

SDA 시니어 논문세션 · 4회차

Optimizing Baseball Fielder Positioning

with Consideration for Adaptable Hitters

Melville · Wise · Nielson · Mott · Archibald · Grimsman

MIT Sloan Sports Analytics Conference 2024

목차

01	Introduction	수비 위치 최적화 문제
02	Modeling Batter Distributions	타자별 타구 분포를 <u>베이지안</u> 으로 학습
03	Out Probability & Optimization	아웃 확률 모델과 수비 위치 <u>최적화</u>
04	<u>Game Theory</u> for Adaptive Hitters	적응형 타자에 대한 균형 위치
05	Discussion & Implications	한계 · 정책 함의

01

CHAPTER 01

Introduction

문제 정의와 4단계 해결 구조

왜 수비 위치 최적화가 중요한가?

야구 수비의 요소

- 임무: 타자 아웃 & 실점 최소화
- 투수 · 포수 외 7명의 야수를 어느 곳에 세울것인가?
- 해당 위치 결정이 Hit/Out을 직접 좌우

논문의 접근 방식

- 통계 (Bayesian, Logistic)
- ML (XGBoost, SGD)
- 게임이론

시프트 / 극단 시프트

시프트?

- 타자에 맞게 수비의 위치 조정
- 극단 시프트 지속적 발생

극단 시프트 규제

- 내야수 모두 Infield Dirt 위에 위치
- 2루 기준 양쪽에 각각 2명씩 배치
- 타구 시점에 위 조건 만족 (위반 시 공격 어드벤처지 부여)
- 자신의 영역 내 위치 조정 가능 (유격수가 2루 베이스와 가까이)



극단적 시프트 사례 (내야 5인 / 외야 2인 시프트)

여전히 내야 5인 시프트는 가능(KBO, MLB 모두)

02

CHAPTER 02

Modeling Batter Distributions

타자별 타구 분포 $p(h, v, s)$ 학습

용어

Spray angle (h)	타구 좌우 각도 · -45° (3루) ~ $+45^{\circ}$ (1루)
Launch angle (v)	타구 상하 각도 · ground ball < 10° < line drive < 25° < fly ball < 50° < pop-up
Exit speed (s)	타구 속도 (mph) · 평균 90 mph 부근
Pulled vs Oppo	당겨치기 vs 밀어치기 · 좌·우타자 부호 통일
wOBA	안타의 평균 점수 가치 측정 지표 · 가중치 - 1B 0.88, 2B 1.24, 3B 1.57, HR 2.00, out 0

통계 분포

Beta(α, β)

Support: (0, 1)

$$f_X(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

Spray angle - p(h)

[-55°, +55°] → [0, 1] scale

$\mathcal{N}(\mu, \sigma)$

Support: $(-\infty, \infty)$

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

launch angle - p(vlh)

타구 상하 각도 - 평균 주변에 대칭으로 분포

Gamma(α, β)

Support: (0, ∞)

$$f_X(x) = \begin{cases} \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} \exp\left(-\frac{1}{\beta}x\right) \\ 0 \end{cases}$$

Exit speed- p(slv)

타구 속도 - 한 쪽 꼬리가 긴 비대칭 분포

z = s_max - s 변환

타자별 타구 결합 분포: $p(h, v, s)$

$$\begin{aligned} p(h, v, s) &= p(h) p(v | h) p(s | v, h) \\ &= p(h) p(v | h) p(s | v) \end{aligned}$$

