LinNet: Probabilistic Lineup Evaluation Through Network Embedding

Konstantinos Pelechrinis

University of Pittsburgh
kpele@pitt.edu

Abstract. Which of your team's possible lineups has the best chances against each of your opponent's possible lineups? To answer this question, we develop LinNet (which stands for LINeup NETwork). LinNet exploits the dynamics of a directed network that captures the performance of lineups during their matchups. The nodes of this network represent the different lineups, while an edge from node B to node A exists if lineup λ_A has outperformed lineup λ_B. We further annotate each edge with the corresponding performance margin (point margin per minute). We then utilize this structure to learn a set of latent features for each node (i.e., lineup) using the **node2vec** framework. Consequently, using the latent, learned features, LinNet builds a logistic regression model for the probability of lineup λ_A outperforming lineup λ_B. We evaluate the proposed method by using NBA lineup data from the five seasons between 2007-08 and 2011-12. Our results indicate that our method has an out-of-sample accuracy of 68%. In comparison, utilizing simple network centrality metrics (i.e., PageRank) achieves an accuracy of just 53%, while using the adjusted plus-minus of the players in the lineup for the same prediction problem provides an accuracy of only 55%. We have also explored the adjusted lineups' plus-minus as our predictors and obtained an accuracy of 59%. Furthermore, the probability output of LinNet is well-calibrated as indicated by the Brier score and the reliability curve. One of the main benefits of LinNet is its generic nature that allows it to be applied in different sports since the only input required is the lineups' matchup network, i.e., not any sport-specific features are needed.

Keywords: Network Science · Network embedding · Sports Analytics · Probabilistic models.

1 Introduction

During the past decade or so, the availability of detailed sports data in conjunction with the success enjoyed by early adopters, has led to the explosion of the field of sports analytics. Part of this can be attributed to the advancements in computing technologies that have facilitated the collection of detailed (spatio-temporal) data that can shed light to aspects of the sport(s) in question that were not possible before. For example, since 2013 a computer vision system installed in all NBA stadiums collects the location of all players on the court and

the ball 25 times every second. Using this information the Toronto Raptors were able to (manually) identify the optimal position for defenders given the offensive scheme. These *optimal* defenders are called *ghosts* and can be used to evaluate defensive skills, an aspect of the game severely underrepresented in traditional boxscore statistics [8]. Since then automated ways for ghosting, and in general for analyzing and understanding fine-grained in-game behavior, have been developed - in various sport - relying on the advancements in representation (deep) learning (e.g., [7, 16, 19, 10]).

However, representation learning can also help answer more *traditional* questions in a new way. For example, one of the decisions that a basketball coach has to constantly make during a game (or even for game preparation) is what lineup to play in order to maximize the probability of outperforming the opponent's lineup currently on the court. This lineup evaluation problem has been traditionally addressed through ranking lineups. More specifically, player and lineup ratings based on (adjusted) plus/minus-like approaches [18], or efficiency ratings (i.e., points scored/allowed/net per 100 possessions [1]) have been used to rank lineups. This ranking can then be used to evaluate which lineup is *better*. Nevertheless, these ratings do not explicitly account for the game situation/context. For instance, a lineup that outperformed its opponent by 10 points in *garbage* time does not provide us with the same information as compared to the same lineup outperforming its opponent during the start of the game. Hence, to account for this the use of in-game win probability models has been proposed [12]. In this case, instead of computing the net points scored in every stint[1] we calculate the win probability added by each of the two lineups.

To the best of our knowledge apart from these basic metrics that are used to rank lineups, there exist no studies in the public sphere that evaluate the predictive power of these metrics and/or introduce other ways of evaluating and predicting lineup matchups[2]. In the current study, we propose a completely different approach that is based on representation learning on networks. In particular, we first define the matchup network $\mathcal{G}$:

Definition 1.1: Matchup Network

The matchup network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, is a weighted directed network where nodes represent lineups. An edge $e_{i,j} \in \mathcal{E}$ points from node $i \in \mathcal{V}$ to node $j \in \mathcal{V}$ iff lineup j has outperformed lineup i. The edge weight $w_{e_{i,j}}$ is equal to the performance margin of the corresponding matchup.

Using the structure of this network we can learn a vector representation of the nodes. For this purpose we utilize a network embedding, which projects the

[1] Stint refers to a time-period during the game when no substitutions happen by either team.

[2] Of course, we expect professional teams to perform their own analysis - potentially beyond simply ranking - but their proprietary nature makes it impossible to study and evaluate.

network nodes on a latent space $\mathcal{X}$. In our study we adopt the **node2vec** [6] framework for learning the latent space. Simply put, the embedding learns a set of features $\mathbf{x}_u$ for node u. These features are then utilized to build a logistic regression model that models the probability of lineup λ_i outperforming lineup λ_j, $\Pr[\lambda_i \succ \lambda_j | \mathbf{x}_{\lambda_i}, \mathbf{x}_{\lambda_j}]$. Figure 1 visually represents the LinNet framework.

