GameRank and MLB illustrator: Ranking, Visualizing and Analyzing Baseball Network

Zifei Shan
EECS, Peking University
shanzifei@pku.edu.cn

Shiyingxue Li
EECS, Peking University
rachellieinspace@gmail.com

ABSTRACT

In the report we present a novel algorithm, modified from Pagerank and HITS, to evaluate the baseball players in Major League Baseball (MLB) in a network perspective. The algorithm could also be easily expanded and applied on any network that has multiple factors interacting with each other, to evaluate the vertex's significance. Relevant analysis is also performed for our MLB data network for both teams and individual players from 1930s to now, with a few interesting conclusions drawn. Last but not least, we have warped up the whole system as a working website, called MLB Illustrator (http://zifeishan.github.com/mlb_illustrator), to let users interact with the data and network itself, making the traditional baseball statistics analysis based on tables and simple graphs evolve into intuitive visualized network analysis.

Keywords

Network visualization, Ranking, Major League Baseball, Data Mining

1. INTRODUCTION

Major League Baseball, or MLB, is the professional baseball league consisting of American League and National League. It has the most attendance of any sports league with more than 70,000,000 fans. Baseball being one of the most popular sports in the States, the statistics analysis of its games and professional players has always been of great interest.

Retrosheet.org [1] keeps an amazingly complete record of play-by-play game data from 1930s to 2010s in a structured form. Due to the complex set of rules of baseball game, it has a rather sophisticated way of keeping score and game situation, and much effort is needed when trying to find out the general performance of any team or player.

While PageRank[4] can be applied for ranking the teams based on the game results, valuable players can't be accurately tracked, because players have multiple abilities, such as ability to hit, to pitch, to run, and to catch balls. To better evaluate players, we've come up with a novel approach, inspired by PageRank and HITS[5], to iteratively measure the performance of the given individual. The algorithm assigns each individual GameRank (GR) values to represent his offensive/defensive ability, and use multiple random walk models to iteratively accumulate the GR value. The results of GameRank, along with other analysis results, will also be shown.

The rest of our report is arranged as such: Section 2 gives a more detailed description of GameRank, proves its validity, and compares GR with simple degree-centered approaches to demonstrate its significance. Section 3 shows the product, MLBI's features, e.g. ranking visualization, game replay, statistics display etc.. Section 4 presents the specific analysis of our MLB data set network, illustrates its structure, basic attributes and evolution over time, and shows a few interesting results. And finally in Section 5 we conclude and propose future works.

2. ALGORITHM: GAMERANK

2.1 Introduction and Motivation

Each player, in a baseball game, has two basic evaluation standard: (a) offensive ability, and (b) defensive ability. For simplicity, we define the offensive ability as a player's ability to bat and the defensive ability as to pitch. These two ability cannot be evaluated independently, in that we tend to think the player a better hitter if he can achieve a home run facing a greater pitcher, and a better pitcher vice versa. Simple PageRank is unable to capture such a network's feature.

Our assumptions in the baseball network are: (a) a player is good at batting if he wins over good pitchers; (b) a player is good at pitching if he wins over good batters.

To iteratively calculate each player's ability, a random walk model is applied to obtain the stationary distribution of the player's ability value, that is, the GameRank value.

2.2 Definition of GameRank

Definition 1. *An Batting Edge from A to B means A wins over B when A is offensive and B is defensive. Similarly, an Pitching Edge from A to B means A wins over B when A is defensive and B is offensive. N is the number of vertices, $DB_{in}(i)$ is the batting in-degree of vertex i, and $DD_{in}(i)$ is the pitching in-degree of it.*

Then Batting Ability is

$$RB(i) = \beta/N - (1-\beta)\sum_j RD(j)/DD_{in}(j),$$

Pitching Ability is

$$RD(i) = \beta/N - (1-\beta)\sum_j RB(j)/DB_{in}(j),$$

where β is the damping factor.

