Prediction of winner in NBA basketball games

Abstract

I apply three machine learning approaches, Linear Regression, Maximum Likelihood Classifier and Support Vector Machine(SVM), to predict the winner of a National Basketball Association(NBA) game. The data is selected from the historical statistics of regular season games and the features are defined and computed from it.

1. Introduction

The National Basketball Association is the premier men's professional basketball league in the world.The winner team within the NBA will gain popularity and get more income, and is beneficial to the league and its players. So predicting the winner in a game based on the previous performance is very valuable to business managers, coaches, players, fans, gamblers, and statisticians alike. I apply approaches to improve this prediction.

So far some people used the statistics of the teams like the average score, the shot efficiency and so on, while some people focused on the past history of the team like the winning percentage of the team, the overall behavior in the past ten games and the points differential. I'm trying to combining the features together to take a deeper look at both the behaviors and the historical data of the team. Now I've gathered the regular seasons game box scores from 2010 to 2016 and tried to compute the feature vectors. After that I'll start to get the teams' behaviors and apply PCA(Principal Component Analysis) to analyze the most important features. At last, I'll use three approaches for machine learning process and compare the result with that generated by the naive majority classifier based on the winning percentage of a team.

2. Model

I'm trying to combine the detailed behaviors statistics with the comprehensive situation of teams together and then use three approaches for training and prediction. The first big problem is where to gather the data. I searched the Internet and read the paper and finally decided to fetch data from http://www.basketball-reference.com. I can also gather the detailed behaviors of teams in each game, but it'll take more time. To define the features, there're at least eight features to take into consideration:

- Winning percentage of team A
- Winning percentage of team B
- Point differential per game of team A
- Point differential per game of team B
- Winning percentage of team A in latest N games
- Winning percentage of team B in latest N games
- Overall behaviors of team A
- Overall behaviors of team B

Right now I've extracted the six features and implemented the naive majority classifier with the features of Winning percentage of team A and B and the result is as follows:

season	accuracy
2011-2012	0.6363636363636364
2012-2013	0.6403580146460537
2013-2014	0.6455284552845528
2014-2015	0.667479674796748
2015-2016	0.6504559270516718

So the aim of the following job is to collect data and generate the feature vectors for the overall behaviors of both teams in a match. And then I'll implement three machine learning processes with eight features to show that the accuracy is better.

Implementations

From the website, I can collect the box scores of regular season game from 2011-2012 to 2015-2016. What I did to extract the data was to copy the box scores into a spreadsheet in $Excel$, and then I wrote a $macro$ to replace all the team names with numbers ranging from 1 to 30. After that, I converted .xlsx file to .txt with space splitting each instance. Note that some teams changed their name

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

in 2013 or 2014, so I had to find it out and replace it manually. Since the data was generated in the order of ascending time. So I deleted the date and detailed time of the games and added an column for labeling whether team A was the winner or not. If so, the label is 1. Otherwise it's 0. After extracting the box scores, I implemented the naive majority classifier based on the winning percentage of two teams in a game.

Naive majority classifier

Params	Meaning
pr	Probability of winning
win-count	Times of winning
count	Total times of games of teams
predict-correct	Times of correct prediction
total-instance	Number of instances

Each time I read one instance and predict the result based on winning probability of the visitor team and the home team. And I check whether the prediction is correct or not, if it's correct, I'll increase predict-correct and win-count of the visitor team by one, otherwise increase win-count of the home team by one and compute the pr for both teams. After all instances processed, I can get the ratio of predict-correct over total-instance to get the accuracy.
This mechanism determines the winner by looking at whose current winning percentage is greater in this season. Of course at the beginning they may be both 0 as no matches between them, and in this case I'll random a real number between 0 to 1 and I predict team A is the winner if this number is less than 0.5. I tested it on dataset of 2013-2014, 2014-2015 and 2015-2016 and the accuracy is around 65%. I'm still trying to grasp the behaviors teams in each game and it's very time-consuming. To deal with the data, I'm still using Java because I don't have too much experience with Matlab, which takes me more time. To improve the beginning accuracy, I plan to use the data from last season to predict the result, which should be better than randomly select one as the winner. This is the first progress when comparing with other people's idea. After gathering behavior's data for both teams in each game, I'll use PCA(Principal Component Analysis) to increase the convergence and reduce the dimension. When all feature vectors are ready, I'll go ahead to train the datasets with Linear Regression, Maximum Likelihood Classifier and SVM. The combination of the historical ranking and overall behaviors of teams should be more accurate to predict the winner of a game.

Linnear Regression

The first method will be implemented is the linear regression. The method consists of multiplication of each feature by a weight, making a summation of all values obtained and a bias, and uses this final value for the classification:
$Y = \omega_0 + \sum_{i=1}^{n} \omega_i * x_i$

I'll compute Y, and if Y is greater than 0, the visitor team is considered the winner and otherwise the home team is considered the winner.