NFL Score Difference Prediction with Markov Modeling

Guy Blanc[1], Eric Luxenberg[1], and Stanley Xie[1]

I. INTRODUCTION

Professional sports lends itself to the field of machine learning because of the natural goal of predicting the winner and score of a game. Additionally, sports strategy is already being merged with rigorous analysis to replace intuition with precise empirical motivation [3][4]. In this paper, we seek to predict score differences for American football games in the National Football League (NFL). We believe the sub-field of in-game score difference prediction is a valuable area of inquiry to build upon, and want to predict not only expected score difference, but a full probability distribution, whereas most current work focuses only on predicting an expected score difference.[1][2][5]. To construct such a distribution, we build a Markov model that simulates the fourth quarter of play. Previous work using Markov models seeks only to output expected drive outcomes, expected play conversion rates, or expected value of personnel changes [2][4][6]. Building a simulator for the fourth quarter logically can be extended to simulating the rest of the game. Additionally, having a simulation tool for the most crucial part of a game, the end, could provide a useful tool for coaches and analysts with some modification to allow visualizing different scenarios.

II. DATA AND PROCESSING

A. Dataset

We used an aggregated dataset of play-by-play data from the 2002-2013 NFL seasons [8][9][10]. The processed dataset contains a total of 467,199 plays in total. The 2012 and 2013 NFL seasons were separated to be used as our test set, and were not used for training our model.

B. Processing and Features

We combined data from multiple sources, each of which had slightly different schema. We preprocessed the data to put it all into the same form and then appended it with features, such as average yards per play for each team.

[1]G. Blanc, E. Luxenberg, and S. Xie are students at Stanford University gblanc at stanford.edu, ericlux at stanford.edu, stanleyx at stanford.edu

C. Handling Missing or Erroneous Values

There were occasionally cases in the data (a few hundred rows) where values would be missing or impossible values (i.e. score jumping from 17 to 42 in one play) would occur. We did not explicitly remove them from the dataset, but instead handled them in the coding of our algorithm. This decision was made because we did not want to have transitions between plays that may not have actually been next to each other in the real game, which could occur if we removed data. Instead, our algorithm just did not count transitions to and from erroneous states or states with missing values.

III. METHODOLOGY

A. Overview

To predict the score difference at the end of the fourth quarter, we imposed a Markov Model on the game of football. Using a Markov Model, we can simulate all possible play sequences for a given number of plays to generate a probability distribution across possible score differences. In order to do so, a game specific estimate for the number of plays in the fourth quarter is required. We model the number of fourth quarter plays as varying normally with mean and standard deviation linear in features from the first three quarters. With a probability distribution of the number of fourth quarter games, we simulate the fourth quarter by iterating through the Markov Model for each possible number of plays, weighted by its respective probability, to generate the final distribution across possible score differences. For a summary, see Figure 2.

B. Estimating the Number of Fourth Quarter Plays

We first made the simplifying assumption that the number of plays in the fourth quarter is independent of the actual plays that occur. This is not completely correct, as some plays always take no time and others tend to take more time, but it makes our model computationally feasible. The first model assumed that

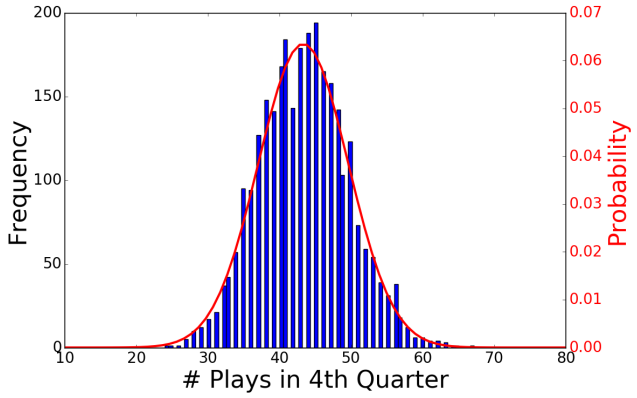

Fig. 1. The distribution of plays in the fourth quarter for all the games in our training set with a Gaussian distribution overlayed.

all games have the same constant number of plays in the fourth quarter, but to improve accuracy, we looked at estimating the number of fourth quarter plays as a distribution, rather than one value.

