Project Midway Report

Lee Richardson
Daren Wang
Chi Zhang
Xiaofeng Yu

November 4, 2014

Question

Our goal for this project is to predict the outcomes of NBA basketball games as accurately as possible. To do this, we have collected data from various sources in to find the best features to use for prediction. Our metric for success we have been using is a 0-1 loss function. We will predict each game in a season, and then check the actual results to see how many games we accurately predicted.

Dataset

We did not have a processed dataset for this project, so we created our own database. This means we had to select data sources and use ETL steps to merge them together.

The three main data sources we used were the ESPN NBA website [7], Basketball Reference [6], and a new website from Jeremias Engleman [5]. We use the ESPN data to get information about all NBA games from 2009-2014. Specifically, this includes the game score, the home and away teams, the players involved and their statistics. Also from ESPN, we have a player database, which has 50 individual statistics for each player in each of the seasons. There are 7139 games in this dataset.

The next data source we used was basketball reference [6]. The key reason we used this site is because they have a larger player database, with player information dating back to the 1950's and more advanced statistics, such as the widely used Player Efficiency Rating (PER), as oppposed to just the box score stats provided by ESPN.

The final data source we used was from a website put together by Jeremias Engleman [5]. which has the Regularized Adjusted Plus Minus (RAPM) statistics dating back to the 1980's. This statistic is widely used in the nba statistics community, and it has the best estimates of individual effects on defense than anything else. As we see below, RAPM is a very useful feature in predicting game outcomes.

To obtain all of these datasets, we used webcrawlers to pull them off their websites. All of these scripts can be found in our Github repository [8]. For the ESPN data, we used the BeautifulSoup package in the Python language. For the other two datasets, we used the XML package in R.

One of the major issues in our profect has been combining these three data sources into one single database. The ESPN dataset had a match_id, and playerID's for each game, so merging the game statistics with the players database wasn't very difficult. However, the basketball reference and RAPM datset didn't have these identifiers, so it was more challenging to put them together. We ended up using Player name, team, and team the join both of these datasets together. Some common probelms we have were inconsistent spelling of names in different datsets, inconsistent team names, some teams were changing cities, etc.. In the end, we were able to sync the idiosyncracies between the datasets, and we think it will be worth it going forward to have all of the extra features to experiment with. However, we don't doubt that there will still be further cleaning to do (I.E: Linking the Oklahoma City Thunder and the Seattle Sonics

together, since they are technically the same Franchise.)

We are using an SQLite database to store all of the tidied data. The design of this database follows the Third normal form to ensure there's no redundancy, and the indexes were built on frequently used keys to ensure the queries are fast.

Literature Review

We looked into the literature to see if anyone had worked on the same problem. We found a couple papers, especially [1] and [2], which were similarly attempting to predict the results of games. These papers used Linear and Logistic regression, Naive Bayes, SUpport Vector Machines, and Neural Networks in order to predict the outcomes of games. They also used the same loss function that we are proposing, which gives us a prediction rate to shoot for when implementing our algorithms. Specifically, [1] acieved the highest classification rate of 73% in the 1996 season with linear regression, and all of the other seasons and techniques had error rates in the mid-high 60's or low 70's.

One advantage we believe we have as opposed to the other groups who have attempted these prediction problems is that we have more features, specifically we have RAPM. RAPM has been anointed by many as the next big thing [4] in basketball statistics, and we hope that this as a feature can help differentiate us between the other attempts at game classification. There is a full explanation of the statistic here [3], but the basic idea is to split each game into miniature games, each one occuring time time periods when there's no substitutions. Then these five players play a certain amount of possessions on offense and defense, and we can estimate their overall effect on both ends of the floor.

Current Status

We have put a substantial amount of time into constructing our feature matrices. So far we have two, one which is using Defensive and Offensive RAPM, and the other using all of ESPN's box score statistics. To create these datsets, we went through each match to find the players on each team, and merged the players statistics from the previous season with the match results in the current season. To form the RAPM matrix, we used each players average minutes played from the season before, and used these as weights to compute a weighted offensive and defensive RPM statistic for each team. Below is a look at the final matrix we use for fitting our models

	ORPM_weight.0	DRPM_weight.0	ORPM_weight.1	DRPM_weight.1	homeWin
1	-0.28	0.89	0.65	0.18	1.00
2	-0.28	0.89	1.15	1.05	1.00
3	-0.29	0.99	-1.44	0.12	1.00
4	0.03	-0.66	0.04	1.09	1.00
5	0.28	1.26	-0.81	-0.20	0.00
6	-0.75	-0.46	0.51	1.02	1.00

After constructing these two matrices, we were able to run classification algorithms on them to test their prediction accuracy. Given the amount of effort that went into creating the matrices, we didn't have a lot of time to experiment with different algorithms and featues. However, we were able to fit a Naive Bayes classifier, trained on the 2009 season and tested on the 2010 season, which gave us 67% accuracy. We see this as a good sign for achieving a rate equal if not higher than the NBA oracle algorithms.

Realistic Goal

As mentioned above, a substantial amount of time was put into collecting and tidying the data, and creating the feature matrices. Now that lots of this work completed, we will have more time to experiment with different classification algorithms and combinations of features, to see if we can find combinations that improve our classification

accuracy. Specifically, we hope to use linear, logistic, and SVM techniques to classify our data and see how well they perform.

We believe it is a realistic goal to achieve a greater 70% classification rate on one of our seasons before the end of th course.

Stretch Goals

One of the parts of our classification we feel could be improved is that we are only usin the previous seasons data to predict the current season. This runs into issues. For instance, some rookies, even if thye are quite productive players, don't have statistics from last season so we have to assign them league average rates. Also, some players, such as Derick Rose, were injured in the previous season, so using just last year and discounting his MVP season would downgrade the Chicago Bulls team substantially. To combat these issues, we think it may be helpful to use more than just last seasons statistics, perhaps the last threee years, or test on projected statistics for 2014. The latter could probably spawn an entirely new project.

Another thing we could try to to is predict the outcome of the numer of wins in a season, as opposed to just single game results. To do this, consider fitting a Naive Bayes model. The outcome is two probabilities of each team winning th game. We could compress these probabilities to sum up to one, and then use a Uniform $\sim$ (0,1) random variable to choose the winner for each game. We could do this for each game in the season and add up the wins and losses for each team. Then we could repeat this process a couple thousand times to see a distribution of total wins for each team.

References

[1] Matthew Beckler, Hongfei Wang, Michael Papamichael *NBA Oracle* 2009.

[2] Dragan Miljkovic, Ljubia Gajic, Aleksandar Kovacevic, Zora Konjovic *The Use of Data Mining for Basketball Matches Outcomes Prediction* 2010: SISY 2010

[3] Paul Fearnhead, Benjamin M. Taylor *On Estimating the Ability of NBA Players*. 2010: http://arxiv.org/pdf/1008.0705.pdf.

[4] Steve Illardi. *The next big thing: real plus minus*. 2014. ESPN.com

[5] Jeremias Engleman. http://stats-for-the-nba.appspot.com/

[6] Basketball Reference. http://www.basketball-reference.com/

[7] ESPN. http://espn.go.com/nba/.

[8] Repository for Game Simulation. https://github.com/leerichardson/game_simulation.