2012 ASE International Conference on Social Informatics (SocialInformatics 2012) / 2012 ASE International Conference on Cyber Security (CyberSecurity 2012) / 2012 ASE International Conference on BioMedical Computing

GameRank: Ranking and Analyzing Baseball Network

Zifei Shan, Shiyingxue Li, Yafei Dai
School of Electronic Engineering and Computer Science, Peking University
Beijing, China
{shanzifei, lsyx09, dyf}@pku.edu.cn

***Abstract*—In the paper we present an algorithm called GameRank, modified from Pagerank and HITS, to evaluate the pitching and batting ability for players in Major League Baseball (MLB) with a network perspective. The model could also be easily expanded and applied on any network that has multiple factors interacting with each other, to quantify the vertex's significance. Then, we evaluate the algorithm by comparing its results to ESPN Ratings, a popular baseball rating method. Our algorithm achieves similar or better results with a way simpler model. Furthermore, relevant analysis is also performed for our MLB data network, with a few interesting conclusions drawn, like (a) players are getting closer in their skills; (b) good pitchers bats better than normal ones.What's more, we have wrapped up the whole system as a working website, called MLB Illustrator (http://mlbillustrator.com), 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. At last, we present a series of examples where GameRank model can be used, to prove that our model is extensive and widely applicable.**

Our contribution lies in the following aspects: (a) we provide a simple model to rank the nodes in networks with multiple indicators interplaying with each other, which expands the functionality of PageRank, and is widely applicable; (b) we initially apply the network theory on the baseball network, handle a set of analysis on it, and have some interesting findings; (c) we provide a powerful method to rank baseball players which is stronger than ESPN Ratings in several aspects.

***Index Terms*—Social Networks, Ranking, Data Mining, Baseball, Algorithm**

I. Introduction

Major League Baseball [1], 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. Among researchers, previous works are mainly focusing on analyzing baseball videos [2], [3], and there are seldom works on ranking players and analysis of baseball network.

Our data is parsed from Retrosheet.org [4], which keeps a 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. From the dataset we parse each play into a win-lose relationship between two players: the batter and the pitcher.

While PageRank [5] 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 field (catch balls). To better evaluate players, we've come up with a novel approach, inspired by PageRank and HITS [6], to iteratively measure the performance of the given individual. The algorithm aims at measuring the pitching and batting ability for each player. It assigns each individual GameRank (GR) values to represent his pitching / batting ability, and use multiple random walk models to iteratively accumulate the GR value. The results of GameRank, its evaluation, along with other analysis results, will also be shown.

The rest of our paper is arranged as following. Section II gives a more detailed description of GameRank, explains how it can be mapped into a random walk model, how it can be computed, and proves its convergence properties. Section III evaluate the GameRank algorithm, compare it with a popular ranking method—ESPN Ratings, and prove that GameRank achieves similar or better results with ESPN Ratings with a simpler and more natural model. Section IV 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. Section V introduces our product—MLB illustrator, to visualize the baseball network according to GameRank, which gives a better view of the relationships among players and illustrates GameRank values to provide further analysis. Section VI gives some examples that GameRank is applicable, to support that this model can be widely used and easily expanded. In Section VII we propose future works, and finally in Section VIII we make a conclusion.

II. Algorithm: GameRank

A. Introduction and Motivation

Each player, in a baseball game, has multiple abilities such as batting, pitching, fielding and running. Batting and pitching abilities are what people care the most, and we aim to evaluate these two qualities of baseball players. These two ability cannot be evaluated independently: a good hitter sometimes achieve a home run facing a great pitcher, and a good pitcher sometimes strikes out a great batter. Most pitchers also have the ability to bat, and some of them are even very good at batting. In this way, simple PageRank is unable to capture such a network's feature with the two abilities, since only one indicator PR is calculated. If we use PageRank separately for pitching and batting abilities on two networks $G_{batting}, G_{pitching}$, we cannot describe the interplay within the two factors; if we do not separate batting and pitching abilities, but only compute PageRank on the network where

978-0-7695-4938-5/12 $26.00 © 2012 IEEE
DOI 10.1109/SocialInformatics.2012.21

edges stand for win-lose conditions, we can only get a coarse significance of all players, which is unnatural and inaccurate.

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.

This model is quite similar to hubs and authorities [7], which is the abstraction of Web presented by HITS algorithm: good hubs link to good authorities, and good authorities are linked by good hubs. Although HITS does not perform well in the context of Web, this intuition fits in well in the baseball network.

To iteratively calculate each player's ability, a random walk model is applied to obtain the stationary distribution of the player's pitching and batting abilities, i.e. GameRank values. One of the strength of the random walk is that it provides a good probabilistic meaning for the algorithm, and fits in well with our assumption. Detailed description of the random walk model is stated in the following subsection.

