MIT SLOAN
SPORTS ANALYTICS CONFERENCE
MARCH 3 - 4, 2017 HYNES CONVENTION CENTER

Possession Sketches: Mapping NBA Strategies

Paper Track
1624

Abstract

We present *Possession Sketches*, a new machine learning method for organizing and exploring a database of basketball player-tracks. Our method organizes basketball possessions by offensive structure. We first develop a model for populating a dictionary of short, repeated, and spatially registered "actions". Each action corresponds to an interpretable type of player movement. We examine statistical patterns in these actions, and show how they can be used to describe individual player behavior. Leveraging this "vocabulary" of actions, we develop a hierarchical model to characterize interactions between players. Our approach draws on the topic-modeling literature, extending Latent Dirichlet Allocation (LDA) through a novel representation of player movement data which uses techniques common in animation and video game design. We show that this model is able to group together possessions with similar offensive structure, allowing efficient search and exploration of the entire database of player-tracking data. We show that our model finds repeated offensive structure in teams (e.g. strategy), providing a much more sophisticated, yet interpretable lens into basketball player-tracking data.

1 Introduction

Player-tracking data present a unique challenge for basketball analytics. Some believe that there is a windfall of quantitative insight available in these data, hidden in spatiotemporal patterns that coaches and analysts typically process with human intuition. While there has been work toward quantifying player ability [6], possession value [4, 3], and play classification based on small set of labeled plays [13], methods for *automatically* organizing, summarizing, and interpreting basketball possessions have not delivered.

As an example, consider the following use case for defensive scouting: an analyst is tasked with finding all possessions in which James Harden drives to the basket and passes the ball to a teammate for a right corner three-point attempt. Ad hoc engineering solutions for this scenario are easy to imagine: first sub-select Rockets possessions with a right corner three-point attempt and then look for passes from Harden that originate in the paint. However, adding search criteria quickly renders this ad hoc solution intractable: find sequences where Harden uses a high screen before driving to the basket and then passes to the corner for a three-point attempt; find sequences where Harden uses a high screen, drives to the basket, passes to the corner and that teammate drives to the basket; find sequences where *any rocket* uses a high screen, drives to the basket, etc. The landscape of relevant basketball scenarios is far too vast for ad hoc search solutions.

Furthermore, this type of sequential query is only one approach to gaining insight from player-tracking data. We can imagine starting a research project by simply asking — what leads to a corner three? What sort of patterns are employed by different offenses in order to get an open three-point attempt? What sorts of actions do specific players tend to do in order to generate an open three-point attempt? Existing methodology falls short of extracting this information from player-tracking data.

In this work, we bridge this gap by formulating a novel machine learning method to describe an entire database of player-tracks. Our method uncovers characteristic patterns of offense in a way that is searchable and interpretable. We first describe individual player's actions by building a data-driven

dictionary of *action templates* derived from a novel statistical model. We then construct a model of possessions that describes patterns in these *action templates* — common co-occurrences that create a signature of offensive strategy. For each play, this yields a *possession sketch*, a concise summary of the offense's actions in a basketball possession. We show that this model structure at multiple levels — in dynamic actions taken by individual players, as well as collective actions present in each possession.

Importantly, we construct our model out of interpretable pieces — each *action template* can be interpreted as a type of on- or off-ball cut. Further, pairs of actions are also interpretable — some clearly correspond to on- and off-ball screens, others correspond drives and passes to various wings. Our use of probabilistic graphical models on an interpretable representation of the data allows for easier-to-understand model output and inferences than recent deep learning approaches [13].

In the following section we describe the components of our method that generate *action templates* and *possession sketches*. After describing our method, we explore the structure it reveals by looking at three of the different organizational tools it makes possible:

- *team possession maps*: low-dimensional visualization of all of the offensive possessions of a team — exploring this map reveals different set calls used by a team.
- *shot possession maps*: low-dimensional visualization of possessions that led to a particular type of shot — we examine the different types of actions that lead to corner threes.
- *possession basis*: common and repeated actions discovered by the model — this establishes the types of player interaction that make up the "vocabulary" of a basketball possession.

By integrating machine learning methods, statistics, and visualization, this work shows that we can organize and systematically explore NBA possessions, allowing us to drive useful basketball intelligence from the NBA's vast and growing store of player-tracking data.

2 Methods

This section details the machine learning model we construct to recognize patterns at two resolutions: spatiotemporal patterns in individual player trajectories (*action templates*), and co-occurrence of actions in each possession (*possession sketches*).

Before we go into further detail, the overall procedure behind our method can be broadly decomposed into the following steps

- *Segmentation*: We cut possession-length (e.g. 5-24 second) player trajectories into shorter, more manageable segments (e.g. .6-8 second) based on moments of sustained low-velocity.
- *Learning action templates*: We formulate a novel statistical clustering algorithm to learn which action is represented by each short segment.
- *Possession modeling*: We represent each possession as a "bag" of pair-actions, and fit a possession-level hierarchical model inspired by the document modeling and natural language processing literature.

The following subsections describe the process of applying the above steps to a large data set of basketball player-tracks.

