Introducing: Ensemble Ratings

The Difficulty in Ranking and Ratings Teams

One of the central problems in sports analytics is ranking teams within a league. Historically, the best method was to poll a number of experts and aggregate their votes to form a consensus. In fact, this subjective method is the *de facto* technique to rank teams in college sports (the AP, coaches Polls). However using humans creates two main biases that we observe. First, polling has a recency bias: teams that have performed well recently are often ranked higher than teams that performed well at the beginning of the season. Second, humans are not good at gauging strength of schedule

More recently, some computerized, objective methods have been created to address these issues. Three of the most popular are RPI (popular amongst CBB fans), Pythagorean expectation (popular amongst baseball/sabermetrics fans), and RAPTOR (previously CARMELO, 538's method, popular amongst data junkies).

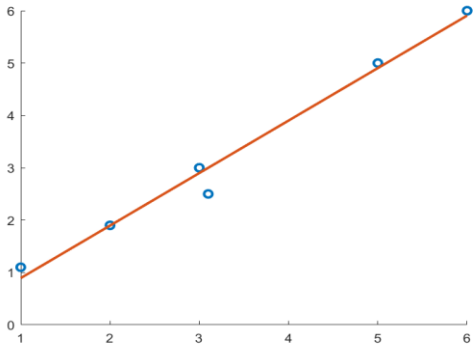
RPI inherently takes into account strength of schedule. The formula is pretty simple: you get points for win percentage (WP), your opponents' WP, and your opponents' opponets' WP. The idea is to reward teams for playing harder strength of schedule. It *seems* like a good idea, but is extremely easily manipulated and has many flaws. As a simple example, suppose team A and team B play each other 3 times and have only played each other (for instance, team A and team B open a baseball season with a series against each other). If team A sweeps team B, then team A and team B will have identical RPI. However, team A is observably better than team B. Because of these flaws most sports entities and analysts, including the NCAA itself, have moved away from RPI.

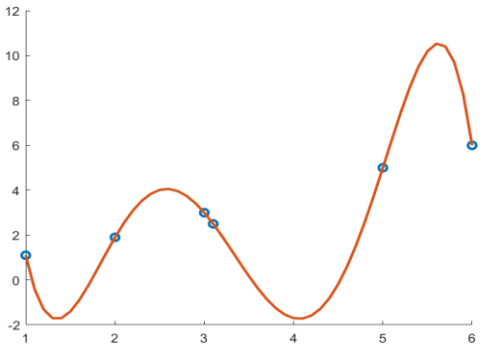
The second method is the Pythagorean Expectation (PE). PE tries to predict your winning percentage by comparing how many runs you score to how many you allow, taking opponents completely out of the picture. Because of this, PE is even more susceptible to anomalies in strength of schedule. PE also has a more technical problem: it requires the determination of an *exponent* which has no physical representation in the real world. For example, in baseball a team's expected winning percentage looks like:

$$Win\ Pct = \frac{Runs\ Scored^{1.83}}{Runs\ Scored^{1.83} + Runs\ Allowed^{1.83}}$$

This 1.83 has no physical interpretation; it doesn't *mean* anything. Even worse, the 1.83 is specific to baseball. If we want to use the Pythagorean idea in basketball, we have to find a new exponent.

RAPTOR is 538's method of rating NBA teams. It combined box statistics and one of my favorite measures of player ability: plus/minus. While RAPTOR is a very successful metric with a large amount of advantages, it has some problems. First, it is only applicable to basketball. You can't use it to create rankings in football or in baseball. Second, it uses massive amounts of data including height, age, draft position, etc. While these are all undoubtedly meaningful in predicting player ability, using all these variables risks *overfitting.* Every model has to trade off predictive accuracy (i.e. predicting which teams are better *in future games*) versus historical accuracy (predicting which teams won in past games). It turns out, if you force your model to fit past data too much, it doesn't actually capture the underlying relationship. The graph below shows an example where increasing the model complexity fits the observed (blue) points better, but probably doesn't explain the underlying relationship as well.





