G-036

マルチエージェント強化学習による野球の戦略推定
Baseball Strategy Analysis via Multi-Agent Reinforcement Learning with Stats Data

島野 雄貴[1)]　田高 礼子[1)]　高橋 正樹[1)]
Yuki Shimano　Reiko Takou　Masaki Takahashi

1　はじめに

本研究では，野球における戦略をマルチエージェント強化学習の枠組みで導出し，その戦略の妥当性について分析した．我々は，視覚障害者を含むあらゆる視聴者が野球中継をより理解できるようにするために，野球中継に対して解説音声を自動で付与する研究開発を進めている．解説すべき項目は種々存在するが，本研究では，野球の先の展開を予測したり現在の状況を戦術の面から解説するための戦略分析に焦点を当て，その分析のための技術について述べる．野球においては長年，セイバーメトリクス [3] による戦略の分析が行われており，近年ではニューラルネットワークを活用した戦略の分析，予測 [2] なども行われつつある．しかし，一般に野球においてはプレーヤー自身の行動選択のみで結果が決まることはなく，相手の行動選択に応じて結果が変わる．すなわち，野球の戦略を分析したり予測する上では，プレーヤーやチーム同士の戦略的相互作用を考慮することが望ましいと考える．そこで我々は，過去に行われたプロ野球のスタッツデータをもとに，野球における状態集合，行動集合，状態遷移関数，および報酬関数を定義し，それらをもとに野球を二人零和マルコフゲームとして定式化した．二人零和マルコフゲームとは，二人のプレーヤーの利得が環境を表す状態とお互いの行動によって決まるゲームであり，その状態遷移はマルコフ決定過程に従う．本研究では，定式化したゲームにおける近似ナッシュ均衡を導出し，その戦略と従来研究 [2] をベースとした統計モデルを比較した．

2　準備

ここでは，提案手法を理解する上で必要な，二人零和マルコフゲームの定義，求める戦略，すなわち均衡方策の性質，そして，均衡方策を求めるためのアルゴリズムについて従来研究を基に述べる．

2.1　二人零和マルコフゲーム

二人零和マルコフゲーム $\mathcal{M}$ を $\mathcal{M} = \langle S, A_1, A_2, T, R, \gamma \rangle$ と定義する [4]．ここで，S は有限状態集合，A_i はプレーヤー $i \in \{1,2\}$ の有限行動集合，$T : S \times A_1 \times A_2 \times S \to [0,1]$ は状態遷移確率関数，$R : S \times A_1 \times A_2 \times S \to \mathbb{R}$ は報酬関数，$\gamma \in [0,1)$ は割引因子を表している．プレーヤー i ではないプレーヤーを $-i$ と表す．二人のプレーヤーの行動の組を $a = (a_i, a_{-i})$ とし，行動の組の集合を $A = A_i \times A_{-i}$ とする．状態 $s \in S$ で行動の組 $a \in A$ が実行され，状態 $s' \in S$ に遷移したとき，プレーヤー 1 が得る報酬 R_1 は $R_1(s,a,s') \coloneqq R(s,a,s')$，プレーヤー 2 が得る報酬 R_2 は $R_2(s,a,s') \coloneqq -R(s,a,s')$ とする．

プレーヤー i が，状態 $s \in S$ で行動 $a_i \in A_i$ を選択する確率を表す関数を $\pi_i : S \times A_i \to [0,1]$ と定義し，これを方策とよぶ．二人のプレーヤーの方策の組を $\pi = (\pi_1, \pi_{-i})$ とする．ある状態 $s \in S$ から各プレーヤーが特定の方策 π に従って行動を選択し続けるときに，将来的に獲得可能な報酬の期待値を価値関数とし，プレーヤー $i \in \{1,2\}$ の価値関数を，

$$V_i^{\pi}(s) = \mathbb{E}\Big[\sum_{h=0}^{\infty} \gamma^h R_i(s^{(h)}, a^{(h)}, s^{(h+1)}) \mid s^{(0)} = s, \\ a^{(h)} \sim \pi(\cdot \mid s^{(h)}),\ s^{(h+1)} \sim T(\cdot \mid s^{(h)}, a^{(h)}), \forall h \geq 0 \Big],$$

