UNIVERSITY OF NEW SOUTH WALES, SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Predicting Basketball Hall of Fame Membership

COMP9417 - Machine Learning and Data Mining: Project

Chi-Chun Tiao

June 9, 2013

1. INTRODUCTION

Enshrinement into the Naismith Memorial Basketball Hall of Fame is an honour bestowed upon exceptional basketball players, coaches, referees, executives and other major contributors to the sport across the world. As one might expect, the population of inductees is dominated by members in the players category - 158 members in the players category out of a total of 326 members as of the enshrinement class of 2013[1]. Of the 158 players in the Hall of Fame, 103 played in the National Basketball Association (NBA), easily rendering it the league with the most players in the Hall of Fame[1]. As such, we shall limit our discussion to individuals who have played in the NBA. [2]

Unlike other major sport's Hall of Fames where sportswriter voters openly debate their choices, the Naismith Hall of Fame has an opaque and closely held selection process[2]. As a consequence of the screening committee's lack of transparency, the criteria for induction is vague at best and discussion regarding the merits of players is subjective. It would be useful to form an objective basis upon which a player's merits can be analysed and used to predict whether he will be inducted into the Hall of Fame. The goal of this project is to develop such a method using machine learning techniques.

[1] Other inductees of this category include college, international and female players, etc., most of whom have their own shrines already: The College Experience (NCAA), the FIBA Hall (international) and the WBHOF (women's).

[2] Indeed, this inevitably places prejudice against players who had decorated international or college careers, but relatively short or less impressive stints in the NBA, but might nevertheless be considered worthy of Hall of Fame induction.

Currently, some popular methods used include the *Keltner List*, a less objective but systematic method adopted from baseball (as many advanced basketball metrics often are) and also a logistic regression model[4], developed by *Basketball Reference*, a site that provides comprehensive basketball statistics. However, no attempts have been made to develop a machine learning model (to the best of my knowledge). Work has been done to solve the same problem in baseball using a Radial Basis Function Network by Smith, Lloyd and Downey, James (2009)[5], and some of the methods here shall be based upon this.

In this project, I strive to harness the capabilities and advantages of machine learning techniques to build a model with better predictive accuracy than that of the aforementioned methods, by experimenting with various different datasets, features and learning algorithms. This report documents the methods employed, work done to collect and preprocess the data, experimentations involved and analysis of the various results.

2. EXPERIMENTATION

2.1. DOMAIN-SPECIFIC FEATURES

In the initial stages of this project, simple basketball statistics were incorporated as features into the training set, alongside other metrics that might be indicative of the relative merit of particular players with respect to other players who played in the league during his time:

- Points Per Game
- Assists Per Game
- Rebounds Per Game
- Career Points
- Career Assists
- Career Rebounds
- Most Valuable Player Awards
- All-Star Game Appearances
- Year Played in the League

Taking into account the fact that a player must be fully retired for five years before being eligible for induction[6] and that the inaugural NBA All-Star Game took place in 1951, we used data from players who debuted in the league during or after the 1951-52 season and retired before the 2008-09 season. This not only avoids differences in basketball eras but ensures comprehensive statistics for each player (no missing or incomplete features for any instance) as all the above statistics were being recorded in that time period. Other key statistics such as *steals*, *blocks* and *field goal percentage* were not incorporated for this very reason, as many players either did not have these statistics recorded at all or only had it recorded for some seasons. Other metrics like player's *draft pick*, *weight* and *height* were also considered but these failed to improve performance.