2.1 Data and preprocessing

We analyze a database of player-tracks from the 2014-2015 season of the NBA. The data are organized into over $N = 190{,}000$ possessions (and possessions into quarters and games). For each possession (indexed by n), we model the trajectories of players on offense. For each player (indexed by j) in possession n, we cut their trajectory (denoted $\boldsymbol{x}_j^{(n)}$) into short segments at locations of sustained low-velocity. To do this, we first detect moments of low velocity by inspecting the smoothed first difference of the

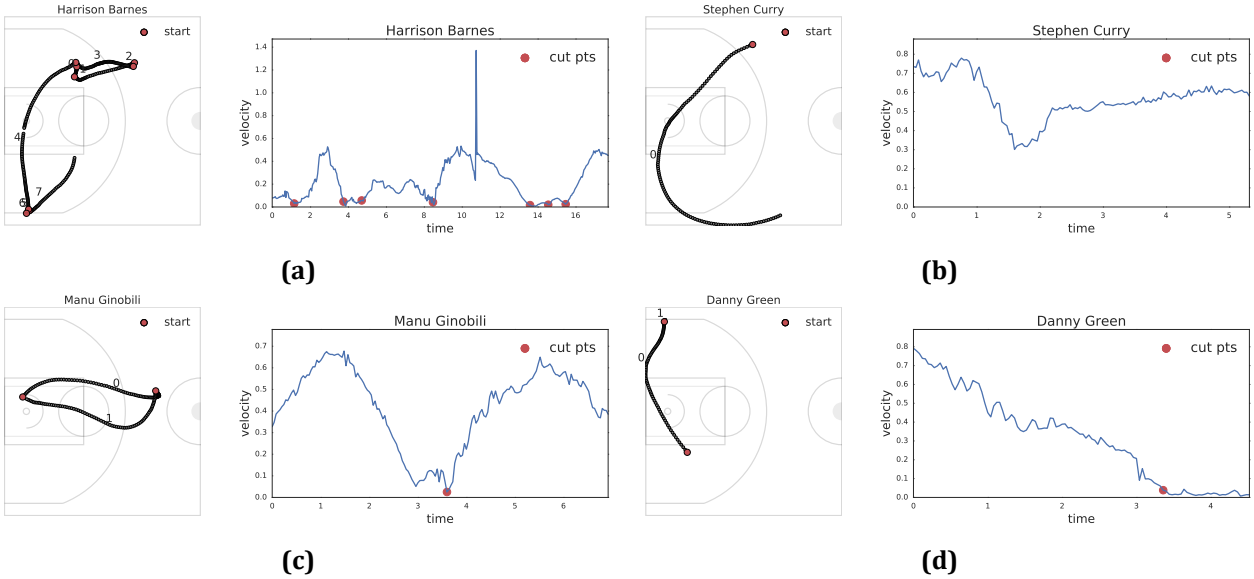


Figure 1: Examples of trajectory segments resulting from the "sustained-low-velocity-moment" finding algorithm. In each example, the left plot depicts the spatial trajectory with cut points denoted by the red dots (with the order of the cuts labeled). The right plot depicts the approximate magnitude of the velocity at each moment during the possession, with the corresponding cut points.

trajectory. At sustained moments of low velocity ($> .25$ seconds below a threshold of $.1$ feet per second), we cut the possession, resulting in a collection of shorter segments. Figure 1 depicts four example trajectories, cut into various number of segments.

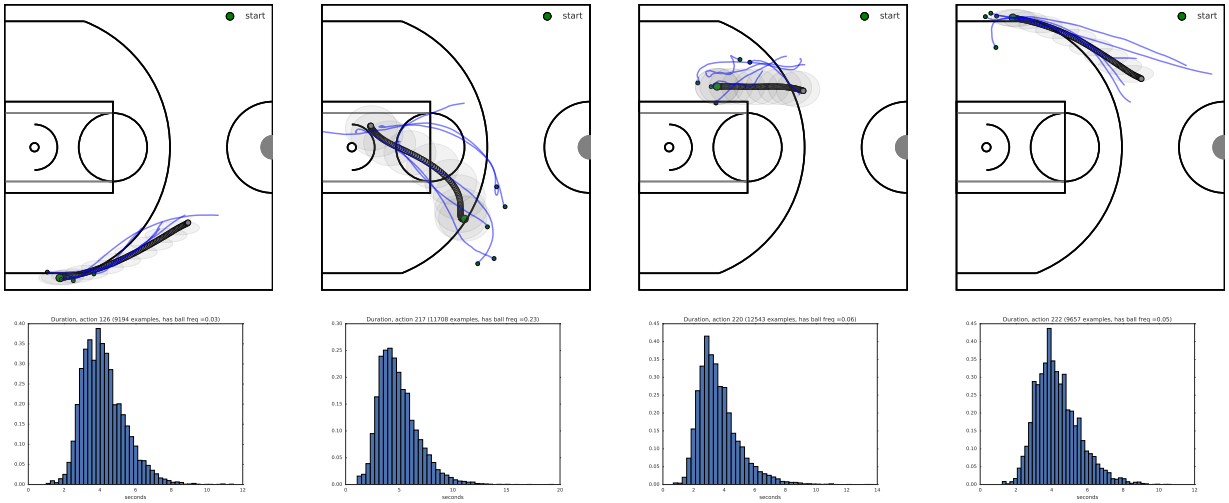
We refer to these shorter segments as $\boldsymbol{x}_{j_1}^{(n)}, \ldots, \boldsymbol{x}_{j_S}^{(n)}$, where it is understood that S varies from possession trajectory to trajectory. The resulting short segments are on average 2.25 seconds (the interior 95 percentiles ranges from 0.6 to 7.96 seconds). Applying this preprocessing step to the full 2014-2015 regular season creates a data set of roughly 4.5 million segment observations.