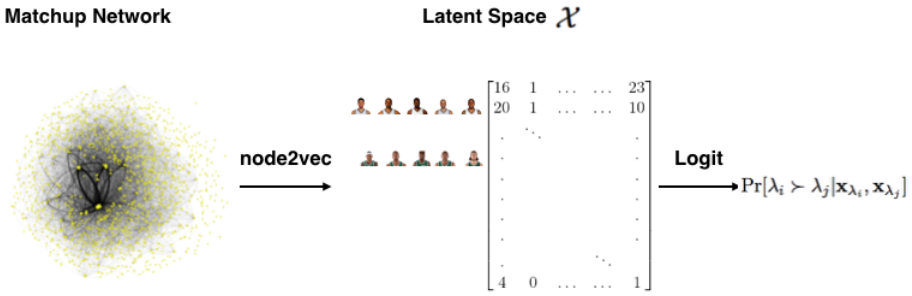

Fig. 1. The LinNet lineup evaluation framework

Our evaluations indicate that LinNet can predict the outcome of a lineup matchup correctly with approximately 68% accuracy, while the probabilities are well-calibrated with a Brier score of 0.19. Furthermore, the probability validation curve of LinNet is statistically indistinguishable from the $y = x$ line, i.e., the predicted matchup probability is equal to the *actual* probability (see Figure 3). Hence, the logistic regression model on the latent space $\mathcal{X}$ captures accurately the lineups' matchup probabilities. In comparison, we evaluate the following three baseline methods inspired both from current approaches in ranking lineups as well as network ranking; (i) a PageRank-based ranking using the same matchup lineup network $\mathcal{G}$, (ii) a model based on the adjusted plus/minus of the players that are part of each lineup, and (iii) a model based on the adjusted plus/minus of the lineups. The lineup adjusted plus/minus has the best performance among the baselines, but still worse than LinNet, with an accuracy of 59%.

The main contribution of our work is twofold:

- We introduce and evaluate a novel approach for evaluating basketball lineups in a probabilistic way using representation learning on networks.
- The proposed method is generic, i.e., it can be adopted in other sports without the need to incorporate sport-specific information.

We also hope that this study will trigger more interest and research in the applications of network science in sports. While network science methods have been used in the literature to study and answer sports analytics questions, these studies are primarily focused on analyzing the most straightforward network structure in sports, namely, passing networks, i.e., *who-passes-to-whom structures* (e.g., [5, 4, 14]). However, networks can also be used to represent complex

structures that might not be *visible* directly – such as the win-loss relationships of teams or lineups that we use in our current study – and can provide new and novel insights.

The rest of the paper is organized as following. In Section 2 we present in details the operations of `LinNet` as well as the datasets we used. Section 3 presents our results, while Section 4 concludes our study and discusses the implications and limitations of our work.

2 Materials and Methods

In this section we will present in detail (a) the design of `LinNet`, (b) the baseline methods for comparison, and (c) the datasets we used for our evaluations.

2.1 `LinNet`

The first step of `LinNet` is defining the matchup network $\mathcal{G}$. There is flexibility in choosing the performance margin that one can use for the edge weights. In the current implementation of `LinNet`, the weights of $\mathcal{G}$ correspond to the point margin per minute for the two lineups.

Once the network is obtained the next step is to learn the network embedding. As our network embedding mechanism we will utilize the approach proposed by Grover and Leskovec [6], namely, node2vec. node2vec utilizes (2^{nd} order) random walks on the network in order to learn the latent features of the nodes, i.e., a function $f : \mathcal{V} \rightarrow \Re^d$, where d is the dimensionality of the latent space. Starting from node u in the network and following the random walk strategy R the network neighborhood $N_R(u)$ of u is defined. Then node2vec learns the network embedding f by solving the following optimization problem:

$$\max_f \sum_{u \in \mathcal{V}} \log(\Pr[N_R(u)|f(u)]) \tag{1}$$

where $\Pr[N_R(u)|f(u)]$ is the (conditional) probability of observing $N_R(u)$ as the network neighborhood for node u. Simply put, the network embedding maximizes the log-likelihood of observing a network neighborhood $N_R(u)$ for node u conditioned on the network embedding f. To keep the optimization tractable, node2vec makes use of two standard assumptions; (i) conditional independence of the nodes in $N_R(u)$, and (ii) the probability of each source-neighborhood node pair is modeled through a softmax of the dot product of their features f (to be learned). When two nodes are *similar*, they are expected to appear within the same random walk often, and the optimization problem (1) ensures that they will be close in the embedding space.

The random walk strategy - which implicitly defines the *similarity* of two nodes - is defined by two parameters, p and q, that offer a balance between a purely breadth-first search walk and a purely depth-first search walk. In particular, the random walk strategy of node2vec includes a bias term α controlled by parameters p and q. Assuming that a random walk is on node u (coming from

node v), the unnormalized transition probability $\pi_{ux} = \alpha_{pq}(v,x) \cdot w_{ux}$. With d_{ux} being the shortest path distance between u and x we have:

$$\pi_{ux} = \begin{cases} 1/p & , \ if \ d_{ux} = 0 \\ 1 & , \ if \ d_{ux} = 1 \\ 1/q & , \ if \ d_{ux} = 2 \end{cases}$$

As alluded to above, parameters p and q control the type of network neighborhood $N_R(u)$ we obtain. Different sampling strategies will provide different embeddings. For example, if we are interested in having a set of nodes that are tightly connected in the original network, to be close to each other in the latent space, p and q need to be picked in such a way that allows for "local" sampling. In our application we are interested more in identifying structurally equivalent nodes, i.e., nodes that are similar because their connections in the network are similar (not necessarily close to each other with respect to network distance). This requires a sampling strategy that allows for the network neighborhood of a node to include nodes that are further away as well. Given this objective and the recommendations by Grover and Leskovec [6] we choose $q = 3$ and $p = 0.5$ for our evaluations. Furthermore, we generate 3,000 walks for each network, of 3,500 hops each, while, we choose as our latent space dimensionality, $d = 128$. Increasing the dimensionality of the space improves the quality of the embedding as one might have expected, however, our experiments indicate that increasing further the dimensionality beyond $d = 128$ we operate with diminishing returns (with regards to computational cost and improvement in performance).