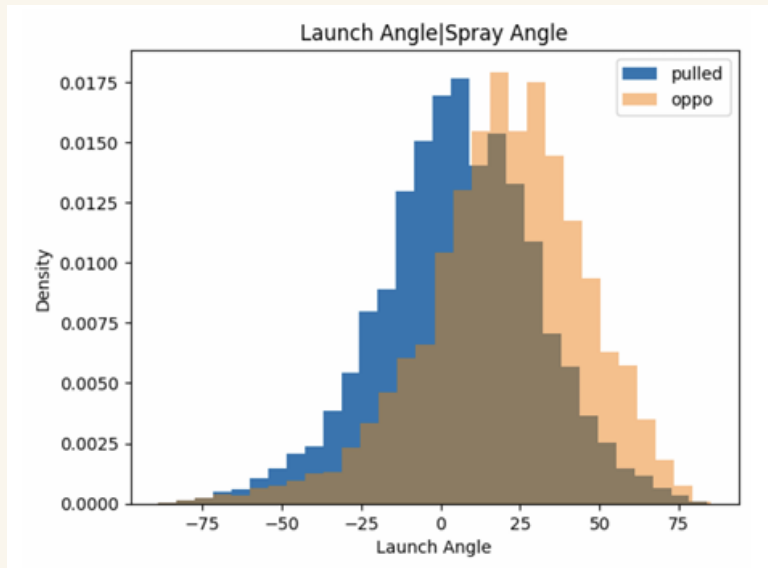
Trick 1: Chain rule

변수 2개: $p(x, y) = p(x) p(y|x)$
3개 변수 결합분포 $\rightarrow$ 1차원 추정 3개로 분해

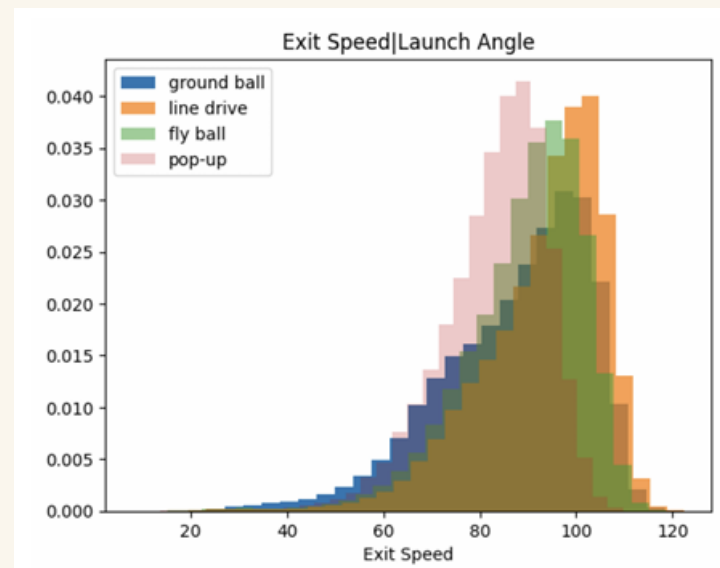
Trick 2: Conditional Independence

$s \perp h | v \rightarrow p(s | v, h) = p(s | v)$
해석: v 를 알면 h 는 s 에 추가 정보 없음 [Figure 3]

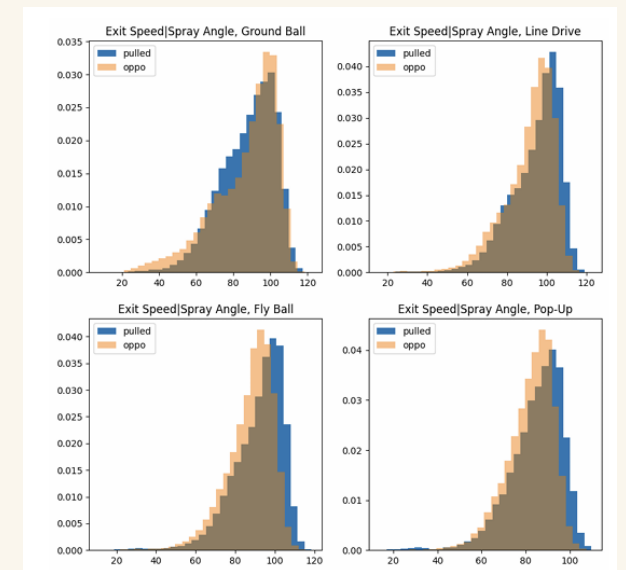
타자별 타구 결합 분포: $p(h, v, s)$

Figure 1 - $p(h)$ 

Pulled vs oppo로 launch angle 분포 다름
 $\rightarrow v$ 는 h 에 종속

Figure 2 - $p(v|h)$ 

GB/line/fly/pop으로 exit speed 분포 다름
 $\rightarrow s$ 는 v 에 종속

Figure 3 - $p(s|v, h)$ 

Launch angle을 추가로
 Conditional해도 모양이 거의 동일
 $\rightarrow s \perp h | v$

[1] Spray angle: $p(h)$ - Beta mixture

좌타/우타 부호 통일 $\rightarrow [0,1]$ 로 scale $\rightarrow$ 2개 Beta 의 mixture 로 fit

Bayesian Hierarchical 구조

Hyperprior $\rightarrow$ Dirichlet, Gamma, Half-Normal

↓

Prior $\rightarrow$ 타자별 π_i, α_i, β_i

↓

Likelihood 식

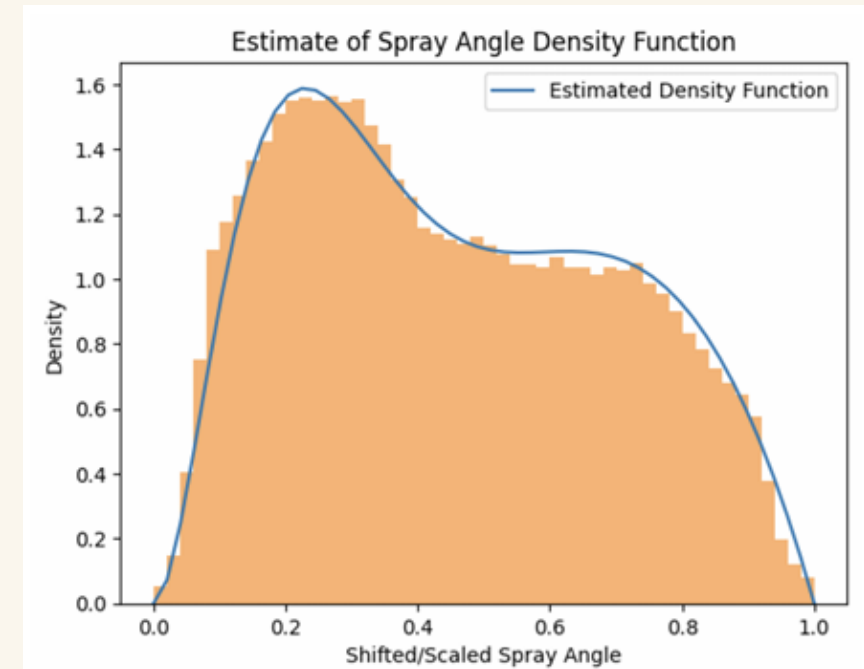
$$h_i \sim \pi_{i1} \cdot \text{Beta}(\alpha_{i1}, \beta_{i1}) + \pi_{i2} \cdot \text{Beta}(\alpha_{i2}, \beta_{i2})$$

↑

Data $\rightarrow$ 타자 i의 실제 Spray angle 관측값 (h_i)

