

Quarterback Cluster Analysis Without Expert Knowledge And How It Failed

Gavin Byrnes

4 February 2016

Abstract

This paper is a companion paper to the introduction of a new approach to portfolio construction in (cite other paper). In this paper, we attempt to remove the need for the expert knowledge provided by the Overall rating in clustering analysis. We use a similar clustering method on quarterback ratings without the assistance of regressing against Overall ratings and demonstrate that the results do not pass a sanity check on known quarterback quality, implying that the portfolio construction archetype we are attempting to build will most likely not be able to stand alone without any expert input.

1 Introduction

In (cite other paper), we establish the groundwork for a new approach to nonlinear portfolio construction using the process of building a football team. We used Madden video game ratings and clustering techniques to construct different archetypes for the quarterback position that we could then use to evaluate players against different heuristics of what makes a good quarterback. Though our primary data set was individual criteria ratings, we also used the game's Overall rating as a guideline of expert knowledge about which criteria are considered most important to the position. While there is nothing inherently wrong with the use of expert knowledge in this way, we wanted to explore the possibility of eliminating the use of the overall rating (and therefore this type of expert knowledge); if we could construct reasonable archetype ratings without using the Overall score as a guideline, this

would be an indication that expert knowledge might not be as necessary to our portfolio analyses and that in situations in which it was unavailable we could still build strong portfolio paradigms with only our analysis.

Unfortunately, we failed to do so. The archetypes built without using expert knowledge of Overall ratings failed to pass the eye test in a number of crucial ways, implying that, while we can use clustering analysis and our overall process to make portfolio construction more efficient, we still need some expert guidance, especially in constructing the archetypes. This paper examines our attempt to remove the Overall rating from our quarterback clustering process, and concludes that for the time being it has been unsuccessful.

2 Data Collection

¹ The data for this project are the ratings for the video game Madden 2016, found at [?].

1. We imported the ratings for quarterbacks into R.
2. We subjectively reduced the number of criteria from 52 to 21, based on the observation that some abilities (Tackling, Catching, Kick Power, etc.) are irrelevant to playing quarterback

3 Processes

²

The first thing we did was to cluster the quarterbacks based on their ratings on these 21 categories. This was useful as an extremely basic sanity check on the clustering algorithm; generally speaking, good quarterbacks clustered with other good quarterbacks, scrambling quarterbacks clustered with other scrambling quarterbacks, etc. However, this was insufficient for several reasons:

¹This section is reproduced mostly verbatim from (cite other paper), except for the attempt to exclude the overall rating.

²This section is mostly reproduced from (cite other paper)

1. We want the archetypes of quarterbacks to be things like "pocket passer" and "running QB", not "good quarterback" and "bad quarterback"; this would ruin the point of comparing and contrasting across archetypes.
2. Clustering on raw attribute scores tends to put good quarterbacks with good quarterbacks and bad quarterbacks with bad quarterbacks, even if their *relative* strengths and weaknesses don't match up. Therefore, we want to normalize scores when we cluster players. (but not when we rate them, of course)
3. The easiest and most intuitive way to normalize the scores is to make every player's ratings add up to the same number. However, the ranges of values differ greatly in both mean and standard deviation. For example, most quarterbacks have an Injury rating between 85 and 95 and a Throw Accuracy Deep rating between 50 and 75, while their Trucking ratings range from 20 to 76!
4. The mathematically intuitive way to ensure that the normalization of ability is not dominated by the more variable attributes is to first convert the scores to z-scores.³
5. Finally, having so many attributes makes the clusters awkward to visualize. In addition, anecdotal evidence suggests that some attributes, like Speed and Acceleration, or Throw Accuracy Short and Throw Accuracy Medium, would be very highly correlated. We decided that it would be a good idea to create clusters of the attributes themselves as well as the players to construct a smaller number of metafeatures to use in the eventual player clustering.

Therefore, we performed the following process:

- Begin with the data frame of the overall ratings and 21 raw attribute scores for quarterbacks, QBOriginal.
- Construct the data frame QBzscores, obtained by replacing each element with its zscore using the means and standard deviations of each column of player ratings.

³Calculated for each data point by subtracting the mean of the overall dataset and dividing by the standard deviation

- Construct the data frame `QBNormzscores`, which removes the Overall score and then adjusts every player's z-scores so that every player's sum of 21 attribute normalized z-scores is 0.
- Construct a covariance matrix `covar` of the z-scores on each of the 21 attributes.