Once the latent space $\mathcal{X}$ is obtained, we can build a logistic regression model for the probability of lineup λ_i outperforming λ_j. In particular, we use the Bradley-Terry model [2]. The Bradley-Terry model is a method for (probabilistically) ordering a given set of items based on their characteristics and understanding the impact of these characteristics on the ranking. In our case the set of items are the lineups and the output of the model for items i and j provides us essentially with the probability of lineup λ_i outperforming λ_j. In particular, the Bradley-Terry model is described by [2]:

$$\Pr(\lambda_i \succ \lambda_j | \pi_i, \ \pi_j) = \frac{e^{\pi_i - \pi_j}}{1 + e^{\pi_i - \pi_j}} \tag{2}$$

where π_i is λ_i's *ability*. Given a set of lineup-specific explanatory variables $\mathbf{z}_i$, the difference in the ability of lineups λ_i and λ_j can be expressed as:

$$\pi_i - \ \pi_j = \sum_{r=1}^{d} \alpha_r (z_{ir} - z_{jr}) + U \tag{3}$$

where $U \sim N(0, \sigma^2)$. The Bradley-Terry model is then a generalized linear model that can be used to predict the probability of λ_i outperforming λ_j. In our case, the explanatory variables are the latent features learned for each lineup, $\mathbf{x}_{\lambda_i}$.

Previously Unseen Lineups: One of the challenges (both in out-of-sample evaluations as well as in a real-world setting), is how to treat lineups that we

have not seen before, and hence, we do not have their latent space representation. In the current design of `LinNet` we take the following simple approach. In particular, for each lineup λ_i of team $\mathcal{T}$ we define the similarity in the players' space $\sigma_{\lambda_i,\lambda_j}$ of λ_i with $\lambda_j \in \mathcal{L}_\mathcal{T}$, as the number of common players between the two lineups (i.e., $\sigma_{\lambda_i,\lambda_j} \in \{0,\ldots,4\}$). One might expect that lineups with high overlap in the players' space, should also reside closely in the embedding space. In order to get a feeling of whether this is true or not, we calculated for every team and season the correlation between the similarity between two lineups in the players' space (i.e., $\sigma_{\lambda_i,\lambda_j}$) and the distance for the corresponding latent features (i.e., $\texttt{dist}(\mathbf{x}_i, \mathbf{x}_j)$). As we can see from Figure 2 all teams exhibit negative correlations (all correlations are significant at the 0.001 level), which means the more common players two lineups have, the more closely they will be projected in the embedding space. Of course, the levels of correlation are moderate at best since, the embedding space is obtained by considering the performance of each lineup, and two lineups that differ by only one player might still perform completely differently on the court. With this in mind, once we obtain the lineup similarity values, we can assign the latent feature vector for the previously unseen lineup λ_i as a weighted average of the lineups in $\mathcal{L}_\mathcal{T}$ (with σ being the weighting factor):

$$\mathbf{x}_{\lambda_i} = \frac{\sum\limits_{\lambda_j \in \mathcal{L}_\mathcal{T}} \sigma_{\lambda_i,\lambda_j} \cdot \mathbf{x}_j}{\sum\limits_{\lambda_j \in \mathcal{L}_\mathcal{T}} \sigma_{\lambda_i,\lambda_j}} \tag{4}$$

It should be evident that this is simply a heuristic that is currently implemented in `LinNet`. One could think of other ways to approximate the latent space features of a lineup not seen before.

2.2 Baselines

For comparison purposes we have also evaluated three baseline approaches for predicting lineup matchup performance. The first one is based on network ranking that operates directly on the matchup network (i.e., without involving any embedding of the network), while the rest two are based on the adjusted plus/minus rating of the players that belong to the lineup, as well as, the lineup itself.

Network Ranking In our prior work we have shown that ranking teams through centrality metrics - and in particular PageRank - of a win-loss network, achieves better matchup prediction accuracy as compared to their win-loss record [13]. The intuition behind the team network ranking is that nodes (teams) with high PageRank have outperformed many more teams or have outperformed *good* teams that themselves have performed many other or *good* teams etc. Therefore, we follow a similar approach using the lineup matchup network and rank lineups based on their PageRank score. The PageRank of $\mathcal{G}$ is given by:

