MIT SLOAN SPORTS ANALYTICS CONFERENCE
presented by ESPN

Deep Learning of Player Trajectory Representations for Team Activity Analysis

Nazanin Mehrasa*, Yatao Zhong*, Frederick Tung, Luke Bornn, Greg Mori
Simon Fraser University
{nmehrasa, yataoz, ftung, lbornn}@sfu.ca, mori@cs.sfu.ca

Abstract

Team sports such as ice hockey and basketball involve complex player interactions. Modeling how players interact with each other presents a great challenge to researchers in the field of sports analysis. The most common source of data available for this type of analysis is player trajectory tracking data, which encode vital information about the motion, action, and intention of players. At an individual level, each player exhibits a characteristic trajectory style that can distinguish him from other players. At a team level, a set of player trajectories forms unique dynamics that differentiate the team from others. We believe both players and teams possess their own particular spatio-temporal patterns hidden in the trajectory data and we propose a generic deep learning model that learns powerful representations from player trajectories. We show the effectiveness of our approach on event recognition and team classification.

1 Introduction

Learning representative features from data is a fundamental problem in sports analysis. Visual data (e.g., images and videos) and tracking data (e.g., player trajectories) are two common types of data sources for analyzing sports games. Computer vision researchers in the past decades have been making substantial progress in developing feature representations and algorithms for understanding human activities, with many benchmarks set up for recognizing sports actions [3, 7, 8]. Millions of pixels in videos provide rich data, but it is necessary to develop accurate and effective algorithms for turning these pixels into useful information regarding the players' actions. Advances in automated player detection and tracking algorithms have made it possible to reliably extract trajectories of players from either purpose-built camera systems or broadcast video feeds.

In this paper we show how such player trajectory data can be used in novel machine learning techniques to obtain a variety of information regarding team and player behaviours. These player trajectories, time series of (x, y) coordinates of player movement, provide a simple yet effective way to encode the spatio-temporal pattern of player motion. With appropriate arrangement of a group of player trajectories, it can even represent the unique underlying motion pattern of a team.

Player tracking data have been extensively studied by researchers in sports analysis. Many of these works study player trajectories from a data mining and knowledge discovery perspective by deploying various statistical models. However, there are few works on learning representative features from player trajectories, which could be useful in many potential applications such as pattern recognition, prediction, clustering, and retrieval. To this end, we propose discriminative feature representation learning using deep convolutional neural networks.

*equal contribution

Existing literature on trajectory learning include [1, 5, 6, 11, 12]. Wei et al. [12] handcrafted a dictionary representation of player movement and game context to predict shot outcome in tennis. [1] proposed a team formation descriptor plus game statistics for identifying team styles. Most similar to our work is [11], where the authors converted trajectory data to an image representation and applied a 2D convolutional neural network for classifying offensive plays. Miller and Bornn [6] analyze NBA team strategies based on variants of probabilistic topic modeling that capture the structure of sets of player trajectories. Le et al. [5] utilize deep imitation learning to generate alternative strategies for defensive teams.

Convolution filter learning is a standard processing paradigm in the machine learning community; with 2D convolution successfully employed in images and 3D convolution in videos [2, 4, 9, 10]. We observe that player tracking data come in the form of one dimensional signals, which is an ideal case to deploy 1D convolutions.

In this paper, we demonstrate that 1D convolutions can learn discriminative features from player and team trajectories. With the learned feature representations, our model can automatically recognize events, identify players, and classify teams. We show that, on an ice hockey dataset, our model with only trajectories as input outperforms C3D [10], a deep neural network that takes videos as input, on the task of event recognition. Our model achieves even better performance when used in combination with videos. We also demonstrate, on a basketball dataset, how our model excels at team classification using only player trajectories.

2 Approach

We conduct three sets of experiments: event recognition, team classification, and star player impact on team classification. Fig. 1 shows the structure of our proposed model. We develop a general-purpose method for representing player trajectories that is used in all three tasks. They all share the same 1D convolutional network that learns to represent trajectories. In this section, we start by describing our 1D convolutional network. Next, we show how to utilize it for modeling the set of trajectories consisting of all the players in a team.

