entire framework consists of two phases: (i) generate a counter-factual batter's ability based on their actual performances, and (ii) simulate games with the batting simulator. To realize (i), we propose a deep learning model that transforms batting ability when batting strategy is changed. This method can estimate the effects of various strategies, which has been traditionally difficult with actual game data. We found that, when the switching cost of batting strategies can be ignored, the use of different strategies increased runs. When the switching cost is considered, the conditions for increasing runs were limited. Our results suggest that players and coaches should be careful when employing multiple batting strategies given the trade-offs thereof. We also discuss practical baseball use-cases to use this simulation.

KEYWORDS: BASEBALL, SIMULATION, MACHINE LEARNING, COUNTERFACTUAL

Introduction

Advancements in measurement technologies now allow a much higher level of sports analyses. Especially in baseball, each scene is discrete, making it easy to allocate responsibilities of each play to players. Many formulas and statistics were developed and employed to evaluate player's performance (e.g., abilities of batting, pitching, and fielding) from a long time ago (James, 2010; Lewis, 2004; Click and Keri, 2006; Tango et al., 2007; Beneventano et al., 2012; Costa et al., 2012; Nakahara et al., 2022a). Conventionally, most statistics are computed using batting, pitching, and fielding results. To continually improve measurement, new technologies are being developed that measure sensor data of all Major League Baseball games and make them accessible on data platforms. These data, such as pitched and batted ball location, spin rate, spin axis, and players' location, allow us to develop statistics to estimate players' performance more accurately. It enables us to develop statistics to estimate player's performance more accurately computed by speed and angle of batted balls, such as expected weighted on-Base Average (xwOBA) and expected Earned Run Average (xERA) (MLB, 2022), which can acquire more important information and findings to win. "Fly ball revolution" clarified the effective speed and angle of batted balls for a long hit.

In *Sabermetrics*, which analyzes baseball based on statistics, the importance of each play is defined by the amounts of contributions with a win. WAR (Wins Above Replacement), which is widely used as overall player evaluation statistics, is computed using the number of the contribution of plays to a win such as pitching, batting, fielding, and base running. For batting, it is computed using the weighted on base average (wOBA). However, wOBA does not consider the game state (e.g., score, out count, runner occupancy, etc.) because each player cannot choose when to have a turn at bat or who is on the pitcher's mound.

In contrast, a player is trained to make his or her play by considering the game state to win the game. However, estimating the effect of a strategy with real game data is challenging for two reasons: First, it is difficult to observe or record a team or player's strategy. Here we can consider a batter's intent and there are a lot of batting strategies. In this study we focus on popular contact (on-base) prioritized and power (long hit) prioritized strategies. Second, various factors affect baseball outcomes (e.g., batting results). We thus employ computer simulation to simulate a baseball game and estimate the effect.

Baseball simulations are used for quantifying the effect of strategies. In this regard, the literature has clarified the optimal batting order for various settings (Freeze, 1974; Bukiet et al., 1997; Soko, 2003; Hirotsu et al., 2016). Norman and Clarke conducted a similar study on cricket (Norman and Clarke, 2010). In the analysis of substitution strategies, researchers have employed Markov chain model to simulate games (Hirotsu and Wright, 2003; Hirotsu and Wright, 2005); this approach helps optimize pinch hitting strategies and pitcher substitution strategies. Researchers have also focused on evaluating players or plays. Some researches focus on evaluating players or plays. Player's batting ability was quantified by simulating games if they batted in all nine positions in the line-up (Cover and Keilers, 1977), and the impact of base-running ability was clarified (Baumer, 2009). Other research (Hirotsu and Bickel, 2016; Hirotsu and Bickel, 2019) estimated the value of the sacrifice bunt. Similarly, scholars have quantified the effect of rule changes through simulation in various sports (Forrest et al., 2005; Sonne and Keir, 2016).

Following the literature (Bukiet et al., 1997; Hirotsu and Wright, 2003), we use a simulator to estimate the effect of strategies using Markov chain modeling. A Markov chain is a probabilistic model that describes a series of possible events, where the probability of each event depends only on the state achieved in the previous event. This approach is especially suited for baseball. Since the transition model for runner advancement used in previous works (Bukiet et al., 1997;

Hirotsu and Wright, 2003) is too simple, we construct a model based on transitions occurring in real games. In general, adopting a different strategy requires a trade-off among statistics. For example, the probability of long hits decreases when a player prioritizes reaching the base. We thus need to estimate a player's ability counter-factually for each strategy. We call this framework *counter-factual simulation*.