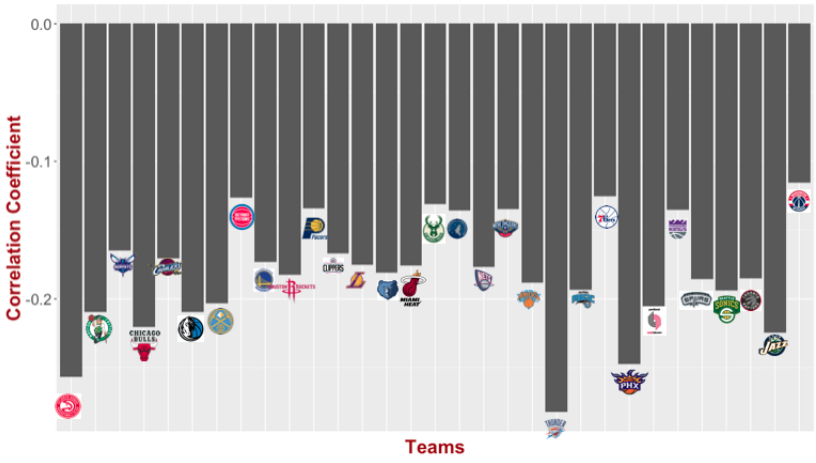

Fig. 2. Lineups with higher overlap in terms of players exhibit smaller distance in the latent embedding space $\mathcal{X}$

$$\boldsymbol{r} = D(D - \alpha A)^{-1}\mathbf{1} \tag{5}$$

where A is the adjacency matrix of $\mathcal{G}$, α is a parameter (a typical value of which is 0.85) and D is a diagonal matrix where $d_{ii} = \max(1, k_{i,out})$, with $k_{i,out}$ being the out-degree of node i. Using the PageRank score differential $\Delta r_{ij} = r_{\lambda_i} - r_{\lambda_j}$ as our independent variable we build a logistic regression model for the probability: $\Pr(\lambda_i \succ \lambda_j | \Delta r_{ij})$.

Player Adjusted plus/minus (PAPM) The APM statistic of a player is a modern NBA statistic - and for many people the best single statistic we currently have - for rating players [15]. It captures the additional points that the player is expected to add with his presence in a lineup consisting of league average players matching up with a lineup with league average players. APM captures the impact of a player beyond pure scoring. For instance, a player might impact the game by performing good screens that lead to open shots, something not captured by current box score statistics. The other benefit of APM is that it controls for the rest of the players in the lineups. More specifically the APM for a player is calculated through a regression model. Let us consider that lineup λ_i has played against λ_j, and has outscored the latter by y points per 48 minutes. y is the dependent variable of the model, while the independent variable is a binary vector $\mathbf{p}$, each element of which represents a player. All elements of $\mathbf{p}$ are 0 except for the players in the lineups. Assuming λ_i is the home lineup[3],

[3] If this information is not available - e.g., because the input data include the aggregate time the lineups matched up over multiple games - without loss of generality we can

$p_n = 1,\ \forall p_n \in \lambda_i$, while for the visiting lineup, $p_n = -1,\ \forall p_n \in \lambda_j$. Then these data are used to train a regression model:

$$y = \mathbf{a}^T \cdot \mathbf{p} \tag{6}$$

where $\mathbf{a}$ is the vector of regression coefficients. Once obtaining this vector, the APM for player p_n is simply a_{p_n}. The rating of lineup λ_i, ρ_{λ_i} is then the average APM of its players:

$$\rho_{\lambda_i} = \frac{a_{p_n}}{5},\ \forall p_n \in \lambda_i \tag{7}$$

Using the lineup rating differential $\Delta\rho_{ij} = \rho_{\lambda_i} - \rho_{\lambda_j}$ as our independent variable we again build a logistic regression model for the probability: $\Pr(\lambda_i \succ \lambda_j | \Delta\rho_{ij})$.

Lineup Adjusted plus/minus (LAPM) The above baseline method assumes that the lineup is simply the sum of its individual parts in a vacuum. However, this is certainly not true in many cases (if not in most/all of the cases). Players can help each other boost their performance, or they might not be in sync and hence, not perform as expected. For example, one should expect that a lineup that includes your best player (e.g., the highest APM) should perform better than one where he is substituted. However, this is not necessarily true. For instance, Skinner [17] used network theory to show that a lineup that does not include the best player of the team, might perform better as compared to lineups including this payer. Thus, simply summing up the individual players' contribution can overestimate or underestimate a lineup's performance. For this reason, we examine another baseline that considers the adjusted plus/minus of the lineups (as opposed to individual players). More specifically, we follow the same approach as with PAPM but our independent variable binary vector now represents lineups rather than individual players. The corresponding regression coefficient represents the adjusted plus/minus of the lineup l_{λ_i}. Using the LAPM differential $\Delta l_{ij} = l_{\lambda_i} - l_{\lambda_j}$ we further build a logistic regression model for the probability: $\Pr(\lambda_i \succ \lambda_j | \Delta l_{ij})$.

2.3 Datasets

In order to evaluate `LinNet` we used lineup data during the 5 NBA seasons between 2007-08 and 2011-12 obtained through basketballvalue.com. This dataset includes information for all the lineup matchups for each of the 5 seasons. In particular, for each pair of lineups (e.g., λ_i, λ_j) that matched up on the court we obtain the following information:

1. Total time of matchup
2. Total point differential
3. Players of λ_i

consider the home lineup to be the one with lower ID number for reference purposes. This is in fact the setting we have in our dataset.

