AN ELO-BASED APPROACH TO MODEL TEAM PLAYERS AND PREDICT THE OUTCOME OF GAMES

A Thesis

by

RICHARD EINSTEIN DOSS MARVELDOSS

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Jean Francios Chamberland
Committee Members,	Gregory H. Huff
	Thomas R. Ioerger
Head of Department,	Scott Miller

August 2018

Major Subject: Computer Engineering

ABSTRACT

Sports data analytics has become a popular research area in recent years, with the advent of different ways to capture information about a game or a player. Different statistical metrics have been created to quantify the performance of a player/team. A popular application of sport data anaytics is to generate a rating system for all the team/players involved in a tournament. The resulting rating system can be used to predict the outcome of future games, assess player performances, or come up with a tournament brackets.

A popular rating system is the Elo rating system. It started as a rating system for chess tournaments. It's known for its simple yet elegant way to assign a rating to a particular individual. Over the last decade, several variations of the original Elo rating system have come into existence, collectively know as Elo-based rating systems. This has been applied in a variety of sports like baseball, basketball, football, etc. In this thesis, an Elo-based approach is employed to model an individual basketball player strength based on the plus-minus score of the player. The plus-minus score is a powerful metric because it quantifies the contribution of a player like good defense, setting up screens, or sledging the opposite team, which are not reflected by metrics that are primarily based on points. Then, the individual player ratings are combined to obtain a team rating, Team rating are compared pairwise to obtain the probability of a win by each of the teams during a matchup. This method not only predicts wins/losses, but offers more information than the Elo rating system as ratings are assigned to each individual player instead of just considering teams. This information includes for example, the effect of mid-season transfers or the impact of injuries to team strengths; these items are overlooked by the standard Elo algorithm.

The performance of the proposed Elo-based rating system is compared to that of the standard Elo rating system for basketball by using sythetic data. The rating systems are also compared by running them over real-life data from past NBA seasons.

DEDICATION

This dissertation is dedicated to my family, professors and friends for their guidance and encouragement which made this work possible.

ACKNOWLEDGMENTS

I would like to thank Dr. Jean Francois Chamberland for his support and guidance throughout the course of my thesis. This phase has helped me understand what research really is and I have learnt life-long lessons. I would also like to thank Dr. Gregory Huff and Dr. Thomas Iorger who offered me the freedom to venture into the domain of my interest.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Jean Francios Chamberland and Professor Gregory H. Huff of the Department of Electrical and Computer Engineering and Professor Thomas R. Ioerger of the Department of Computer Science.

All other work conducted for the thesis was completed by the student independently.

NOMENCLATURE

+/-	Plus-minus score
NBA	National Basketball Association
NCAA	National Collegiate Athletic Association
FIDE	Fédération Internationale des Échecs
USCF	United States Chess Federation
ATL	Atlanta Hawks
BOS	Boston Celtics
BKN	Broklyn Nets
CHA	Charlotte Hornets
CHI	Chicago Bulls
CLE	Cleveland Cavaliers
DAL	Dallas Mavericks
DEN	Denver Nuggets
DET	Detroit Pistons
GSW	Golden State Warriors
HOU	Houston Rockets
IND	Indiana Pacers
LAC	Los Angeles Clippers
LAL	Los Angeles Lakers
MEM	Memphis Grizzlies
MIA	Miami Heats
MIL	Millwakee Bucks

MIN	Minnesota Timberwolves
NOP	New Orleans Pelicans
NYK	New York Knicks
OKC	Okhalohoma Thunder
ORL	Orlando Magic
PHI	Philadelphia 76ers
PHX	Phoneix Suns
POR	Portland Trailblazers
SAC	Sacramento Kings
SAS	San Antonio Spurs
TOR	Toronto Raptors
UTA	Utah Jazz
WAS	Washington Wizards

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TABLE Page

1. INTRODUCTION AND LITERATURE REVIEW

1.1 Rating system

A rating system analyzes the outcomes of games and assigns a number/value to the strength of a team/player relative to others. Sports rating systems have been in existence for about 80 years. Initially, rating systems were computed on paper. Common systems include rating based on expert voters, layman majority voting and computer systems. These sports rating systems have been applied to a wide variety of sports like chess, basketball, baseball, etc A rating system can be used to:

1) Predict the outcome of a match;

2) Form brackets so that the teams that are likely to win do not end meeting in the earlier stages of the tournament;

3) Select participants in elite tournaments;

4) Allow team to gauge how well they are doing.

In the last 20 years, the development of strength assessment tools has roused the curiosity of statisticians in applying them to sports rating systems. This general approach has proven to be more robust and reliable than systems that are based on guesses from domain experts. Additionally, the advent of higher processing power and big data technology has made the processing of game-play possible. This information also allows teams and coaches to make wise decision regarding drafts and team formation. For example, Nate Silver a well known statistician, has succesfully used game-log data to come up with rating systems for various team sports[2].

1.1.1 Rating system for basketball tournaments

Rating system models are applied to a wide variety of sports, and they have become popular in modelling and predicting professional basketball games in particular. Several such contributions have appeared in the literature, especially to predict the brackets for the NCAA basketball tournament and the NBA finals. In almost every professional basketball game, game play and event log data are recorded. These statistics can accurately describe the performance of a team or player. For example, Fig 1.1 is a sample of the various measures that are collected after every NBA game for every player. These metrics include minutes played, points scored, plus-minus score, field goals made, field goals attempted, etc. This information can be used to build mathematical models that can quantify the relative strengths of NBA teams. A simple, elegant approach is the Elo rating system applied to basketball. Variations of the Elo-based approach consider wins/losses, victory margin, home court advantage to create a system that quantifies the strengths of NBA teams. These systems are used to predict who will win in future contests.

PLAYER	TEAM	DATE	MATCHUP	W/L	MIN	PTS	FGM	FGA	FG%	3PM
Shaun Livingston	GSW	03/27/2018	GSW vs. IND	L	19	8	4	6	66.7	0
Brandon Jennings	MIL	03/27/2018	MIL @ LAC	L	12	0	0	2	0.0	0
Kevin Love	CLE	03/27/2018	CLE @ MIA	L	7	1	0	2	0.0	0
Jabari Parker	MIL	03/27/2018	MIL @ LAC	L	19	2	1	12	8.3	0
Pat Connaughton	POR	03/27/2018	POR @ NOP	W	11	0	0	4	0.0	0
Kyle Collinsworth	DAL	03/27/2018	DAL @ SAC	W	26	3	1	2	50.0	0
Jeff Green	CLE	03/27/2018	CLE @ MIA	L	31	5	2	10	20.0	0
JR Smith	CLE	03/27/2018	CLE @ MIA	L	16	2	1	4	25.0	0
Buddy Hield	SAC	03/27/2018	SAC vs. DAL	L	25	14	6	15	40.0	2
LaMarcus Aldridge	SAS	03/27/2018	SAS @ WAS	L	18	13	4	6	66.7	0

Figure 1.1: Example of the statistics collected for an NBA game, reprinted from[1].

1.2 Elo algorithm

The Elo algorithm is a popular rating system in the sports community. The Elo algorithm was developed by Arpad Elo to rate players in chess tournaments. It is used to determine chess player ratings in the Fédération Internationale des Échecs (FIDE) and the United States Chess Federation (USCF). Nowadays, the Elo algorithm is widely used for rating players/teams in football, basketball, and even multiplayer video games. The Elo algorithm is used to develop a relative rating system based on a match by match basis. The Elo system models a match as a pairwise comparison. By pairwise comparison, we mean that when two teams are compared, one of the team is preferred to win over the other. The Elo algorithm assumes that, over a period of time, the estimated player strength (player rating) approaches its true value. The difference between the rating of two teams determines the predicted outcome of a game. This is expressed as a probability of one team winning over the other. A player rating is represented as a number which increases/decreases based on whether the team wins or loses. The Elo rating system states that the team with the higher rating is more likely to win. The greater the difference in rating between the teams is, the more likely the stronger team is to win. If a highly rated team wins, then the increase in its rating is not as much when compared to the difference low rated team beats a highly rated team. There is a larger transfer in points for an upset win. Also, the Elo algorithm maintains the overall sum of the ratings in a manner similiar to a zero sum game. This is due to the fact that after every game the points gained by the winning team is the points lost by the defeated team. This allows the comparison of the team's performance throughout the course of the tournament possible.

1.3 Mathematical Model

In the Elo algorithm [3], the realized strength of each team is assumed to come from a normally distributed random variable where the mean corresponds to the true strength of the team. The normal distribution is a good assumption because, on an average, a team performs roughly with the same strength in every game. Variations do exist, but larger deviations are less common than smaller variations. All these properties are satisified by the normal distribution, and this assumption works suprisingly well. The estimate of a team's strength is updated iteratively based on observed outcomes, namely wins and losses. If Team i plays Team j, then the rating update is given by:

$$R_{inew} = R_{iold} + K(S_{ij} - \mu_{ij}). \tag{1.1}$$

1.3.1 K-factor

Parameter K controls how much weight should be given to a recently concluded game as compared to past games (prior information). The higher the K value is, the faster the estimate adapts. Still, an undesirably high K can lead to large oscillations in the rating estimate. On the other hand, a lower K leads to slower adaptation. The K value is callibrated based on factors that govern the game.

