NBA Game Result Prediction based on Individual Player

JunYi Jiang; Lyu Bo; Yu Zezhong

Abstract

This report contains a project focusing on predicting the result of a basketball game with the players' statistics as feature. The report will focus on how to produce a higher accuracy with different feature selections. Finally, the report also contains some experiments on real world situation based on the experiment result.

1 Introduction

NBA is an American Professional basketball league consist of 30 teams and, during the regular season, each team playing 82 games, 41 at home and 41 away. The problem that is being explored in this paper is how accurately can the result of a basketball game be predicted. Sports prediction has been widely studied by different groups such as the team manager or the sports betting industry. After looking into many studies of predicting game results based on passed records of the team, we would like to explore the features as starters' statistics, and we would like to pick out the most significant and influential statistics to the test accuracy.

We would like to treat the ten starters from both team in a game as our features. Their career statistics would be aligned and input into various machine learning algorithm to predict a result of the game, either the away team losing ('0') the game or winning ('1'). Also, to check if our experiment is not trivial, we will use their passed records as our benchmark accuracy. We expected to find an algorithm that produces a relatively high accuracy and then could be used for later experiments.

Then, various combinations of different summary statistics would be explored and we expected to see some combinations may be producing a lower accuracy while some would be able to produce a higher accuracy. Finally, we would like to explore the flexibility and applicability of our model. Since we are not building the result upon passed records of a whole team but upon the career performance of players actually on the court, we would be able to switch players on a team and see the difference and therefore exploring how the existence of or the absence of one player could affect the game results and answering some questions that could not be realized in the real world.

2 Problem Definition

The main goal is to predict a basketball game result as accurate as possible. Many papers online have been exploring methods with the passed record of the team. To put them in a very simple term, if the team won more in the past, it would have a higher probability of winning. However, we think the model oversimplified the basketball game. From seasons to seasons, a lot of different aspects of a team will change. There might be trades that completely change the construction of a team. For example, LeBron James, currently the best player in the league, leaving Cleveland has caused the last year's NBA Playoffs Final contender to slide down to the 13th in Eastern Conference (15 teams in each Conference) right now. therefore, we decided to use the career statistics of the ten starters of the game to predict its result. First of all, we choose the starters because they typically play the most minutes in a game and they can best characterize the team. We expect to have a higher accuracy than predicting on past records since the players on the court are the more determinant factors to the game results. And we could then explore many different sets of summary statistics of players and compare the accuracy. One very important experiment that can be carried out with our model is that because we are

using each player's statistics, we can switch a player on the team with another player to see the influence on game result, which makes our study very applicable in the real world. We can switch to players that are currently playing to analyze possible influence of specific trades in the league or to retired players to see their influence in nowadays' basketball games

3 Methodology

The Datasets we used came from https://www.basketball-reference.com. We have taken data from the website and transformed the data of past five years into two dataframes. One showing the games' outcome and starters' names, and the other one showing the career statistics of each player. For all the algorithms, data of games of the recent 5 years are being used, and a random subset of 30% of the whole feature set and label set will be used as the test set and the remaining will be used as the training set.

We started by setting a benchmark for the experiments. Using the game dataframe, the past records of a team can be extracted out as features. The input consists of 30 Boolean features representing 30 NBA teams. For each input, two teams will be set as 'on' (set as '1'). We also added an indicator to see if the first team appeared in the 30 Boolean features that is 'on' is an away team or not. So, the input for the model will 31 Boolean features and the labels would be whether the away team won the game ('1') or not ('0').

Many papers use SVM as a start to analyze their models. Since SVM is a robust classifier that maximize the separating hyperplane margin to find an optimal solution, we chose SVM with a random state of 42 to classifier these data. The accuracy will be treated as a baseline.

