15TH ANNUAL
MIT SLOAN
SPORTS ANALYTICS CONFERENCE
APRIL 8-9, 2021 | VIRTUAL
presented by ESPN 42ANALYTICS

Decoding MLB Pitch Sequencing Strategies via Directed Graph Embeddings

Abstract

This paper presents a novel analysis of pitch sequencing in Major League Baseball (MLB). By leveraging high-resolution pitch tracking data from ~3.6 million pitches across the 2015-2019 seasons, this work introduces directed graph (network) embeddings that successfully map short- and long-term patterns in pitch sequences. This quantitative approach to pitch sequencing captures the intuition that pitchers exhibit a sense of long-term memory when on the mound that is not adequately represented by individual pitch selection or simple pitch-to-pitch correlation. By interpreting the graph embeddings as forward dependencies, there is compelling evidence that pitchers construct their sequences via strategic components, denoted “setup” and “knockout” pitches. Model-based clustering on the graph embeddings suggest that MLB pitch sequences can be grouped into a finite collection of universal patterns with respect to both pitch-type and zone selection. Exploratory data analysis of these sequence clusters indicate that a pitcher’s sequencing strategies are distinct—though not inseparable—from their available pitch arsenal; that is, in addition to pitch selection, pitch ordering is a significant component of pitcher decision-making. Moreover, pitchers exhibit patterns or styles in how they sequence their pitches over the course of a single matchup, game, season, and career. Ultimately, this paper introduces an analytical framework to study and visualize MLB pitch sequences with potential applications in matchup preparation, player evaluation, and player development.

1. Introduction

Each moment in a baseball game is an extension of the battle between pitcher and batter. This conflict perhaps most closely resembles that of a high-performance mixed martial arts fight, where there is immense pressure—and corresponding incentive—for pitchers to optimize their approach against each batter they face. In this matchup, a pitcher’s arsenal can be imagined as a set of techniques that each pitcher has at their disposal to defeat their opponent. However, a pitcher’s pitch arsenal alone (i.e., the moves they have available) is an incomplete representation of a pitcher’s aggregate style. By treating both the selection and ordering of baseball pitches, pitch sequencing is much more informative of pitcher decision-making by taking into account how pitchers learn, build upon, and order their pitches.

A major analytics challenge in sabermetrics is capturing differences between pitchers with respect to not only their performance on the field but also the styles they exhibit and the strategies they deploy. Although each at-bat between a pitcher and batter is ultimately either won or lost using an approximately unique collection of pitches, patterns may emerge wherein sets of pitch sequences recur across various in-game settings. If pitchers share pitch arsenals or face similar batter-matchup situations, it is also expected that sets of pitch sequences manifest across many individuals and across seasons. Conversely, pitchers can also distinguish themselves from others in their role that share their pitch arsenal through differences in their pitch sequencing. This is advantageous to the pitcher since it separates them from their peers and is critical for teams to diversify talent in their roster.

However, preexisting analytical research in pitch sequencing currently limit sequence analysis to pitch pairs via pitch-to-pitch correlation; that is, a pitch sequence is defined exclusively in terms of consecutive pitches [3]. This memoryless approach is likely not reflective of how pitchers pursue decision-making since it constrains pitchers to very short-term learning. These early methods gloss over some of the crucial qualities that make pitch sequencing as captivating as it is and generally fail to provide insights as to the strategic motivations behind pitch sequence selection.

This paper introduces a graph-based solution to capture the complexity of patterns that are latent in pitch sequences. Unlike prior statistical approaches that focus primarily on pitch-level predictability or pitch pairings, this method successfully embeds short- *and* long-term patterns in pitch sequences in a finite-dimensional feature space. Each graph embedding, given its interpretation as a forward dependency between two pitches, also preserves the directed (i.e., ordered) nature of baseball pitch sequences. From this property of the embeddings, this paper presents a suite of tools that can be valuable in a strategic setting within a team front-office or player development unit. In order to more closely analyze how pitchers construct their sequences, the analysis in this paper separately considers pitch sequences based on pitch-type constitution and zones. Through model-based clustering, this paper identifies a collection of pitch sequencing strategies that are common throughout the MLB and vary in their constitution, ordering, and entropy. Comparative analysis between sequence clusters provide insight into core structural elements of pitch sequencing strategy. In particular, this work finds evidence for “setup” and “knockout” pitches by drawing from the concept of sinks in graph theory. With respect to how players select pitch sequences, it is clear from these clusters that pitchers who share pitch arsenals can also differ drastically in how they order those pitches in game situations. Applications of this work to baseball operations, analytics, and scouting are suggested and explored throughout the paper.