Introducing: Ensemble Ratings

The Model in a Nutshell

First, why do we call our method Ensemble Ratings? We have two reasons, one mathematical reason and one sports reason. First, the mathematical technique we use generates a Gaussian ensemble to model games in our league. Second, as will become more evident as we work on a more advanced technique based on the same idea, basketball teams' ability can truly be seen as an aggregate of individual players' abilities. That is, a team's rating can be broken down into an ensemble of individual player ratings. Let us begin with the simple method and save the advanced version for the future. The ensemble ratings are a sports rating system which:

- Assigns to each team a single number which may be interpreted as quality. The average rating is fixed to 100 and the difference between two ratings is the predicted margin of victory. For example, if Boston has a rating of 105.1 and New Orleans has 99.5 rating,

then on a neutral court we would predict Boston to beat New Orleans by 5.6 points, on average.[1]

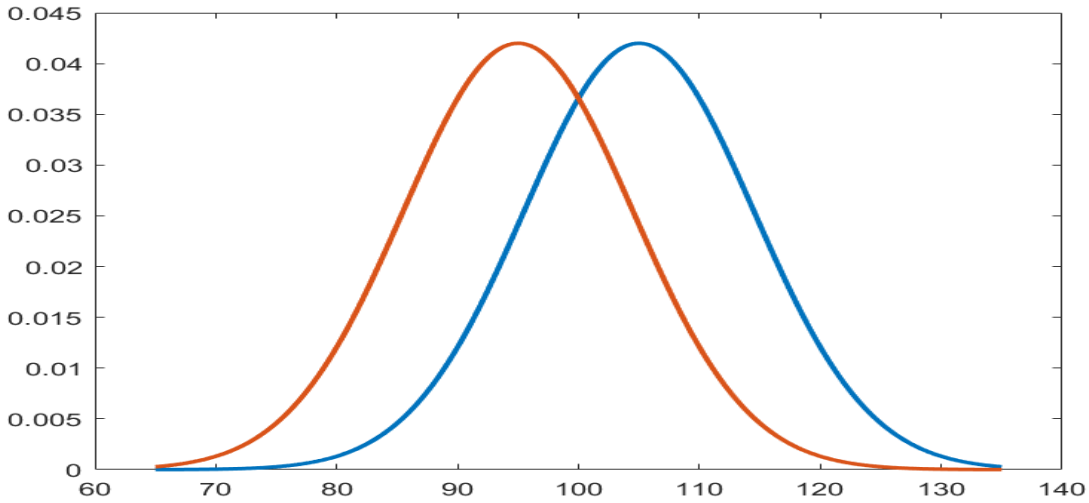
- Implies a win probability based on the predicted margin of victory. Milwaukee has a current rating of 109.6 while Oklahoma City has a rating of 103.3. So, if Milwaukee plays at Oklahoma City, we predict a margin of victory of 3.3 points (assuming a 3 point ome court advantage) and our model also tells us that Milwaukee should win this game 59.7% of the time.
- Uses **only** the final scores and location of each game to generate the ratings. Our model has no knowledge of box stats. For example, if Houston lost to Portland by 1 in OT but only shot 11% on 40 3PA, this game would have the same impact on the ratings as if they lost by 1 in regulation and shot 80% on 50 3PA. Since we use the bare minimum amount of information to make the ratings, the risk of overfitting is extremely low. This allows users of the ratings to use their own observations that aren't already `baked in' to the model.
- Can be used in any sport where the victory is determined by who has more `points' (runs, goals, points, etc.) at the end of the game. We can use this to rate teams in football, baseball, hockey, basketball, etc.
- Inherently uses margin of victory and strength of schedule in determination of ratings. Unlike RPI, strength of schedule only matters to the extent that you perform well against good teams. That is, losing by 2 to Milwaukee (rating 111) is seen as a better performance in our model than beating Cleveland by 15 (rating 92).

How do we do it? [MATHEMATICS, may be skipped]

The model assumes that each team's performance on a night-to-night basis is approximately normally distributed. Our model assigns each team a normal distribution where the mean is the team's rating and the standard deviation is the same for each team[2]. Then, the difference between the peaks of each distribution is the predicted difference. Below, the red team has a rating of 95 and the blue team has a rating 105. We expect blue to beat red by 10 on average.