2.3 Random Walk Model

GameRank can be seen as two random walk models interacting with each other. Say, the Mighty O Drunk first play as the batter i, in the OFFENSIVE part of the models, and has a pre-determined probability $p(ij)$ to win over a pitcher j. As he loses to j, he takes the position of pitcher j, in the DEFENSIVE part, and win over a hitter k with probability $p(jk)$. As he continues this never-ending game, after a sufficiently long time, the probability that he is acting the batter (pitcher) i, called $\pi(i)$, represents i's batting (pitching) ability to win.

2.4 Computation

With N vertices in the network, we first assign $1/N$ as the initial GameRank, and makes sure GR values sum up to 1. Then iteratively, using the formula we have in Definition 1,

$$RB(i) = \beta/N - (1-\beta)\sum_j RD(j)/DD_{in}(j),$$

$$RD(i) = \beta/N - (1-\beta)\sum_j RB(j)/DB_{in}(j),$$

we collect the GR values of each vertex. The process is repeated until GR values converges to the stationary distribution.

2.5 Convergence Properties

It happens that some vertices have no out-links, as they haven't beat anyone in batting or pitching, in turn making the network not connected. And even worse, it will lead to non-convergence of the GR network, where these vertices serve as absorbing states. To deal with it, we choose to add the damping factor β to allow random victory: Mighty O Drunk, like in the comics, at the most desperate moment could always beat the enemy. With these miracle links, the network become connected without dangling nodes hanging around, and stationary distribution could be obtained.

2.6 Measurements

In order to prove that our GameRank algorithm has good performance, we conducts a series of measurements. In the measurements we pick the match data of all the teams in 2011, as it is the most recent and there are some official statistics in this year. And we analyze the top players according to the algorithm.

We list the top players selected by our algorithm in two tables with the official rankings from ESPN. The ESPN rankings are 2011 regular season, "Opp Batting Stats" for pitchers, and "ESPN Ratings" for batters.

The comparisons are shown in table 1 and table 2.

Table 1: Top-10 Batters

GR Rank	Name	Team	Official Rank
1	Prince Fielder	MIL	6
2	Matt Kemp	LAN	2
3	Justin Upton	ARI	17
4	Joey Votto	CIN	8
5	Troy Tulowitzki	COL	18
6	Hunter Pence	HOU	21
7	Albert Pujols	SLN	12
8	Ryan Braun	MIL	2
9	Carlos Beltran	SFN	33
10	Mike Stanton	FLO	Out of 50

Table 2: Top-10 Pitchers

GR Rank	Name	Team	Official Rank
1	Daniel Hudson	ARI	36
2	Matt Cain	SFN	12
3	Chris Carpenter	SLN	32
4	Roy Halladay	PHI	2
5	Tim Hudson	ATL	24
6	Clayton Kershaw	LAN	1
7	Ian Kennedy	ARI	11
8	Cliff Lee	PHI	3
9	Tim Lincecum	SFN	7
10	James Shields	TBA	9

According to the comparison, we find that GameRank algorithm is quite effective, since most of them are ranking at the top of the official ranks.

It turns out that GameRank is a simple, effective algorithm, which fits in the situation where there are multiple interplaying factors. And it also proves that our assumptions of baseball games are reasonable.

3. PRODUCT: MLB ILLUSTRATOR

We build an online system to visualize and rank all the MLB data from 1930 to 2011, including ranking teams, ranking players in one team, and ranking all the players in one year. Our ranking system uses the algorithm of GameRank.

The website of our system is: **mlbillustrator.com**; or **zifeishan.github.com/mlb_illustrator**.

Our website is built upon D3.js [2], jQuery [3], and basic javascript and html.

3.1 Ranking and visualization

The current system provides three ranking systems: (1) Player Rank by Team ranks the player using all the games played by a certain team during one year. (2) Player Rank by All Teams ranks the player using the data of all games of that year. (3) Team Rank uses the game data of the whole year to rank every team.

First two player rank systems have two metrics: players can be ranked by their GameRank value, or simply by their out-degree. Team rank uses the team's PageRank value, modified to adjust to the MLB network.

Figure 1 is the main interface of the system.

Figure 1: Main Window

For ranking player by team, the user can choose year, team and ranking metric. The network of your choice will be shown automatically in the central part of the window.

