A Model for College Football Season Simulations

Making Season Projections and Assessing Playoff Potential

by

Steven M. Pugh, MS

June, 2017

Introduction

In this paper, we examine one possible system for generating probabilities and statistics for NCAA Division I College Football. The model is broken down into a series of four processes which must be executed in order:

1. Generate team ratings based on game scores and preseason bias
2. Compute probabilities of team season outcomes
3. Perform many random season simulations and count the frequency of teams making the playoffs or championship game
4. Aggregate the data from Steps 1 through 3

The objective of this document is to describe the methodologies employed during each sub-process to produce the final statistics and probabilities. The following table depicts the various season projections and probabilities produced by the routines along with their method of derivation.

Projection/Probability	Derived From
projectedWins	Win/loss probabilities (Step 1)
projectedLosses	Win/loss probabilities (Step 1)
projectedConferenceWins	Win/loss probabilities (Step 1)
projectedConferenceLosses	Win/loss probabilities (Step 1)
projectedConferenceDivisionWins	Win/loss probabilities (Step 1)
projectedConferenceDivisionLosses	Win/loss probabilities (Step 1)
probabilityToWinNextGame	Win/loss probabilities (Step 1)
probabilityToWinConference	Poisson binomial (Step 2)
probabilityToWinConferenceDivision	Poisson binomial (Step 2)
probabilityToFinishWithJustOneLoss	Poisson binomial (Step 2)
probabilityToFinishWithTwoOrFewerLosses	Poisson binomial (Step 2)
probabilityToWinOut	Poisson binomial (Step 2)
probabilityToLoseOneMoreGame	Poisson binomial (Step 2)
probabilityToLoseTwoOrMoreGames	Poisson binomial (Step 2)
probabilityToMakePlayoffs	Monte-Carlo (Step 3)
probabilityToMakeChampionshipGame	Monte-Carlo (Step 3)

Step 1: Generate Team Ratings

The CompughterRatings model (http://www.compughterratings.com/theory [1]) uses an advanced proprietary mathematical model to produce a variety of ratings estimates for individual teams. A Linear Least Squares model is used to produce initial ratings estimates and those estimates are refined using a Maximum Likelihood technique. The ratings produced are overallPerformanceRating, offensiveRating, and defensiveRating. From these ratings, additional ratings are derived: powerRating, strengthOfScheduleRating, and futureStrengthOfScheduleRating. Ordinal rankings are then assigned to each team in each of these categories.

The algorithm takes into consideration the following factors:
- Game scores
- Margin of victory
- Home field advantage
- Strength of schedule
- Strength of opponents, opponents' opponents, etc
- Game recency (later games weighted higher)

Factors such as conference strength are implicitly included in the model.

The following factors are not included in the model:
- Field / weather conditions
- Player injuries or disciplinary actions
- Coaching changes
- Time of day
- Betting lines
- Officiating crews
- Any other external factors

Game Predictions

As noted above, the CompughterRatings model [1] generates an Offensive Rating and Defensive Rating for each team. These two ratings can be used together to make predictions about future games. To understand how these two ratings are computed, consider a series of n games played between a finite group of competitors of order p. If n is large enough and the p teams are all "connected" in the series, then we can use the score outcomes from the games played to predict future scores between any two teams. Furthermore, we can compute the probability of any given team defeating another team.

Let X represent the design matrix as follows:

$$X = \begin{bmatrix} & & & & \vdots & & & & \\ x_{1,1} & x_{1,2} & \cdots & x_{1,p} & \vdots & x_{1,p+1} & x_{1,p+2} & \cdots & x_{1,2p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} & \vdots & x_{2,p+1} & x_{2,p+2} & \cdots & x_{2,2p} \\ x_{3,1} & x_{3,3} & \cdots & x_{3,p} & \vdots & x_{3,p+1} & x_{3,p+2} & \cdots & x_{3,2p} \\ x_{4,1} & x_{4,2} & \cdots & x_{4,p} & \vdots & x_{4,p+1} & x_{4,p+2} & \cdots & x_{4,2p} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{2n-1,1} & x_{2n-1,2} & \ddots & x_{2n-1,p} & \vdots & x_{2n-1,p+1} & x_{2n-1,p+2} & & x_{2n-1,2p} \\ x_{2n,1} & x_{2n,2} & \cdots & x_{2n,p} & \vdots & x_{2n,p+1} & x_{2n,p+2} & \cdots & x_{2n,2p} \\ & & & & \vdots & & & & \end{bmatrix}$$

