Predicting NBA Player Performance

Kevin Wheeler

Introduction

Predicting the outcomes of sporting events and the performances of athletes represents a natural application for machine learning. Major, professional sports such as the NBA, NFL, and MLB contain a significant amount of easily accessible data whose outcomes and player performances tend to be randomly distributed and offer attractive data to predict. The NBA is particularly well suited for machine learning applications because the performance of NBA players across different positions can be measured using the same set of statistics. This is unlike the NFL, for instance, where different positions are measured using different metrics (e.g. QB's are measured based on how many TDs they throw while linebackers are measured based on how many people they sack/tackle).

Furthermore, applying machine learning algorithms to predict the performance of individual athletes also has a natural extension in predicting the outcomes of games. By predicting the scoring performances of each player and summing them up, it can be determined if predicting individual performances can be used an accurate win-loss classifier.

In this project, I set out to develop a model using linear regression (with Naïve Bayes and SVM implementations to compare) to predict how many points NBA players would score against an opponent.

Methods

Feature Selection

The ability to accurately predict the performance of NBA players depends upon determining and using features that are strongly correlated with the number of points an NBA player will score against any given opponent. A χ^2 test was used to analyze a set of 46 team and individual, gave-level statistics to determine the subset of features to be used in the prediction models. For instance, the minutes a player plays or the number of shots a player takes in a game represent game-level statistics while Pace and opponent turnover rate are team-level statistics. Using a χ^2-value of 0.05 as a cutoff (all statistics below this value were subsequently disregarded), 16 features were ultimately used to calculate the parameter vectors for each player. The features are shown in table 1 below.

Table 1. Features that correlated best with the number of points NBA players scored in a game.

Team-level features	**Description**
Pace	Average possessions per game
OPTS	Opponents points scored
OFG%	Opponents field-goal %
OTOR	Opponents turnover rate
DRR	Percentage of defensive rebounds made
O%Rim	Opponents shots taken near rim
O%Short	Opponents shots taken w/in 10 feet
OXeFG%	Expected FG% allowed
OeFG%	Effective FG% allowed
TRR	Total rebound rate

Player-specific features	**Description**
MINS	Minutes played
FGA	Shots attempted
FTA	Free throw shots attempted
eFG%	Effective FG% (weighted for 3PA)
TS%	True FG% (adj. for FT and 3PA)
OffRtg	Offensive rating (measure of points produced per 100 possessions)

Linear Regression

Linear regression was used to predict both the player-level statistics and the number of points each player would score against a particular opponent using equation 1 below. The training set consisted of statistics from every game each player played in over the last three seasons.

$$\theta = (X^T X)^{-1} X^T y \qquad (1)$$

Above, θ was the parameter vector, χ was the training set where the rows represented each game the player played in and the columns were the features, and y were the training results. Once the regression parameters were calculated for each player, they were used to make scoring predictions. As stated above, the parameters were calculated based on all of the player's games played over the last three seasons. Predictions were then made for the current (2012-2013) season and the end of last season (2011-2012) using cross-validation.

Win-loss classification

The linear regression model was then extended to determine whether the outcomes of games could be predicted. This was done by summing up the predicted points scored by each player and comparing the results between two opposing teams.

To compare the performance of this type of classifier, Naïve Bayes and support vector machine (SVM) models were implemented. A multinomial event model with Laplace smoothing was implemented for a Naïve Bayes model. Maximizing the likelihood expression given in equation 2, resulted in the likelihood parameters given by equations 3-5

$$\mathcal{L}(\emptyset, \emptyset_{k|y=0}, \emptyset_{k|y=1}) = \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}) = \prod_{i=1}^{m} \left(\prod_{j=1}^{n_i} p\left(x_j^{(i)} \middle| y; \emptyset_{k|y=0}, \emptyset_{k|y=1}\right)\right) p(y^{(i)}; \emptyset_y) \qquad (2)$$

$$\emptyset_{k|y=0} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n_i} \left(1\{x_j^{(i)} = k \wedge y_j^{(i)} = 0\} + 1\right)}{\sum_{i=1}^{m} \left(1\{y^{(i)} = 0\} n_i\right) + V} \qquad (3)$$

$$\emptyset_{k|y=1} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n_i} \left(1\{x_j^{(i)} = k \wedge y_j^{(i)} = 1\} + 1\right)}{\sum_{i=1}^{m} \left(1\{y^{(i)} = 1\} n_i\right) + V} \qquad (4)$$

$$\emptyset_y = \frac{\sum_{i=1}^{m} \left(1\{y^{(i)} = 1\}\right)}{m} \qquad (5)$$

where $\emptyset_y = p(y)$, $\emptyset_{k|y=1} = p(x_j = k|y = 1)$, $\emptyset_{k|y=0} = p(x_j = k|y = 0)$. The likelihood parameters were calculated using the same set of features used in the linear regression model (as seen in table 1).

Since the statistical categories were a combination of percentages and integers, each x_{ij} in the training set was classified and converted into a number from 0-9 based on what decile the value belonged to relative to its category. For instance, if a player's eFG% was 0.768 for one training example (i.e. one game played in) and the 80^{th} and 90^{th} percentiles were 0.650 and .810 respectively, then this percentage was converted into a 7 for being this decile. This was done to relatively weight each category and put the calculation of the likelihood parameters on even footing. It should also be noted that the SVM incorporated a linear kernel and the training sets used were consistent across the various models.