In an effort to improve predictive accuracy on this model built on a relatively small number of features, several advanced basketball statistics were considered. Namely, *win shares*[9], a metric adopted from baseball that is designed to estimate a player's contribution to his team's win, which is made up of two other metrics, *defensive win shares* and *offensive win shares*, each of which considers a player's offensive and defensive contributions respectively. Since it has been said that certain players are inducted into the Hall of Fame based but on their defensive prowess, which may not reflect well on a stat sheet, I decided it would be beneficial to incorporate both defensive and offensive win shares. Another statistic considered was the *Player Efficiency Rating (PER)*, a rather arcane but popular metric which summarises a player's performance and contributions into a one number using a very detailed formula[10]. Finally, the statistics *effective field goal percentage (eFG%)* and *true shooting percentage (TS%)* were considered, both of which incorporate 3-point and 2-point field goals, the former of which adjusts for the fact that a 3-point field goal is worth one more point than a 2-point field goal and the latter also considers free throw percentage.

The last metric considered was the number of *NBA Championship Titles* the player has won and the player's *longest stint with any team*. The latter is potentially interesting and valuable as this could serve as an indication that he was a franchise player, an elite status bestowed upon (usually) the best players on their respective teams, whom the franchise is built around. Unfortunately, this was not experimented with due to the difficulty in obtaining this feature.

As noted earlier, since many statistics were not being recorded prior to the 1973-74 season, we cannot simply incorporate these advanced basketball metrics into our model as most rely on these missing statistics. Luckily, *Basketball Reference* provides a complete system with estimations to account for missing statistics in particular eras, the formulae for which are quite detailed. The method for calculating win shares and player efficiency rating can be found at `http://www.basketball-reference.com/about/ws.html` and `http://www.basketball-reference.com/about/per.html`.

2.2. Data Collection and Preparation

A considerable challenge faced in this project was the absence of a readily available data repository relevant to the problem, so a substantial amount of work was devoted overcoming this by web data extraction (scraping) and parsing data into a format that could be analysed with machine learning software. The documentation of the data collection process is kept as brief as possible since while part of the overall machine learning and data mining process, it has little to do with machine learning itself.

At first, official NBA stats portal `http://stats.nba.com/` seemed like a promising avenue for data collection. However, not only were there no datasets available for download, the site did not provide any APIs. To avoid resorting to scraping the pages, I inspected the source of the site's pages to find ways to access the backend scripts that drove the site's data and was able to find Javascript sources that provided results in a friendly Javascript Object Notation (JSON) format. Since each source only provided a subset of features for the particular player queried, I was required to cross reference these data sources to create a complete set of features for each player. The data sources and respective features provided are described below:

- `http://stats.nba.com/stats/commonallplayers` - *List of all NBA players*

- `http://stats.nba.com/stats/commonplayerinfo/?PlayerID=<player_id>` - *All-Star Appearances (Not available for active players), Position*

- `http://stats.nba.com/feeds/players/profile/<player_id>_Profile.js` - *Draft Pick, Weight, Height*

- `http://stats.nba.com/feeds/players/awards/<player_id>_Award.js` - *Most Valuable Player Awards, Hall of Fame Inductee*

- `http://stats.nba.com/stats/playerprofile` - *Points Per Game, Assists Per Game, Rebounds Per Game, Career Points, Career Assists, Career Rebounds*

At this stage of the data collection process, I hadn't conceived of using the advanced statistics described in the previous section, so these features seemed to be enough to get started on preliminary experimentation. This data source obviously failed to yield the number of championship titles, the aforementioned advanced statistics and the full set of simple statistics that were required to calculate complex metrics like the PER. This posed a problem after no further improvements could be obtained with the original dataset and its limited feature space and an alternative data source was required.

Since Basketball Reference provides all of the aforementioned features, with all of the advanced metrics already calculated (and much more), I finally resorted extracting data from its web pages since there was no API or datasets available for download. The list of players and their respective identifiers were obtained from `http://www.basketball-reference.com/players/<first_letter_last_name>` and a players complete profile could be obtained from `http://www.basketball-reference.com/players/<first_letter_last_name>/<identifier>.html`.

