Shot Recommender System for NBA Coaches*

[Extended Abstract] †

Raymond E. Wright
SAS Institute Inc.
100 SAS Campus Dr.
Cary, NC 27513
ray.wright@sas.com

Jorge Silva
SAS Institute Inc.
100 SAS Campus Dr.
Cary, NC 27513
jorge.silva@sas.com

Ilknur Kaynar-Kabul
SAS Institute Inc.
100 SAS Campus Dr.
Cary, NC 27513
ilknur.kaynar-kabul@sas.com

ABSTRACT

In basketball, knowing the odds of success of a shot is critical. Many factors affect shot success, including the location of the shot, the shot style (e.g., jump shot, finger roll, dunk), and of course the player's skill. Crucially, the list of high-probability shots is different for each player. We predict the success of shot made by NBA basketball players in the 2015-2016 season, and show that using traditional methods such as logistic regression or support vector machine regression is problematic because many style-location-player combinations do not occur in the training data, i.e., the data are highly sparse. Hence, we propose a shot recommender system based on factorization machines. Factorization machines have been used successfully in recommendation problems because they handle sparse data, scale well to very large datasets, and provide latent factors that capture underlying rater (player) preferences and item (shot) features. For the NBA player data, a 25-factor model predicts log-odds of shot success with high accuracy. It also identifies both highly recommended shots and shots to be avoided, including shots that are not represented in the training data.

Keywords

Sports; Basketball; Matrix factorization; Factorization machines; Recommender systems

1. INTRODUCTION

Predicting shot success in basketball is a challenging task. Shots are influenced by many factors besides player skill, such as location, shot style, range, and naturally also the actions of opposing players. One of the difficulties encountered by standard prediction methods is the fact that existing training data is sparse. In other words, not all player-style-range-location combinations occur in the data, which means that the outcomes of such combinations are unknown. The proportion of missing data is too high to be handled by simple imputation techniques. The same sparsity problem is also encountered in the area of recommender systems [7].

Recommender systems are a diverse class of algorithms which aim to learn user preferences in order to recommend items (e.g., movies, book, songs). In this paper, we consider that the ability to automatically suggest player- and situation-specific shots may be useful in its own right to basketball coaches, players and other interested parties. Therefore, we have chosen to explore similar methods. Specifically, we apply a factorization machine [5, 6] model and compare its performance to other models, namely logistic regression and support vector machines. As an added benefit, the factorization machine (FM) model learns latent features which characterize players and shot types.

1.1 Prior research

While there is significant literature on machine learning applied to basketball [1], little of it explores the use of recommender systems. One notable exception unrelated to basketball is [12], where recommender systems are advocated as a decision support tool for sports in general. Forecasting the outcome of basketball games is addressed in [10]. In [3], player and ball tracking data are used to analyze offensive and deffensive formations. For the task of shot prediction, in [8] the authors predict free-shot success based on visual cues, while [2] is based on biomechanical variables. Non-negative matrix factorization is employed in [4], using spatial information. In [11], a conditional random field model is used for predicting events (pass, shoot, hold). To the best of our knowledge, ours is the first work to quantitatively evaluate the predictive performance of a recommender system for shot success based on factorization machines on sparse NBA data.

1.2 Data

The data for this paper consist of recorded shots taken during the 2015-2016 basketball season, from the beginning through March 2016. The dataset was downloaded using the API avalable from the NBA.com website. Each shot taken by all players is recorded, excluding free throws. Figure 1 exemplifies how shot success varies by court location and player position.