B. Intuition: Random Walk

GameRank can be seen as two random walk models interacting with each other. The following example illustrates the intuition behind GameRank, which is exactly the same process as the algorithm we will present later.

Say, Ellie, a big fan of MLB, has all the games' data recorded in pieces of "nice plays". A nice play from i to j is defined as one play in the game that pitcher i wins over batter j, or batter i wins over pitcher j. Ellie wants to find out who is the best player of the year, so she randomly starts from a batter A, and randomly picks a nice play that pitcher B defeats A. Then she look over B's data and picks a nice play that a batter C defeats B. As she continues this never-ending game, after a sufficiently long time, the probability that she is watching the play of a batter x, called $GRB(x)$; or of a pitcher x, called $GRP(x)$, represents x's batting (pitching) ability.

Considering that there might be a batter i is so lucky that no pitcher wins over him throughout the year, or vise versa when i is a pitcher. If Ellie gets to batter (pitcher) i, she will randomly pick a pitcher (batter) from all the players, and restart her journey.

Also, Ellie sometimes (with a probability β) gets bored when watching plays of batter (pitcher) i, and directly jump to a random pitcher (batter) j to look for some surprise.

In this process, the more nice plays x has, the higher possibility x is visited, thus gaining the higher GR values. If x is a batter and he defeats many pitchers with high GRP, then his GRB will be high; if x is a pitcher and he defeats many batters with high GRB, then he will get high GRP. This is the intuition of our algorithm, and it is in accordance with our assumption of baseball games.

C. GameRank Definition

The GameRank algorithm is defined as formulas in this subsection.

Definition 1: In a simple unweighted network, a Batting Edge from A to B means that A wins over B when A is batting and B is pitching. Similarly, a Pitching Edge from A to B means that A wins over B when A is pitching and B is batting.

N is the number of vertices. $DB_{in}(i)$ is the in-degree of vertex i when i is batting, i.e. the number of pitching edges targeting at i. $DP_{in}(i)$ is the in-degree of vertex i when i is pitching, i.e. the number of batting edges targeting at i. $outlinks_P(i)$ is the set of endpoints of pitching edges starting from i. $outlinks_B(i)$ is the set of endpoints of batting edges starting from i.

Then Batting Ability is

$$GRB(i) = \beta/N - (1-\beta) \sum_{j \in outlinks_B(i)} \frac{GRP(j)}{DP_{in}(j)}, \qquad (1)$$

Pitching Ability is

$$GRP(i) = \beta/N - (1-\beta) \sum_{j \in outlinks_P(i)} \frac{GRB(j)}{DB_{in}(j)}, \qquad (2)$$

where β is the damping factor, which equals 0.15 in our calculation.

In our real case, the edges of the network is weighted. The weight of edges indicates the significance of the edge. Similar to weighted PageRanks, we revise the GameRank algorithm as the following:

Definition 2: $WDB_{in}(i)$ is the weighted in-degree of vertex i when i is batting, i.e. the number of pitching edges targeting at i. $WDP_{in}(i)$ is the weighted in-degree of vertex i when i is pitching, i.e. the number of batting edges targeting at i. $w_P(i,j)$ is the weight of pitching edge from i to j. $w_B(i,j)$ is the weight of batting edge from i to j.

Then Batting Ability is

$$GRB(i) = \beta/N - (1-\beta) \sum_{j \in outlinks_B(i)} \frac{w_B(i,j) GRP(j)}{WDP_{in}(j)}, \qquad (3)$$

Pitching Ability is

$$GRP(i) = \beta/N - (1-\beta) \sum_{j \in outlinks_P(i)} \frac{w_P(i,j) GRB(j)}{WDB_{in}(j)}, \qquad (4)$$

GRB and GRP values describe the batting and pitching abilities of players. With these values, we can rank the players by batting and pitching abilities separately, thus got *GR batting rank* and *GR pitching rank*.

D. 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 (3) and (4), we collect the GR values of each vertex. The process is repeated until GR values converges to the stationary distribution.

To make our rankings more accurate, we set different weights for various type of edges. The weights of edges indicate the significance of the edge, or how nice the play

TABLE I
WEIGHT FOR DIFFERENT KINDS OF EDGES

Edge Class	Edge Type	Weight
Batting	Single Base	1
Batting	Double Base	2
Batting	Triple Base	3
Batting	Home Run	4
Batting	Sacrifice Hit	0.5
Batting	Walk / Base-on balls	0.5
Batting	Others	0.5
Pitching	All	1

is. According to our algorithm, higher edge weights will contribute more in the computation of GR values.

