Axel Fries, Drew DeRubertis, Michael Freeman
MAT 395: Applied Math in Data Science
Spring 2020

Bayesian Draft Analysis

Intro:

The NFL Draft is one of the biggest events in football as teams are able to select new team members for their upcoming season. Each team wants to pick the best player possible but it is often quite difficult to know who that is, and sometimes they do not get who they want due to their position in the draft order. Our model can help teams navigate the draft by providing a table which projects the probability that each player will be picked at each spot.

The probability distribution is determined with the help of the Bayesian Model. Bayesian Models begin with a 'prior' probability distribution, used as a reasonable first guess. As more information is added, your prior belief is either confirmed or revised to some degree. The degree to which it is refined is a function of how reliable the new information is. The model can be implemented into the NFL draft by using Consensus player rankings or BIG BOARD rankings as the prior, and expert projections as the added information.

What is produced is a probability distribution of selection at each pick number for each player which in turn can be very helpful for an NFL team when preparing for the draft.

Collecting the Data:

Using the Bayesian Statistical Model described below, we needed to establish a dataset that consisted of player rankings and mock drafts for the 2016 through 2020 drafts. We first needed to construct a ‘prior’ probability distribution reflecting where a player was likely to be available. Following Brian Burke’s outline, we constructed this ‘prior’ distribution by using the top 150

overall player rankings from Gil Brandt, Todd McShay, and Mel Kiper, three highly revered NFL Draft analysts. In order to establish an average pre-draft ranking distribution for each year, we averaged these analysts' rankings for each player, breaking ties by looking at their average mock draft projection. An example would be if CeeDee Lamb (Wide Receiver) was ranked 15,17,20 overall by our 3 analysts' overall player rankings we say his average ranking is 17.33. If another player also had an average ranking of 17.33 this tie would be broken by seeing which player is projected to go higher in our mock drafts. After ties were broken and the average overall player rankings were established, we had our 'prior' distribution.

Next, we needed to acquire expert projections to accumulate our likelihood distribution. We scraped mock drafts from McShay, Kiper, Daniel Jeremiah, NBC Sports, Sports Illustrated, and CBS Sports. In order to get a measure of how historically accurate each analyst or source is, we scraped these mock drafts for 2016-2019 and looked at the accuracy of each analyst. Once we established historical accuracies, we scraped these six mock drafts for 2020, as close temporally to the actual draft as we could get. Additionally, Kiper, McShay and NBC Sports had two-round mock drafts available for each year. We ran our first round iteration of this model with all 6 analysts' first round mock drafts. The accuracy of this is expected to be better with a more populated field of analysts. We then used the three aforementioned second round mock drafts to create a less accurate but hopefully useful model of the second round as well.

In terms of the specifics of the player rankings and mock drafts that we scraped, we used the most updated and final rankings and mock drafts that each analyst put out each year. This limits the chance that new information came out or new injuries occurred that caused an analyst to be less accurate than the typical field of analyst because they didn't have as much information.

Taking rankings and mock drafts from a day or two before the actual draft evened the playing field for the analysts and isolated which analyst was truly the most accurate.

The data scraping method utilized was the package Beautiful Soup in Python 3. The coding was done in Jupyter Notebooks and the scraping files are included with the submission. To scrape ESPN+, which requires an account login and password, the library 'urllib' was also utilized in Python 3. Beautiful Soup allows us to parse through the source code of a website. We read in source code and navigate to find the proper object class that player information is stored in and we retrieve the player names from these objects. For some websites like NFL.com, this navigation is easy because player names are stored in lightweight container objects, however on other websites like NBC Sports, player names were embedded in complicated lists and we needed to establish unique indexing functions to find the player names within the lists. We populated a comma delimited list of player names and their corresponding ranking or draft slot, and wrote this list into a CSV, which we would then read into MATLAB to populate our data matrices.

The Algorithm:

http://advancedfootballanalytics.com/index.php/home/research/draft/235-bayesian-draft-analysis-tool-is-now-live

As mentioned above, our model utilizes the consensus player rankings and expert mock drafts and weights each depending on its historical accuracy. The weights are needed in order to use Bayes Theorem. The theorem can be described as the formula below:

$P(Act|Proj) = P(Proj|Act) * P(Act) / P(Proj)$

- P(Proj/Act) is the likelihood of how often historically the actual pick was selected at the projected spot, by the expert.
- P(Act) is how often the ranking is correct for that draft spot.

Perform these steps for each possible pick number for that spot in the draft.

- Sum up P(Proj|Act) * P(Act) for that pick at each spot.
- P(Proj) is therefore 1 – above sum.
- Once P(Proj) is determined, we can find P(Act/Proj) for each player.

