Predictive Modeling on 2017 NBA Playoffs

Qingyuan, Zhang   Jihan, Wei   Ruxue, Peng   Zixuan, Guan   Yue, Jin

April 27, 2017

1 Introduction

The Playoffs champion goes to...

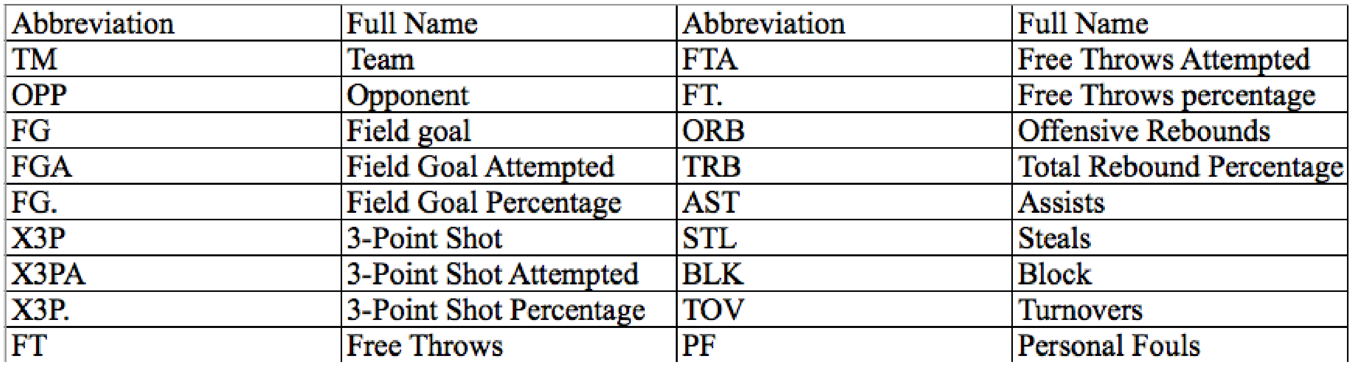
NBA 2015-2016 season was unforgettable. The Golden State Warriors 73-9 made history. For the playoffs, OKC ended their journey even they got 3-0 over GSW at first. Then the same situation happened again to GSW, at the finals, GSW won over CLE by 3-0, but they dramatically lose the following 4 games.

This is why NBA is so enchanting. You can never predict what the game would be like till the last minutes.

Then why this project? Because we want to train our machine on helping us make predictions without subjective opinions.

2 Data

The data used for this project is mainly scrapped from basketball-reference.com and stats.nba.com. Among all kinds of measurement for a player, for a game, or for a season, we chose to use these key features as measurement for the performance of a team during one game played.



Abbreviation	Full Name	Abbreviation	Full Name
TM	Team	FTA	Free Throws Attempted
OPP	Opponent	FT.	Free Throws percentage
FG	Field goal	ORB	Offensive Rebounds
FGA	Field Goal Attempted	TRB	Total Rebound Percentage
FG.	Field Goal Percentage	AST	Assists
X3P	3-Point Shot	STL	Steals
X3PA	3-Point Shot Attempted	BLK	Block
X3P.	3-Point Shot Percentage	TOV	Turnovers
FT	Free Throws	PF	Personal Fouls

Besides above features gained directly from game logs, we add two more features into the model. One is called Efficiency, the other is ELO rating.

- EFF Feature

The NBA publishes online all the basic statistics recorded officially by the league. Individual player efficiency is expressed there by a stat referred to as 'efficiency' and abbreviated EFF. It is derived by a simple formula below: NBA Efficiency recap = ((Points + Rebounds + Assists + Steals + Blocks) - ((Field goals attempts - Field goals made) + (Free throws attempts - Free throws made) + Turnovers)) To construct the feature, we found out for each game, the player lineup of the two teams and their minutes played in that game, then query their efficiency score of the corresponding game season, at last we average over players' EFF score for the two team.

By incorporating the players' efficiency score, we took into consideration the influence of change of player at each game within our model.