2.2 Action Templates: Segment Clustering

Our method assumes that each short trajectory segment represents some discrete *action*, and each player performs a series of actions throughout the course of a possession. For instance, a player might (i) make a cut along the baseline and then (ii) camp out in the corner. Alternatively, a player can (i) make a cut along the 3-point line, (ii) stand at the break, and then (iii) cut toward the basket. In order to decompose a player's trajectory into a set of actions, we must first infer a meaningful set of actions that all players share. We use a data-driven approach to infer this set of actions, each action's structure, and the action label for each trajectory segment.

To accomplish this, we construct a probabilistic clustering algorithm tailored for functional data (i.e. continuous trajectories). Our model posits that each trajectory segment represents one of V discrete actions, where each action is characterized by a *template*. Each template can be thought of as a cluster center in a clustering model — each trajectory segment is centered around a cluster with some deviation. We specify each template as a *Bezier curve* – a tool commonly used to model movement in the computer graphics community – which specifies a function $B(t)$ that maps time to a two-dimensional point, $B : [0, 1] \mapsto \mathbb{R}^2$. This maps out a dynamic curve through space, which describes the movement of each action. See the appendix for technical details.

Figure 2 depicts a sampling of learned templates resulting from fitting a mixture of $V = 250$ Bezier curves to the processed trajectory segments. The result of the model fit allows us to succinctly represent



(a) Action 126 **(b)** Action 217 **(c)** Action 220 **(d)** Action 222

Figure 2: A sampling of *action templates*. Our method automatically builds a taxonomy of commonly repeated movements shared among all players (i.e. *actions*). In each column, the top plot depicts the spatial trajectory for a single action template. The light blue lines are real segment trajectories that fall in that cluster. Below each action plot is a histogram of segment lengths (in seconds) for all segments that fall into that cluster — some actions are shorter or longer (on average) than others. For a more dynamic picture of an action template, please view this animated figure: https://youtu.be/-a6_Ot6etmk

each trajectory segment as a single discrete integer, $v = 1, \ldots, 250$. We view these actions as a kind of *vocabulary* — each possession combines words in the vocabulary to describe structured interactions that characterize the possession. Following this thread, we turn to statistical methods originally devised for modeling documents, and adapt them to basketball sequences.

2.3 Possession Model

Offensive possessions are highly structured. When James Harden drives toward the basket, drawing defender attention, his teammates are not distributed randomly on the floor — it is likely at that least one teammate is in the corner waiting for a pass; it is likely that other teammates vacate the paint, and begin jockeying for a rebounding position. The structure of an offensive possession can be viewed as structure in the actions that each player performs throughout the possession. Which actions tend to simultaneously co-occur? Which actions tend to precede or follow other actions? Our possession model seeks to answer these questions by first observing that these actions are a lot like words. Words are interwoven sequentially to express a coherent idea; player actions are interwoven sequentially to implement a coherent strategy. We run with this analogy by adapting *topic models* [1] to describe sequences of actions in basketball possessions.

We use Latent Dirichlet Allocation (LDA) [2], a topic model for unsupervised structure discovery in a corpus of text documents. LDA is a latent factor model, similar to factor analysis or principal components analysis. In document modeling, LDA describes each document as a mixture *topics*, where each topic is a distribution over the entire vocabulary of words. As a concrete example, LDA applied to a corpus of *Science* articles finds topics corresponding to *cancer* (e.g. probable words are “tumor”, ”cell”, ”cancer”, etc.), *neuroscience* (“synaptic”, “neurons”, “hippocampal”, etc.), among many others (see [8, 1]).

We use LDA to describe each possession as a mixture of *strategies*, where each strategy is a distribution over co-occurring actions that are frequently observed in the data. LDA requires that we represent

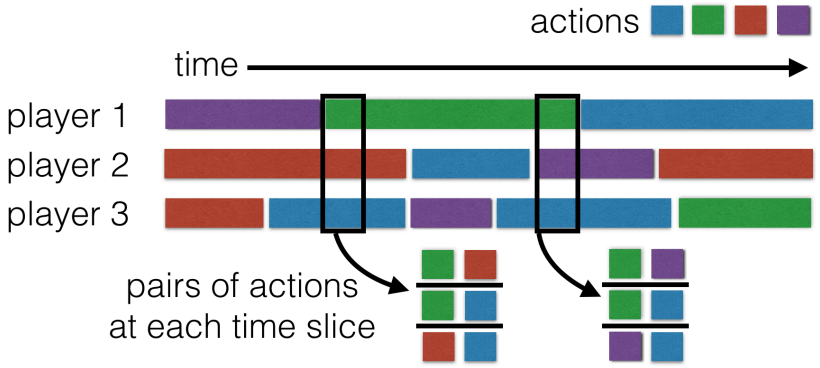


Figure 3: The "bag of words" construction of each possession. Each "word" represents two actions that occur simultaneously throughout the course of the possession. In the toy example depicted, we have three players, each performing a sequence of actions (corresponding to the four colors). At each moment in time, we enumerate all unique pairs of actions. We represent the entire possession as a bag of these pair-action counts.

each possession as a vector of "word counts". To do this, we need to first establish a vocabulary.[1] Our first approach was to simply count the number of $v = 1, \ldots, V$ actions that occur in each possession. This approach is appealing in its simplicity, and does reveal interesting structure. However, this representation completely ignores temporal structure in the possession.

