Optimizing Preseason Fantasy Football Rankings

John O'Hollaren (jpo4)
Department of Electrical and Computer Engineering

Abstract

Fantasy football is a fast growing, multi-billion dollar industry centered around projecting which players will perform the best over the course of a season. Correctly predicting which players will play well greatly improves your team's year-long performance, whereas wasting a high pick on a low achiever can ruin your season. This paper focuses on an optimal strategy for predicting a list of the top 30 non-rookie NFL wide receivers for a season, using only data available prior to the start of the season.

which NFL wide receivers will have the best fantasy football season using data available prior to the beginning of the season.

1 Introduction & Related Work

According to the Fantasy Sports Trade Association, over 10 percent of the US population spent 12 billion dollars playing fantasy football in 2012. Given such a large market, a slight statistical advantage could provide significant profits in fantasy football leagues. Specifically, a correct pre-season ranking of wide receivers (WRs) will allow owners to pick the best players to join their teams during the pre season fantasy football draft, setting them up for year-long success. However, ranking players is difficult. Preseason rankings from teams of experts at ESPN and Yahoo are correct for only a small minority of players. This paper compares several machine learning techniques for ranking WRs, including linear regression, feature reduction, K-Means clustering, mixture models, and PCA. A PCA regression technique which outperforms ESPN and Yahoo experts is presented.

While countless fantasy football websites exist, the major players are ESPN, Yahoo, the NFL, 4for4, and FantasyPros. None of these entities releases their methods publicly. Tim Mathews at the University of Southampton[1] explored using Bayesian reinforcement learning for week-to-week team management during the season. Matt Bookman at Stanford used linear regression to predict week-to-week performances during the season with limited success only in certain weeks[2]. The week-to-week prediction problem, however, is simplified through the use in-season data from each player, whereas in the pre-season, year-long ranking problem, predictions must be made for players who are 8 months removed from public games. Some attempts at machine learning approaches for the year-long ranking problem have been made[3], but did not outperform human experts. Other attempts use machine learning to outsmart other players in the same league[4].

1.1 Data Summarization

My dataset includes NFL statistics for over 700 players for every year from 2007 to 2012, however only a subset are used for prediction. There are 13 features in total: Receiving Yards, Receiving Touchdowns, Receiving Targets, Receiving Catches, ESPN Preseason Ranking, Yahoo Preseason Ranking, $\Delta 1$ year Receiving Yards: (Yards in year Y-1) - (Yards in year Y-2), $\Delta 1$ year Receiving TDs: (TDs in year Y-1) - (TDs in year Y-2), $\Delta 2$ year Receiving Yards: (Yards in year Y-1)-(Yards in year Y-3), $\Delta 2$ year Receiving TDs: (TDs in year Y-1) - (TDs in year Y-3), $\Delta 3$ year Receiving Yards: (Yards in year Y-1) - (Yards in year Y-4), $\Delta 3$ year Receiving TDs: (TDs in year Y-1) - (TDs in year Y-4), Fantasy Points in Previous Year: $6 \cdot \text{TDs} + 0.1 \cdot \text{Yards}$

---

[1]Mathews, Tim, et. al. Competing with Humans at Fantasy Football: Team Formation in Large Partially-Observable Domain. Association for the Advancement of Artificial Intelligence, 2012.

[2]Bookman, Matt. "Predicting Fantasy Football - Truth in Data." Stanford CS 229 Final Project. December 14, 2012.

[3]Kapania, Nitin. "Predicting Fantasy Football Performance with Machine Learning Techniques." Stanford CS 229 Final Project. December 14, 2012

[4]Becker, Adrian, et. al. "An Analytical Approach for Fantasy Football Draft and Lineup Management." MIT ORC Technical Report. November 2010.