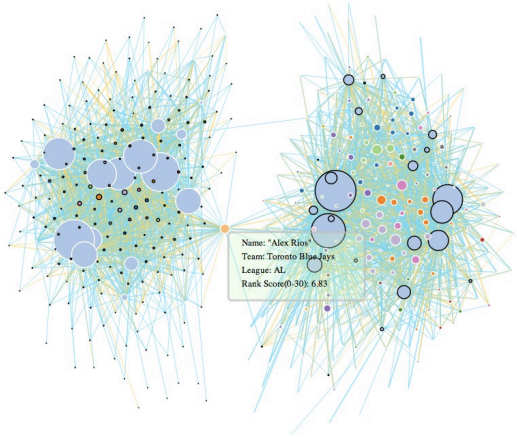

Figure 2: Ranking Player by Team, 2009, Chicago White Sox with GameRank

Take Chicago White Sox, in 2009, as an example in Fig 2. Mouse over any vertex or arc, the basic information (name, team, league etc.) of the vertex or arc (source, target, offensive or defensive) will be shown. Vertices with black borders are pitchers, white are batters. Teammates vertices will be in the same color. Yellow edges indicates that a batter makes a successful attack; blue edges indicates that a pitcher makes a successful defend. Click on a vertex will light up all its neighbors and corresponding links. The user can also choose to search for any specific player in the search box.

In Fig 3. is the similar interface for ranking players by all teams. The notation are all the same, and it is easy to find out from the network the most significant player and team.

In Fig 4, it shows that when you click a node, it will only show its ego-network and make the others opaque.

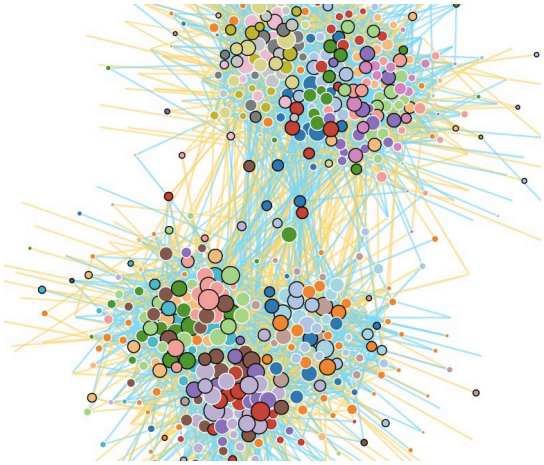

Figure 3: Ranking Player by ALL Teams, 2005 with GameRank

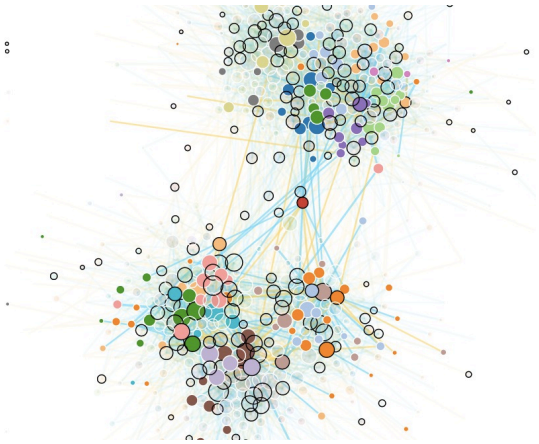

Figure 4: Click a node to show its Egonet

Fig. 5 shows the interface of ranking teams. User can change the year and ranking metric (PageRank or Out-Degree) The size of the vertex indicates its rank, and actual rank value can also be checked upon mousing over / clicking the vertex.

4. DATA ANALYSIS

We calculated the degree distribution of the player network, and recorded a few years' for reference.

In Fig 6 is the CDF of the degree distribution of all player throughout the year of 1950, 1960, 1970, 1980, 1990, 2000 and 2010 respectively.

We see from the figure that the degree distribution is almost linear, which indicates that the number of players in different levels are similar.