3 ELO Algorithm

ELO - The algorithm made famous by Facebook & depicted in the movie *Social Network*

- Basic Chess Algorithm proposed by Arpad Emrick Elo:

$${}_nR_i ={}_o R_i + K(S_{ij} - \mu_{ij})$$

Where

$$\mu_{ij} = \frac{1}{1 + 10^{({}_oR_i - {}_oR_j)/400}}$$

is the expected result, K factor is adjusted for each domain. For Chess, K = 10; for soccer it varies from 20 to 60. $S_{ij} = 1$ or $1/2$ or 0.

- ELO Algorithm adjusted to basketball game:

  For chess game, no score needed to be considered except Win, Lose or Draw; but ball games have scores that need to be accommodated.

  In this project, ELO rating for each team is updated per game using following formula:

$${}_nR_i ={}_o R_i + K * M_{ov}(S_{ij} - \mu_{ij})$$

where

$$M_{ov} = \frac{(abs(PD) + 3)^{0.8}}{7.5 + 0.006 * (elo\_diff)}$$

In this algorithm, K factor takes value 20.

$$elo\_diff = (abs(R_i - R_j) + 100 * home\_win) * won\_underdog$$

where $home\_win = 1$ if the home team won the game, 0 otherwise. $won\_underdog$ takes value of 1 if the team got expected outcome as suggested by the ELO rank, -1 otherwise.

By using the ELO rating system, we can also calculate one team's probability of winning:

$$Pr(A) = \frac{1}{10^{-elo\_diff/400} + 1}$$

4 Model Selection

At the very beginning, we tried various models and picked the model with the highest prediction accuracy, which is XGBoost. The steps to train and test each model are as following:

1. We picked the 2015 Regular Season data and 2016 Regular Season data from www.basketball-reference.com/ as training and testing data respectively.
2. We trained the model based on the 2015 Regular Season data, with parameters selected from 10-fold cross-validation. For training the model, we treated each game as one observation. And for each game, we used the performance of both teams, say Team A and Team B, and other game related data (such as the players' information and the weather the game is played in Team A's home field) as features.
3. Based on our model, we predicted the outcome of each game on 2016 Regular Season. For each team, we found all the historical games it played and used weighted average of the history data as the measure of performance of one team. For predicting the results for each game, knowing which two teams will play, we combined the historical performance of the two teams and added other information as test features. Once a game finished, we updated the historical performance of both participated teams, and continued the process until the last game.
4. Following the above procedure, we finally found the XGBoost had the best prediction accuracy, which was 70.2%.

