Pairwise Comparison Models: A Two-Tiered Approach to Predicting Wins and Losses for NBA Games

Tony Liu

Introduction

The broad aim of this project is to use the Bradley Terry pairwise comparison model as the basis for finding strong predictive models for NBA games. Bradley Terry model is commonly used in power rankings in sports. It is primarily used to calculate win probabilities based off of a team's win/loss record. I argue, however, that using win percentages alone might not be the most effective method.

For instance, it is not necessarily true that if team A has a greater than half chance of beating team B and team B has a greater than half chance of beating team C that team A will have a greater than half chance of beating team C.

Instead, I hypothesise that it is possible to come up with a better predictive model by first predicting features that have a high correlation with win rate. Therefore, my model would have a two-tiered approach. First, I calculate the features that are predictive of win rate and then I feed those predictions into a model that has those features as the predictors and win rate as the response.

Here, I turn to Dean Oliver's "Four Factors of Basketball Success."[i] Oliver argues that most the variation in wins can be explained by Shooting, Turnovers, Rebounding and Free Throws. He assigns the weights to each factor, 40%, 25%, 20% and 15%, respectively. In my model, I will determine the coefficients for each factor, which have the greatest predictive power. The four factors are defined in the following ways.

Shooting:
Effective Field Goal Percentage =
(Field Goals Made + 0.5*Three Pointers Made)/Field Goals Attempted

Turnovers:
Turnover Percentage =
Turnovers/(Field Goals Attempted + 0.44*Free Throw Attempts + Turnovers)

Rebounding:
Offensive Rebound Rate =
Offensive Rebounds/(Offensive Rebounds + Opposition Defensive Rebounds)
Defensive Rebound Rate =
Defensive Rebound Rate = Defensive Rebounds/(Opposition Offensive Rebounds + Defensive Rebounds)

Free Throws:
Free Throw Factor = Free Throws Made/Field Goals Attempted

As these values are all rates, we can predict the each of the four factors for both teams. Let's consider a game between team A and team B. Consider a prediction for A's Turnover Percentage. We would need to know A's mean turnover percentage, the league's mean turnover percentage and the mean turnover percentage of teams when they play against B. Given these values, I am able to apply the Bradley Terry Model, where the three agents are A, B and the league.

Why use the Bradley Terry Model at all?

Other models that make use of data at a far more granular level will likely have greater potential for predictive power. The reason I choose the Bradley Terry Model, however, is its simplicity and its potential to make sensible predictions with very limited data. It is clear that the model I suggest only requires data at the team level for each game. Thus, there are far fewer features.

Methodology

There are two predictive layers in the model – optimise a model for predicting the four factors and a model for predicting win rate from the four factors. I use the 2010-2011 NBA season as my data set. There are (82*30)/2 = 1230 games per season. I split the data set into a training set and a test set. The training set consists of the 70% of the season and the test set consists of the remaining 30%. There are 861 observations in the training set and 360 observations in the test set. As a point of comparison for my model, I also tune the model that only uses the win/loss record.

Predicting the four factors

In predicting the four factors, the key parameter I consider is the number of games that I should use in the prediction of a single game. Since this is a purely predictive model, I can only predict on a game using past games. An obvious choice would be to include every game leading up to the prediction game. There are, however, some potential disadvantages with this method. By the time a team plays its 70^{th} game of the season, the first twenty games might not be so predictive of the outcome of that game. Also, injuries and roster changes can decrease the importance of earlier games.

The alternative to this approach is a moving window. Therefore in my model, I tune the size of the window. I train and test on my original training set and calculate the test MSE error. The size of the training and test sets within the original training set varies depending on the size of the window. Given a window size d, the training set, here, is essentially the number of games that takes place in the league before every team plays d games. For instance, for a window size of 1, which is the case that we only use information from the previous game to predict the next, by the 18^{th} game of the season, every team has played at least one game. I calculate the mean squared error for the five different window sizes and also for the case, in which I include every game leading up to the prediction game.