The upside of this was that unlike the official NBA stats portal where it was necessary to piece together data from different sources, it was possible to obtain all the information required from a single page. Using the Python library *Beautiful Soup*, I was able to extract all the required features for every player. Additional statistics were also obtained in case they were required, such as statistics from Playoff games, All-Star Games, etc. This data source seemed far more comprehensive, and appeared to also include players from the American Basketball Association (ABA), which was later merged with the NBA in 1976. This also meant a larger training set to work with.

Furthermore, the Most Valuable Player Award data is expressed in terms of cumulative award shares for the player's career, which is calculated by the number of points a player received for a particular award over the total points of all first-place votes[7] - far more useful than merely the raw number of MVP awards in the case of players such as Jason Kidd, who has never won the award but has been a serious contender in many seasons due to his stellar performance.

Both of the aforementioned datasets have been saved in the ubiquitous *JSON* format and is included with the submission files. Both datasets can also be parsed into Weka's *ARFF* or vectorised to work with the Python machine learning library *scikit-learn* with the provided scripts.

2.3. Training the Model

In the initial experiment, the original dataset described in the previous section was used and trained on using different learning algorithms with the popular machine learning workbench *Weka*[8]: *Logistic Regression*, *Multilayer Perceptrons*, *Decision Trees* and *Radial Basis Function Networks*, using the default parameters for each respective algorithm. To remove irrelevant instances, the dataset filtered out players who played for less than one season (this might be a poor criterion of relevance, since players who play for many seasons don't play many actual games. Games played were not available in this dataset so could not be filtered accordingly).

The performance of these algorithms were estimated with 10-fold cross-validation repeated 10 times and the predictive accuracy of models were compared using a paired t-test. Surprisingly, the logisitic regression model outperformed all other models, but not by a statistically significant amount, with the exception of the tree learner. As seen in table 2.1, the percentage of correct classifications under these data schemes were all estimated to be above 97%.

Table 2.1: Predictive Accuracy (Percent Correct)

Dataset	Logistic	RBF Network	Multilayer Perceptron	Decision Tree (J48)	
Initial NBA Data	98.08±0.83	97.79±1.01	97.83±0.83	97.41±0.93	•

∘, • statistically significant improvement or degradation

The suspiciously (and somewhat deceptively) good predictive accuracy attained by these learning schemes might be explained by the skewness of the training set toward negative examples. Indeed, the prestigious honour of enshrinement into the Hall of Fame is only bestowed upon a very select few individuals - 80 out of the 1875 players (4.2%) in the training set (in fact one could attain 95.8% predictive accuracy simply by classifying all instances as negative!).

Consequently, it was necessary to consider other cost-sensitive metrics to evaluate the performance of the learning schemes, the key one being *recall*.

Table 2.2: Recall

Dataset	Logistic	RBF Network	Multilayer Perceptron	Decision Tree (J48)
Initial NBA Data	0.67±0.16	0.70±0.16	0.65±0.16	0.60±0.18

∘, • statistically significant improvement or degradation

In table 2.2, we indeed see that the employed learning schemes did not have high recall, so 30-40% of actual Hall of Fame members were being misclassified as not being in the Hall of Fame, with the RBF network model yielding the highest recall [3].

[3] Obviously, it was also important not to compromise the prestige of the Hall of Fame by classifying instances as

Table 2.3: Confusion Matrix

		Predicted		
		HOF	not HOF	Total
Actual	HOF	57	23	80
	not HOF	16	1779	1795
	Total	73	1802	1875

Table 2.3 shows the confusion matrix for the RBF network model. Optimisations were performed on this model [4] in an effort to minimise the false negatives and thereby increase recall. The first attempt was to optimise the parameters of the model using the `CVParameterSelection` meta-classifier in Weka, namely, number of clusters, minimum standard deviation for the clusters, and the random seed to be used by *k*-means clustering. Additionally, I tried to use Weka's feature selection options to either eliminate features that yielded little information gain, or otherwise transform the possibly correlated features into linearly uncorrelated features using principle component analysis. Unfortunately, these efforts were mostly in vain, as these had little to no effect on the predictive accuracy of the model or its ability to correctly classify positive instances. Some minor improvements where obtained, however, with a cost-sensitive meta-classifiers using two different methods: reweighting training instances according to the total cost assigned to each class and predicting the class with minimum expected misclassification cost[11]. The respective results of which can be seen in table 2.4. Note the overall predictive accuracy remained practically unchanged at around 97%.