4. Players of λ_j

We would like to note here that these data include aggregate information for matchups between lineups. For example, if lineup λ_A played against λ_B over multiple stints - either during the same game or across different games - the performance over these matchups will be aggregated. The benefit of this approach is that we now have a longer, and potentially more robust (see Section 4), observation period for the matchup between λ_A and λ_B. On the other hand, aggregating information does not allow us to account for home-field advantage. Nevertheless, the latter is typically considered to be 3 points per game (i.e., per 48 minutes) in the NBA [18], which means that during a 5 minute stint there will be an approximately 0.3 points adjustment missed. Hence, we should not expect a big impact on our final results. Furthermore, in our dataset only approximately 10% of the lineup pairs have matched-up over separate stints.

We used these data in order to obtain both the matchup network as well as to calculate the APM for every player in each season. Using these data we build the lineup matchup networks. Table 1 depicts some basic statistics for these networks. Note here that the dataset for the 2011-12 season includes only approximately 75% of that season's games and this is why the network is smaller. All networks have similar size and density and exhibit similar diameter. Furthermore, they all have right-skewed degree distributions. Table 1 also presents the power-law exponent obtained for every network after fitting a power-law distribution, i.e., $P(k) \propto k^{-\gamma}$, where k is the node degree.

Season	Nodes	Edges	Diameter	Power-Law Exponent γ
2007-08	10,380	50,038	15	3.5
2008-09	10,004	48,414	16	2.8
2009-10	9,979	49,258	15	2.6
2010-11	10,605	49,694	18	2.5
2011-12	8,498	35,134	17	2.8

Table 1. Basic network statistics for the lineup matchup networks used.

3 Analysis and Results

We now turn our attention to evaluating LinNet. Our focus is on evaluating the accuracy of LinNet in predicting future lineup matchups, as well as the calibration of the inferred probabilities. For every season, we build LinNet (both the network embedding as well as the Bradley-Terry model) using 80% of the matchups and we evaluate them on the remaining 20% of the matchups. Our evaluation metrics include: (i) prediction accuracy, (ii) Brier score and (iii) the probability calibration curve.

3.1 Prediction Accuracy

Table 2 presents the accuracy of each method predicting the outcome of lineup matchups over all seasons. As we can see LinNet outperforms all the baselines

during all five seasons. Out of the three baselines we evaluated, LAPM performs the best. This further indicates that the performance of a lineup cannot be simply described by the sum of its individual parts; metrics that evaluate each player individually cannot capture well the performance of a lineup.

Season	Page Rank	PAPM	LAPM	`LinNet`
2007-08	52%	55%	59%	67%
2008-09	53%	56%	57%	69%
2009-10	52%	54%	58%	68%
2010-11	54%	55%	59%	68%
2011-12	53%	56%	58%	67%

Table 2. `LinNet` outperforms all three baselines with respect to accuracy. LAPM performs the best among the baselines.

3.2 Probability Calibration

Accuracy figures cannot fully evaluate a probabilistic model as it does not provide any insight on how well-calibrated and accurate the output probabilities are. To evaluate the probability calibration of `LinNet` we rely on the Brier score and the reliability curve.

Brier Score In a probabilistic model, its classification accuracy paints only part of the picture. For example, two models M_1 and M_2 that both predict lineup λ_A will outperform λ_B will exhibit the same accuracy. However, if $\Pr_{M_1}(\lambda_A \succ \lambda_B) = 0.9$ and $\Pr_{M_2}(\lambda_A \succ \lambda_B) = 0.55$, the two models have different probability calibration. The latter can be evaluated by calculating the Brier score [3] of each model, which can essentially be thought of as a cost function. In particular, for the case of binary probabilistic prediction, the Brier score is calculated as:

$$\beta = \frac{1}{N}\sum_{i=1}^{N}(\pi_i - y_i)^2 \tag{8}$$

where N is the number of observations, π_i is the probability assigned to instance i being equal to 1 and y_i is the actual (binary) value of instance i. The Brier score takes values between 0 and 1 and as alluded to above evaluates the calibration of these probabilities, that is, the level of confidence they provide. The lower the value of β the better calibrated the output probabilities are – recall that Brier score is essentially a cost function. Continuing on the example above a 0.9 probability is better calibrated compared to a 0.55 probability (when the ground truth is label 1) and hence, even though M_1 and M_2 have the same accuracy, M_1 is better calibrated (lower Brier score – 0.01 compared to 0.2025). The Brier scores for `LinNet` and the baselines examined are presented in Table 3.

As we can see `LinNet` exhibits a lower Brier score as compared to the baselines. Furthermore, typically the Brier score of a model is compared to a *climatology* model [9]. A climatology model assigns the same probability to every

Season	Page Rank	PAPM	LAPM	LinNet
2007-08	0.23	0.22	0.22	0.19
2008-09	0.23	0.23	0.22	0.19
2009-10	0.23	0.23	0.21	0.19
2010-11	0.23	0.23	0.22	0.19
2011-12	0.23	0.22	0.21	0.18

Table 3. LinNet exhibits better probability calibration compared to the baselines examined.

observation, which is equal to the fraction of positive labels in the whole dataset, i.e., a base rate. Therefore, in our case the climatology model assigns a probability of 0.5 to each observation. As alluded to above we do not have information about home and visiting lineup so our model estimates the probability of the lineup with the smaller ID outperforming the one with the larger ID. Given that the lineup ID has no impact on this probability the climatology model probability is 0.5. The Brier score for this reference model is $\beta_{climatology} = 0.25$, which is of lower quality as compared to LinNet and also slightly worse than our baselines.