Then we started to create features based on players and test various classifiers on them. The feature matrix for each game would be going into the first game dataframe and take the starters and find the match in the player dataframe. The feature matrix has ten rows corresponding to the ten starters from the home team and the away team. In each row, each column will represent career statistics of the player. Initially, we used 12 basic statistics to summarize a basketball player into a feature that influence the game result. The 12 statistics are Field Goal % (FG%), effective FG% (eFG%), Free Throw % (FT%), Offensive Rebounds (ORB), Defensive Rebounds (DRB), Total Rebounds (TRB), Assists (AST), Steals (STL), Blocks (BLK), Turnovers (TOV), Personal Fouls (PF), Points (PTS). eFG% adjusts the FG% for the fact that a 3-pointer is worth more than a 2-pointer. These are very basic statistics to evaluate a basketball player's performance. The labels are still whether the away team won the game ('1') or not ('0').

Again, to best compare the result, SVM are chosen to evaluate this model. And decision trees with different maximum depths will be used as another classifier. Finally, we would like to explore an ensemble method, bagging. We would randomly choose bootstraps with replacement from feature set and train them on the decision tree that, among those different maximum depths, achieves the highest accuracy.

For the baseline, we expected to have an accuracy around 55% because the past records can only represent a team's performance with limitations. And we expected to see accuracy to rise above 60% as the players' statistics were used as the feature for the reason of a better summary of a team's performance. For the different algorithms, we expect to see SVM and Decision Tree to have a similar final accuracy while since bagging ensembles, it should give out a higher accuracy of all classifiers.

Finally, we conduct features selection to compare the influences of different features and whether if the machine can learn features by itself. we use stepwise features selection method including forward search and backward search as well as heuristic method. Our expectation is that the algorithm itself will already be able to select features that do more help for the prediction, given the fact that we are not using a lot of features comparing to the training set size.

4 Results

Using a team's past record as features and SVM as classifier yielded a higher accuracy than expected. So, we are setting our baseline at 58%. As expected, with a more expressive feature set representing the ability of players actually playing on the court, SVM classifier produces a

higher test accuracy while also producing a much higher train accuracy.

And as expected, Decision Tree produces a similar test accuracy as SVM does and to show that the increase of expressivity of the model on the training set, a decision tree with max depth is also implemented.

Finally, among different depths, Decision Tree with maximum depth of 6 produces the highest test accuracy and therefore is used as the base estimator for the bagging algorithm. The bagging algorithm produces the highest test accuracy of all classifiers and would be used in later experiments for the real world.

Algorithms	Train Accuracy	**Test Accuracy**
Baseline SVM	0.5807	0.5827
SVM	0.9587	0.6038
Decision Tree (Max Depth)	1.0	0.5675
Decision Tree (Depth = 6)	0.6899	0.6043
Bagging (Base=(DT, Depth = 6))	0.7821	0.6385

Table 1. Performance of Different Algorithms

In the feature selection session, as for forward search, each time add one best feature to the previous result; as for backward search, each time delete one worst feature from the previous result. And the final five features selected by three methods are shown below:

Forward Search	Backward Search	Heuristic
DRB	3P	FG%
STL	3PA	eFG%
FT%	PTS	FT%
3P	FTA	TRB
TOV	FGA	AST

Table 2. Features selected by different methods

The features selected by these three different methods were almost the same, and when the number of features is greater than 3, the performance would not increase significantly, in other words, using 3 features of ten players can generate almost the same accuracy as using all the features. This indicates that the 10 players itself contains the information to predict the outcome, and too much features cannot increase the accuracy of the algorithms.

5 Discussion

Our baseline SVM gives a higher test accuracy than our expectations. Also, looking at the train accuracy, we conclude that there is not a strong evidence for overfitting or underfitting. With a relatively simple feature structure, the model is not too expressive on the training set. Also, since we are looking at the team's record for the past five years, the main strategy or the core players for each team haven't had very dramatic changes and therefore the SVM could learn the pattern and produce an accuracy that is close to 60 percent. However, 58 percent is just a bit higher than a random guess, we are hoping for a classifier that can generalize better.

