Predicting Margin of Victory in College Football Games

Isaac Haberman
Center for Data Science
New York University
New York, NY 10011
ijh216@nyu.edu

Stephen Carrow
Center for Data Science
New York University
New York, NY 10011
swc419@nyu.edu

Chris Rogers
Center for Data Science
New York University
New York, NY 10011
cdr380@nyu.edu

1 Introduction

Sports betting, the industry known best for the Black Sox Scandal and Michael Jordan's first retirement, has grown more lucrative to data scientists in recent years with the advent of better data collection methods at sporting events leading to rich data sets used in the analysis of athletic performances. In this project, we apply data science techniques to sports betting for NCAA football, the premier college football league in the United States. Of the sports bets made by participants, "betting the spread" is the most popular bet and therefore, its prediction will be our central task in this project. To bet the spread is to bet on the predicted difference in total points scored by the home and away teams, the spread, which is determined by a sports booking service along with the amount a better would need to bet to win $100, the odds. Ultimately the goal of deploying a predictive algorithm in this context is to maximize the monetary return of placing such bets. As an initial step towards this end, we will train a model to predict the margin of victory of future match-ups, which could then be used to inform the placement of bets.

1.1 Data

Our initial data-set included game results, several in-game stats for the major Division 1 teams, and odds across several markets for seasons between 2013 and 2016 inclusive. Given the limited games teams plays per year, we hypothesized, and showed, that substantially increasing the amount of data was beneficial to model training. While we could not replicate all statistics for a longer historical time period, we discovered and acquired several other data-sets with historical data leading back to 2001.

We spent time manipulating the "Ultimate Football Database" BlueSCar [2017], Reddit User's personal project. Despite the dubious authorship, it had many positive reviews, and the database had extensive drive and play-by-play data that seemed ideal for feature generation. Unfortunately the data proved to have too many errors and inconsistencies to provide reliable stats (play-by-play records were often repeated, drive results were mislabeled or combined, classifications changed significantly over the years, weeks mislabeled, etc). After discussions with the author and attempts to clean the data up ourselves, we eventually felt forced to abandon it. Its team name reference however proved to be reliable, and we continued to use it. It is a collection of team names, nicknames, and identifiers that allowed us to assign unique ids to each team.

Searching for other sources of historical data, we settled on the "Snoozle" data from Snoozle Sports Wagner [2017]. This is a collated set of stats available through a free public API. Each year's stats were downloaded as a CSV. The data was comprised of stats for each game broken down by home and visitor, including the final score, as well as a number of other game statistics (Team Name, Rushing Yards, Rushing Attempts, Passing Yards, Passing Attempts, Passing Completions, Fumbles Lost, Interceptions Thrown, Final Score, Season, Year, Month,Day,Week, Team ID, Team Conference, D1 Game). This data was collected per year for 2001-2017. We merged this data with team ids and conference data, and computed the week number for each game. There were a total of 13511 games between 239 different teams, almost all involving one Division 1 team. Bowl games were

included, and simply considered to be later "weeks" in the season. A series of sanity checks were run to ensure the data was internally consistent, which showed mostly positive results. Some missing or inconsistent data was corrected by referencing ESPN Sports website ESPN [2018], which most other resources view as authoritative. Starting in 2002 the site also provided a point spread for every game, which we similarly integrated. Finally, to distinguish the major Division I conferences, we scraped data from Sports Reference, and merged teams with their given conferences. Teams that were not members of major conferences were marked as 'NotMajor'.

1.2 Success Evaluation

To evaluate our models, we predict the margin of victory (home team score - visiting team score) for each game, and compute the mean square loss with the ground truth margin of victory. During model training, we use held out validation data, the 2016 season, to tune our models for generalization, leaving the previous seasons as training data. Since many of our features required at least one if not multiple games to have been played by a team to compute, we chose to use the first four weeks of each season for feature building only, and did not train or evaluate model performance on them.

We developed several naive models as baselines to improve against and to provide initial assessments of the difficulty in producing accurate results on our task. These naive models include the "Home Field Advantage" rule and a simple single feature linear regression. The "Home Field Advantage" rule simply predicts that the home team will win by 3.5 points. In our single feature linear regression, we compute the within season average point differential of the home games (up to the game preceding the sample of interest) for the home team and similarly the away team's within season average for away games. Using these two averages, we compute the "Difference of Average Point Differentials" by taking the difference between the home team average and away team average. As an additional benchmark, and high water mark, we used the median of the published spreads from up to 8 different professional book makers in 2013 to 2016 to train a single feature linear regression model, the "Market Baseline". These spreads sometimes contained missing data, which we imputed for a given game using an average of the available values for that game.