Here, X is a $2n$ x $2p$ matrix. The first p columns on the left-hand side of the matrix store the offensive rating coefficients for all p teams. Similarly, the right-hand side of the matrix stores the defensive rating coefficients for all p teams. For each game played, there are 2 rows in X. The first row represents how team i performed against team j and the second row represents how team j performed against team i. We will use values of 0's, -1's, and 1's to make these representations clear.

Example 1.1: Let's suppose we have a very simple series of $n = 3$ games between $p = 4$ opponents. To meet our criteria of connectedness, let's also assume that Team 1 played Team 2, Team 2 played Team 3, and Team 3 played Team 4. The hypothetical outcomes of these 3 games are depicted as follows:

Team 1 defeats Team 2 by a score of 45-28.
Team 2 defeats Team 3 by a score of 17-10
Team 3 defeats Team 4 by a score of 31-30

We can represent these equations in matrix form as

$$\begin{bmatrix} X_{off\,(6\,x\,4)} & \vdots & X_{def\,(6\,x\,4)} \end{bmatrix}_{(6\,x\,8)} * \begin{bmatrix} b_{off\,(4\,x\,1)} \\ b_{def\,(4\,x\,1)} \end{bmatrix}_{(8\,x\,1)} = \begin{bmatrix} y_{1\,(2\,x\,1)} \\ y_{2\,(2\,x\,1)} \\ y_{3\,(2\,x\,1)} \end{bmatrix}_{(6\,x\,1)}$$

And using our scores in the column vector y, we get:

$$\begin{bmatrix} & & & & \vdots & & & & \\ 1 & 0 & 0 & 0 & \vdots & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & \vdots & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \vdots & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & \vdots & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & \vdots & 0 & 0 & 0 & -1 \\ 0 & 0 & \dots & 1 & \vdots & 0 & 0 & -1 & 0 \\ & & & & \vdots & & & & \end{bmatrix}_{(6\,x\,8)} * \begin{bmatrix} b_{off_1} \\ b_{off_2} \\ b_{off_3} \\ b_{off_4} \\ b_{def_1} \\ b_{def_2} \\ b_{def_3} \\ b_{def_4} \end{bmatrix}_{(8\,x\,1)} = \begin{bmatrix} 45 \\ 28 \\ 17 \\ 10 \\ 31 \\ 30 \end{bmatrix}_{(6\,x\,1)}$$

From here, we can use techniques described in the Compughter Ratings Theory [1] to ensure our matrix equation is solvable for the ratings vector b. ♦

Now that we have our offensive and defensive ratings estimates, we can use those together to predict the game score between any two teams. First note that every game prediction is actually a pair of predictions: 1) Team A's offensive performance against Team B's defensive performance; and 2) Team B's offensive performance against Team A's defensive performance.

Let $team_i^{off}$ represent the offensive rating of $team_i$ and $team_i^{def}$ represent the defensive rating of $team_i$ for any $1 \leq i \leq p$. Then the number of points that $team_i$ is expected to score against $team_j$ is given by $team_i^{off} - team_j^{def}$.