Reliability Curve Finally, we evaluate the accuracy of the probability output of LinNet by deriving the probability validation curves. In order to compute the accuracy of the predicted probabilities we would ideally want to have every matchup played several times. If the favorite lineup were given a 75% probability of outperforming the opposing lineup, then if the matchup was played 100 times we would expect the favorite to *win* approximately 75 of them. However, this is clearly not realistic and hence, in order to evaluate the accuracy of the probabilities we will use all the matchups in our dataset. In particular, if the predicted probabilities were accurate, when considering all the matchups where the favorite was predicted to win with a probability of x%, then the favorite should have outperformed the opponent in (approximately) x% of these matchups. Given the continuous nature of the probabilities we quantize them into groups that cover a 5% probability range. Figure 3 presents the predicted win probability for the reference lineup (i.e., the lineup with the smaller ID) on the x-axis, while the y-axis presents how many of these matchups this reference lineup won. Furthermore, the size of the points represents the number of instances in each situation. As we can see the validation curve is very close to the $y = x$ line, which practically means that the predicted probabilities capture fairly well the actual matchup probabilities. In particular, the linear fit has an intercept of 0.1 and a slope of 0.85.

3.3 Dimensionality of LinNet

One of the parameters that we must choose for the network embedding, which forms the core of LinNet is its dimensionality d, i.e., how long is the vector representation of each lineup/node in $\mathcal{G}$. In all of our experiments above we have

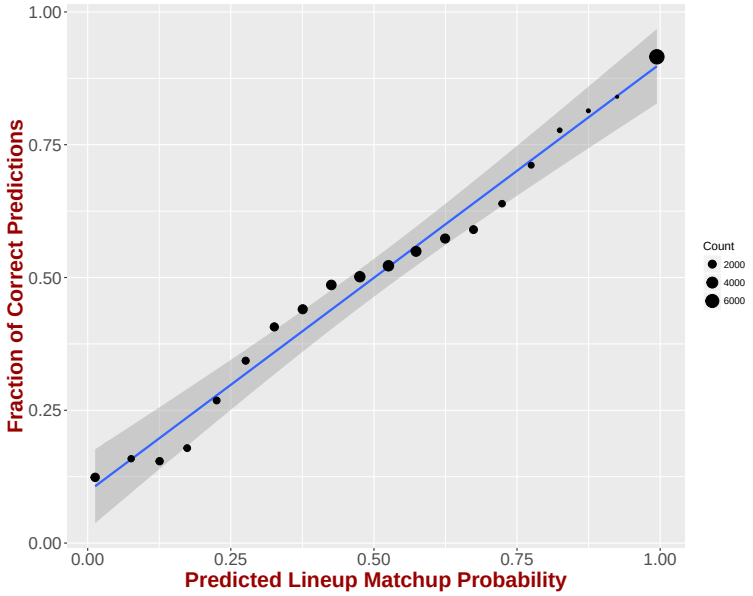

Fig. 3. The `LinNet` probability validation curve is very close to the $y = x$ line, translating to fairly accurate probability estimations (for $d = 128$).

used a dimensionality of $d = 128$. However, we have experimented with different embedding dimensionality values and the accuracy results[4] are presented in Figure 4. As we can see, low dimensionality does not provide any significant benefits with respect to the accuracy of the model over the baselines. Increasing the dimensionality further, improves the model performance. However, for values higher than $d = 128$ we see a plateau in the performance. In fact, we even see a slight decrease for dimensionality greater than 128. Higher dimensions lead to solutions that might not be as robust, since there are many more variables to optimize for the same amount of data. This can also lead to overfitting, which consequently degrades the out-of-sample performance.

3.4 Season Win-Loss Record and Lineup Performance

How well can lineup *ratings* obtained from `LinNet` explain the win-loss record of a team? One should expect that there is a correlation between `LinNet` lineup ratings and the record of a team - which as we will see indeed is the case. However, this correlation should not be expected to be perfect, since it relies also on coaching decisions as well as availability of the lineups (e.g., a lineup can be unavailable due to injuries). In order to examine this we focus on lineups that

[4] The Brier score exhibits similar qualitatively behavior but the differences are much smaller compared to the model accuracy and hence, we omit their presentation.

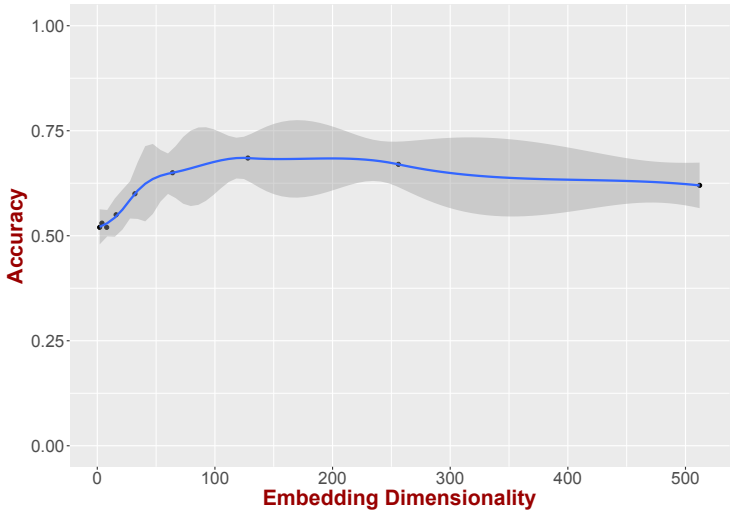

Fig. 4. The choice of $d = 128$ for the embedding dimensionality of LinNet provides a good tradeoff between accuracy and (computational) complexity.