↓

Posterior $\rightarrow p(\pi_i, \alpha_i, \beta_i \mid h_i \text{ 데이터})$



집단 평균 spray angle 분포 — 두 봉우리(pulled · oppo)
봉우리 1 (가중치 0.4) - 강한 pulled / 봉우리 2 (가중치 0.6) - 약한 oppo

[2] Launch angle: $p(v|h)$

두 개의 Normal 분포 — pulled / oppo 분기

Bayesian Hierarchical 구조

Prior → $\mu_i \sim N([5.1, 21], [5.2, 5.7])$ (각 타자 mean)

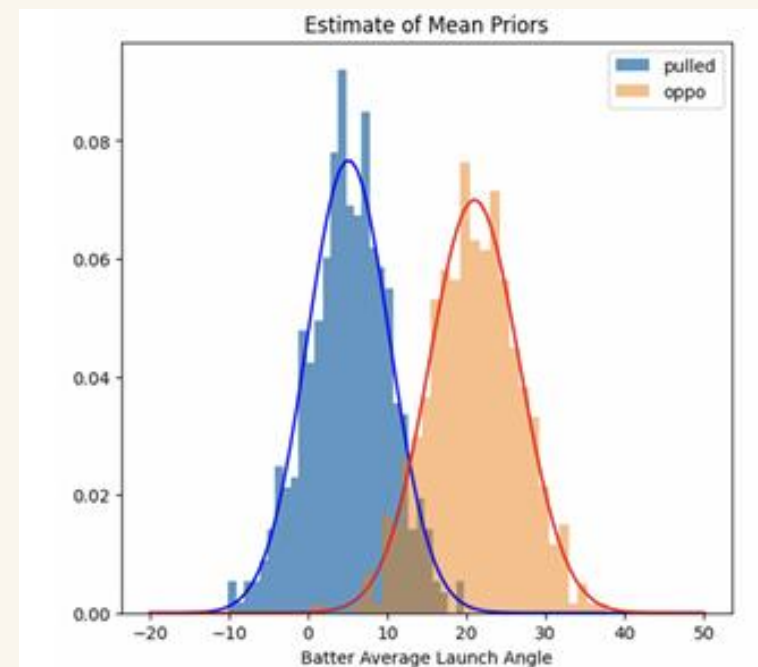
↓ $\sigma_i \sim \text{Gamma}(\alpha = [137.9, 76.8], \beta = [5.4, 2.9])$ (각 타자 std)

Likelihood 식

$$v_i | h_i \sim \begin{cases} N(\mu_{i1}, \sigma_{i1}) & h_i < 0 \quad (\text{pulled}) \\ N(\mu_{i2}, \sigma_{i2}) & h_i \geq 0 \quad (\text{oppo}) \end{cases}$$

↑ Data → 타자 i의 실제 launch angle 관측값 (v_i), h_i 는 조건

↓ Posterior → $p(\mu_i, \sigma_i | v_i, h_i \text{ 데이터})$



“Full-hitter(당겨치기) - 평균 5°·표준편차 5°,”

Oppo(밀어치기) - 평균 21°·표준편차 6°”

[3] Exit Speed: $p(s|v)$

네 개의 Gamma 분포 — launch angle 카테고리별 분기

Bayesian Hierarchical 구조

Prior $\rightarrow \alpha_i \sim \text{Gamma}(\mu = [6.0, 6.9, 9.1, 16.7], \sigma = [1.5, 2.4, 3.2, 6.1])$

$\downarrow$
 $1/\beta_i \sim \text{Gamma}(\mu = [6.4, 4.6, 4.0, 2.6], \sigma = [1.6, 1.4, 1.3, 0.8])$

Likelihood 식

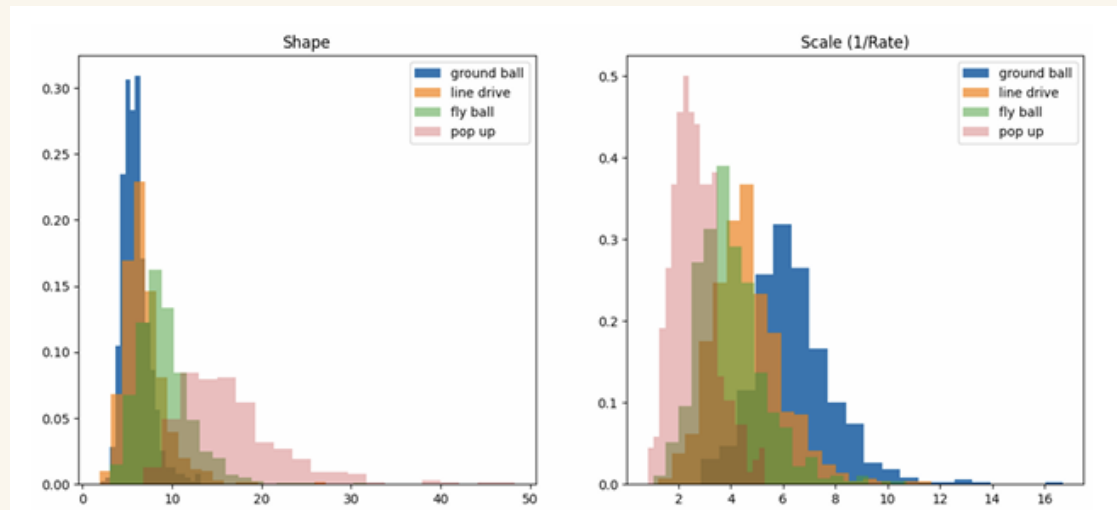
$$z_i | v_i \sim \begin{cases} \text{Gamma}(\alpha_{i1}, \beta_{i1}) & v_i \leq 10 & \text{(Ground Ball)} \\ \text{Gamma}(\alpha_{i2}, \beta_{i2}) & 10 < v_i \leq 25 & \text{(line drive)} \\ \text{Gamma}(\alpha_{i3}, \beta_{i3}) & 25 < v_i \leq 50 & \text{(fly ball)} \\ \text{Gamma}(\alpha_{i4}, \beta_{i4}) & v_i > 50 & \text{(pop-up)} \end{cases}$$

