Proceedings of Artificial Intelligence and Machine Learning for Engineering Design
AI & ML for ED
Spring, 2012, ,

PREDICTING NBA CHAMPIONSHIP BY LEARNING FROM HISTORY DATA

Jackie B. Yang
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: jackieyang@cmu.edu

Ching-Heng Lu
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: chinghel@andrew.cmu.edu

ABSTRACT

We apply a machine learning approach, Support Vector Machine (SVM), to predict the playoff results of National Basketball Association (NBA). Here the features are composed of the historial statistics of regular seasons, and each possible game is viewed as a sample. We also create pseudo-labels for the samples that didn't happen in the history. With 10-year records, the classifier performs 55.00% with conflict correction respectively. They show the results can be predicted better-than-chance but still have many unpredictable factors.

INTRODUCTION

The National Basketball Association (NBA), which is one of the most popular sports events conduct in United State annually. Sports lottery and all kinds of analytic reports spread around the world trying to get a closer guess of the final champion of the year; ESPN has a similar predictor[1]. Can we learn anything from history statics and results to see who will be the winner of sports game, such as NBA? This is what we are going to approach in this report.

A predicting system of 2012 NBA champion is developed by learning from history statics using Support Vector Machines (SVM) method. First, we collect some of regular season statics of 16 playoff teams in the past 10 years (2002-2011) as our features, then label them with results of history playoff games and pseudo labels. Then we trained our training data by SVM method with cross validation to build up our classifier. Eventually we will test our classifier by statics of 2012 and have win/lose results as output, then find out who are the winning teams of Conference Semifinals, Conference Finals, and last but not least, the 2012 winner of NBA championship.

BACKGROUND

Basketball is a kind of ball game that contains one ball, two teams each has five players on court, and two basket for scoring. The professional league of United States is NBA. There are 30 teams involve in the association, half of the teams are from east conference and the other half are from west. Each team usually plays 82 games per season and half of the games are at home and the other half are road games.

After regular season during November and April, 16 best performance teams (top 8 teams of each conference) out of 30 are qualified to continue to the playoff games in May till June. In playoff, there will be total 15 game series, the finalists is the champion of the year and will be rewarded by a trophy and champion rings for every players.

Tens of years in NBA history has come out with so many different champion teams. These champion teams may have some unique, critical performance and characteristics in common relative to other teams which makes them the best team. Trying to find out ways to predict new NBA champion teams from some of their regular season's performance, we look for our answers in the past NBA history.

DATA COLLECTION

There are hundreds of different statistics have been recorded in NBA, some of them are teams' statistics, some are individual players' and many other kinds of detail records. All of these data

[1] http://espn.go.com/nba/playoffs/predictions

TABLE 1. *A FEATURE SAMPLE BEFORE PRE-PROCESSING.*

Team A	Team B	Overall Statistics			Split Statistics		
		Info	Team A	Avg Opponents	Info	Team A	Opponent
A	B	G_A, W_A, L_A	$\vec{x}_A^I, \vec{x}_A^O, \vec{x}_A^D$	$\vec{x}_{\bar{A}}^I, \vec{x}_{\bar{A}}^O, \vec{x}_{\bar{A}}^D$	$G_{A,B}, W_{A,B}, L_{A,B}$	$\vec{x}_{A,B}^I, \vec{x}_{A,B}^O, \vec{x}_{A,B}^D$	$\vec{x}_{B,A}^I, \vec{x}_{B,A}^O, \vec{x}_{B,A}^D$

may possess useful data to help us analyze the results of games. However, we were unable to evaluate which statistic numbers are the most relative to game results, we have decided to use the most common ones. In our project, we only consider teams' statistics. There are 31 general terms of numbers are selected as our training features. We have collected history data samples from a reliable websites[2] and pasted them onto Excel sheet for later process.

We decided to look back past ten years' data of 16 playoff teams of each year. Many of other analysis or predictions of the winner of playoff were made in between regular season and playoff season, and most of them are based on teams' regular season performance and other individual status and statistics. Therefore, in order to find the possible key of winning championship, we listed 16 playoff teams of each year and found their regular season statistics. In these statistics, the most important part is that we found all the average numbers not only the overall of team self's performance and the rest of teams' performance in 82 regular season games, but also, the team-split data. In team-split data, we listed every combination of matchup, for example, for team A, we will have overall team A's performance and team B's performance within their match ups during the regular season, so as for team C, team D and so on. Thus, $16(\text{team}) \times 15(\text{opponent}) \times 10(\text{year}) = 2400$ is the number of our data samples.

Features Extraction

The original data samples we have include two parts, one is overall data and another is team-split data. Table 1 shows an example for original sample. In the data for each team, there will be 1 data sample represents the overall and 15 in team-split data, in total there are 16 data samples for each team of a year. In each single data sample includes performance of itself and opponent. All of the features are shown in Table 2. Feature are in total 17 different ones, except the G, W and L features only shows onece in each data sample, other features are both in overall and team-split data. Thus, every data sample has $3(\text{G,W,L}) + 14(\text{rest of features}) \times 2$ (self and opponent) $= 31$ features.