In this study, we consider the effect of batting strategies as one example of using multiple strategies. Based on the player's attitude toward the game's context and the player's wOBA, we categorize batting strategies into three types: (1) simply maximizing wOBA without considering other elements, (2) considering the game's context and the player's wOBA (e.g., light hitting at scoring opportunity), and (3) only considering the game context and trying to move a runner to the advanced base (e.g., sacrifice bunt). Strategy (1) is effective for getting runs as wOBA explains runs well, while some studies show that strategy (3) cannot effectively maximize runs (Hirukawa, 2019; Tango et al., 2007). Finally, the effect of strategy (2) has not been clarified because of difficulties in estimating.

In our simulation, we assume that a batter adopts three strategies to maximize runs: (1) normal, (2) prioritizing reaching a base, and (3) prioritizing making a long hit. Which strategy a player should adopt depends on game contexts. For estimating the counter-factual ability (prioritizing reaching a base or long hit), we build a *batting strategy converting model* that learns the relationships among real players' batting ability using machine learning. This enables us to simulate different batting strategies and, thus, estimate the effect.

The main contributions of this paper are as follows: (i) estimating the effect of adopting different batting strategies through counter-factual simulations, (ii) proposing a new method for counter-factually converting batting ability using machine learning, and (iii) clarifying the relationship between a batting strategy and runs thereof.

Methods

Dataset

For this study, we used the data of professional Japanese baseball games from 2018 to 2020 provided by the Research Center for Medical and Health Data Science at the Institute of Statistical Mathematics and Data Stadium Inc. Data acquisition was based on the contract between the baseball league and Data Stadium, Inc., but not between the players and us. We obtained the data by participating in a competition hosted by the academic organizations. The central idea of this study was independent of the competition (not restricted by the competition).

The data includes pitch-by-pitch data and individual player statistics per game. Pitch-by-pitch data provide, for example, game context (inning, out count, and runner), pitch results, and batting results. We used pitch-by-pitch data to compute the correspondence between batting result and game context transition. Individual player statistics are indicators of batting performance (plate appearance, hit, homerun, strikeout, and so on) per game, and we use these statistics to compute a player's batting ability vector (i.e., the probability of each batting result).

Batting simulator

We use *batting simulator* to simulate a baseball game virtually. First, we define a batting ability vector P as follows:

$$P = (p_{1b}, p_{2b}, p_{3b}, p_{hr}, p_{bb}, p_k, p_g, p_f), \quad (1)$$

where $p_{1b}, p_{2b}, p_{3b}, p_{hr}, p_{bb}, p_k, p_g$ and p_f represent the probability of a one-base hit, a two-

base hit, a three-base hit, a homerun, a walk and hit by pitch, a strikeout, an easy grounder out, and an easy fly ball out, respectively. These probabilities are regularly computed based on actual batter's statistics (e.g., if a batter makes 10 single hits in 100 at bats, $p_{1b} = 0.1$). P can be also interpreted as a discrete probability distribution, and the sum of each probability is 1. Batting results other than those listed above do not occur. An easy grounder out and easy fly ball out include reaching a base because of the opposing fielder's mistakes; these are thus recorded as a batter's batting result. We generate nine (the number of batters in a game) P, denoted as $\boldsymbol{P}$.

The *Batting simulator* is the function with $\boldsymbol{P}$ as the input and team's runs as the output. Different from real games, it only considers a batting phase (i.e., consists of nine half-innings). The *Batting simulator* consists of the *batting function* and the *transition function*. The former is the function with the game context (out count and runner), with P as the input and batting result as the output. Each batting result is determined by the respective P. A player has three types of P and uses them differently depending on the game context (see Evaluation subsection). Although extant work (Bukiet et al., 1997; Hirotsu and Wright, 2003) has mainly used a simple model (D'ESOPO, 1997), we construct a sophisticated transition model using game context transitions in real games. Then, using pitch-by-pitch data, we compute the distribution of game context transitions. Further, the *transition function* represents the function with the previous game context (out count and runner occupation), with batting result as the input and game context (out count, runner occupation, and run) after batting as the output. For example, when a batter makes a single hit, with 0 outs with 1st-base loaded, various transitions are expected as Figure 1. On the contrary, because the *transition function* takes as input only the game context and the batting result, a transition caused by a factor other than the batting result (e.g., wild pitch and stolen base) does not occur. This also implies that the running ability of a runner and the fielding ability of a fielder are always average. Despite its simplicity, the *batting simulator* outputs the run distribution similar to real games (see Model validation subsection). The *batting simulator* repeats the *batting function* and *transition function* for nine innings, and then computes the runs.