2. Methodology

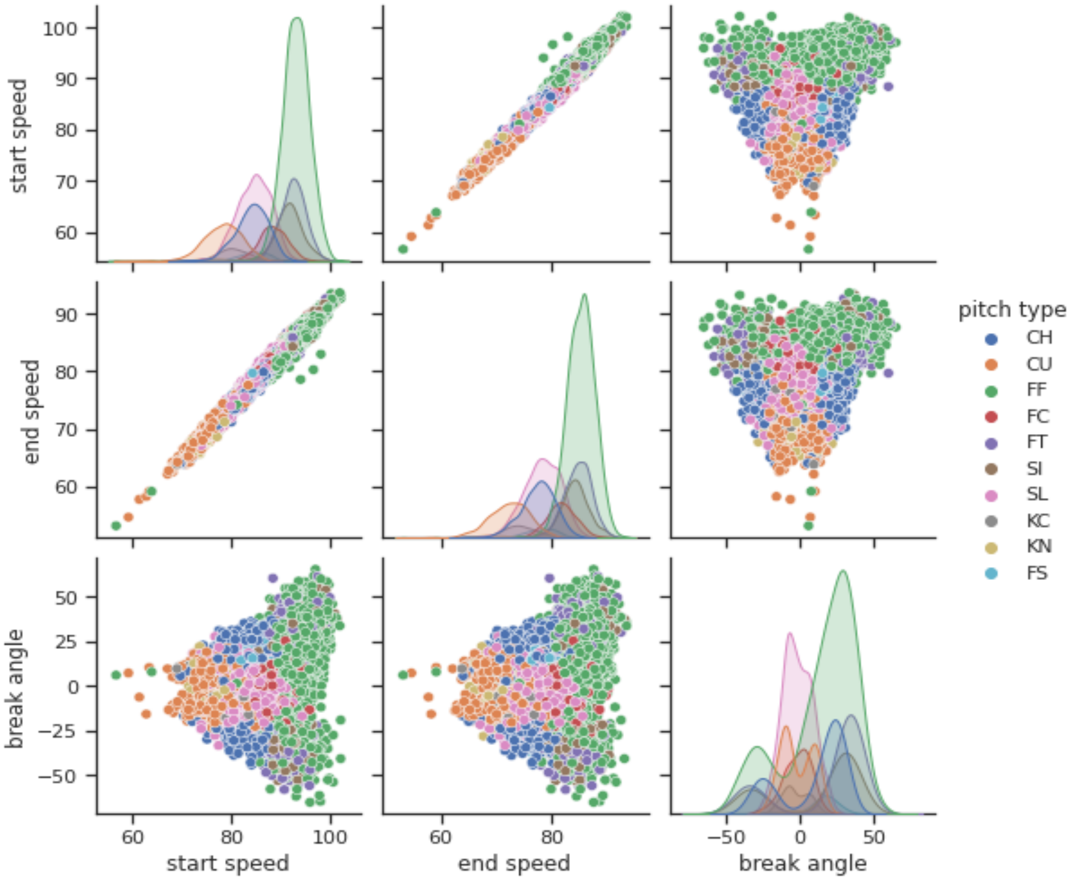
The following sections will describe the data and statistical methodology that underlies the findings. Section 2.1 provides an overview of the paper’s primary dataset(s) and discusses the model-specific implications of data quality. Section 2.2.1 introduces Sequence Graph Transform (SGT), the graph embedding algorithm that encodes each pitch sequence into a feature space. Furthermore, Section 2.2.2 details the specific mathematical properties of SGT that make it attractive for practical application to this problem. Finally, Section 2.3 specifies the clustering technique that is applied to determine optimal groupings of pitch sequences without relying on pre-existing labels or assumptions.