In this work, we use a vocabulary of *pair-actions*, where each "word" in the vocabulary is a unique pair of of the V actions, (v_i, v_j) for $v_i, v_j \in \{1, \ldots, V\}$ and $v_i \neq v_j$. We then represent each possession as a "bag of simultaneous pair-actions", mapping the "bag of words" concept from topic models to basketball actions. For each possession, we simply count the number of times each unique pair of actions ($v = 1, \ldots, V$) occurred simultaneously. We string these counts into a single vector, which represent possession n

$$Y_{n,d} = \text{\# times action action pair } d = (v_1, v_2) \text{ appears in possession } n. \tag{1}$$

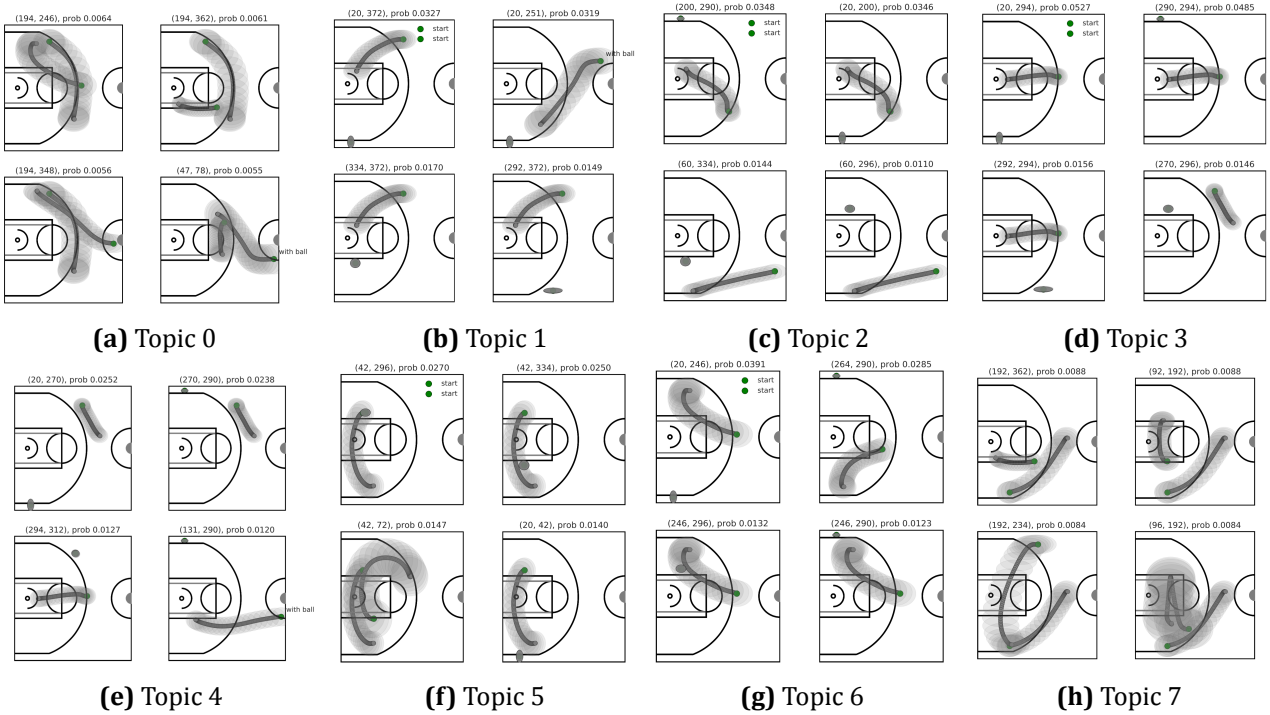
Figure 3 illustrates the construction of our pair-action vocabulary that we use to succinctly represent each possession. This representation allows us to easily apply LDA to basketball possessions. To scale this model to the over $190{,}000$ possessions in the season, we use a newly developed scalable Bayesian inference technique [9]. See the appendix for technical details.

This model yields a low-dimensional embedding of every NBA play that allows us to quickly assess similarities between possessions and explore the space of team offensive strategies. We can create interactive graphics (a dynamic version of Figure 5a), where each point in space represents an NBA possession and nearby points indicate "similar" possessions — possessions that share the same pattern of actions. The following section dives deeper into this exploration tool, and what it can afford an analyst. The *topics* themselves encode strategic co-occurrences of actions, and using these topics we can shed light on the fundamental building blocks of collective action on the basketball court. Inspecting these topics can help us quantify what exactly makes a unique offense unique.

3 Analysis

In this section, we explore the output of the possession level model to see which patterns are represented. We focus on the following aspects of model output

[1] In document modeling, the vocabulary is typically the vocabulary of the language itself, with minimal preprocessing. Bigrams and tri-grams are sometimes included as vocabulary words to improve the model.