Determining P(Act) and P(Proj/Act) for each position in the draft is the tricky part of this method. The rest is quite straightforward. The P(Act) or prior distribution was found by first compiling each of the consensus player rankings for 2016-2019 and putting them into a matrix. With 3 rankings of 64 players (1st and 2nd round draft picks) per year for 4 years, the matrix had 64 rows and 12 columns. Each row represents the actual draft spot, and each column represents where each ranking had ranked that player. This matrix is then put into the MATLAB code which determines the prior distribution. The code takes in the matrix and finds how often a certain pick was ranked at each position. For instance, the number one pick from 2016-2019, was ranked first 4 times, second 6 times, and third 2 times. Thus, the first row in the prior distribution will be [0.33 0.5 0.167 0... 0]. Doing so for each draft pick will create a 64x64 matrix which will be used as P(Act).

The method for determining P(Proj/Act) or likelihood distribution is quite similar. We start by compiling each expert mock draft of 64 players for 2016-2019 and using them to build a matrix. With 6 mocks of 64 players per year for four years, the matrix built had 64 rows and 24 columns. This matrix is then put into the same MATLAB code as above which produces our likelihood distribution. The likelihood distribution is a 64x64 matrix with each row representing

the pick and each column representing how often each pick was projected to go at each spot. For instance, if the number one pick from 2016-2019 was projected first 12 out of 24 times, then the [1,1] position in the matrix will contain 0.5. Doing so for each draft pick will create a 64x64 matrix which will be used as P(Proj/Act).

Once we had P(Act) and P(Proj/Act), we noticed that there were a lot of values that were zero. Because we only used six mock drafts, and 3 rankings for a short span of four years, our dataset was not huge which led to all these zeros. To solve this issue, we used the exponential dampening function within excel. We performed exponential dampening on all P(Act) rows and all P(Proj/Act) columns individually with a scaling factor of 0.1. The function smoothes out the data by evening out the values, thus getting rid of a lot of zeros. Another method of smoothing the data is by using KernSmooth in R. None of us are proficient in R which is why we used excel instead.

Now that P(Act) and P(Proj/Act) was smoothed and ready, we could perform Bayes Theorem on each player for the 2020 draft. The theorem was used by first taking the row of P(Act) that corresponded to the average ranking for that player for 2020 and transposing the column of P(Proj/Act) that corresponded with the average projected pick for that player for 2020. We then multiplied the first value in P(Act) with the first value in P(Proj/Act), the second with the second and so on. Then, we found the sum of these products and had it be subtracted by 1 in order to get P(Proj) for that player. Lastly, we divided each of the above products by P(Proj) which gave us the final probability distribution for that player. Doing the above for each player in the first two rounds of the draft gave us our full distribution probability.

The Results:

The final results are outputted by the Matlab code and once they are in excel they should show the probability of each individual player being picked at every slot in the draft. For example in our model Joe Burrow had a .93 probability of being the first pick. There was data on 64 different players in our model, but it can work with more or less. We formatted the results to show the probability to 4 decimals places since anything less would be considered insignificant to the model and for decision making.

After the probability table was complete it was very easy to create a distribution that showed the probability of each player being available at each pick in the draft, which is what the Carolina Panthers used in their draft process. We did this by showing a probability of 1 for every player for the first pick and then subtracting the probability that a player was picked at the previous slot. So if we were trying to find the probability a player would be available for the team drafting 3rd in the draft, then we would subtract the probability they were drafted at pick 2 from the probability they were available at pick 2. So the equation would be: P(Available at 3) = P(Available at 2) - P(Picked at 2). This was very helpful to the Panthers especially after we used conditional formatting that highlighted the box green if it was very likely they would be available to be picked and gradually faded to white and then from light red to dark red as that probability dropped. This allowed us and the Panthers to see any outliers from the data and potentially over or undervalued players in the draft.

These final results in both tables we created were great to look at before the draft, but once picks were made they could change drastically. A simple way we combatted that problem was to change the probabilities a player was picked at each slot in the draft, which was in our table that was first outputted. Since the probability of availability table used formulas from the first table, changing the value in the first table would change the probability of availability. Once

a player was picked we changed the probability he was taken at that pick to 1. Then we changed all other player's probabilities to zero at that pick. This gave an accurate representation of the draft board in real time. The only hitch was the availability becomes negative eventually which can mess up the conditional formatting, but the formatting is not necessary to interpret the results.

Improving the Method:

Overall this method is a very creative way to attack the problem of predicting the draft and it is also accurate, in theory. The only thing holding this model back is the amount of data needed to obtain a high degree of accuracy. Since we only had 6 mock drafts and 3 player rankings for 4 years, there were many places where players ranked at certain spots had not gone in the draft, therefore making the P(Act) or the P(Proj|Act) equal to 0. Therefore when multiplying the two together it only took one of them being 0 to make the result 0. We tried to solve this problem by smoothing the distributions at each pick so there weren't holes, but that will not be as accurate as having actual data in its place. The only way to do that is to add more credible mock drafts and player rankings throughout the years. There are many of them out there, but few that have been consistently released from 2016-2020. Unfortunately the only way to get these new data is to wait for the next draft to occur or have another trusted source emerge for mock drafts or player rankings. If this project is done again Pro Football Focus is a resource that may produce fairly accurate mock drafts and player rankings despite being heavily reliant on analytics, which makes their data unique.