Results

Game-by-game scoring fluctuations were difficult to model and predict
Overall, the linear regression model was able to predict a player's points scored per game (PPG) to within 3.5% of his season average. However, the standard deviations in the predictions compared to a player's actual performance differed dramatically. On average, the standard deviations in the predicted PPG were 7.7 times less than the player's actual standard deviation of PPG. The standard deviation in the actual PPG of a player was roughly 20%-40% of his season average while the standard deviation in the predicted PPG was between 2.5%-5%.

Predicting player-level statistics using linear regression was better than using player averages
Despite being unable to accurately predict the large deviations in the PPG and player-level statistics, simply using a player's season averages for their player-level statistics resulted in less variance in the predictions. For instance, in stead of trying to predict how many shots a player will take/make or how many minutes he will play, using his season and career averages for these statistics resulted in average standard deviations in the predicted PPG of 0.5%.

PPG predictions did not scale well to accurately classify win-loss
Using the linear regression model to predict a player's PPG and then adding up the predictions for each player on opposing teams to predict win-loss outcome had an error rate of 53%. This is unlike the Naïve Bayes and SVM classifiers that had error rates as low as 31% (as seen in figure 1).

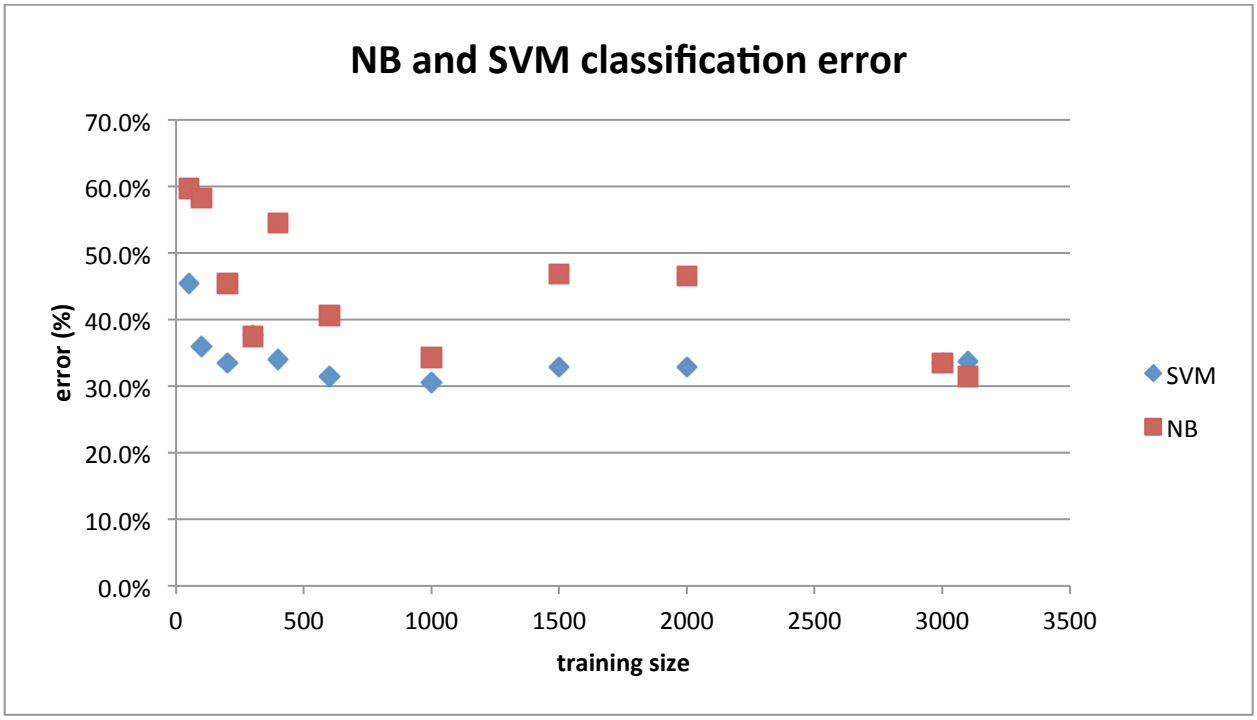

Figure 1. Shows the error rates from the multinomial event and SVM models when used to classify matchups as wins or losses.

Conclusion/Future directions

Accurately predicting player-level statistics is key

The inability to accurately predict the deviations in a player's PPG was most likely due to the inability to accurately model player-level statistics. The challenge in predicting these statistics lies in difficult-to-quantify and not-well-understood factors. For instance, the number of minutes a player plays and shots he takes can depend in factors such as how well he is playing on that particular game, which specific player is defending him, how well the other players on his team are playing, and his confidence level may influence his performance, but it is not straightforward how they should be represented in a regression model. Furthermore, the coach's decision to play a player can also depend on a number of behavior, philosophical, and social factors (each of which may vary from coach-to-coach).

Social media and weighted regression provide possible strategies for improving predictions

A possible strategy to quantify the behavior, philosophical, and social aspects stated above is to use social media and weighted regression as part of the feature set and regression model, respectively. For instance, one strategy is to attempt to classify the confidence level of a player (which may influence how well he plays) by weighting his recent performances more than his earlier performances. One could further classify confidence by quantifying his social media and press coverage (e.g. positive twitter postings and press-coverage may indicate higher confidence levels).

Attempting to predict player performances still has value and potential future applications

Though the predictions made here were less than optimal, this was an important first step in predicting player-level performance. If improvements can be made to the model and the above strategies can be successfully incorporated, this will represent an important step in building models that can accurately predict and outperform classifiers based on Naïve Bayes and SVM models.

References

1. http://www.basketball-reference.com/
2. http://espn.com/nba
3. http://hoopdata.com/
4. http://nba.com

Acknowledgements

Thank you to the CS229 staff and Prof. Ng for a great class.