1.3.2 Actual Score (S_{ij})

Variable S_{ij} denotes the value used in the Elo update equation. The actual score considered is the win/loss information. The defintion of S_{ij} is mentioned below:

$$S_{ij} = \begin{cases} 1, & \text{if } Team_i \text{ beats } Team_j \\ 0, & \text{if } Team_j \text{ beats } Team_i \\ 0.5, & \text{if } Team_i \text{ draws with } Team_j. \end{cases}$$

1.3.3 Expected Score (μ_{ij})

Parameter μ_{ij} is the expected measure of the win/loss of $Team_i$ against $Team_j$. It is a logistic function of the difference in rating between r_i and r_j. Two teams competing in a game is modelled using the pair wise comparison model called Thurstone-Mostelller Model[4]. Under the Thurstone-Mostelller Model[5][6], the realized team strength is assumed to be governed by a random variable with a normal distribution. Another competing pair comparison model is the Bradley-Terry model [7]. In the Bradley-Terry model, the logistic function is chosen as an approximation when the two strengths are exponential distributed random variables. When two players meet, the performance of each player can be modelled as a normal random variable. For example, when a player with Elo rating 1500 meets a player with Elo rating 1900(fixed standard deviation of 400 is assumed.). This can be represented as in Fig 1.2.

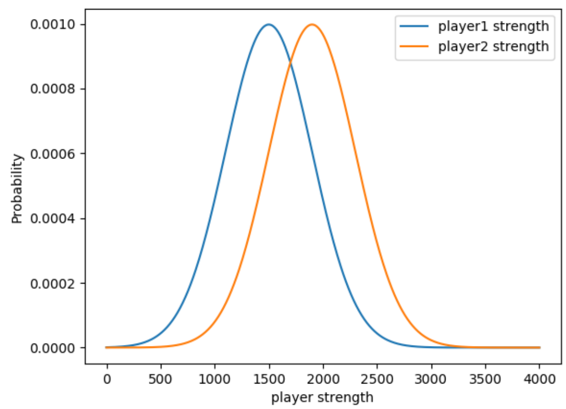


Figure 1.2: This figure is the distribution of player strengths with player 1 having an average rating of 1500; and player 2, a rating 1900.

For the sake of analysis, the difference between the ratings can be considered a random variable. This random variable would again be normally distributed, albeit with a mean of -400 (1500-1900) and a standard deviation of $\sqrt{2}\sigma$. This is shown in Fig 1.3. Since we are interested in findng the probability of player 1 winning/lossing against player 2, the shaded region in Fig 1.3 represents the probabilty that player 1 wins against player 2.

The area under the curve in Fig 1.3 can be represented as a cumulative distribution function. Thus for a difference in rating (D), we have a probilistic function depicted by blue curve Fig 1.4 where the y-axis represents the probability that player 1 wins against player 2. Though the analysis above assumes normal distributions, Stern [8] has shown that, when analyzing paired comparison data, there is no significant difference whether one assumes logistic distributions or normal distributions for the player strengths. The preference of one model over the other is a

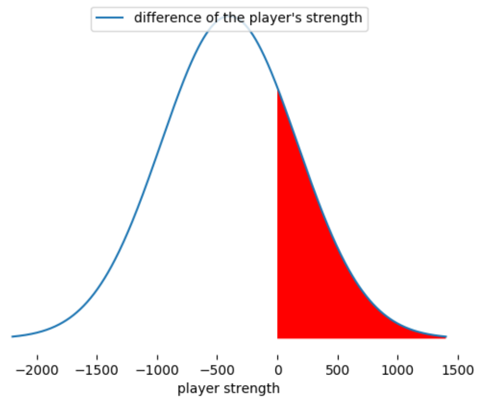


Figure 1.3: This figure represents the distribution of difference in player strengths with player 1 having an Elo rating of 1500 and player 2 with an Elo rating 1900 respectively.

matter of debate. From Fig 1.4, it can be seen that logistic function is approximately the same as the normal distribution. Hence, the logistic function can be employed to compute the probability of a win. The use of the logistic function is widely accepted among chess federation systems, like FIDE and USCF.

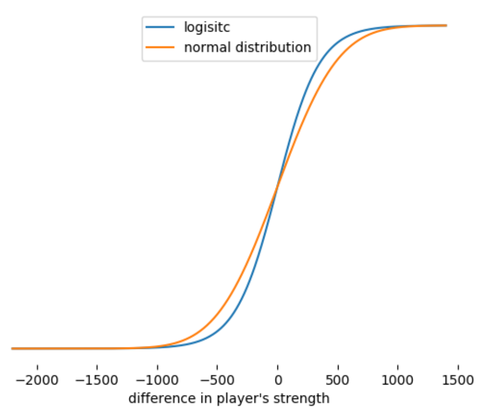


Figure 1.4: Comparsion of logisitc distrbution and normal distribution

The probability that Team i wins against Team j is given by

$$\Pr(i > j) = \frac{P_i}{P_i + P_j},$$

where P_i and P_j are real-values scores assigned to Teams i and Team j. When an exponential score is considered, the above expression reduces to a logistic function as follows:

$$\Pr(i > j) = \frac{e^{r_i}}{e^{r_i} + e^{r_j}},$$

where r_i and r_j are the ratings of Team i and Team j. The standard Elo is of the form:

$$\Pr(i > j) = \frac{1}{1 + 10^{\frac{r_j - r_i}{400}}}, \tag{1.2}$$

where 400 captures the spread of the logistic curve. Let $\Pr(i > j)$ be denoted by μ_{ij} for all further discussion. For example, when a Team i with rating 1500 plays against Team j with rating 1900, then μ_{ij} is given by

$$\mu_{ij} = \frac{1}{1 + 10^{\frac{-(1500-1900)}{400}}} = 0.09.$$

This means that there is a 9% chance for a team with a 1500 rating to win against a team with a rating of 1900.

The Elo update equation is designed to incorporate upset wins. An upset win, i.e. when a team with a lower rating wins, will lead to a larger rating change than that associated with an anticipated win. From the above example, if Team i (1500) beats Team j (1900), the $(S - \mu)$ factor becomes 0.91, which leads to a bigger change in the update equation of $Team_i$. The above algorithm can be run over matchups to obtain probabilistic predictions and to update team ratings as well.

1.4 Home advantage

Sometimes, the Elo-based algorithms incorporate home advantage as a factor in estimating wins and losses in matchups. This can yield significant improvements to the performance of the Elo algorithm. Especially in basketball, home court advantage has historically played a big role in helping the home team win, even though they may not be as strong as the opposition [9]. This is due to factors like crowd cheering and the familiarity of the home team with the environment, giving them an advantage in the game.

Home advantage can be incorporated by adding a fixed amount of points to the home team before predicting who the winner will be. In the Elo algorithm, the fixed points added is typically 100 points [10].

1.5 Update Algorithm

Algorithm 1 Update to team rating according to Elo algorithm

Initialize all the teams ratings to 1000
for all matchup between two teams $team_i$ and $team_j$ **do**
  Compute μ_{ij} and μ_{ji} which corresponds to the probability of $team_i$ and $team_j$ winning respectively
  Update ratings for $team_i$ and $team_j$ are according to equation (1.1)
end for

2. PROPOSED HYPOTHESIS

2.1 Motivation

The Elo algorithm was orginally designed to develop a rating system for chess players. As the Elo system gained in popularity, statisticians started to apply it to sports like football and basketball. Yet, in applying the Elo system to a team sport, the players are not given individual attention and the whole team is treated as a single entity for the sake of the rating system. In this thesis, we explore the benefits of modelling the strength of each basketball player, and we develop an Elo-based system to estimate the strength of players to predict wins and losses when two teams compete against each other.

Since the true strength of a player cannot be measured, one can try to estimate the strength of the player by using observable metrics of a player like points scored, minutes played, plus-minus score, field goals made, etc. The plus-minus score is the observed metric used in our algorithm. This is similiar to a basic inference problem where a hidden parameter is estimated using observable value. Using prior information a likelihood function is calculated using the observable metrics and posterior values are computed. This offers the advantage of being able to obtain more information about individual basketball players. Our algorithm is designed to track the performance of individual basketball players, and it can perform all the functionalities that a conventional team Elo-based basketball rating system offers.

2.1.1 Plus-Minus Score (+/-)

The plus-minus score started as a metric in hockey. It was compiled in the National Hockey League (NHL) statistics. The metric quicky gained popularity in basketball as well. In fact, it seems to be more meaningful in basketball than hockey because of its high scoring nature. The higher the scoring is in a game, the more meaningful the metric becomes. On a basic level, the plus-minus score measures the contribution of the player to the team. It assesses the performance of the team when the player is on the court. Additionally, the plus-minus score has the ability to capture good defense, self-less play, and other contributions that cannot be captured by points cored alone. For example, during the 2016-17 regular season in the Golden State Warriors and San Antonio Spurs game, Draymond Green played a key factor in the Warrior's win despite only scoring 11 points. This is evident from the fact that his plus-minus score was +28, which is in fact more than that of Stephen Curry who had a plus-minus score of +26 despite scoring 42 points. This shows the power of the plus-minus score as compared to point-based metrics. The plus-minus score is denoted as +/-.