5 XGBoost

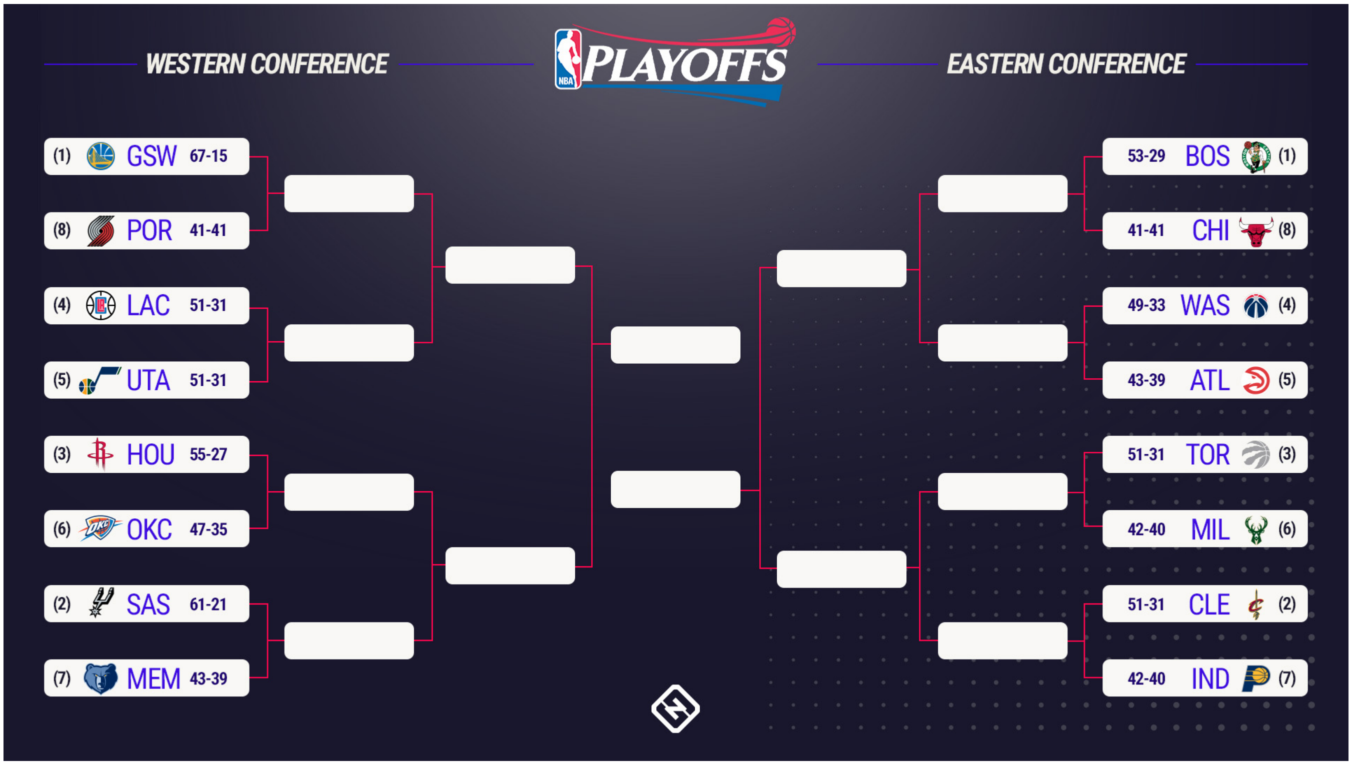
We take the following steps to train and fit XGBoost model to our data:

1. For predicting the champion for the 2017 playoff, we retrained the XGBoost Model, on the 2015 Regular Season, 2016 Playoff, and 2016 Regular Season data, with the best parameters (depth, eta) selected from cross-validation.

2. After training the model, we could predict the wining probability for Team A in each game as long as we have the information for that game.
3. Cumulatively add all the data available before the date of each test data point to the test data, and use dynamic moving average of the history data as test features
4. Transform the data to the desired form for xgboost model
5. Select the best parameters (depth, eta) of the XGBoost model through cross validation
6. Train the XGBoost model using the best parameters to the entire dataset and obtain predictions for this season

In more details, for 2016-2017 season playoffs predictions, we train the XGBoost model using xxx datasets, and use weighted average of most recent 82 games as input of the model (except for features home-away, EFF and ELO).

6 Simulation



The first round of the NBA playoffs, or conference quarterfinals, consists of four match-ups in each conference based on the seedings (1–8, 2–7, 3–6, and 4–5). The four winners advance to the second round, or conference semifinals, with a match-up between the 1–8 and 4–5 winners and a match-up between the 2–7 and 3–6 winners. The two winners advance to the third round, or conference finals. The winner from each conference will advance to the final round, or the NBA Finals.

According to the NBA rules, all rounds are *best-of-seven* series. Series are played in a 2-2-1-1-1 format, meaning the team with home-court advantage hosts games 1, 2, 5 and 7, while their opponent hosts games 3, 4, and 6, with games 5-7 being played if needed.

Based on our model, given two teams Team A and Team B, we will have two probabilities based on whether Team A plays in the home filed. According to the schedule and playoff rules, we simulated the result for each game and predicted the winners for each layer, from the first round: 8 teams in both Western and Eastern Conference, to the final layer: only 1 team left in both Conference and fight for the championship.

Importantly, in each layer, we can update our features and wining probabilities by incorporating the most up-to-date data. And as the real games go on, we can correct the winner of each round and continue the simulation process.

7 Our Predictions

Computed from our model, the western conference champion goes to:

WESTERN
ONFERENCE
WESTERN
RRIORS

The eastern conference champion goes to:

CAVALIERS
23

And the champion for the Finals goes to:

YOU KNOW WHO :-)