

Various Machine Learning Approaches to Predicting NBA Score Margins

Grant Avalon, *gavalon*; Batuhan Balci, *bbalci*; and Jesus Guzman, *guzmanj*

INTRODUCTION

A single basketball game contains many metrics that can be used to predict which team will win. Our goal in this project is to predict the outcome of a National Basketball Association (NBA) game using box score statistics. We use various machine learning techniques to predict game scores and to understand which features of a team make it successful. We want to figure out which attributes or stats of a team will be critical for winning against the other team.

We employed linear regression, gaussian discriminant analysis, principal component analysis coupled with support vector machines, random forest and adaptive boosting. Once trained, the input to our models is the name of two teams in the NBA, and the output is the point margin between these two teams. Our training data, includes team statistics along with point margins of different games of one NBA season. By point margin, we mean the difference between the home team score and the away team score. For one specific case, namely gaussian discriminant analysis, we decided to analyze not the point margin, but the winner/loser of the matchup. This will be detailed in the later sections.

RELATED WORK

Much research has attempted to model NBA game results and simulate games in an effort to understand what makes a winning team. Here we present two papers related to our work.

The first paper, "Prediction of NBA games based on Machine Learning Methods," by Renato Torres, uses team statistics to predict

games. His chief features include win percentage and average scoring margin. Rather than aiming for predicting a scoring margin, he sought to simply predict winners and losers. Similar to our project, Torres used linear regression and support vector machines, however, he applied principal component analysis over linear regression and support vector machines because he expected his features to be highly correlated. According to Torres' results, linear regression was his best predictor of winners and losers with a prediction rate of 0.7009. [1]

Jasper Lin, Logan Short and Vishnu Sundaresan in "Predicting National Basketball Association Winners," were able to obtain high test accuracy with logistic regression, adaptive boost, random forest and support vector machines. These accuracies were in the range of 63.3 to 65.1%. They used box score statistics from games starting with the 1991-1992 season through the 1997-1998 season. Their main conclusion is that the win record of past games had played a crucial role in predicting which team would win a game. When they took out the win records from their training, their accuracy dropped. This suggests that box score statistics may not capture all the elements of a winning team and that further research must be done. [2]

DATASET AND FEATURES

We retrieved all of our data from *basketball-reference.com* and divided it into two parts. First, we scraped and cleaned all the game scores for the NBA season of 2013-2014. We then retrieved the cumulative season statistics of all the teams in NBA for the same season. These statistics include traditional box score statistics

(e.g. points, rebounds, assists), shooting tendencies (i.e. range of shot selection and the associated percentages), and advanced statistics (e.g. eFG%, TS%, Pace).

The related works we looked at only took into account stats like past records and scoring margins. With our basketball knowledge, we decided to calculate other important statistics in addition to box scores. These stats had to be derived because they are not typically found in public. We derived these stats from data found in *basketball-reference.com*. However, in order to create these stats, a number of assumptions had to be made. For each NBA team, we derived advanced stats ranging from run-of-the-mill information like points and rebounds, to advanced statistics like shooting percentages between ten and sixteen feet and true shooting percentages.

Despite the fact that some of the derived statistics may not amount to significant data when observed as a single number, we hope that when considered as unit, all the advanced statistics end up helping our learning algorithms by providing some key missing insight.

For each team, we calculated advance statistics and joined it with box scores to get a total of 109 stats per team. For every game that we trained on, we placed the statistics of the home team side by side with the statistics of the away team to obtain a feature vector containing a total of 218 stats. We decided to place the statistics together and not merge or normalize them in some way because in basketball games, home court is a very significant advantage, and we wanted our features to reflect this advantage. Combining the features in some way may rid our learning algorithms of important metrics that capture the home court.

We trained our models on 1052 games and tested them on 264 games by performing 20-fold random-sample validation. Our predictors output the difference between the scores of the two teams, one home and one away, face off against each other. From the score, we can pick winners and losers. More importantly, however, is the parameters of our models. They tell us what data the predictors judge useful in making their predictions.

METHODS

Linear Regression

We began to analyze our data by implementing linear regression first. Linear regression is a relatively simple algorithm where the algorithm finds a line in higher dimensions such that the sum of the squared distance between the line and the data points is minimized. After having fit this line, the algorithm predicts the outcome of an unseen data point by plugging in the point's features to the line equation.

