Predicting Career Paths of NBA Players

Suril Shah, Alex Cousland, Daniel Robbins
CS 229 Machine Learning Final Project

Abstract

The goal of our project was to predict how successful athletes would be in professional sports. A noted blogger on emptythebench.com recently said, "Advanced statistics is one area that baseball is lightyears ahead of basketball." [1] This rung with us. We decided to use ML to contribute what we could to the statistics of basketball, by predicting the success of players in the NBA. Another reason we chose to focus on basketball was the fact that most players have a lot of stats in common as there are only 5 unofficial playing positions which leads to better availability of data. We initially were interested in applying supervised learning to predict the success of college athletes just entering the NBA. After running into a few problems finding labels indicative of success, we rephrased our problem statement into something more interesting: the prediction of players' future career paths. Using a combination of unsupervised and supervised learning, we were able to predict with moderate accuracy how a player's skills would change over time.

Data

We collected data from the NBA section on the ESPN website as it was pretty comprehensive and up-to-date. We put together a scraper in Python using the BeautifulSoup library. To briefly describe our scraping method, we first fed in the URLs of the pages containing the roster for each team in the NBA to the scraper. These rosters listed players as well as links to their profile pages. The scraper visited each player's profile page and collected metadata info about the player such as name, height, weight, age etc. Next, it navigated to their stats page for both NBA and NCAA and collected the averages data for all the listed seasons. Finally, we inserted all the data into a MySQL database to enable us to extract data efficiently via SQL queries for further use. Below are some statistics providing an idea of the kind of data we were able to gather:

Number of Teams (NBA)	29
Number of Players (NBA)	418
Number of NBA players with NCAA data	293
Max number of seasons for a player (NBA)	18
Total number of seasons (NBA)	3059
Total number of seasons (NCAA)	772
Number of types of stats per player per season (NBA) eg. Rebounds	20
Number of types of stats per player per season (NCAA)	18

Below are the different kinds of stats we collected for each player per season:

MIN: Minutes
PTS: Points
FGM-A: Field Goals Made-Attempted
FG%: Field Goal Percentage
3PM-A: 3-Point Field Goals Made-Attempted
3P%: 3-Point Field Goal Percentage
FTM-A: Free Throws Made-Attempted
FT%: Free Throw Percentage
OR: Offensive Rebounds
DR: Defensive Rebounds
REB: Rebounds
AST: Assists
BLK: Blocks
STL: Steals
PF: Personal Fouls
FLAG: Flagrant Fouls
TO: Turnovers
AST/TO: Assists Per Turnovers
DBLDBL: Double Doubles
STL/TO: Steals Per Turnovers
TRIDBL: Triple Doubles
RAT: NBA Rating
DQ: Disqualifications
SCEFF: Scoring Efficiency
EJECT: Ejections
SHEFF: Shooting Efficiency
TECH: Technical Fouls

Approach

We needed to find labels for our NBA players, which would both correspond to how talented the players are and not be overly subjective. Initially, we chose salary as the label for our feature vectors, which actually led to several problems. First of all, salary increases the more time you spend in the NBA. In our original model, we did not take this into account and ended up with pretty meaningless predictions. We then added an "experience" feature, which indicated how many seasons each NBA player has played, so that our model could find a relationship between experience and salary. After adding this feature, though, it became apparent that pretty much every NCAA player we would want predictions on would have experience=0, since they have not played in the NBA yet. Although there are a fair number of new NBA players in our training set, it doesn't quite make sense to predict the salary of new recruits using data of very experienced players. Moreover, the average number of seasons a player had played in NCAA in our data set was just ~2.6, since there are at most 4 seasons in a player's college basketball career. As a result of these complications, we decided to change the trajectory of our project to predicting career paths (change in performance over time) within NBA itself.

While looking at our results using salary as a label, we realized that many players had exactly the same salaries as each other. In fact, digging into the issue further showed that the NBA establishes minimum salaries, which increase with each year of experience the player has. In order to make our model non-piecewise, we had to leave out any feature vector corresponding to a salary at a minimum value. This decreased the error by 5%. However, it was still unstable at the early season level for each player. These results resonated with that of Fleenor [2], who also experimented with predicting NBA salaries in his thesis and concluded that "players do not get paid according to their performance on the court only. A variety of other factors come into play, such as fan appeal and who represents them as an agent."

In light of these issues we began searching for a better label for success in the NBA. Something that immediately came up was the Player Efficiency Rating (PER) metric. However, upon further investigation we found that PER had some major weaknesses such as not incorporating enough defense statistics into its calculation which resulted in some of the best defense players having below-average PERs. Moreover, the PER formula was created by adjusting the terms to validate the beliefs of its creator, John Bollinger, and thus did not follow a scientific process [3]. There are similar problems with other alternatives of PER such as Plus-Minus (+/-), Wins Produced (WP) etc [5].