Example 1.2: Suppose the offensive and defensive ratings for Team A and Team B are computed as in the following table:

Team	Offensive Rating	Defensive Rating
Team A	34.0111	2.7216
Team B	32.8420	10.2093

Then in a hypothetical matchup between Team A and Team B, the expected outcome would be:

Team A scores $34.0111 - 10.2093 = 23.8018$ points against Team B; and
Team B scores $32.8420 - 2.7216 = 30.1204$ points against Team A. ♦

Preseason Ratings

Early in the season when few or no games have been played, our assumption of "connectedness" between teams is invalid. Our matrix equations are simply not solvable. We can fix this by adding rows to our matrix equations to ensure teams are connected. Since little is known about any team's potential performance when no games have been played, preseason team ratings can be represented as a function of previous seasons' ratings, along with a Bayesian correction to account for the loss of key players. These terms can then be progressively dampened out as the season progresses and the teams become more and more connected.

To think about this in equation form, consider the fact that each team has two pieces of information to be added to our matrix: a preseason offensive rating and a preseason defensive rating. So with p teams, there would be $2p$ equations of data to add to our matrix equation. Furthermore, observe that adding rows to the design matrix X and the column vector y will not change the dimension of our solution vector b. To see this in matrix form, our previous equation

$$\begin{bmatrix} X_{off\,(2n\,x\,p)} & \vdots & X_{def\,(2n\,x\,p)} \end{bmatrix}_{(2n\,x\,2p)} * \begin{bmatrix} b_{off\,(p\,x\,1)} \\ b_{def\,(p\,x\,1)} \end{bmatrix}_{(2p\,x\,1)} = \begin{bmatrix} y_{1\,(2\,x\,1)} \\ y_{2\,(2\,x\,1)} \\ \vdots \\ y_{n\,(2\,x\,1)} \end{bmatrix}_{(2n\,x\,1)}$$

becomes

$$\begin{bmatrix} X_{off}{'}_{(2p\ x\ p)} & \vdots & X_{def}{'}_{(2p\ x\ p)} \\ X_{off\,(2n\ x\ p)} & \vdots & X_{def\,(2n\ x\ p)} \end{bmatrix}_{(2p+2n\ x\ 2p)} * \begin{bmatrix} b_{off\,(p\ x\ 1)} \\ b_{def\,(p\ x\ 1)} \end{bmatrix}_{(2p\ x\ 1)} = \begin{bmatrix} y_{1\,(2\ x\ 1)} \\ y_{2\,(2\ x\ 1)} \\ \vdots \\ y_{p\,(2\ x\ 1)} \\ y_{p+1\,(2\ x\ 1)} \\ \vdots \\ y_{p+n\,(2\ x\ 1)} \end{bmatrix}_{(2p+2n\ x\ 1)}$$

where

$X_{off}{'}_{(2p\ x\ p)}$ and $X_{def}{'}_{(2p\ x\ p)}$ represent the *2p x p* matrices of offensive ratings coefficients, respectively, and

$$\begin{bmatrix} y_{1\,(2\ x\ 1)} \\ y_{2\,(2\ x\ 1)} \\ \vdots \\ y_{p\,(2\ x\ 1)} \end{bmatrix}_{(2p\ x\ 1)}$$

represents the vector of offensive and defensive ratings pairs for each team $1 \le i \le p$.

Step 2: Forecast Season Outcomes

More complex season projections, related to a variety of scenarios and not related to a single game, are estimated using the Poisson Binomial Model. Examples of these include a team's probability to finish the season with just one loss or its probability to lose two or more of its remaining games.

Poisson Binomial Distribution

Since a team's win probabilities are computed each week (in Step 1) for all of its remaining opponents, Poisson's Binomial model was determined to be sufficient solution for predicting k wins or losses out of n remaining games. As a matter of computational convenience, the Poisson Binomial probabilities are estimated using a well-known recursive method [3] which is known to be stable when n is less than approximately 20 (which is always the case in College Football).