played for a total of more than a game (i.e., 48 minutes) during the season. Let p_{λ_i} be the average probability of lineup λ_i (of team τ) to outperform each of the opponent's lineups. I.e.,

$$p_{\lambda_i} = \frac{\sum_{\lambda_j \in \mathcal{L} \setminus \mathcal{L}_\tau} \Pr(\lambda_i \succ \lambda_j)}{|\mathcal{L} \setminus \mathcal{L}_\tau|} \tag{9}$$

where $\mathcal{L}_\tau$ is the set of all lineups of team τ and $\mathcal{L}$ is the set of all league lineups. Then the LinNet team rating of team τ is:

$$r(\tau) = \frac{\sum\limits_{\lambda_i \in \mathcal{L}_\tau} \gamma_i \cdot p_{\lambda_i}}{\sum\limits_{\lambda_i \in \mathcal{L}_\tau} \gamma_i} \tag{10}$$

where γ_i is the total time lineup λ_i has been on the court over the whole season. Our results are presented in Figure 5. The linear regression fit has a statistically significant slope (p-value < 0.001), which translates to a statistically important relationship. However, as we can see there are outliers in this relationship, such as the 2008-09 Cavaliers and the 2011-12 Nets. The linear relationship explains 27% of the variability at the win-loss records of the teams. This might be either because teams do not choose (due to various reasons) their best lineup to matchup with the opponent, or because the time that a lineup is on court is important for its performance (we discuss this in the following section), something that LinNet currently does not account for (see Section 4). Overall, the

correlation coefficient between the LinNet team rating and the win-loss record is moderate adn equal to 0.53 (p-value < 0.0001).

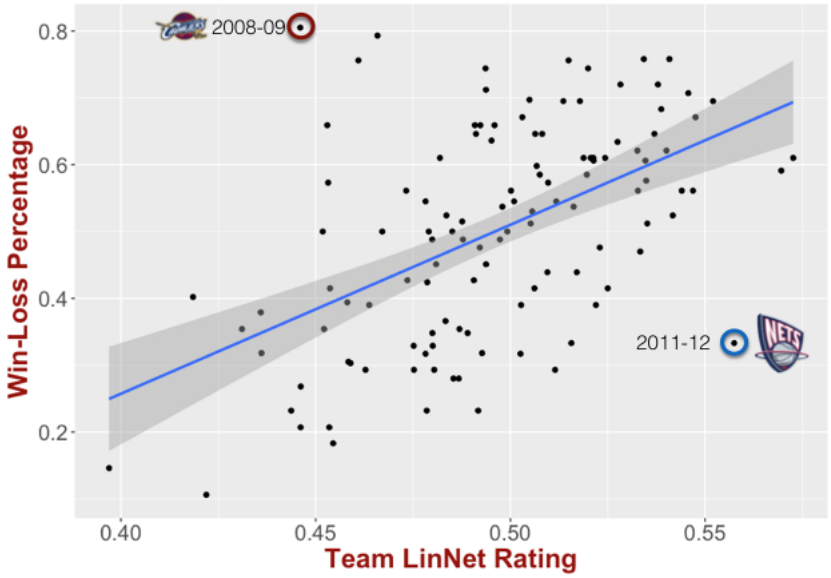

Fig. 5. The team ratings we obtain from LinNet explain 27% of the win-loss variability of teams.

4 Discussion and Conclusions

In this work, we presented LinNet, a network embedding approach for evaluating lineups. Our evaluations indicate that the probability output from LinNet is well calibrated and more accurate than traditional lineup evaluation methods. More importantly, while we have evaluated LinNet using basketball lineup data, the proposed method is sport-agnostic and not specific to basketball, i.e., there are no basketball-related features used. In particular, LinNet can be used to evaluate lineups in other sports as long as they involve frequent substitutions (e.g., hockey, volleyball etc.) and an appropriate performance metric is defined. Furthermore, it can also be used for evaluating and rating teams, as well as, predicting future games. In this case network $\mathcal{G}$ will capture the win-loss relationships between teams rather than lineups.

However, there are still open issues with the design of LinNet. More specifically, a matchup between lineups might last only for a few minutes (or even just a couple of possessions). This creates a reliability issue with any prediction that one tries to perform with similar information. Even though we adjust the performance margin on a per minute basis, it is not clear that a lineup can keep

up its performance over a larger time span. It is a very plausible hypothesis that a lineup has its own skill curve, similar to the players' *skill* curves introduced by Dean Oliver [11]. Other contextual factors can also impact the performance of a lineup (e.g., foul troubles, current score differential etc.) that we have not accounted for. However, note that these issues exist with every lineup evaluation metric and to the best of our knowledge they have not been addressed. In addition, the limited temporal observation that we might have for some stints can lead to unreliable labeling for the Bradley-Terry model. More specifically, if lineup λ_A outperformed lineup λ_B by 1 point during a stint of a total of 90 seconds, is it correct to declare that λ_A outperformed λ_B (for training and evaluating our model)? In fact, as one might have expected there is a significant, moderate, correlation of 0.44 (p-value < 0.01) between the matchup temporal length and the final point margin observed. How can we incorporate this label uncertainty in our model? To answer these questions we plan to explore the concept of *fuzzy classification* as part of our future work, where the category membership function will integrate the temporal dimension. The latter might also require the extension of the model from binary classification to multi-class classification, where we have a third class corresponding to the two lineups being *equally matched*.