と定義する．ここで $h \in [0,1,2,...]$ はゲームのステップ数を表す．二人零和マルコフゲームでは，任意の状態 $s \in$ について $V_i^{\pi}(s) = -V_{-i}^{\pi}(s)$ が成り立つ．ある状態 $s \in S$ で各プレーヤーが行動 $a_1 \in A_1, a_2 \in A_2$ を選択し，その後は特定の方策 π に従って行動を選択する場合に将来的に獲得可能な報酬の期待値を行動価値関数とし各プレーヤー $i \in \{1,2\}$ の行動価値関数を，

$$Q_i^{\pi}(s,a) = \sum_{s' \in S} \big(R(s,a,s') + \gamma T(s' \mid s,a) V_i^{\pi}(s') \big),$$

と定義する．価値関数と同様に，任意の状態 $s \in S$ と行動の組 $a \in A$ について $Q_i^{\pi}(s,a) = -Q_{-i}^{\pi}(s,a)$ が成り立つ．また，行動価値関数 Q_i^{π} を用いて価値関数 V_i^{π} を表すことができる：

$$V_i^{\pi}(s) = \sum_{a \in A} \pi(a \mid s) Q_i^{\pi}(s,a).$$

2.2　均衡方策

二人零和マルコフゲーム $\mathcal{M}$ において，以下の性質を満たす方策 π^* をナッシュ均衡の方策と呼ぶ：

$$\forall i, \forall s, \forall \pi,\ \ V_i^{\pi_i, \pi_{-i}^*}(s) \leq V_i^{\pi^*}(s) \leq V_i^{\pi_i^*, \pi_{-i}}(s).$$

定義が示すように，プレーヤーがナッシュ均衡の方策に従っている限り，各プレーヤー $i \in \{1,2\}$ は他の方策 π_i に変更する誘引を持たない．自身の獲得報酬の期待値は最大化できないものの小さくされにくい方策であるため，ナッシュ均衡の方策に従うことでゲームに負けにくくなると捉えることができる．

2.3　Nash-DQN

本研究では，近似均衡方策 π^* を求めるアルゴリズムとして，Nash-DQN [1] をベースとしたアルゴリズムを用いた（アルゴリズム 1）．Nash-DQN は，状態-行動空間が巨大な設定でもマルコフゲームの均衡計算を可能としたアルゴリズムである．各プレーヤー $i \in \{1,2\}$ の行動価値関数をパラメータ θ_1, θ_2 を用いたニューラルネットワークで近似し $Q_1^{\theta_1}, Q_2^{\theta_2}$ とする．

環境から現在の状態 $s \in S$ を観測し，各プレーヤーは確率 $\epsilon \in [0,1]$ でランダムに，確率 $1-\epsilon$ で現在の方策 π_i に従って行動を選択する．なお，現在の方策は以下の線形計画法により導出する：

$$\begin{aligned} maximize\ \ & V_i(s) \\ s.t.\ \ & \sum_{a_i \in A_i} \pi_i(a_i|s) Q_i^{\theta_i}(s, a_i, a_{-i}) \geq V_i(s) \quad \forall a_{-i} \in A_{-i} \\ & \pi_i(a_i|s) \in [0,1] \quad \forall a_i \in A_i \\ & \sum_{a_i \in A_i} \pi_i(a_i|s) = 1. \end{aligned}$$