2.1 Player trajectory features: 1D convolutional network

We propose a direct way of interpreting a trajectory. Recall that a person trajectory is essentially a continuous signal. A 2D trajectory in world coordinates (player position in court / rink coordinates) has two separate continuous signals, one for the x series and one for y series. We can split the input $[(x_1, y_1), (x_2, y_2), \cdots, (x_T, y_T)]$ into two sequences $[x_1, x_2, \cdots, x_T]$ and $[y_1, y_2, \cdots, y_T]$, each being a 1D continuous signal. In our approach we treat these two sequences as two channels. We build a convolutional neural network on top of these inputs, with 1D convolution operating on each input. By stacking layers of 1D convolution, we can learn combinations of x and y movements that are indicative of particular motion patterns.

In detail, let $X \in \mathbb{R}^{N \times T}$ denote the input, $F \in \mathbb{R}^{N \times W \times M}$ denote the filters in a convolutional layer and $O \in \mathbb{R}^{M \times T}$ denote the output, where N is the number of input channels, T is the length of input sequence, W is the filter size and M is the number of filters. To model the behaviour of a convolutional layer[1], we do the basic operation as follows:

$$O_{k,t} = \sigma\left(\sum_{i=1}^{N}\sum_{j=1}^{W} X_{i,t+j-1} F_{i,j,k}\right). \tag{1}$$

In the above formula, $\sigma(\cdot)$ can be any activation function. In our case, we choose $\sigma(x) = max(0, x)$

[1] We choose a step size of 1 when doing convolution and pad zeros to the input if the domain of a filter goes outside of the input.

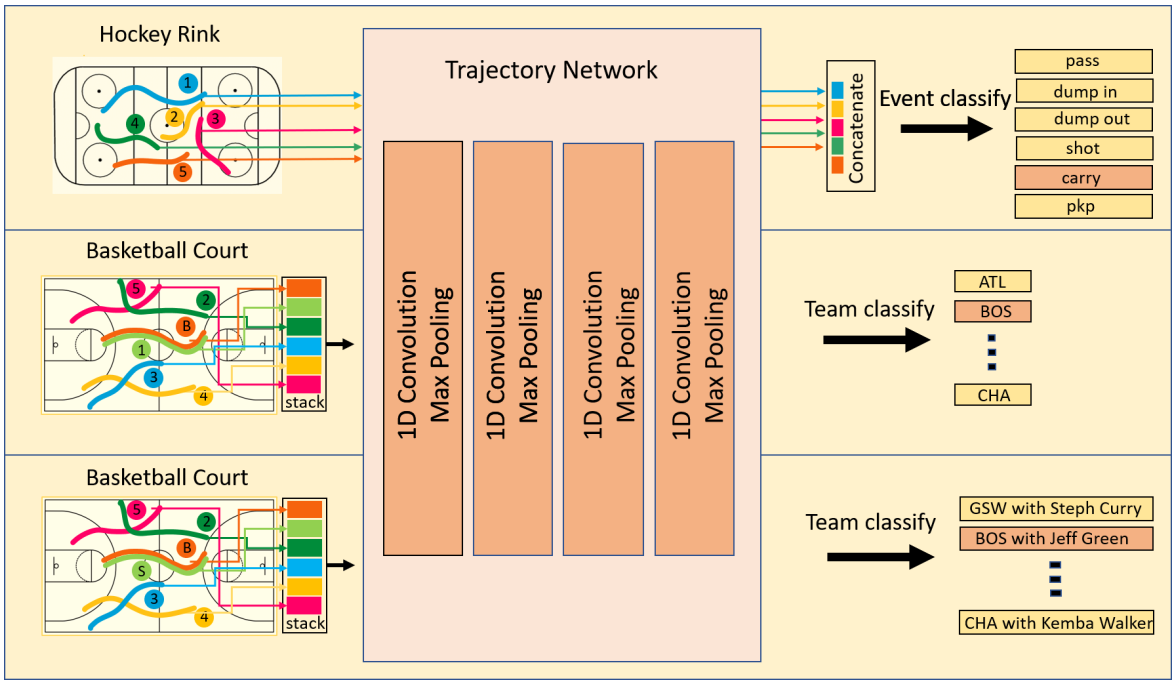


Figure 1: Our proposed trajectory network.

for all layers in our network. Each convolutional layer is followed by a max pooling layer to make the model shift-invariant and help reduce the dimension of the output.

