BreakoutPredict: A LASSO-Based Approach to Predicting Early-Season Baseball Offensive Performance Gains

Ruairi Moore,
Alex Hammerman,
Liam Ramsay,
& Chris Jung

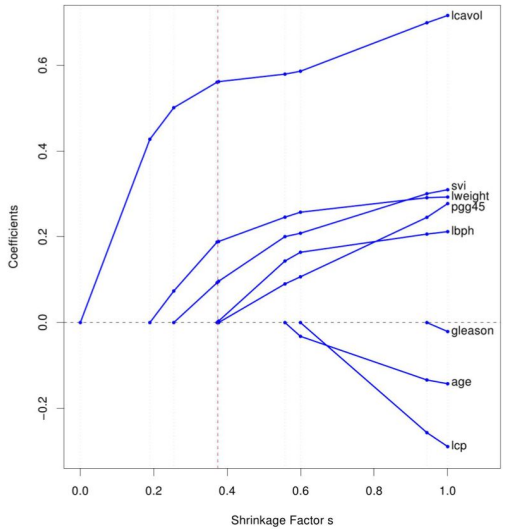
Introduction

- Every year new players take the league by storm
- Extremely difficult to predict who has made sustainable improvements
- Offseason information/spring training performance can be misleading
- How to interpret stats and know which indicate total offense?

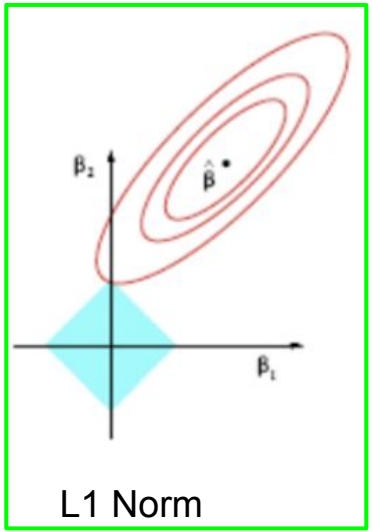
- Goal: To create a machine learning models that can use changes in offensive statistics between seasons to predict “breakouts”

Shooting LASSO Algorithm

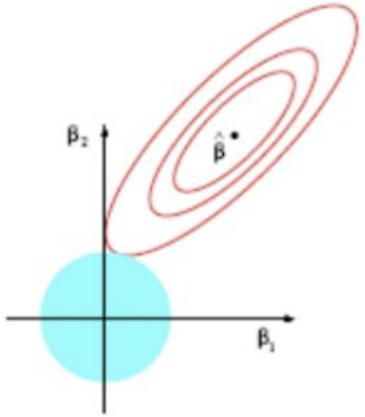
- Coordinate Descent
  - Optimizing one parameter at a time
- L1 Norm for regularizer
  - Relaxes exact sparsity condition



- Eliminates ineffective parameters
  - Intelligent feature selection
- Statistic-heavy problem
  - Compare performance of feature-heavy data to narrower feature set