After reviewing the data, there appeared to be a normal distribution across the number of fourth quarter plays, as seen in Figure 1. Initially, we used a normal distribution with sample mean and sample variance. However, we desired higher model specificity that would give a different prediction for each test game. To do this, we then modeled the number of fourth quarter plays through a normal distribution whose mean was linear in features from the first three quarters and variance was just the sample variance. The MLE is equivalent to a normal linear regression with squared error. The likelihood of this model on the training set was slightly better than the likelihood of the model with constant mean and variance. Initially, we modeled the variance as linear in our features, but we later switched to the standard deviation to guarantee that the variance was positive. We found that this model outperformed the prior one in terms of log likelihood on the test data.

Based on the improvement gained by adding this model complexity, we considered tailoring the normal distribution even further to each game. To do this, we assume that the number of plays in the fourth quarter is normally distributed with mean and standard deviation linear in features from the first three quarters. That is, we assume if y^i is the number of plays in game i, and x^i is feature vector characterizing the first three quarters of game i,

$$y^i \sim N(\theta_1 x^i, (\theta_2 x^i)^2)$$

Using this model, we derive the partial derivatives with respect to θ_1 and θ_2.

$$L(\theta_1,\theta_2) = \Pi_{i=1}^{m} \frac{1}{\sqrt{2\pi(\theta_2 x^i)^2}} exp(\frac{-(y^i-\theta_1 x^i)^2}{2(\theta_2 x^i)^2})$$

$$LL(\theta_1,\theta_2) = \sum_{i=1}^{m} -log(\theta_2 x^i) - \frac{1}{2}log(2\pi) - \frac{(y^i-\theta_1 x^i)^2}{2(\theta_2 x^i)^2}$$

$$\frac{\partial LL(\theta_1,\theta_2)}{\partial \theta_1}_j = \sum_{i=1}^{m} \frac{(y^i-\theta_1 x^i)x_j^i}{(\theta_2 x^i)^2}$$

$$\frac{\partial LL(\theta_1,\theta_2)}{\partial \theta_2}_j = \sum_{i=1}^{m} \frac{x_j^i((y^i-\theta_1 x^i)^2-(\theta_2 x^i)^2}{(\theta_2 x^i)^3}$$

With the gradient fully characterized, we ran Stochastic Gradient Descent to find the optimal θ_1 and θ_2 via Maximum Likelihood Estimation. This new model outperformed both of the prior models by measure of log likelihood, so we utilized this method to predict the number of fourth quarter plays.

C. Markov Model

We first bucketed each play using positional data - namely what yard line the play starts at, the down of the play, number of yards needed for a first down, and which team had possession. We then preprocessed the the training data to first count the number of times each state s transitions to s_1, which is assigned to C_{s,s_1} and then determine transition probabilities as follows

$$P_{s,s_1} = \frac{C_{s,s_1}}{\sum_{s_2 \in S} C_{s,s_2}}$$

Where P_{s,s_1} is the probability of transitioning from state s to state s_1, and S is our state space. We also denote $R(s)$ to be the change in score difference, or reward, that occurs at position s. For most states, $R(s)$ is 0, but some, such as the state associated with a scored field goal, will have nonzero values. Note that when the algorithm is initialized it must be told for which team all scores give positive rewards, and for which team all scores give negative rewards.

Our goal with the Markov model is to predict the probability of a specific score difference occurring. We define $p(d|s,n)$ to be the probability of getting reward d given we are in state s with n plays left. It can be computed using the following recurrence relation:

$$p(d|s,n) = \begin{cases} \sum_{s_1 \in S} P_{s,s_1} \cdot p(d-R(s)|s_1,n-1) & n \geq 1 \\ 1 & n=0, d=R(s) \\ 0 & \text{otherwise} \end{cases}$$

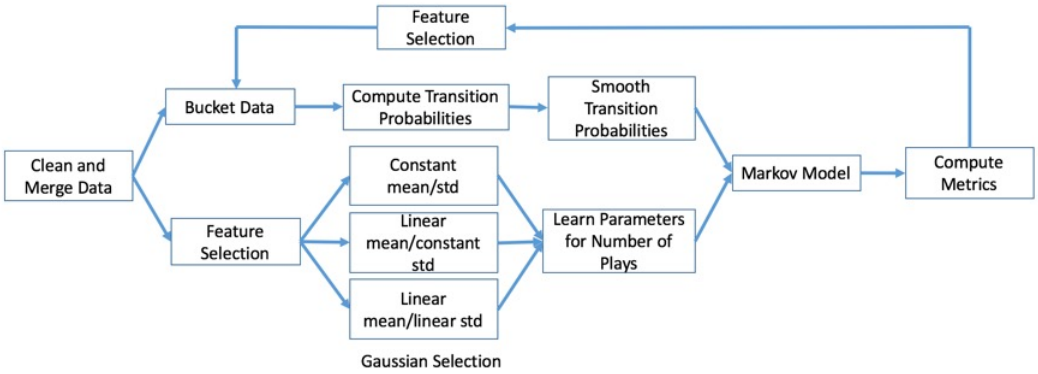

Fig. 2. Our pipeline from data acquisition to prediction/scoring.