We chose to implement linear regression first mainly because it was a relatively easy algorithm to implement and it could provide a quick benchmark for our other results. In other words, if the results of our algorithms had been significantly worse than the results yielded by linear regression, then we could tell that either something had gone wrong with the implementation or that the algorithm was totally not suitable for this task. Thus, linear regression on the dataset was our first choice. The fact that the dataset is labeled was another factor that contributed to our decision of implementing a supervised learning algorithm as linear regression.

Principal Component Analysis & Support Vector Machines

With over 200 features, we had many that were derivatives or slight adjustments of each other, and thus some very apparent multicollinearity issues. We used principal component analysis (PCA) to reduce our features into the most essential, linearly uncorrelated components. Of course, 218 features is not such a vast amount, but we felt given the set of features we had, PCA would be prudent.

After we boiled it down to our principal components, greatly reducing the feature space, we used support vector machines (SVM) to linearly separate the data in possibly higher dimensions. In order to improve score predictions, rather than simple binary win/loss results, we used a multi-class implementation. From this, we could classify wins and losses, but also get an idea of the score margin. We accomplished this by looking at it as a series of binary classification problems.

Random Forest

A random forest is made of decision trees. Each decision tree can be thought of as a representation of the training data that is split into subpopulations based on a strong differentiating variable. Because each tree is built on a different subset of the training observations, random forest can easily handle outliers and can prevent overfitting by randomizing new trees during learning. Our data has a plenty of features and a random forest can help unravel complex unknown interactions between predictor variables.

Adaptive Boosting

Like random forest, adaptive boosting (also known as adaboost) is an ensemble learning technique that builds a strong classifier by combining multiple weaker ones. Adaboost works by creating a model of the training data

and then refining it with other models that attempt to correct errors in the first model. The number of models to combine is an input parameter of adaptive boosting.

We chose to use adaptive boosting because it has no parameters to tune (except for maximum models to combine). No prior knowledge is needed about weak learners and it can select features that result in a relatively simple classifier.

Gaussian Discriminant Analysis

After analyzing many regression algorithms, we decided to also approach the issue from a binary classification standpoint. Specifically, in addition to finding the score margin in a matchup between two teams, we also decided to predict the winner of a matchup between two teams. Since we could already determine the winner from the score margins (if the margin is negative then the away team is the winner, and if it is positive then the home team is the winner) we also decided to implement a classification algorithm to compare its accuracy against the other algorithms.

Gaussian Discriminant Analysis (GDA) is a generative algorithm, which assumes that $p(x|y)$ is distributed according to a multivariate normal distribution. The model assumes that:

$$\begin{aligned} y &\sim \text{Bernoulli}(\phi) \\ x|y = 0 &\sim N(\mu_0, \Sigma) \\ x|y = 1 &\sim N(\mu_1, \Sigma) \end{aligned}$$

The model calculates the parameters of the multivariate normal distribution according to a maximum likelihood estimate, and then when it receives a new query, just like a generative algorithm would do, it computes $p(x|y = 0)$ and $p(x|y = 1)$ and assigns the query to the class with the higher probability. Another reason as to why we decided to experiment with GDA was that the assumption that the features (given the class) are distributed according to a Gaussian around

the mean and variance, which are calculated according to a maximum likelihood estimate of the training data, is a very suitable assumption in our case. In other words, it is suitable to assume that the likelihood of offensive rating (one of our features) given win or loss, for example, is normally distributed with a mean and variance that are equal to the MLE average to those of the training data.

RESULTS & ANALYSIS

Since our approaches included a classification approach and a regression approach as detailed above, our results and analysis of the results will be based on these two approaches as well. We will start by analyzing our regression results. Below is a graph visualizing the cumulative distribution of score margin errors for the various approaches. This graph shows what percentage of the test data (y-axis) was how many points (or less) apart from the real margin (x-axis) for a given approach. We can observe that all of our three approaches, PCA/SVM, adaptive boost and random forest, have a similar distribution on the distance of the predicted margin from the real margin. Linear regression, however, has performed very badly when compared to the other algorithms in predicting the score margins.

