MLB Ranking & Prediction Documentation

Release 0.0.1

Josh Rogan

June 08, 2016

Contents

1 Project Overview **3**

1.1 Version . . . 3
1.2 Repository . . . 3
1.3 Author . . . 3
1.4 Preface . . . 3
1.5 Goals . . . 3
1.6 Quick Links . . . 4
1.7 Assumptions and Constants . . . 4
1.8 Development . . . 4
1.9 Navigation . . . 4

Welcome to the MLB Ranking Documentation!

CHAPTER 1

Project Overview

1.1 Version

0.0.1

1.2 Repository

Github

1.3 Author

Josh Rogan (JoshJRogan@gmail.com) - Developer Site

1.4 Preface

Having played baseball for such a long time this project is very interesting to me on many levels. Much of the baseball specific analysis will be based upon is The Hidden Game by John Thorn and Pete Palmer. However, a significant portion of the project will be based on principles of data science not specific to baseball including data mining (with scheduling), association rules, recommender systems utilizing **Jaccard Similarity**.

1.5 Goals

A application to predict which MLB teams will be contenders in the range of 3-5 years. Also suggest what a particular team can do to make their team a contender in 3-5 years. Focus which stage a team is in (buying, selling, rebuilding, etc.) and how aggressive they are in that mode. Determine value of players focused on WAR and years of control as primary factors.

This will also be an exploration of ES2015 / ES6, Map and Reduce, and draw many ideas and algorithms from data science.

1.6 Quick Links

Links to some of the important sections.

- *Algorithms* - Algorithms for similarity comparisons to be used
- *Player Model* - Shows which stats will be focused on and why
- *Simulator Application* - High level synopsis of how the simulator will work

1.7 Assumptions and Constants

Work in progress

1.7.1 Assumptions

Decide on source for WAR - Baseball Reference, FanGraphs, ESPN, etc

1.7.2 Constants

Upcoming

1.8 Development

- **Node.js** - event-driven I/O server-side JavaScript environment based on V8
- **ES6** - NoSQL database system which stores data similar to JSON documents
- **Mongo DB** - standardized single modern database management system
- **Nginx** - high performance HTTP server
- **Digital Ocean** - simple cloud infrastructure for hosting
- **Front End Framework** - TBD
- **Statistical Analysis Framework** - TBD
- **Read the Docs** - Incrementation hosting with Sphinx generator

1.9 Navigation

1.9.1 Tasks

Simple todo lists to manage overall tasks.

Gathering Data

Phase 1: Current 40 Man Rosters

- [DONE] Download appropriate pages for currently players
- [IN-PROGRESS] Move downloads to AWS
- [DONE] Parse downloads to create JSON files
- [ ] Implement MongoDB or other DBMS to optimize performance
- [ ] Create weights for stats of each year to allow comparisons across years. (Simple idea - store season averages for each stat)

Phase 2: Retired Players History

- [ ] Download pages for past players to AWS
- [ ] Convert to JSON
- [ ] Find ways to group players based on rule changes and other weighting mechanisms

Phase 3: Prospects

- [ ] Download snapshot lists
- [ ] Download moving lists
- [ ] Analyze scouting ranks

Creating API

None Yet

1.9.2 Project Overview

Version

0.0.1

Repository

Github

Author

Josh Rogan (JoshJRogan@gmail.com) - Developer Site

Preface

Having played baseball for such a long time this project is very interesting to me on many levels. Much of the baseball specific analysis will be based upon is The Hidden Game by John Thorn and Pete Palmer. However, a significant portion of the project will be based on principles of data science not specific to baseball including data mining (with scheduling), association rules, recommender systems utilizing **Jaccard Similarity**.

Goals

A application to predict which MLB teams will be contenders in the range of 3-5 years. Also suggest what a particular team can do to make their team a contender in 3-5 years. Focus which stage a team is in (buying, selling, rebuilding, etc.) and how aggressive they are in that mode. Determine value of players focused on WAR and years of control as primary factors.

This will also be an exploration of ES2015 / ES6, Map and Reduce, and draw many ideas and algorithms from data science.

