Modeling the Probability of a Successful Stolen Base Attempt in Major League Baseball

Cade Stanley
Director of Thesis: Dr. Joshua Tebbs
Second Reader: Dr. Ting Fung Ma

University of South Carolina Honors College

April 13, 2023

UNIVERSITY OF South Carolina
South Carolina Honors College

Introduction

Background

- In sports, teams constantly search for a competitive edge
  - In the front office: evaluating talent, spending money wisely
  - On the field: selecting optimal lineups and strategies, making sound in-game decisions
- An important decision made several times a game in the MLB: whether to attempt to steal second base
  - Success: the runner reaches "scoring position," where any hit will likely score the runner
  - Failure: the runner is removed from the basepaths, and an out is recorded

Existing Research

- Existing research: identifying the minimum success rate needed for stolen base attempts to be worthwhile
  - Failed attempts are more harmful than successful attempts are helpful
  - In general: success rate of 75% is needed to add positive value (MLB-Advanced-Media)
  - Keyes (2022): identifies "breakeven" success rate for specific situations (according to the number of outs and the runners on base)
- Less focus on estimating the probability of success of a particular stolen base attempt
  - With an estimate of the likelihood of success, previous research can be used to make a decision rule for attempting to steal

Approach

- Binary classification models – logistic regression and random forests
- Use data about the game situation and the players involved in the stolen base attempt to predict the outcome
  - Baserunner speed, catcher arm strength
  - Pitcher and batter handedness
  - Number of outs, number of balls and strikes
  - Number of pickoff throws, number of pitchouts
  - Presence of a runner on third base
  - Type and speed of the pitch thrown

Methodology

Data Collection

- Retrosheet's play-by-play game files
  - Lineups, at-bats, and plays from every game of the 2018 MLB season
- Paul Schale's pitch data sets on Kaggle
  - Type, speed, and location data for every pitch from the 2018 MLB season
- Baseball Savant's Statcast data sets
  - Player attributes, including baserunner speed and catcher arm strength
  - Measures of players' historical success with stolen base attempts (2017 season)
- Baseball Reference's player data sets
  - Success rates for catchers at defending against stolen bases in 2017 (not included in Statcast data)

Data Processing

- **Goal:** record the game situation at the time of every stolen base attempt during the 2018 season
- **Problem:** Retrosheet event files are fairly simple – they only encode the batter name, the sequence of pitches, and the outcome for each play
- **Solution:** Use Python scripts to keep an updated account of the game situation after each play
  - Record the kinds of pitches preceding each play
  - Update the number of outs after each play
  - Track the movement of runners around the basepaths
  - Log any substitutions of pitchers, catchers, and baserunners

Data Processing – Example 1

```python
# count outs and keep track of runners reaching base and advancing
play = record[6]

batter_play = play.split('.')[0]
basic_play = batter_play.split('/')[0]

if('CS' in basic_play and 'POCS' not in basic_play):
    is_stolen_base_attempt = True
    if('E' not in basic_play[basic_play.find('CS'):]):
        is_successful = False
    else:
        is_successful = True
elif('SB' in basic_play):
    is_stolen_base_attempt = True
    is_successful = True
elif('POCS' in basic_play):
    is_stolen_base_attempt = True
    if('E' not in basic_play[basic_play.find('POCS'):]):
        is_successful = False
    else:
        is_successful = True
else:
    is_stolen_base_attempt = False
    is_successful = False
```

Figure: Python code to identify stolen base attempts in the Retrosheet play-by-play data.

Data Processing – Example 2

```python
# interference
if(batter_play in ['C/E2', 'C/E1', 'C/E3']):
    if(not batter_advance_noted):
        on_first = True
        runner_on_first = batter

# single
elif(basic_play[0] == 'S' and basic_play[0:2] != 'SB'):
    if(not batter_advance_noted):
        on_first = True
        runner_on_first = batter

# double or ground rule double
elif(basic_play[0] == 'D'):
    if(not batter_advance_noted):
        on_second = True
        runner_on_second = batter

# triple
elif(basic_play[0] == 'T'):
    if(not batter_advance_noted):
        on_third = True
        runner_on_third = batter

# error
elif(basic_play[0] == 'E'):
    if(not batter_advance_noted):
        on_first = True
        runner_on_first = batter
```

Figure: Python code to log a play and record the batter's advance to the appropriate base.