We entered this phase of the process with the following data frames:

- `QBzscores`: Z-scores on the Overall score and 21 attributes
- `QBNormzscores`: Z-scores on 21 attributes, adjusted so that each player row sums to 0.
- `covar`: The covariance matrix for all 21 attributes, based on relatively correlated player performance on pairs of attributes

We performed a cluster analysis on the above covariance matrix, which in this case we elected to place into four clusters:

- Power Run: Carrying, Trucking, Stiff Arm, Throw On The Run
- Accuracy: Awareness, Throw Accuracy Short, Throw Accuracy Mid, Throw Accuracy Deep, Stamina, Toughness, Play Action
- Power Throw: Strength, Throw Power, Injury
- Speed: Speed, Acceleration, Agility, Elusiveness, Ball Carrier Vision, Spin Move, Juke Move

We tried several methods of weighting the subcriteria within these four clusters, from allowing them to be all equal to estimating based on a new covariance matrix that indicated which ones were more related to each other. In all these cases, we constructed the data set `QBNormClusters`, with four columns for each quarterback's aggregated performance on the four meta-features listed above (Power Run, Accuracy, Power Throw, and Speed). After renormalizing the data set, we performed another clustering, experimenting with the number of clusters to use and settling on six, as follows:

- Scrambler: High Speed, High Power Run, Slightly Low Power Throw, Very low Accuracy

- **Balanced:** High Accuracy, Slightly Low Power Run/Power Throw/Speed
- **West Coast:** Average Power Run, Average Accuracy, Low Power Throw, High Speed
- **Deep Ball:** Average Power Run, Slightly High Accuracy, Very High Power Throw, Very Low Speed
- **Pocket Passer:** Low Power Run, Extremely High Accuracy, High Power Throw, Extremely Low Speed
- **Inaccurate Big Arm:** Average Power Run, Low Accuracy, High Power Throw, Average Speed

With these archetypes constructed, we scored every player's normalized attributes against these archetypes. We did this as follows:

1. Convert the cluster means for each archetype into cluster weights of each meta-feature. To do this, we took the pnorm (cumulative density function) of each value and normalized so that each row summed to 1. For example, the cluster weights for Scrambler were constructed by taking the pnorms of (0.357, -0.925, -0.144, 0.712) to get raw weights of (0.639, 0.177, 0.443, 0.762) and then dividing by the sum to get the overall weights of $0.316 \cdot \text{Power Run} + 0.088 \cdot \text{Accuracy} + 0.219 \cdot \text{Power Throw} + 0.377 \cdot \text{Speed}$.
2. Sum up the original z-scores multiplied by the cluster weights to get raw z-score-based scores for each player on each archetype.
3. Take the p-norm of this number and multiply by 100 to return to a 0 to 100 scale based on a normal distribution (50 is average).

3.1 Example Calculation

Ryan Tannehill has z-scores of 0.934 on Power Run, 0.845 on Accuracy, 0.647 on Power Throw, and 1.055 on Speed. Multiplying these by the Scrambler weights and adding them up gets 0.909. Take the pnorm, multiply by 100, and round to two decimal places, and Ryan Tannehill's score on the Scrambler archetype is 81.84.

4 Results and Tables

(insert R Markdown document)

5 Analysis

The results reveal the fundamental problem of attempting this analysis without reference to overall scores or any other expert metric. We are effectively treating all the aspects of being a quarterback as starting out as equal, and only then being adjusted based on clustering. Actual passing ability is hardly more highly weighted than stiff arming or carrying the ball, and sometimes given less weight.

The result is a set of results that simply do not pass the eye test. Cam Newton and Aaron Rodgers excel under all circumstances, as we would expect, but running specialists like Terrelle Pryor and Johnny Manziel end up scoring higher than Peyton Manning and Tom Brady, not just as scramblers (which might be justifiable) but even as "Balanced" quarterbacks. There are many more examples of this fundamental truth; the cluster analysis by itself is not sufficient without some reference to the fact that some attributes are in truth more important than others.

6 Conclusions and Future Research

We conclude that reference to the overall grade or some other expert-vetted measurement of performance is necessary for our clustering analysis to make sense. Thus future research will return to the method outlined in the previous paper.

However, we note that in cases in which expert knowledge proves unavailable or unnecessary, the tactics we have used here to weight the clusters could be revisited.

References