Quick Links

Links to some of the important sections.

- *Algorithms* - Algorithms for similarity comparisons to be used
- *Player Model* - Shows which stats will be focused on and why
- *Simulator Application* - High level synopsis of how the simulator will work

Assumptions and Constants

Work in progress

Assumptions

Decide on source for WAR - Baseball Reference, FanGraphs, ESPN, etc

Constants

Upcoming

Development

- **Node.js** - event-driven I/O server-side JavaScript environment based on V8
- **ES6** - NoSQL database system which stores data similar to JSON documents
- **Mongo DB** - standardized single modern database management system
- **Nginx** - high performance HTTP server
- **Digital Ocean** - simple cloud infrastructure for hosting
- **Front End Framework** - TBD
- **Statistical Analysis Framework** - TBD

- **Read the Docs** - Incrementation hosting with Sphinx generator

1.9.3 Algorithms

The data science algorithms that will be used to determine similarities among players will be determined based on how accurate the results are during the learning phase.

Basis of the prediction simulator is to determine similarities between players and teams to provide a better model to predict player and therefore team performance. Requires a **learning phase** to be successful and determine accuracy. This should be easy for most of the common stats as there is a massive amount of data dating back many years.

Determining Accuracy

To determine the accuracy of the predictions the models have we allocate all previous data for the system to be trained on and compare the predicted values with the known values.

Algorithms & Models

Many of the algorithms that employed will be that of **collaborative filtering** which often are used in recommender systems to give recommendations of products, movies, shows etc. However, in this case the recommendation will either be players that are similar. Biographical information will be used along with stats to determine how similar to players are.

Variables

Table 1.1: Variables Table

Name	Description
n	number of unique stats we are analyzing
p	number of players
v	number of values for the stats (includes stats over years)
u	undermined stats for a given player
Pt	prediction time of all players and all of their stats
Lt	learning time used by the algorithm to build a dataset in order to determine predictions

Similarity Measures

Jaccard Similarity A statistic used for comparing the similarity and diversity of sample sets of individual player stats.

Pearson Correlation Coefficient Measures how well two stats fit on a straight line

Adjusted Cosine Similarity Treat stats for each player as vectors in n-dimensional space (n = number of players) and determine the angle between the two vectors. **Important Adjustment** - weight all values with the average of each stat for that particular year as year to year factors change.

Algorithms

The table belows some of the algorithms that will be used to create a hybrid between memory and model based collaborative filtering.

Table 1.2: Algorithms Table

Name	Description	Performance	Use Case
K Nearest Neighbors	Training phase stores only feature vectors. Classification phase assigns labels which is most frequent among the k training samples nearest to the query point.	TBD	Very simplistic uses lazy learning
Slope One	Item-based collaborative filtering.	Lt = pn 2, Pt = (n-x)	Simple to use

Potential Problems

This entire approach of collaborative filtering might produce poor predictions.

Stats

- Look up the Curse of dimensionality when choosing which stats and what **distance algorithm** to use. *Most often not euclidean.*

1.9.4 Getting Started

Installation and requirements pages.

Requirements

These requirements are targed at linux (debian) users specifically). However, their own install documents should be able to help.

Babel & Application Requirements

- Node & NPM

Sphinx & Read The Docs Building

- Python
- PIP - Install Docs
- Virtualenv - Install Docs

Installation

Note: Check the *requirements* before proceeding!

Initial Setup

1. `npm install` - will install everything necessary to run
2. Download the files either by reparsing or pulling down from AWS. *More coming soon*
3. `npm run-script build && node lib/<script to run>` - Transpile ES6 with babel and allow to run anywhere

More coming soon

Babel and JavaScript

- Babel Easy - Plugin

More coming soon

Sphinx & Read the Docs (Optional)

Note: Simply editing the `docs/*.rs` files will automatically regenerate via a webhook with readthedocs.com when you push. Local generations are useful for testing the docs without having to commit and see them on readthedocs.com

1.9.5 Database Information

Information for building the database and the associated data models that will be used.

Data Sources

List of some of the sources that can be used to build the database.

Player Sources

All of the useful sources for players.