*(Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

†A full version of this paper is available as *Author's Guide to Preparing ACM SIG Proceedings Using LaTeX2ε and BibTeX* at `www.acm.org/eaddress.htm`

KDD '16, CA USA
© 2016 ACM. ISBN 123-4567-24-567/08/06.
DOI: 10.475/123_4

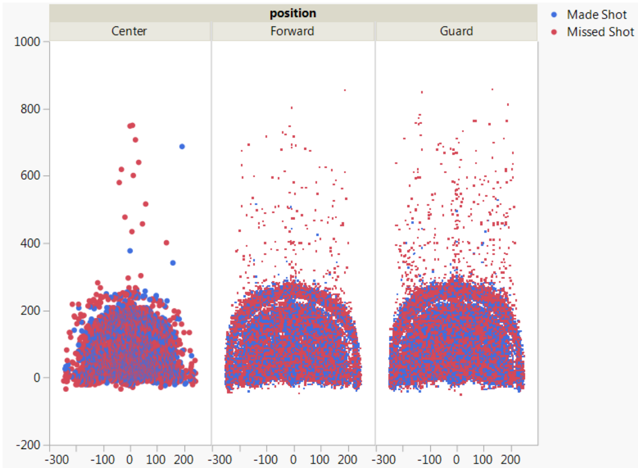

Figure 1: Shot success by court location and player position.

There are 420 players in the dataset. Information for each shot includes the player, shot zone range (e.g., < 8 feet, 8-16 feet, etc), shot zone area (center, left, right center, etc.), whether the shot was successful, and other characteristics. Of particular interest in this paper is the Action Type variable, which records the style of each shot. For example, “Alley Oop Dunk Shot”, “Finger Roll Dunk Shot”, “Turnaround Fadeaway Dunk Shot”, and “Driving Reverse Layup Shot”. There are 53 action type categories, which are sparsely represented in the dataset. That is, while players vary, any individual player tends to take shots of only a handful of the possible shot styles.

2. FACTORIZATION MACHINES

Here we provide a brief technical description of factorization machines. Assuming a training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$ with $i = 1, \ldots, n$, where $\mathbf{x}_i$ refers to the i-th input feature vector and y_i to the i-th label, the factorization machine model of order two is written as

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j x_j + \sum_{j=1}^{p} \sum_{j'=j+1}^{p} x_j x_{j'} \sum_{f=1}^{k} v_{jf} v_{j'f} \qquad (1)$$

where $\mathbf{x} = (x_1, \cdots, x_p)$ is an observed p-dimensional input feature vector, $\hat{y}$ is the predicted target, w_0 is a global bias, w_j are per-feature biases, and v_{jf} denotes coordinate f of the vector $\mathbf{v}_j \in \mathbb{R}^k$. The overall factor matrix $\mathbf{V} \in \mathbb{R}^{p \times k}$ is the concatenation of the row vectors $\mathbf{v}_j$ for $j = 1, \ldots, p$. The number of factors is k. The model parameters to be estimated are $w_0, w_1, \ldots, w_p$ and $\mathbf{V}$. The estimation is done by minimizing the root mean square error (RMSE), which is defined by

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2} \qquad (2)$$

over the training set. The vectors $\mathbf{v}_j$ provide a latent characterization of feature j in $\mathbf{x}$.

Interestingly, it is known that factorization machines approximate polynomial-kernel support vector machines, and are more robust to overfitting when the design matrix is sparse [5].

3. METHODOLOGY

Our primary goal is to predict shot success using the following inputs: player, shot zone range, shot zone area, and action type. These are all nominal (categorical) inputs. After pre-processing we retain 359 players, 4 zone ranges, 5 zone areas, 36 action types. Thus, following the notation in (1), and using one-hot encoding, we have $p = 404$ features in $\mathbf{x} = (\text{player}, \text{action}, \text{area}, \text{range})$. Of particular interest is, for a given player, whether we can accurately predict their probability of making a shot with particular characteristics. This amounts to estimating a logistic regression-type model. Prior to building the model, we aggregate the shots taken by each player and compute the proportion P_i of success of the i-th player / shot combination. As proportions with small denominators (e.g., 3/5) tend to be unstable, we filter out any player-shot combination that occurs fewer than 10 times in the training set. Because using proportions as a target variable could result in predictions outside the 0,1 range, we transform the target from the proportion of shots made to a log-odds scale. Hence we define the target $y_i = \frac{P_i}{1-P_i}$ for the regression task.