1）　日本放送協会，Japana Broadcasting Corporation

選択した行動 a_1, a_2 をもとに次の状態 s' に遷移し，プレーヤーは報酬 r_1, r_2 を受け取る．この一連の結果 $(s, a_1, a_2, s', r_1, r_2)$ を 1 つの経験とし，メモリ M に追加する．メモリ数 $|M|$ が一定量 W を超えた場合，M から D 個データをサンプリングし，それらを用いて各プレーヤーの行動価値関数 $Q_1^{\theta_1}, Q_2^{\theta_2}$ を学習する．なお，学習における損失関数は以下の通りである：

$$\mathcal{L}(D, \theta_i) = \frac{1}{|D|} \sum_{(s,a_i,a_{-i},r_i,s') \in D} \left(Q_i^{\theta_i}(s, a_i, a_{-i}) - (r_i + \gamma V_i(s')) \right)^2.$$

r_i は実際に観測された報酬であることから $r_i + \gamma V_i(s')$ は $Q_i^{\theta_i}(s, a_i, a_{-i})$ の真値に近いと解釈し，$Q_i^{\theta_i}(s, a_i, a_{-i})$ が $r_i + \gamma V_i(s')$ となるように学習する．なお，$V_i(s')$ は先に示した線形計画法を用いて導出可能である．$Q_1^{\theta_1}, Q_2^{\theta_2}$ の学習が E 回行われるまで同様の処理を繰り返す．

3 提案手法

本章では，スタッツデータを元にした野球の二人零和マルコフゲームの定式化について述べる．プレーヤーは $i \in \{1, 2\}$ として定義する．本研究では，ビジターチーム（先攻）を $i = 1$ とし，ホームチーム（後攻）を $i = 2$ として扱う．スタッツデータには，データスタジアム株式会社が 2023 年に収集したプロ野球 261 試合分のデータを用いた．ある試合の任意のシチュエーションにおける投球結果や打撃結果が記載されており，合計で 78,327 球分のデータがある．

3.1 状態集合

本研究では，野球における各状態 $s \in S$ を 11 個の要素 $s = (in, tb, pa, np, bo, d, rs, bc, sc, oc, t)$ で表す．各要素を以下で定義する．

- $in \in \{1, 2, ..., 12\}$ は現在のイニングを表す．
- $tb \in \{1, 2\}$ は各イニングの表 (1), 裏 (2) を表す．
- $pa \in \{1, 2, ..., 10\}$ はイニング内打席数を表す．なお，打席数が 10 を超えた場合は全て 10 として扱う．
- $np \in \{1, 2, ..., 10\}$ は現在の投球が現在のバッターに対して何球目に該当するかを表す．なお，10 球を超えた場合は全て 10 として扱う．
- $bo \in \{1, 2, ..., 9\}$ は現在の打順を表す．
- $d \in \{-10, -9, ..., 9, 10\}$ はホームチームの立場における点差を表す．例えば，ビジターチームが 0 点，ホームチームが 1 点獲得している場合 $d = 1$ となる．なお，点差が-10 未満の場合は全て-10 として扱い，点差が 10 を超えた場合は全て 10 として扱う．
- $rs \in \{0, 100, 10, 1, 110, 101, 11, 111\}$ はランナーの出塁状況を表す．百の位は一塁ランナーの有無，十の位は二塁ランナーの有無，一の位は三塁ランナーの有無をそれぞれ表す．例えばランナー一，二塁の状況であれば $rs = 110$ となる．
- $bc \in \{0, 1, 2, 3\}$ は現在のボールカウントを表す．
- $sc \in \{0, 1, 2\}$ は現在のストライクカウントを表す．
- $oc \in \{0, 1, 2\}$ は現在のアウトカウントを表す．
- $t \in \{0, 1\}$ は試合終了フラグを表す．$t = 1$ の場合，試合終了である．