$\uparrow$

Data $\rightarrow$ 타자 i 의 변환된 exit speed 관측값 ($z = s_{\max} - s$)

$\downarrow$

Posterior $\rightarrow p(\alpha_i, \beta_i | z_i, v_i \text{ 데이터})$



변환 트릭: $z = s_{\max} - s$ Gamma는 $(0, \infty)$ right-skewed에 적합

$p_k(h, v, s)$ 의 의미

한 타자(k)가 칠 수 있는 모든 가상 타구의 분포

3개 분포의 결합

$p(h)$ Beta mixture · 타자가 어디로 칠지

$p(v|h)$ Normal $\times 2$ · 발사각도 (pulled/oppo)

$p(s|v)$ Gamma $\times 4$ · 속도 (4 카테고리)

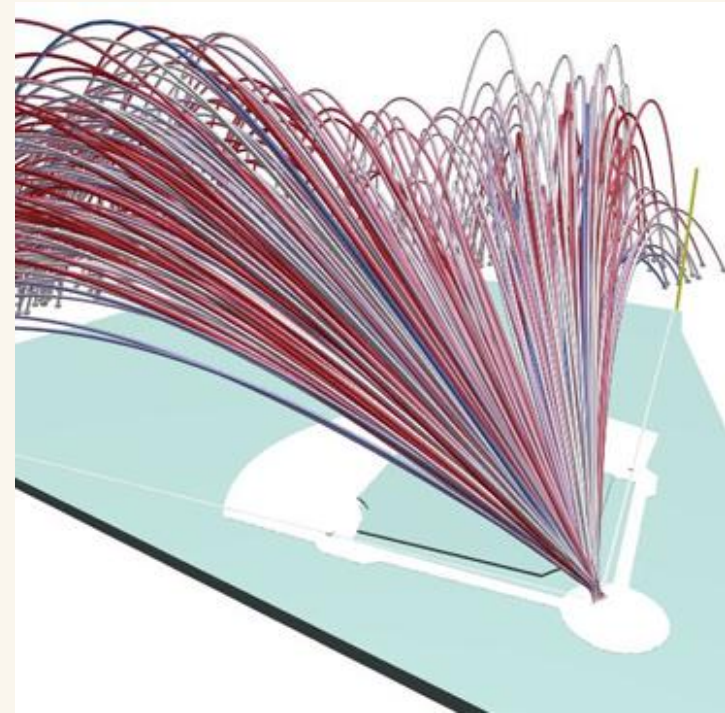
Chapter 3

가상타구 sampling $b_j = (h_j, v_j, s_j) \sim p_k(h, v, s), \quad j = 1, 2, \dots, N$

→ 아웃 확률 $f(b_j, p_s) = f_{gb}(b_j, p_s) \cdot \mathbb{I}_{gb}(b_j) + f_{\neg gb}(b_j, p_s) \cdot (1 - \mathbb{I}_{gb}(b_j))$

$$\mathbb{E}_{b \sim p_k}[f(b, p_s)] \approx \frac{1}{N} \sum_{j=1}^N f(b_j, p_s)$$

→ 최적 수비 위치 $p_s^* \in \arg \max_{p_s \in P_s} \mathbb{E}_{b \sim p_k}[f(b, p_s)]$



3D 타구 궤적 분포

03

CHAPTER 03

Out Probability & Optimization

아웃 확률 모델 + Projected SGD 최적화

[1] 가상 타구 sampling

$p_k(h, v, s)$ 분포 $\rightarrow$ 입력 변수

$$b_j = (h_j, v_j, s_j) \sim p_k(h, v, s), \quad j = 1, 2, \dots, N$$

Notation

$b = (h, v, s)$ 한 개의 타구 (3-tuple)

b_j j번째 가상타구 $\cdot j = 1, 2, \dots, N$

$d = p_k(h, v, s) \cdot$ 타구 분포 (어떤 분포에서 b 를 뽑는지)

분포에서 N개의 가상 타구를 뽑아 Section 3 입력 변수로 기능

[2] Model - 땅볼 아웃 (1루 송구)

Logistic Regression - 입력 변수 : a_d & b_t

Logistic Regression

이항 결과의 확률 모델

$Y \in \{0, 1\}$ 종속 변수(결과)가 1 아니면 0

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \dots$$

Sigmoid σ : Logistic 활성화 함수

ML 관점 — 선형 결합을 (0, 1) 확률로 변환

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad \text{입력 } (-\infty, \infty) \rightarrow \text{출력 } (0, 1)$$

[2] Model - 땅볼 아웃 (1루 송구)

Logistic Regression - 입력 변수 : a_d & b_t

$$f_{gb}(b, p_s) = \sigma(-2.78 - 0.15 a_d + 5.87 b_t - 1.32 b_t^2)$$

a_d spray angle 차이 (야수와 타구) · 값 낮을수록 좋음

b_t 공이 야수 깊이 도달 시간 = depth / exit_speed

b_t^2 역U — 너무 짧으면 반응 X, 너무 길면 송구 X. 최적 $b_t \approx 2.2$ 초.

