Beautiful Data:
NBA Statistics Analyzer

PANG C. WONG
SONG XIAO
California State University, Los Angeles

aidanw7@gmail.com
yoho.song@gmail.com

Abstract

Big data is a popular term used to describe the exponential growth and availability of data, both structured and unstructured. For the most part, big data projects have been an exercise in computer science that, while being of enormous potential benefit to the business, has had little relevance to the sports, such as the National Basketball Association. And for now, big data may be as important to sports, as the Internet has become. Why? More data may lead to more accurate analyses,which may lead to more confident decision making for trading, drafting and coaching. A better decision can mean greater operational efficiencies, cost reductions and reduced risk. For our project, we will be using a widely used statistical analysis method,K Nearest Neighbour, to find out how the height and weight of nba players affect the wining percertage.

I. Introduction

Height and weight are probably the most common and basic data we can find for a NBA player. Normally, we could say, a player with higher height or bigger weight may have greater advantages than other. However, if a player has heavey weight but short in height, or is taller than other players but lack of strength,it is hard for us to judge if he can make contribution to the team. Therefore, it is meanful for us to take the incluences of the weight and height of players into consideration and analysis how those two values affect the winning percentage of a team. if we are able to find the relationship, we can predict the perfomance of a team in the current season based on previous seasons' data.

II. Collection

We collect the data of height and weight for each player in each team from NBA-Referance website by using a python library called Beautiful Soup. This library will allow us to scrap and parse the data on webpage and store them in the data structure we want to use.The Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree. It automatically converts incoming documents to Unicode and outgoing documents to UTF-8 so we don't have to think about encodings. We store height and weight into arrays so that we can extract and use them laer.

Table 1: *Example table*

NBA Season 2010 - 2011	
Team name	Winning Percentage
BOS	0.683
MIA	0.707
CHI	0.756
OKC	0.671
LAC	0.390
DEN	0.610
GSW	0.439
HOU	0.524

Table 2: *Example table*

Team: BOS Season 2010 - 2011		
Player name	Height	Weight
Paul pierce	6.06	230
Ray Allen	6.05	205
Carlos Arroyo	6.02	202
Avery Bradley	6.02	180
Marquis Daniels	6.06	200
Glen Davis	6.09	289
Semih Erden	7.00	240
Kevin Garnett	6.11	220
Jeff Green	6.09	235
Luke Harangody	6.08	246
Chris Johnson	6.11	210
Nenad Krstic	7.00	240
Troy Murphy	6-11	245
Jermaine O'Neal	6.11	226
Nate Robinson	5.09	180
Rajon Rondo	6.01	171

III. Method

For the Analyzation part, We calculate the mean of height and weight for total players of each team and we will get the table like below:

Table 3: *Example table*

Season 2010 - 2011		
Team Name	Mean Height	Mean weight
BOS	6.53	224
MIA	6.47	225
CHI	6.58	219
OKC	6.71	227
LAC	6.89	218
DEN	6.85	223
GSW	6.76	224
HOU	6.65	219

$$MeanHeight = TotalPlayerHeight/PlayerNumbers \tag{1}$$

$$MeanWeight = TotalPlayerWeight/PlayerNumbers \tag{2}$$

Then, we plot each team's mean weight and height into a two dimensional graph, which x-axis stands for height and y-axis stands for weight. Those are our training data. Next initiate the kNN algorithm and pass the trainng data to train the kNN (It constructs a search tree).

After that, we can grab one team's height and weight values for the current season and calculate the mean values, which will be regarded as our new input data. Now we can plot the new input data into our graph and our program will classify it based on the winning percentage with the help of kNN in OpenCV. Before going to kNN, we need to know something on our test data (data of new input). Our data should be a floating point array with size number of testdata * number of features.Then we find the nearest neighbours of new input data. We can specify how many neighbours we want. It returns: 1.The label given to newcomer depending upon the kNN theory we saw earlier. If you want Nearest Neighbour algorithm, just specify k=1 where k is the number of neighbours. 2. The labels of k-Nearest Neighbours. 3. Corresponding distances from new input data to each nearest neighbour.

IV. Conclusion

Finally, we could predict how good a team can be for the current season based on past data. For example, if we input the mean height and weight of Miami Heat, after applying the K-Nearest Neighbour method, we could find one closest team that has the same height and weight distribution from previous season. Then, we can predict Miami Heat's winning percentage for the current season since it close to the winning percentage of the team we have in the trainning data.

References

http://www.basketball-reference.com/
http://www.http://opencv-python-tutroals.readthedocs.org