Table 2.4: Confusion Matrix

		Predicted		
		HOF	not HOF	Total
Actual	HOF	61	19	80
	not HOF	24	1771	1795
	Total	85	1790	1875

(a) reweighted training instances

		Predicted		
		HOF	not HOF	Total
Actual	HOF	59	21	80
	not HOF	21	1774	1795
	Total	80	1795	1875

(b) minimize expected misclassification cost

This was a good improvement, but I wanted to see if further advances could be made. Unfortunately, as mentioned in the data preparation section, this initial experiment only used the very basic set of features and more features were required at this point to make progress on this problem.

With the new dataset acquired from Basketball Reference, players who played less than 100 games were filtered out, along with any players who did not have the complete records of their

positive too liberally.

[4] There was no reason why the other learning schemes could not have been optimised as well. Radial Basis Function networks seemed especially appealing and promising due to the results achieved in solving the problem for the baseball Hall of Fame.

respective statistics. Consequently, the training set consisted of all the features described in the previous section (including the features in the original dataset), with a total of 1591 players and 85 Hall of Fame members. Initial experimentation with this dataset used all of the features in training the new models.

Table 2.5: Predictive Accuracy (Percent Correct)

Dataset	Logistic	Decision Tree (J48)	Multilayer Perceptron	RBF Network
Basketball Reference Data	97.81±1.04	97.27±1.17	97.83±0.98	97.18±1.11

◦, • statistically significant improvement or degradation

Again, as shown in table 2.5, the models were all of comparable predictive accuracy. Upon further inspection, table 2.6 shows RBF Network model yielded higher recall.

Table 2.6: Recall

Dataset	Logistic	Decision Tree (J48)	Multilayer Perceptron	RBF Network
Basketball Reference Data	0.74±0.15	0.68±0.16	0.73±0.15	0.80±0.13

◦, • statistically significant improvement or degradation

Table 2.7 shows the confusion matrix for the RBF network model. As one might expect, though there were relatively few false negatives, the false positives were quite high. Consequently, I sought to strike a good balance between precision and recall using similar approaches taken in the initial experiment.

Table 2.7: RBF Network Model - Confusion Matrix

		Predicted		
		HOF	not HOF	Total
Actual	HOF	68	17	85
	not HOF	28	1478	1506
	Total	96	1495	1591

Precision	0.708
Recall	0.8
F-Measure	0.751
Percent Correct	97.17%

Firstly, by analysing the various new features used, a few of them yielded relatively low information gain, and were thus removed. The features that reduced overall predictive accuracy or had little effect were TS%, eFG%, total games played, and the offensive and defensive win shares respectively. Table 2.8 shows the improvements gained from this.

Table 2.8: RBF Network Model (irrelevant features removed) - Confusion Matrix

		Predicted		
		HOF	not HOF	Total
Actual	HOF	67	18	85
	not HOF	20	1486	1506
	Total	87	1504	1591

Precision	0.77
Recall	0.788
F-Measure	0.779
Percent Correct	97.61%

Finally, I sought to optimise the parameters of the RBF network model. By using 3 as the clustering seed with 2 clusters and the minimum standard deviation of 0.1 in addition to removing the win shares feature. Good results were obtained, as depicted in table 2.9.

Table 2.9: RBF Network Model (final model) - Confusion Matrix

		Predicted		
		HOF	not HOF	Total
Actual	HOF	68	17	85
	not HOF	17	1489	1506
	Total	85	1506	1591

Precision	0.8
Recall	0.8
F-Measure	0.8
Percent Correct	97.86%

A comparison of the model's F-measure subject to the variations of parameters of the model and features of the dataset can be seen in table 2.10. The overall predictive accuracy across these models remained unchanged.