ここで，試合終了を表す終端状態集合を

$$S_t = \{s | s \in S, t = 1\}, \tag{1}$$

と定義する．

アルゴリズム 1 Nash-DQN

```
Initialize Q_i^{θ_i}(s, a_1, a_2) ∀i, s, a_1, a_2
M ← {}
t ← 0
s ← s^init
while e < E do
    Return a_1, a_2 uniformaly at random or with π_1, π_2
    Moving from s to s', and receiving r_1, r_2
    Add (s, a_1, a_2, r_1, r_2, s') to M
    if W < |M| then
        Sampling mini-batch D from M
        for i ∈ {1, 2} do
            Optimization Step of L(D, θ_i)
        end for
        e ← e + 1
    end if
    if s' is terminal then
        s ← s^init
    else
        s ← s'
    end if
end while
```

3.2 行動集合

野球においては，攻撃時と守備時それぞれで選択可能な行動が異なるため，攻撃時および守備時の行動集合をそれぞれ定義する．攻撃時に選択可能な行動の例としてヒッティングや盗塁がある．このような行動をスタッツデータから収集することで，攻撃時の行動集合を定義する．スタッツデータを元にした攻撃時の行動集合を

$$A_o = \{\text{Hitting, Bunt, 2ndSteel, 3rdSteel, Hit\&Run, Run\&Hit, Squeeze, SafetySqueeze}\}, \tag{2}$$

と定義する．

守備時に選択可能な行動としては，外角低めストレートを投げるといった “投球行動” と，敬遠や牽制といった “特殊行動” に分けられる．投球行動は 3 つの要素の組み合わせで定義する：

$$A_{d,p} = PitchType \times Hight \times Width. \tag{3}$$

$PitchType$ は球種を表す．スタッツデータを元にして $PitchType \in \{\text{Fast, Curve, Drop}\}$ として定義した．ここで Fast はストレート，Curve はスライダーやカットボールのような曲がる系の球種，Drop はフォークなどの落ちる系の球種を表す．$Hight \in \{\text{High, Middle, Low}\}$ は制球時の高さ位置を表し，$Width \in \{\text{Inside, Middle, Outside}\}$ は制球時の横位置を表す．以上から，投球行動数 $|A_{d,p}| = 27$ となる．特殊行動は，スタッツデータを元に以下で定義した：

$$A_{d,u} = \{\text{IntentionalWalk, 1stPickoff, 2ndPickoff}\}. \tag{4}$$

なお，3rdPickoff は，スタッツデータ上に数件しか存在しないことから，特殊行動の対象から除外した．以上か

ら，守備チームの行動集合 A_d を以下で定義する：

$$A_d = A_{d,p} \cup A_{d,u}. \tag{5}$$

各プレーヤーが選択可能な行動は状態 $s \in S$ に依存する．具体的には，現イニングが表なのか裏なのかで選択可能な行動が決まる．ある状態 $s \in S$ における表裏 $tb \in \{1,2\}$ を基準として，プレーヤー $i \in \{1,2\}$ の行動集合を

$$A_i(s) = \begin{cases} A_o & \textbf{if} \quad tb = i \\ A_d & \textbf{if} \quad tb \neq i, \end{cases} \tag{6}$$

と定義する．

3.3 状態遷移確率関数

本研究では，スタッツデータを基にニューラルネットワークを用いて状態遷移確率関数を近似する．状態遷移確率関数は，ある状態 $s \in S$ で行動の組 $a \in A$ が選択されたときに状態 $s' \in S$ に遷移する確率を表すため，パラメータ ϕ を用いて $T(s'|s,a;\phi)$ として関数近似するのが最も単純な方法である．しかし，本研究の設定では状態数は 1 億を超えるため，次の状態への遷移確率を直接学習することは非常に困難である．一方，野球のルール上，ある状態 $s \in S$ で行動 $a \in A$ が選択された結果，以下の要素が順に判明すれば次の状態が決まる．