We define "Edge class" as pitching or batting edge according to the source of the edge, i.e. a pitcher wining over a batter is mapped into a pitching edge. In real baseball games, the nice plays are not merely win-lose conditions, but a various types such as Single Base (1B), Home Run(HR), Sacrifice Hit, Walk, etc. In the table I, we specify how we assign a weight according to the type of edges. For simplification, we assume that the value of base-hits (1B, 2B, 3B, HR) are proportional to the bases the batter run, so we assign weights equal to the number of bases, i.e. 1, 2, 3 and 4. Other types of battings like Sacrifice Hit and Walk, are assigned a 0.5.

Different edge types have different weights, since we think when batter i faces pitcher j, a home run might contribute more than a single base to evaluating i's batting ability.

The weight can be adjusted freely, and cause different results. We picked basic and intuitive weights, to prove that this model is effective and improvable. By picking a set of weights with more professional knowledge, we can make the algorithm more accurate.

This computation can be easily parallelized, using the same metrics of PageRank. It can be calculated in a method of MapReduce model [8], which is applicable for large scale networks.

E. Convergence Properties

It happens that some vertices have no in-links, as no one can beat him 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. With these miracle links, the network become connected without dangling nodes hanging around, and stationary distribution could be obtained.

For simple illustration of convergence, we first take a look at the convergence proof of the original PageRank. Let π^T be the $1 * n$ PageRank row vector, we can describe the iteration at the $k - th$ step as

$$\pi^{(k+1)T} = \pi^{kT} H$$

where H is the row-normalized adjacency matrix.

To ensure stochasticity, H should be of non-negative elements and every row in it should sum to one, yet by definition there could be rows summing up to zero. We define stochastic $H = H + \frac{ae^T}{n}$, where $a_i = 1$ is a column vector if $\sum_{k=1}^{n} H_{ik} = 0$ and $a_i = 0$ otherwise, e is the unit column vector.

To guarantee there exists unique stationary distribution vector π^T, H should be irreducible, which happens if and only if the corresponding graph is strongly connected [9]. This is where damping factor β comes in. With $0 \leq \beta \leq 1$ and $E = ee^T/n$, we get irreducible, row-stochastic matrix $H = \beta H + (1 - \beta)E$. Then we can rewrite the iteration equation as

$$\pi^{(k+1)T} = \pi^{kT} H$$

provided that π will converge to the unique stationary distribution.

Back at GameRank, we build and modify the adjacency matrices in accordance to that above. The elements are by definition non-negative and normalized, the dangling nodes are taken care of to make sure each row sums up to one, and β make the matrices irreducible. GRB and GRP then, should converge to their corresponding unique stationary distribution.

III. Evaluation

In order to evaluate that our GameRank algorithm has good performance, we conduct a series of experiments.

A. ESPN Ratings

Firstly, we want to prove that GameRank algorithm achieves at least the similar effect with a well-recognized and prestigious ranking method, ESPN Ratings [10], and the GameRank model is simpler, and in some aspects stronger, than ESPN Ratings.

ESPN Ratings is defined by Jeff Bennett in the official site of ESPN. In its algorithm, batters, starting pitchers and relief pitchers are separated into different ranking groups. For each group, it calculates the weighted average of all the factors as a single score to describe the players' value. As an example, the ESPN rating of batters are calculated by gathering the following factors: batting bases accumulated, runs produced, OBP, BA, HRs, RBIs, runs, hits, net steals, team win percentage, difficulty of defensive position, etc. It includes more than 10 factors. Both of the other groups also include more than 5 factors.

The calculation of ESPN Ratings involves computing the weighted average of many statistics of players. Its algorithm is only a simple sum-up to a bunch of indicators, which makes it very complex and unnatural. In addition, none of those indicators can take the detailed relationships among players into consideration. Furthermore, it separates pitchers with batters, thus cannot compare pitchers' and batters' batting abilities. What is worse, a lot of players fail to get a score according to the algorithm, thus a large number of players accounting for more than 60% cannot get a rank, while our GR algorithm can rank all the players according to game data. Compared to ESPN Ratings, our algorithm is simpler, more natural, capable to compare pitchers and batters in terms of batting ability, and covers a larger majority.

B. Comparing GameRank and ESPN Ratings

We pick the match data of all the teams in 2011, as it is the most recent and complete in Retrosheet, and official ratings can also be found in ESPN. We calculate the GameRank values for all the players according to the match data in 2011, and get GR values for 1295 players. Then we collect the ESPN Ratings for pitchers and batters, which only involves 161 pitchers and 310 batters.

