NBA Outcomes Prediction

Yue Sun, YuNing Qiu,HeMing Huang

Introduction

In recent years, lots of talk have been risen in NBA circles regarding the rise of statistical analysis, additionally, due to the stability of the data has been improved recently, most notably play by play data which provides a record of each event. So, our team decided to do some work trying to predict the match outcomes based on the available NBA data. The basic idea is to split each game into miniature games.

Goal

Our team decided to do some work trying to predict the match outcomes based on the available NBA data. The basic idea is to split each game into miniature games.
In this project, we have used linear and logistic regression, naive Bayes, neural networks, and Support Vector Machine's and trees as the classifiers to predict the match outcomes.

The Data

The two main sources we used were ESPN's NBA website, basketball reference. Our dataset includes game score, the home and away teams, the players involved and their individual statistics, and also 50 individual statistics for each player in each season from ESPN database (ESPN data provide basic statistics such as the average number of points per game or the average number of rebounds per game for a given season).

Prepare data for model

To create the features, we considered each match and listed the players on each team first, after that, the players' statistics were merged from the previous season with the results in the matches of the current season. For example, suppose that there are n players in each team, with the i-th player playing mi minutes per game and having RAPM score ri in the previous season. Then, the weight for the i-th player and the team offensive RAPM.

$$w_i = \frac{m_i}{\frac{1}{n}\sum_{i=1}^{n} m_i}$$

Model Selection

Logistic regression:

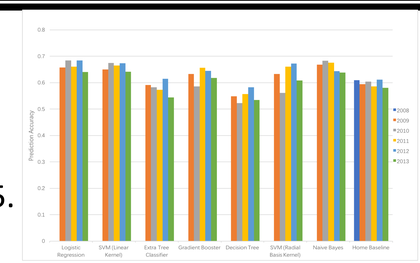
We use 0 and 1 to denote wins and losses, and we predict that the home team wins a new game if the predicted value for that game is greater than or equal to 0.5.

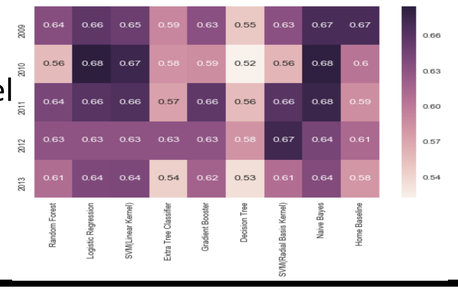
SVM:

We tried different kernels, one this should be noticed was that even though different kernels generate different decision boundaries, none allows to exactly classify all of the data points. For this reason, we use soft margin to train the model

Decision tree:

A decision tree is useful when one needs a good way to split a dataset with many features. In our case, there are 44 features overall. we choose the subtree that minimized the regularized training error to avoid the overfitting.





Conclusion

In conclusion, based on our machine leaning results, we can see that based on several basic statistic features, we can do some correct predictions (nearly 70%). However, as we can see, even though the best performance from logistic regression, the accuracy still was not higher than 70%.

Future Work

We could use multiple players' data from the previous season in order to predict the current season, which can avoid the situation if a player misses the previous season. Additionally, we could consider a weighted average of the past 3-5 seasons to get an estimate for the quality of each player