Predicting Optimal Game Day Fantasy Football Teams

Glenn Sugar and Travis Swenson
Department of Aeronautics and Astronautics
Stanford University
gsugar@stanford.edu, swens558@stanford.edu

1 Introduction

The popularity of fantasy sports has exploded in recent years. Websites such as Draft Kings and FanDuel offer weekly competitions where people pay to build a fantasy team under a salary constraint for the chance to win money. Each team must have one quarterback (QB), two running backs (RB), three wide receivers (WR), one tight end (TE), one kicker (K), and one team defense (D). While machine learning has been applied to fantasy sports prediction, very little has been published as these algorithms are usually proprietary [1]. The goal of this project is to use machine learning techniques to obtain positive expected returns when playing these games. In order to do this we break our project into two parts. The first uses machine learning algorithms to predict the number of points any given player will score in a given game, while the second uses convex optimization to assemble teams with maximum expected return and minimum risk.

2 Dataset and Feature Selection

In order to accurately predict the performance of every player, we naturally considered their individual histories. However, it is also important to consider the history of their opponents, as a player is more likely to have a good game against a poor defense than against a good defense.

Injuries also play an important role in predicting a player's performance. If a star defensive player is injured, the opposing team's offense is likely to perform better. If a WR is injured, other WRs on the team might see more targets and their projected points might increase while the QB's projected points might decrease. We quantify the impact of an injured player by creating features that contain the total game averaged stats for all injured players. For example, if two WRs are injured and they have averaged 10 and 5 targets per game respectively, all of their teammates would have 15 for the "injured targets" feature.

We used the urllib and BeautifulSoup Python libraries to access and parse the data on Pro-Football-Reference and Rotoworld for every game that active players have played since 2010 and recorded their performance, the injured players, and the opposing team's average performance [2,3]. This resulted in over 40 statistics for every active players performance in every game played since 2010.

The statistics were grouped into 3 basic feature types: career features, current features, and recent history. Career features consist of running averages of player performance metrics over their entire career (e.g. average rush yards per game). Current features give information about the game we are trying to predict (e.g. home vs away). Recent history consists of performance metrics of the player and their opposition in the n most recent games (e.g. rushing touchdowns in each of the most recent 5 games). This data could either be averaged or included as n individual features per statistic. We then process the data so that each feature has zero mean and unit variance.

To select the optimal subset of features we used forward search selection to obtain the best feature vector for each position. We also experimented with n, the number of games to consider in a player's recent history ($n = 5$ gave the best results). We considered several machine learning algorithms,

see Section 3, but found the optimal features were relatively insensitive to the machine learning algorithm used. An example set of features selected using our forward selection algorithm is given in Table 1.

Table 1: Features selected for RB using forward search. Colored boxes indicate the features used for each feature type.

	Recent History	Current Game	Career
FD Points			
Targets			
Rush TDs			
Receptions			
Reception TDs			
Injured Rush Attempts			
Injured Pass Attempts			
Injured Targets			
Defensive Turnovers			
Game Location			
Defensive Points Allowed			
Defensive Rush Yards Allowed			
Opposition Offensive Points			

3 Model Selection

Initially we tried to train a hypothesis for each player individually, in the hopes of achieving the most accurate predictions possible. Unfortunately this resulted in high variance, and we were forced to abandon this approach. Instead we grouped the players by position, thus generating six hypotheses for the six positions of the FanDuel team. Training and testing sets were created by randomly dividing every game played by every player into 70% training examples and 30% testing examples, where a train/test example is a single game. We experimented with many machine learning algorithms using the Scikit-Learn Python package and found Ridge Regression, Bayesian Ridge Regression, and Elastic Net gave the lowest root mean squared error (RMSE) [4].

3.1 Ridge Regression (RR)

$$\underset{w}{\text{minimize}} ||Xw - y||_2^2 + \alpha ||w||_2^2 \tag{1}$$

The objective here is the same as linear regression, except for the addition of an L_2 penalty on the feature weights, w, weighted by α. We found $\alpha = 0.1$ works well.

3.2 Bayesian Ridge Regression (BR)

$$p(w|\lambda) = \mathcal{N}(w|0, \lambda^{-1}\mathbf{I_p}) \tag{2}$$

Bayesian Ridge Regression is similar to ridge regression, however the weighting on the L_2 norm is chosen based on the data supplied, thus it is more robust to changes in the feature vector. The normalization coefficient is chosen using the assumed prior on w, given in equation 2.

3.3 Elastic Net (EN)

$$\underset{w}{\text{minimize}} \frac{1}{2m} ||Xw - y||_2^2 + \alpha\rho ||w||_1^2 + \frac{\alpha(1-\rho)}{2} ||w||_2^2 \tag{3}$$

EN is similar to Ridge Regression, but includes an additional L_1 penatly. The parameters α, and ρ set the relative weights of the penalties. We found $\alpha = 0.5$ and $\rho = 0.1$ works well.