Then we analyze the players who have ESPN Ratings, and rank them according to both GR values and ESPN Ratings, then compare their GR ranks and ESPN ranks. To be specific, we normalize GR and ESPN rank by selecting the players whose GameRank values and ESPN Ratings are both provided. Actually, all the players who have ESPN Ratings also have their GR values calculated, so the selected set includes 161 pitchers and 310 batters. We separately sort all the selected pitchers and batters by their GR values, and the order is $GRrank$; similarly we sort the players by ESPN ratings and get $ESPNrank$. Therein, both $GRPitchingRank$ and $ESPNPitchingRank$ have a range $[1, 161]$, and both $GRBattingRank$ and $ESPNBattingRank$ are ranged in $[1, 310]$. So GR and ESPN ranks can be compared.

First, we plot the scatter diagram of $GRrank - ESPNrank$, arranged by $GRrank$. Figure 1(a) for batting, and figure 1(b) for pitching. The value is a indicator of a certain player how close the two ranks are.

Then we plot Cumulative Distribution Functions (CDFs) to see the distribution of absolute difference values, figure 1(c) for batting, figure 1(d) for pitching. The horizontal axis refers to $|GRrank - ESPNrank|$, and the vertical axis is its cumulative function, i.e. how many users have the difference below this value. The difference value shows the closeness of the two ranks.

From the plot we can see that, for 50% batters, the difference between their GR rank and ESPN rank is less than 37; for 80% batters, the difference is less than 85. For 50% pitchers, the difference is less than 40; for 80% pitchers, the difference is less than 81. A smaller difference demonstrates a better similarity of GR rank and ESPN rank. As the range of pitching rank is $[1, 161]$ while the range of batting rank is $[1, 310]$, it comes out that the GR ranks are more close to ESPN ranks in terms of batting than pitching.

To provide some specific statistics, the top 10 batters and pitchers or the year 2011 according to GR ranks are listed in the table II and table III, with comparison to the ESPN ranks. From these tables, we can also confirm that the difference between our rankings and ESPN rankings are small, and they are more close in terms of batting than pitching, which is in accordance with the above results.

Secondly, we want to prove that GameRank algorithm is in some way more persuasive than ESPN Ratings, in an experimental approach. As ranking the players is quite a subjective procedure, there is no definite criteria to judge which ranking method is better. However, we come up with a intuitional assumption: players with better rankings should have higher probability to win in games.

TABLE II
TOP-10 BATTERS

Name	GR Rank	ESPN Rank
Matt Kemp	1	1
Prince Fielder	2	6
Justin Upton	3	17
Hunter Pence	4	21
Ryan Braun	5	2
Joey Votto	6	8
Albert Pujols	7	12
Adrian Gonzalez	8	5
Jacoby Ellsbury	9	3
Jose Bautista	10	7