3.1 Model for shot prediction

Traditional methods like logistic regression are inappropriate due to sparsity, particularly for estimating interaction terms such as player * action type * shot zone range * shot zone area. In fact, we were unable to fit even the simpler model $\hat{y} = w_0 + w_{\text{player}} + w_{\text{action}} + V_{\text{player,action}}$ as the problem was ill-conditioned due to sparsity.

Thus, we employ the following factorization machine model for basketball shot prediction.

$$\hat{y}(\mathbf{x}) = w_0 + w_{\text{player}} + w_{\text{action}} + w_{\text{area}} + w_{\text{range}} + \langle \mathbf{v}_{\text{player}}, \mathbf{v}_{\text{action}} \rangle + \cdots + \langle \mathbf{v}_{\text{area}}, \mathbf{v}_{\text{range}} \rangle . \qquad (3)$$

The sum in (3) includes all pairwise interactions between player, action type, shot zone area and shot zone range. The notation $\langle \cdot, \cdot \rangle$ represents inner product.

4. RESULTS

In our experiments, we randomly hold out 20% of the data and use the remainder for training. We estimate a FM model to predict log-odds of shot success. We then evaluate the RMSE criterion on the held-out set and compare with a support vector machine (SVM) regression model. The comparison is shown in table 1. The FM model uses $k = 25$ factors and was optimized using stochastic gradient descent. The best performance for the SVM model is with a 2nd degree polynomial kernel and 25 support vectors, but it requires considerable tuning effort and training is far slower than FM. Neither a linear kernel nor a Gausian radial basis function kernel perform any better. This is not unexpected, and it does not mean that factorization machines outperform SVMs in general – but it is well documented that they do consistently outperform SVMs when the data are sparse [5]. Conversely, a SVM model should do better with dense design matrices. Also, while we tried standard logistic regression, we ran into ill-conditioning problems due to the sparsity in the dataset. Using ℓ_2–regularized logistic regression did mitigate ill-conditioning, but was not sufficient to obtain competitive accuracy.

4.1 Model recommendations

Table 1: Test set accuracy

Method	RMSE
SVM	1.39092
FM	0.92956

Besides achieving good prediction performance, the factorization machine model yields bias terms and latent factor vectors. The bias terms are highly informative by themselves. For instance, Figure 2 show that it is possible to differentiate between player positions based on the player bias terms. The figure shows box plots for all the w_{player} biases grouped by player position, and it is apparent that the mean and variance of center players significantly differ from forward and guard players.

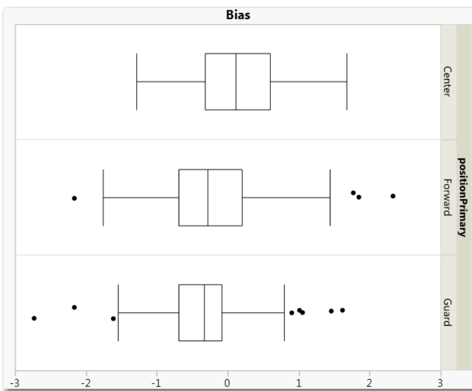

Figure 2: Player bias w_{player} values grouped by position. Dots represent outliers.

In addition, Figure 3 shows the action type bias values. Higher values are associated with higher log-odds of success. It is striking to notice that dunk shots dominate all other types of action.

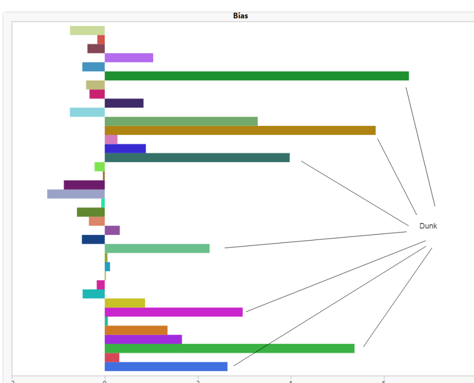

Figure 3: Bias for dunk-type shots.