More specifically, the plus-minus score for a player A on Team 1 playing against Team 2 is calculated as follows: if player A is on the court for 10 minutes and during that time Team 1 scores 20 points while Team 2 scores 18 points, then the plus-minus score is +2.

A positive number indicates that the player has a postive effect in that their team performs well (scores more) when they are present. Typically, the plus-minus score can vary from -45 to +45 in an NBA season.

2.2 Elo-Based Approach

The contribution of each player is modelled as a normally distributed random variable with mean proportional to the player's strength and a constant variance. An Elo-based approach is employed. The Elo algorithm attemps to iteratively bring the estimated value of a player's strength close to its actual value. It can be described according to the following equation:

$$player\hat{r}ating = playerrating + Kf(actualscore - expectedscore). \tag{2.1}$$

The plus-minus score of the player is the observed metric used in this algorithm. That is, the plus-minus score is substituted for the actual score in the equation (2.1). A mathematical model is developed to compute the expected score. Function $F(\cdot)$ is used to normalize the difference. Herein, we adopt a modified version of the logistic function.

2.2.1 Strength of a Player

The parameter used for an NBA basketball player is the point contribution per minute. Let it be denoted as p. Then, p quantifies the strength of a player. Each basketball player is initialized with a p value of 1000, i.e, a normalizing constant η is multiplied. This makes the interpretation of data easier. During the calulation of the expected score the p value is normalized again.

$$1000 = \eta p \tag{2.2}$$

A suitable value of η is found out empirically.

2.2.2 Expected Value

To estimate the plus-minus score of a player, a mathematical model is constructed. Whenever two teams have a matchup, the individual player strengths are combined to generate a weighted strength parameter for the team. The strength of the $Team_j$ is calculated as follows:

$$m_j = \frac{\sum_{n=1}^{N} t_{jn} p_{jn}}{\sum_{n=1}^{N} t_{jn}}, \tag{2.3}$$

where t_{jn} is the minutes played by the nth player of the jth Team and p_{jn} is the estimated player strength of the nth player of the Team j.Then, m_j is the average point scored per minute by Team j. From the defintion of plus-minus score, the expected plus-minus score is the difference in the point scored by the two teams while the player is on the court. The strength of a $Team_j$ without $player_i$ is calculated as:

$$\hat{m}_{ji} = \frac{\sum_{n=1,n\neq i}^{N} t_{jn} p_{jn}}{\sum_{n=1,n\neq i}^{N} t_{jn}} \tag{2.4}$$

Since a team has 5 players on the court at any point in the game, our model assumes that the total points scored by a $Team_j$ in t minutes is:

$$PTS_j = 5 m_j t, \tag{2.5}$$

where PTS_j denotes points scored by $team_j$. To calculate the plus-minus score, we need to find the point scored by the player's team and the opposite team. The underlying assumption is that, while they are on the court, a player contributes to the strength of their team. The remaining strength components come from the other players.

Then the total number of points scored by the a team during the time the player is on court is well approximately by

$$PTS_j = (p_{ji} + 4\hat{m}_{ji})t_{ji} \tag{2.6}$$

During that time, the expected number of points scored by the opposite team ($team_k$) can be computed as

$$PTS_k = 5m_k t_{ji}. \tag{2.7}$$

Let t_{ji} be the time played by the ith player belonging to $team_j$. The estimated plus-minus score (μ_{ij}) can be written as

$$\mu_{ij} = (PTS_j - PTS_k)/\eta. \tag{2.8}$$

Parameter η in (2.8) is the normalizing constant to compensate for the initialization of 1000 points for the initial strength. Then,

$$\mu_{ij} = t_{ji}(p_{ji} + 4\hat{m}_{ji} - 5m_k)$$

.

Let the actual plus-minus score of $player_i$ be denoted as S_{ij}. Substituting this value in (2.1), we get

$$\hat{p}_{ji} = p_{ji} + Kf(S_{ij} - \mu_{ij}). \tag{2.9}$$

2.2.3 K-factor

The K factor in this thesis is dependant on the minutes played by a basketball player. A player who spent considerable amount on the court should have a higher weightage in updating the rating than a player who spent less time on court as the game information for that player would have a weaker significance. The K factor can be expressed as

$$K = \gamma \hat{t}, \tag{2.10}$$

where $\hat{t}_i$ is the normalized version of minutes played by the player,

$$\hat{t} = \frac{t}{48}. \tag{2.11}$$

Time t is the minutes spent on the court by the player and 48 is choosen as the normalizing constant because that is the maximum number of minutes a player can be on court in a regular game. We make the assumption that there is no overtime play. Parameter γ is choosen by running it over synthetic or real-life data and K is chosen so that maximum predictions are obtained; γ helps in controlling the effect of the likelihood function and the prior information in obtaining the posterior information.

2.2.4 F(x)

This function is used to normalize the difference of the actual score and expected score to a value between -0.5 and 0.5. The function choosen is a slight variation of the logistic function that is used in a standard Elo algorithm. A large difference between the actual score and the expected score suggests that the estimates have not converged to the actual strengths. This function ensures that a large difference would lead to a change of large magnitude up to a point, Figure 2.1 represents the curve that is used.

2.2.5 Point Conservation

The standard Elo algorithm is designed in a way that the total rating across all teams remains a constant. Applying the point conservation to (2.9) for each player involved in a game between $Team_i$ and $Team_j$ we get

$$\sum_{l=1}^{N_j} \hat{p}_{jl} + \sum_{m=1}^{N_k} \hat{p}_{km} = \sum_{l=1}^{N_j} p_{jl} + \sum_{m=1}^{N_k} p_{km} + \sum_{l=1}^{N_j} K f(S_{lj} - \mu_{lj}) + \sum_{m=1}^{N_k} K f(S_{mk} - \mu_{mk}), \tag{2.12}$$

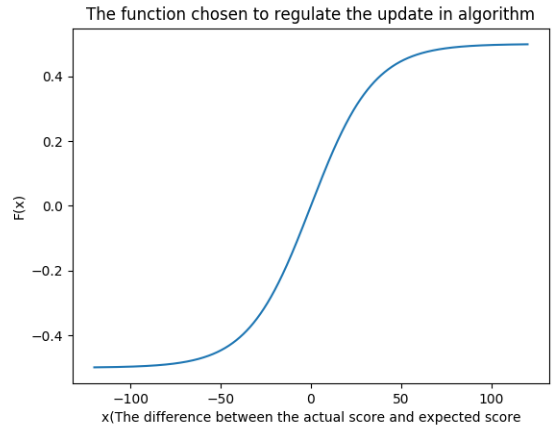


Figure 2.1: The function used to normalize the difference between the actual plus-minus and expected plus-minus score.

The above equation can rewritten as

$$\sum_{l=1}^{N_j} \hat{p}_{jl} + \sum_{m=1}^{N_k} \hat{p}_{km} = \sum_{l=1}^{N_j} p_{jl} + \sum_{m=1}^{N_k} p_{km} + \sum_{l=1}^{N_j} t_l b_l + \sum_{m=1}^{N_k} t_m b_m \tag{2.13}$$

where $Kf(S_{mi} - \mu_{mi})$ is replaced by $t_m b_m$. Let $\hat{b}$ be the value for which the point conservation is satisfied. In order for points to be conserved, the left hand side of (2.13) should cancel out with the first two term on the right hand side. Applying this condition to (2.13) gives us

$$\sum_{l=1}^{N_j} t_l \hat{b_l} + \sum_{m=1}^{N_k} t_m \hat{b_m} = 0. \tag{2.14}$$

In matrix form it can expressed as

$$\begin{bmatrix} T_l & T_m \end{bmatrix} \begin{bmatrix} \hat{B}_l \\ \hat{B}_m \end{bmatrix} = 0 \tag{2.15}$$

where T_l, T_m are row block vectors of dimensions $1 \times N_j$ and $1 \times N_k$, respectively. Also $\hat{B}_l$, $\hat{B}_m$ are column vectors of dimensions $N_j \times 1$ and $N_k \times 1$.

From (2.13), we can obtain the offset d as

$$\sum_{l=1}^{N_j} t_l b_l + \sum_{m=1}^{N_k} t_m b_m = d. \tag{2.16}$$

So a transformation is done over b that would allow (2.4) to be achieved. Equation (2.6) can be represented in a matrix form,

$$\begin{bmatrix} T_l & T_m \end{bmatrix} \begin{bmatrix} B_l \\ B_m \end{bmatrix} = d, \tag{2.17}$$

where T_l, T_m are row block vectors of dimensions $1 \times N_j$ and $1x \times N_k$ respectively. Also B_l,B_m are column block vectors of dimensions $N_j \times 1$ and $N_k \times 1$.