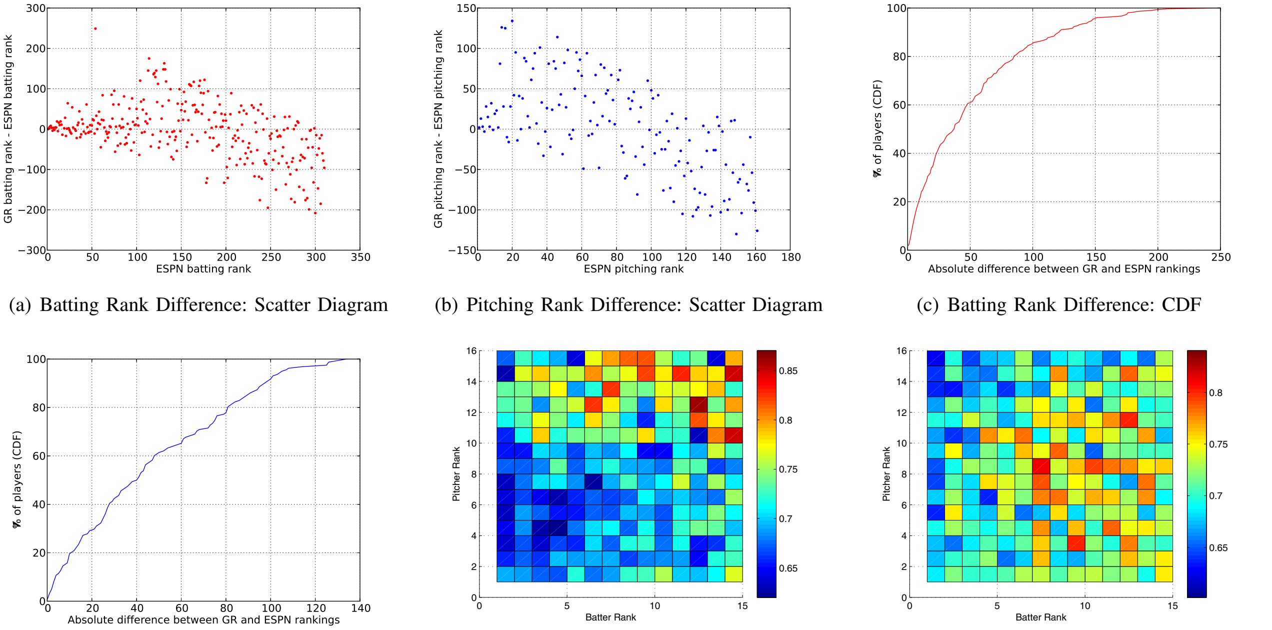
TABLE III
TOP-10 PITCHERS

Name	GR Rank	ESPN Rank
Cliff Lee	1	4
Matt Cain	2	18
Clayton Kershaw	3	1
Daniel Hudson	5	38
Roy Halladay	6	3
Tim Lincecum	7	17
Ian Kennedy	8	9
Tim Hudson	9	23
James Shields	10	7

With this assumption, we plot two figures visualizing the frequency for pitchers at different rank levels to win over batters at different rank levels. Figure 1(e) for GR ranks, and figure 1(f) for ESPN ranks. In these figures, the horizontal axis refers to batters, and the vertical axis refers to pitchers at different levels in the according ranking algorithm. The maps are cut into grids for every 10 pitchers and every 20 batters, and the color of grids refers to average frequency for pitchers at specific rank levels to win over batters at specific rank levels. The redder, the higher frequency for pitchers to win. In specific, for each grid with the left-bottom corner at point (x, y), the color of that grid refers to the average frequency for pitchers with rank in $(10 \times (y-1), 10 \times (y)]$ to win batters with rank in $(20 \times (x-1), 20 \times (x)]$.

Based on the assumption, the figures show that GR ranks are better than ESPN ranks in terms of batters: When x-axis is growing which means pitchers meets “weaker” batters, the frequency for pitchers to win gets more obviously higher in GR ranks than in ESPN ranks. However, the GR ranks for pitchers do not seem to have good patterns. Probably we can say that we achieve better rankings for batters using GameRank, but for pitchers it is still hard to make such a conclusion.

According to the above comparison, we find that GameRank algorithm is quite effective in the following ways: (a) it achieves at least similar results with ESPN rankings, and it is even better in terms of batting rankings if we set the criteria as wining frequency. (b) What is more, it is such a simple model that only uses win-lose relationships between batters and pitchers in games, featuring a perspective of networks. (c) At last, it can give rankings to all the players as long as they are in the network, while ESPN Ratings only have a small rated set of 161 pitchers and 310 batters.



(a) Batting Rank Difference: Scatter Diagram

(b) Pitching Rank Difference: Scatter Diagram

(c) Batting Rank Difference: CDF

(d) Pitching Rank Difference: CDF

(e) Frequency for pitchers to win batters at different rank levels in GameRank. Pitcher ranks are divided by 10; batter ranks are divided by 20.

(f) Frequency for pitchers to win batters at different rank levels in ESPN rank. Pitcher ranks are divided by 10; batter ranks are divided by 20.

Fig. 1. Comparison between GR and ESPN rankings

GameRank can be made more precise if we dig into the edge weights: how much is the weight for Home Runs, Sacrifice Flys, and Walks? What if we consider more complicated and various situations in baseball games? By customizing weights of different kind of edges, we can easily extend this method, and make it more powerful at ranking baseball players.

IV. Data Analysis

Then we handled analysis with the calculated $GRrank$, and some interesting results are found in the baseball network: (a) By studying its out-degree distribution in different years, we found that recent players are getting closer in their skills than before. (b) By analyzing the pitchers' batting ability, we found that good pitchers are better than normal pitchers at batting.

A. *Out-degree Distribution Analysis*

First, we calculated the out-degree distribution of the player network, and recorded a few years' for reference. The out-degree is defined as the total number of outlinks, including both pitching and batting edges, indicating the total numbers of nice plays a player achieves inside a year.

In figure 2 is the CDF of the out-degree distribution of all player throughout the year of 1950, 1960, 1970, 1980, 1990, 2000 and 2010 respectively.

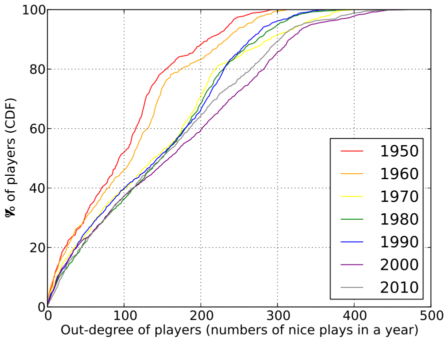


Fig. 2. Players' out-degree distribution of historical years

We see from the figure that the out-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 to statistics not shown, has been ever-increasing over time. Despite of this, we see that the out-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 players dominating the game—with a lot of nice plays (higher out-degree), but now there are more players contributing edges targeted at others, i.e. contributing nice plays. As the time goes by, the long tail is slowly but surely disappearing, i.e. players are gradually getting closer in skills.