2.1 Data

It is no secret that baseball has experienced a technological revolution, largely thanks to an explosion of reliable high-resolution data. All in-game data—including all measurements on pitches, id tracking, and catcher data that are included in this paper—are publicly available through the MLB Statcast API [2]. The pitching data includes 3,595,944 pitches from the 2015-2019 MLB seasons that are provided alongside 40 features that describe pitch-level characteristics and information that contextualizes when each pitch was thrown. In aggregate, 1523 distinct pitchers and 1894 distinct batters are represented in the data. This dataset—or a shortened form of it—has been used extensively in recently published baseball analytics research [4].

Since this paper is primarily concerned with sequential data, it is necessary to rely on discrete pitch-classification labels. The unique pitch-type and zone labels in the MLB dataset serve as the alphabet that the embedding algorithm will learn from. The accuracy and precision of the pitch-classification labels in this dataset is thus critical to the success of this research. This methodology assumes homogeneity within pitch-types with respect to pitch-level characteristics, though it is clear in Figure 1 that this assumption has shortcomings. It is important to note that pitch classification itself is a non-trivial problem. Currently, in-house MLB models perform real-time pitch-classification for nearly 750,000 pitches each season using patented neural networks [2]. Irrespective of the pitch-classification labels that are utilized in this analysis, independent researchers can easily adapt this work using their own pitch classification systems or datasets.

Figure 1: *Pitch Characteristic Pairplot*



Several adjustments to the dataset are made to improve the quality of modeling. At-bats with incomplete measurement/label data are discarded from the dataset to minimize the inadvertent effects of noise or calibration issues. Several features are also added to the raw dataset. To avoid look-ahead bias in predictive modeling, rolling statistics are used to report pitcher and batter performances and track changes in player outcomes overtime [4]. Additionally, to compare the sequencing tendencies of pitchers in various roles and contexts, "starter" designations for each pitcher are inferred from the information available in the first at-bat in top of the first inning and the first at-bat in the bottom of the first inning [4].

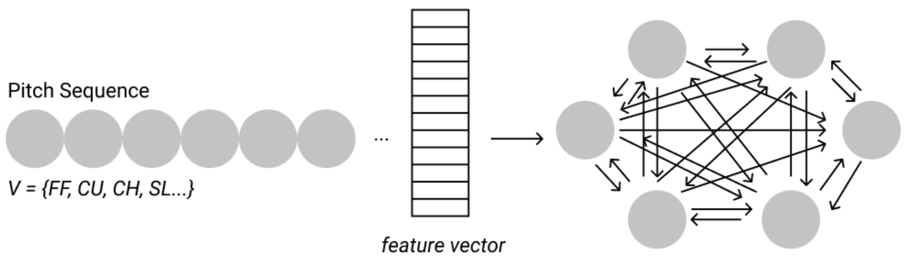
2.2.1 Sequence Graph Transform – Overview and Intuition

This paper applies a graph-based feature extraction method to encode for patterns in MLB pitch sequences. This method—Sequence Graph Transform—achieves the goal of representing the complexity of pitch sequencing in terms of both pitch constitution and ordering.

A sequence can be defined as a set of discrete items that are structured in an ordered series [5]. Sequences are one of the most common data types in natural, physical, and computational settings. For example, sequencing mining techniques have found value bioinformatics and computational genomics. Pitch sequences naturally follow this sequential structure; both the order and constitution of a pitch sequencing are important in disrupting a batter's calibration to the pitcher's strategies. However, since pitch sequences represent unstructured data— defined by arbitrarily placed alphabets of an arbitrary length—their mapping into a Euclidean space is non-trivial. Moreover, since pitch sequences generally extend beyond two-pitches, this analysis requires a feature extraction method that specifies long-term dependencies (i.e., the effect of distant elements in a sequence on each other) in addition to short-term dependencies (i.e., the effect of elements in a sequence that occur in short succession of each other).