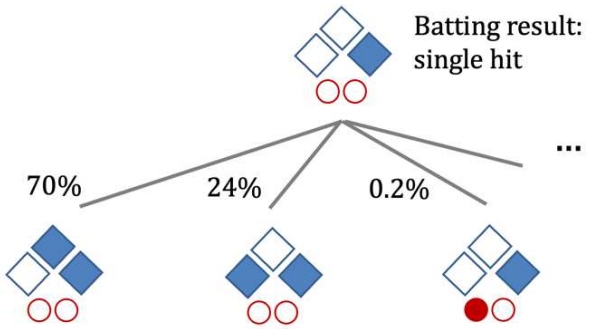


Figure 1. **Transition probability when a batter makes a single hit where 0-out with 1st-base loaded in *transition function.***

Counter-factual batting simulation

Next, we propose the counter-factual batting simulation using the batting simulator, which estimates the effect of the specific strategy with the counter-factual batter's ability conversion (Figure 2). The entire framework of the counter-factual batting simulation consists of two phases: (1) generate a counter-factual batter's ability based on their actual performances, and (2) simulate games with the batting simulator. The architecture of (2) corresponds to the *batting simulation*. Thus, we first describe a general formulation of the problem of converting the batting

ability vector, and then describe a formulation for converting the batting ability vector in the case of changing to the on-base/long hit strategy, which is our focus. Finally, we explain how to convert the batting ability vector using machine learning.

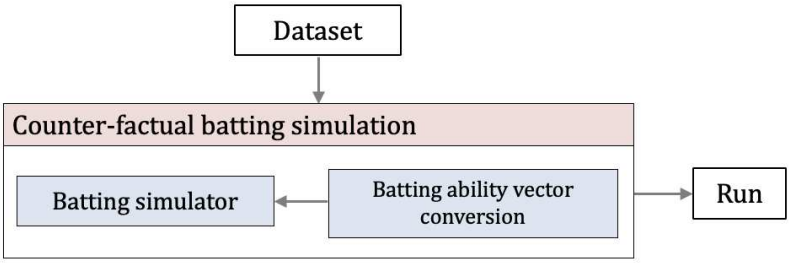


Figure 2. **The architecture of *counter-factual batting simulation*.**

General formulation of the batting ability vector conversion

We consider a change of the individual batting ability vector when converting from the normal strategy P_n to a different strategy (e.g., on-base/long hit prioritized) P_*, denoted as $\Delta P = P_* - P_n$. This P_* estimation problem is divided into two sub-problems: (1) estimating a feasible maximum $|\Delta P|$ (i.e., the scale of conversion) and (2) estimating an appropriate $\Delta P/|\Delta P|$ (i.e., the direction of conversion). We consider it difficult to solve (1), and thus simulate with various $|\Delta P|$. On the other hand, $\Delta P/|\Delta P|$ can be estimated from the batting ability vectors of real players, which help us develop a reasonable batting strategy conversion model.

We regard *batting strategy conversion model* as the function with $P_n, \Delta\alpha$, and $\Delta wOBA$ as the inputs and P_* as the output, where $\Delta\alpha$ and $\Delta wOBA$ are the parameters of the batting strategy conversion model. α is the magnitude of the batting strategy conversion and can be computed with batting ability vector. $\Delta\alpha$ is the difference in α before and after conversion and is computed by $\Delta\alpha = \alpha_* - \alpha_n$. $\Delta wOBA$ is computed by $\Delta wOBA = wOBA_* - wOBA_n (\leq 0)$, and α_* and $wOBA_*$ indicate α and wOBA after conversion, respectively. $\Delta wOBA$ is the cost of the batting strategy conversion because we consider that some amount of wOBA reduces when a batting strategy changes. Because estimating appropriate $\Delta wOBA$ is difficult, we simulate with various settings.

Converting the on-base/long hit prioritized batting ability vector

We assume that a batter chooses three batting strategies: (1) normal, (2) on-base prioritized, and (3) long hit prioritized strategies. In general, α is defined as follows:

$$\alpha = f(P) = \frac{\Delta_{1b} \times p_{1b} + \Delta_{bb} \times p_{bb}}{\sum_{\{1b,bb,2b,3b,hr\} \in r} \Delta_r \times P_r}, \quad (2)$$

where Δ_r is the run value of batting result r, and $\Delta_{1b} = 0.437, \Delta_{bb} = 0.294, \Delta_{2b} = 0.786, \Delta_{3b} = 1.117$, and $\Delta_{hr} = 1.408$ (Okada, 2017). In other words, α is the percentage of singles and walks in the scoring contribution of the batter with P.