Let $Z \in \mathbb{R}^{M \times \lceil \frac{T}{S} \rceil}$ be the output of max pooling, where S is the step size in the pooling operation, then we have

$$Z_{k,t} = \max_{1 \le j \le S} O_{k,\,(t-1)\cdot S+j}. \tag{2}$$

To build a network with stacked convolutional and max pooling layers, we use the output Z^{l-1} at layer $l-1$ as the input X^l at layer l:

$$X^l = Z^{l-1}. \tag{3}$$

We repeat the process described in Eq. 1 and Eq. 2 for a number of layers. To obtain the final feature representation, we flatten the output of the last layer.

2.2 Team trajectory representations: permutation invariant sorting

To build features from a set of player trajectories, the most the intuitive way is to (i) concatenate features extracted from each of the player trajectories or (ii) extract features directly from concatenated player trajectories. Either way, we need to do concatenation. However, when we do concatenation, we implicitly enforce an order among this set of players. Arbitrarily enforcing such order is problematic. To resolve this issue, we have to renumber the players.

We propose a permutation invariant sorting scheme based on the distance from a candidate player to the "anchor". An "anchor" can be the ball or a key player defined based on task-specific criteria. The trajectory of the "anchor" is always placed in the first position. Other players are numbered according to their distances to the "anchor". The closest player is placed next to the "anchor" while the farthest player is appended to the end. In the experiments, we use variants of the above concatenation and sorting procedure, which we describe in detail below.

Event recognition in hockey: We annotate the player who is carrying the puck and we select this player as the "anchor". Other players are numbered according to this key player. Player trajectories

are first fed into the same network for feature extraction, then all resulting features are concatenated according to this order.

Team classification in basketball: The ball is selected as the "anchor". Players are numbered according to the ball. We first stack the ball and the 5 offensive players into an ordered list of trajectories, then we feed them all into the network to extract features.

3 Datasets

We conduct experiments on two datasets. The first incorporates visual and trajectory features: player positions and appearances obtained from broadcast video footage of NHL hockey games. The second includes trajectory features only: player tracks extracted from an external tracking system recording player positions in NBA basketball games

3.1 The SPORTLOGiQ NHL Dataset

The SPORTLOGiQ NHL dataset has both video and trajectory data. Unlike the NBA dataset where player trajectories are obtained from a multi-camera system, the player positions in the SPORTLOGiQ NHL dataset are estimated using a homography, which maps a pixel in image coordinates to a point in world coordinates. SPORTLOGiQ Inc. utilizes state of the art algorithms to automatically detect and track players in raw broadcast videos. If we have the bottom-mid point of a player bounding box, we can map this point to world coordinates with a homography matrix, hence acquiring the player position. Similarly, the SPORTLOGiQ NHL dataset also has detailed event annotation for each frame, each event being categorized into a super class and a fine-grained class. In our experiment, we use 8 games with 6 classes: pass, dump out, dump in, shot, carry, and puck protection. Fig. 2 shows the fraction of each event in the 8-game dataset. A reader might notice that this is a highly unbalanced dataset in terms of events. We will describe how we handle such imbalance in Sec. 4.1.

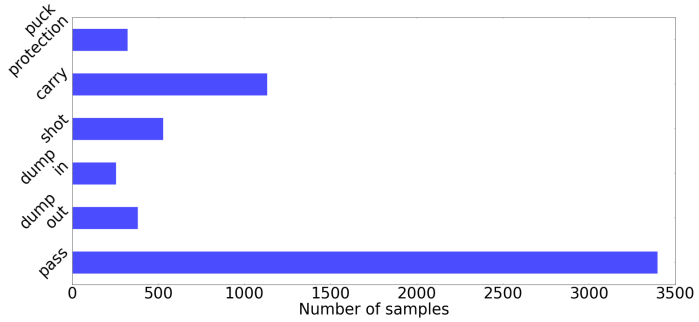


Figure 2: Number of samples per event in the hockey dataset.