1.2 Performance Criteria and Baseline

Discounted cumulative gain (DCG) will be used to assess WR rankings. It rewards ranking good players highly and penalizes ranking bad players too high. A higher DCG is better:

$DCG = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2(i)}$, and the relevance weights are: $rel_i = \frac{1}{\text{actual end of year rank}}$

Using DCG, the dataset can be analyzed by looking at the performance of ESPN, Yahoo, and simple regression across the years.
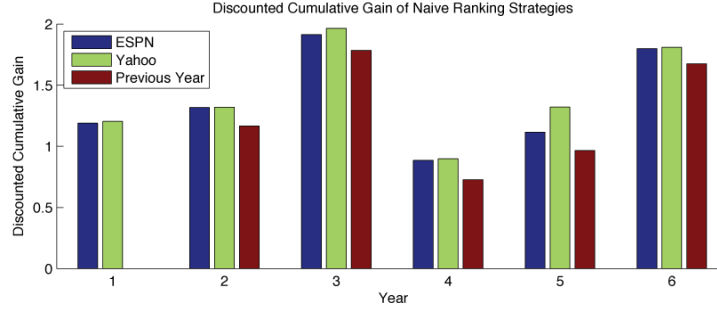

Figure 1: : DCG of ESPN / Yahoo preseason rankings over the years, along with a nave method of using the previous years final rankings as the next years preseason rankings.

2 Models and Methods

Linear Regression: β will be calculated using X_{train} as all data up until 2010, and Y_{train} as the 2011 end-of-season performances. X_{test} will then be all data up until 2011, and the resulting $Y_{test} = \beta X_{test}$ will be the 2012 end-of-season performance which will be compared with the actual, ESPN, and Yahoo results using DCG. All features will be used for this 13-dimensional β.

$\beta = (X^T \cdot X)^{-1} \cdot X^T \cdot Y$

$\beta = [w_{Points}, w_{Yds}, w_{TDs}, w_{Targets}, w_{Catches}, w_{\Delta 1yrYds}, w_{\Delta 1yrTds}, w_{\Delta 2yrYds}, w_{\Delta 2yrTds}, w_{\Delta 3yrYds}, w_{\Delta 3yrTds}, w_{ESPN}, w_{Yahoo}]$

Feature Reduction: Feature reduction will be utilized to remove features that may not be necessary for prediction. This helps reduce computation overhead, allows the results to be easier to interpret because they are lower dimensional, and also can help reduce overfitting. Different subsets, L, of features will be experimented with for linear regression using $\beta_L = [w_1 \dots w_L]$.

K-Means Mixture Model: 2 and 3 dimensional K-Means will be used to separate players in different groupings, and linear regression will be run on each of the K mixtures individually. Each mixture, $1 \dots K$, will have it's own learned weights, $\beta_1 \dots \beta_K$. Both the features used for K-means and the number of features used for regression will be varied.

PCA Regression: PCA will be used to estimate the regression coefficients. Different numbers of principle components will be experimented with. PCA on X gives a principle component score, $\Omega_{n \times p} = $ a representation of X in the principle component space, and also the principle component loadings in each column of $\Lambda_{p \times p}$. β is initially estimated as before, but by replacing X with Ω and mean centering Y to give: $\beta = (\Omega^T \cdot \Omega)^{-1} \cdot \Omega^T \cdot (Y - \mu_Y)$. Then to make the results easier to interpret, β is transformed to regression coefficients for uncentered variables by adding a mean term and adding a column of 1's to X during prediction: $\beta = [\mu_Y - \mu_X \cdot \Lambda \cdot \beta | \Lambda \cdot \beta], Y = [1's|X] \cdot \beta$.

3 Results

Using linear regression, β is calculated and then used to predict point totals for each player at the end of the next year. The top 30 players are then ranked by their point totals, and the ranking is scored using DCG. Note that as we use more training data, the RMSE of the point predictions reduces, which in turn leads to the DCG increasing as shown in Figure 2.

The linear regression results can be improved by picking out specific features to perform regression on. Multiple combinations of features were experimented with. The best results were shown with less than the full 13 features. Specifically, 2, 3, and 7 feature regression worked well on the features as shown in Table 1.