The batting ability vectors for the normal, on-base prioritized, and long hit prioritized strategies denote P_n, P_o, and P_l respectively, and the α corresponding to each batting ability

vector are α_n, α_o, and α_l. We set P_n, P_o, and P_l such that $\alpha_l < \alpha_n < \alpha_o$ is satisfied. In other words, we define a strategy with a larger α than the normal strategy as the on-base prioritized strategy and a strategy with a smaller α as the long hit prioritized strategy. For simplicity, all conversions satisfy $\alpha_o - \alpha_n \simeq \alpha_n - \alpha_l$ for a player's α_n, α_o, and α_l. Note that the conversion also considers the constraints on $\Delta wOBA$, as described in "General formulation of the batting ability vector conversion".

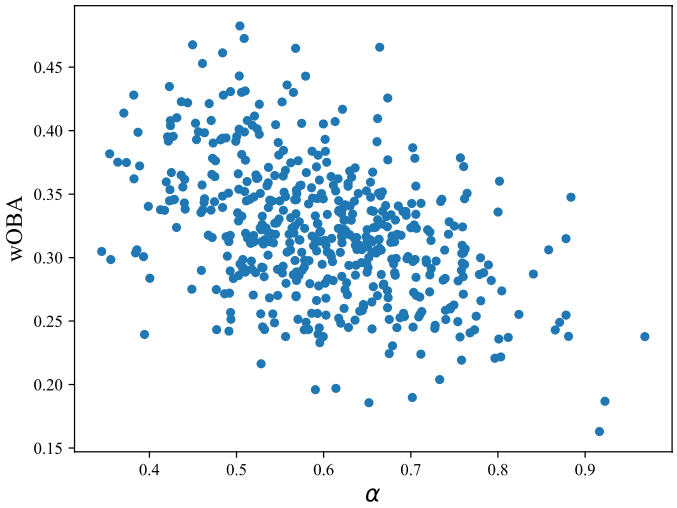


Figure 3. **Each batter's α and wOBA with 100+ at-bat in 2018-2020 (502 batters in total).**

Batting ability vector conversion with machine learning

Next, we consider the batting ability vector conversion when changing $\Delta\alpha$ and $\Delta\, wOBA$ from the baseline (i.e., using actual game data), which does not exist in the actual game data. To compute the nonexistent batting ability vector, we construct the batting strategy conversion model using machine learning, which is trained by the relationships among batters. In other words, our method can estimate the effects of various strategies, which has been traditionally difficult with actual game data.

At first, using each batter's batting statistics over 100 at-bats per season, we computed 502 batting ability vectors (Figure 3: here we count the same batters in different seasons as different batters). Then, P were matched for all combinations, which corresponds to P before/after the batting strategy conversion. Ultimately, $N = 502 \times 501/2 = 125,751$ combinations were generated. For each pair, $\Delta\alpha$ and $\Delta wOBA$ were computed and used as the train data for the batting strategy conversion model.

We employed a multi-layer perceptron to build the batting strategy conversion model because the batting strategy conversion is assumed to be a non-linear conversion. The input was $P', \Delta\alpha$, and $\Delta wOBA$, and the output was $P_*{}' - P'$, where P' denotes that the 7-dimensional vector $P' = (p_{1b}, p_{2b}, p_{3b}, p_{hr}, p_{bb}, p_k, p_g)$. This is because, based on the constraint that the sum of the probabilities is 1, p_f is determined if P' is computed. The middle layer has 2 layers with 100 neurons each and the rectified linear unit (ReLU) was used as the activation function. The loss function was as follows:

$$loss = \frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{\overline{p_i}'\in\overline{P_i}',\widehat{p_i}'\in\widehat{P_i}'}(\overline{p_i}' - \widehat{p_i}')^2 - \gamma\sum_{\widehat{p_i}'\in\widehat{P_i}'}\min(\widehat{p_i}', 0) + \beta\left(wOBA(\overline{P_i}') - wOBA(\widehat{P_i}')\right)^2\right\}, \quad (3)$$

where $\overline{P'}$ and $\widehat{P'}$ are the ground truth and estimated P', respectively. $wOBA(P')$ represents wOBA of a player whose batting ability vector is P'. The first and third terms refer to the mean squared error of the batting ability vector and wOBA, respectively. The second term is the constraint probability is greater than or equal to 0. The soft-max function is not used, because it shows inferior performance to the function employed. γ and β represent each constraint's weight; we set $\gamma = 0.1$ and $\beta = 0.5$ empirically. Eighty percent of data was used for training and twenty percent of data was used for validation.

