Project Report

A Robust And Explainable Way To Predict NBA Game Results

Ziwei Wei

Abstract

The current paper presents a more robust and explainable way to predict NBA game results. Using the abstracted team composed by on-court player stats, the model will be able to generalize across seasons and perform consistently even if the team changes by a player injured in season, leaving the team, and joining the team achieving the same accuracy in prediction as to the current best accuracy rate. It can also generate insight about what is more important in the current NBA games.

Background

Statistics and data have become an ever more attractive component in professional sports in recent years, particularly with respect to the National Basketball Association (NBA). Massive amounts of data are collected on each of the teams in the NBA, from the number of wins and losses of each team to the number of field goals and three-point shots of each player to the average number of minutes each player plays.[1] But the lack of advanced data source hinders the progress of prediction accuracy, from my research the highest prediction results come from students in stanford[1] or in Wisconsin[3] which can predict from 66% to 68% which is a little bit better than simple history-based prediction. However, we also saw others claim they have achieved higher results like 80%[2] but using invalid methods such as using played game statistics to predict a game result which is not possible in real life prediction and is a

totally wrong way to train a predictor. By using the played game data 2P, 3P, we can easily achieve 100% accuracy without any machine learning method. From my perspective, we may need more data such as detailed one-on-one performance, detailed game match-up during different intervals, detailed player fatigue and body measurement between games, etc. in order to achieve better results.

Introduction

My proposal is a new way of predicting NBA game results using currently available game data. Instead of using the average team performance to predict the game results like what other papers did[1][2][3], it will be more realistic to use avg on-court player performance in 5 positions(PG, SG, SF, PF, C) to represent the team to achieve more useable prediction for real games and exclude player match-up and injury's negative influence on average team data. In this way, we can also use cross season data as a whole for training, since we have recorded the game as the match of two abstract teams using current players and their playtime. There will be no need to train the model within one season for validity like other papers have been doing[1][3]. We can actually train a few recent years together to predict the future season. Therefore we can have a more robust and reusable model for real-life prediction.

The work will be comprised of 2 parts. First, we need to construct the abstract team and match between them using the data scraped from basketball-reference.com. The second, We will run Ridge Regression, SVR, XGBoost on them and test the final accuracy.

Data Preparation

3 data scrappers are built to scrape data in 3 domains from the basketball-reference.com. These 3 data sets are boxscore(every in-game player stats in detail), players(season total stats), and schedule(match-ups ordered by time). In the end, I scraped data in the recent 5 years(2015-2019). There are 6150 games in total during the 5 years without the play-offs which are intrinsically different from a normal game in NBA season.

In the box score data set, there are 6150 games with home and away players data in the game including Players, MP, FG, FG%, 3P, 3PA, 3P%, FT, FTA, FT%, ORB, DRB, TRB, AST, STL, BLK, TOV, PF, PTS, +/-. In the players' data set, there are players stat from each year including Height, Weight, Pos, MP, 3P, 3PA, 2P, 2PA, FT, FTA, ORB, DRB, AST, STL, BLK, TOV, PF, PTS. In the schedule data set, there are match date, Home/Neutral, Visitor/Neutral, Difference. All 3 data sets will be combined into the abstract game data set which will be explained in the next chapter. The final data set will exclude features like FG, FG%, 3P%, FT% which is very closely correlated to 2P, 2PA, 3P, 3PA, FT, FTA with relationships like FG = 2P + 3P, FG% = (2P + 3P)/(2PA + 3PA), 3P% = 3P/3PA, FT% = FT/FTA.

Model Overview

The abstract team and match for training will be generated for each game. In each game, each team will be represented by 5 abstract players from 5 different positions C, PF, SF, SG, PG. Each position is a time-weighted sum of all players with the same position. It is comprised of by Height, Weight, MP(minutes played), 2P(2 pointers), 2PA(2 pointers attempted), 3P(3 pointers), 3PA(3 pointers attempted), FT(free throws), FTA(free throws attempted), ORB(offensive rebound), DRB(defensive rebound), AST(assist), STL(steal), BLK(block), TOV(turnover), PF(personal foul). Then we can construct each match as 5 abstract players vs 5 abstract players. So the final dataset for training will be like home_C, home_PF, home_SF, home_SG, home_PG, away_C, away_PF, away_SF, away_SG, away_PG, game_score_difference. All these data should use the player's season average performance to build. So that if we want to predict a new game, what we need is estimated playtime or coach designed playtime for each player and their average performance over a period of time. Using this data we can train years of games without worrying about season and injury's influence upon our model thus we can achieve a more robust model. Also, this model will also be easier to be explained since a team is not a team, a team is actually a bunch of players on-court.