In order to get a simple comparison between overall and team-split samples, we decided to let each sample in team-split data be divided by overall data sample to get a relative performance of one team. In this case, we can get rid of the one data sample of overall performance and keep the relative performance in team-split data, which reduces the samples for each team of a year from 16 to 15. Now the new data samples are all the same kind of team-split, which is more resonable to use as trainning data samples. Also, in our trainning data samples, we regard team A versus team B and team B versus team A as different samples due to the overall statistics in this two cases are different. Thus, our real trainning data is a 2400 (total data samples) $\times$ 31 (features) matrix.

Pseudo-Labels for Training

In order to predict the results of playoff season, the actual playoff labels are needed as our training labels. However, there are only 15 series games in one playoff, each match up represents two situations, thus, only 30 playoff labels are available for 2400 data samples in one year data. We need all the labels for training our classifier, so we have made our own pseudo-labels for the rest of data samples.

We have created an algorithm for labeling pseudo-labels, which shows in Algorithm 1.

Algorithm 1 Pseudo-label creating algorithm

Require: y_i: label;
if $W_A > L_A$ **then**
  $y_i \leftarrow +1$
else if $W_A < L_A$ **then**
  $y_i \leftarrow -1$
else
  if $PTS_A > PTS_B$ **then**
    $y_i \leftarrow +1$
  else if $PTS_A < PTS_B$ **then**
    $y_i \leftarrow -1$
  else
    if $\frac{FG_A}{FGA_A} > \frac{FG_B}{FGA_B}$ **then**
      $y_i \leftarrow +1$
    else
      $y_i \leftarrow -1$
    end if
  end if
end if

[2] http://www.basketball-reference.com

TABLE 2. *ALL THE FEATURES WE USE IN OUR PROJECT.*

Feature		Name	Description
Single game	Info	G	Number of games
		W	Number of winning games
		L	Number of losing games
Single team	Info ($\vec{x}^I$)	TOV	Avg. turnovers
		PTS	Ave. points
		PF	Avg. totall personal fouls
	Offensive ($\vec{x}^O$)	FG	Avg. field goals
		FGA	Avg. field goal attempts
		3P	Avg. 3 pointes
		3PA	Avg. 3 pointers attempts
		FT	Avg. free throws
		FTA	Avg. free throw attempts
		AST	Avg. assists
	Defensive ($\vec{x}^D$)	ORB	Avg. Offensive rebouds
		TRB	Avg. total rebounds
		STL	Avg. steals
		BLK	Avg. blocks

LEARNING APPROACH

From the methods and theories we have learned from the class this semester, we choose Support Vector Machine (SVM) [2] as our learning method. In addition to convenience and usefulness provided by an opensource toolkit LibSVM [3], SVM performs very well for separating data from different classes and lowering the generalization error of the classifier. SVM creates a hyperplane to separate data from different classes, as showned in Figure 1. If a data point is close to the hyperplane, the distance is smaller and the probability of each class is about 0.5, where $P(C1)+P(C2)=1$, which means this data is more difficult to classify; in contrast, if the point is far away from the hyperplane, it is much more distinguishable between two classes.

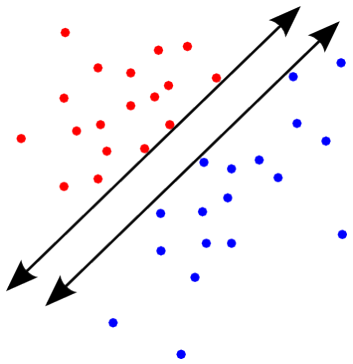

FIGURE 1. Principle of SVM [1]

Training Procedure

In this project, two classes are defined as Team A wins $(+1)$ and Team A lose (-1). Given some training data $\mathscr{D}=\{(\vec{x}_i,y_i)\mid \vec{x}_i\in\mathbb{R}^k, y_i\in\{-1,+1\}\}$, where y_i is either $+1$ or -1, indicating the class $\vec{x}_i$ belongs., and $\vec{x}_i$ is a k-dimension real vector. The optimization problem is to minimize $\vec{w}$ in $\vec{w}$ and b, subject to (for any $i=1...n$),

$$y_i(K(\vec{w},\vec{x}_i)-b)\geq 1, \tag{1}$$

where we use Polynomial kernel

$$K(\vec{x}_i,\vec{x}_j)=(\vec{x}_i,\vec{x}_j+1)^d \tag{2}$$

and Gaussian Radial Basis Function (RBF) kernel

$$K(\vec{x}_i,\vec{x}_j)=\exp(-\gamma\cdot\mid\vec{x}_i-\vec{x}_j\mid^2),\gamma>0. \tag{3}$$

In this project, we include LibSVM [3] in our code to build the predicting system. Regular season statistics of 16 playoff teams and win $(+1)$/lose(-1) labels for the past 10 years are provided as training data for SVM training.