(a) Topic 0 **(b)** Topic 1 **(c)** Topic 2 **(d)** Topic 3

(e) Topic 4 **(f)** Topic 5 **(g)** Topic 6 **(h)** Topic 7

Figure 4: The result of fitting a $K = 100$ topic possession model. A "topic" in our framework corresponds to a distribution over *pairs* of actions. Above, we show common *pairs* of actions from 8 of the 100 topics. We observe that topics tend to pick up on combinations of actions that include common actions. For instance, the top two pair-actions in topic 0 includes a cut along the 3-point line while a teammate cuts nearby (perhaps setting an off-ball screen).

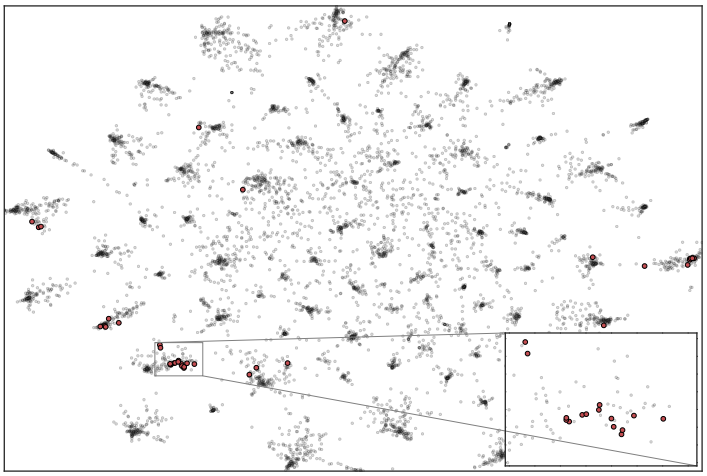
- *basketball topics*: we see which *pair-actions* are represented by each of the $K = 100$ topics. This tells us not only which pair-actions occur frequently, but which pair-actions co-occur in possessions — this will reveal structure
- *possession sketch*: each possession can be described as a distribution over topics, and "similarity" can be measured using this topic-loading vector. We explore what "similarity" means for basketball possessions according to our model, and we empirically test this notion of similarity by measuring distances between set plays we know are similar.

In the following sections we explore the above concepts by visualizing and exploring possessions in ways newly afforded by our framework.

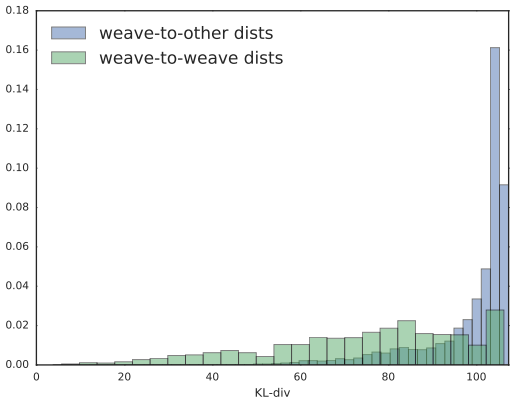
3.1 Basketball Topics

Figure 4 graphically depicts a small sampling of "topics" discovered by the possession model. The topics reveal which pair-actions are most common in our data set, and we do see patterns emerge. As a concrete example, if a particular possession "loads" onto topic 3[2] then that possession is more likely to include the pair-actions depicted in Figure 4d — a cut to the basket while a teammate is standing in either of the two corners. Topic 5 prominently includes possessions with a baseline cut from the right block to the left break. Note that there are *many more* pair-actions with significant probability than the ones depicted, and there are many more topics than we depict.

[2] i.e. the possession sketch vector is large along the dimension corresponding to topic 3



(a) Warriors Possessions (t-SNE map)



(b) Weave-to-weave and weave-to-other distances

Figure 5: Left: map of 2014-2015 Warriors Possessions, with a small set of known "weave" plays highlighted in red. The weave plays tend to cluster together in this visualization. We verify this by computing the average distance between two weave possessions and between a weave and a random warriors possession of a similar length. This indicates that our topic-model-based representation is picking up on patterns that are able to (mostly) distinguish between semantically different plays.

Sparsity We also notice that each possession topic vector is quite sparse — on average only 8 of the 100 entries are non-zero (on average). This makes intuitive sense — each possession can only include a small number of offensive patterns from the wide array of available tactics.

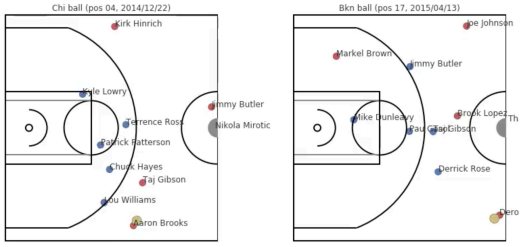
3.2 Possession Map Exploration

Each possession has an associated *possession sketch* — a per-topic vector that describes how much of each of the *basketball topics* (a subset illustrated in Figure 4) are featured in that possession. We can use these *possession sketches* to reason about large sets of basketball possessions. In this section we select the offensive possessions of the 2014-2015 Golden State Warriors (over $6{,}000$ possessions). With each possession succinctly described by a (sparse) 100-dimensional topic vector, we use the dimensionality reduction technique t-SNE [10] to visualize these vectors in 2-dimensions. This method finds a 2-dimensional representation of each 100-dimensional vector such that the distance in 2-d is similar to the distance in 100-d (emphasizing the preservation of local distances).[3] Figure 5 visualize all warriors possessions in 2014-2015.