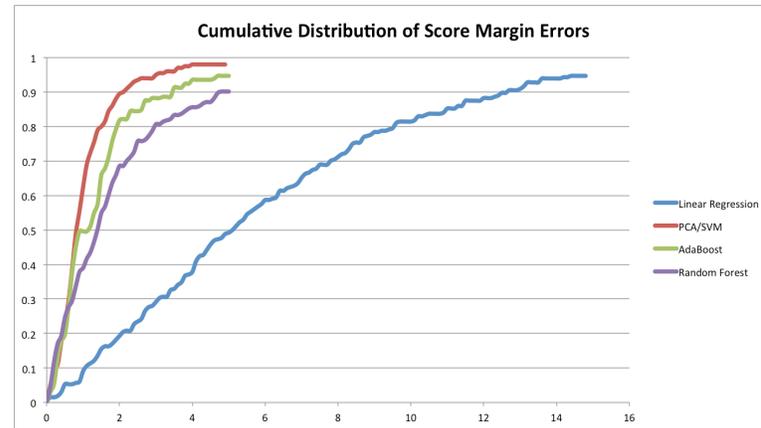


Figure 1: The cumulative distribution of score margin errors for various learning algorithms

We knew that linear regression would not be the best algorithm for predicting score margins but we still decided to implement it for the reasons detailed in Methods section. We think that the reason that it did not work as well as the other algorithms is due to a few factors. Firstly, linear regression works best when the features of the data are independent, but our features were not all independent. We had 218 features and some features, such as offensive rating and turnovers, are highly dependent on each other. We could have improved our results on linear regression through removing dependent features, but then there would be another main concern of whether the data points are linearly separable or not. Moreover, linear regression is relatively sensitive to outliers, which we can see many examples of in the NBA. Some teams that are not expected to win some games at all end up winning those games more frequently than they do in other sports, and this creates a lot of outliers in the data. These outliers shift the decision boundary significantly, resulting to a misinterpretation of the data. It is mainly due to these reasons that we believe why linear regression performed very poorly when compared to the other results, which explains the graph above.

We now shift our focus to analyzing our classification results. The chart below summarizes our results succinctly.

Algorithm	Accuracy	Precision	Recall	F1 Score
Linear Regression	64.26%	65.36%	78.52%	71.34%
Gaussian Discriminant Analysis	65.53%	67.90%	73.83%	70.74%
PCA & SVM	61.96%	64.04%	80.22%	71.22%
Random Forest (100 trees, 50 max depth)	61.36%	55.71%	35.40%	43.33%
Adaptive Boost (100 estimators)	60.23%	54.10%	30.00%	38.60%

Figure 2: The accuracy, precision, recall and F1 score of various learning algorithms

One conclusion we can make from this chart is the superior performance of gaussian discriminant analysis. It has the highest accuracy amongst all the approaches. We think that this is mainly due to the assumption that it makes regarding the distribution of the data, which was explained above. Thus, from the performance of gaussian discriminant analysis, we can conclude that the features of our data conditioned on the win/loss class are more or less normally distributed.

Furthermore, we had seen the poor performance of linear regression on regression tasks; however in the classification task, it has performed surprisingly well. We are still working as to understand why such a case has happened, but this result reinforces an advice Prof. Ng has given in class, which implied that the best way to understand if a model would work well on a dataset is to actually test the model on the data.

Moreover, despite the general accuracy that random forests can model, our random forest, sadly, did not perform as well as we had hoped. We experimented with different number of trees and maximum tree depth but found negligible change in accuracy, precision, recall. While the

accuracy of random forest is comparable to that of principal component analysis coupled with support vector machines (61.36% and 61.96% respectively), the recall of random forest was a meager 35.40%. This means that it was able to correctly detect only around 35% of the winning games.

FUTURE WORK

In summary, we have seen that PCA topped with SVM is a very stable algorithm that has performed very accurately in regression tasks and fairly well in classification tasks.

In a future rendition of this project, it would help us if we could train with lineup data rather than team data. Using the machine learning algorithms in this project, we now know how to predict the results of the matchup between two lineups at any given time in a game, and along with other data, we are also able to predict the likelihood of two lineups facing each other in a game. Using these two approaches, we are hoping to get more accurate and detailed results for predicting the score margin in a game.

Additionally, a future version of this project would train a neural network to predict values that go against transitivity between two lineups. Anecdotally, we know that transitivity does not hold, but being able to calculate it in a smart manner is difficult.

Lastly, here we present the most important features found by the respective algorithms in determining the outcome of a game, which was one of our motivations.

Algorithm	Top Feature	Second Feature	Third Feature
Linear Regression	ORTg/A	Opp_PF	DRB
PCA + SVM	ORTg/A	TOV	eFG%
Random Forest	Opp_FG%	Opp_DRB	Opp_3P
AdaBoost	Md.	Opp_FG%	%FGA

Figure 3: Top features for our learning algorithms

REFERENCES

[1] *Amorim Torres, Renato. "Prediction of NBA games based on Machine Learning Methods." Computer-Aided Engineering, University of Wisconsin, Dec. 2103.*

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

[3] *Ng, Andrew. CS 229 Lecture Notes*