This paper utilizes Sequence Graph Transform (SGT), an embedding function that extracts patterns in sequences and maps them into a finite-dimensional Euclidean space. There are multiple advantages that are implied in the function and make SGT particularly attractive to the problem of pitch sequencing. SGT operates by quantifying the observable patterns in a sequence by scanning the positions of each item relative to other items. The outputted embeddings can be interpreted as a directed graph, where each alphabet (e.g., pitch-type) becomes a node and a directed connection between two nodes explains their association. In order to maximize the utility of sequencing mining in baseball pitch sequence analysis, it is important to have a sequence-level measure of (dis)similarity between various sequences. Similarity is given explicitly by the dot-product between two separate embeddings where higher results indicate higher similarity. Additionally, SGT provides a solution to extract length-insensitive patterns in pitches. As a feature embedding tool, SGT is particularly valuable because it provides a machine-interpretable representation of a complex data-type. The graph interpretation of each embedding allows for pitch-types to form nodes and the directed connection between nodes as a signal of the strength of their association. Graph embeddings also enable unsupervised learning and clustering for pattern analysis. Moreover, each graph embedding provided by SGT is directed; that is, both the constitution and the ordering dynamics of each pitch sequence are maintained. [5]

Figure 2: *SGT Application to Pitch Sequences*



Although there are embedding alternatives to SGT, these existing methods either fail to account for long-term patterns, produce false positives, or substantially increase computation. For example, if each sequence is transformed into a string, there are several metrics which define the distance between two unique patterns (e.g., edit distance, hamming distance). However, these metrics improperly account for the complexities of pitch sequences and do not offer the advantage of representing pitch sequences in terms of forward dependencies. Measures drawn from information theory such as Shannon entropy, though informative of the level of "uncertainty" or "randomness" in a sequence, also do not provide enough granularity to explain sequence characteristics, namely pitch constitution and ordering [6].

If pitch sequences do not correspond in terms of both short and long patterns, SGT embeddings will not produce a high similarity match. While the pitch sequences *FF-CU-FF*, *FF-CU-FF-FF-CU-FF*, and *FF-SL-FF-CU-FF-SL-FF-SL-SL* all share a subsequence in *FF-CU-FF*, SGT distinguishes the first two examples from third given the overall, long-term differences in the sequence pattern. Conversely, traditional subsequence matching methods would produce a similarity match between all three sub-strings [5]. Given the application setting, SGT is preferable since traditional subsequencing matching methods would produce a false positive. Finally, SGT is widely applicable across multiple domains and does not exhibit setting bias. The SGT algorithm is publicly available, and its low computational requirements allow it to be run in local environments [5].

2.2.2 Sequence Graph Transform – Definition

Suppose a set of sequences S is contained in a dataset. Any individual sequence in the dataset can be represented by $s \in S$, where s is constituted by an alphabet $\mathcal{V}$. For the purposes of pitch sequencing, alphabet $\mathcal{V}$ either contains a collection of pitch-type labels or discrete zone-location labels. Each sequence s has at least one instance of one element from the alphabet $\mathcal{V}$, though it is not necessary for every element in alphabet $\mathcal{V}$ to be represented in each set s. In baseball terms, a pitch sequence is defined by at least one pitch that is thrown in an at-bat; moreover, due to pitch arsenal constraints, strategic decision-making motivations, and in-game feasibility, it is not expected that a pitcher throws every pitch-type in a single sequence. The length of a specific sequence s is denoted by $L^{(s)}$. For each sequence, s_l will represent the element that occurs at position l where $l = 1, \dots, L^{(s)}$ and $s_l \in \mathcal{V}$. $\varphi_\kappa(d)$ is a function that takes a distance as input and κ as a tuning hyper-parameter. The features for sequence s are extracted in the form of associations between alphabet instances. Each association is denoted as $\psi_{uv}^{(s)}$, where $u, v \in \mathcal{V}$ correspond to specific alphabet instances and ψ corresponds to a helper function of φ.

Since SGT considers the relative positions of instances to form each feature, $\varphi\big(d(l,m)\big)$ quantifies the information that is extracted from the relative positions of two instances, where l, m are the positions of two distinct instances and $d(l,m)$ is a corresponding distance output. Specifically, $\varphi\big(d(l,m)\big)$ denotes the effect of the first instance on the later instance.

To apply SGT on a set of sequences S, the following conditions must hold on φ: (1) The function output is strictly greater than 0 such that $\varphi_\kappa(d) > 0; \forall\kappa > 0, d > 0$; (2) The function strictly decreases with d such that $\frac{\delta}{\delta d}\varphi_\kappa(d) < 0$; and (3) the function strictly decreases with κ such that $\frac{\delta}{\delta\kappa}\varphi_\kappa(d) < 0$. The first condition is designed to maintain the interpretability of each SGT embedding, whereas the second condition maintains that closer neighbors have higher corresponding dependencies, and the last condition maintains that the effect of distant neighbors can be effectively tuned.