However, we still needed to use some metric by which to measure the skill of a player. We used a rough approximation for player skill, which involved taking the inverse of negative stats (such as fouls and turnovers) followed by normalizing all stats to go from 0 to 1. This made all stats weighted equally. Then, for each player, we took the L2-norm of the player's stats for each season. Obviously, weighting stats equally is not a perfect metric for comparing players against one another; however, we felt that it would do a good job of showing an individual players change in performance over time.

The next thing we did was come up with what the most common career paths were. To determine this, we took all of the players with careers of a certain length (i.e. N=7 seasons), computed their performance for each year using the norm metric we created, and then put them in an N-years long vector. Once we had these vectors of performance over the span of a career, we then performed k-means clustering to determine characteristic career paths.

Once we had these distributions, we attempted to predict what a player's career path would be. Formally, given the stats of the i^{th} player, who had played N_i seasons (where $N_i < N$), we use the characteristic distributions to predict the trajectory of the player's stats for the remaining $N-N_i$ years. By finding the characteristic distribution which most closely matched the player's stats over N_i seasons, we would be able to predict the player's stats for the coming years by taking the N_i^{th} through N^{th} years of the characteristic distribution and appending them to the player's distribution.

Our first naive attempt at assigning players to clusters involved assigning a player the cluster that had the smallest norm for the number of years we were looking at. I will present an example with specific values to

make the procedure more concrete. We want to predict a player's seven year career path given their first three years of performance. To decide which characteristic career path they belong to we would simply compute the Euclidean distance between the three year input vector for the player and all of the first three year segments of the 7 year characteristic distributions. At first, when we performed this operation we were excited by the results as it did a reasonably good job of assigning people to the correct cluster. However, when we tested the accuracy of our algorithm on random data versus the actual data, we found that the while the accuracy was clearly higher on our data, random data still reported a disconcertingly high accuracy as well and made it obvious that our method was not predicting as much as we previously thought.

To attempt to combat this problem we moved to using a multiclass SVM with a linear kernel to classify which distribution the data belonged to. In the case of the example given above, the respective X's where the norm scores of the first three years and the label was the number of the cluster to which they belong. The results of this method are shown below. Note that these accuracies were obtained using the cross validation flag in liblinear:

Cross-Validation Accuracy (%) using SVM, norm of stats, N = 7

		number of clusters			
		3	**4**	**5**	**6**
length of seasons	**3**	48.2517	33.5664	29.3706	39.1608
	4	54.5455	46.8531	40.5594	33.5664
	5	67.8322	51.7483	61.5385	55.9441
	6	70.6294	64.3357	69.9301	58.7413
	7	79.7203	79.021	73.4266	69.2308

Cross-Validation Accuracy (%) using SVM, full stats, N = 7

		number of clusters			
		3	**4**	**5**	**6**
length of seasons	**3**	43.3566	36.3636	28.6713	26.5734
	4	47.5524	46.8531	37.0629	28.6713
	5	57.3427	52.4476	45.4545	44.7552
	6	62.9371	54.5455	47.5524	46.8531
	7	62.9371	51.7483	48.951	55.2448

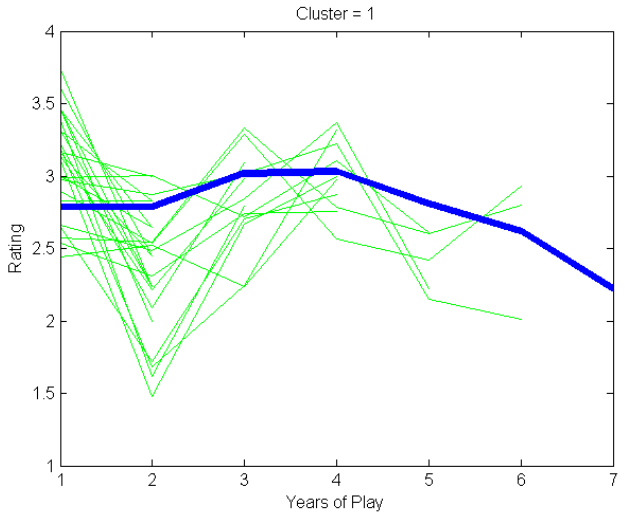
The "length of seasons" indicates the number of seasons used in each training example feature vector. As was expected, the fewer seasons left to predict (i.e. the closer the "length of seasons" is to N), the more accurate the prediction is.