Furthermore, currently for lineups that we have not seen before we use as its latent features a weighted average of already seen lineups, weighted based on their similarity in the players' space. Nevertheless, there are other approaches that one might use for this task that could potentially provide even better results. For example, a regression model (similar to the one used for calculating the adjusted plus/minus) can be used to infer the latent features based on the players in the lineup.

Finally, currently LinNet utilizes a generic network embedding framework from the network science literature (i.e., node2vec), with a number of parameters that need to be tuned and optimized[5]. However, optimizing the neighborhood objective that node2vec does might not be the most appropriate objective for evaluating lineups. Thus, a task-specific embedding might perform better than a generic framework. For example, one of the problems in ranking sports teams (and lineups) is the several intransitivity relationships (e.g., lineup λ_A outperforms lineup λ_B, lineup λ_B outperforms lineup λ_C, but lineup λ_C outperformed lineup λ_A). These relationships manifest themselves as triangles in the matchup network. An objective function that incorporates these cycles might be more appropriate. Moreover, modeling the point performance margin between two lineups is also of interest, since in many cases a lineup needs to outscore its opponent more than just one point in order for the team to win or obtain the lead. All these are promising directions for future research on the usage of network science and representation learning for basketball analytics in general, and on evaluating lineups in particular. Despite these open issues, we firmly believe that our current study makes a solid contribution in the problem of evaluating

[5] In the current version parameters p, and q, as well as, the size and number of random walks have not been necessarily optimally chosen.

lineups and a strong case for the use of network science methods and tools in the field of sports analytics in general.

References

1. Nba advanced stats: Lineup efficiency. https://stats.nba.com/lineups/advanced/, accessed: 2018-04-04
2. Agresti, A.: An introduction to categorical data analysis. Wiley series in probability and statistics, Wiley-Interscience, Hoboken (N.J.) (2007)
3. Brier, G.W.: Verification of forecasts expressed in terms of probability. Monthly weather review **78**(1), 1–3 (1950)
4. Fewell, J.H., Armbruster, D., Ingraham, J., Petersen, A., Waters, J.S.: Basketball teams as strategic networks. PloS one **7**(11), e47445 (2012)
5. Gonçalves, B., Coutinho, D., Santos, S., Lago-Penas, C., Jiménez, S., Sampaio, J.: Exploring team passing networks and player movement dynamics in youth association football. PloS one **12**(1), e0171156 (2017)
6. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864. ACM (2016)
7. Le, H.M., Yue, Y., Carr, P., Lucey, P.: Coordinated multi-agent imitation learning. In: International Conference on Machine Learning. pp. 1995–2003 (2017)
8. Lowe, Z.: Lights, cameras, revolution (2013), http://grantland.com/features/the-toronto-raptors-sportvu-cameras-nba-analytical-revolution/
9. Mason, S.J.: On using "climatology" as a reference strategy in the brier and ranked probability skill scores. Monthly Weather Review **132**(7), 1891–1895 (2004)
10. Mehrasa, N., Zhong, Y., Tung, F., Bornn, L., Mori, G.: Deep learning of player trajectory representations for team activity analysis
11. Oliver, D.: Basketball on paper: Rules and tools for performance analysis. Potomac Books (2004)
12. Pelechrinis, K.: Lineup evaluations through in-game win probability models and bayesian adjustment. Technical Report (https://www.pitt.edu/ kpele/TR-SCI-PITT-032018.pdf) (2018)
13. Pelechrinis, K., Papalexakis, E., Faloutsos, C.: Sportsnetrank: Network-based sports team ranking. In: ACM SIGKDD Workshop on Large Scale Sports Analytics (2016)
14. Peña, J., Touchette, H.: A network theory analysis of football strategies. In: In C. Clanet (ed.), Sports Physics: Proc. 2012 Euromech Physics of Sports Conference, p. 517-528, Editions de l'Ecole Polytechnique, Palaiseau, 2013.(ISBN 978-2-7302-1615-9)
15. Rosenbaum, D.: Measuring how nba players help their teams win. Available at: http://www.82games.com/comm30.htm. (Last accessed: 5-6-2018) (2004)
16. Seidl, T., Cherukumudi, A., Hartnett, A., Carr, P., Lucey, P.: Bhostgusters: Real-time interactive play sketching with synthesized nba defenses
17. Skinner, B.: The price of anarchy in basketball. Journal of Quantitative Analysis in Sports **6**(1)
18. Winston, W.L.: Mathletics: How gamblers, managers, and sports enthusiasts use mathematics in baseball, basketball, and football. Princeton University Press (2012)
19. Zhan, E., Zheng, S., Yue, Y., Lucey, P.: Generative multi-agent behavioral cloning. arXiv preprint arXiv:1803.07612 (2018)