L1 Norm



L2 Norm

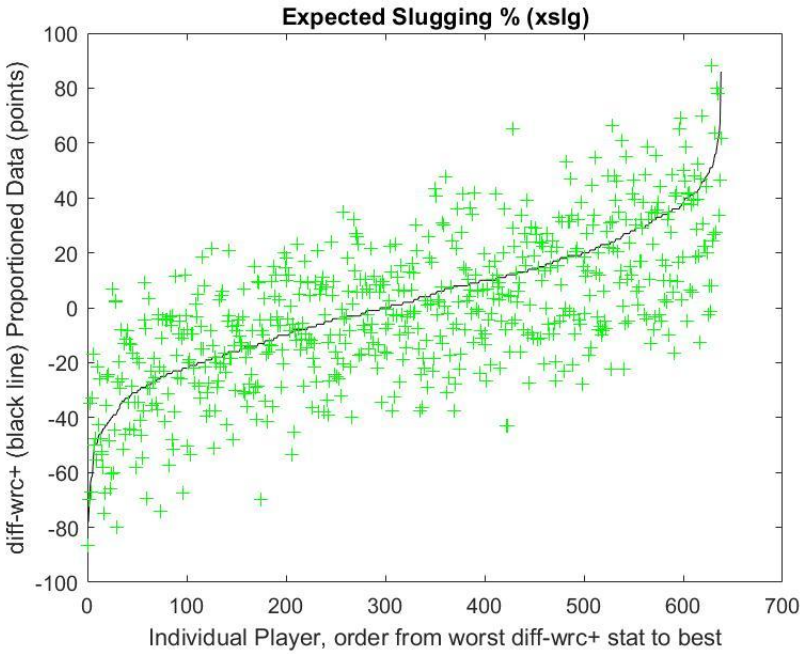
Feature and Label Selection

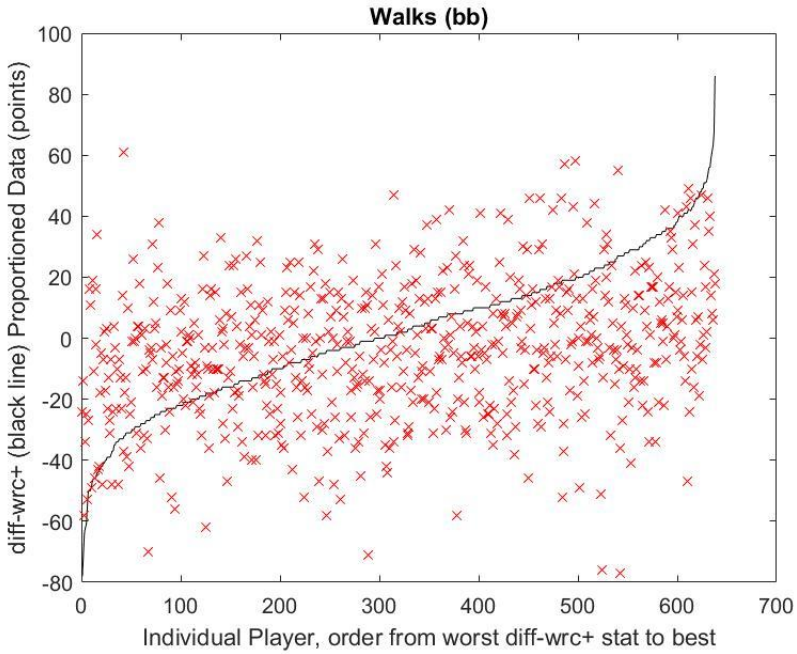
- Two LASSO Models trained
  - Standard Breakout Model
  - Rapid Breakout Model
- wRC+ (weighted runs created +) used for labels
  - Average player = 100; each point above/below = percentage point above/below average
  - Normalizes for factors such as ballpark, league, and era
- Data Source
  - Fangraphs