Table 1. The batting statistics in the simulation using all Pacific league games data from 2018 to 2020. OBP is the on-base average and SLG is the slugging percentage**.**

	OBP	SLG	wOBA	α
1	.337	.377	.320	.64
2	.324	.369	.310	.65
3	.393	.476	.383	.54
4	.360	.464	.363	.50
5	.335	.411	.331	.59
6	.329	.408	.327	.58
7	.316	.369	.307	.58
8	.288	.331	.278	.63
9	.292	.308	.273	.68

Evaluation

P was computed using the average batting ability vector by batting order. We used all Pacific league games data from 2018 to 2020 (Table 1) given that DH system is adopted in all Pacific league games. $\Delta\alpha$ and $\Delta wOBA$ were empirically set as $\Delta\alpha = \{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}, \Delta wOBA = \{0, -0.005, -0.01, -0.015\}$, respectively. Unless otherwise noted, the condition for employing the on-base prioritized strategy is as follows: (1) the out count is zero or (2) the second and/or third base is loaded. Similarly, the conditions for employing the long hit prioritized strategy are as follows: (1) the out count is two and (2) neither the second nor the third base is loaded. We simulated 100,000 games for each combination of parameters and obtained the average runs and its standard error.

Results

Model validation

For the batting simulator, the average runs scored in the simulator was 4.27 per game compared with 4.22 for all Pacific league games held in 2018–2020. Figure 4 shows the run distribution of the simulation and real games. We considered that the average runs and the run distribution were similar appropriately. For the batting strategy conversion model, mean squared error of P' and wOBA as well as the probability constraint were 1.11×10^{-3}, 4.00×10^{-4}, 0, respectively.

The results suggest that the simulator and the hitting strategy conversion model could output realistic values.

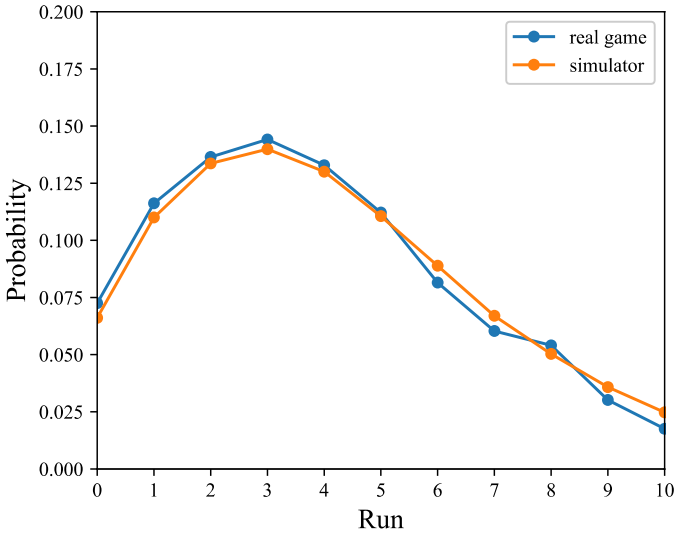


Figure 4. **The run distribution comparison between real games and simulation results.**

Effect of using multiple strategies

Figure 5 shows the result of simulation with parameters $\Delta\alpha$ and $\Delta wOBA$. Note that the average runs are expressed as the difference from the baseline (4.27), which is fixed with the normal strategy. When $\Delta wOBA = 0$, the average runs increase as $\Delta\alpha$ increased, and was greater than the baseline. This shows that using multiple batting strategies increases the average runs when the wOBA loss is ignored. On the other hand, if we consider wOBA loss, the average runs increase, where (1) $\Delta\alpha \geq 0.15$ and $\Delta wOBA \geq -0.005$, or (2) $\Delta\alpha \geq 0.2$ and $\Delta wOBA \geq -0.01$.