B. *Pitcher's Batting Ability Analysis*

As we know, pitchers also have batting ability, some of which bats well. We pick all the 660 pitchers, who have

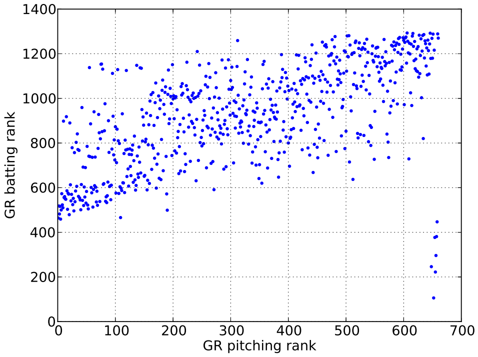


Fig. 3. Pitcher's batting ranks

TABLE IV
BOTTOM PITCHERS WHO ARE GREAT BATTERS

Name	Batting Rank	Pitching Rank
Wilson Valdez	246	648
Michael Cuddyer	106	652
Darnell McDonald	377	654
Skip Schumaker	222	655
Bryan Petersen	296	656
Mike McCoy	381	657
Mitch Maier	447	658

pitching ability in the year 2011 in all leagues. For them, both GRP and GRB are meaningful. We plot their pitching ranks and batting ranks in one scatter diagram in figure 3, and find something interesting.

As the figure shows, among all the pitchers, there is a trend that "better pitchers bat better". This might be opposite to someone's intuition that great pitchers put all their minds in developing their pitching skills, and thus fall behind others on batting. Actually, the statistics indicates that good pitchers are usually more talented or well-trained than normal pitchers, not only in pitching, but also in batting.

Another interesting thing in this figure is that among the bottom pitchers, there are 7 pitchers who bats really well: their pitching ranks are behind 640, but their batting ranks are before 450, some of which even rank about 100, which is far better at batting than any other pitcher in the leagues. We pick them up and list them in table IV.

We manually check these players, and found that most of them do not take pitchers as their major fielding positions. Although they have the ability to pitch, and they all once pitched in 2011 regular season, they are actually better at batting, and usually do not pitch.

V. Visualization: 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 contribution of our visualization tool is: (a) providing a perspective of network to display the baseball statistics rather than the traditional tables of statistics, leading a prospective new trend of baseball game analysis; (b) helping potential further analysis of baseball network, by presenting a straightforward view of GameRank values and the relationships among pitchers and batters.

The website of our system is: **mlbillustrator.com** [11].

Our website is built upon the visualization toolkit D3.js [12], jQuery [13], and basic javascript and html.

A. *Definition*

In our visualization, Nodes are players. Bigger nodes are stronger players. Nodes with same color inside are in the same team. Nodes with black border are pitchers, with white border do not pitch.

Edges are plays, i.e. winning relationship between two players. Color of edges indicates the kind of a play: a successful defense (blue) or attack (yellow). If a batter makes a hit or other successful attacks, then he initiates a yellow (attack) link towards the rival pitcher. If a pitcher strikes out a batter or leads his team to a successful defense, then he initiates a blue (defense) link towards the rival batter. Nicer plays lead to thicker links, and multiple thin links will aggregate to a thick one.

And users can easily interact with the system according to the manuals on the website.

B. *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 outdegree. Team rank uses the team's PageRank value, modified to adjust to the MLB network.