SGT uses an exponential function for φ since it satisfies all three conditions. Using this approach, the distance between two distances can be taken simply as $d(l, m) = |m - l|$.

$$\varphi_\kappa\big(d(l,m)\big) = e^{-k(m-l)}, \forall\kappa > 0, d > 0 \tag{1}$$

Since pitchers throw multiple pitches in a sequence, there are likely to be several instances of alphabet pairs (u, v). The SGT algorithm is initially concerned with determining how many of each alphabet pair exists in the sample. Each alphabet pair is ultimately stored in a $|\mathcal{V}| \ \times |\mathcal{V}|$ asymmetric matrix Λ. Λ_{uv} will contain every alphabet pair (u, v) that is contained a specific sequence s such that the v's position can always be inferred to be after u in each instance pair.

After computing φ for each (u, v) pair instance that is contained in a sequence, the association feature $\psi_{uv}^{(s)}$ is defined as the normalized aggregation of all instances. Since all analysis conducted in this paper is concerned with length-insensitive problems, the effect of length is controlled for by normalizing $|\Lambda_{uv}|$, the size of the set Λ_{uv} or the number of unique (u, v) pairs, with the sequence length $L^{(s)}$as shown in Eq. 2.

$$\varphi_{uv}(s) = \frac{\sum_{\forall(l,m)\Lambda_{uv}(s)} e^{-k(m-l)}}{|\Lambda_{uv}(s)|/L^{(s)}} \tag{2}$$

The SGT feature representation of an entire sequence can be represented as the aggregation of $\Psi(s) = [\psi_{uv}(s)], u, v \in \mathcal{V}$. The features $\Psi^{(s)}$can be interpreted as a directed graph with edge weights ψ and nodes in $\mathcal{V}$ for each sequence s in the dataset.

SGT contains a single tuning parameter κ. κ modulates the extent to which long-term dependencies are captured in each embedding. A small value of κ preserves longer-term dependencies. Since the average pitch sequence is approximately 5 pitches at the at-bat level, a small $\kappa = 1$ (default value) is selected. [5]

2.3 Model-Based Clustering – Gaussian Mixture Models

Unsupervised learning via model-based clustering can be used to discover patterns in at-bat pitch sequences. This paper relies on Gaussian Mixture Models (GMMs), a model-based clustering technique that uses an expectation-maximization (EM) algorithm, to find clusters that capture the common patterns that manifest and recur in pitch sequences. Given there is no set of intuition that informs how number of these clusters or patterns exists, this analysis requires an approach that self-determines an optimal number of target clusters K from the graph embeddings.

Other popular algorithms, such as hierarchical clustering, are also able to produce K without user labels, using measures of in-cluster and out-of-cluster distance and variance [6]. However, current hierarchical clustering algorithms scale in $O(n^3)$ time and often produce unstable results [6]. This computational hurdle is especially relevant when dealing with a high volume of data across multiple seasons.

To assess the potential instability of any results, it is desirable to obtain a probability estimate that a sequence belongs to their assigned cluster. Additionally, GMMs can frame K-selection as a model-selection problem using likelihood-based measures such as Bayesian Information Criterion (BIC), which benefits interpretability [7].

A Gaussian Mixture is a function that treats several Gaussians, each identified by $k \in \{1, \dots, K\}$. Each Gaussian k includes a mean μ that defines its center, a covariance Σ that defines its width, and a mixing probability π that defines the size of the Gaussian function, as shown by Figure 2 [8].

Eq. 3 explicitly defines the Gaussian Mixture Model Likelihood Function used in this approach. The Gaussian Mixture Model Likelihood-Function computes a Maximum-Likelihood Estimate (MLE) to find the optimal distribution that underlies the dataset.

$$L(\theta|X_1, \dots, X_n) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k N(x_i; \mu_k, \sigma^2) \tag{3}$$

All modeling and associated data analysis were conducted in Python in a Jupyter Notebook environment. The Gaussian Mixture Models and Expectation Maximization algorithm were built and implemented from scratch.