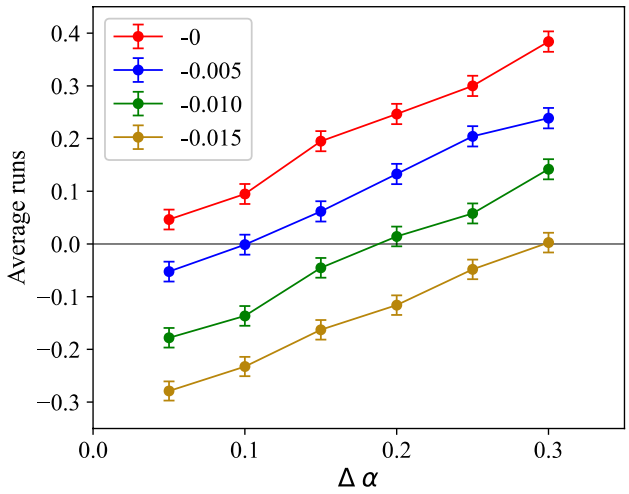


Figure 5. **Runs by the combination of $\Delta\alpha$ (X-axis) and $\Delta wOBA$ (plot color).** The runs with normal strategy are set to the baseline(0.0).

Effect by strategy activation conditions

We also clarified the relationship between strategy activation conditions and average runs. Although we should perform simulations with all combinations of parameters (out count and

runner), it is difficult to simulate all settings ($2^{3\times8}$). Generally, the on-base prioritized strategy is employed when the run expectancy is high, whereas the long hit prioritized strategy is employed when the run expectancy is low. Thus, each strategy activation condition was defined by the run expectancy. More specifically, the on-base prioritized strategy was employed when $RE_c \geq \theta_o$, and long hit prioritized strategy was employed when $RE_c \leq \theta_l$. Note that RE_c represents the run expectancy of game context c has, and θ_o and θ_l denote the thresholds of the on-base/long hit prioritized strategy. A normal strategy was employed if both conditions were not satisfied. The other parameters were fixed as $\Delta\alpha = 0.1$ and $\Delta wOBA = -0.005$.

Figure 6 shows the result. The lower left corner shows looser triggering conditions (employed frequently) and the upper right corner shows stricter triggering conditions (employed less frequently). The average runs increased when the strategy activation conditions were strict (employed less frequently). The combinations of θ_o and θ_l with the highest runs (4.33) were $\theta_o = 1.057$ and $\theta_l = 0.306$. $\theta_o = 1.057$ corresponds to 0 out with second base loaded, and $\theta_l = 0.306$ corresponds to 2 out with first base loaded.

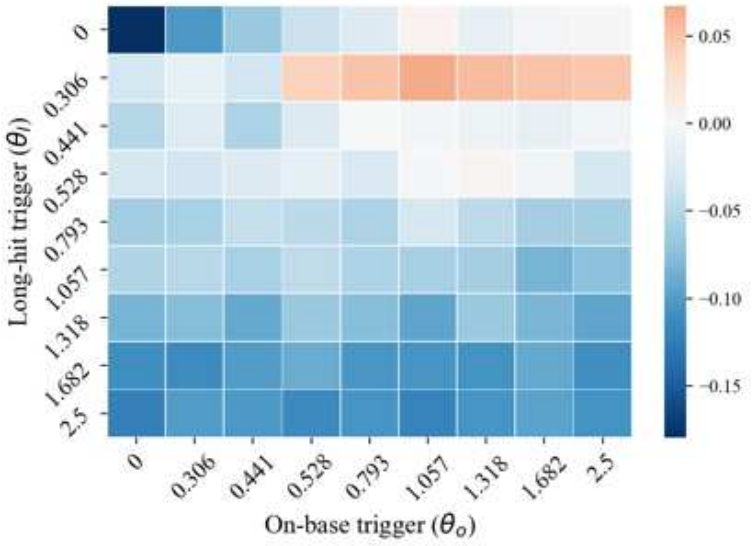


Figure 6. Runs by the combination of θ_o and θ_l**.** The X-axis shows the on-base prioritized strategy's trigger. The higher the θ_o, the less frequently the on-base prioritized strategy is employed. Similarly, the Y-axis shows the long hit prioritized strategy's trigger. The higher the θ_l, the more frequently the long hit prioritized strategy is employed. The normal strategy was employed if both conditions were not satisfied. The other parameters were fixed as $\Delta\alpha = 0.1$ and $\Delta wOBA = -0.005$. The runs with normal strategy were set to the baseline(0.0) as Figure 5**.**

Discussion

In this study, we proposed the counter-factual batting simulation and the batting strategy conversion model to simulate a game in which the players use multiple batting strategies. The validation results suggest that our simulation would be useful to clarify the effect of using multiple batting strategies. Here we discuss our experimental results, the limitation, and future work.

As shown in Effect of using multiple strategies subsection, we clarified that using multiple batting strategies increased the average runs when the wOBA loss was ignored. Thus, the appropriate batting ability vector changes depending on the situation. The strategy of changing batting ability vectors depending on the situation was thus confirmed to be reasonable

as a batter does.