2 Feature Engineering and Performance Improvement Strategy

2.1 Rating Systems

Our effort to improve upon the baseline model focused on implementing a number of ranking systems that, through research, we found to perform well when predicting the winner of a game or the final season rankings of the major Division 1 teams in college football. These include, Elo - the original chess rating system, which we adjust for the use of point spread rather than game outcome; Mills - developed by Matt Mills, which uses Continuous-Time Markov Chains with scores as transition rates; Glicko2 - developed by Mark Glickman as an updated Elo system that uses rating reliability; Pythagorean Wins - an estimate of expected wins developed by Bill James. Many of these systems use team records to develop rankings, however we have attempted to select systems to implement that are sufficiently different in their approach to raking with the intention that each feature will provide new information to our model.

We implemented FiveThirtyEight's version of Silver [2017] Elo Ratings . Similar to the original chess ratings, these ratings are all initialized to the same value at the beginning of the training period, the 2001 season, and are updated with every win. To control for the difference in conferences, we initialized major teams with a rating of 1500 and 'NotMajor' teams with a rating of 1300, a number chosen by cross-validated testing performance. Like FiveThirtyEight, we incorporated the margin of victory into the ratings update and also produced predicted scores and probabilities of winning for each game. At the beginning of each subsequent season, we used mean regression to adjust the ratings, regressing the major teams and 'NotMajor' teams separately.

We also found Pythagorean Wins Lieblich [2017], the expected wins formula developed by Bill James, to be an appropriate ranking system for our task. Since, Pythagorean Wins is based on points scored and given up, we theorized that the Pythagorean Wins of the home and visitor teams might be powerful features. We used Football Outsiders variant, using a football specific exponent, and multiplying by games played at each iteration. Unlike Elo Ratings, we reset Pythagorean Wins each

year, but kept weekly markers for teams that over and underperformed according to the Pythagorean Wins.

The Mills systems Mills [2015] was of particular interest as it makes use of the final points scored by each team in a game as the transition weights in a Markov Chain, which leads to rankings based on the very components used to compute the margin of victory. To extract team rankings from this transition matrix we solve for the steady state distribution of the Markov Chain, which describes the long-term probability of being in each state, "a team", where better teams have lower steady state probabilities. For each season, we computed the first steady state from weeks 1-4 and use the ranking as a feature for week 5 games. We then updated the Markov Chain rankings each week for the remainder of the season.

We also implement the Glicko2 rating system. This rating system generates not only an estimate of team rankings, but also an estimate of the variance of those rankings, which is not produced by any of the other systems. The system works by initially assuming the same rating mean and variance for each team, although prior knowledge of team strength could be incorporated. Each week, the mean and variance of a team's rating are updated using the mean and variance of the opponents faced through that week, along with the outcome (win or loss) of each game as described by Glickman [2013]. As with the Markov Chain features, we use weeks 1-4 of each season to compute the first Glicko2 features and update the rating means and rating variances for each subsequent week.

In addition, we incorporate strength of schedule features into our training data. There are a variety of ways to capture strength of schedule, or the average strength of your opponents that season. One rather simplistic one we implemented was the BCS calculation Wikipedia [2018], which uses the team's win/loss record (wins/total games). It is a simple weighted average of a team's opponent's record, and their opponent's opponent's record or (2*OR + OOR) / 3. This was reset for each season, starting fresh with 0-0 records for all teams. Another version, the Ratings Percentage Index LAXPower [2017], which incorporates the team's record itself, was also used as a feature (.25 * TR + .5 * OR + .25 * OOR). We then repeated these two calculations, but for the "record" we instead used the games point differential (negative if they lost) averaged over the games played. This again was reset to 0 at the beginning of a season. The "point differential" record showed a decent improvement over a head to head comparison with the w/l record. Both were used in the complete model.

