Emma McLean and Alex Goodman
Profs. Sadigh and Charikar
CS 221
24 May 2019

Predicting the NBA Draft

We are trying to create a model that predicts the order of the NBA draft. For this problem, we found that the best accuracy achieved by other models in terms of draft order was about eighty percent, with first round prediction (first 30 draft picks) being much easier than the second round prediction (next/last 30 picks). At a high level, our own model takes in two factors: player skill and team needs, and uses these two factors to determine the best player who has not been drafted yet. Teams will always choose the best player for them, but two teams will likely have different best players because of their differing needs.

Measuring Player Skill

The first step in generating our two factors is to use a player's college stats to try and predict his NBA stats, specifically trying to predict their NBA win share (WS). We will use this as a approximate measure of player skill. To do this we will use a reflex-based model inputting various college statistics of players who were drafted into the NBA to learn what best predicts these players' success in the NBA. Our features are the following college statistics:

`(Games, Minutes Played, Player Efficiency Rating,True Shooting Percentage, Effective Field Goal Percentage, Offensive Rebound Percentage, Defensive Rebound Percentage, Total Rebound Percentage, Assist Percentage, Steal Percentage, Block Percentage, Turnover Percentage, Usage Percentage, Points Produced, Offensive Rating, Defensive Rating, Offensive Win Shares, Defensive Win Shares, Win Shares, Offensive Box Plus/Minus, Defensive Box Plus/Minus, Box Plus/Minus)`

and our target is the NBA win share.

Measuring Team Fit

After determining player skill, we will next use a team's roster to determine the current needs of the team. We will take in the team's roster, which provides us with a list of players, along with their playing position and win share (example below with made-up numbers).

```
Lakers: {(LeBron James, Pos: F, WS: 12.4), (Lonzo Ball, Pos: G, WS: 6.1), (Mo
Wagner, Pos: F, WS: 0.1), ...})
```

While not implemented in our preliminary model, we want to look at a couple of different possible ways that we can operationalize a team's "needs":

1. Replace the player with the lowest win share: perhaps a team will try and replace the player of their roster who is performing the worst (measured by the lowest win share of the team) by picking a player from the same position. This new player would probably be the player with the highest predicted WS (from Part 1 of our model) of all the players left at that position.
2. Add a player to the position with the fewest players: perhaps a team may look at the position where they have the fewest number of players. This option seems a bit naive but is worth trying anyway.
3. Add a player in their weakest overall position, calculated by the lowest average winshare by position.

No matter how we implement this factor in code, it will not itself be a separate model. It will likely just be the result of data cleaning and wrangling to come up with the measure on which we settle from the information we gather from different data sources.

Combining Our Factors

Once we have these two factors, we can learn the relative weighting of the two by using a reflex model that takes in the prior draft history data to output the weights. Once we have the weights, we can then finally model the whole problem of the draft.

Let's make an example of how our model would work for the Lakers. This year, the Lakers have the fourth overall pick in the draft, which means that the only players who will be unavailable to them are Zion Williamson, Ja Morant, and RJ Barrett (the consensus top 3 picks this year). Then the roster would look as it does in the example above, and the draft board would contain everybody except the top three players who were already picked. Assuming Mo Wagner has the lowest WS value on the Lakers and we are using Method 1 for team needs, the Lakers would want to pick a new forward to replace Wagner. The first sub-model, predicted WS, would take in the college stats of all the remaining players and output predicted WS for each one. Then the Lakers would be predicted to take the forward with the highest predicted WS, which in this case might be Cam Reddish or De'Andre Hunter. However, depending on the weights on skill and fit in the overall model, the Lakers might take Jarrett Culver (a guard) instead if they think that his skill outweighs their need for another forward. Whichever player we predicted to be chosen in that spot would be taken off the board, and the model would advance to the next pick.

Preliminary Experimental Results

We built a model using the features above and linear regression to predict NBA WS for each player in the draft. From there, each team picked the player with the highest projected WS, regardless of team needs. Our preliminary model does not have all of the features that we wish to implement, but it does give us a good starting place from which to move forward. Alongside adding in the team fit factor, we can also add in other features to a player's statistics, like the player's rank within his draft class by a subjective analysis (such as ESPN or CBS Sports). All of this should increase the accuracy of the model.

In our testing, out of the 60 picks in the 2018 NBA Draft, we found college data for 42 of them. We trained our model on the 2015-2017 drafts (about 170 picks' worth of data) and then tried to predict the order of the 42 players we had in the 2018 draft. 14 out of the 42 predictions (33%) were within 10 picks and 28 were within 20 picks (66%). We achieved these numbers with the missing data (the other 18 picks), which we hope to fill in later and would surely help a fair amount. Additionally, 18 out of the 30 players (60%) we predicted to be picked in the first round were picked there.

Our model still has relatively high error (mean error: 16.8 picks, median error: 16.5 picks). Adding in team fit will address some of this error, and we might move to a more complicated reflex-based model like a neural network. Other than that, we just need to work on obtaining more data and cleaning it from the messy format in which it comes. This becomes especially important if we turn to a neural network, but either way we are limited by the tiny number of NBA draft picks in history (only a few thousand).