1. 次のボールおよびストライクカウント $nbs \in \{0\text{-}0, 0\text{-}1, ..., 4\text{-}1, 4\text{-}2\}$
2. 打者変化の有無 $cb \in \{0,1\}$
3. アウトカウント増加の有無 $oi \in \{0,1,2,3\}$
4. 得点の有無 $pi \in \{0,1,2,3,4\}$
5. 次の出塁状況 $nrs \in \{0,100,10,1,110,101,11,111\}$

以上から，本研究では状態 $s \in S$ および行動 $a \in A$ を所与としたとき nbs, cb, oi, pi, nrs それぞれの生起確率を出力するモデルをパラメータ ϕ を用いて作成することで，間接的に状態遷移確率関数を表現した．

モデル構造を図 1 に示す．要素が順に決定することを考慮して Masked Attention を適用した Transformer Encoder を採用した．学習における損失関数は交差エントロピーを用いた．推論時は，まず最初に状態 $s \in S$ および行動 $a \in A$ を入力して nbs を獲得する．その後，獲得した nbs も入力として使用し cb を獲得する．同様の操作を nrs を獲得するまで繰り返す．

3.4 報酬関数

試合終了の際に，勝利したプレーヤーは報酬 10 を得，敗北したプレーヤーは報酬 -10 を得る設定とした．なお，試合終了の際に同点だった場合は両プレーヤーは報酬 0 を得る設定とした．すなわち，任意の状態 $s \in S \backslash S_t$ および 任意の行動 $a \in A$ について，

$$R_i(s,a,s') = \begin{cases} -20i + 30 & \textbf{if} \quad s' \in S_t, d' < 0 \\ 20i - 30 & \textbf{if} \quad s' \in S_t, d' > 0 \\ 0 & \textbf{if} \quad s' \in S_t, d' = 0, \end{cases} \tag{7}$$

と定義する．なお，d' は状態 $s' \in S_t$ における点差を示す．

あるプレーヤーが得点した場合は，得点をそのまま報酬として与える設定とした．すなわち，任意の状態

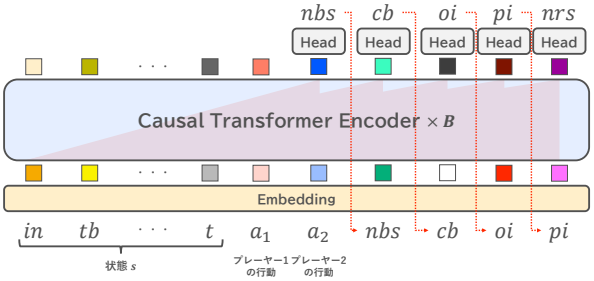

図 1　状態遷移関数モデル構造

表 1　任意の方策に従った際の各プレーヤーの勝利数

		$i=2$	
		π_2^{stat}	π_2^*
$i=1$	π_1^{stat}	(8,745, 1,049)	(8,444, 1,385)
	π_1^*	(7,738, 1,673)	(6,673, 2,489)

$s, s' \in S \backslash S_t$ および任意の行動の組 $a \in A$ について，

$$R_i(s,a,s') = (2-i)(d-d') + (i-1)(d'-d) \tag{8}$$

と定義する．なお，$i \in \{1,2\}$ であり，d は状態 $s \in S \backslash S_t$ における点差，d' は状態 $s' \in S \backslash S_t$ における点差を示す．

行動に対する報酬は，特定の行動に対するコストという形で与える設定とした．ここで，攻撃時の特定の行動は $A_{o,u} = A_o \backslash \{\text{Hitting}\}$ とし，守備時の特定の行動は $A_{d,u} = \{\text{1stPickoff}, \text{2ndPickoff}\}$ とした．いずれかのプレーヤーが特定の行動を選択した場合，行動に対するコストとして報酬 -0.1 を受け取る．なお，両プレーヤーが特定の行動を選択した場合は，報酬が相殺されるため互いが受け取る報酬は 0 となる．以上から，任意の状態 $s, s' \in S \backslash S_t$ について

$$R_i(s,a,s') = \begin{cases} -0.1 & \textbf{if} \quad tb = i, a_i \in A_{o,u}, a_{-i} \notin A_{d,u} \\ -0.1 & \textbf{if} \quad tb \neq i, a_i \in A_{d,u}, a_{-i} \notin A_{o,u} \\ 0 & \textbf{if} \quad a_i, a_{-i} \in A_{o,u} \cup A_{d,u}, \end{cases} \tag{9}$$