2.2 Error Analysis

We examined the Mean Square Error of our training and validation sets for each week of the season to determine our models performance throughout a season. From this analysis we found that, as expected, earlier weeks in the season are more challenging to predict as shown in Figure 1. Using this analysis we trained several different models using different configurations of pre-season ratings for the Markov Chain and Glicko2 features and compare our models' performance on validation. For the Markov Chain rating system we generated pre-season ratings by building the steady state probabilities for the prior season, but decreasing all margins of victory by $1/2$ by adjusting the winning team's score. Similarly, for the Glicko2 ratings, we initialized the current season ratings and rating variances with those from the final week of the previous season, but translated all ratings and variances towards the default rating and variance by a factor of $1/2$ the distance from the current values to these default values.

Figure 2 shows that using the Markov Chain and Glicko2 features initialized with pre-season ratings improves MSE approximately through week 11 of the validation set, but afterwards the un-initialized features outperform all other configurations. Our final model makes use of this insight and utilizes the pre-season initialized Markov Chain and Glicko2 features from weeks 5 to 11, and the un-initialized features for weeks 12 to 19.

3 Results

Our final model is a Kernalized Ridge regression using the RBF kernel. We performed a grid search to set the hyper-parameters for both the regularization parameter and the gamma parameter from the RBF kernel, which are set to 10^{-6} and 10^{-10} respectively. Thus only minimal regularization and RBF basis functions with high variance produce the best performance on this task. We also

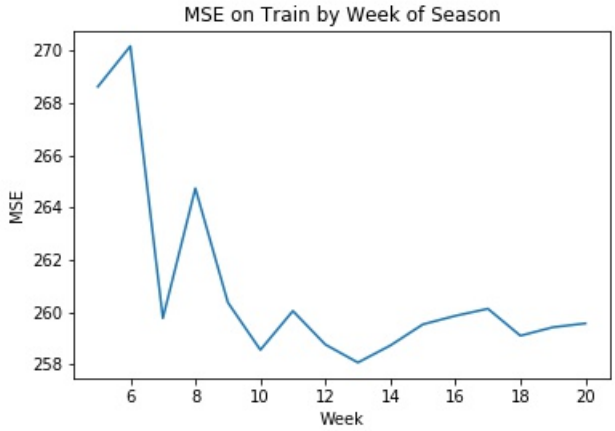

Figure 1: MSE by week on training data. The MSE decreases until approximately week 10 before stabilizing for the remainder of each season.

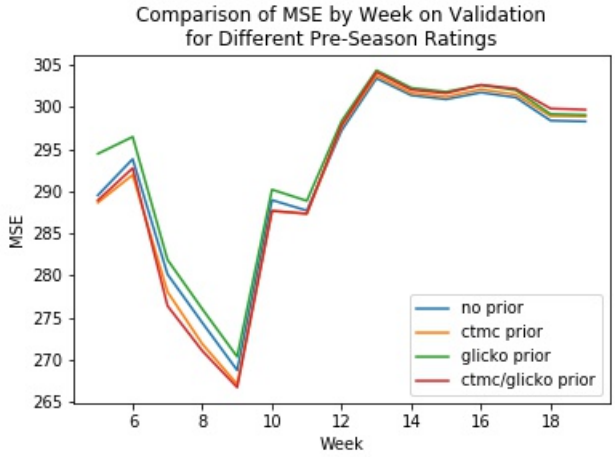

Figure 2: A comparison of the MSE by week on validation data using different configurations of pre-season ratings for Markov Chain and Glicko2 rating systems.

tested Linear Regression, Random Forest and Gradient Boosted Trees, however the Kernel Ridge outperformed these other approaches.

The Kernelized Ridge model improved substantially over the performance of both our naive baselines as shown in Table 1. We achieve an MSE of **284.55** on the Test data set, compared to an MSE of **500.80** and **366.88** for the "Home Field Advantage" and single feature linear regression baselines respectively. Our model showed a marked improvement in R-squared, slightly more than doubling the single feature regression model performance. We also compare our final model to the "Market Baseline", which is a linear regression using the median of the odds from up to 8 different marketplaces, but this comparison is performed on validation (not test) data and the models are trained using data from 2013-2015 only. Table 2 shows that our final model performs close to the "Market Baseline", which we view as a high water mark.

Table 1: Results on Test

Model	MSE	R-Squared
Home Field Advantage	500.80	-0.02
Single Feature Regression	366.88	0.175
Our Model	284.55	0.361

Table 2: Comparison to Odds Model on Validation

Model	MSE	R-Squared
Market Baseline	288.88	0.399
Our Model	302.70	0.370

While the weights of our Kernalized Ridge model are restricted to the dual formulation, and thus the solution is a weighted combination of the training points, we examine the weights of a Linear Regression to gain insight into the importance of the features we have generated. In Table 3 we show the features with a p-value ≤ 0.05 sorted by the absolute value of the learned coefficient. The Linear Regression spreads weight across many of our generated features. The Elo Spread is our most important feature in this model and many of the Pythagorean features are concentrated in the top half of the significant features.