3 Pitch-Type & Zone Sequencing Strategies

During MLB gameplay, pitchers engage in advanced decision-making. Pitchers are incentivized not only to promote unpredictability in their sequencing decisions but also adapt to the strengths and weaknesses of the batter they matchup with. In each of these subsequent analyses, it is assumed that a pitcher is capable of "calling their own pitches;" that is, pitchers are largely responsible for what pitches they throw. The intuition to support this assumption is strong—pitchers do not completely redesign their pitching styles whenever their catcher or umpire changes, for example.

Generally, pitch sequencing can be distilled into two distinct components: Pitch-type selection and zone (location) selection. In order to fit MLB pitching data to the required input structure of the Sequence Graph Transform algorithm, pitch sequences are represented as ordered lists of pitch-type labels and discrete zone-location labels, respectively (See Section 2.2.1). From these embeddings, unsupervised learning (Gaussian Mixture Models) is used to cluster pitch-type and zone sequences independently (See Section 2.3). Then, each at-bat sequence is assigned with two cluster labels (pitch-type and zone) and is paired with relevant in-game information (e.g., score, inning, result of the sequence).

This analytical design has multiple advantages: (1) It is possible to discern which sequencing strategies are common in a variety of in-game contexts; (2) The covariance of pitch-type and zone-labels can be analyzed, and the complexity between pitch-type selection strategies and zone selection strategies can be compared; and (3) Sequencing styles across players and time periods can be systematically evaluated.

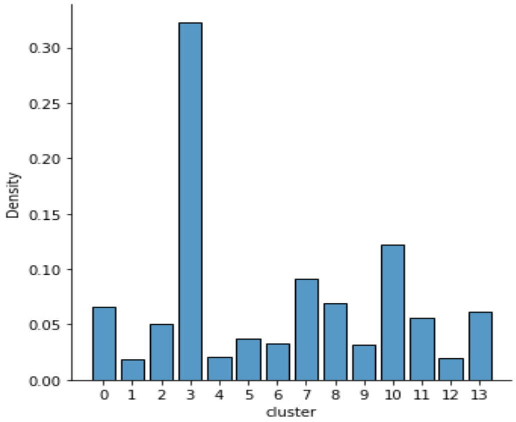
3.1.1 Pitch-Type Sequence Clusters

Modeling results from pitch-type sequences show that there are many diverse groups of pitch sequences at the at-bat level that are thrown in the MLB. Since the optimal number of clusters is selected on Bayesian Information Criterion (BIC), the number of pitch-type sequence clusters is informative of the aggregate complexity of pitch-type sequencing strategies. Each sequence cluster varies in their pitch-type constitution and ordering (See Table 1; Note that only the top three pitch-types by relative frequency are displayed in Table 1 for each cluster for the sake of clarity).

The fitted Gaussian Mixture Model supplies a probabilistic distribution for each sequence's assignment to a pattern cluster. The overwhelming majority of at-bat sequences (99.87%) are assigned to clusters with a high probability (greater than or equal to 99% assignment probability). The 0.13% of at-bats that are assigned to a cluster with less than an associated 99% probability all have an assignment probability greater than 50% and have a mean assignment probability of 86.37%. Within this 0.13% of at-bats, it is important to note there is no explicit pattern as to which sequences seem difficult to assign to a cluster. Given this distribution, it is clear that most sequences map very strongly to a single cluster and that the model-results can be interpreted in terms of a primary cluster assignment.

From Table 1, it is clear that pitchers rely on a variety of pitch sequences across matchups. Since clustering takes into account all pitch sequences for pitchers that meet an at-bat minimum, the model results are expectedly correlated with which arsenals are most popular in the MLB (See Figure 3, Table 1). For example, cluster 3 contains pitch sequences that include three of the most popular pitch-types in the MLB, so it is expected that this cluster is also relatively popular among pitchers.

Figure 3: *Pitch-Type Sequence Cluster by Relative Frequency/Density*



Within each pitch sequence, it is observable that there is a split between fastball types and off-speed pitches that are thrown; in other words, every sequence pattern involves both fastball pitches and off-speed pitches, though in varying frequencies. These results present significant evidence that pitch-type mixing is a critical aspect of all sequencing strategies across a diverse set of arsenals. Note that sequence clusters that share pitch-types at high frequencies also differ in their ordering and entropy (See Appendix: Table 2, Table 3).