More specifically, multiple batting strategies ($\Delta\alpha = 0.2, \Delta wOBA = -0.005$) increased 0.13 runs per game compared with the normal strategy. Assuming that each team has 143 games per a season (NPB) and 10 runs worth 1 win, using multiple batting strategies increases 1.9 wins in a season. However, WAR does not evaluate team batting, as it does not consider game context in the evaluation. Given that WAR quantifies the contribution to runs (wins), we must evaluate the contributions of team hitting.

On the other hand, when we did consider the wOBA loss, we found that the conditions for increasing runs were limited. Thus, it is better not to use multiple batting strategies if $\Delta wOBA \leq -0.015$. For a P of average ability, the standard deviation of the wOBA distribution per 400 at-bat is 0.026, whereas the value of 0.015 is comparatively lower. Although changing batting strategies are welcomed in Japan, the benefit of doing so for a batter may be too small. Players and coaches should be careful when employing multiple batting strategies given the trade-offs thereof.

As shown in Figure 6, the average runs increased where strategy activation conditions were strict (lower employed). In most situations, the runs were lower than in the normal strategy. Runs were thus highly dependent on $\Delta\alpha$ and $\Delta wOBA$, and ideally, a simulation is done for each setting in practice.

Second, the question about the multiple batting strategies cannot be answered from these results, because the relationship between $\Delta\alpha$ and ΔwOBA remains unclear, and it is difficult to estimate this relationship using real game data. This is because we cannot check if a batter changes his or her batting strategy in a game. We can consider the idea that a batter's batting strategy is completely determined by game situations (e.g., A batter employs the on-base prioritized strategy when a runner is on third base). However, this assumption is too hard to represent actual players' behavior. It depends not only on game situations, but a player's preference and team strategy. For future studies, we suggest experiments to estimate this relationship such that (1) An experimenter instructs a subject (batter) to employ the specific batting strategy (normal/on-base/long hit). (2) A subject hit a ball many times and an experimenter calculates individual or teams' relationship between $\Delta\alpha$ and ΔwOBA. In practical baseball use-cases, we can similarly suggest the following scenarios: (i) interviewing players in a team and obtaining their individual $\Delta\alpha$ and ΔwOBA, (ii) calculating average $\Delta\alpha$ and ΔwOBA in a team, and (iii) checking Figure 5 and if average $\Delta\alpha$ and ΔwOBA satisfy the criteria to employ a team batting strategy.

Third, counter-factual prediction method can be extended. In this study, we used a simple matching method (e.g., see Stuart (2010) for a review) to compute the counter-factual batting ability when changing the strategy. Even if we obtain the batting ability when changing the strategy in the above approach, in real-world baseball data, there should be biases in the data with and without changing the strategy. In sports studies, to investigate the causal effect of some plays or timeouts in many sports, propensity score matching or other approaches were used in the previous work (Fujii et al., 2022; Gibbs, Elmore, & Fosdick, 2020; Nakahara et al., 2022b; Toumi & Lopez, 2019; Vock & Vock, 2018; Yam & Lopez, 2019). Such extension to reduce the biases of data in our setting would also be future work.

Lastly, the batting strategy conversion model still has room for improvement. Note that it does not consider the batter's character except for α, and the conversion might be inappropriate. For example, we can consider the ratio of $BB\%$ to $1B\%$, where $BB\%$ and $1B\%$ are the percentage of walk and first-base hit, respectively. Then, player A's ability with high $BB\%/1B\%$ may be converted into an ability with low $BB\%/1B\%$. In other words, only $\Delta\alpha$ is

considered, and other facts are ignored. Clustering players before building the model may reasonably address this problem.

Acknowledgements

This work was supported by JSPS KAKENHI (Grant Numbers 20H04075 and 21H05300) and JST Presto (Grant Number JPMJPR20CA).

References

Baumer, B. S. (2009). Using simulation to estimate the impact of baserunning ability in baseball. Journal of Quantitative Analysis in Sports, 5(2).

Beneventano, P., Berger, P. D., & Weinberg, B. D. (2012). Predicting run production and run prevention in baseball: the impact of sabermetrics. Int J Bus Humanit Technol, 2(4), 67–75.

Bukiet, B., Harold, E. R., & Palacios, J. L. (1997). A markov chain approach to baseball. Operations Research, 45(1), 14–23.

Click, J., & Keri, J. (2006). Baseball between the numbers: Why everything you know about the game is wrong.

Perseus Books Group. Costa, G. B., Huber, M. R., & Saccoman, J. T. (2012). Reasoning with sabermetrics: Applying statistical science to baseball's tough questions.