Equation (2.17) can be rewritten as

$$\begin{bmatrix} T_l & T_m \end{bmatrix} \begin{bmatrix} B_l \\ B_m \end{bmatrix} = \frac{d}{2} 2. \tag{2.18}$$

Further, it can be rewritten as

$$\begin{bmatrix} T_l & T_m \end{bmatrix} \begin{bmatrix} B_l \\ B_m \end{bmatrix} = \frac{d}{2} \begin{bmatrix} T_l & T_m \end{bmatrix} \begin{bmatrix} \frac{1}{\sum_{l=1}^{N_j} t_l} \\ \frac{1}{\sum_{m=1}^{N_k} t_m} \end{bmatrix}. \tag{2.19}$$

Moving the $\frac{1}{2}$ to the left-hand side and moving the factor d into the column block matrix, we get

$$\begin{bmatrix} T_l & T_m \end{bmatrix} \begin{bmatrix} 2B_l \\ 2B_m \end{bmatrix} = \begin{bmatrix} T_l & T_m \end{bmatrix} \begin{bmatrix} \frac{d}{\sum_{l=1}^{N_j} t_l} \\ \frac{d}{\sum_{m=1}^{N_k} t_m} \end{bmatrix} \tag{2.20}$$

Rearranging the equation, we obtain

$$\begin{bmatrix} T_l & T_m \end{bmatrix} \begin{bmatrix} 2B_l \\ 2B_m \end{bmatrix} - \begin{bmatrix} T_l & T_m \end{bmatrix} \begin{bmatrix} \frac{d}{\sum_{l=1}^{N_j} t_l} \\ \frac{d}{\sum_{m=1}^{N_k} t_m} \end{bmatrix} = 0 \tag{2.21}$$

$$\begin{bmatrix} T_l & T_m \end{bmatrix} \left(\begin{bmatrix} 2B_l \\ 2B_m \end{bmatrix} - \begin{bmatrix} \frac{d}{\sum_{l=1}^{N_j} t_l} \\ \frac{d}{\sum_{m=1}^{N_k} t_m} \end{bmatrix} \right) = 0 \tag{2.22}$$

$$\begin{bmatrix} T_l & T_m \end{bmatrix} \begin{bmatrix} 2B_l - \frac{d}{\sum_{l=1}^{N_j} t_l} \\ 2B_m - \frac{d}{\sum_{m=1}^{N_k} t_m} \end{bmatrix} = 0 \tag{2.23}$$

Comparing (2.15) and (2.23) , the transformed b that satisfy the points conservation,

$$\hat{B}_l = 2B_l - \frac{d}{\sum_{l=1}^{N_j} t_l}, \tag{2.24}$$

$$\hat{B}_m = 2B_m - \frac{d}{\sum_{l=1}^{N_k} t_l}. \tag{2.25}$$

This results in a two-step update process. The first step involves finding b as

$$b_k = \gamma f(S_{mk} - \mu_{mk}). \tag{2.26}$$

The second step involves tranformating of the offests to account for point conservation

$$\hat{b_k} = 2b_k - \frac{d}{\sum_{l=1}^{N_k} t_l}. \tag{2.27}$$

The second step allows us to conserve the total sum of all the indivdual player ratings. This is critical as it enables comparison of the performance of the team/player over a length of time, and the performance of two different players/teams.

The final update equation is

$$\hat{p}_{ji} = p_{ji} + \hat{t}_{ji}\hat{b}_{ji}. \tag{2.28}$$

2.2.6 Outcome of a game

Using (2.3) overall team rating is computed from the stengths of individual basketball players. Soppose the team rating for Team i is r_i and for Team j is r_j, the probability that Team i wins over Team j is obtained using (2.29). For the sake of simulation, a win is predicted if Team i has a higher team rating that Team j.

$$\Pr(i > j) = \frac{1}{1 + 10^{\frac{r_j - r_i}{400}}}. \tag{2.29}$$

2.3 Update Algorithm

Algorithm 2 below shows how the player rating are updated as the teams play one another. Algorithm 3 describes how the outcome of a games is determined based on the rating of the players.

Algorithm 2 Update player ratings

Initialize all player ratings to 1000
for all $Team_j$ and $Team_k$ in matchups **do**
  for all p_{ji} in players of $team_j$ **do**
    b_{ji} is computed according (2.26)
  end for
  for all p_{ki} in players of $team_k$ **do**
    b_{ki} is computed according (2.26)
  end for
  for all b_l in $team_k$ and $team_j$ **do**
    Compute the normalized $\hat{b}_k$ from (2.27)
  end for
  for all p_{ji} in players of $team_j$ **do**
    Compute updated player rating $\hat{p}_{ji}$ from (2.28)
  end for
  for all p_{ki} in players of $team_k$ **do**
    Compute updated player rating $\hat{p}_{ki}$ from (2.28)
  end for
end for

Algorithm 3 Predict winner of match between $team_i$ and $team_j$

Find effective strength of $team_i$ using (2.3).
Find effective strength of $team_j$ using (2.3).
The one that has the higher strength of the two teams is expected to win.

---

3. DATASET

The previous chapter discussed the mathematical model that is used in predicting player strengths. In this section, two datasets have been generated/used in order to measure the performance of the proposed algorithm. Real-life data is obtained by screen-scrapping data from the NBA website. The NBA website was used to scrape post-game information of the regular and final season for both teams and indiviudal NBA players. Since the obtained datasets is limited, synthetic data was generated and used as well. The methodology behind obtaining the data will be discussed in the following sections.

3.1 Real-life Data

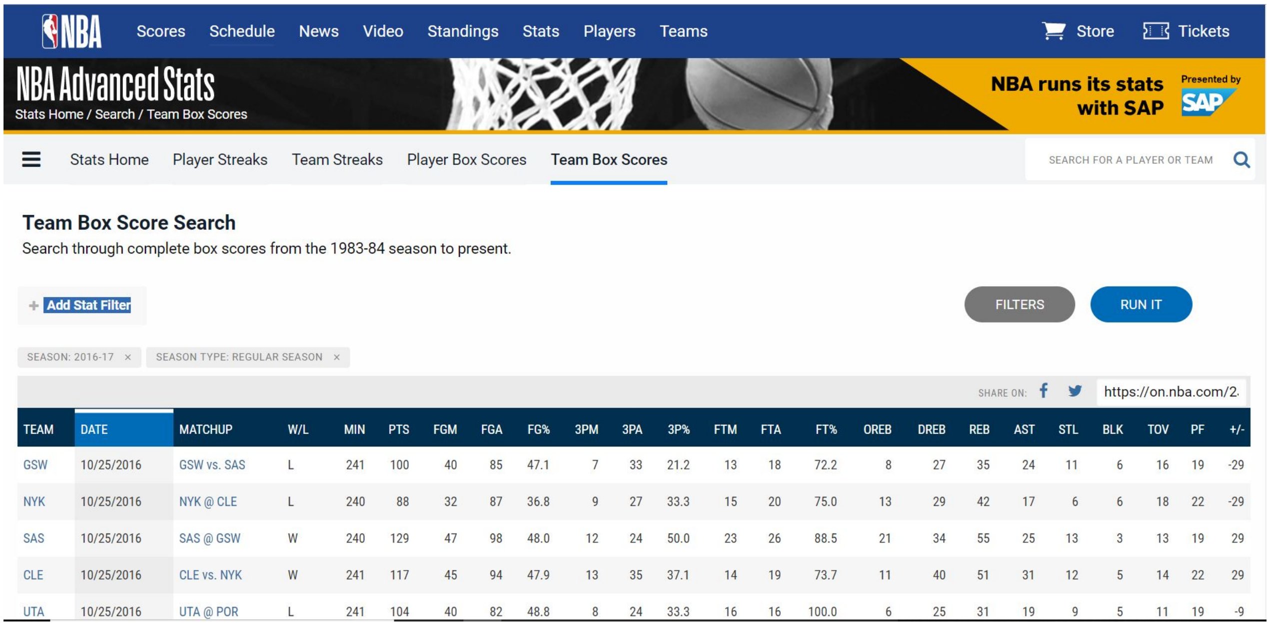
The datasets is collected by scrapping data from the NBA website. The data obtained is from the 2015-16, 2016-17 and 2017-18 regular seasons. Both the team box score and player box score are accounted for.

3.1.1 Data Scrapping

It is a computer program that extracts data from a website and stores it offline in a suitable data structure format. Webpages are typically written in a text-based markup language like HTML or XHTML. A Python program was written to extract the required statistics from tables and is stored locally as a csv file. An example is shown in Fig 3.1 which is webpage that is converted to a csv file format as shown in Fig 3.2 that can be stored locally.