Then, we introduced extra features from the first three quarters of data: the average yards per a play of the team currently on offense and score difference. To make the computation tractable, we assumed both of these features would remain constant throughout the fourth quarter. We discretized these features into 5 buckets each and then reran our model. The result was worse than without extra features. We noted three potential reasons for poor performance:

1) The model was overemphasizing the first three quarters - we examined for which games in the test set our model was performing most poorly and found that they were often games in which one team dominated during the first three quarters, but the fourth quarter was more evenly matched.
2) We assumed that the features were constant for the entire fourth quarter, but this is clearly not true (average yards per play changes after every play), and each feature would have a reasonable chance of moving to neighboring buckets over the course of the fourth quarter.
3) By introducing more features, we were increasing the number of buckets and decreasing the number of plays in each bucket. This increases the variance of our transition probabilities, potentially making them less accurate.

We decided on a solution that attempted to remedy all three of the potential problems. When computing transition probabilities, we would smooth over buckets neighboring the start state. We decided to not smooth on buckets that had different positional features (such as down), because if we did, some plays would jump from first to third down, which is not a legal NFL play. Instead, we smoothed over our non-positional features, such as average yards per play. More formally, we defined a smoothing function F that takes in two states in the same positional bucket and determines how close their non-positional features are. The output of F is then used to weigh our counts during the smoothing process. For each non-positional feature, we needed to tune one hyperparameter, τ_i, indicating how much we are smoothing feature i. We define $\phi(s)$ to be the non-positional features of s, and:

$$F(s,s_1) \sim N(\phi(s) - \phi(s_1), \Sigma)$$
$$\Sigma = \text{diag}(\tau_1^2, ..., \tau_m^2)$$

Given this smoothing function, we recompute our transition probabilities. First, we denote $B(s)$ to be the set of all states with the same positional features as s. Note that $B(s) \subset S$. Then:

$$C'(s,s_1) = \sum_{s_2 \in B(s)} F(s,s_2) C(s_2,s_1)$$

$$P'_{s,s_1} = \frac{C'(s,s_1)}{\sum_{s_2 \in S} C'(s,s_2)}$$

After tuning the smoothing parameters, and recomputing score different distributions with the smoothed transition probabilities, we found that the model indeed performed better with non-positional features than they did without them.

Finally, instead of assuming every game had the same number of plays in the fourth quarter, we combined our Markov model with our work on determining a game-specific distribution for the number of plays in the fourth quarter. For each game in the test set, we determined the Gaussian distribution for number of plays in the fourth quarter. We then ran our Markov model over the range of reasonable number of fourth quarter plays, weighing the results by the probability

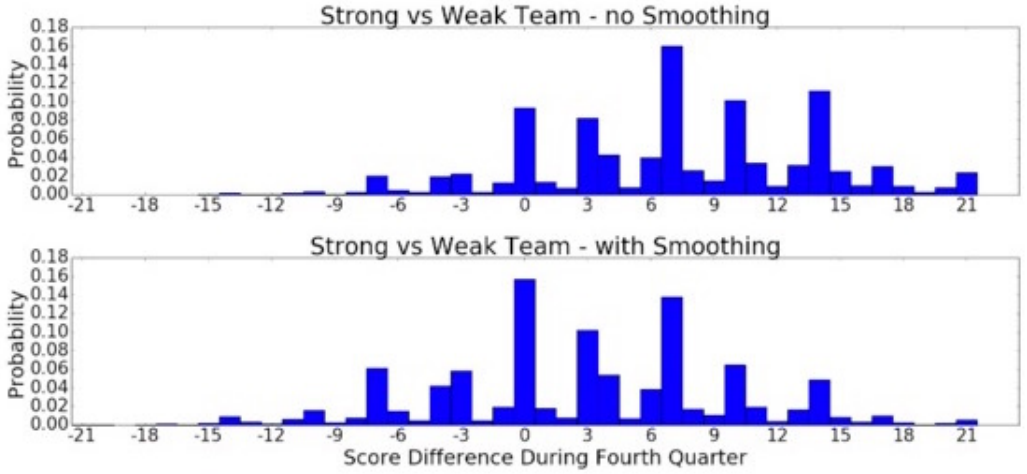

Fig. 3. A sample probability density plot for 4th quarter score difference between two teams of different strength. We see that smoothing causes reversion to the mean.

of the fourth quarter containing that number of plays, and produced one combined probability distribution.