Model Training and Data Pipeline

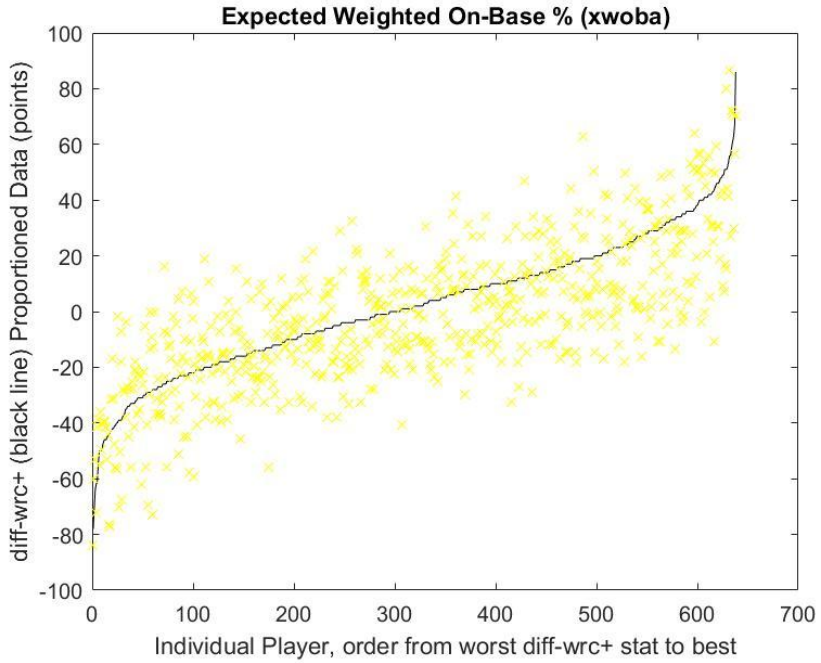
- Data Processing:
  - Extract batting data
  - Remove unwanted features (Player Name, Team Name, etc)
  - Extract wRC+ stats for ‘y’ outputs, leaving the rest for ‘X’ inputs.
- Resulting Datasets:
  - Xtrain, ytrain (includes 2015-2016, 2016-2017, 2017-2018 season stat differences)
    - Standard Xtrain (638x44), rapid Xtrain (638x10)
  - Xval, yval (includes 2018-2019 seasons, used for hyperparameter tuning)
    - Standard Xval (187x44), rapid Xval (187x10)
  - Xtest, test (includes 2019-2020 seasons and 2019-2021 seasons)
    - Standard Xtest (200x44), rapid Xval (200x10)
- Sample Constraints
  - Minimum of 300 PAs for 2015-2019, 120 PAs for 2020
  - Players must have reached thresholds for both adjacent seasons

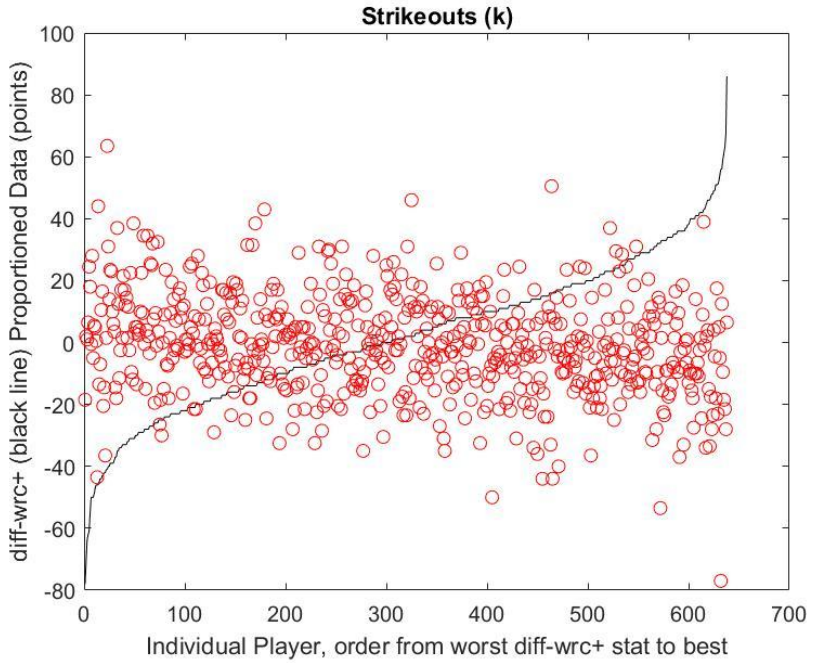
Visualization of Parameters





Visualization of Parameters (Cont.)