3.2 Synthetic Data

The number of basketball teams choosen are 16 with each team containing 10 players. The true strength of each basketball players is drawn randomly from a beta distribution with α=1.5 and β=5 normalized to give strengths ranging from 900-1500. The Beta distribution is chosen because it can effectively model processes that take up values in a particular range and needs to be unsymmetric. The requirement of unsymmetry is due to the fact that most professional basketball player strengths



TEAM	DATE	MATCHUP	W/L	MIN	PTS	FGM	FGA	FG%	3PM	3PA	3P%	FTM	FTA	FT%	OREB	DREB	REB	AST	STL	BLK	TOV	PF	+/-
GSW	10/25/2016	GSW vs. SAS	L	241	100	40	85	47.1	7	33	21.2	13	18	72.2	8	27	35	24	11	6	16	19	-29
NYK	10/25/2016	NYK @ CLE	L	240	88	32	87	36.8	9	27	33.3	15	20	75.0	13	29	42	17	6	6	18	22	-29
SAS	10/25/2016	SAS @ GSW	W	240	129	47	98	48.0	12	24	50.0	23	26	88.5	21	34	55	25	13	3	13	19	29
CLE	10/25/2016	CLE vs. NYK	W	241	117	45	94	47.9	13	35	37.1	14	19	73.7	11	40	51	31	12	5	14	22	29
UTA	10/25/2016	UTA @ POR	L	241	104	40	82	48.8	8	24	33.3	16	16	100.0	6	25	31	19	9	5	11	19	-9

Figure 3.1: Webpage containing the statisitcs for NBA teams, reprinted from[1].

1	TEAM	DATE	MATCHUP	W/L	MIN	PTS	+/-
2	GSW	10/25/2016	GSW vs. SAS	L	241	100	-29
3	NYK	10/25/2016	NYK @ CLE	L	240	88	-29
4	CLE	10/25/2016	CLE vs. NYK	W	241	117	29
5	SAS	10/25/2016	SAS @ GSW	W	240	129	29
6	POR	10/25/2016	POR vs. UTA	W	239	113	9
7	UTA	10/25/2016	UTA @ POR	L	241	104	-9
8	TOR	10/26/2016	TOR vs. DET	W	240	109	18
9	DEN	10/26/2016	DEN @ NOP	W	240	107	5
10	MIA	10/26/2016	MIA @ ORL	W	241	108	12
11	PHX	10/26/2016	PHX vs. SAC	L	240	94	-19
12	ORL	10/26/2016	ORL vs. MIA	L	240	96	-12
13	PHI	10/26/2016	PHI vs. OKC	L	240	97	-4

Figure 3.2: The scrapped data is stored as a csv file.

tend to concentrated in a particular range because of the competitiveness of professional basketball with a few stray cases on either side of the spectrum. The distribution is plotted in Fig 3.3. The performance of a player during a game varies according to a normal distribution with the mean as the true strength and a finite variance. The standard deviation of player perfromance is assummed to be 400, which is common for Elo algorithms.

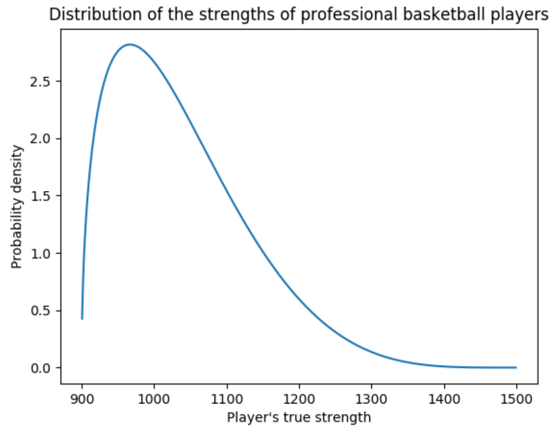


Figure 3.3: The figure depicts the distribution used to draw the player's true strength.

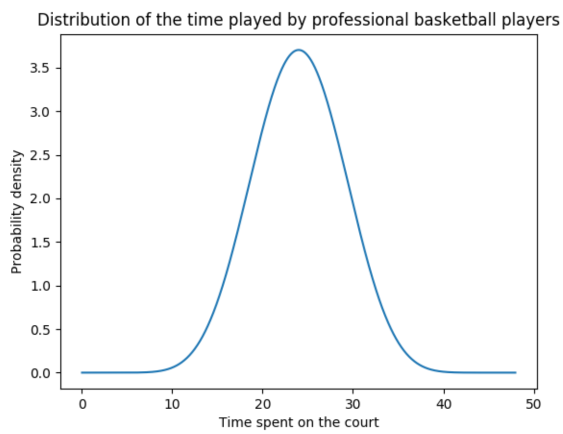


Figure 3.4: The figure depicts the distribution used to assign time played to each basketball player.

3.2.1 Initialization

The algorithm selects two teams out of the 16 teams and the 10 players in each team are assigned player rating from the Gaussian distribution.

The time played is modelled as a symmetric beta distribution with parameters $\alpha = \beta = 11.02$. The value of the parameter is obtained by using the empirically obtained variance value from real life data of 10 random numbers. An additional constraint, the sum of the time spent by all players should be a constant equal to 240 minutes. The distribution is plotted in Fig 3.4.

3.2.2 Metric Calculation

The mathematical model to calcuate the plus-mius score in Section 2.2.1 is used here. The initialized values from the above section are used to initialize each player. The effective strengths of both the teams are calculated according to (2.3). The team having a higher strength is declared the winner. The metrics are entered in two different tables one storing the team matchup information while the other stores individual information.

matchnumber	Player Name	Team	Minutes Played	+/-
0	P2T13	T13	22.0	-4.21385518044057
0	P10T13	T13	25.0	-4.963336920663551
0	P9T13	T13	18.0	-4.0087435655601995
0	P6T13	T13	25.0	-5.007457877079351
0	P4T13	T13	18.0	-4.203899082973941
0	P7T13	T13	33.0	-7.0189041560128755
0	P1T13	T13	18.0	-3.497633785359912
0	P8T13	T13	25.0	-5.1296892641587055
0	P5T13	T13	29.0	-6.588839256803876
0	P3T13	T13	25.0	-5.033927712516916

Figure 3.5: This is an example of the synthetic data generated for team matchups.

Team1 v/s Team2	Game Number	W/L
13 v/s 12	0	L
11 v/s 2	1	L
7 v/s 12	2	W
3 v/s 14	3	W
12 v/s 1	4	W
6 v/s 12	5	W
5 v/s 7	6	L
5 v/s 3	7	W

Figure 3.6: This is an example of the synthetic data generated for players.

4. NUMERICAL SIMULATIONS

The Elo-based rating system algorithm is applied to data proposed in this thesis. The same datasets is applied over the conventional Elo algorithm and is used as the benchmark to compare the performance of the proposed algorithm.

4.1 Synthetic Data

The synthetic data discussed in Section 3.2 is used to check for the convergence of the proposed algorithm. In this case, we have access to the true strength of each and every basketball player. The proposed algorithm is used to estimate the true team strengths. The mean square error is used as a metric to choose the value of hyper-parameter that minimizes the mean square error.

4.1.1 Elo Algorithm

For each K value, 1000 instances of synthetic data is generated and the empirical average is calculated. The K-value that gives the minimum of all the empirical average is chosen. From Figure 4.1 and Table 4.1, it can be seen that for K=2 the mean square error is minimum with a value of 2713.

4.1.2 Proposed Algorithm

The empirical average of the mean square error for various hyper parameters is obtained.
Figure 4.2 shows the plot of variation of mean square error with γ value for a constant η value. The datapoint used are depicted in Table 4.2. From Table 4.2, the best performance is obtained for γ value=100 and η value =2500 with a mean square error value of 2030.35.

The proposed algorithm has a better perfromance when compared to the Elo algorithm. This result is based on the assumption that the data generated is based on the proposed model, which is a optimistic assumption. This motivates for testing the performance on real-life data. Additonally, the convergence of the proposed algorithm has been verified. Figure 4.3 below shows the approximate convergence of mean square error value with the number of matches.

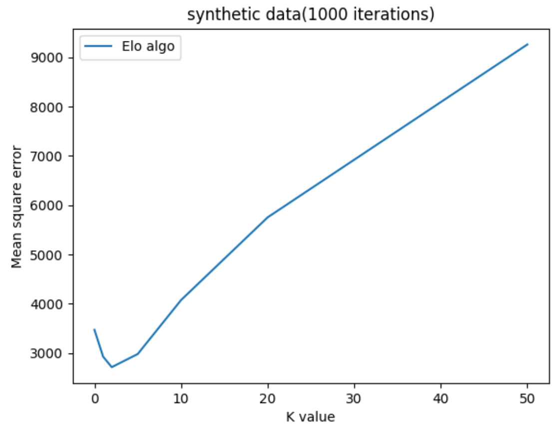


Figure 4.1: Figure show the plot of Mean Square values for various values of K using the Elo algorithm

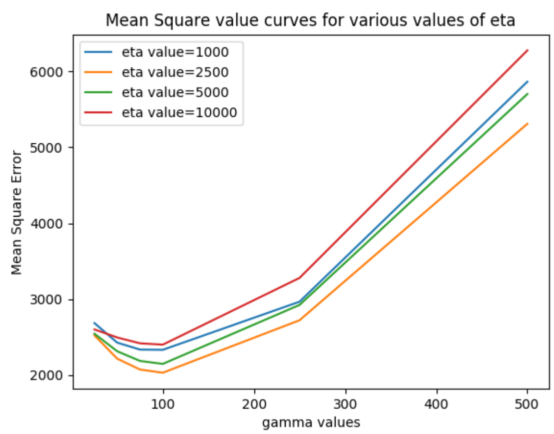


Figure 4.2: This figure represents the plot of Mean Square values for various parameter values of γ and η using the proposed algorithm