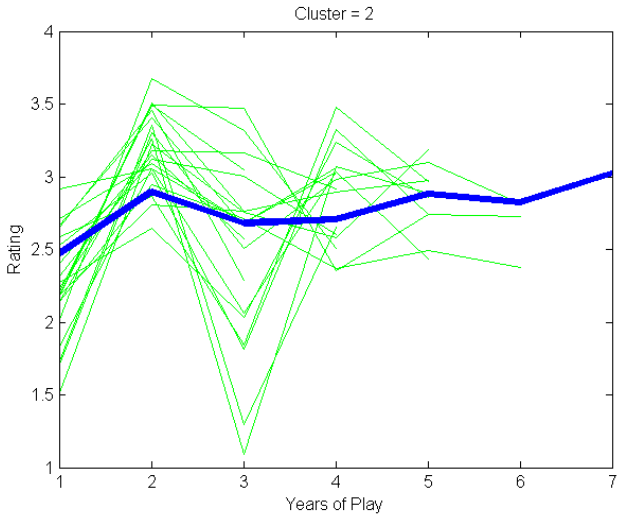
The "number of clusters" indicates how many clusters were created using k-means. As expected, the fewer clusters we have, the less room there is for error when making predictions about what career path a player will take.

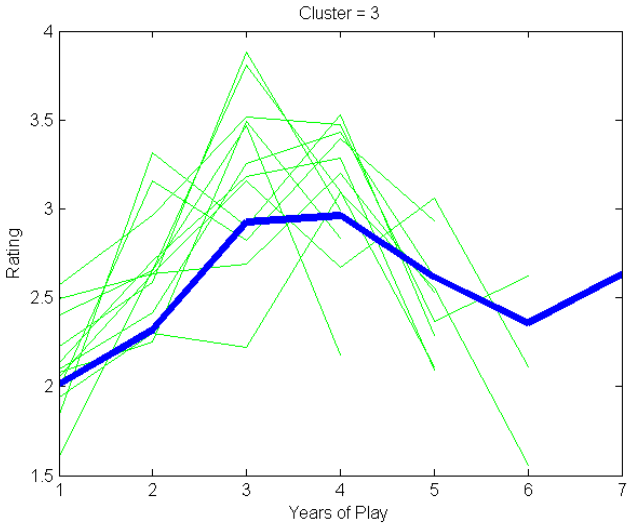
One interesting thing we found is that the SVM was less accurate when using all of a player's stats than when using our norm metric. This told us that there are most likely features that were unimportant to the SVM and should be ignored to improve generalization error. As an experiment, we implemented the forward search feature-selection algorithm to see if we could make a more accurate SVM model that does not use the norm metric. By picking the five stats with the highest cross-validation accuracy, we were able to improve accuracy from 28.7% to 33%. This comes fairly close to our norm metric's accuracy for the same setup (33.6%). This experiment also showed us that the most informative stats included points scored, three-point percentage, and age.

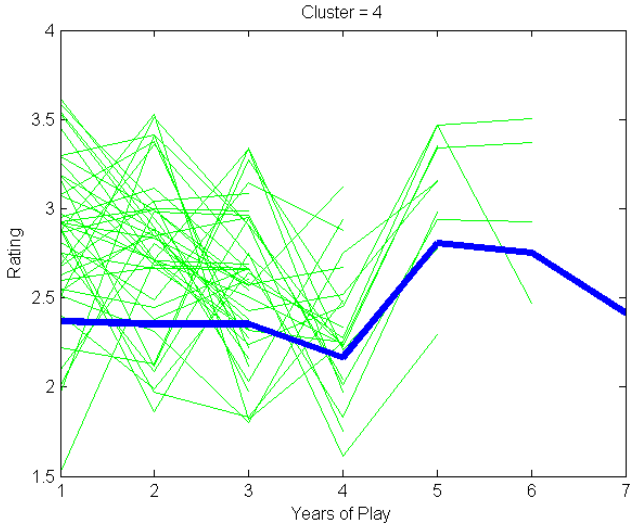
We also implemented model selection in order to select the best possible cost parameter for our SVM. We varied C from 2^-15 to 2^50 and found C=1 to produce the best cross-validation accuracy.