Data Preprocessing: In a hockey game, typically there are 4 on-ice officials and 12 players (6 on each team). Thus, there can be at most 16 persons on the rink at the same time. In the following we do not make any distinction between officials and players and we use "player" to refer to all people on the rink. Because the dataset is created from NHL broadcast videos where not all players are visible in each frame, we need to set a threshold N_p so that our model can handle a fixed number of players. If the number of players available in a frame is less than N_p, we pad with zeros the part where players are unavailable. Each training sample consists of data from N_p players. The data of each player includes a T-frame video clip (cropped from raw video using bounding boxes) and the corresponding T-frame trajectory estimated from this video clip. Note that our model supports variable-length input. If in some frames a player is not available, we set the data in these frames to zeros. In our experiments, N_p is set to 5 and video frame size is set to 96×96. We set T to 16 by first locating the center frame where an event

happens and then cropping 7 frames before the center frame plus 8 frames after it. If the center frame of a certain event happens to be close to that of another event within 15 frames, we drop this sample.

3.2 The STATS SportVU NBA Dataset

The STATS SportVU data consist of positions of players and the ball in 2D world coordinates captured by a six-camera system at a frame rate of $25Hz$. Each frame has complete annotations of the events happening in this frame, such as dribble, possession, shot, pass and rebound. The dataset we use has around a thousand games during the 2013–2014 NBA season with around 10^6 frames in each game. We will use this dataset for team classification – determine the identity of a team from player trajectories.

Data Preprocessing: We extract 137176 possessions from the dataset for experiments. Each possession starts with an offensive team having possession of the ball and ends with a shot. We fix possession length to 200 frames. If a possession is longer than 200 frames, we crop it starting from the last frame and count the number of frames backward until it reaches 200. If a possession is shorter than 200 frames, we pad zeros to it. Originally there are 25 frames per second, but we sample only half of the frames in a second, so the sampled 200 frames actually represent a 16 second [2] long sequence.

4 Experiments

4.1 Hockey event recognition

The events used are pass, dump out, dump in, shot, carry and puck protection. The goal is to predict the event label given the short video clips and trajectories of 5 players on the rink. The number of samples of each event in the dataset are shown in Fig. 2. It is obvious that this dataset is highly unbalanced with the pass event taking up half of the dataset. To resolve this problem, we minimize a weighted cross-entropy loss function during training. The weighting for each class is in inverse proportion to its frequency in the dataset, which is 0.07, 0.6, 1, 0.4, 0.2 and 0.7 for pass, dump out, dump in, shot, carry and puck protection respectively.

We use mean average precision (mAP) as the metric. The proposed model is implemented with a stack of 4 convolutional layers and we compare it with C3D, a deep learning model which applies 3D convolutions to videos for action recognition. The result is shown in Table 1. It is worth noting that the proposed 1D convolutional network with only player trajectories as input can beat the C3D model that takes videos as input. This result is counter-intuitive because millions of video pixels definitely provide a lot richer information than a short series of (x, y) coordinates. Our conjecture is that, in ice hockey games, events are firmly associated with spatial locations. As is shown in Fig. 3, different events tend to have different spatial distributions over the rink. For example, carry happens near the three lines in the middle; dump in happens within the neutral zone; dump out mostly happens around the corner and boundary. This strong spatial correlation is more explicit in player tracking data than videos, therefore easier to learn by directly observing player trajectories. This result explains the importance of trajectory data for analyzing player behaviours.

We proceed by constructing a more powerful model by concatenating the output features of 1D convolutional network and C3D and training this composite model end-to-end. When used in combination with videos, the proposed model achieve even higher performance, as is shown in Table 1.