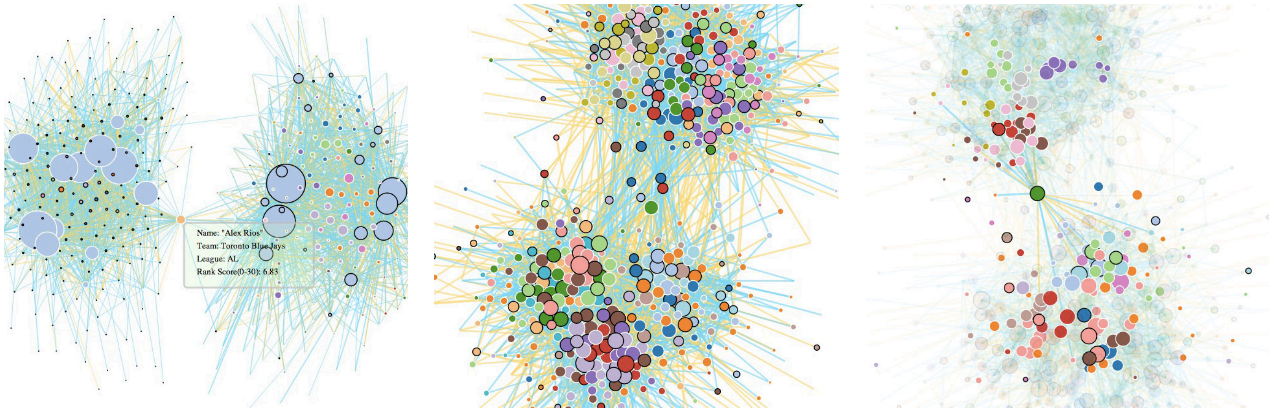
For ranking player by team, the user can choose year, team and ranking metric. The network of your choice will be shown automatically in the central part of the window. By using this function, we can easily see who are the core players in each team. Figure 4(a) shows an example of Chicago White Sox, in 2009.

In figure 4(b) 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 his team.

In figure 4(c), it shows that when you click a node, it will only show itself and its neighbors, and make the others opaque.

C. *Analysis based on visualization*

We handle some simple analysis based on our visualization system. We take the all-teams-in-one-year network, like in figure 4(b), and find that in every year, the network consists of two large communities. The layout is force-driven layout provided by D3 library, and this result shows that the links within each community are dense, while links between two communities are sparse.



(a) Ranking Player by Team, 2009, Chicago White Sox with GameRank (b) Ranking Player by ALL Teams, 2005 with GameRank (c) A node and its neighbors in the network

Fig. 4. Ranking and visualization by MLBillustrator

Why does this happen? We found that in MLB there is American League (AL) and National League (NL), and the two clusters are almost exactly AL and NL communities. The effect is because both AL and NL play more inside themselves, but less across leagues (only in league championship series).

And the players who is the bridge of two communities, like the selected node in figure 4(c), have links with both leagues. We manually check some of these nodes, and find that all of them used to change their teams across the league during the year. For example, Dan Haren was traded from Arizona Diamondbacks (NL) to Anaheim Angels (AL) in 2010, thus he is a bridge in the layout of 2010.

VI. Other use cases

It turns out that GameRank is a simple, effective algorithm, which fits in the situation where there are multiple interplaying factors.

In this section, we provide some other use cases that GameRank is also applicable.

A. Football Network

In Football, each player has attacking and defending ability. Nice forwards are usually good at attacking, backfielders are good at defending, and midfielders might be good at both attacking and defending.

The assumption is that: if a player often beats rivals who are good at defending when he is attacking, then he is a good attacker. In the contrary, if he successfully defends good attackers, then he is a good defender.

Each player has two GameRanks: attacking and defending.

Definition 3: An Attacking Edge from A to B means A wins over B when A is attacking and B is defending. Similarly, a Defending Edge from A to B means A wins over B when A is defending and B is attacking. N is the number of vertices. $DA_{in}(i)$ is the in-degree of vertex i when i is attacking, i.e. the number of defending edges targeting at i. $DD_{in}(i)$ is the in-degree of vertex i when i is defending, i.e. the number of attacking edges targeting at i. $outlinks_D(i)$ is the set of endpoints of defending edges starting from i. $outlinks_A(i)$ is the set of endpoints of attacking edges starting from i.

Then Attacking Ability is

$$GRA(i) = \beta/N - (1-\beta) \sum_{j \in outlinks_A(i)} \frac{GRD(j)}{DD_{in}(j)}, \quad (5)$$

Defending Ability is

$$GRD(i) = \beta/N - (1-\beta) \sum_{j \in outlinks_D(i)} \frac{GRA(j)}{DA_{in}(j)}, \quad (6)$$

where β is the damping factor.

By calculating the GameRanks, we can measure the attacking and defending abilities for all football players.

B. Network with three interplaying factors

Imagine there is a network in which nodes have three attributes A, B and C, and there are three types of edges: $< A, B >$, $< B, C >$ and $< C, A >$, using different attributes of nodes. In this network, if node X point to (wins) node Y through an $< A, B >$ edge, then the attribute B of Y contributes to A of X, and similar rules are adopted with the other types of edges: wining higher C leads to higher B, and wining higher A leads to higher C, through accordant edge types.