IV. Results and Discussion

A. Metrics

We utilized three different metrics to quantify and compare our results. Metric 1 is the average squared error between the expected value of score difference in the fourth quarter generated by our model, and actual score difference. This is perhaps the most intuitive metric. Metric 2 is the mean log-likelihood assigned to the actual score difference over all test games. One of the key advantages of our model is that we generate a probability distribution rather than an expected value, a benefit captured in this metric. Thus, Metric 2 is the one we aimed to optimize. Our third, Metric 3, is the percentage of games where the actual score difference appeared in our top 5 score differences (ranked by probability).

B. Model Tuning and Evaluation

Algorithm	Metric 1	Metric 2	Metric 3
Naive	66.1013	-3.5145	0.2687
Position state, all quarters	51.2583	-3.0165	0.5374
Position state, 4th quarter	51.0961	-2.9934	0.5408
Extra features no smoothing	56.1134	-3.0152	0.5136
Extra features/smoothing	51.2906	-2.9698	0.5374
Extra features/smoothing, #plays est.	51.3062	-2.9692	0.5374

Fig. 4. Performance of various iterations of our model measured by the three metrics.

To provide a baseline, we created a naive model which took in the third quarter score difference and multiplied it by a factor of $\frac{4}{3}$ to get the mean of the predicted probability distribution. We then computed a variance by looking at the average variance between this naive prediction and actual score difference over all test games. The naive model was then simply a Gaussian with the above mean and standard deviation.

Our first model used only positional data, and we needed to tune the buckets into which we discretized each feature. If the performance is equal, using less buckets is advantageous because it is more computationally efficient and easier to train with less data. We determined that our model no longer improved with more buckets after using 5 buckets for yard line, 5 buckets for yards until a first down, and 4 buckets for down. As expected, even our most basic Markov model outperformed the naive model significantly.

We then tried training on only fourth quarter data. Since teams may play differently in the fourth quarter, due to tiredness, sportsmanship, or other factors, this could make our model more specific, but it also had the potential downside of decreasing the size of our training set. Overall, the results were positive, indicating that the difference between fourth quarter play and the other three quarters overshadowed concerns of losing training data.

Next, we added in extra non-positional features and actually got worse results, as discussed in the Markov model section. To remedy this, we introduced smoothing and tuned the smoothing parameters for each of our non-positional features one by one, as demonstrated in Fig. 5. The resulting model performed significantly better on Metric 2 than the one without non-positional

features. While still allowing non-positional features to impact transition probabilities, smoothing tends to limit the effects of either extreme for those features, as seen in Fig. 3.

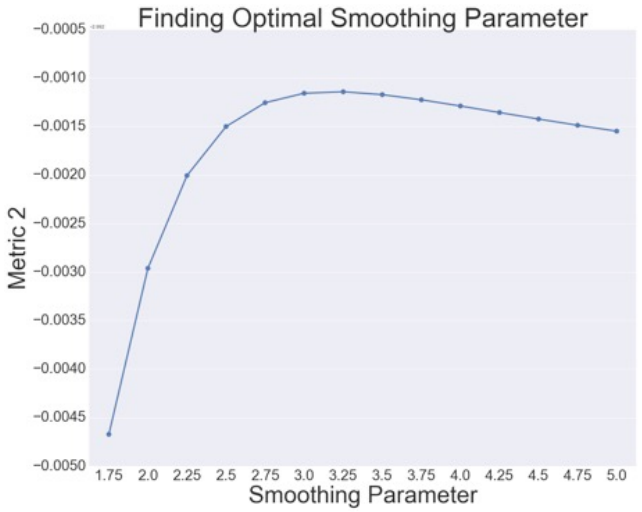

Fig. 5. Tuning the smoothing parameters for average yards per a play. At the limit of very large tuning parameter (standard deviation), average yards per play does not contribute to transition probabilities

Finally, we changed from assuming every fourth quarter had a constant number of plays, to combining our work on determining a game-specific distribution for number of plays in the fourth quarter. Doing so did improve Metric 2, but only slightly. This is likely because the change in distribution of number of fourth quarter plays did not vary too much from game to game, and that distribution does not have a huge effect on the distribution of score difference. Still, it is an interesting and beneficial addition.

