**Predicting MLB 2017 Regular Season Games using Social Network Analysis**

**By: Matthew Miller**

**BIA 658A - Prof. Gomez**

# Introduction:

Major League Baseball is known as the first of the four major sports leagues in the US to kick off the sports analytics movement. In general, the sport of baseball lends itself well to statistical analyses. The events within each game are discrete and individual, making it easier to distill individual events into numerical values. For as long as baseball’s existed, the box score, a numerical representation of the game has accompanied it. The Society of American Baseball Research was formed in 1971, which led to the creation of Sabermetrics, a term used for empirical analysis of baseball statistics. Most famously, Billy Beane utilized Sabermetrics as General Manager of the Oakland A’s to identify underutilized assets. Beane shifted the focus of MLB front offices from traditional wisdom to quantitative approaches to team building. Seeing how baseball is so well suited to these advanced statistical analyses, I decided that it would also lend itself well to the type of analysis studied in this class.

For this project, I decided to model the 2017 MLB Regular Season as a directed, weighted network based on run differential. Run differential is measured by the difference in scores at the end of each game; i.e. a game ending Mets 2, Pirates 1 would yield a +1 run differential for the Mets, and a -1 run differential for the Pirates. Theoretically, this same model could be used for different metrics such as Win/Loss ratio or advanced stats like WAR. I decided to use Run Differential over Win/Loss ratio because run differential can show not just whether a team won or lost, but also the degree to which they won or loss. One of the major conclusions of early sabermetrics research was that run differential could be used to better understand a team’s true abilities; i.e. teams that lose many close games are likely better than their actual record would show. The network would represent the overall average run differential per game between teams.

# Objective:

The objective of this project was to attempt to first model the 2017 MLB Regular Season as a network, and then to use various statistics and methodologies associated with social network analysis to analyze the network and draw conclusions on how future games may play out. There were three methods used to attempt to predict future games. One method was based on solely the previous matchups between two teams. The next method was based on the teams’ overall performance over the course of the season. The final method was based on looking at the performance based on similar matchups. These methods were then formed into statistical formulas discussed later.

# Data and Transformation:

Originally, just the list of all games played this season up to May 31st was taken from Baseball Reference and brought into Excel. Then a new sheet was made with two entries for each game, listing the team, the opponent, and the run differential. Once these entries were created, I also used Excel to get the total run differential for each Team/Opponent combination and divided that by the total number of games played between the two teams. This is the Average Run Differential Per Game, and became what I used for the edge weights in the network. Here is an example of how the calculations were done:

Table 1 - Initial List of Games:

|  |  |  |  |
| --- | --- | --- | --- |
| Home Team | Home Team Score | Away Team | Away Team Score |
| Chicago White Sox | 1 | Arizona D'Backs | 5 |
| Chicago White Sox | 4 | Arizona D'Backs | 5 |
| Chicago White Sox | 6 | Arizona D'Backs | 8 |

Table 2 - Entries Showing Run Differential:

|  |  |  |
| --- | --- | --- |
| **Team** | **Opponent** | **Run Difference** |
| Chicago White Sox | Arizona D'Backs | -4 |
| Chicago White Sox | Arizona D'Backs | -1 |
| Chicago White Sox | Arizona D'Backs | -2 |
| Arizona D'Backs | Chicago White Sox | 4 |
| Arizona D'Backs | Chicago White Sox | 1 |
| Arizona D'Backs | Chicago White Sox | 2 |

Table 3 - Edge Weights:

|  |  |  |
| --- | --- | --- |
| Source (Team) | Target (Opponent) | Weight(Avg. Run Diff.) |
| Chicago White Sox | Arizona D’Backs | -2.333 |
| Arizona D’Backs | Chicago White Sox | 2.333 |

As seen above, for every edge that exists, there is an edge going in the opposite direction with the negative weight. The interpretation here is that the Arizona D’Backs outscored the Chicago White Sox by an average of 2.33 runs per game. One question raised at this step was whether to use the average run differential or total run differential. Average run differential did have the issue of treating matchups where teams have only played each other 3 times the same as matchups where teams have played each other 12 times, but this was addressed later on by using weighted averages based on number of games played.