Feature reduction greatly improves the results! However, performance is still short of experts at ESPN and Yahoo. Now the method takes advantage of the fact that there are many types of receivers in the NFL, including those who play in the slot and catch lots of short passes, those who play out wide and catch very few longer passes, and all-stars who do it all. This insight prompts clustering receivers by type period to prediction, which is done using K-Means as shown in Figure 3.
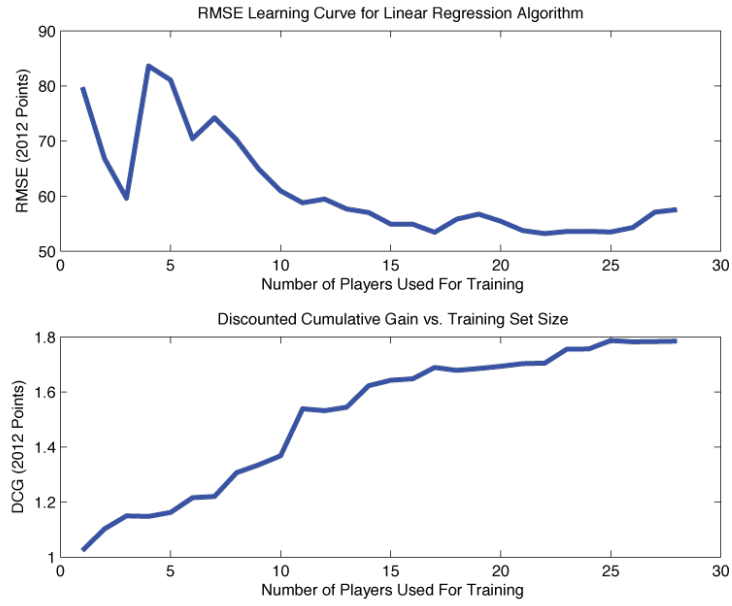
Figure 2: Graph of RMSE and DCG as the training set size increases. Results improve with larger training sets.

# Features	Features	DCG
13	Points, Yds, TDs, Targets, Catches, $\Delta 1yds$, $\Delta 1yds$, $\Delta 2yds$, $\Delta 2tds$, $\Delta 3yds$, $\Delta 3tds$, ESPN, Yahoo	1.7179
7	Points, Yds, TDs, Targets, Catches, $\Delta 1yds$, $\Delta 1yds$, ESPN, Yahoo	1.7722
6	Points, Yds, TDs, Targets, Catches, $\Delta 1yds$, $\Delta 1yds$, ESPN	1.7621
3	Points, Yds, TDs	1.7806
2	Points, TDs	1.7896
2	Points, Yds	1.7863

Table 1: Results of feature reduction showing improved results in lower dimensional spaces.
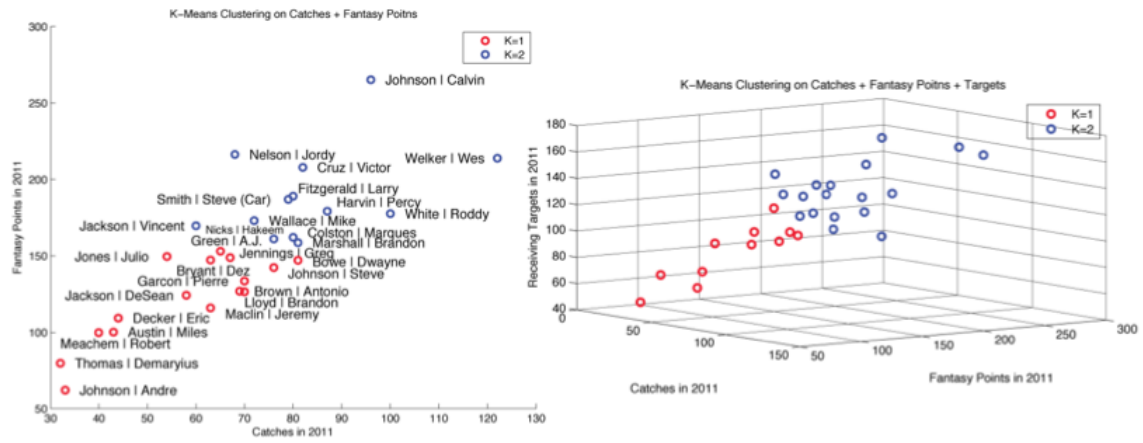
4   Conclusions

Figure 3: Results of k-means clustering in 2 and 3 dimensions.

5 Bibliography