[1] Adding in about a 3 point home court advantage, we could predict Boston to win by 8.6 points at home and only 2.6 points at New Orleans. Our model allows us to plug in any home court advantage we like and can also learn the home court advantage that best explains the data we observe.

[2] Interestingly, if we allow for different standard deviations, we get better accuracy but the optimization problem is much, much more difficult to solve.



What else our model does: the probability that blue beats red is the probability that a random point on the blue curve is larger (i.e. the team scores more points) than a random point on the red curve. This built-in randomness lets us encapsulate the idea that team's performances are inconsistent, they vary night to night in an unpredictable way.

The mean (peak) of each team's assigned curve is picked to be the location that minimizes the difference between the predicted margins of victory and the observed margins of victory through the games that have been played so far[3].

How does our model do?

Our model is designed to do two things: predict a winner and report a predicted probability of our team winning. How well does it work? The following table lists correctly predicted winners using a few popular basketball rating methods (there have been 803 NBA games this year).

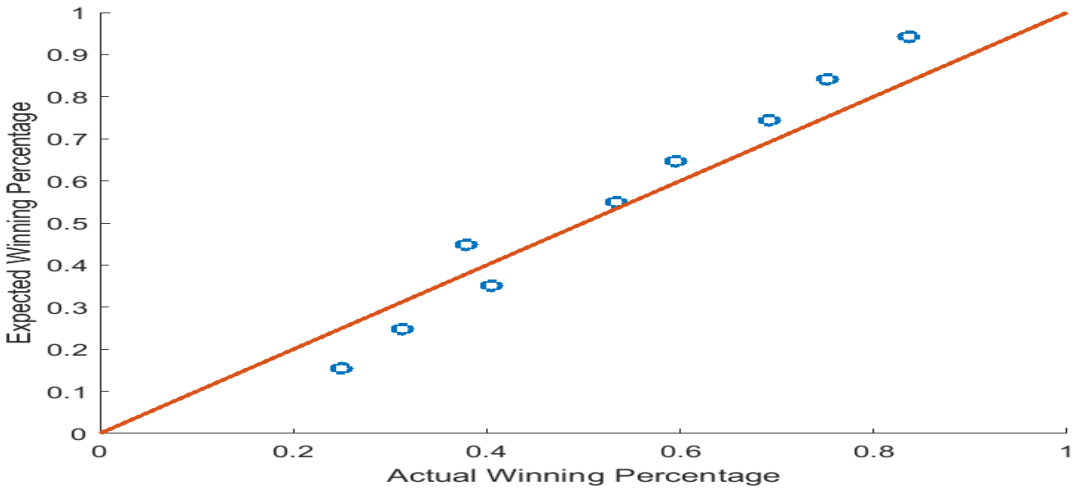
Method	Correct Predictions	Prediction Percentage
ESPN BPI	532	66.2%
Sagarin	530	66%
Ensemble	528	65.8%
Pythagorean	512	63.8%

For all intents and purposes, the top 3 method have indistinguishable accuracy while the Pythagorean method performs significantly worse. The first argument for our method is that it predicts winners straight-up with almost optimal accuracy using minimal information.

[3] We use Maximum Likelihood Estimation to determine the set of means and a common variance for each distribution. It turns out, using this idea reduces the problem of simultaneously finding each team's mean to a matrix inversion problem which is quite low-dimensional.

The second argument in favor of our rating system is that it provides a probability of a specific team winning using the method described above. In the following paragraph we test how accurate these probabilities are.

First, we split all the games into a number of groups where the probability of the home team winning in each group was approximately equal. For example, we will group up all the games where the home team has between a 30 and 35% chance of winning. Then, we predicted the number of wins in each group (by calculating the win probability for each game!) and plot one blue dot per group. Each blue dot has x coordinate at the actual group win percentage and y-coordinate at the predicted group win percentage The red line is y=x, so the dots being close to the line means our predicted winning percentage is close to the actual winning percentage.



How we'll Use this Going Forward

Our method provides ratings that will be hosted on this site shortly in the future. However, the reason our method is interesting to us is because of the win probabilities it provides. This idea lets us simulate games, tournaments, seasons, etc. which will allow us to write articles about any number of topics which require simulation.