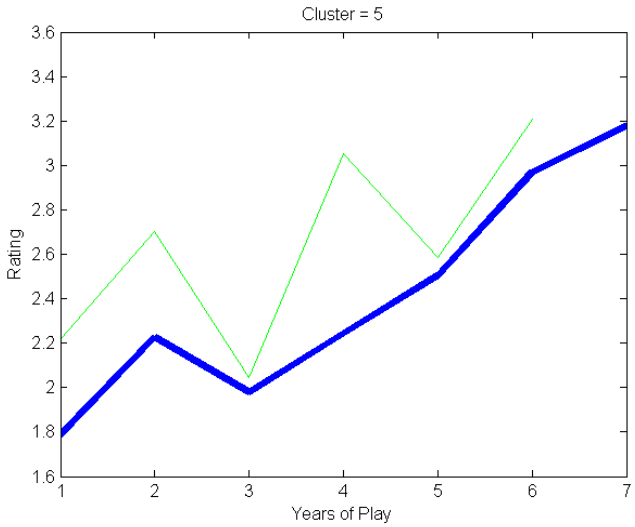
To test the qualitative accuracy of our algorithm, we generated characteristic graphs of 6 clusters using all players in the NBA with more than 7 years of experience. For every player in the NBA with less than 7 years of experience, we then predicted which career path (i.e. which cluster) that player would take using the multinomial norm-metric SVM. Below are our results. Blue lines represent the characteristic distributions. Green lines show each predicted player's rating history. If a player's green line appears on one of the graphs, it means that player was assigned to that cluster.

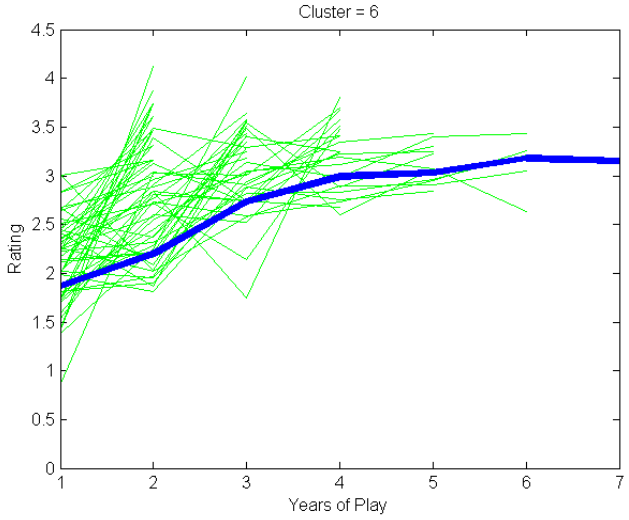












In case you know anything about basketball, here are some of the best players that ended up in each cluster, along with their supposed rankings: (1) Andre Miller #130; (2) Tony Parker #22; (3) Chris Paul #17; (4) Metta World Peace #69; (5) Kobe Bryant #3; (6) Lebron James #1. So, who is the single guy assigned to cluster 5, who's going to be the next Kobe Bryant? Turns out it is J.J. Redick, currently rated #88. While this may not seem so likely, our algorithm did assign Kevin Durant (currently rated #2) to cluster 6: the cluster that contains most of NBAs superstars.

Conclusion

In the end, we got mixed results. Our predictions were obviously not as accurate as we would like, as our SVM had accuracy between 30 and 60 percent depending on the conditions. However, the expected accuracy of guessing is only 16 percent, meaning that while our models is not perfect, they are able to glean some insight into what makes players change over time. In the end, our models were in many cases able to make qualitatively sensible predictions and could potentially produce results that are interesting to basketball fans.

References

[1] Blott, Zachariah. "Empty the Bench." *An Introduction to Advanced Basketball Statistics*. N.p.,n.d. Web. <http://www.emptythebench.com/2009/12/11/advanced-basketball-statistics-101/>.

[2] Fleenor, Andrew Thomas, "Predicting National Basketball Association (NBA) Player Salaries" (1999). University of Tennessee Honors Thesis Projects. <http://trace.tennessee.edu/utk_chanhonoproj/306>.

[3] "A Comment on the Player Efficiency Rating." *The Wages of Wins Journal*. N.p.Web. <http://wagesofwins.com/2006/11/17/a-comment-on-the-player-efficiency-rating/>.

[4] Kurylo, Mike. "A Layman's Guide to Advanced NBA Statistics." *KnickerBloggerNet RSS*. N.p., n.d. Web. <http://knickerblogger.net/a-laymans-guide-to-advanced-nba-statistics/>.

[5] "An Advanced Stats Primer for the NBA." N.p., n.d. Web. <http://www.goldenstateofmind.com/2011/12/6/2602153/advanced-stats-primer>.

[6] Fromal, Adam. "Top 100 Players in the NBA." <http://bleacherreport.com/articles/1153445-top-100-players-in-the-nba-right-now>.

[7] "NBA Total Player Ratings." <http://www.cbssports.com/nba/playerrankings>