# NBA Single Game Score Difference Prediction

Xiao Fang[1] / Yang Hong[1] /Guanda Li[1] /Da Li[1]

[1] EECS Department, Northwestern University, Evanston, IL 60208 USA

**Historical data can be used in future score difference prediction. In this project, we employ more than ten attributes and several machine learning algorithms to predict score difference in a single game.**

***Index Terms* — Machine Learning, NBA, score prediction, Bayes Net, Decision Tree, KNN, Logistic Regression, Weka**

## I. Introduction

The result of an upcoming NBA match is always a popular topic among NBA basketball fans. People always have a drastic debate on the results of games and one will never surrender to others. As long as we could have a well-performed predictor, we can not only foresee the result of a single game, but also predict the fate of a team in playoffs and final. Besides, lottery companies often use these predictions, which means they have business values.

In this project, our target is to predict the score difference between two arbitrary teams. However, we won't provide the difference as a precise number, instead, a difference range the difference will be in is rendered.

## II. Data collection

We collect data from *www.basketball-reference.com*, a professional basketball database. To predict sports game results, data amount is considered very important so that we pick all game records over the last four NBA regular seasons – 5000 records in all. In addition, we are provided more than 20 attributes for each game record by this website, which helps us a lot in creating data models.

## III. Feature selection & Data processing

Feature selection and data processing are two key parts of our data model as it directly affects our model accuracy. In this project, 27 features are chosen manually to be our data attributes and several data processing techniques are employed to optimize our model.

### *A. Features Selection*

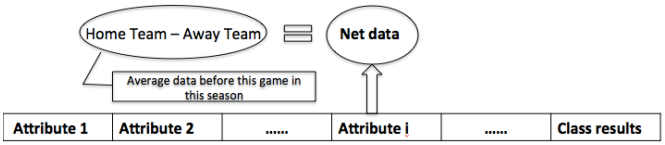
Specifically, we manually choose FG, FGA, FG%, 2P, 2PA, 2P%, 3P, 3PA, 3P%, FT, FTA, FT%, PTS, ORtg, FTr, 3PAr, TS%, eFG%, FT/FGA, ORB, DRB, TRB, AST, STL, BLK, TOV and PF these 27 features as our data attributes.

Received June 10, 2015 Corresponding author: Xiao Fang
(email: xiaofang2016@u.northwestern.edu)

### *B. Data Processing*

Data processing is one of the parts of our project that contains most workload. The reasons for that we cannot simply use the game records directly provided by online database are shown below:

Firstly, we need to generate a small database for each single game record because each game record needs average performance data of both the two teams before this game in this season as its attributes.




Secondly, we give each past games different affect weight according to how far away a game is before the target game. In our last status report we arbitrarily set the last-10-game weight as 0.7 and other-game weight 0.3. Now we are using a softer weight assigning system, making the weight falls down like a logistic shape. The weight function is

$$w(i) = \frac{1}{\log(1 + offset + i)}$$

where i is the number of games between this game and target game.

Thirdly, we try to unify the attribute difference scale in number, which means we try to make each attribute has a number falls equally in [0,10]. Attributes like 'XXX rate' and 'XXX percentage' are multiplied by 100 in this case.

For class result, as we mentioned, we calculate the score difference of two teams and then classify the results based on their values. A simple example is like 'score difference range 1' shown in the table below.

| Level | Score Difference Range 1 | Score Difference Range 2 |
|---|---|---|
| -3 | <-15 | <-10 |
| -2 | -15~-5 | -10~-3 |
| -1 | -5~0 | -3~4 |
| +1 | 0~5 | 4~9 |
| +2 | 5~15 | 9~15 |
| +3 | >15 | >15 |

However, as we mentioned in our last status report's 'future plan', we modified the division value by manually testing different values so that we can almost split the class results equally. The new division values are shown in column 'score difference range 2'.

## IV. MACHINE LEARNING MODEL & RESULTS

### A. *Feature Number Reduction*

This is also what we mentioned in our status report's 'future plan'. To reduce the feature number, we first applied J48 decision tree model to train our data and use 10-fold validation to prune the decision trees. Then we try to find those features that are close to the root in each decision tree. We selected XX of features as our final features of our data set. The final features are: FG, 2P, 2PA, 3PA, FT, FTA, PTS, ORtg, eFG%, DRB, STL and BLK.

### B. *Different Machine Learning Algorithms & Results*

Finally, after all those pre-process, our data set has 4051 data rows (this number also got reduced because we require both two teams finish 10 more games before the current game record considered valid) with each row having 12 attribute columns.

We then tested a number of different machine learning classification model and algorithms. For each of them we applied 10-fold validation to calculate the accuracy. We did not use f-measure because our result classes are not binary.

| **Techniques** | **Accuracy (6 result classes)** | | **Accuracy (2 result classes)** | |
|---|---|---|---|---|
| ZeroR | 18.0988 % | | 51.0123 % | |
| Bayes Net | 21.8519 % | | 59.5062 % | |
| J48 Decision Tree | 17.1852 % | | 59.6543 % | |
| K-Nearest Neighbor | K = 11 | 19.8272 % | K = 11 | 58.6667 % |
| | K = 101 | 22.1481 % | K = 101 | 60.716 % |
| Multilayer Perceptron | 22.716 % | | 63.1111 % | |
| SVM-SMO | Poly | 22.9383 % | Poly | 61.4815 % |
| | Puk | 22.0247 % | Puk | 63.1852 % |
| Logistics Regression | 24.5185 % | | 64.716 % | |

From what is shown above in the table, we can see that different algorithms/models appears to have different efficiencies on our data model. One simple conclusion is that J48 decision tree is definitely not suitable for this kind of data set, this happens mainly because the data set attributes have a high relationship with each other. On the other hand, logistic regression works significantly better than any other models.

Also, if compared to our last status report, we can see our accuracies dropped down. This happens for a obvious reason as we tried to equally split data according to their result class. As a result, the baseline accuracy (ZeroR model) is also decreased (closer to 1/number of result classes). However, our models do perform better considering the difference in accuracy between baseline and other models get increased.

One thing is that even though KNN, Multilayer Perceptron, SVM-SMO and Logistics Regression performs considerably well for our data set, the time for building and evaluating models are different to a large degree. KNN implemented by some good search algorithms and SVM-SMO with polynomial kernel function are the fastest. Then comes logistic regression while Multilayer Perceptron and SVM-SMO with Puk kernel function are really slow. As a conclusion, Logistic Regression and SVM-SMO with polynomial kernel function are best choices for our project.

## V. CONCLUSION & FUTURE WORK

From the result we can also find that even though classified into only 2 results, our best predictor can only perform out an accuracy of 65%. Although it is 14% higher than baseline, this low accuracy still make our predictor far from useful in real life. However, viewing in another direction, if we do have a predictor that has an accuracy of more than 80%, then we can use it to buy lotteries and earn money easily from lottery companies. Apparently lottery company people are clever and they will not allow this to happen, which also means that it is indeed very difficult to predict a single NBA game's score difference range even though we have all data we want to use to predict.

In the future, we first will read some more papers related to this topic. This will give us more ideas about how to improve our date set and machine learning algorithms. Besides, more test experiments are needed to find the best combinations of our current available method as we do not have enough time to find the global-best method combination for this project.

## VI. ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. They are also grateful to Prof. Doug Downey for optimizing our models.