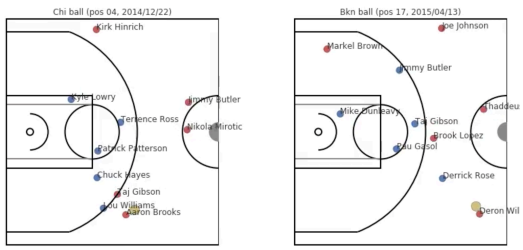
We test the notion of "similarity" in topic space by examining a group of hand-labeled *set plays*, a "weave play". We animate two examples of the weave play in this animated figure: https://youtu.be/KRDsTLMm7FY. We hand-label 40 weave plays in the 2014-2015 season, and visualize them in the t-SNE Warriors map (Figure 5a, in red). We can visually verify in Figure 5 that the possession sketch preserves this notion of similarity — weave plays tend to cluster around other weave plays.

We can further measure this clustering by comparing two distributions of possession sketch distances: (i) the distribution of distances between two weave plays, and (ii) the distribution of distances between one weave, and one non-weave play. Figure 5b illustrates these two distributions. The average distance between the known weave plays is much smaller than the average distance between weave and non-weave plays. In fact, the nearest neighbor of each weave play is most often itself a weave play,

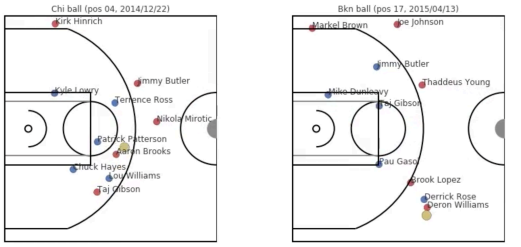
[3] For intuition, t-SNE tends to yield a visualization where locally clustered points are close in distance in the full, 100-dimensional topic space; points that are farther away from each other tend to be far, but could also be close.



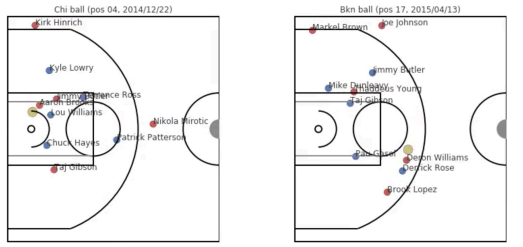
(a) frame 1: (left) Aaron Brooks brings the ball along the left; (right) Deron Williams brings the ball along the left



(b) frame 2: (left) Taj Gibson sets a high screen in the left frame; (right) Deron Williams waits for a screen in the right frame.



(c) frame 3: (left) Aaron Brooks curls around the screen and drives; (right) Brook Lopez sets a high screen for Deron Williams



(d) frame 4: (left) Aaron Brooks attacks the basket; (right) Deron Williams curls around the screen and drives

Figure 6: An example of two very similar possessions: each sub-figure displays key frames from two possessions — one where Chicago has the ball and one where Brooklyn has the ball. These frames highlight similar features between the two possessions. For a clearer picture of "possession similarity", please navigate to https://youtu.be/0Jlj6xekxeI to see these plays animated.

highlighting the potential of our technique to quickly find a collection of plays similar to a chosen play.

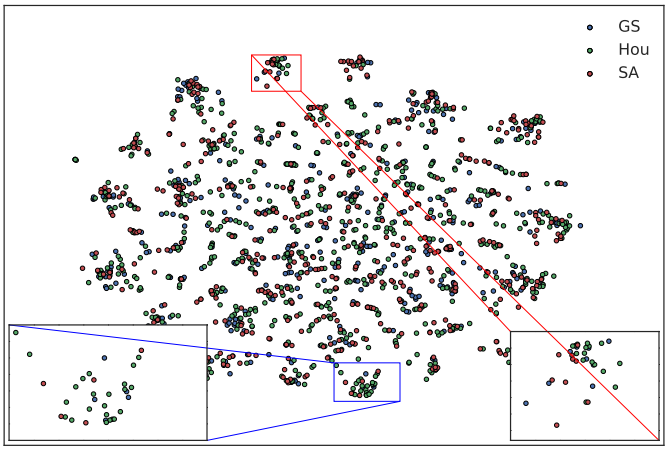
3.3 Between Team Nearest Neighbors

Our method also identifies similar possession structure between different teams. To highlight this, we select a play at random, and search through the entire database of 190,000 possession sketches to find the most similar play. The resulting two possessions are compared in Figure 6. Chicago is on offense in our first possession, and Brooklyn is on offense in the nearest-neighbor possession.