Final training data after feature selection and reconstruction:
home_C meaning all players played in C position for the home team in this game's average season stats weighted by playing time which is a good prediction for performance in that position.

	home_C _Height	home_C _Weight	home_C _MP	home_C _2P	...	away_PG _TOV	away_PG _PF	away_PG _PTS	diff
0	84.129116	252.9987	38.46666	4.37665	...	3.468103	4.076216	13.7795	17.0
1	82.628819	253.7118	48.00000	8.57355	...	3.829040	5.485043	22.8159	1.0
2	82.466667	243.1666	30.50000	5.20856	...	1.956260	3.887881	13.4459	-18.0
3	82.657269	272.8969	60.30000	12.1151	...	1.101233	1.349631	4.95140	2.0
...	...	...	...	...	...	...	...	...	...
6146	83.711297	249.3305	55.76666	11.9268	...	3.980422	7.239754	33.4016	11.0
6147	83.718024	248.7703	50.95000	9.94501	...	1.032393	1.713333	10.2800	4.0
6148	81.943017	240.0000	50.60000	11.3091	...	0.000000	0.000000	0.00000	6.0
6149	83.758797	253.2463	55.41666	5.39597	...	4.048626	4.459583	28.5021	5.0

Training Results

The final model is trained and tested upon the abstract game data set containing 6150 games from 2015 to 2019 explained in the last chapter using 3 different methods. K-fold cross-validation with shuffling and a random state on is used in all training processes where K = 5, meaning there are roughly 4 seasons about 4920 games is trained, and 1 season about 1230 games is used for testing. The final accuracy will be calculated from the average of all iterations.

Model	SVM (SVR)	Linear Regression (Ridge Regression)	Decision tree (XGBoost)
Training accuracy	69.2%	64%	70%
Testing accuracy	67.2%	63.5%	60%

In all 3 machine learning method, the SVM performed the best overall reached the same accuracy that previous papers have achieved[1][2]. In the SVR model, I used SVM for regression. The final results are the best which has the training accuracy of 69.2%, testing accuracy of 67.2%. In the Ridge Regression model, the final results are not as good as in the Stanford example[1]. It reached 64% in training, 63.5% in testing. In the XGBoost tree model, the final results show that the decision tree did not perform well enough on the data set with 70% in training, 60% in testing.

Insight From Results

Using the customized data sets like this, we can gain insights about the recent 5 years game, and see what position or stats is more important for a modern basketball game by taking the coefficient of the linear regression model. I took the most important coefficient as following.

For the home team the top 3 important stats which have positive effects on a game are
('home_PG_3P', 0.5422934714572395),
('home_SF_3P', 0.4257312772880185),
('home_PF_AST', 0.4542551564125257).
For the home team the top 3 important stats which have negative effects on a game are
('home_PG_TOV', -0.3510520954006565),
('home_SG_TOV', -0.2783324117535572),
('home_SF_PF', -0.2701965691642854).

Since we used games from the last 5 seasons, we can clearly see how pointer guard and 3 pointers have changed the game. The game-winning rate depends mostly on the pointer guards' 3P and turnover. Other positions' 3P like SF also have huge effects on the game. Whereas PF's assist is quite a surprisingly important factor. Although these factors may be wrong, they provide a fresh perspective about what is more important in the modern NBA game.

Conclusion

The prediction model's accuracy after all these tests from various teams[1][2][3] is quite limited with various models used. I think the main reason for the limited accuracy is the data granularity/details and the lack of key player performance data in both physical and decision making. For example, a good way to predict will a shot goes in will need data like the distance from the defender to the shooter, the speed change of the shooter, and how the shooter normally performs under such condition. However, current NBA data can only provide basic stats in a whole match in a very limited field from the data it provided, there is no way to reconstruct the game itself. In order to get higher accuracy, we need better data about the game and every round in the game in detail with timestamp and defense assignment(who guards who).

Teams	My results	Omid Aryan& Ali Reza Sharafat
Training accuracy	69.2%	67.7%
Testing accuracy	67.2%	68%

Comparison with Omid Aryan&Ali Reza Sharafat's results from Stanford

Although my current model did make a breakthrough in the accuracy measurement, it provides a more universal framework for NBA prediction. This model can be actually used in real-life scenarios since it can generalize with the effect of injury, trade, and coach decision about a player's playing time. It can be easily trained across seasons to learn more data which opens up the opportunity to use more sophisticated and modern machine learning models on it. Furthermore, with such explicit training data, we can also explain models easily and extract insight into the current games in the NBA could potentially help managers and coaches adjust their squad and trade targets.

Future Work

This project can be developed further adding a temporal player status into the framework such as recent performance to better help the model to predict in real life. Also more advanced machine learning models such as ANN, CNN, etc. can also be tested upon using these abstract games. By using better stats explained in the previous chapter, we can achieve better results in the accuracy.

Citation:

[1]A Novel Approach to Predicting the Results of NBA Matches.
http://cs229.stanford.edu/proj2014/Omid%20Aryan,%20Ali%20Reza%20Sharafat,%20A%20Novel%20Approach%20to%20Predicting%20the%20Results%20of%20NBA%20Matches.pdf
[2]NBA Game Result Prediction Using Feature Analysis and Machine Learning.
https://fadifayez.com/wp-content/uploads/2019/03/NBA-Game-Result-Prediction-Using-Feature-Analysis-and-Machine-Learning.pdf
[3]Prediction of NBA games based on Machine Learning Methods.
https://homepages.cae.wisc.edu/~ece539/fall13/project/AmorimTorres_rpt.pdf