4.2 Real-life Data

Since we do not have prior information about the strength of the basketball players, the players are assigned a value of 1000 for every player who plays their first game. The dataset composed of regular season from the 2015-2018 season is split into training, cross-validation and testing set. The 2015-2017 season is split into 75% training set and 25% cross validation set. The training set is used to update the rating of players without any performance metric. The validation set is used

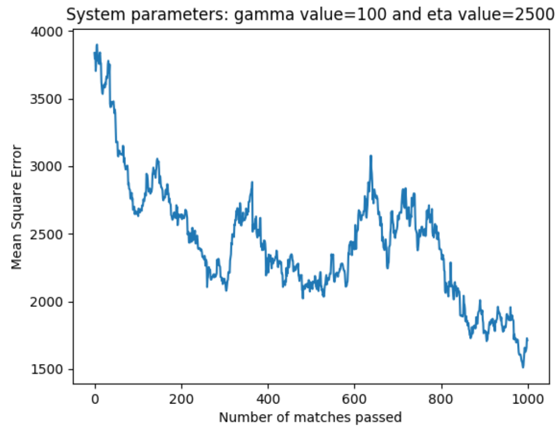


Figure 4.3: Figure depicts the convergence of the proposed algorithm as the algorithm run over the dataset.

K value	mean square error
0	3471.34
1	2926
2	**2713**
5	2980
10	4073.91
20	5752.63
50	9260.93

Table 4.1: Table containing the various mean Square Error values for various K value using the Elo algorithm.

to choose the optimal model parameters like γ, η, K . The perfomance of the proposed algorithm is compared with the Elo algorithm by measuring the predictive discrepancy averaged throughout the course of the season and the prediction rate.[11]

4.2.1 Prediction Discrepancy

For a match between Team i and Team j, the predictive discrepancy is given by :

$$-\left(y_{ij}\log\hat{p}_{ji} + (1 - y_{ij}\log(1-\hat{p}_{ji})\right) \tag{4.1}$$

γ value	η value	mean square error
25	1000	2684.47
25	2500	2522.53
25	5000	2544.45
25	10000	2600.48
50	1000	2426.67
50	2500	2215.44
50	5000	2312.99
50	10000	2492.92
75	1000	2335.38
75	2500	2072.15
75	5000	2185.66
75	10000	2417.65
100	1000	2332.27
100	**2500**	**2030.35**
100	5000	2146.72
100	10000	2400.39
250	1000	2966.21
250	2500	2721.70
250	5000	2923.67
250	10000	3279.50
500	1000	5865.55
500	2500	5310.43
500	5000	5703.31
500	10000	6278.10

Table 4.2: This table contains mean square error values for various parameter values using the proposed algorithm.

where y_{ij} is the binary match outcome

$$y_{ij} = \begin{cases} 1, & \text{if } Team_i \text{ beats } Team_j \\ 0, & \text{if } Team_j \text{ beats } Team_i \end{cases}$$

$\hat{p}_{ij}$ is the probability that Team i beats Team j. This metric is commonly known as the logarithmic loss and used widely in prediction challenges such as kaggle.com. This metric was used in Glickman's paper on comparing different rating system performance for women's volleyball[11].

γ	η	Log-loss x 10000	Correct Predictions(Total Games:615)
100	2500	9070	337
100	10000	10878	372
100	25000	13895	382
250	2500	9359	328
250	10000	11508	390
250	**18781**	**11933**	**391**
500	2500	10049	312
500	10000	11933	387
500	25000	20882	388

Table 4.3: This table contains the redictions results from running the proposed algorithm over real-life data

K value	Log-loss x10000	Correct Predictions(Total Games:615)
1	6630	362
2	6511	366
5	6379	387
10	6362	385
20	6421	387
30	**6471**	**391**
50	6592	389
100	6953	376
200	7942	365
500	12605	353

Table 4.4: This table contains the predictions results from running the ELO algorithm over real-life data

4.2.2 Testing on the NBA 2017-18 season

From the above section, the hyperparameter that gives the maximum accuracy is fixed and the algorithms are run over the NBA 2017-2018 season. Algorithms 1 and 3 are used to predict the outcome of a matchup for the Elo algorithm and the proposed algorithm, respectively. Even as the outcomes are predicted, the ratings of players and teams are updated during the testing phase as well. The update algorithm for the proposed algorithm is Algorithm 2 in Section 2.3.

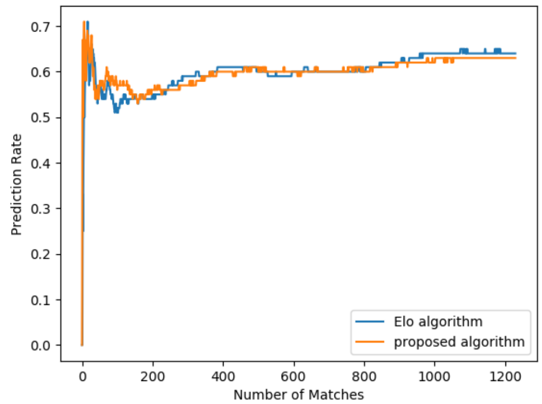


Figure 4.4: This plot compares the prediction performance of the two algorithms

Algorithm	Log-loss x10000	Correct Predictions (Total Games:1230)
Proposed Algorithm	18112	771
Elo Algorithm	8789	786

Table 4.5: This table contains the predictions results from running the Elo algorithm over real-life data

4.2.3 Testing using bracket scores

Another method of comparing the performance of the above disussed algorithm with that of the Elo algorithm is using bracket scores. This metric has been historically used for the NCAA March madness bracket predictions. The system used to rank the various submitted brackets is discussed in Section 4.2.3.1. The system in table 4.7 is used to assign a metric for the proposed algorithm and is compared with the standard Elo algorithm. The modified systen is mentioned in Table 4.4. A maximum of 32 points (8 points each round) can be obtained.The assumption made in this metric is that the seeded team for the playoffs are given rather than extracted from the algorithm, i.e, the first 16 seeded NBA teams in the playoffs are fixed and the rating obtained from our algorithm is used to evaluate who will win the matchups. Additionally, the home advantage is given to a team that has a better regular season record.

4.2.3.1 *March Madness NCAA brackets*

March madness is an annual bracket prediction ranking for the basketball teams competing in the NCAA. Every year, roughly 70 million brackets are submitted by fans all across the USA. The brackets involve coming up with the matchups that might happens from the first round until the winner. The users are given the 64 seeded teams, with that they come up with their own set of brackets which they think is the likely one to happen in real life. The total amount of money involved in this competition roughly comes up to 10.4 billion dollars. March madness uses a point system [12] to rank the different participants competing in the bracket submission contest. The point assigned for every correct prediction is assigned according to Table 4.6. A maximum of 192 points can be earned if all the games are predicted correctly.

Challenge Round	Points
1(First Round)	1 point for each correct selection
2(Second Round)	2 point for each correct selection
3(Sweet Sixteen)	4 point for each correct selection
4(Elite Eight)	8 point for each correct selection
5 (Final Four)	16 point for each correct selection
6 (Championship)	32 point for each correct selection

Table 4.6: The table represents the point assignment system used in the NCAA brcket predicition competition

Challenge Round	Points
1(Sweet sixteen)	1 point for each correct selection
2(Elight Eight)	2 point for each correct selection
3(Final Four)	4 point for each correct selection
4(Championship)	8 point for each correct selection

Table 4.7: The table represents the point assignment system used for comparison of the two algorithms.

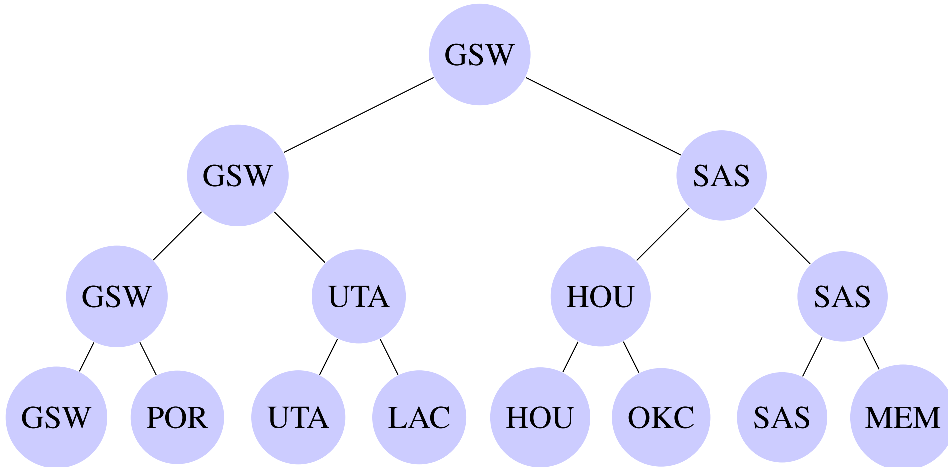


Figure 4.5: This figure represents the winning team bracket for the western conference playoffs.

