Maneesh Apte
Simar Mangat
Keyur Mehta

CS221 Project Proposal

Over the last few years, there have been a number of systems created with the intention of predicting NBA games. The majority of these systems have used macro-game indicators as their features, including statistics like Home Team Points Scored per Game and Away Team Points Against per Game, to make predictions on how a specific game will turn out. Our system seeks to expand on existing models by diving into micro-game indicators to better predict which team will win and by how much. Micro-game indicators, or specific components of how a team performs on the offensive and defensive side of the game, of each team are used as features with the hopes that understanding how teams stack up in part, as opposed to in whole, will better inform a prediction of how the game will go. Therefore, the **input for our system will be the team's offensive and defensive metrics calculated over the season prior to the game we are predicting and the output would be a score differential prediction for the game itself.** For example, imagine that our game of choice was the Warriors vs. the Spurs, game 73 in the 82 game season. Using the data provided by the SportVu player tracking system for all the games that the Warriors and Spurs had played in the season so far, each team would have a metric for their offensive ability in the following categories:

- Half-court & Fast-break Drives/Layups (shots within 6ft of the basket)
- Mid-range 2 pt shots (6-15 ft from basket)
- Long-range 2 pt shots (15-22 ft from basket)
- 3 pt shots (22+ ft from basket)
- Foul-drawing
- Free throws
- Rebounding
- Turnover capacity

All of the shot taking ability will include the percent accuracy of the shooters as well as how hard the shot attempt would be (determined by the distance to closest defender, and ability of that defender)]. Both teams would also have calculated defensive ability in the same aforementioned offensive categories except instead of measuring how well they performed in the categories, it would measure how badly they had played performed. Depending on how the Warriors offensive ability stacked up against the Spurs defensive ability, the system would output the predicted number of points scored by the Warriors. Similarly, the system would also spit out the Spurs' points. These scores would then be used to classify the game as a win/loss and additionally to produce the predicted score differential. To evaluate success in the test set and see how the system performed, we would compare our classification to the true result as

well as the predicted differential to the actual game differential. **In the case of classification, our baseline involves always choosing the home team, often the favored team in professional sports, to win each game it played.** If employed in the 2014-2015 season, this baseline would have 53.7% accuracy. **The oracle, on the other hand, can be estimated with the Accuscore algorithm for NBA (win/loss) bets.** We chose this algorithm as our oracle because Accuscore is a best-in-class betting prediction company with the strongest track record of success. For the 2014-2015 season, their model had an accuracy of 70.3%.

Firstly, in working with a dataset with such a huge range of features, it's hard to engineer features that will have high predictive power in the model. Thankfully, there is a suite of analyst research on basketball that we can use to inform important features like number of rebounds or drives down the court, but there will likely be additional information hiding in the data and also potential key features that are not in our dataset to begin with. Another key challenge will be generalizability. When training our model, we will likely face overfitting. An ML algorithm could foreseeably weight features from basketball game that are in truly irrelevant, but are learned nonetheless. Additionally, we are attempting a fairly difficult regression challenge by trying to predict the spread of a game since chance is especially prominent in sports games. To address the challenges in feature selection we could analyze expert opinions on important game characteristics and use quantitative selection techniques like dimensionality reduction or clustering. For overfitting, we'll have to use cross validation and/or regularization.

There have been a few projects in the past that have explored using machine learning techniques on predicting win-loss of NBA games. The first project found was a project that had similar goals as ours[1]. One of the main takeaways from this project was that their baselines looked at all previous games (in the season) between two teams and picked the one with the fewest losses as the winner. Second main takeaway was that their feature set stuck to using data about wins and losses regarding the team and did not dig deeply into incorporating features regarding the player data. We plan to take this further by exploring features regarding players and their positions. The second project we found was a former cs229 project[2]. Most notably, the project used SVMs in conjunction with a lasso regression and bootstrap aggregating. Again, consideration of specific player data was not done (it seems that the data was not as readily available for them).

We believe there are exciting possibilities to apply machine learning on NBA games.

[1] http://homepages.cae.wisc.edu/~ece539/fall13/project/AmorimTorres_rpt.pdf
[2] http://cs229.stanford.edu/proj2013/ChengDadeLipmanMills-PredictingTheBettingLineInNBAGames.pdf