Pairwise Comparison Models

A Two-Tiered Approach for Predicting Wins and Losses in the NBA

Motivation

- Test Bradley Terry model as the basis for finding strong predictive models for NBA games
- Test the success of an indirect, two-tiered approach to predicting wins
- Only using Win/Loss record might not be optimal

Hypothesis

- Could be more effective to use a two-tiered approach
- First identify the broad features that have high correlation with win rate
- Model wins based off of those features
- Predict for those features first and then predict wins

Dean Oliver’s Four Factors

- Effective Field Goal Percentage =

(Field Goals Made + 0.5*Three Pointers Made)/Field Goals Attempted

- Turnover Percentage =

Turnovers/(Field Goals Attempted + 0.44*Free Throw Attempts + Turnovers)

- Offensive Rebound Rate =

Offensive Rebounds/(Offensive Rebounds + Opposition Defensive Rebounds)

- Defensive Rebound Rate =

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

- Free Throw Factor =

Free Throws Made/Field Goals Attempted

Bradley Terry Application

- The four factors are all rates
- To calculate A’s turnover rate against B, we need
  1. A’s mean turnover rate
  2. The league’s mean turnover rate
  3. The mean turnover rate of teams that play against B

Why Bradley Terry?

- Simple
- Very little data required (only at the team level)
- Far fewer features to predict

Methodology

- Data set 2010-11 NBA season
- (82*30)/2 = 1230 observations
- 861 in training set and 369 in test set (70%/30%)

Models

- Two predictive layers in model
  - a model for predicting the four factors
  - a model for predicting win rate from the four factors
- Reference model
  - Only uses win/loss record to predict win rate

Predicting Four Factors

- Only predict on a game using past games
- How many games to include in training sample?
- Two possible options
  - Use every game leading up to prediction game
  - Use a moving window of size *d* games to predict

Tuning window size

- Objective: tune *d* with training set
- Set d = 1, 2, 5, 10, 20
- Train on different number of observations
- E.g. when d = 1, I start training when every team has played at least 1 game
- Compute *MSE* for the 5 values of *d* and also for the case in which every game is included

Results

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 four factors

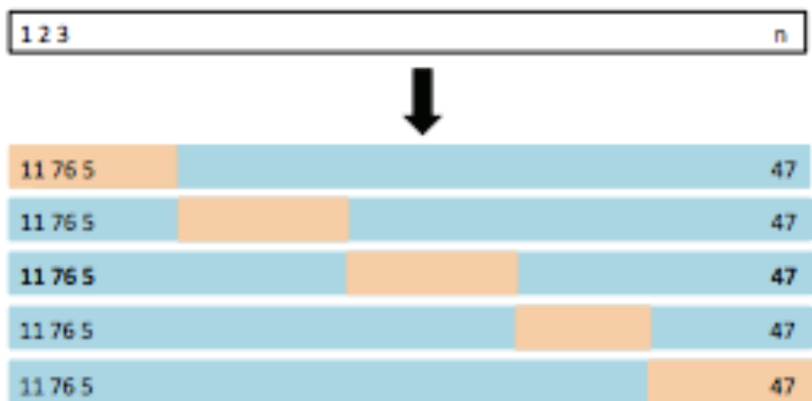
- Linear models
  - Least squares regression
  - Logistic regression
- Non-linear models
  - Regression tree
  - Classification tree
- Point differential vs Win/Loss
- Multicollinearity with Rebound features

Feature Set

```
> head(train.df)
  TurnoverRate   EFGRate FreeThrowRate OffReboundRate DefReboundRate
1   0.18367347 0.5217391    0.23188406      0.2222222      0.7555556
2   0.10752688 0.5161290    0.10752688      0.4390244      0.8108108
3   0.09973404 0.4635417    0.23958333      0.2745098      0.6521739
4   0.13430545 0.4814815    0.20987654      0.2000000      0.8285714
5   0.10831036 0.5250000    0.21250000      0.2500000      0.7777778
6   0.15408320 0.4759036    0.09638554      0.2380952      0.8484848
  OppTurnoverRate OppEFGRate OppFreeThrowRate OppOffReboundRate
1      0.16666667  0.4189189        0.2432432         0.2444444
2      0.18992403  0.5472973        0.1486486         0.1891892
3      0.16217970  0.4615385        0.2857143         0.3478261
4      0.18954509  0.4930556        0.2222222         0.1714286
5      0.05870841  0.4939759        0.1927711         0.2222222
6      0.16220600  0.5362319        0.3333333         0.1515152
  OppDefReboundRate PointDifferential WinLoss
1         0.7777778                 8     Win
2         0.5609756                14     Win
3         0.7254902                 2     Win
4         0.8000000                 8     Win
5         0.7500000                 3     Win
6         0.7619048               -10    Loss
```

Model Selection

- 10-fold cross validation, i.e. randomly divide training set into 10 folds



Results

- Best model is logistic regression with a moving window of 10 games

10-Fold Cross Validation

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

Logistic Regression Model

```
Coefficients:
                  Estimate Std. Error z value Pr(>|z|)    
(Intercept)        -28.976      4.716  -6.145 8.01e-10 ***
TurnoverRate      -109.504     13.350  -8.203 2.35e-16 ***
EFGRate            111.361     12.237   9.101  < 2e-16 ***
FreeThrowRate       26.971      3.572   7.550 4.35e-14 ***
OffReboundRate      30.590      4.257   7.186 6.69e-13 ***
DefReboundRate      30.117      4.300   7.005 2.48e-12 ***
OppTurnoverRate     97.897     11.676   8.384  < 2e-16 ***
OppEFGRate        -109.903     11.910  -9.228  < 2e-16 ***
OppFreeThrowRate   -27.710      3.875  -7.151 8.64e-13 ***
---
```

Tune single-tier model

- Compare 0-1 Loss
- Select window size of 20 games

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

Performance on test set

- Test set of 369 observations

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

Compare with other popular models

- Omidiran
- 0-1 Loss
- Dummy model
  - Home court advantage 0.4024
- Plus-minus models
  - Least squares 0.4073
  - Ridge regression 0.3732

Compare with other popular models

- Errors seem to be at least as small as errors in the plus-minus model
- However, motivation of APM is to measure player performance
- But, our models require far fewer features

Conclusion

- Reasonable evidence that models that indirectly predict wins can be successful
- Bradley Terry model can be applied beyond win/loss record
- Sample size in predicting game, i.e. window size