The recursive formula used for estimating the Poisson Binomial probability mass function [3] is given by

$$\Pr(K=k) = \begin{cases} \displaystyle\prod_{i=1}^{n}(1-p_i) & k=0 \\ \displaystyle\frac{1}{k}\sum_{i=1}^{k}(-1)^{i-1}\Pr(K=k-i)\,T(i) & k>0 \end{cases}$$

where

$$T(i) = \sum_{j=1}^{n}\left(\frac{p_j}{1-p_j}\right)^i$$

Using the estimated win probabilities (computed in Step 1) for every team against their future opponents, this formula can be used to estimate the probabilities of winning all remaining games, winning all but x remaining games, losing at most x games, and many other scenarios.

Sample PHP code to test out the Poisson Binomial recursive formula can be found in the following repository on Github:

https://github.com/stevenmpugh/PHP_Statistics/blob/master/PoissonBinomialDistribution.php

Step 3: Run Random Season Simulations

For predicting the probabilities of making the 4-team playoff or the national championship game, a proprietary Monte Carlo method is employed. Once team ratings are generated based on all games played, we can use random variations of team offensive and defensive ratings to produce a robust simulation of the remaining season. We can then simulate the remaining season hundreds or even thousands of times and rank the teams at the end of each season simulation based on factors that we believe represent the human components of team rankings (i.e., the College Football Playoff Selection Committee). After this process is complete, we are left with a subset of teams whose probabilities of making the 4-team playoff sum up to 400%. Similarly, we can identify the subset of teams most likely to finish in the Top 2, with their respective probabilities summing to 200%.

Monte-Carlo Simulations

Each Monte-Carlo iteration consists of three steps repeated hundreds or thousands of times:

1. Randomize offensive and defensive ratings for teams based on known input estimates
2. Simulate the remaining season using the randomized offensive and defensive ratings

3. Rank the teams at the end of the simulated season by a known set of quantifiers and log the results

Once all of the iterations have completed, we are left with many possible scenarios that are legitimate season outcomes. Moreover, not only do we know which outcomes are possible, but we also know the probability of each scenario occurring. For example, if Team A ended up in the Top 4 of the rankings 357 times out of 1,000 iterations, then the model would suggest that Team A has about a 35.7% probability of making the 4-team playoff. Similarly, if Team A ended up in the Top 2 of the rankings 135 times, then the model suggests that Team A has a 13.5% chance to make the Championship Game.

There are two parts of our Monte-Carlo method that involve more art than science:

- How much variance do we apply to the offensive and defensive ratings?
- How do we quantitatively assess the strength (and therefore, the rank) of each team at the end of each iteration.

We will look at each of these questions in the following sections.

Randomizing Team Strength

Suppose we have computed the offensive and defensive ratings for p teams (this is done in Step 1). Let s^{off} and s^{def} represent the standard deviations of the offensive and defensive ratings, respectively. Let $R(k,g)_i^{off}, R(k,g)_i^{def} \in (0,1)$ be two distinct randomly generated real numbers associated with the i^{th} team and the g^{th} game of the k^{th} iteration, where $1 \leq i \leq p$, $1 \leq k \leq K$, and K is the total number of iterations.

Then we can randomly adjust the offensive and defensive ratings for each team, each game, and each iteration using the inverse of the Cumulative Normal Distribution (NORMINV) function (sometimes referred to as the "probit" function [2]) as follows:

$$\left(team_i^{off}\right)' = NORMINV\left(R(k,g)_i^{off}, team_i^{off}, c \cdot s^{off}\right)$$

and

$$\left(team_i^{def}\right)' = NORMINV\left(R(k,g)_i^{def}, team_i^{def}, c \cdot s^{def}\right)$$

where c is a scalar chosen large enough to vary the ratings by the desired magnitude (i.e., the artistic choice).

By randomly adjusting each team's offensive and defensive ratings for every simulated game within every simulated season, and staying within a certain threshold of variance, we are able to introduce an element of random chance into every game. The next step is then to measure team performance based on each simulation's randomized outcome.