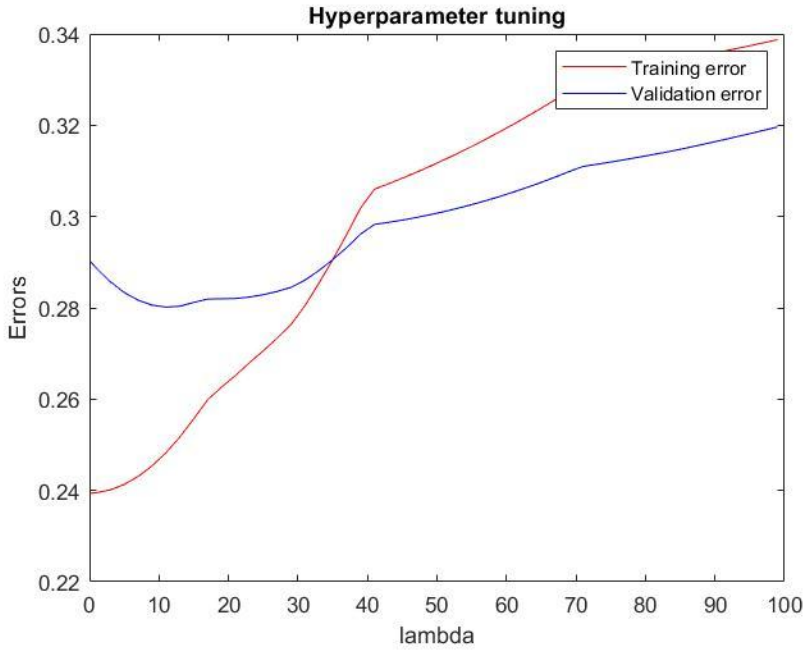


Validation: 2019 Season

- 2018-2019 season used for validation (last adjacent full seasons)
- Ran models on data with lambdas from .01 to 100
- Selected lambda based on smallest validation loss for each