4.2 NBA team classification

In this set of experiments, we classify 30 NBA teams with a stack of 5 convolutional layers. We measure the performance according to the following metrics: accuracy and hit-at-k accuracy[3], both of which are

[2] $16s = \frac{200frames \times 2}{25fps}$

[3] Hit-at-k accuracy means if any one of the top-k predictions equals the ground truth label, we claim it as being correctly classified.

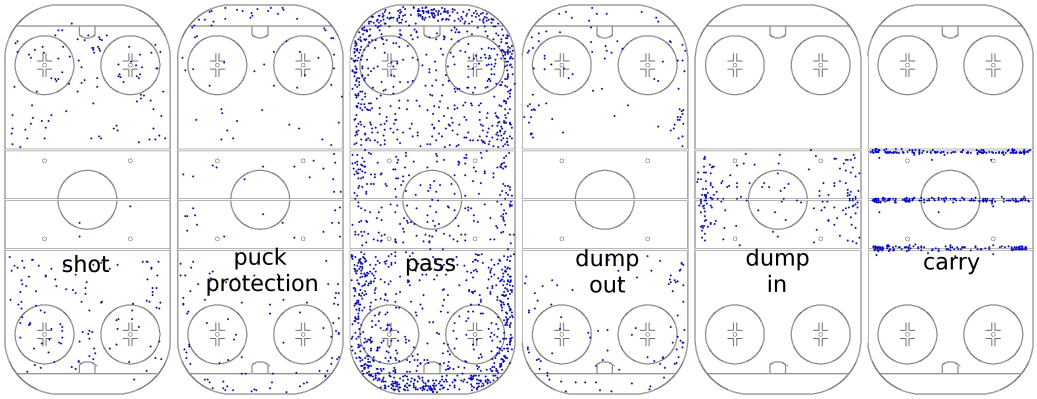


Figure 3: Visualization of locations where events happen. Samples are drawn from the test set.

	C3D	1D conv	C3D+1D conv
pass	77.30%	77.73%	79.15%
dump out	10.17%	22.30%	23.27%
dump in	10.25%	39.39%	37.29%
shot	34.17%	42.42%	50.86%
carry	86.37%	77.21%	86.21%
puck protection	11.83%	9.87%	8.43%
mAP	38.35%	44.89%	47.54%

Table 1: Average precision for each event.

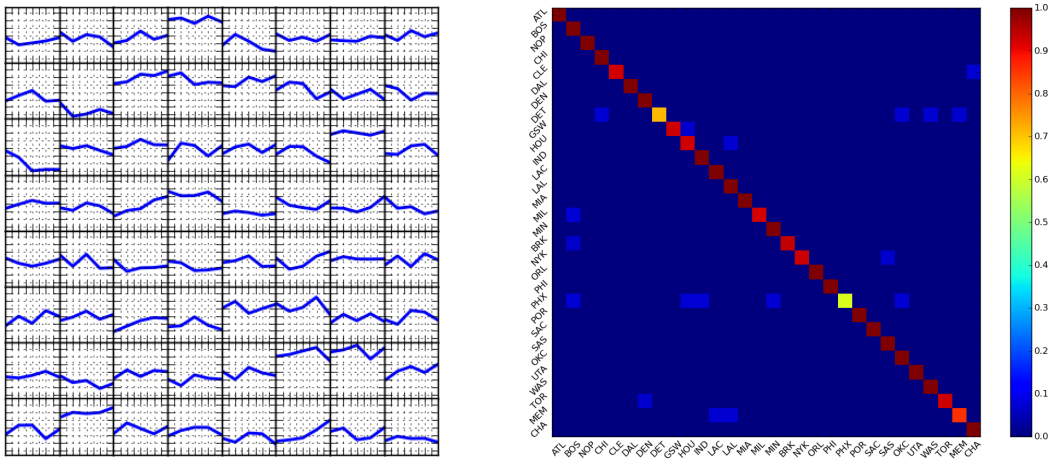
calculated over possessions. However, a single trajectory series can hardly display the full underlying pattern a team might possess. To resolve this issue, we propose to use all possessions in a game and classify the game as a whole using majority voting. For example, if most possessions in a game are predicted as the Golden State Warriors, then the model predicts this game to be a Golden State Warriors game. Table 2 shows the result. Our experiments demonstrate that the per-possession accuracy can be largely improved when aggregated to game level and our best accuracy is surprisingly high – 95.91% – compared to the chance performance of $\frac{1}{30} = 3.33\%$.