In this network, each node can have three GameRanks for A, B and C. $outlinks_A(i)$ is the set of endpoints of $< A, B >$ edges starting from i. $outlinks_B(i)$ is the set of endpoints of $< B, C >$ edges starting from i. $outlinks_C(i)$ is the set of endpoints of $< C, A >$ edges starting from i. $DA_{in}(i)$ is the in-degree of i when counting $< C, A >$ edges. $DB_{in}(i)$ is the in-degree of i when counting $< A, B >$ edges. $DC_{in}(i)$ is the in-degree of i when counting $< B, C >$ edges. And the three abilities can be quantified like:

$$GRA(i) = \beta/N - (1-\beta) \sum_{j \in outlinks_A(i)} GRB(j)/DB_{in}(j),$$

$$GRB(i) = \beta/N - (1-\beta) \sum_{j \in outlinks_B(i)} GRC(j)/DC_{in}(j),$$

$$GRC(i) = \beta/N - (1-\beta) \sum_{j \in outlinks_C(i)} GRA(j)/DA_{in}(j),$$

where β is the damping factor, and j is in the corresponding set of endpoints.

This example shows a case where GameRank algorithm can be extended to fit in networks with multiple interplaying indicators.

VII. Future Work

With the dataset and the visualization system, we might handle further measurements on the baseball network.

First, we can test the robustness of each team based on the knowledge of network resilience. For each team a network can be built: 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? He might be bought in a low price, and that can be a good bargain. Which pitchers are the toughest to the players in one team? He might not be a top pitcher, but he keeps winning you every time, and your team should be cautious about him.

Moreover, we can use specific knowledge in baseball games to optimize our algorithm, such as dividing starting pitchers and relievers, and selecting precise edge weights. We can also try to predict the result of certain games by a comprehensive study of the network.

VIII. Conclusion

In this paper we present a novel approach to analyze the complex statistics 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 effective 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.

We evaluate the GameRank algorithm by comparing its results and rankings according to ESPN Ratings, a famous and well-recognized approach to rate baseball players. The result shows that our model is excellent in the following aspects: first, we get the similar results with ESPN ranks, if not better. Second, our method only use a simple model which only needs the win-lose relationships in plays, which is far more independent than ESPN Ratings. Third, our method is capable of calculating every player's rankings, while in ESPN more than half of players will not get a score so that they cannot be ranked. Fourth, our method is capable to join the pitchers with batters and compare their batting ability, while ESPN Ratings fail to do so.

Besides, other popular network analysis techniques are also applied on both team and individual in the baseball game. We calculate the out-degree distribution of the network for all the past years, and find that the head is getting smaller and the tail is shorter, indicating that recent players are getting closer in their skills than before. We discuss the pitcher's batting ability, and find that good pitchers are better than normal pitchers at batting.

Then, we provide a visualization system as our product, MLB illustrator featuring GameRank on-line calculations. With this working system out on the street for users to interact with the data, more interesting patterns and knowledge are there to be discovered.

At last, we give some cases that GameRank can also be applied, to demonstrate that this model is flexible and applicable for any network featuring multiple interplaying indicators.

Acknowledgment

The authors would like to thank Professor Qiaozhu Mei, our course advisor, for giving us a lot of encouragement and suggestions on this project, which helps a lot in forming this paper.

The authors would like to thank Yilun Zhang, for helping us in ploting some figures.

References

[1] http://mlb.mlb.com.

[2] W.-C. Tsai, H.-T. Chen, H.-Z. Gu, S.-Y. Lee, and J.-Y. Yu, "Baseball event semantic exploring system using hmm," in *Proceedings of the 17th international conference on Advances in multimedia modeling - Volume Part II*, ser. MMM'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 315–325. [Online]. Available: http://dl.acm.org/citation.cfm?id=1950054.1950091

[3] Y.-F. Huang and J.-J. Huang, "Semantic event detection in baseball videos based on a multi-output hidden markov model," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC '11. New York, NY, USA: ACM, 2011, pp. 929–936. [Online]. Available: http://doi.acm.org/10.1145/1982185.1982390

[4] http://www.retrosheet.org/.

[5] Google, "The pagerank citation ranking:bringing order to the web."

[6] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, Sep. 1999. [Online]. Available: http://doi.acm.org/10.1145/324133.324140

[7] J. M. Kleinberg, "Hubs, authorities, and communities," *ACM Comput. Surv.*, vol. 31, no. 4es, Dec. 1999. [Online]. Available: http://doi.acm.org/10.1145/345966.345982

[8] J. Lin and M. Schatz, "Design patterns for efficient graph algorithms in mapreduce," in *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, ser. MLG '10. New York, NY, USA: ACM, 2010, pp. 78–85. [Online]. Available: http://doi.acm.org/10.1145/1830252.1830263

[9] C. D. Meyer, Ed., *Matrix analysis and applied linear algebra*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2000.

[10] http://sports.espn.go.com/mlb/news/story?id=2897967.

[11] http://mlbillustrator.com.

[12] http://d3js.org/.

[13] http://jquery.com/.