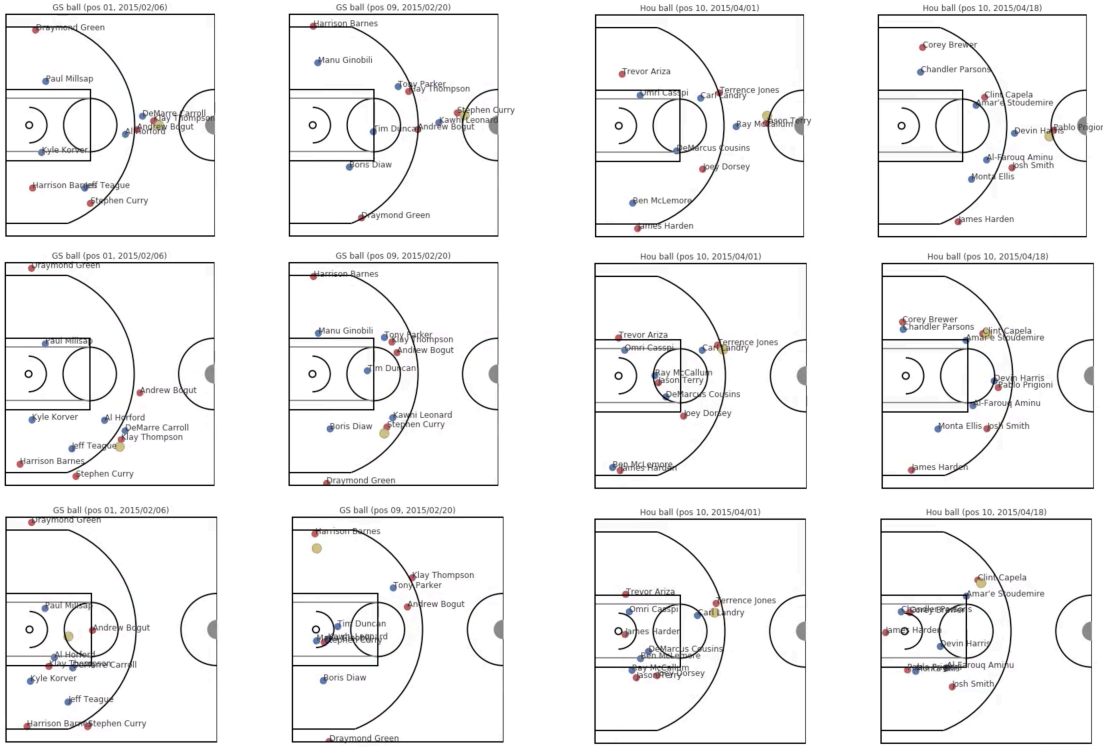
We examine what exactly is similar between these two possessions — which basketball patterns are being modeled by our method. There are a handful of salient similarities: (i) the point guard brings the ball up the left side of the floor in each possession; (ii) a player sets a high screen on the left side, and the point guard curls around the screen toward the middle with the ball; (iii) through both possessions a player camps out in the weak-side corner three; (iv) the point guard attacks through the middle of the paint. The possession sketch contains this information — and we can further inspect the particular basketball topics for this possession to see how this information is summarized in our model.

3.4 Corner Threes

In this section we explore possession sketch similarity in the context of a particular type of shot — a corner three. We first sub-select the 2014-2015 data to possessions that include corner three-point



(a) Corner 3s



(b) Left cluster example: key frames

(c) Right cluster example: key frames

Figure 7: Corner Three. The left pane cluster examples are similar in that they include a drive to the basket, and a pass to a teammate camping out in the corner. The right pane cluster esamples are similar in that they include a baseline cut toward the corner in which the shot is taken. Please see the animated figures at https://youtu.be/hUuPkE06rX4 (left), and https://youtu.be/mMcWuqgrj1w (right).

shots for three teams: the Warriors, the Rockets, and the Spurs. We then apply t-SNE to visualize the resulting shots in Figure 7a. We can immediately notice that the possession sketches that lead to corner threes overlap significantly between teams, however there are some regions of the space in which the Rockets are more likely to inhabit than the Spurs.

We examine the structure of the possession-map clusters by zooming in on two groups on the op-

posite side of the map. Figure 7 compares two possessions in the cluster in the left-pane to two possessions in the cluster in the right pane. An immediate difference between the two clusters is that the right pane includes a baseline cut toward the corner in which the shot is taken, whereas the left pane includes a drive into the middle, and a pass out to a player camping out in the corner.[4] Indeed, these are two very different ways of ending up with a corner three point attempt, and our method identifies this and allows us to efficiently explore this different structure.

4 Discussion

Related work This paper develops a framework for exploring interpretable patterns in player-tracking data — applications of this framework can enhance player evaluation and media consumption. A similar system for measuring play similarity was developed in [12], based on point-wise similarities in trajectories. We take a more data-driven and global approach — we use scalable machine learning techniques to fit a probabilistic model to an entire season's worth of player tracking data, directly modeling player interactions. The result is a more interpretable, succinct, and scalable decomposition of possessions.

In [4, 3], the authors propose a stochastic process model to measure the moment-by-moment expected possession value (EPV) of a basketball sequence. They handcraft a set of basketball states that are used in the model. Our approach is more of a data-driven decomposition of basketball states that we use for exploration (but could be used within an EPV model). Other examples that develop data-driven representations from player-tracking data can be found in [11, 7, 6].