Fig. 4b shows the confusion matrix obtained using the model aggregated to game level. For most teams, our model can correctly predict the identity. The worst case is the Phoenix Suns (PHX in Fig. 4b), who the model correctly classifies with 65% correctly. While the lowest of any team, this is still a significant improvement over random chance.

To see what kind of patterns the model learns over the time dimension, we visualize a small fraction of the filters in the first convolutional layer. In Fig. 4a, we show 64 filters learned from the input sequence of x coordinates of the ball. They appear to be shaped in various forms. Some of them are similar, so there could be redundancy in these filters. These temporal patterns are the building blocks that form discriminative representations to distinguish teams.

acc	hit@2	hit@3	game acc
24.78%	35.61%	42.95%	95.91%

Table 2: Team classification result.



(a) (b)

Figure 4: (a) Visualization of the filters in the first convolutional layer. (b) Confusion matrix based on game-wise classification.

4.3 Impact of star players

Are the Golden State Warriors easier to recognize when Steph Curry is leading the play? In this section, we examine the impact of the presence of a "star player" on our model's ability to recognize NBA teams.

We re-use the same network structure as in the task of team classification, except that the star player is anchored in the second position (the ball remains anchored in the first position). We define a team's "star player" as the player with the most points during the 2013–2014 season. For training and evaluation, we consider only possessions in which the star player is on the court. Our model is trained to perform 30-way team classification, conditioned on the star player being on the court.

The trained model obtains 42.45% per-possession accuracy on the 30-way classification task, which is significantly better than the 24.78% result in Table 1. Intuitively, this result tells us that we get nearly twice the classification rate when we remove possessions without the team's leading player by points. This indicates that there is a much more concise team identity in its starters relative to its role players coming off the bench.

5 Conclusion

In this paper we have proposed to use 1D convolutions for learning discriminative feature representations from player tracking data while also resolving the permutation problem inherent in player tracking data. To show the generality of our approach, we conducted extensive experiments on challenging team sports such as ice hockey and basketball, which involve complex player interactions. Our method obtained surprisingly good results on both datasets and various tasks, demonstrating our approach is generic and effective for capturing critical features of player tracking data.

References

[1] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, Sridha Sridharan, and Iain Matthews. Identifying team style in soccer using formations learned from spatiotemporal tracking data. In Data Mining Workshop (ICDMW), 2014 IEEE International Conference on, pages 9–14. IEEE, 2014.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[3] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In Computer Vision and Pattern Recognition (CVPR), pages 1725–1732, 2014.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (NIPS), 2012.

[5] Hoang M. Le, Peter Carr, Yisong Yue, and Patrick Lucey. Data-driven ghosting using deep imitation learning. In MIT Sloan Sports Analytics Conference, 2017.

[6] Andrew C. Miller and Luke Bornn. Possession sketches: Mapping nba strategies. In MIT Sloan Sports Analytics Conference, 2017.

[7] S. M. Safdarnejad, X. Liu, L. Udpa, B. Andrus, J. Wood, and D. Craven. Sports videos in the wild (svw): A video dataset for sports analysis. In 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), volume 1, pages 1–7, May 2015. doi: 10.1109/FG.2015.7163105.

[8] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402, 2012.

[9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[10] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In International Conference on Computer Vision (ICCV), pages 4489–4497. IEEE, 2015.

[11] Kuan-Chieh Wang and Richard Zemel. Classifying nba offensive plays using neural networks. In MIT SLOAN Sports Analytics Conference, 2016.

[12] Xinyu Wei, Patrick Lucey, Stuart Morgan, Machar Reid, and Sridha Sridharan. "The thin edge of the wedge": Accurately predicting shot outcomes in tennis using style and context priors. In MIT SLOAN Sports Analytics Conference, 2016.