Testing Procedure with Conflict Correction

As previous mentioned, each game result is regard as two data samples, therefore their labels should be a pair, one win and one lose. However, the trained classifier does not know these two samples are in pairs, hence there is no guarantee that they are predicted in the logically way. Thus, for team A vs B and team B vs A, we have the probabilities of $+1$ and -1 for both them. We add up the probabilities of A is winning and compare with probabilities of B is winning, then find out which one is higher, and decide who wins. Conflicts corrections can let our prediction be more reasonable. Figure 2 shows numbers of win and lose games are not equal before we correct the conflicts.

EXPERIMENTAL RESULTS

Evaluation by Cross Validation

After conflict correction we are able to compare the predicting labels and actual game results. Here 10-fold cross validation is involved to evaluate the classifier. Table 3 shows the performance of SVM classifiers using different kernel functions.

We find that the best average accuracy out of 240 possible games is 86.75% and actual accuracy out of 30 real playoff games is 55.00% with conflict correction.

Because Polynomial kernel function performs best for 30 real games, so we apply this function to do following experi-

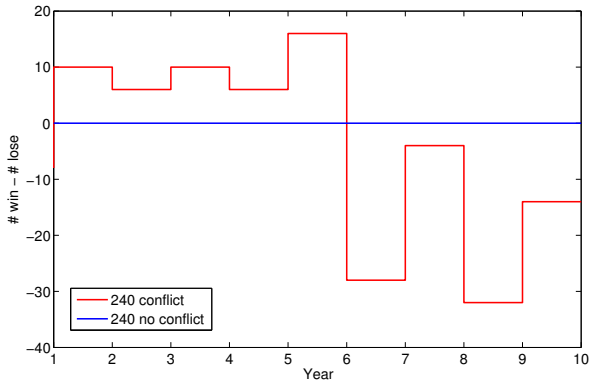

FIGURE 2. Number of labels changed after conflict correction

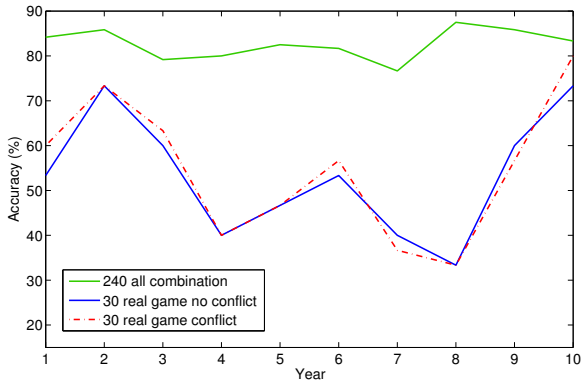

FIGURE 3. Accuracy of 10-fold cross validation

TABLE 3. *COMPARISON OF DIFFERENT KERNEL FUNCTIONS.*

-t	Kernel Function	Avg. ACC (%) (30 real games)	Avg. ACC (%) (240 all combinations)
0	linear	54.33	**86.75**
1	polynomial	**55.00**	85.75
2	radial basis	54.67	82.67
3	sigmoid	51.33	73.50

ments. Figure 3 shows the accuracy after conflict correction regarding to each fold.

Prediction from Features of 2012

We use LibSVM [3] to test for 2012 regular season statistics and correct the conflicts to finalize predicting labels. The trained classifier predict that San Antonio Spurs would become NBA champion of the year.

According to previous accuracy 85.75% for all possible game combination we set upset rate to 14.25%, in which the predicted losing team have a chance to defeat the winning one. In this case, the champion is still most likely San Antonio Spurs but had other possible competitive candidates.

DISCUSSIONS

First, manual created pseudo-labels are not 100% consistant to the actual playoff game results hence cause some confusion during the training process. Though we still have the average accuracy more than 50% .

Also, there is a large varience between accuracy of different folds of cross validation. They range from 33.33% to 73.33% after conflict correction as blue line in Figure 3. This condition is probably due to differences between each year's data.

Finally, we have tried four different kind of kernel functions and results are shown in Table 3, we can see that the sigmoid function performs the worst in our case. Since a SVM model using sigmoid function is equivalent to a two layer perceptron neural networks model, we can assume that if we use neural networks as our training model, we won't have a better results.

CONCLUSIONS

Sports game has too many features to consider that makes NBA championship become almost unpredictable. Although predicting accuracy of a single game can be improved, but playoff only contains 15 game series with single-elimination system which make every game very important. Therefore the champion of the year may be knocked home at the very beginning according to our prediction.

ACKNOWLEDGMENT

Thanks to Professor Levent B. Kara for his excellent lectures during the semester which gave us enough background knowledge to start and finish this project. Also we appreciate the help from Yun-Nung V. Chen, her opinion always enlighten us.

REFERENCES

[1] C. Clarkson Ken, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm, in *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, 922-931.

[2] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and other kernel-based learning methods, in *Cambridge University Press*, 2000.

[3] C. Chang and C. Lin, LIBSVM : a library for support vector machines, in *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.