Fantasy Basketball using Knapsack

Kevin Luo

Ilya Yudkovich

Day to day fantasy is up and coming so it is a great way to make some quick buckets of cash while not doing too much work.

Fantasy basketball is a great way to make money...if you know how to predict a good lineup. A good line up will generate more points than the other people that you are winning with. The basics of a line up consist of nine players. More specifically, 2 point guards, 2 shooting guards, 2 small forwards, 2 power forwards and a center. The way to gain points from your lineup all depends on the scoring schematics of the application you are using. For this project, Kevin and I based our algorithm to work with FanDuel (http://Fanduel.com).

How does Fanduel work?

Fanduel provides a list of all the players playing that night. On the weekends, it gets a bit tricky where it does not show all the players, but only those that are playing at 8pm or later. They also have a bunch of contests that vary in how many people participate in each one and what amount of participants are actually winners. We ignored this part -for now- since we are only focussed on predicting a line up. Each player playing that day has an average Fantasy Points per Game (FPG) statistic, who they are playing, how much they cost, what team they play for and what position they play. The FPG are calculated the following way:

For each:

* 3 pointer made - 3 points
* 2 pointer made - 2 points
* Free throw made - 1 point
* Assist - 1.5 points
* Rebounds - 1.2 points
* Steal - 2 points
* Block - 2 points
* Turnover - -1 points

Every user on Fanduel is allotted 60000 to spend on these players. The most points gained during an evening, the better the chance to win the contest that the user is participating in.

Our program:

Fanduel was kind enough to provide a download as CSV\* button for all the players. Using this, although we had to do it manually, we were able to download the csv files and analyze them using python and the csv library. Once we imported the csv file, we could begin our analysis. We created a Player Object so that it would allow us to easily access pertinent information for each player without parsing the csv file over and over again. Each Player Object contained the name, price, and position. Then, we split the players into the 5 different lists of positions that they play, as well as had a separate list that was all the players playing that day. We wrote the knapsack algorithm and then applied it to the sorted list of all players. The weight for each player was the price that they cost and the value was the amount of average FPG that they generated, the total weight that the knapsack could hold was 60000. We then passed into our knapsack algorithm the list of all the players and 60000.

Things to work on:

Currently, our knapsack does not account for the different position, but instead simply finds the best players to fill up the knapsack. We need to integrate the fact that there can only be 9 players and of those 9 players, we can only have a certain amount of players at each position. This will be fixed with conditional statements, checking to see if we have met the requirements. It might also be possible to implement some sort of binary pairing algorithm that could factor in all of these datum. Another thing that we can work on is instead of simply going by FPG, we can try and get a better estimate of how many points a player will generate by using some outside resources. One that we had in mind was Rotowire defense versus position. (<http://www.rotowire.com/daily/nba/defense-vspos.htm>). Rotowire is a tool that most people who partake in NBA fantasy use to get the latest news on players; however, they also provide a tool that lets you compare how many points each team gives up per position. Incorporating this tool would add another layer of information that would help in predicting how each player would do that night.

Another great tool that is out there is ESPN, they have great data bases that we could use to compare past performances of teams against teams, and even more in depth of player versus player analysis. Although this would be tougher, and we would have to figure out how to compare rookies. There are a couple of tools that are already out there that would let us parse the ESPN website to get this information. Two of these tools are Requests (<http://docs.python-requests.org/en/latest/>) which is a replacement for the urllib2 API that python presents. Then there’s also Beautiful Soup (<http://www.crummy.com/software/BeautifulSoup/>) which allows to grab data off the screen.

One last and definite future implementation would have to be an automated script that gets the csv file from Fantasy Duel. Although it is trivial work to login and click the single button, it would not be of true computer science nature to do that. Why not write a medium to lengthy sized script to do it for us. Since this is trivial work though, we most likely will not focus on this until we have the algorithm padded a bit more. This would be a polishing feature and those tend to be done last.

----------------------------------------------------------------------------------------------------------------------------\*For now we have been using just one single csv file from December 5, 2015, it doesn’t make sense for us to currently keep updating it everyday.