Player Stats

Table 1.3: Stats

header "Name"	URL	Notes	Use Case	Use Grade
Baseball Reference	baseball-reference.com	Detailed stats about players, teams, and much more.	Stats, salaries, bios, standings	9/10

Prospect Sources

All of the useful sources for prospects.

General Information

Table 1.4: General Information

Name	URL	Notes	Use Case	Use Grade
MLB Prospect Pipeline	mlb.com	Detailed information about propspects	Informative	9/10

Lists & Rankings

Table 1.5: Lists/Rankings

Name	Years	URL	Notes	Use Case	Use Gra
MLB Top 100	2011-2015	mlb.com	Moving list of top 100 prospects.	Ranking Lists, Scouting Grades	7/10
MLB Top 30 by Team	2011-2015	mlb.com	Moving list of teams top 30 prospects.	Ranking Lists, Scouting Grades	5/10
MLB Draft Lists	2011 - 2015	mlb.com	Ordered list of the draft.	Ranking Lists, Draft position to future outcome	8/10
MLB Top International	2011 - 2015	mlb.com	Moving list of top 30 international prospects.	Ranking Lists, Scouting Grades, International players rating to future outcome	TBD
Prospectus Top 101	2007 - 2016	prospectus.com	Snapshot List of top 101 prospects.	Ranking Lists	8/10
ESPN Prospect Lists	TBD - 2015	espn.com	ESPN doesn't have a centralized system for ranking. Will be more difficult to analyze	Ranking Lists	5/10

...

Player Model

Stats

General idea is to stick with runs/wins as the primary unit of measurement. All stats shouldn't be counters, they should be averaged and weighted. Counters should only be used to permit the stats as statistically relevant. Common stats such as batting average, era, and more traditional stats should all be weighted in the current season. However, stats that don't have as much as a history might not be possible to be accurately weighted per season.

The simulator will allow the user to weight other stats differently but the goal will be to have defaults with the best accuracy.

** Important: Different resources calculate some stats differently. Most notably WAR. **

Resources

- Fangraphs Counting vs Rate
- Normalization

Emphasis These are the stats my initial thoughts are to focus on and weigh the most heavily.

Definitions

Counting Stat A raw “tallying” number (ex. number of hits)

Rate Stat A value divided by other value (ex. hits per at bat)

Traditional w/weight A commonly used baseball stat but weighted across years for more accuracy.

Overall Value A catch all stat that encapsulates as many factors as possible into one number.

Overall **Qualifiers** - Innings played

Table 1.6: Overall Stats

Stat	Description	Stat Type	Justification	Notes
WAR	All inclusive Wins above replacement	Overall Value	Cross comparable stat	N/A

Offensive **Qualifiers** - Plate Appearances

Table 1.7: Offensive Stats

Stat	Description	Stat Type	Justification	Notes
RC/27	Runs created per 27 outs	Rate Stat	Accuracy	N/A
AVG	Batting average normalized across years.	Traditional w/weight	Popularity	N/A
wOBA	Credit a hitter for the value of each outcome	Overall Value	Sabermetrics Popularity & Accuracy	N/A

Pitching **Qualifiers** - Innings Pitched (differentiate starters and relievers)

Table 1.8: Pitching Stats

Stat	Description	Stat Type	Justification	Notes
ERA	Earned runs per 9 innings normalized across years	Traditional w/weight	Popularity	N/A
dERA	Projects what a pitcher’s earned run average (ERA) would have been, if not for the effects of defense and luck on the actual games in which he pitched.	Overall Value	Sabermetrics Popularity	Notes
BABIP	How often non-home run batted balls fall for hits	Rate Stat	Sabermetrics Popularity & Accuracy	N/A

Defensive **Qualifiers** - Innings played

Table 1.9: Defensive Stats

Stat	Description	Stat Type	Justification	Notes
dWAR	Defensive wins above replacement	Overall Value	Sabermetrics Popularity & Accuracy	N/A

Non Statistical Based Measurements

Analyzing prospects (domestic & international) will be one of the most difficult parts to build on the database. It will based on a lot of different baseball analysts top prospect lists.