と定義できる．

ここまでの定義に該当しないケースでは，各プレーヤーは報酬 0 を受け取る．

4 実験

4.1 実験設定

今回は計算環境の都合上，ゲームのスタートは 1 回の表とはぜず 9 回表からとし，初期状態 $s^{init} = (9,1,1,1,2,-1,0,0,0,0,0)$ とした．$d = -1$ なので，ホームチームがが 1 点差で負けている状況である．アルゴリズムのハイパーパラメータは $E = 5.0 \times 10^5, W = 5.0 \times 10^4, \gamma = 0.99$ として計算した．また，ϵ は初期値を 1.0 とし，$E/2$ で $\epsilon = 0.1$ となるように，学習が進むにつれて線形減衰させた．

比較対象として，従来研究 [2] をベースとしたニューラルネットワークを用いたモデルによる方策を用いた．なお，本研究では，3 章で述べたスタッツデータを用いて教師あり学習をしてモデルを作成し，このモデルを統計モデルと呼ぶこととする．現在の状態を入力として，3.2 節で定義した攻撃時もしくは守備時の行動の確率分布，すなわち方策を出力する．統計モデルは，実際の試

表 2　$s = (9, 1, 1, 1, 2, -1, 0, 0, 0, 0, 0)$ における統計モデル方策と近似均衡方策

π_1^{stat}	π_1^*	π_2^{stat}	π_2^*
Hitting: 0.946	Hitting: 0.906	Fast-Outside-High: 0.259	Curve-Inside-Middle: 0.576
Bunt: 0.054	Bunt: 0.094	Fast-Middle-Middle: 0.189	Drop-Middle-Low: 0.424
		Fast-Inside-High: 0.099	

表 3　$s = (9, 1, 3, 1, 4, -1, 100, 0, 0, 1, 0)$ における統計モデル方策と近似均衡方策

π_1^{stat}	π_1^*	π_2^{stat}	π_2^*
Hitting: 0.982	Bunt: 0.713	Fast-Outside-Middle: 0.143	1stPickoff: 0.923
Bunt: 0.018	2ndSteel: 0.179	1stPickoff: 0.087	Curve-Inside-Low: 0.043
	Run&Hit: 0.108	Drop-Outside-High: 0.085	Curve-Inside-Middle: 0.034

合で実行された行動を高い確率で出力するように学習するため，相手の行動を明示的に考慮して方策を出力できていないと捉えることができる．

4.2　シミュレーションによる方策評価

各プレーヤーが任意の方策に従ってゲームを複数回行った時の勝利数を計測することで，方策の評価を行った．プレーヤー $i \in \{1, 2\}$ は，統計モデルの方策 π_i^{stat} および近似均衡方策 π_i^* のいずれかに従いゲームを行うものとした．初期状態 s^{init} から試合終了までを 1 エピソードとし，合計 10,000 エピソード実施し，各プレーヤーの勝利数を算出した．なお，引き分けの場合は各プレーヤーに対して勝利数は加算されないものとした．

表 1 に各プレーヤーが任意の方策に従ってゲームを行った際の勝利数を示す．例えば，各プレーヤーが統計モデルの方策 π_i^{stat} に従って行動選択した場合，$i = 1$ の勝利数は 8,745 であり $i = 2$ の勝利数は 1,049 であることを示している．$i = 1$ が統計モデルの方策に従っているときに $i = 2$ が近似均衡方策に従って行動を選択することで，統計モデルに従う時よりも勝利数を 336 回向上させることができた．これは，近似均衡方策が勝利を目指す上では優れていることを示しており，戦略的相互作用を考慮した上で戦略を導出することの重要性が示唆された．