McFarland. Cover, T. M., & Keilers, C. W. (1977). An off ensive earned-run average for baseball. Operations Research, 25(5), 729–740.

D'esopo, D. (1977). The distribution of runs in the game of baseball. Optimal Strategies in Sports, Ladany.

Forrest, D., Beaumont, J., Goddard, J., & Simmons, R. (2005). Home advantage and the debate about competitive balance in professional sports leagues. Journal of Sports Sciences, 23(4), 439–445.

Freeze, R. A. (1974). An analysis of baseball batting order by monte carlo simulation. Operations Research, 22(4), 728–735.

Fujii, K., Takeuchi, K., Kuribayashi, A., Takeishi, N., Kawahara, Y., & Takeda, K. (2022). Estimating counterfactual treatment outcomes over time in complex multi-agent scenarios. arXiv preprint arXiv:2206.01900.

Gibbs, C., Elmore, R., & Fosdick, B. (2020). The causal eff ect of a timeout at stopping an opposing run in the nba. arXiv preprint arXiv:2011.11691.

Hirotsu, N., & Bickel, J. E. (2019). Using a markov decision process to model the value of the sacrifi ce bunt. Journal of Quantitative Analysis in Sports, 15(4), 327–344.

Hirotsu, N., & Eric Bickel, J. (2016). Optimal batting orders in run-limit-rule baseball: a markov chain approach. IMA Journal of Management Mathematics, 27(2), 297–313.

Hirotsu, N., & Wright, M. (2003). A markov chain approach to optimal pinch hitting strategies in a designated hitter rule baseball game. Journal of the Operations Research Society of Japan, 46(3), 353–371.

Hirotsu, N., & Wright, M. (2005). Modelling a baseball game to optimise pitcher substitution strategies incorporating handedness of players. IMA Journal of Management Mathematics, 16(2), 179–194.

Hirukawa, K. (2019). Introduction to sabermetrics. Suiyosha.

James, B. (2010). The new bill james historical baseball abstract. Simon and Schuster.

Lewis, M. (2004). Moneyball: The art of winning an unfair game. WW Norton & Company.

MLB.com. (2015a). Expected era (xera). Retrieved from https://www.mlb.com/glossary/statcast/expected-era.

MLB.com. (2015b). Expected weighted on-base average (xwoba). Retrieved from https://www.mlb.com/glossary/statcast/expected-woba.
MLB.com. (2022a). On-base average (obp). Retrieved from https://www.mlb.com/glossary/standard-stats/on-base-percentage.
MLB.com. (2022b). Slugging percentage (slg). Retrieved from https://www.mlb.com/glossary/standard-stats/slugging-percentage.
Nakahara, H., Takeda, K., & Fujii, K. (2022a). Evaluating a third base coach's decision making via game theory and machine learning. MathSport International.
Nakahara, H., Takeda, K., & Fujii, K. (2022b). Pitching strategy evaluation via stratifi ed analysis using propensity score. arXiv preprint arXiv:2208.03492.
Norman, J. M., & Clarke, S. R. (2010). Optimal batting orders in cricket. Journal of the Operational Research Society, 61(6), 980–986.
Okada, Y. (2017). Delta baseball report 1. Suiyosha.
Sokol, J. S. (2003). A robust heuristic for batting order optimization under uncertainty. Journal of Heuristics, 9(4), 353–370.
Sonne, M. W., & Keir, P. J. (2016). Major league baseball pace-of-play rules and their infl uence on predicted muscle fatigue during simulated baseball games. Journal of sports sciences, 34(21), 2054–2062.
Stuart, E. A. (2010). Matching methods for causal inference: A review and a look forward. Statistical science: a review journal of the Institute of Mathematical Statistics, 25(1), 1.
Tango, T. M., Lichtman, M. G., & Dolphin, A. E. (2007). The book: Playing the percentages in baseball. Potomac Books, Inc.
Toumi, A., & Lopez, M. (2019). From grapes and prunes to apples and apples: Using matched methods to estimate optimal zone entry decision-making in the national hockey league. In Carnegie mellon sports analytics conference 2019.
Vock, D. M., & Vock, L. F. B. (2018). Estimating the eff ect of plate discipline usinga causal inference framework: an application of the g-computation algorithm. Journal of Quantitative Analysis in Sports, 14(2), 37–56.
Yam, D. R., & Lopez, M. J. (2019). What was lost? a causal estimate of fourth down behavior in the national football league. Journal of Sports Analytics, 5(3), 153–167.