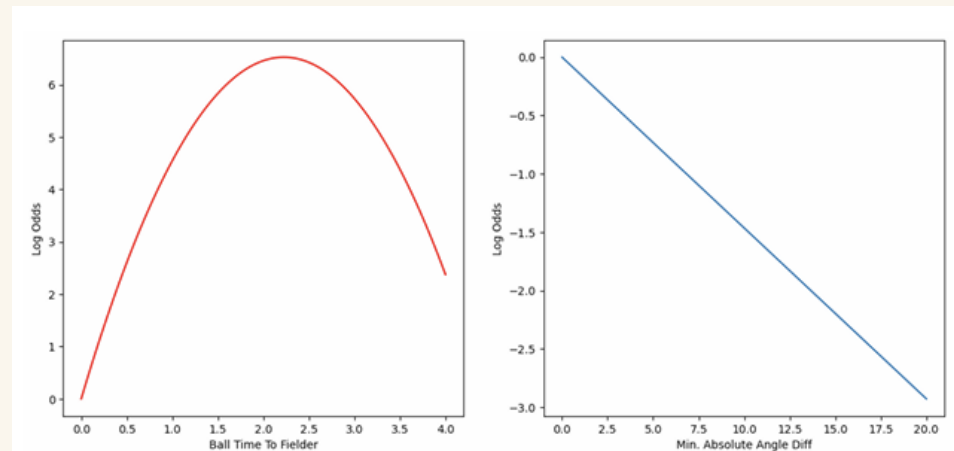


Figure 9 왼쪽: b_t 효과 (역U) · 오른쪽: a_d 효과 (단조 감소)

[3] Model - 땅볼 외 아웃 (line drive, fly ball, pop-up)

공중 catch mechanism - 입력 변수: δ_t

$$\delta_t = \text{hangtime} - \frac{\text{distance}}{27}$$

Hangtime: 공이 떨어지기까지 걸리는 시간

Distance: 공 낙하 지점까지의 거리

"공이 떨어지기 전에, 야수가 도달할 시간 여유가 얼마인가" · 27 ft/s = MLB 평균 sprint speed

$$\delta_t > 0$$

: 공이 떨어지는 시간 > 야수가 평균 속도로 도달하는 시간: 아웃 가능성 ↑

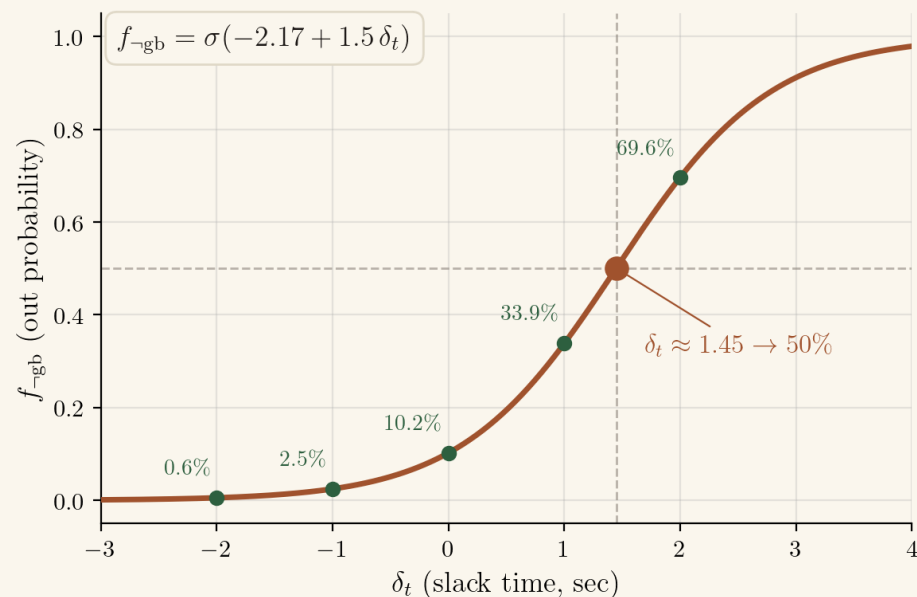
$$\delta_t < 0$$

: 공이 떨어지는 시간 < 야수가 평균 속도로 도달하는 시간: 아웃 가능성 ↓ (다이빙, 평균 보다 빠른 속도 등 필요)

[3] Model - 땅볼 외 아웃 (line drive, fly ball, pop-up)

XGBoost + Logistic Regression

$$f_{\neg\text{gb}}(b, p_s) = \sigma(-2.17 + 1.5 \delta_t)$$



δ_t	$f_{\neg\text{gb}}$ (out prob.)
$\delta_t = -2$	0.6 %
$\delta_t = -1$	2.5 %
$\delta_t = 0$	10.2 %
$\delta_t = +1$	33.9 %
$\delta_t = +1.45$	50.0 %
$\delta_t = +2$	69.6 %
$\delta_t = +3$	91.1 %

threshold (50%) at $\delta_t \approx 1.45$

[3] Model - 땅볼 외 아웃 (line drive, fly ball, pop-up)

XGBoost + Logistic Regression

$$f_{\neg\text{gb}}(b, p_s) = \sigma(-2.17 + 1.5 \delta_t)$$

STAGE 1

XGBoost

Input:

h, v, s (spray, launch, exit speed)