Measuring Team Performance

At the end of each iteration, we use a simple arithmetical technique to rate each team's performance and assign an ordinal ranking. This technique is comparable to the RPI (Rating Percentage Index) system utilized by the NCAA for ranking College Basketball teams. The strength factors employed and their associated weights can be specified in an infinite number of combinations, but we will aim to identify a system that produces reasonable results and keep it consistent throughout all iterations of the season simulations.

The following table depicts the factors included in the performance metric alongside one possible combination of weights to be used. Note that these factors and weights are currently in development and are subject to change.

Factor	Min	Max	Weight
Overall Performance Rank	1	128	1
Power Rank	1	128	1
Total Expected Wins (Actual + Projected)	0	13	10
Actual Top 10 Wins	0	5	40
Projected Top 10 Wins	0	5	20
Actual Top 25 Wins	0	5	20
Projected Top 25 Wins	0	5	10
Probability to Win Conference	.01	.99	128
Strength of Schedule	1	128	.5
Future Strength of Schedule	1	128	.5
Out of Conference Strength of Schedule	1	128	.5

It's important to note that there may be certain conditions which disqualify a team from being ranked at all. Since we are primarily interested in the Top 4 and Top 2 teams from each iteration, there's simply no need to rank a team which has already exceeded a certain number of losses, particularly when their schedule is known to be relatively weak. For example, in College Football, if a team such as Western Michigan who plays in a non-Power 5 conference (MAC Conference) has already achieved 2 losses, then its ranking will be assigned a zero and the team will be disqualified from the Top 4 and Top 2 rankings (for all iterations).

At the end of each iteration, teams which are assigned non-zero rankings are stored for later reference. Once all iterations have completed, the number of Top 4 and Top 2 rankings for each team are counted. If we divide each team's count by the total number of iterations, we finally arrive at the team's probability of finishing the season ranked in the Top 4 or Top 2. To illustrate this, consider the output of Top 4 probabilities in the table below. Note that teams with a probability less than about 2% have been omitted for brevity, but the comprehensive sum of all probabilities will be exactly 400%.

team	probability	team	probability
Ohio State	0.2962	Wisconsin	0.0487
Clemson	0.2923	Georgia	0.0474
Alabama	0.2744	Southern Cal	0.0449
Tennessee	0.2526	LSU	0.0436
Oklahoma	0.2487	Florida	0.0436
TCU	0.2256	Iowa	0.0385
Michigan	0.2038	Notre Dame	0.0372
Baylor	0.1628	North Carolina	0.0359
Stanford	0.1513	Nebraska	0.0308
Arkansas	0.1397	Western Kentucky	0.0295
Mississippi	0.1385	North Carolina St	0.0231
Michigan St	0.1013	Oregon	0.0231
Mississippi St	0.1013	Texas A&M	0.0231
West Virginia	0.0846	San Diego St	0.0218
Utah	0.059	UCLA	0.0218
California	0.0513	Florida St	0.0205
Washington	0.05		

Step 4: Aggregate the Probabilities and Statistics

The final step is simply to aggregate the statistics and probabilities computed in the first 3 steps, which can be done with any choice of software. At the same time, we can include some other useful statistics which were derived from scores data during our calculation, but did not contribute directly to any of our derived results.

Other Supporting Statistics

A number of other basic statistics are derived by applying simple arithmetic to the actual game outcomes. These statistics include but are not limited to:

- Top 25 win/loss record
- Largest margin of victory/defeat
- Best quality win
- Record against teams with winning/losing records

References

[1] Pugh, S.M., “Compughter Ratings Theory”, http://www.compughterratings.com/theory, (Aug, 2007)
[2] Wikipedia, “The Probit Function”, https://en.wikipedia.org/wiki/Probit, (Feb, 2017)
[3] Wikipedia, “Poisson Binomial Distribution”, https://en.wikipedia.org/wiki/Poisson_binomial_distribution, (May, 2017)

Contact

Steve Pugh: stevepugh@compughterratings.com
LinkedIn: https://www.linkedin.com/pub/steve-pugh/1/457/7b4