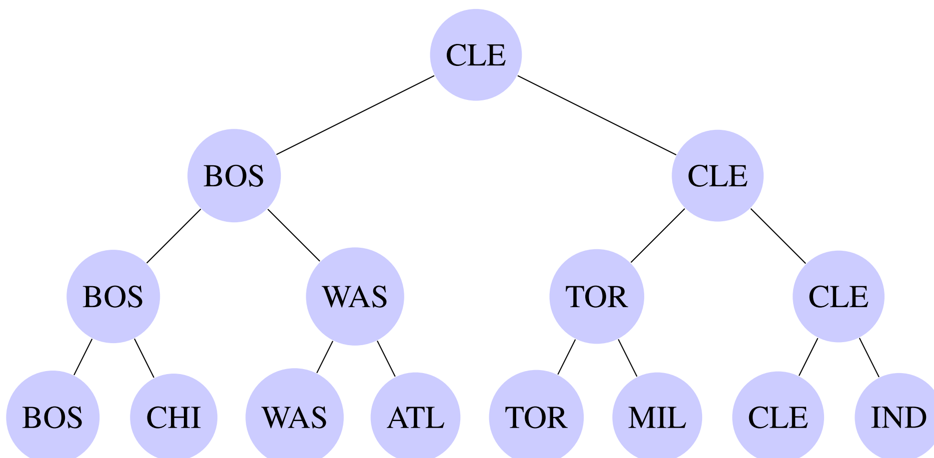


Figure 4.6: This figure represents the winning team bracket for the eastern conference playoffs.

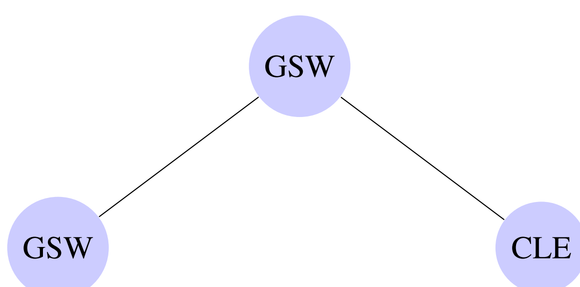


Figure 4.7: This figure represents the brackets for the winner of the easter and western conference to decide the NBA championship.

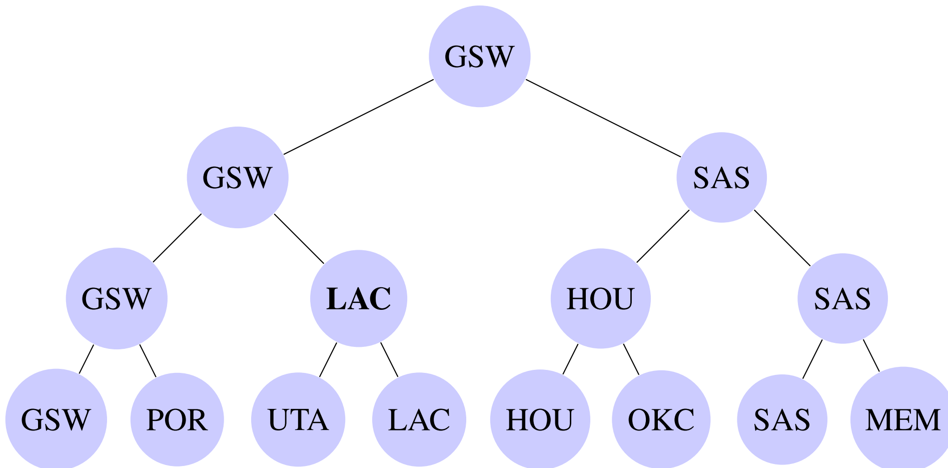


Figure 4.8: This figure represents the predicted brackets for the western conference playoffs using the Elo algorithm.

4.2.3.2 Bracket score prediction using the standard Elo algorithm

Algorithm 1 is used over the NBA dataset for the 2016-17 season and the predicted team ratings are obtained at the end of the regular season. This is shown in Table 4.8. The seedings of teams are obtained from the actual data. The rating obtained is used to determine the brackets in the playoffs. The predicted brackets are shown in Figure 4.8,4.9 and 4.10. The bolded teams represents the mismatch with the actual brackets.

4.2.3.3 Bracket score prediction using the proposed algorithm

The procedure discussed in Section 4.2.3.2 is repeated with the proposed algorithm. The team rating is obtained from the proposed algorithm. This team rating is used to compute the bracket. Then, the bracket score is obtained according to Table 4.4. The predicted brackets is shown in Fig 4.11,4.12 and 4.13. The bracket score comes out to be 29.

4.3 Player Performance

The proposed algorithm offers the advantage of providing information about each individual player and how their performance varies as the tournament progresses. Figure 4.11 shows the estimated rating variation of LeBron James over the season 2016-17. Additionally, this allows the

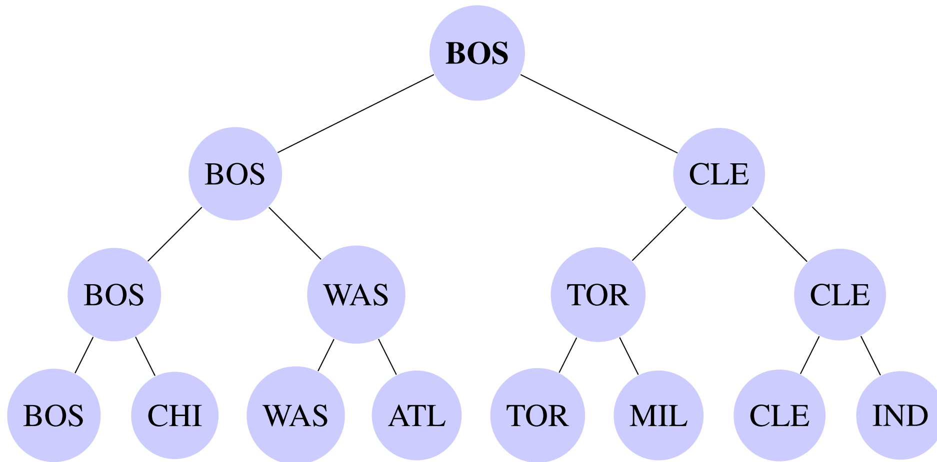


Figure 4.9: The figure represents the predicted brackets for the eastern conference playoffs using the Elo algorithm.

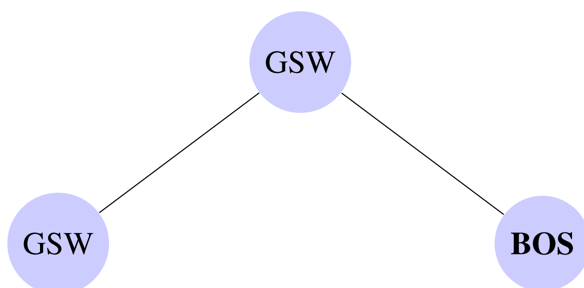


Figure 4.10: The figure represents the predicted brackets for the winner of the easter and western conference to decide the NBA championship using the elo algorithm.

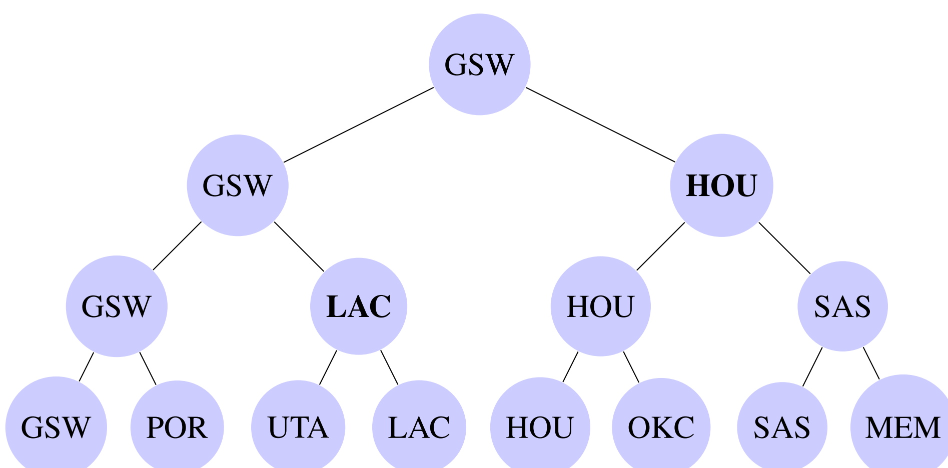


Figure 4.11: This figure represents the predicted bracket for the western conference playoffs using the proposed algorithm.