Table 2.10: F-measure

Dataset	All features	Filtered features 1	Filtered features 2
RBF Network (default)	0.75± 0.11	0.77±0.11	0.78±0.11
RBF Network (optimised parameters)	0.74± 0.11	0.77±0.11	0.79±0.10

◦, • statistically significant improvement or degradation

Evidently, the final model achieved a good balance between precision and recall as a result of optimisations and feature selection. These approaches were experimented on the different learning schemes, logistic regression, tree learning and multilayer perceptrons but none outperformed the RBF network model, even with cost-sensitive classifiers. The learning schemes that did perform well were on par with the RBF network model.

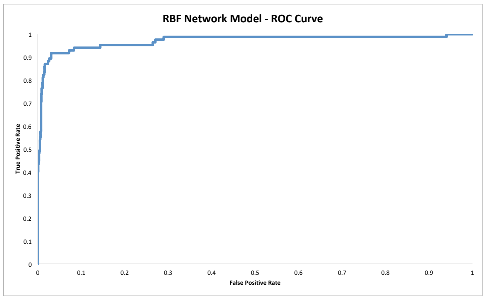

Figure 2.1: ROC Curve

The ROC curve of the RBF Network model is depicted in 2.1 with an area of 0.971 under the curve (AUC).

3. Predictions

Using the predictive model developed in the previous section, it would be interesting for practical purposes to apply domain knowledge to reason about the misclassifications made by the model and also to analyse the generalisation power of the model by evaluating it on active or recently retired players.

4. Conclusion

A. Appendix

References

[1] *"Hall of Famers"* Retrieved 6 June 2013 from Naismith Memorial Basketball Hall of Fame: `http://www.hoophall.com/hall-of-famers-index/`

[2] Aschburner, Steve. (12 August 2012) *"Hall of Fame selection process leaves much to be desired"*. Retrieved 6 June 2012 from NBA: `http://www.nba.com/2010/news/features/steve_aschburner/08/12/hall.process/index.html`

[3] Ziller, Tom. (30 March 2010) *"Fans to Vote for Basketball Hall of Fame Inductees"*. Retrieved 1 June 2013 from AOL News: `http://nba.fanhouse.com/2010/03/30/fans-to-vote-for-basketball-hall-of-fame-inductees/`

[4] *"Hall of Fame Probability"*. Retrieved 1 June 2013 from Basketball Reference: `http://www.basketball-reference.com/about/hof_prob.html`

[5] Smith, Lloyd and Downey, James (2009) *"Predicting Baseball Hall of Fame Membership using a Radial Basis Function Network,"* Journal of Quantitative Analysis in Sports: Vol. 5: Iss. 1, Article 6.

[6] *"Guidelines For Nomination and Election Into the Naismith Memorial Basketball Hall of Fame"* Retrieved 7 June 2013 from Basketball Hall of Fame: `http://www.hoophall.com/enshrinement-process/`

[7] James Bowman *"WNBA Most Valuable Player Shares"* Retrieved 7 June 2013 from Swish Appeal: `http://www.swishappeal.com/2010/2/21/1320444/wnba-most-valuable-player-shares`

[8] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009) *"The WEKA Data Mining Software: An Update,"* SIGKDD Explorations, Volume 11, Issue 1.

[9] *"Calculating Win Shares"* Retrieved 7 June 2013 from Basketball Reference: `http://www.basketball-reference.com/about/ws.html`

[10] *"Calculating PER"* Retrieved 7 June 2013 from Basketball Reference: `http://www.basketball-reference.com/about/per.html`

[11] *"CostSensitiveClassifier"* Retrieved 7 June 2013 from Weka Documentation: `http://weka.sourceforge.net/doc/weka/classifiers/meta/CostSensitiveClassifier.html`