Predicting Results for Professional Basketball Using NBA API Data

Jacob Perricone (jacobp2@stanford.edu), Ian Shaw(ieshaw@stanford.edu), Weronika Święchowicz (wswiecho@stanford.edu)

CS229: Machine Learning

Objectives

Advance the branch of sport statistics specializing in predicting the winner of the National Basketball Association (NBA) Regular Season Games:

- Novel application of common Machine Learning feature selection and model training techniques in developing more accurate predictions of the winner.
  - Utilization of aggregated team and game statistics provided by the NBA `API`.
  - Inclusion of temporal components of a team's statistics.
  - Identification of feature sets most representative of the quoter and full season data.
  - Development of a mathematical model that learns the strengths of the basketball team and predicts the final outcome of the game.

Background

The study of predicting a winner of a team based competition has been of interest to the scientific community for years. The recent popularity of the fantasy team construct only popularized this branch of Machine Learning and Bayesian updating based predictions. With the best models predicting about 70% of the winning teams correctly, there is still a room for improvement.

Data

In this project we utilized detailed statistical data provided by the NBA (see, `http://stats.nba.com/`). That includes **season type** data, **conference**, **division**, **team**, as well as the **player** based statistics for the period of last 30 years. To efficiently store and parse this vast amount of information, we employed the `goldsberry` library in `Python`.

Features

To optimize data analysis we selected a normalized feature vector of size **50** that aggregate team performance throughout the season. That included

- **Defensive rebound**
- **3pt field goals** % and **field goals attempted**

Features

To reduce the variance associate with over-fitting we applied the `Extra Tree Classifier`. That allowed us to reduce the number of features to **17** when producing the outcome of the game based on the full seasonal data and **15** when predicting game's winner using only quarterly game data.

Models

To identify the best model we capitalized upon machine learning libraries of `Scikit.py`. That allowed us to train and test a vast variety of different models, including

- Lasso
- Logistic Regression
- Orthogonal Matching Pursuit
- SVM with linear, RBF and polynomial kernels

Important Result

Model	Mean-Squared Error	Training Accuracy	Testing Accuracy
SVM w/ RBF	0.321	0.673	0.676
Logistic Regression	0.326	0.672	0.671
Algo SVM w/ Sigmoid	0.329	0.671	0.678
K-Neares Neighbors	0.331	0.747	0.666
Ridge Regression	0.592	0.192	0.096

Table 1: Errors and Scores of most successful methods

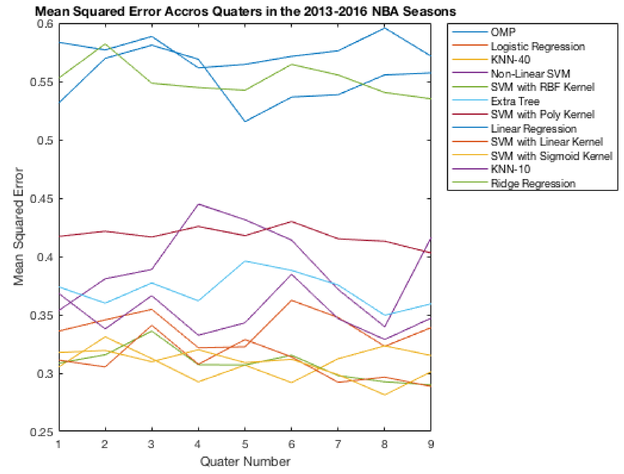
Discussion

The project allowed us to get acquainted with a wealth of machine learning procedures. In addition to obtaining better results by employing an innovative partition of the seasonal data, we also learned which techniques are more adequate for different sets of data. In general, it should be stated that the best results in predicting the winner of the NBA matchup across different partitions of the original data set were obtained using the **SVM** model with the RBF kernel. These results are compatible with the theoretical understanding of the inter-workings of the method. That is, given a large feature space we expect the non-linear kernel to be able to model the complexities of the data better than its linear counterpart.

Results

We trained and tested several models using seasonal, as well as quarterly data. In general we registered a steady trend of **32-35%** mean squared error across predictions for the full seasonal data. The best result with **32.3%** mean squared error was delivered by the SVM with RBF kernel model.

A slightly better predictions were developed using only quarterly data. Here, the mean squared error ranged between **29%** and **33%**, with the best results coming from the Logistic Regression and the SVM with the RBF kernel (**29.2%** and **28.9%** respectively). These results are better than outside work [1], [2], [3], with the later have prediction error of 31.4% when using Multilayer Perceptron (It should be noted that our results were bolstered by capitalizing upon the rich data provided by the NBA API).



Future Work

- Develop a near-continuous time based prediction of matchups winner.
- Cluster game data to allow for prediction within a game by determining its similarity to past games.
- Integrate player-based features and their contribution to the game.
- Explore the value older data (the NBA API goes all the way back to the 1984-85 season) to develop team and player based clusters impacting the game.
- Incorporate live data into prior updates.

References

[1] Logan Short Jasper Lin and Vishnu Sundaresan. Predicting national basketball association winners. 2014.

[2] Ali Reza Sharafat Omid Aryan. A novel approach to predicting results of nba matches. 2014.

[3] Renator Amorim Torres. Prediction of nba games based on machine learning methods. University of Wisconsin, Madison, 2013.

Acknowledgements

We would like to extend the most heartfelt thanks to Bradley Fay whose development of the py-Goldberry facilitated the data extraction from the NBA API.