Output:

hang time, landing distance

δ_t

= hangtime - distance/27

STAGE 2

Logistic Regression

Input:

δ_t (slack time)

Output:

$f_{\neg\text{gb}} = \sigma(-2.17 + 1.5 \delta_t)$

Ground Truth (training data)

Alan Nathan baseball calculator — *ballistic physics simulation*

[4] 통합 아웃 Model

Indicator 함수

$$f(b, p_s) = f_{gb}(b, p_s) P(g = gb) + f_{\neg gb}(b, p_s) P(g = \neg gb)$$

$$I_{gb}(b) = \begin{cases} 1 & \text{if } g = gb \\ 0 & \text{if } g = \neg gb \end{cases}$$

관측된 b 가 땅불인지 아닌지 여부: 데이터에서 이미 아는 정보

→ $P(g = gb \mid b) \in \{0, 1\}$: deterministic indicator $I_{gb}(b)$ 로 표기

→ total probability law 의 두 항이 정확히 한쪽만 살아남는 형태

⇒ 학습 단계는 분리, 최종 모델은 한 식으로 통합

[5] 목적함수 2가지

Out 최대화 vs 실점(wOBA) 최소화

(1) Outs 최대화

$$p_s^* \in \arg \max_{p_s \in P_s} \mathbb{E}_{b \sim d} [f(b, p_s)]$$

: 타자의 타구 분포 $p(h, v, s)$ 에서 기대 out 확률이 가장 높은 위치

안타, 2루타 모두 같은 'hit'으로 간주
아웃 빈도만 확인

(2) wOBA 최소화

$$p_s^* \in \arg \min_{p_s \in P_s} \mathbb{E}_{b \sim d} [(1 - f(b, p_s)) \cdot f_{\text{wOBA}}(b)]$$

: Out이 안 났을 때 기대 wOBA를 가중하여 최소화

단타와 홈런을 다른 가중치로 파악
(0.883 / 1.244 / 1.569 / 2.004)

→ wOBA 최소화쪽이 Out도 더 잘잡는 결과, 해당 모델 채택.

[6] Projected SGD

Stochastic Gradient Descent(표본 기반 경사 하강법) + Shift Rule (시프트 룰)

3-Step

[1] Gradient Descent — 가장 가파른 방향으로 한 걸음

최댓값은 $+\nabla$ 방향, 최솟값은 $-\nabla$. η 는 step size (learning rate).

[2] Stochastic — 분포 전체 대신 sample로 gradient 근사

기댓값 $\mathbb{E}_{b \sim p_k}[f]$ 정확 계산 어려움 $\rightarrow$ 가상 타구 b_j 몇 개로 평균

[3] Projected — 시프트 룰 제약 강제

매 step 후 가장 가까운 P_s 안 위치로 사영 (1루수, 홈 위, 균형)

P_s — 시프트 룰 제약 (2023 MLB)

매 step 후 세 조건이 모두 만족되도록 사영 (projection)

- ① 1루수 1루로부터 40 ft 이내
- ② 모든 내야수 pitching rubber로부터 95 ft 이내 (= infield dirt 안)
- ③ 2루 양쪽에 각각 2명씩 배치 (2023 시프트 금지 룰)

$\rightarrow$ 조건 위반 시 가장 가까운 허용 위치로 사영

업데이트 규칙 — 매 iteration

Gradient step: $\tilde{p}_s \leftarrow p_s + \eta \cdot \nabla_{p_s} \hat{f}(b_j, p_s)$

Projection step: $p_s \leftarrow \text{Proj}_{P_s}(\tilde{p}_s)$

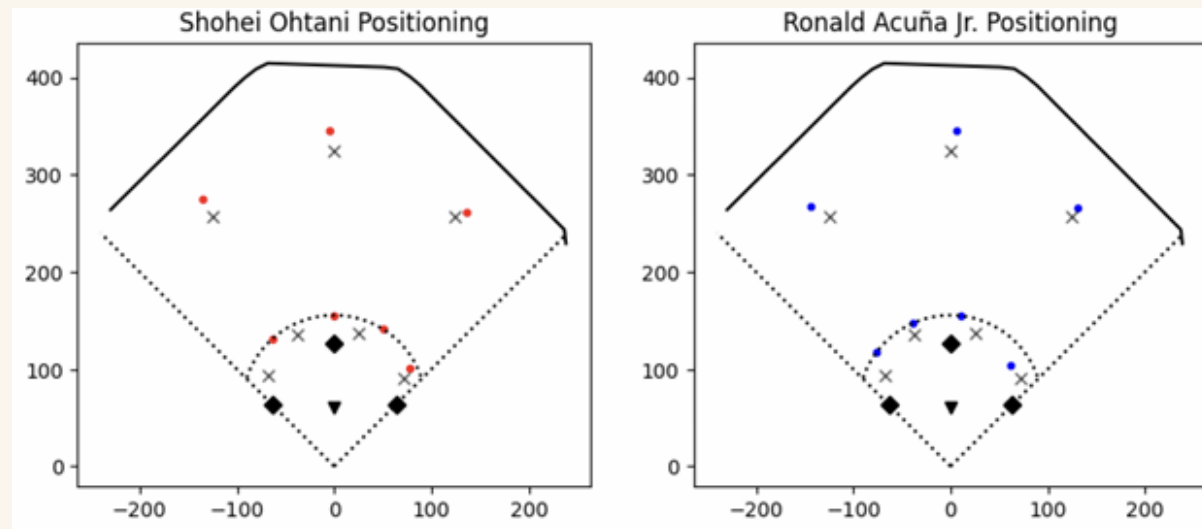
두 단계 — gradient step + projection step

Gradient — Monte Carlo 추정

$$\nabla_{p_s} \mathbb{E}_{b \sim p_k}[f(b, p_s)] \approx \frac{1}{N} \sum_{j=1}^N \nabla_{p_s} f(b_j, p_s)$$