After formatting the player statistics as features, the accuracy increases as expected. SVM and Decision Tree produce an accuracy that is over 60 percent. One thing that is worth noticing is that with the increase in the test accuracy, the accuracy of training set also significantly increases. This is because the more complicated and more expressive feature set makes the classifier learn a less generalizing model. To give an extreme case, a Decision Tree with maximum depth is implemented and it gives a perfect result on the training set while only give a 57 percent test accuracy.

Then, we used a Decision Tree with depth 6 to produce a bagging classifier and it, as expected, produce the highest test accuracy. This is because bagging ensembles classifiers and therefore grows a more power classifier to predict. Since there are different classifiers voting for the final result in bagging, the classifier could learn the players' performance and their influence to the game result better, and therefore, also bringing a larger train accuracy then just using one Decision tree with depth 6.

The feature selection kind of match our expectations by showing that SVM is already selecting the important features to predicting the results by not having a significant upgrade in accuracy. SVM is the algorithm that find the best separator for the data points and maximize the margin (the performance of the separator). Therefore, SVM would be able to learn which features would take more weights during training. Also, as explained in the expectation part, since we only have 12 features for each data point to start with, given the much larger training size, it wouldn't be too hard for the algorithm to discriminate which feature is more or less important.

Finally, regarding to the fact that fewer features are generating almost the same accuracy as the original feature matrix. There can be many different approaches. For example, one can conclude that predicting game result alone without many other effects is itself not a correct approach. Career data maybe not generalized enough for the algorithm to provide a better accuracy. On the other hands, one can also argue that data could be too representative that if you simply added players' who scored a lot and assisted a lot in their careers, it will be a very strong team. In fact, if you take five players that had a very strong career performance on their stats sheets, it will form a team that would be believed to be very hard to defeat. Therefore, many future works could be done to improve the experiment, for instance more sophisticated summary statistics or more feature aspects.

6 Evaluation

We would like to use this section to identify some of our evaluations and some improvements that could be applied to our project.

There are several more things that can be done on the data set and the features to improve our model. First of all, we are using a player's career statistics as feature. However, for some players at almost the end of their career while still playing as a starter, for example, Kobe Bryant, can have a very big gap between, his career statistics and actual performance on the court. Also, there may be some sudden slide down of and some step up of statistics for a particular season and therefore although they didn't play much before and had bad career statistics, they are in the starter and playing a significant role on the court for that season. Therefore, using the career statistics may not be correctly generalizing a player's performance on the court. To solve this problem, for each game we are training or test on, we could use the starters statistics from a previous season. Our problem is that it is very hard to obtain the data set. Since we pulled all the data from the Internet, and getting data for player statistics for each season requires a more advance technique and large amount of time. Also, there is also a problem that some rookies also play important role on the court while they do not have statistics for the previous season. Therefore, if we were using precious season statistics, we would be having some missing data points. And solving the missing data problem is not easy since there will be a lot of estimations for each different categories of player statistics.

Also, the reason that we are not achieving a higher accuracy could be coming from that we are only using basic statistics. There are some more advance statistics that may be able to generalize a player's performance better, for example, Player Efficiency Rating, which measures per-minute production, Win Share. Finally, our data set could also be improved to have position taken into account. Since our starter lineup dataframe doesn't specify each starter's position and sometimes have a pretty random order, adding the position for the classifier might help the classifier to understand more that maybe rebounds performance of a shooting guard is not very important and it is normal for a center to have a very high FG%.

Finally, Centering or some dimensional reduction could be applied to our feature matrix. However, since each statistic represents a particular ability of a player, dimensional reduction might not achieve its function while centering is expected to help generalize each statistic category better and therefore train a better classifier.

7 Application

To show the applicability of our model, we started several experiments that can be apply to the real-world situation. We chose the classifier that gives the highest test accuracy, bagging, to carry out the experiment.