一方，$i = 2$ が統計モデルの方策に従っている時に $i = 1$ が近似均衡方策に従って行動を選択した場合は，統計モデルに従うときよりも勝利数が減少していた．この要因の一つとして，今回の実験設定では π_1^* が均衡に収束しきれていないことが考えられる．2.2 節に示す均衡方策の定義上，π_1^* が均衡に収束していた場合は，少なくとも π_1^{stat} に従った際に達成可能な勝利数と同程度となることが期待される．学習回数を増やすなど，実験設定の見直しや使用するアルゴリズムの見直しが必要と考える．

4.3　統計モデルとの方策比較

ある状態 $s \in S$ における，統計モデルの方策 π^{stat} と近似均衡方策 π^* を比較する．表 2 および表 3 に，ある状態 s における各プレーヤー $i \in \{1, 2\}$ の各方策 π_i^{stat}, π_i^* 上位 3 件を示す．表 2 は初期状態 s^{init}，すなわち 9 回表ノーカウント 2 番打者の状況における各プレーヤーの戦略を示す．$i = 1$ では，両モデルとも Hitting を高い確率で選択する方策となっているため，統計モデルの方策と近似均衡方策はほぼ同じものと考えられる．一方，$i = 2$ では，統計モデルの方策ではストレートを中心とした分布となっているのに対し，近似均衡方策では変化球を主体とした分布となっている．

表 3 は 9 回表ワンアウトで，ランナー一塁の打者 4 番の状況における各プレーヤーの戦略を示す．各プレーヤーについて統計モデルの方策と近似均衡方策が大きく異なることが分かる．$i = 1$ では，統計モデルの方策では Hitting を高い確率で選択する一方，近似均衡方策では Bunt を高い確率で選択しており，次点が 2ndSteel となっている．これは学習の結果，Bunt や 2ndSteel を選択したほうが状況が好転しているケースが多いが故に選択する確率が高くなっていると考えられる．一般に，Bunt や 2ndSteel を選択した場合，作戦失敗によりアウトカウントだけが増えてしまい，状況が逆に悪化してしまうためリスクが比較的高い．直感的には統計モデルの方策のように Hitting を選択することが妥当であると考えられる．$i = 2$ についても同様に，近似均衡方策は比較的リスクの高い行動を選択する傾向が見られる．

近似均衡方策は全体的に，リスクの高い行動をしがちであることが今回の比較によりわかった．このような結果になった要因の一つとして，野球を定式化する上で考慮すべき項目が不十分であることが考えられる．例えば，今回の設定では打者や投手の個人性までの考慮はできていない．同じ 4 番打者だとしても，球団ごとに成績や調子，得意な球種が異なることが考えられるため，打率等を考慮する要素として追加する必要があると考えられる．

5　おわりに

本研究では，スタッツデータを基にした上で野球を二人零和マルコフゲームとして定式化し，そのゲームの近似ナッシュ均衡，すなわち近似均衡方策を導出した．近似均衡方策の導出には Nash-DQN をベースとしたアルゴリズムを用いた．今後，より現実に即した分析を行うために，定式化の際に考慮する要素を再考する予定である．

参考文献

[1] Philippe Casgrain, Brian Ning, and Sebastian Jaimungal and. Deep q-learning for nash equilibria: Nash-dqn. *Applied Mathematical Finance*, 29(1):62–78, 2022.

[2] Connor Douglas, Everett Witt, Mia Bendy, and Yevgeniy Vorobeychik. Computing an optimal pitching strategy in a baseball at-bat. *The International FLAIRS Conference Proceedings*, 36(1), May 2023.

[3] Bill James. *The Bill James historical baseball abstract*. Villard Books, rev. ed edition, 1988.

[4] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, pages 157––163, 1994.