[7] Optimization Result

수비 최적 위치 탐색



2023 시즌 적용 결과

	xBA	xwOBA	wRAA	wRAA/팀
Model	0.305	0.308	-520	-17.3
MLB Avg	0.320	0.321	+156	+5.2

→ 팀당 22.5점 절감 ≈ 약 2.25 wins (WAR)

Texas Rangers proprietary 데이터로 검증

모델 BA 0.292 ≈ 실제 BA 0.293 ✓

→ xBA: 기대 타율, xwOBA: 기대 wOBA, wRAA: 평균 대비 가산 득점, wRAA/팀: 팀당 wRAA

04

CHAPTER 04

Game Theory for Adaptive Hitters

타자가 위치를 보고 적응한다면?

Game Theory

타자 분포 고정 → 현실의 타자는 야수 위치를 보고 적응

(1) 정적 optimization

타자 타구 분포: $p_k(h, v, s)$ (하나의 분포, 고정)

야수 → 그 분포에 대해 최적 위치 탐색

한계

"타자가 시프트 보고 *oppo*로 칠 수도 있는데?"

"번트 위협이 있는 타자는?"

(2) Zero-sum Game

타자 타구 분포: $d \in \mathcal{D}$ (분포의 집합 — 타자가 선택)

야수 먼저 위치 선택 → 타자가 보고 분포 선택

균형

Nash equilibrium — 양쪽 다 최선의 응답

타자 적응을 가정한 *robust*한 위치

Zero-sum Sequential Game + Equilibrium

수비 vs 타자: 합리적 선택

균형 식 — Outs 최대화 (식 3a, 3b)

야수 입장 — minimax

$$p_s^* \in \arg \max_{p_s \in P_s} \min_{d \in \mathcal{D}} \mathbb{E}_{b \sim d} [f(b, p_s)]$$

“타자의 최선 대응을 가정한 야수의 최선(robust) 위치”

타자 입장 — maximin

$$d^* \in \arg \min_{d \in \mathcal{D}} \max_{p_s \in P_s} \mathbb{E}_{b \sim d} [f(b, p_s)]$$

“야수의 최선 대응을 가정한 타자의 최선 분포 (outs 최소)”

**두 식이 동시에 만족되는
(p_s^* , d^*) 가 Nash 균형**

$\mathcal{D}$ 분포의 집합 — 타자 행동 범위

d^* 균형 분포 — 게임 결과 타자가 선택

p_s^* 균형 위치 — 게임 결과 야수가 선택

Case 1. Shift-Beater Oppo

시프트 보고 밀어치는(oppo) 타자 · Pull vs Oppo의 균형

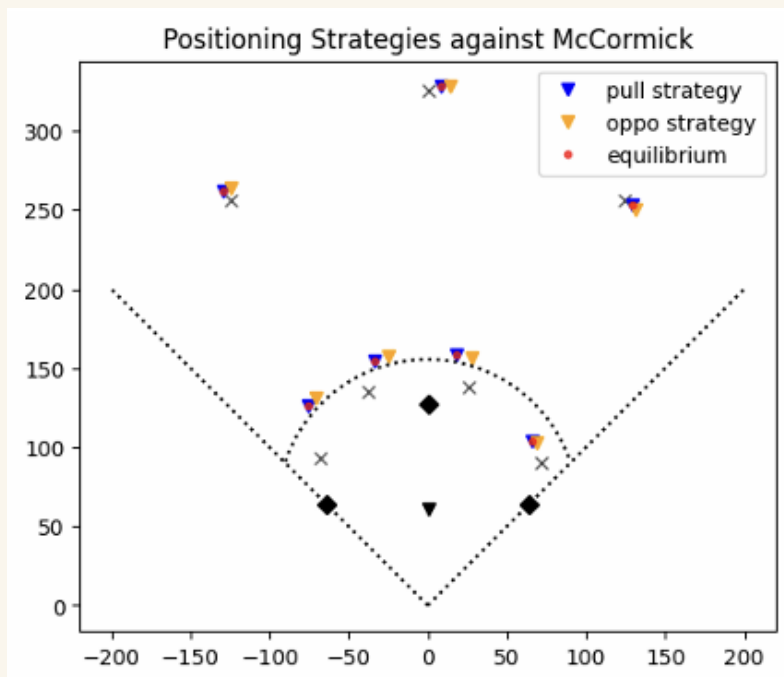


Figure 16 예시: Chas McCormick — 균형(빨강) vs pull/oppo 각각의 최적(파랑/주황)

- (1) Shift-Beater 가정해도 균형이 한쪽으로 편향 — 타자가 항상 pull이 최선
- (2) McCormick의 시프트 적응(oppo)이 사실 자기에게 손해

야수 시프트 보고 oppo 방향으로 칠 의지·능력 있는 타자

ex) Josh Bell, Chas McCormick, Adam Frazier 등

수비 대응, 타자 행동

$$\mathcal{D} = \{d_{\text{pull}}, d_{\text{oppo}}\}$$

Nash Equilibrium Equation

$$p_s^* \in \arg \max_{p_s \in P_s} \min_{d \in \{d_{\text{pull}}, d_{\text{oppo}}\}} \frac{1}{N} \sum_{j=1}^N f(b_j, p_s)$$