Window Size	num obs.	Rebound MSE	Turnover MSE	eFG% MSE	FT factor MSE	Sum of MSE
1	844	0.016501403	0.002960085	0.011684333	0.022131734	0.053277555
2	776	0.011073513	0.002020287	0.007479058	0.02408846	0.044661318
5	693	0.007100297	0.00142125	0.005043673	0.01293233	0.02649755
10	**536**	**0.0063628**	**0.001249419**	**0.004432883**	**0.002776665**	**0.014821767**
20	371	0.005733524	0.001195112	0.004259816	0.005780949	0.016969401
All games	844	0.00608761	0.001254227	0.004407891	0.009369296	0.021119024

Predicting wins from the four factors

Now, I select a model that predicts win rate from the four factors. I consider both linear and non-linear models, including least squares regression, logistic regression, regression and classification trees. I perform 10-fold cross validation on the training set to determine the 0-1 loss and/or mean squared errors of the various models. For the regression models, I use Point Differential as the response and, for the classification models, I use the two classes – win and loss.

I removed Opposition Offensive Rebounds and Opposition Defensive Rebounds from the feature set because a model that includes them would have high multicollinearity since they can be derived from the Offensive Rebounds and Defensive Rebounds features.

From the table, it seems that the window size that performs the best is 10 using the criteria of summing the errors across the four factors. The errors appear to decrease until size 10 before increasing again.

The linear models perform the best and within the linear and non-linear approaches, classification methods outperform their regression counterparts. I chose logistic regression for my two-tiered model.

10-Fold Cross Validation Results

Model	MSE	abs(y_hat – y)	0-1 Loss
Least squares	9.84896582	2.54035922	0.04298316
Logistic regression	n/a	n/a	0.03716921
Regression tree	74.90737	6.877177	0.2078856
Classification tree	n/a	n/a	0.1962978

Predicting wins from Win/Loss record only

I tune the window size for the model that predicts win rate only using the win/loss record. Here I use the 0-1 loss error to find the optimal parameter, which appears to be a window size of 20. Using every single game prior to the prediction games has an insignificantly smaller error so I opt for the simpler model that requires less information.

Window Size	num obs.	0-1 Loss
1	844	0.4490521
2	776	0.4379562
5	693	0.4007732
10	536	0.3708514
20	**371**	**0.3451493**
All games	844	0.3414948

The final predictive models

To determine if the two-tiered approach performs better than the model that only uses win/loss record, I compare the following two models.
1. A two-tiered model that uses window size 10 games to predict the four factors and then uses those predicted four factors to predict wins through the logistic model.
2. The single-tier model that predicts wins using only win/loss record with window size 20 games.

Results

I refit the logistic model on the entire training and then predict the four factors and win rates of the test set of 369 observations. Similarly I predict win rate with the single-tier model on the same set. I find that the two-tiered approach has a lower test error than the single-tier model with a 0-1 loss of 0.360 against 0.385.

Model	0-1 Loss	Correct Guesses	Total Games
Two-tier model	0.3604336	236	369
Single-tier win/loss	0.3848238	227	369

We can compare these error rates with more established models that use player statistics to predict wins. We can refer to Omidiran's[ii] paper, which compares the performance of different models based on Adjusted Plus-Minus (APM) scores of players, which considers the overall contribution of a player to the point differential. He uses the same season for his data set. He trains on the first 410 games before predicting on the final 820 games of the season. His dummy model, which only considers home court advantage, had a 0-1 loss of 0.4024. The least squares model achieved a 0-1 loss of 0.4073 and the ridge

regression model 0.3732. His Subspace Prior Regression models managed to achieve a 0-1 loss of under 0.3.

Therefore it is interesting to see that the two-tiered model is at least comparable if not a better predictor of wins than several of the plus-minus models, while requiring far less information. It must be noted, however, that a primary motivation of APM models is to measure player performance.

Conclusion

The results of this project are encouraging for several reasons. First, It seems that there is reasonable evidence that indirectly predicting wins, as in the two-tiered approach, could be a more successful paradigm for modelling NBA games. Second, the Bradley Terry model can be successfully applied to statistics beyond wins. Third, in predicting a game the size of the sample that should be considered is an important consideration. Based off of my findings, a sample size between 10 and 20 games seems to be optimal.

[i] http://www.basketball-reference.com/about/factors.html

[ii] Omidiran, Dapo. Low-Dimensional Models for PCA and Regression. Diss. U of California at Berkeley, 2013.