V. Conclusion and Future Work

Our model provided a predictive edge for end of game score difference over a naive model but our additional features and model engineering only slightly improved our algorithm's performance. Overall, our model performs quite well in an intuitive sense, with over 50% of score differences in our test set being ranked in the top 5 most likely score differences by our algorithm.

There are several interesting further lines of inquiry. First, we would like to utilize even more in-game information, perhaps including previous plays in the game and allowing them to disproportionately affect the transition probabilities of the model. To do so, we would inflate transition probabilities for transitions that actually occured in the first three quarters of the test game by some factor, and potentially inflate neighboring transitions as well. Second, we would like to include prior information such as power rankings, momentum (e.g. win streaks), etc. into our model. This would make our model more extensible to earlier quarters in the game. The earlier we start our model, the less data we have from the game and the more important prior information becomes.

There are also two interesting extensions we can make to smoothing and how we bucket each play. Currently, we characterize each transition by its start state and end state, which is the simplest representation; however, this makes smoothing over positional data less effective. For example, if we were to smooth over down, then on first down we would have some probability of transitioning straight to third down. This is due to the fact that smoothing makes it possible to take the transition of neighboring buckets, so starting at first down it would be possible to take the transition of the neighboring second down bucket and end up at third down. A potential solution is to represent each transition as a difference. For example, a play can result in a gain of 5 yards and increase in 1 down. Doing so would require encoding many rules from football. For example, a transition from 3rd to 1st down is not equivalent to a transition from 2nd to 0th down even though both are minus 2 downs, but a transition from 1st to 2nd down is roughly equivalent to a transition from 2nd to 3rd down. Handling these cases is certainly possible.

After encoding transitions as differences, we can imagine another extension where buckets are not needed at all. Since there are no buckets, we would be unable to model all possible sequences of fourth quarter, and instead would need to do a monte carlo simulation of possible sequences of fourth quarter plays. At each step of the simulation, the algorithm would need to loop over all pairs of (starting state, transition) in the training state. For each of these pairs, the algorithm would use a smoothing function to give it some weight, and have a probability of picking each transition directly proportional to its weight. This has the potential to be very accurate and lose any disadvantages of discretization, but is also very computationally expensive. With this algorithm, the learning process would just be learning a smoothing function, which would likely boil down to learning a smoothing parameter for each feature.

REFERENCES

[1] Gimpel, K. (2006) Beating the NFL football point spread. Available at: http://ttic.uchicago.edu/~kgimpel/papers/machine-learning-project-2006.pdf (Accessed: 12 November 2016).

[2] Glickman, M.E. and Stern, H.S. (1998) A state-space model for national football league scores, Journal of the American Statistical Association, 93(441), p. 25. doi: 10.2307/2669599.

[3] Glickman, M.E. and Stern, H.S. (2016) Estimating team strength in the NFL. Available at: http://www.glicko.net/research/nfl-chapter.pdf (Accessed: 19 November 2016).

[4] Hirotsu, N. and Wright, M. (2002) Using a Markov process model of an association football match to determine the optimal timing of substitution and tactical decisions, The Journal of the Operational Research Society, 53(1), pp. 8896. doi: 10.2307/822882.

[5] Matthews, G.J., Wilbur, J.D. and Vernescu, B.M. (2005) Improving paired comparison models for NFL point spreads by data transformation. Available at: https://web.wpi.edu/Pubs/ETD/Available/etd-050505-144934/unrestricted/Thesis.pdf (Accessed: 12 November 2016).

[6] Pea, J.L. (2014) A Markovian model for association football possession and its outcomes. Available at: https://arxiv.org/pdf/1403.7993.pdf (Accessed: 1 December 2016).

[7] Warner, J. (2010) Predicting Margin of Victory in NFL Games: Machine Learning vs. The Las Vegas Line. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.402.8415&rep=rep1&type=pdf (Accessed: 28 October 2016).

[8] B. Burke, "Play-by-play data," 2014. [Online]. Available: http://archive.advancedfootballanalytics.com/2010/04/play-by-play-data.html. Accessed: Nov. 1, 2016.

[9] "NFLsavant.com: Advanced NFL statistics,". [Online]. Available: http://nflsavant.com/about.php. Accessed: Nov. 2, 2016.

[10] K. Inc, "Detailed NFL play-by-play data 2015," 2016. [Online]. Available: https://www.kaggle.com/maxhorowitz/nflplaybyplay2015. Accessed: Nov. 1, 2016.