Table 3: Learned Coefficients from Linear Regression

Feature	Weight	p-values	t-values
Elo Spread	4.82	5.44e-06	4.55
Visiter RPI	4.40	1.08e-125	24.27
Visiter Pyth Pct	-4.31	2.29e-09	-5.98
Home Pyth Wins	-3.97	4.78e-09	-5.86
Home Pyth Pct	3.92	4.28e-08	5.48
Home CTMC Rating	3.62	4.38e-13	7.26
Visiter Elo	-3.40	8.21e-06	-4.46
Visister Pyth Wins	2.73	3.87e-05	4.12
Visiter CTMC Rating	-2.65	2.66e-07	-5.15
Home Elo	2.50	1.13e-03	3.26
Visiter Win Pct	2.33	4.15e-04	3.53
Home Glicko Rating	2.27	1.09e-04	3.87
Away Glicko Rating	-2.18	1.58e-04	-3.78
Home Glicko Rating Deviance	-2.18	1.12e-05	-4.40
Away Glicko Rating Deviance	1.93	1.80e-04	3.75
Home Conf NotMajor	1.22	1.34e-04	3.82
Current Season Diff Avg Point Differential	0.69	1.69e-02	2.39

We also examined the correlation between our generated features as shown in Figure 3. There are several highly correlated features, in the range of 0.7 - 0.9, and multicollinearity in the Linear Regression is very likely. However, our use of Kernelized Ridge largely avoids the issue of multicollinearity as discussed in Wibowo [2009].

4 Challenges and Next Steps

Perhaps one of our biggest challenges was obtaining clean historical data. There is no easily digestible authoritative source for historical statistical data, while historical betting data is largely unavailable. Therefore, even the most reliable sources that we discovered contained errors, with limited methods for detection and correction. Perhaps, our greatest headache was the inconsistency between team names and abbreviations across data sources. We were often forced to adjust team abbreviations by hand to allow for merging. To that end, we used a few techniques for fixing the historical statistical data. For both the Snoozle and Ultimate CFB Database data, we manually checked for bad data, such as negative point scores, and checked the opposing values, and confirmed between the two data sources.

When training, the relatively few number of games played by each team and the infrequency of repeat matches made it difficult to learn much structure. Furthermore, the inherent variance of team performance in college football leads us to believe there is a low ceiling on the accuracy one can achieve when predicting the margin of victory for a given match. In addition, while certain teams are observed to be consistently better than others across seasons, there is variation in team performance from season to season and less information about team performance at the outset of each season.