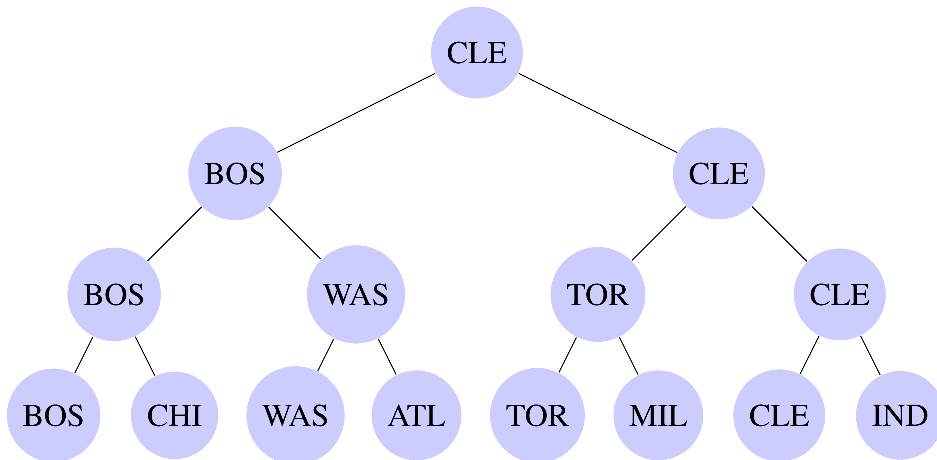


Figure 4.12: This figure represents the predicted brackets for the easter conference playoffs using the proposed algorithm.

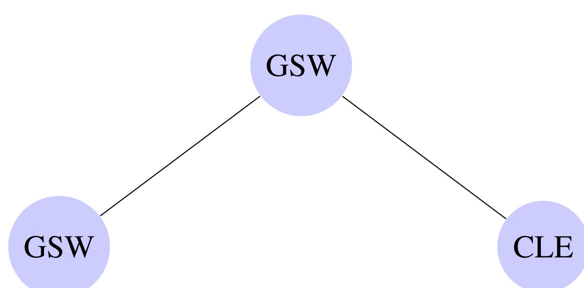


Figure 4.13: The predicted brackets for the winner of the easter and western conference to decide the NBA championship using the proposed algorithm

Team	Team Rating from elo algorithm	Team Rating from the proposed algorithm
ATL	988	1188
BOS	1111	1112
BKN	855	1140
CHA	930	1059
CHI	1010	1138
CLE	1008	1329
DAL	931	1265
DEN	1042	1146
DET	918	982
GSW	1244	1327
HOU	1097	1312
IND	1019	1177
LAC	1108	1157
LAL	890	1149
MEM	956	1144
MIA	1071	1120
MIL	1031	1168
MIN	915	1074
NOP	979	1390
NYK	879	1036
OKC	1057	1193
ORL	886	1154
PHI	846	1242
PHX	824	1092
POR	1072	1221
SAC	923	1216
SAS	1131	1116
TOR	1100	1314
UTA	1118	1145
WAS	1068	1140

Table 4.8: This table contains the team ratings estimated at the end of the regular season using elo algorithm and proposed algorithms.

ability to compare the performance of two different players which are not easily captured by the point based metrics. The average point scored per game (PTS) by LeBron James in the 2016-17 season is 26.4, while that of Draymond Green is 10.2[13]. The PTS metric fails to consider the impact that Draymond Green brings to the team through his defense and play making skills. But the proposed algorithm with the help of the +/- score has the ability to track a player performance.

It can be seen that from Fig. 4.12 that there are instances where Draymond Green has a higher rating than Lebron James.

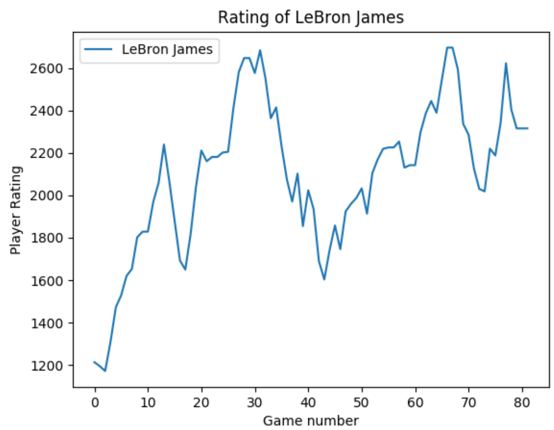


Figure 4.14: The plot represents the variation of LeBron James's Player Rating according to the proposed algorithm.

Additionally, the proposed algorithm can also measure the impact that a player has on the team by removing their contribution from the team and running the simulation. In Fig. 4.13, we can see the impact of Stephen Curry in Golden State Warriors success.

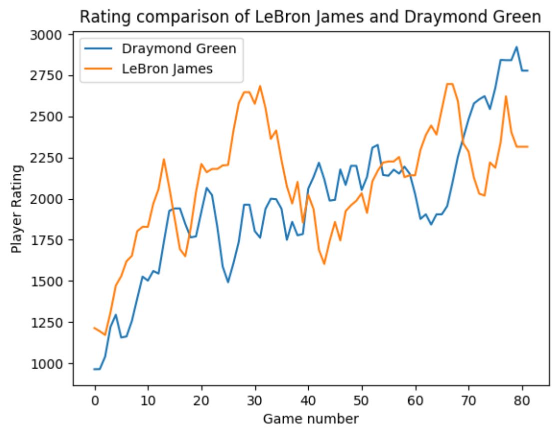


Figure 4.15: The plot represents the comparison of LeBron James and Draymond Green's player rating obtained from the propsed algorithm.

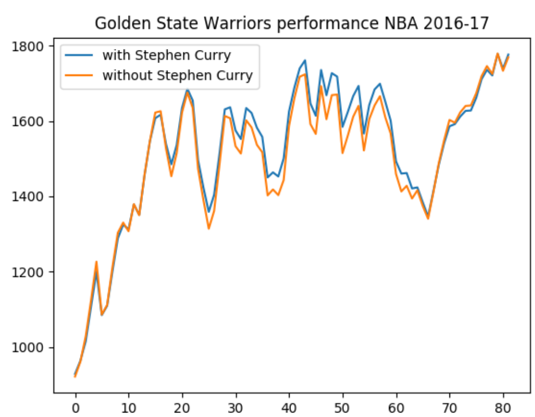


Figure 4.16: This graph represents the comparison of Golden State Warriors perfromance with and without Stephen Curry using the proposed algorihm.

5. SUMMARY AND CONCLUSIONS

The Elo algorithm is a popular rating system, not just because it offers higher prediciton accuracy, but it is also simple in nature. The Elo algorithm does not venture into the dynamics of teams but rather consider the team as the fundamental units. This encompasses a lot of variation that might arise from the variance caused by each individual player. This gives rise to simplicity in the model. In this work, a variation of the Elo algorithm is used where the performance of individual basketball players is modelled using the +/- metric. Individual player ratings are combined to get the team rating and to predict the outcome of games. The proposed algorithm is compared with the standard Elo algorithm by measuring the perfromance of the algorithms over synthetic generated and real-life data. From, the above numerical simulations, it can be seen that the standard Elo algorithm outperforms the proposed algorithm. This is due to the fact that, by considering the individual players's rating, the complexity of the proposed algorithm is increased. The principle of Occam's Razor states that a simpler model is always preferred until the data justifies the use of more complex models. The proposed algorithm has a comparable performance in predicting the outcome of games in comparison to the Elo algorithm. Additionally, this algorithm has the ability to offer insights into player strengths. This enables the analysis of injuries and mid-season transfers, which the Elo algorithm is ignorant about. This information is valuable for the team managment to quantify the worth of a player to the team and can help make decision with regards to trading player.

REFERENCES

[1] "Nba statistics." Available at `http://stats.nba.com/search/player-game/#`.

[2] N. Silver, "2017 predictions." Available at `https://projects.fivethirtyeight.com/2017-nba-predictions`, 2017.

[3] A. E. Elo, *The rating of chessplayers, past and present*. Arco Pub., 1978.

[4] H. A. David, *The method of paired comparisons*, vol. 12. London, 1963.

[5] L. L. Thurstone, "A law of comparative judgment.," *Psychological review*, vol. 34, no. 4, p. 273, 1927.

[6] F. Mosteller, "Remarks on the method of paired comparisons: I. the least squares solution assuming equal standard deviations and equal correlations," in *Selected Papers of Frederick Mosteller*, pp. 157–162, Springer, 2006.

[7] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[8] H. Stern, "Are all linear paired comparison models empirically equivalent?," *Mathematical Social Sciences*, vol. 23, no. 1, pp. 103–117, 1992.

[9] K. S. Courneya and A. V. Carron, "The home advantage in sport competitions: A literature review," *Journal of Sport and Exercise Psychology*, vol. 14, no. 1, pp. 13–27, 1992.

[10] K. Massey, "Statistical models applied to the rating of sports teams," *Bluefield College*, 1997.

[11] M. E. Glickman, J. Hennessy, and A. Bent, "A comparison of rating systems for competitive womens beach volleyball."

[12] "NCAA tournament bracket." Available at `https://www.foxsports.com/fantasy/collegebasketball/tourney/help/rules/`, 2017.

[13] "basketballreference." Available at `https://www.basketball-reference.com/players/`.