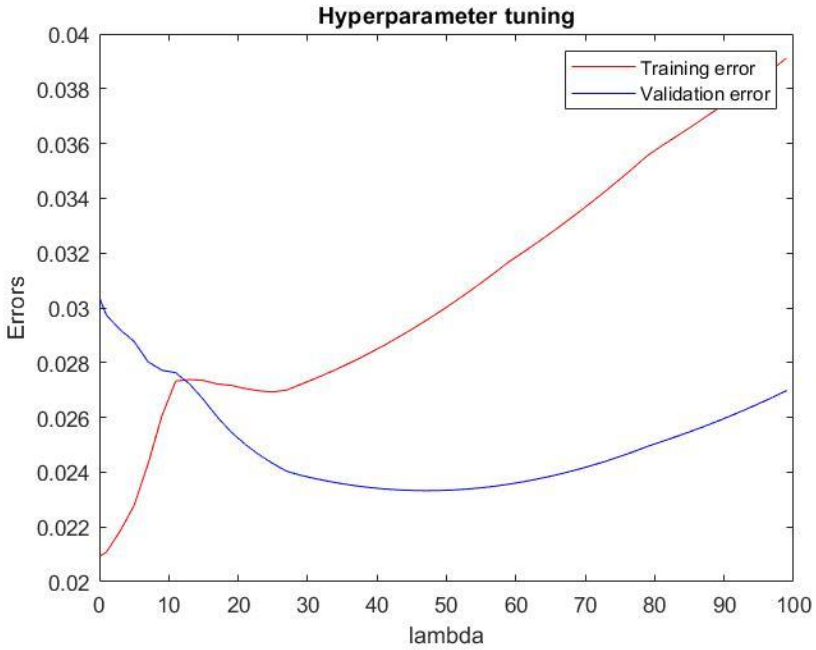
Validation/training Errors

Rapid Model



Optimal lambda = 10

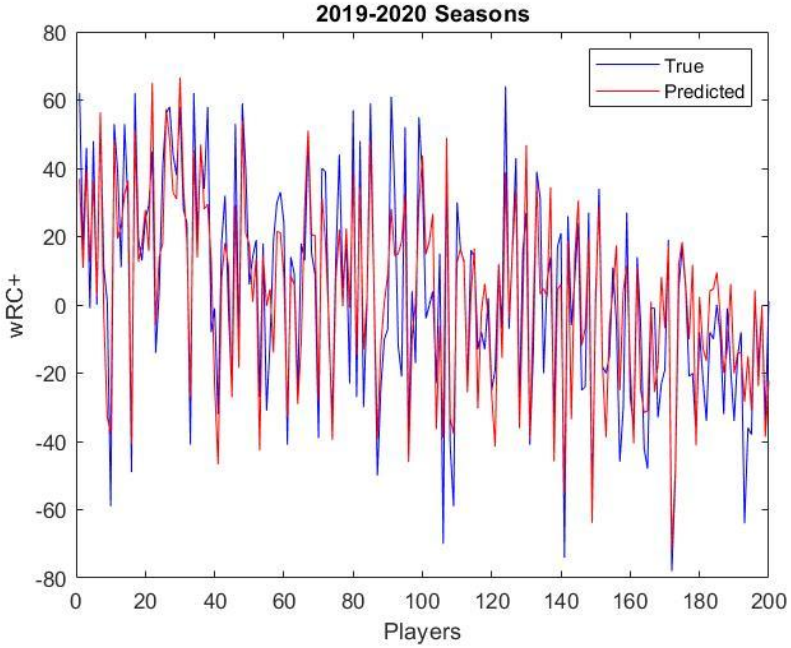
Standard Data



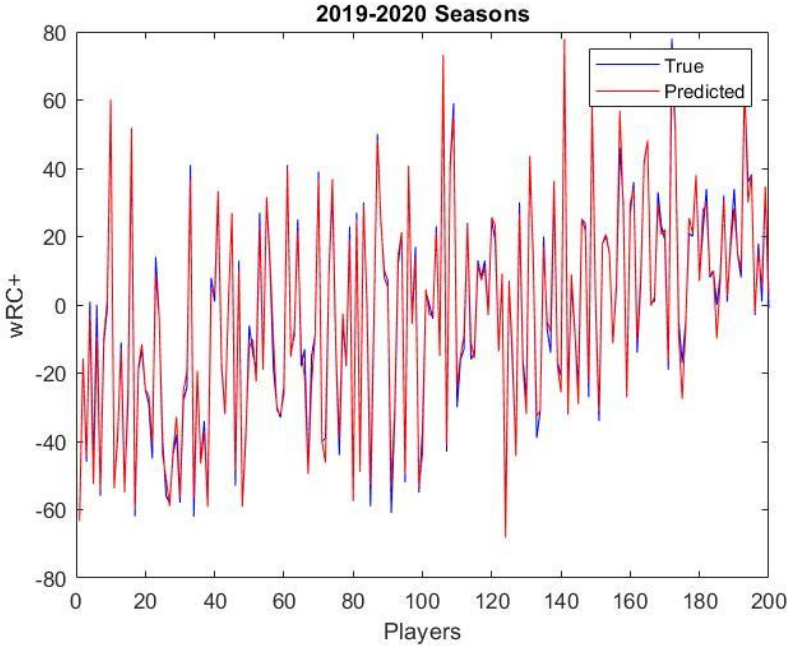
Optimal lambda = 50

Test 1: wRC+ 2019-2020 Plot

Rapid Model
Test Error = 15.069



Standard Model
Test Error = 3.395

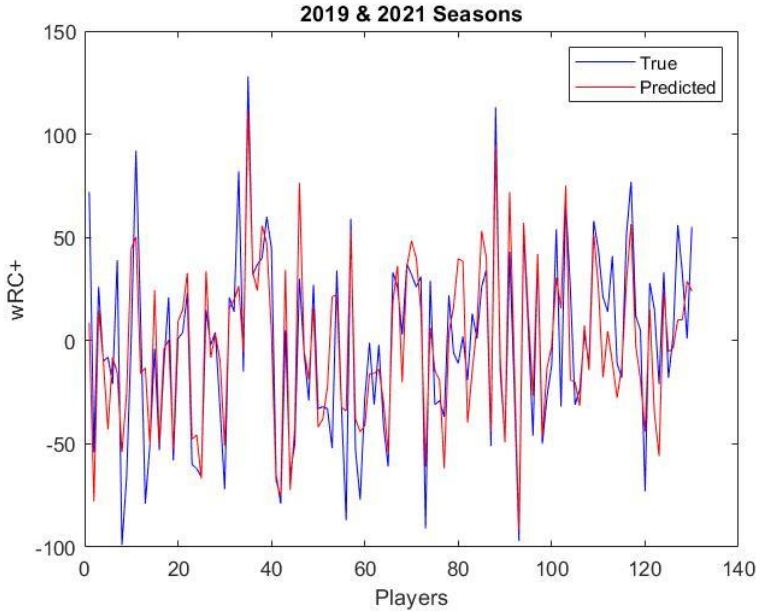


Test 1: wRC+ 2019-2020 Data

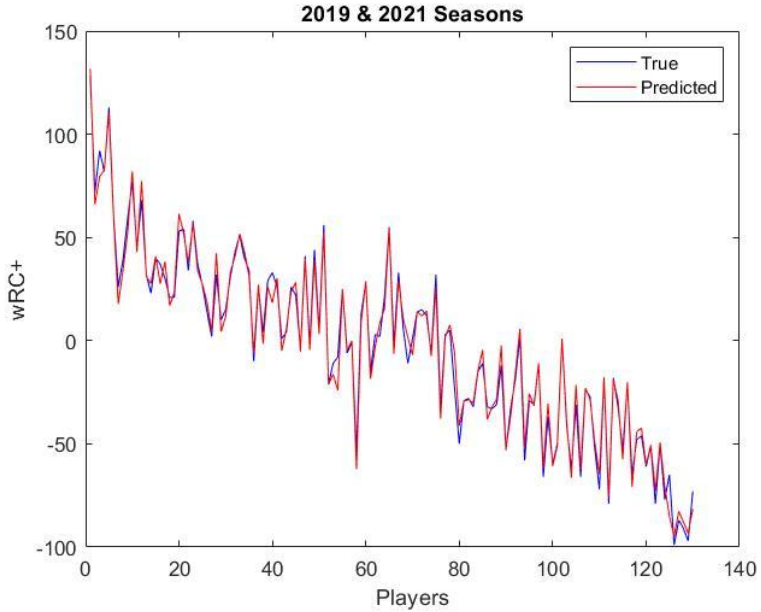
Player	Standard Model Predicted ΔwRC+	Rapid Model Predicted ΔwRC+	Actual ΔwRC+
Brandon Belt	77.9	55.1	74
Marcell Ozuna	73.2	38.9	70
Jose Iglesias	67.0	72.1	78
J.D. Martinez	-59.9	-51.3	-62
Christian Yelich	-63.4	-36.9	-62
Scott Kingery	-68	-38.8	-64

Test 2: wRC+ 2019 & 2021 Plot

Rapid Model
Test Error = 24.558



Standard Model
Test Error = 5.468



Test 2: wRC+ 2019 & 2021 Table

Player	Standard Model Predicted ΔwRC+	Rapid Model Predicted ΔwRC+
Ronald Acuna Jr.	131.7	111.2
Vladimir Guerrero Jr.	110.5	94.6
Justin Turner	82.7	26.4
Kevin Newman	-87.8	-61.0
Jonathan Schoop	-93.3	-92.4
Keston Hiura	-94.8	-54.0

Conclusion and Improvements

- LASSO is better at parameter selection/optimization than us!
  - The more features the better
- More features = more regularization
- Would’ve been nice to source more features cross-platform
- Reframe project data for intended goal
  - Train on early/partial season data, test with wRC+ for that season
- Rapid model still has value
  - More tame predictions on 2021 test data