One further advantage of employing a recommender system is that we can suggest new action-range-area combinations by scoring them and sorting according to predicted logit value. Tables 2 and 3 display the action types with the highest and lowest expected success scores, respectively.

Dunk shots are overrepresented among the successful shots, which seems to confirm intuition. In fact, whenever a player

Table 2: Most recommended action types

Description	p logit
Running Dunk Shot	0.74636
Running Layup Shot	0.73087
Driving Layup Shot	0.71003
Alley Oop Sunk Shot	0.69447
Dunk Shot	0.69338
Cuting Dunk Shot	0.68687
Putback Layup Shot	0.65994
Driving Dunk Shot	0.64115
Tip Dunk Shot	0.63171
Driving Dunk Shot	0.62458

Table 3: Least recommended action types

Description	p logit
Turnaround Hook Shot	0.51875
Turnaround Jump Shot	0.51449
Fadeaway Jump Shot	0.50921
Driving Floating	0.50515
Turnaround Fadeaway Shot	0.50380
Running Jump Shot	0.49988
Hook Shot	0.49475
Step Back Jump Shot	0.47068
Pullup Jump Shot	0.42709
Jump Shot	0.18849

misses a dunk, that usually comes as a surprise and can prove quite entertaining for fans of the opposing team. The non-recommended shots tend to be jump shots and more complicated combinations involving contortions and acrobatics, e.g, the turn around fadeaway. It is somewhat surprising, however, that jump shots rank so low, considering as they are so common.

We also used the model to score every possible combination of the shot characteristics (action type, shot zone range, shot zone area). This allowed us to identify highly recommended shots as well as shots to be avoided, including some combinations of shot characteristics that did not appear in the training data. The combinations are shown in tables 4 and 5. Since these scores ignore player, they can be regarded as recommendations for a typical or 'unknown' player.

All of the most recommended shots are dunk shots taken within 8 feet of the basket. Note that, of the five top shots, only the top two of these combinations actually occurred in the training data.

As for the least-recommended shots, these are predominantly jump shots taken near the basket. The low success rate for these shots may be due to being highly contested by defenders. Interestingly, none of the top five worst shot combinations occurred in the training data.

Table 4: Most recommended shots

plogit	Action Type	Range	Area
6.60419	Cutting Dunk Shot	< 8 ft.	Center
6.90675	Tip Dunk Shot	< 8 ft.	Center
6.90675	Cutting Dunk Shot	< 8 ft.	Left Center
6.90675	Cutting Dunk Shot	< 8 ft.	Right Center
6.90675	Cutting Dunk Shot	< 8 ft.	Right

Table 5: Least recommended shots

plogit	Action Type	Range	Area
-2.41169	Jump Shot	< 8 ft.	Left
-2.33894	Reverse Layup	< 8 ft.	Right
-2.28626	Tip Layup	< 8 ft.	Right
-2.14424	Jump Shot	< 8 ft.	Right
-2.06419	Jump Shot	< 8 ft.	Right Center
-1.97495	Jump Shot	< 8 ft.	Center
-1.93047	Hook Shot	< 8 ft.	Right Center
-1.9031	Turnaround Jump	< 8 ft.	Right Center
-1.82045	Jump Bank	< 8 ft.	Left
-1.77857	Turnaround Fadeaway	8-16 ft.	Center

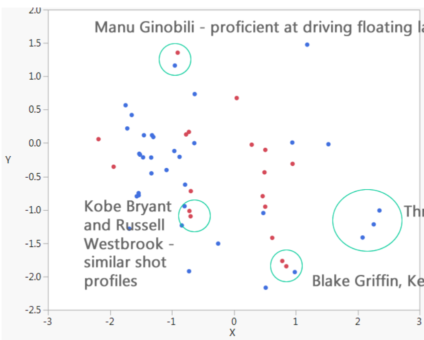

Figure 4: t-SNE embedding of player (red) and action (blue) factors. Best viewed in color.