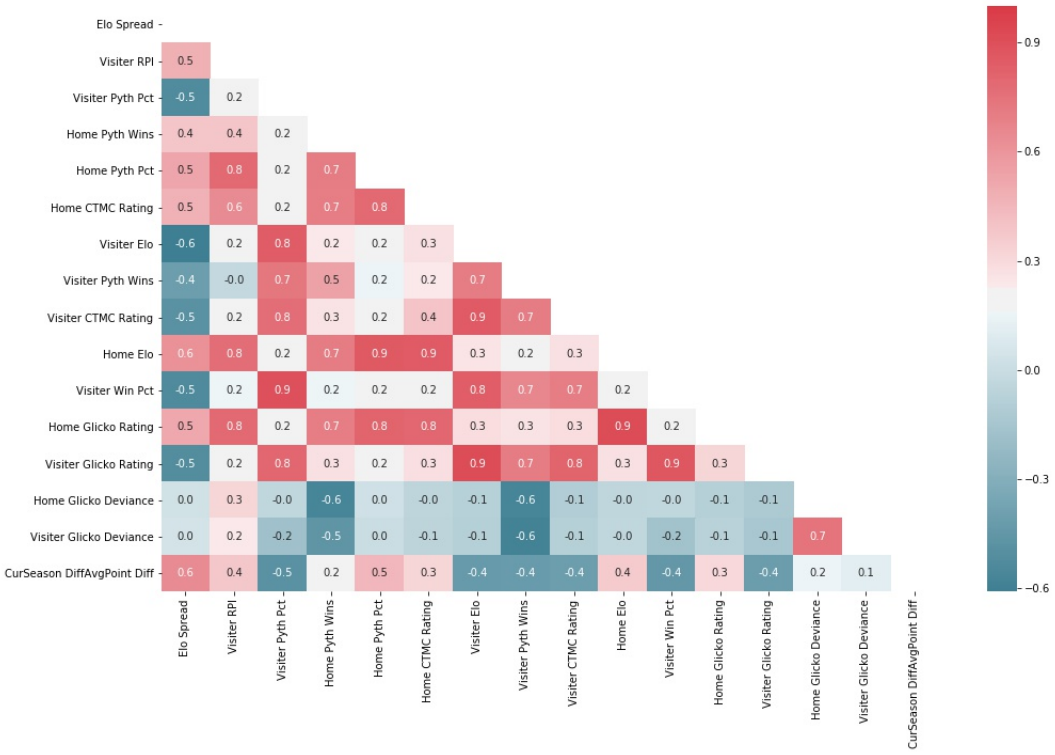

Figure 3: Correlation between generated features found to be significant in Linear Regression model.

Because of these limitations, we see higher Mean Squared Error in the early weeks of each season as shown in Figure 1. While we adjusted some of our ranking methods to handle early season predictions, the methods we implement are simple and a more sophisticated approach could lead to additional improvement in performance.

Our limited work with drive and play level data did not appear to increase our predictive power, but ideally more data in this area should prove beneficial. Obtaining a clean source of this data would allow us to experiment with various new features and implement features such as Adjusted Net Yards Per Passing Attempt, which requires play level information. We could also work to recreate some proprietary metrics such as Defense-adjusted Value Over Average (DVOA) or Clutch-weighted Quarterback Rating (QBR), which may provide additional predictive power but require detailed game statistics.

Much of the historical odds data is unavailable, perhaps at the behest of the casinos who do not want their this type of analysis done. Work was done to to use a recommender system to impute some missing data on a yearly basis. This showed good results on filling in missing odds for our full venue odds data (2013-2016), but could not be extended beyond that without more yearly odds data.

In addition to future efforts to improve predictive performance of our model, we can build on the current work by exploring different betting strategies given various constraints and utilize different model building strategies in order to simulate real-world use of our model. One possible model modification for that would be to train on a weekly basis within a current season, thus using the most up to date data. This presents interesting challenges in validation. We would also want to test different betting strategies, for example how to distribute a bankroll over the best M bets each week. We could add constraints such as restricting the bankroll to a specific weekly allowance or to a specific allowance per season. Finally, under the latter constraint it would make sense to test strategies for distributing bets throughout the season since our model generally performs better as each season progresses.

Our github repository for this project is located at Carrow et al. [2018].

References

BlueSCar. Ultimate college footbal database, 2017. URL https://www.reddit.com/r/CFBAnalysis/comments/7chhqs/cfb_database_week_11_updates/.

Stephen Carrow, Isaac Julius Haberman, and Chris Rogers. Predicting margin of victory, 2018. URL https://github.com/settonull/NYUMLProject.

ESPN. Fbs (i-a) schedule, 2018. URL http://www.espn.com/college-football/schedule.

Mark Glickman. Glicko2, Nov 2013. URL http://www.glicko.net/glicko.html.

LAXPower. Ratings percentage index, 2017. URL http://www.laxpower.com/update18/ex_rpi.php.

Ben Lieblich. pyth, 2017. URL http://www.footballoutsiders.com/2017/stat-analysis/presenting-adjusted-pythagorean-theorem.

Matt Mills. Using continuous-time markov chains to rank college football teams, 2015. URL http://thespread.us/continuous-markov-ratings.html.

Nate Silver. elo, 2017. URL http://www.fivethirtyeight.com/features/how-our-2017-college-football-playoff-predictions-work.

Gary Wagner. Snoozle sports, 2017. URL http://sports.snoozle.net/search/fbs/index.jsp.

Antoni Wibowo. Robust kernel ridge regression based on m-estimation. *Computational Mathematics and Modeling*, 20(4):438, Nov 2009. ISSN 1573-837X. doi: 10.1007/s10598-009-9049-7. URL https://doi.org/10.1007/s10598-009-9049-7.

Wikipedia. Strength of schedule, 2018. URL https://en.wikipedia.org/wiki/Strength_of_schedule.