Our first experiment is to see how a very legendary player could affect a team. We tested to see how the legendary NBA star Kobe Bryant could contribute to Los Angeles Lakers. We switched all different shooting guards of Lakers to Kobe. With our model, Lakers could have gotten 48 wins, 34 lost with Kobe Bryant starting 78 games in 2016-2017 season while they actually got 26 wins, 56 lost in that season. We think this is a very possible outcome. So, our model can be utilized by team managers to analyze the possible outcome of a trade.

Then, we try to test a very interesting topic that has long attracted a lot of attentions in many basketball talks. Michael Jordan has long been recognized as the best basketball player in the history. He and his Chicago Bulls won 72 games in 1995-1996 season and this record was kept for 20 years. We would like to see how this almost unbeatable Chicago Bulls and its starting five would perform nowadays. And our model predicts that Chicago Bulls could get 46 wins, 36 lost in 2016-2017 season with their starting five back in 1995-1996 season. This might still be a wild prediction but at least the model can provide some support to people arguing that modern basketball is played at a faster rhythm and more efficiently than basketball is played in the old days. And a more scientific training and healthier diet for modern athletes will be able to blow up their statistics to affect our model to incline to modern day players. Also, such result can also be viewed as a support of the previous explanation on feature selection. We can say that since there are more to a player and his influence to the game outcome than just numbers on the statistic sheets, simple summary statistics might not be able to represent all the aspects that had the 1995 Chicago Bulls won 72 games back then. And therefore, if some advance data or aspects like coaches or the year or the time a basketball game was played could be added to the experiment, the result might be generated differently.

Finally, we would like to use our model to predict the final ranking of this year's NBA regular season. This is actually first time we are using our own trained model to predict something in the future for this course. Here, we attached a prediction for the final record of Eastern Conference:

By 12/08/2018, the real eastern conference standing (*playoffs):

	Win	Lost	W%
Toronto Raptors*	21	6	0.777778
Philadelphia 76ers*	18	9	0.666667
Milwaukee Bucks*	16	8	0.666667
Indiana Pacers*	16	10	0.615385
Boston Celtics*	15	10	0.600000
Detroit Pistons*	13	10	0.565217
Charlotte Hornets*	12	13	0.480000
Orlando Magic*	12	14	0.461538
Miami Heat	11	14	0.440000
Washington Wizards	11	15	0.423077
Brooklyn Nets	10	18	0.357143
New York Knicks	8	19	0.296296
Atlanta Hawks	6	20	0.230769
Cleveland Cavaliers	6	20	0.230769
Chicago Bulls	6	21	0.222222

Figure 1. Current Eastern Conference Standing

Our model predicts that the final eastern conference standing in 2018-2019 season (*playoffs):

	Win	Lost	W%
Toronto Raptors*	51	31	0.621951
Boston Celtics*	50	32	0.609756
Philadelphia 76ers*	49	33	0.597561
Indiana Pacers*	49	33	0.597561
Washington Wizards*	49	33	0.597561
Detroit Pistons*	48	34	0.585366
Milwaukee Bucks*	47	35	0.573171
Orlando Magic*	42	40	0.512195
Charlotte Hornets	41	41	0.500000
Brooklyn Nets	35	47	0.426829
Cleveland Cavaliers	34	48	0.414634
Miami Heat	31	51	0.378049
Atlanta Hawks	20	62	0.243902
Chicago Bulls	20	62	0.243902
New York Knicks	11	71	0.134146

Figure 2. Predicted Final Standing

References

Lin, Jasper, Logan Short, and Vishnu Sundaresan. "Predicting National Basketball Association Winners." (2014): n. pag. Web.

Torres, Renato Amorim. Prediction of NBA Games Based on Machine Learning Methods. homepages.cae.wisc.edu/~ece539/fall13/project/AmorimTorres_rpt.pdf.

Hoffman, Lori, and Maria Joseph. A Multivariate Statistical Analysis of the NBA.