Of particular interest for coaching individual players, the model can make recommendations for players with relatively short (sparse) or even nonexistent shot histories. For example, consider Tony Wroten, a guard with 3 years NBA experience who plays for the New York Knicks. He took very few shots (17) in the season to date, all of which were layups attempted near the basket. Only 5.9% of the shots were successful. Given his shot profile and the relative difficulties of each combination of shot types, the model predicts that Tony Wroten would be successful at dunk shots from less than 8 feet, center (predicted logit = 2.48) as well as running dunk shots attempted from less than 8 feet, center (predicted logit = 2.94).

Additionally, Figure 4 shows the player factors and the action factors vectors on the same 2D plot. Since these are 25-dimensional vectors, an embedding is created by using t-Studentized stochastic neighbor embedding (t-SNE) [9], which is a non-linear projection method particularly suited for visualization. In the plot, points colored in blue are actions while those colored in red are players. Similar shots appear close together, as do players with similar shot profiles (Kobe Bryant and Russel Westbrook). Also, it appears from the figure that Manu Ginobili is proficient at driving floating layups, as his latent factor vector is embedded very near that of the corresponding action.

5. CONCLUSIONS

In this paper, we have proposed a recommender system approach for predicting shot success in NBA basketball games. Coaches can use this model to have better undertanding of overall factors affecting shot success, as well as for generating targeted recommendations for individual players, who may have little or no shooting history. By using a factorization machine model, we were able to overcome the sparsity of player-shot combinations in the dataset and outperform established predictive models such as SVMs. Using stochastic gradient descent optimization allows the model to scale to a large dataset, which is also of interest.

As a caveat, ours is a relatively simple model, even though it considers all pairwise interactions between inputs. We did not have access to individual or team defensive actions taken by the opposition, which surely affects shot performance. For example, whether a player is being defended by a defensive star, or is being double-teamed, shot success probability may decrease drastically. Also, one can not always dunk. Dunking requires getting close enough to the basket.

Future work will include using additional inputs such as spatiotemporal information, which is often considered in sports analytics (rightly so) and may allow generating real-time shot suggestions.

6. REFERENCES

[1] K. Goldsberry. Courtvision: New visual and spatial analytics for the nba. In *2012 MIT Sloan Sports Analytics Conference*, 2012.

[2] J. L. Hudson. Prediction of basketball skill using biomechanical variables. *Research Quarterly for Exercise and Sport*, 56(2):115–121, 1985.

[3] P. Lucey, A. Bialkowski, P. Carr, Y. Yue, and I. Matthews. How to get an open shot: Analyzing team movement in basketball using tracking data. *MIT Sloan Sports Analytics Conference*, 2014.

[4] A. Miller, L. Bornn, R. Adams, and K. Goldsberry. Factorized point process intensities: A spatial analysis of professional basketball. In *ICML*, pages 235–243, 2014.

[5] S. Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

[6] S. Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[7] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[8] Y. Uchida, N. Mizuguchi, M. Honda, and K. Kanosue. Prediction of shot success for basketball free throws: Visual search strategy. *European journal of sport science*, 14(5):426–432, 2014.

[9] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[10] P. O. Vaz de Melo, V. A. Almeida, A. A. Loureiro, and C. Faloutsos. Forecasting in the nba and other team sports: Network effects in action. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(3):13, 2012.

[11] Y. Yue, P. Lucey, P. Carr, A. Bialkowski, and I. Matthews. Learning fine-grained spatial models for dynamic sports play prediction. In *2014 IEEE International Conference on Data Mining*, pages 670–679. IEEE, 2014.

[12] J. Zeleznikow, C. MacMahon, and T. Barnett. Providing automated decision support for elite athletes. In *Proc. 7th International Symposium of Computer Science in Sport*, pages 240–248, 2009.