Future work and conclusion There are multiple avenues for future work. Firstly, we can improve the action-template method by inferring the number of actions using more sophisticated methods, such as Bayesian nonparametrics. The action templates should also have more temporal structure — autocorrelation and dynamic variance. Further, our possession sketch ignores much of the temporal information in each possession (a trade-off for statistical and computational efficiency). A future project could further describe the time-varying nature of possession strategies, which, for example, would allow us to identify which possessions may have started out in a "weave" set, but broke down into a different sequence.

Insight derived from player-tracking data has been promised more than delivered. We reduce this gap by devising a method that will have a profound impact on the use of player-tracking data for analysis — from summarizing situational statistics (e.g. how often did the "weave" play succeed?), to searching for similar plays (e.g. for post-game analysis), to discovering and quantifying previously unknown habits of interaction between players (e.g. for team-specific scouting).

References

[1] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[3] Dan Cervone, Alexander D'Amour, Luke Bornn, and Kirk Goldsberry. Pointwise: predicting points and valuing decisions in real time with NBA optical tracking data. 2014.

[4] Daniel Cervone, Alexander D'Amour, Luke Bornn, and Kirk Goldsberry. A multiresolution stochastic process model for predicting basketball possession outcomes. *arXiv preprint arXiv:1408.0777*, 2014.

[4] Please refer to animated figures https://youtu.be/hUuPkE06rX4 (left pane) and https://youtu.be/mMcWuqgrj1w (right pane).

[5] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[6] Alexander Franks, Andrew Miller, Luke Bornn, and Kirk Goldsberry. Counterpoints: Advanced defensive metrics for nba basketball. In *2015 MIT Sloan Sports Analytics Conference*, 2015.

[7] Alexander Franks, Andrew Miller, Luke Bornn, Kirk Goldsberry, et al. Characterizing the spatial structure of defensive skill in professional basketball. *The Annals of Applied Statistics*, 9(1):94–121, 2015.

[8] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[9] Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[11] Andrew Miller, Luke Bornn, Ryan Adams, and Kirk Goldsberry. Factorized point process intensities: A spatial analysis of professional basketball. In *Proceedings of the 31th International Conference on Machine Learning (ICML-14)*, 2014.

[12] Long Sha, Patrick Lucey, Yisong Yue, Peter Carr, Charlie Rohlf, and Iain Matthews. Chalkboarding: A new spatiotemporal query paradigm for sports play retrieval. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 336–347. ACM, 2016.

[13] Kuan-Chieh Wang and Richard Zemel. Classifying nba offensive plays using neural networks. 2016.

A Appendix

A.1 Segment Clustering Model Details

Our clustering model specifies V Bezier curve components, $B_v : [0,1] \mapsto \mathbb{R}^2$, each parameterized by $\theta_v \in \mathbb{R}^{P\times 2}$, where P is the number of *control points* used to characterize the curve.[5]

$$B_v(t;\theta_v) = \theta_v^\intercal D_P(t) \tag{2}$$

$$D_P(t) = \binom{P}{p} t^p (1-t)^{P-p} \quad \text{for } p = 0,\ldots,P-1) \tag{3}$$

Importantly, each curve can be specified as a linear function with respect to parameters θ_v, with a non-linear (but fixed) basis in time, $D_P(t)$. Bezier curves are a natural choice for these data — they are flexible, concisely parameterized, and easy to fit. The non-linear basis in time allows for a wide variety of template shapes.

The complete functional clustering model is specified as

$$z_{j_s}^{(n)} \sim Pr(\textit{action}|\pi) \qquad \text{action type} \tag{4}$$

$$\boldsymbol{x}_{j_s,t}^{(n)} \sim \mathcal{N}(B_v(t,\theta_v),\Sigma_v) \qquad \text{location at moment } t \tag{5}$$

We use maximum likelihood to learn parameters θ_v, π, and Σ_v (and therefore each action) directly from the data set of 4.5 million trajectory segments. To do so efficiently, we devise expectation maximization [5] updates that exploit the linear structure of Bezier curves — each maximization step can be computed using weighted least squares. Further, each expectation step can operate on each segment in parallel, allowing us to scale our method up to the 4.5 million trajectory segments. We omit the technical details of the inference procedure in this writeup for brevity.

A.2 Topic Model Details

Conceptually, LDA defines K *topics*, ϕ_k, each a distribution over *actions*. Each observed possession is characterized by some latent distribution over *topics*, $\pi^{(n)}$, which describes the probability that a particular topic is expressed in possession n. These two distributions — possession-specific proportions and global topics — determine the probability of observing any particular action in possession n. LDA posits the following data generating process to give rise to the matrix of counts

$$\phi_k \sim \texttt{Dir}_V(\alpha_0) \text{ for } k = 1,\ldots,K \tag{6}$$

$$\pi^{(n)} \sim \texttt{Dir}_K(\alpha) \text{ for } n = 1,\ldots,N \tag{7}$$

$$Y_{n,:} \sim \texttt{Mult}(M_n, p = \sum_k \pi_k^{(n)} \phi_k) \tag{8}$$

where M_n is the total number of actions present in possession n (a fixed constant). We use statistical inference techniques to infer both the global topics, Φ, and the possession-specific proportions, $\pi^{(n)}$ for all possessions. Due to the size of the dataset, we use stochastic variational inference [9], a scalable method for Bayesian inference in hierarchical models.

[5] More control points allow for more flexibility in fitting shapes — we use 10 control points in our experiments