# The Network:

Once the edge list was created, I brought the list into R Studio to perform the predictive analyses, and into Gephi for the data visualizations and getting the statistics for the overall network. Using Gephi, I found that the visualizations of the actual network were not very descriptive because each edge had an edge in the opposite direction with a negative weight. For a better visualization, I filtered out the negative edges. Here is the network in Gephi with the negative edges filtered out:



Figure 1: Network Visualization in Gephi with Negative weight edges not shown

Generally, better performing teams would have high out-degrees. Out-degree would represent the sum of the different run differentials for each team. In the above visualization, the team names were sized based on the out-degree. As a bit of a sanity check, teams such as the Colorado Rockies, Washington Nationals, and Houston Astros have larger names, and they all were near the top of the standings as of May 31st. To zoom in on the graph a bit, I also took the Egocentric network for the Arizona D’Backs seen here:

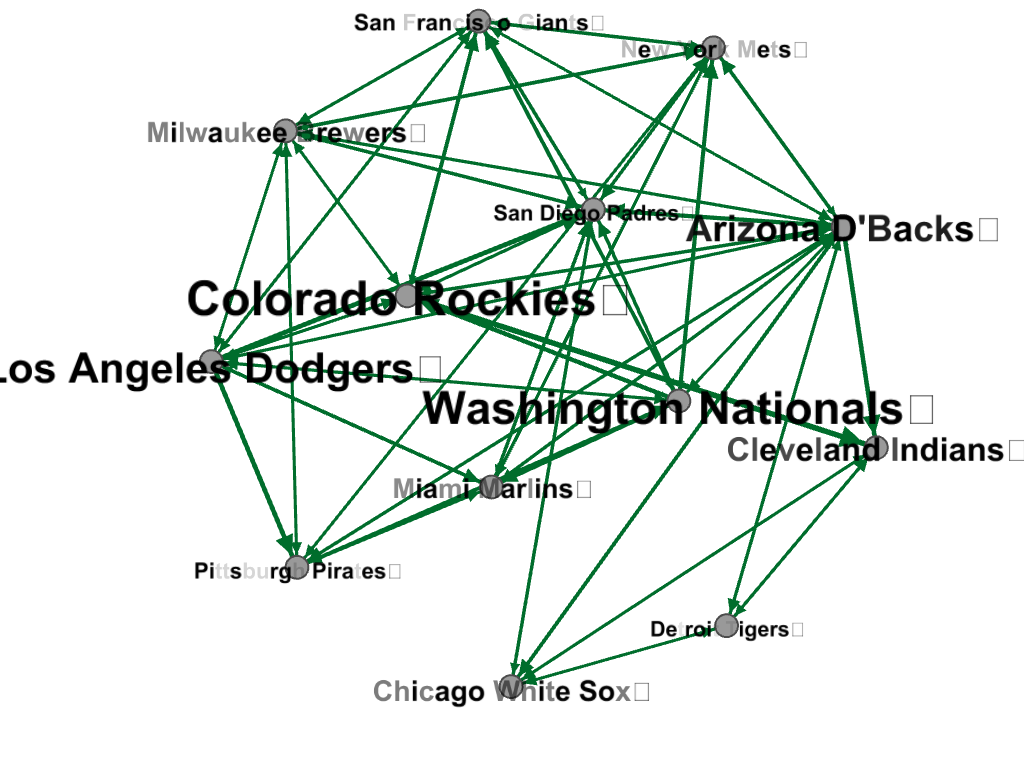


Figure 2: Egocentric Network for the Arizona D’Backs

This network still has the negative edges not pictured, but it does show more of the interactions between teams. A team like the Colorado Rockies has many edges directed out from them towards other teams, meaning that there are many teams that they have a positive run differential against. A team like the San Diego Padres, however, has many edges directed towards it meaning that they have a negative run differential against many teams.