3.4 Bias vs Variance

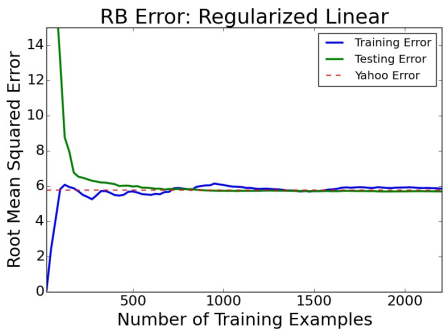

Figure 1: Test and training root mean squared error (RMSE) for the regularized linear model using holdout cross validation for RB. 30% of the data was used in the test set.

In order to diagnose the role of bias and variance in our generalization error we ran our algorithms on training sets of increasing size and recorded the resulting training and testing error. As can be seen in Figure 1, both errors converge to roughly the same value when using a large number of training examples, indicating variance is playing a low role in our generalization error. Instead, bias error is limiting the performance of our algorithms. This indicates we must further improve our model in order to reduce error.

3.5 Final Model Selection

After running forward selection and using hold out cross validation, we selected the best models and features for each position as given in Table 2. For each position, we used the regressor and the feature set (either all possible features or the features chosen by forward search) that gave the lowest RMSE for the test set used during cross validation.

Table 2: The optimal model used for each position. Note that "limited" features correspond to the feature set selected in the forward search, while "all" corresponds to using all possible features.

	QB	**WR**	**RB**	**TE**	**K**	**D**
Regressor	BR	BR	EN	BR	EN	BR
Features	All	Limited	All	Limited	All	All

4 Results

4.1 Error Comparison

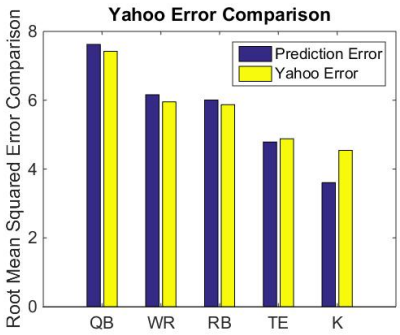

Figure 2: RMSE comparison between our projections and Yahoo's.

We evaluated the performance of our algorithms by comparing the RMSE of our predictions with the publicly available Yahoo projections. Yahoo is a multi-billion dollar company that is active in the fantasy football market, and therefore provides a good baseline for the state of the art in fantasy

sports prediction. As can be seen in Figure 2, our RMSE values are only slightly worse for QBs, WRs, and RBs (2.7%, 3.5 %, and 2.3% respectively), while we beat Yahoo for TE and K (2.0 % and 20.7% respectively). Note that team defense is omitted due to difficulties in obtaining Yahoo defense projections. Because our RMSE values are so close to Yahoo's (and in some cases even better) we consider our point predictor a success.

4.2 Team Optimizer

After computing the predicted points for every player, we can now construct the optimal team. FanDuel gives users a \$60,0000 salary to use when building a team of 1 QB, 2 RBs, 3 WRs, 1 TE, 1 K, and a team's defense. We can formulate the team selector as a binary linear program to pick the team that will produce the maximum number of predicted points:

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & f^T x \\ \text{subject to} \quad & x_i \in \{0,1\},\ i = 1,\ldots,m. \\ & Ax = b \\ & G^T x \leq h \end{aligned} \tag{4}$$

where f_i is the predicted points for the i^{th} player, x_i indicates whether player i is chosen, b contains the position requirements, A_{ji} indicates if player i plays position j, h is the maximum salary of the team, and G_i is the i^{th} player's salary given by FanDuel.

Many of the FanDuel tournaments are structured such that the top 50% of entrants win a fixed prize, and thus there is no incentive to do better than the winning threshold. In this case it makes more sense to minimize risk, subject to the constraint that the winning threshold is met. Fortunately, FanDuel published the average number of points required to win these leagues for the 2014 season (111.21 points) [5]. We use this threshold to solve the Markowitz Portfolio Optimization Problem to construct a team with minimal variance [6].

The problem now becomes:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & v^T x \\ \text{subject to} \quad & x_i \in \{0,1\},\ i = 1,\ldots,m. \\ & Ax = b \\ & G^T x \leq h \\ & f^T x \geq P \end{aligned} \tag{5}$$

where v_i is the variance of player i and P is the minimum number of total points for the team. Figure 3 shows the results from using both optimization methods on weeks 3-9 of the 2015 season. The team picked using Equation 4 has a 71.4% winning percentage, while the minimum variance team picked using $P = 120$ had a 57.1% winning percentage. Even though the sample size is small, our point projections coupled with the team optimizer shows promise of providing positive expected returns when playing FanDuel games.