결과 — McCormick의 expected BA

야수 \ 타자	pull	oppo
pull 대응	0.274	0.271
oppo 대응	0.275	0.249

→ pull이 oppo를 dominate, 균형은 (야수: pull 대응, 타자: pull)

Case 2. Shift-Beating Bunt

시프트 보고 번트하는 타자 · Swing vs Bunt의 균형

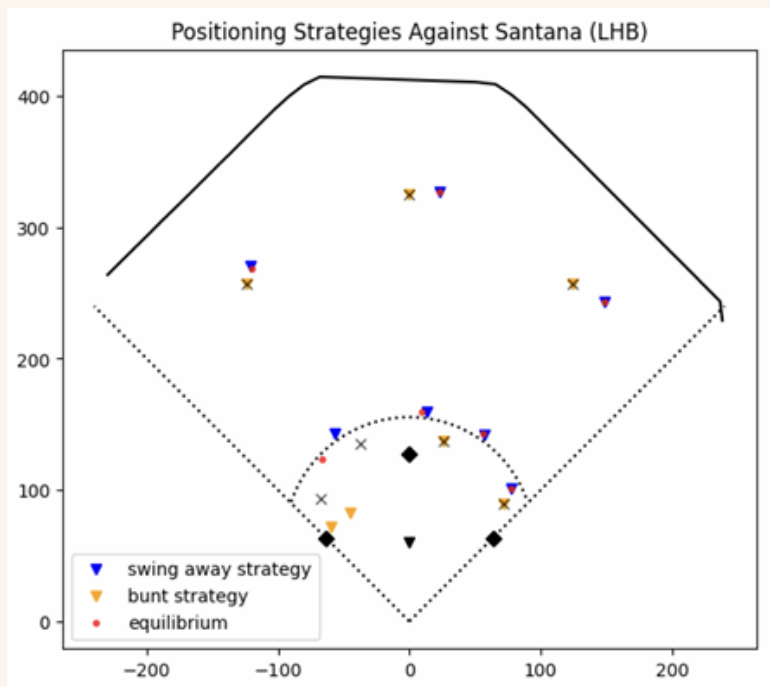


Figure 18 예시: Carlos Santana(좌타) — 균형 위치(빨강), swing 대응(파랑)에서 살짝 3루 쪽 이동

Pull-heavy 타자가 시프트 깨려 3루 라인 쪽 번트 가능성 有

ex) Left Hitter — 번트 시 3루 라인이 비어있음 활용

수비 대응, 타자 행동

$$\mathcal{D} = \{d_{\text{swing}}, d_{\text{bunt}}\}$$

Nash Equilibrium Equation

$$p_s^* \in \arg \max_{p_s \in P_s} \min_{d \in \{d_{\text{swing}}, d_{\text{bunt}}\}} \frac{1}{N} \sum_{j=1}^N f(b_j, p_s)$$

결과 — Santana의 expected BA

야수 \ 타자	swing	bunt
blue (시프트)	0.268	0.453
orange (번트 대비)	0.354	0.062
equilibrium	0.272	—

← 시프트가 깨짐

→ 번트할 의지만 있어도 야수는 균형 위치로 이동

$$\text{swing}(0.272) > \text{swing against blue}(0.268) - 0.004 \uparrow$$

3루수 위치 — 절충

깊이 알게 + 3루 라인 쪽으로

번트 대비 ↔ swing 막기 사이의 애매한 절충 위치

05

CHAPTER 05

Discussion & Implications

강점 · 한계 · KBO 적용 가능성

논문의 함의

방법론: adaptive 타자에 대한 robust 수비 최적화 프레임워크

01

적응형 타자

Game Theory로 모델링

"타자가 시프트 보고
opposite으로 칠 수 있다면?"

"번트 위협이 있다면?"

→ minimax 균형으로
타자 적응까지 흡수

02

일반화 프레임워크

D 집합만 바꾸면 적용

$D = \{\text{pull, oppo}\}$
→ Shift-Beater

$D = \{\text{swing, bunt}\}$
→ 시프트 깨는 번트

어떤 적응형 타자에도
동일 패턴 적용 가능

03

공개 데이터로 작동

Statcast만으로 의미있는 결과

팀당 22.5점 절감

≈ 약 2.25 wins (WAR)

Texas Rangers 구단 데이터 검증
BA 0.292 ≈ 실제 0.293

연구실 시뮬 X — 실용 가능

KBO 승부예측 대회 활용 가능성

좌표 최적화 ≠ 승률 예측 · 그러나 모델링 패턴은 transferable

01

선수 단위 분포 모델링

논문

타자별로 spray/launch/exit 분포 학습

대회 적용

현재: 팀 단위 통계 위주 (home_prior_ops 등)

→ 선수 단위 정보를 라인업 feature로 집계

모델별 활용

XGBoost/LightGBM: player ID + target encoding

딥러닝: player embedding

BART: hierarchical prior (자체)

02

Two-stage — 도메인 + ML

논문

물리 시뮬 → XGBoost → Logistic

대회 적용

도메인 지식을 sub-model로 학습

→ sub-model 출력을 메인 모델 feature로

적용 예시

- 라인업 platoon advantage

- 볼펜 피로도 (recent3d 활용)

- 구장/날씨 효과

→ 어떤 ML 도구든 sub-model로 활용 가능