Ideas

- Position on top ranking lists
- Scouting power rankings (difficult to analyze many different sites have different definitions for this)
- High school stats **difficult to acquire**
- Analyze bio, height, etc when predicting.

Additional Ideas

- Mainting the movement a player undergoes on top lists

1.9.6 Simulator Application

Application that utilizes the collected data and computes predictions based on a variety of algorithms and formulas.

Team Simulator

The simulator allows the user to use different stats to predict which teams will be successful in certain year ranges.

Simulation Options

- Team Status (Conservative, Win Now, Rebuild)
- Factors of importance on stats (suggest values by default)
- Use the player simulator to predict how players will progress

Player Simulator

The player simulator will use big data philosophies such as the jaccard similarity to find patterns among other players to help predict the future of a player.

Simulation Options

- Look at age, height, weight, and position as base factors
- Determine which stats best predict a players future. Similar to how MIT noted the correleation of run differential to wins

1.9.7 Frequently Asked Questions

Working Draft

1.9.8 Results & Analysis

Page used to discuss the outcome of the predictions and simulators.

First Run Results

The first similarity runs on the data were very simple. Using N-Dimensional euclidean distance analyzing the stats:

Table 1.10: Stats Analyzed

Stat	Abbrev	Qualifier
Plate Appearances	pa	> 502
Hits	h	N/A
Homeruns	hr	N/A
Runs Batted In	rbi	N/A
Batting Average	ba	N/A

```
"meta": {
    "year": 2015,
    "stats": [
      "pa",
      "h",
      "hr",
      "rbi",
      "ba"
    ],
    "date": "2016-02-14T07:47:40.385Z",
    "distanceAlgorithm": "euclidean",
    "similairtyAlgorith": "1/(1 + distance)",
    "qualifier": "plate appearances > 502",
    "pairs": 5565
},
```

Most Similar Players

Table 1.11: Most Similar Players

Player	Player	Distance	Similarity
Avisail García	Brett Lawrie	5.1961532887319635	0.16139045523914872
Christian Walker	Nick Castellanos	5.291503000093641	0.15894453201168565
Asdrubal Cabrera	Cameron Maybin	6.557438829299134	0.13231995952426898
Jean Segura	Andrelton Simmons	6.708208702776025	0.12973182727134272
Yunel Escobar	Cristhian Adames	7.416211701401195	0.11881830394469668
Matt Kemp	Jesús Aguilar	7.549834700177216	0.11696132557735561
Stefen Romero	Corey Seager	7.615775469379333	0.11606616299995552
Joe Mauer	Elvis Andrus	7.681148937496265	0.1151921257427949
Charlie Blackmon	Adam Eaton	7.937253933193772	0.11189119247086728
Cameron Maybin	Derek Norris	8.062275671297776	0.11034755907583128
Asdrubal Cabrera	Derek Norris	8.831773604435295	0.101711048304538
Alexei Ramírez	Kolten Wong	9.165160609612904	0.09837522872529218
Darnell Sweeney	Rob Refsnyder	9.219544457292887	0.09785171972967724
David Peralta	Alex Dickerson	9.327380393229387	0.09682997642418578
Billy Butler	Avisail García	9.433983040052594	0.0958406771566843
Francisco Cervelli	Leury García	9.486836353600708	0.09535764326642181
Francisco Cervelli	Daniel Castro	9.94988889385203	0.09132512756010375
Chris Coghlan	Pablo Sandoval	10.049876864917302	0.0904987460244865
Yangervis Solarte	Starlin Castro	10.148892796753742	0.08969500543508438
Yangervis Solarte	Keon Broxton	10.246951205114621	0.08891298466247846
Neil Walker	Nick Castellanos	10.24696032977585	0.08891291252735571
Jean Segura	Didi Gregorius	10.295633249101291	0.08852978650661861
Kyle Seager	Kole Calhoun	10.295634997415167	0.08852977280417035
Brandon Belt	Dariel Álvarez	10.44030693993237	0.08741024215963139
Freddy Galvis	Marcus Semien	10.44030823299772	0.08741023227990148

Conclusions