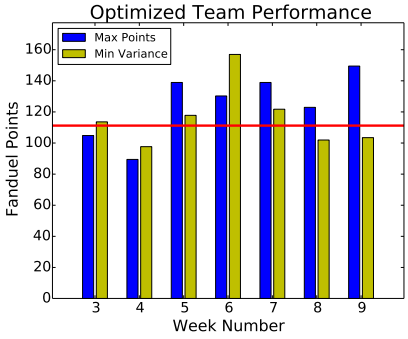

Figure 3: The performance of teams picked using both optimization methods (maximum points and minimum variance). The minimum variance team was chosen to have at least 120 total predicted points. The horizontal red line shows the average required points needed to win a FanDuel league.

4.3 Prediction Improvement: Clustering Players

Even with the encouraging results so far, it could be possible to improve our predictions by recognizing that there are subsets of players within a position. For example, some RBs are often utilized as WRs, while others will rarely be targets for passes. By identifying and training on subsets of a position, we should be able to lower our overall error and potentially beat Yahoo's prediction across all positions. While we have not performed all of this analysis, we have investigated potential subsets in the WR and RB positions. Note that this is a very preliminary analysis and there is much work to be done.

Because some players that are classified as WRs by FanDuel are often utilized as RBs (and vice versa), we looked for clusters within the WR/RB set of players. First, we normalized the data to have zero mean and unit variance for seven features per player. We then used principal component analysis (PCA) to extract the first 2 principal eigenvectors and projected the 7 dimensional RB and WR scaled data onto the reduced 2 dimensional PCA space. The left plot of Figure 4.3 shows the player classification used for our current predictions, and the right plot shows a new classification using K-Means with K=5.

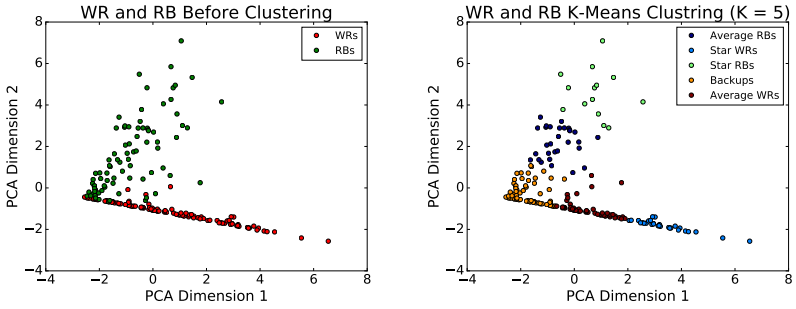

Figure 4: Initial WR and RB classification (left) and classification after K-Means clustering (right). The dimensionality was reduced to 2 via PCA for plotting purposes.

These plots have many interesting features. The distribution of WRs and RBs suggests that the x-axis roughly corresponds to a player's receiving ability, while the y-axis corresponds to their rushing ability. The player with the largest x coordinate is Odell Beckham Jr., and player with the largest y coordinate is Adrian Peterson. These are arguably the best players at their respective positions. The points around (-2,0) correspond to both WRs and RBs that are not very active and have few receptions and rushing attempts. These are mostly backup players, and their production is likely to be affected by injuries much differently than a normal starter. This is because if a starter gets injured, a backup player has a chance to start and have a drastic increase in points, whereas a starter would not notice much of a difference if his teammate misses a game due to injury. Because of these differences, we can generate more effective subsets using clustering algorithms. Furthermore, we can do this without the risk of overfitting the models. Figure 1 shows that we can reduce the training sets by a factor of 3 without significantly increasing variance.

5 Conclusion and Future Work

We have demonstrated a machine learning approach to predict fantasy football player performance that is on par with state of the art methods. With our point predictions, we used a binary linear program solver to pick the optimum team given salary and position constraints. Using these methods, we obtained positive returns playing FanDuel games in weeks 3-9 of the 2015 season.

We also demonstrated a possible path to improve our point predictor by building models that use player performance rather than their prescribed position labels. We used K-Means clustering to obtain subsets of RBs and WRs that could produce better models than both Yahoo's and our current predictor. More work needs to be done to explore the performance of new models that are trained on the smaller player subsets, as well as how these subsets are generated via different clustering methods.

References

[1] Dunnington, Nathan. “Fantasy Football Projection Analysis”. Honors Thesis. University of Oregon. 2015. Web. <http://economics.uoregon.edu/wp-content/uploads/sites/4/2015/03/Dunnington_Thesis_2015.pdf>

[2] Pro-Football-Reference. Sports Reference LLC. <http://www.pro-football-reference.com/>.

[3] Rotoworld. NBC Sports Digital. <http://www.rotoworld.com/teams/injuries/nfl/all/>.

[4] Pedregosa *et al.* Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830, 2011.

[5] Gonos, David. “FanDuel NFL Benchmarks: Points to Target in Each Contest Type.” 7 Aug. 2015. Web. <https://www.fanduel.com/insider/2015/08/07/fanduel-nfl-benchmarks-points-to-target-in-each-contest-type/>.

[6] Markowitz, Harry. Portfolio Selection. The Journal of Finance 7.1 (1952): 77-91.