The number of nodes and edges of the network, according other statistics, has been ever-increasing over time. Despite of this, we see that the degree distribution has been changing: transformed into a probability density distribution, the tail is getting shorter, and the head is getting smaller. This illustrates the fact that there used to be only a few elite

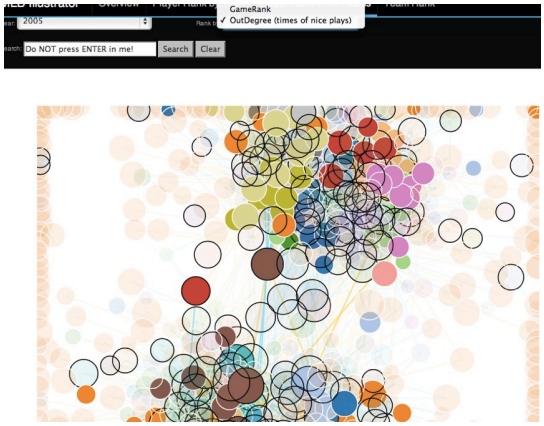

Figure 5: Ranking teams with Out-Degree

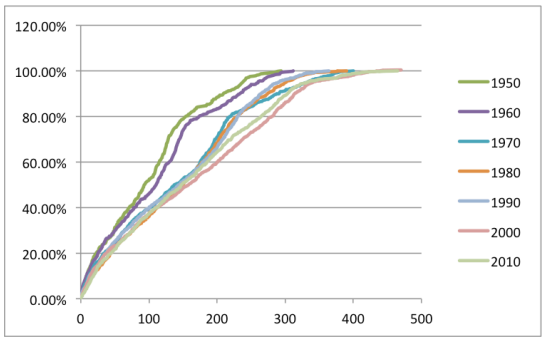

Figure 6: Degree Distribution

players dominating the game–with a lot of nice plays (higher degree), but now there are more players contributing edges targeted at each other, i.e. the skill of everyone are getting closer. As the time goes, the long tail is slowly but surely disappearing.

5. CONCLUSION

In this report we've come up with a novel approach to analyze the complex statistics data of MLB data, that is, to transform the data from simple numbers and situations into a network with multiple indicators interplaying with each other. And in such network, GameRank algorithm is introduced as a simple and neat approach to evaluate individual players in the league. Modified from PageRank and HITS, it takes a player's performance into consideration as a probability estimation problem and models up the problem as a Markov process with a twist. Other popular network analysis techniques are also applied on both team and individual in the baseball game. Finally, the networks along with the analysis are visualized using tools like d3, and a working system is out on the street for users to interact with the data.

6. FUTURE WORK

6.1 Improving product

We want to make our product in the following ways, to make it more useful to all the baseball analysts and general baseball fans.

First, we definitely want to add a "Game Replay" function, to display any game as a "network over time", i.e. describing the procedure of the game as a changing network. We can layout the nodes according to their positions on the field, and generate and hide links in the time order, then it will be a live show of the whole game. The work is fun and have some benefits: you can view any game in the history, even though you cannot find the video of it; and you only need to download a 1MB JSON file to watch the whole game.

Second, we want to add multiple statistics to make our site more useful and complete: the top players in each year, the typical boxing scores (AVG, SLG, etc) of each player, the geographical location of each team on a colored map, and so on.

6.2 Future Analysis

Because of the time limit, we cannot conduct various analyses of our data. Here lists what we can do with our data in the future.

First, we can test the robustness of each team based on the knowledge of network resilience. For each team we build a network: the nodes are the players, the directed edges from A to B indicates that A gives a support to B when A is batting. And we can analyze this network for each team. If it is a highly-centralized network, it shows that the team is too dependent on certain players, and it will be dangerous for the team to lose him. Otherwise if the network is robust, we can say that the team has many good players and is stable.

Second, we can dig into some interesting facts: which players have a high GR but do not play much? they might be unfairly treated, or they are not endurable to play many games. Which players have a high GR but a low salary? You can hire him in a low price, and that will benefit your team. Which pitchers are the most tough to all the players in your team? He might not be the strongest, but he just wins over you every time, and he is the dangerous one to your team.

7. REFERENCES

[1] `http://www.retrosheet.org/`.
[2] `http://d3js.org/`.
[3] `http://jquery.com/`.
[4] Google. The pagerank citation ranking:bringing order to the web.
[5] J. Kleinberg. Hubs, authorities, and communities.