The network contains 30 nodes representing each of the 30 teams, and 398 edges representing 199 unique matchups that have occurred this season up to May 31st, accounting for each matchup generating 2 edges. The diameter of the graph was 5, and the average path length was 2.039. A path length of 2 would mean that for two nodes, there is at least one team that they both played. Since the average path length was very close to 2, I felt confident moving into the analysis of common neighbors.

# Analysis:

After creating the network, I wanted to attempt to use the data here to attempt to predict the results of future games. The games from June 1 to June 21 were used as the data to be predicted. For these games, I tracked the run differential of each game, as well as the win/loss outcome. As stated before, the three methods for predicting future head to head matchups were based on previous games between the two teams, the overall performance of each of the two teams, and the performance of each team against teams that they both played. These were then transformed into metrics that utilized the social network format. For Method 1 regarding previous matchups, the predicted value was the existing edge between two teams. For Method 2 regarding overall performance, the predicted value was the difference in out-degrees of the two teams. For Method 3 regarding performance against similar teams, the predicted value was the average out degree to teams with similar neighbors. Where applicable, average actually means a weighted average based on total games played between the two teams. I used R to create a new matrix containing all the games from June 1 to June 21, the actual run differential of each game, and the predicted run differential of a game between each pair of games using the three different methods.

Once I found the predicted value for each game, I then ran three separate univariate linear regressions using SAS comparing the predicted run differential of each method to the actual run differentials. Here were the resulting fit plots according to SAS:

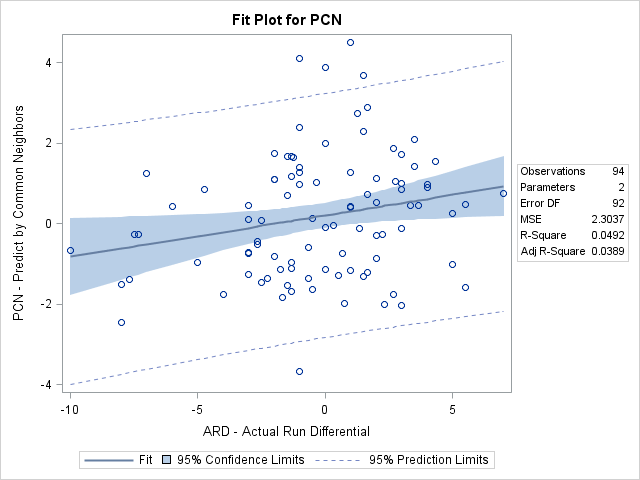


Figure 3: Fit Plot Using Method 1-Head to Head Matchups

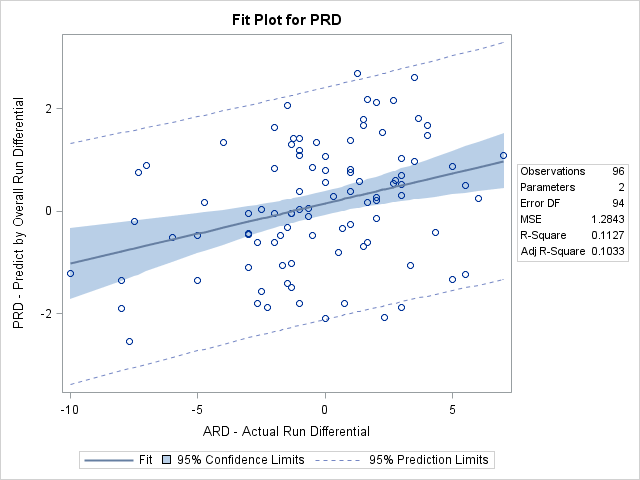


Figure 4: Fit Plot Using Method 2 – Overall Run Differential

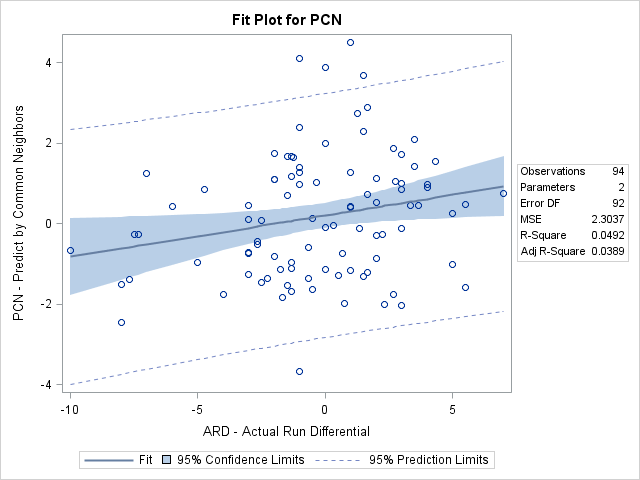


Figure 3: Fit Plot using Method 3 – Run Differential against Common Matchups

And here are the relevant statistics from the tests:

|  |  |  |
| --- | --- | --- |
| Method | R-Square | P-Value |
| 1 | .0082 | .3796 |
| 2 | .0492 | .0316 |
| 3 | .1127 | .0008 |

Table 4: Results from Linear Regression Tests

From this, it seems that Method 3 has the greatest correlation of predicted run differential to actual run differential, Method 2 has a smaller but still statistically significant correlation, and Method 1 does not have a statistically significant correlation.

However, when it came to actual predictions of game outcomes, these were the results:

|  |  |  |
| --- | --- | --- |
| Method | Games Predicted | Success Rate |
| 1 | 88/146 | 60% |
| 2 | 126/255 | 49% |
| 3 | 131/258 | 51% |

Table 5: Results based on predicted outcomes

Whichever team predicted a higher run differential was the team that was designated as the predicted winner. From this chart, it would seem that method 3 had the highest success rate, while method 2 actually performed worse than just picking randomly. However, Method 1 did not predict any games where there was no previous matchup, which may have been harder to predict, thus artificially inflating its success rate.

# Conclusions and Takeaways:

Ultimately, it would seem that Method 2 of measuring the overall average run differential of teams and comparing them would be the best method to predict the outcome of games, though Method 3 also has predictive ability when trying to predict the run differential of games. It also seems that, when applicable, Method 1 can be a strong predictor of games when the teams have faced each other previously, but since there are many times when teams haven’t faced each other, it cannot be a predictive factor on its own.

Overall, there are definitely insights to be found using this social network framework to analyze a sports season. This project did reveal two methods that produced statistically significant (if small) correlations with actual run differentials of games. Admittedly, this did not generate great results when actually predicting the win/loss outcomes of games, so I would not advise going gambling with this newfound information. Still, any sport, especially baseball, will be hard to predict in this manner. Considering how long the baseball season is, even the worst teams in the league can still manage to win about 1/3 of their games just due to randomness. Even if the predictive record isn’t great, the predictive run differential definitely gives me more hope regarding the success of this project. I am not too scared of the small correlations found, given just how much randomness and luck is inherent to the sport. The starting pitcher of a team can have a disproportionate amount of effect on a team’s game to game performance. Teams may change as players are injured or traded. Even the best predictive model will not perform incredibly well when predicting baseball games just because of randomness. In addition, there will always be an issue of sample size when predicting baseball games. This project was performed about a third of the way into the season. While I could test the model on previous seasons to better figure out a way to weight the factors, there is really no solution to the amount of games available to be used for prediction.

One idea for an addition to this project would be to have the analysis constantly updating as more information is known, and looking at the results in more of a live fashion with the network constantly updating and making new predictions with even more information available. In the MLB, games are typically played in series of 3 games with one opponent consecutively. It would be interesting to see how performance within a series can predict later games in the series. I would also consider using some advanced statistics as the metric for the edge weights in future analyses using this framework. I currently do not have enough statistics readily available to better calculate some advanced statistics aggregated by opponent to do such analyses. Finally, I would be interested in trying to weight the games based on recency and see if that would improve the predictive power of these methods.