Data Merging

- With the game situation of every stolen base attempt recorded, it remained to merge the rest of the data
- Match pitch data (Kaggle) with the stolen base attempt on which it occurred
  - merge on the game (home team, away team, date/time), the batter, the number of outs, and the number of pitches in the at-bat
- Match player data (Statcast, Baseball Reference) with each stolen base attempt involving that player
  - Merge on player names
  - Some name discrepancies – corrected these by hand

Data Analysis

- Using pitch data improves prediction, but has problems:
  - Introduces missing observations – about 200 (out of 2800) stolen base attempts did not occur on a pitch
  - Cannot be used for making in-game decisions – coaches and players cannot know what kind of pitch is coming
- Solution: perform the analysis with two data sets – one with pitch data and one without
- Train a logistic regression model and a random forest on each of the two data sets, for a total of four models
  - Use cross-validation to evaluate prediction performance

Results

Model Performance

	Logistic Regression	Random Forest
AUC	**0.6915**	0.6561
Prediction Accuracy	0.7611	**0.7613**
Sensitivity	0.9678	**0.9684**
Specificity	0.1574	0.1564

Table: Performance of the models including the pitch data.

	Logistic Regression	Random Forest
AUC	0.6759	0.6638
Prediction Accuracy	0.7245	0.7361
Sensitivity	0.9585	0.9400
Specificity	0.1381	**0.2253**

Table: Performance of the models excluding the pitch data.

Conclusion

Limitations

- For predictors measuring a player's historical success, we only used data from the 2017 season
  - Using full career data would better represent a player's ability and would reduce the number of missing values
- These models may not perform well in the aftermath of 2023 MLB rule changes
  - Bigger bases, limits on pickoff throws may improve the viability of stolen base attempts in general
  - Revisit these models, re-train them with data from 2023 and beyond

Future Work

- Improve model performance
  - Explore additional predictors: pitcher's delivery time, runner's lead distance
  - Try different methods: support vector machines, k-nearest neighbors, neural networks
- Identify an optimal cutoff value for classification
  - Improve specificity to account for the cost of misclassifying an unsuccessful attempt as successful
- Evaluate the reliability of these models' probability estimates (not just their binary predictions)
  - For example, stolen base attempts for which the models estimate a 70% probability of success should be successful around 70% of the time

References

References I

MLB-Advanced-Media. Stolen-base percentage. URL https://www.mlb.com/glossary/standard-stats/stolen-base-percentage. Accessed January 29, 2023.

Christopher Keyes. Talking baseball: When should you steal second base?, 2022. URL https://www.math.emory.edu/~ckeyes3/blog_stealing_bases.html. Accessed January 29, 2023.

Baseball-Savant. Statcast custom leaderboard. URL https://baseballsavant.mlb.com/leaderboard/custom. Accessed October 25, 2022.

Ben Baumer. Using simulation to estimate the impact of baserunning ability in baseball. *Journal of Quantitative Analysis in Sports*, 5, 2009. doi: https://doi.org/10.2202/1559-0410.1174.

Bruce Bukiet, Elliotte Rusty Harold, and José Luis Palacios. A markov chain approach to baseball. *Operations Research*, 45(1):14–23, 1997. URL http://www.jstor.org/stable/171922.

References II

Greg Stoll. Expected runs in an inning. URL https://gregstoll.com/~gregstoll/baseball/runsperinning.html. Accessed January 15, 2023.

Nobuyoshi Hirotsu and J. Eric Bickel. Using a markov decision process to model the value of the sacrifice bunt. *Journal of Quantitative Analysis in Sports*, 15 (4):327–344, 2019. doi: https://doi.org/10.1515/jqas-2017-0092.

Benjamin Morris. When to go for 2, for real, 2017. URL https://fivethirtyeight.com/features/when-to-go-for-2-for-real. Accessed January 28, 2023.

Charles Pavitt. An estimate of how hitting, pitching, fielding, and basestealing impact team winning percentages in baseball. *Journal of Quantitative Analysis in Sports*, 7(4), 2011. doi: https://doi-org.pallas2.tcl.sc.edu/10.2202/1559-0410.1368.

Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M. Ingersoll. An introduction to logistic regression analysis and reporting. *The Journal of Educational Research*, 96(1):3–14, 2002. URL https://www.jstor.org/stable/27542407.

References III

Retrosheet. Play-by-play data files (event files), 2022. URL https://www.retrosheet.org/game.htm. Accessed October 25, 2022.

Paul Schale. Mlb pitch data 2015-2018, 2020. URL https://www.kaggle.com/datasets/pschale/mlb-pitch-data-20152018. Accessed November 10, 2022.

Sports-Reference. Major league baseball catchers, 2017. URL https://www.baseball-reference.com/leagues/majors/2017-specialpos_c-fielding.shtml. Accessed November 17, 2022.

David Hitchcock. Nonparametric classification methods. Lecture, 2022. URL https://people.stat.sc.edu/hitchcock/stat530.html.