While it is more difficult to summarize ordering in a table or visualization, Table 2 (Appendix) presents a side-by-side example-based comparison of cluster 3, cluster 7, and cluster 10. Each at-bat example yields an estimated 99% probability of membership or greater, which implies that these sequences are exemplary of the identified sequence cluster. It is observable that sequence clusters with similar frequencies of pitch-types also differ in ordering.

The ordering component is highlighted further by which pitch-types tend to be thrown early in a sequence versus later in the sequence (Table 2; colored by beginning, middle, and end of the sequence). Finally, it is evident that each cluster is approximately the same length, which is expected since the graph embeddings learn length-insensitive patterns in pitch sequences.

Table 1: *Pitch-Type Cluster Membership*

Key
🟥 Fastball
🟦 Off-Speed

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Pitch-Type Freq.	CU (0.398)	FC (0.292)	FF (0.436)	FF (0.598)	FF (0.232)
Pitch-Type Freq.	FF (0.325)	SI (0.269)	FC (0.282)	SL (0.291)	CH (0.185)
Pitch-Type Freq.	SL (0.165)	FF (0.151)	KC (0.243)	CH (0.023)	KC (0.154)

	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9
Pitch-Type Freq.	FF (0.330)	SI (0.519)	FF (0.446)	SI (0.570)	FT (0.294)
Pitch-Type Freq.	CU (0.302)	CH (0.302)	CH (0.383)	SL (0.258)	FF (0.228)
Pitch-Type Freq.	CH (0.191)	SL (0.066)	SL (0.133)	FF (0.127)	CU (0.209)

	Cluster 10	Cluster 11	Cluster 12	Cluster 13
Pitch-Type Freq.	FT (0.382)	FF (0.560)	CU (0.264)	FT (0.580)
Pitch-Type Freq.	FF (0.290)	FS (0.165)	CH (0.142)	CH (0.212)
Pitch-Type Freq.	SL (0.220)	CU (1.141)	FC (0.137)	FC (0.098)

3.1.2 Defining Setup and Knockout Pitches with Graph Sinks

By interpreting the graph embeddings as forward dependencies, this analysis finds evidence for "setup" and "knockout" pitches in various pitch-type sequence clusters. Drawing from the concept of a "sink" in graph theory, it is possible to dissect each pitch sequence cluster in terms of its strategic components.

In operations research and graph theory, a sink is formally defined as a node which only has incoming flow; that is, multiple nodes direct to the sink but the sink does not direct itself to any other nodes in the system [9]. This definition can be strictly maintained for certain applications and contexts, such as when modeling fluids in a pipe system, currents in an electrical circuit, or even when approximating web page importance such as in the PageRank algorithm [10]. However, given the nature of baseball pitching, it is unlikely that pitchers reserve the use of a certain pitches only for the end or beginning of a sequence. This is counterintuitive to the incentive each pitcher has to minimize their predictability [3]. Instead, by loosening this formal definition of graph sinks to allow for limited non-zero outflow from sink nodes, it is possible to identify setup and knockout pitches in the context of pitch sequencing. Without loss of generality, each graph embedding *(A, B)*, for a pitch-type A and pitch-type B, provides information on the preferred ordering between two pitches. A high value in the feature *(A, B)* indicates that *A* is followed by a significant number of *B*s in the sequence. However, it is not guaranteed that this relationship holds for (B, A). It is possible that *B* is not followed by a significant number of *A*s, even if the converse is true. Specifically, the frequency of pitch-type *A* may not be dependent on pitch-type *B* to the extent that the frequency of pitch-type *B* is on pitch-type *A*.

In particular, a setup pitch is defined within each sequence cluster as a pitch whose forward dependencies are generally similar across all associated pitch-pairings and order permutations. Given this definition, setup pitches are thus more common relative to knockout pitches in sequence construction by design (See Table 4). A knockout pitch is defined within each sequence cluster as a pitch that follows many pitches but has limited corresponding outflow. Additionally, a knockout pitch could function as a local or universal sink within each sequence cluster graph; that is, a knockout pitch could have a large positive difference between its inflow and outflow for a *single* specific pitch (See Appendix: Table 4 – "Greatest Diff.") or have positive differences between inflow and outflow for *multiple* setup pitches. Within each cluster, both the setup and knockout pitches needed to meet a frequency threshold to be identified as such—this prevents pitches that rarely manifest within a sequence from being designated as either a setup or knockout pitch for the sequence. Given general definitional constraints, it is expected that there are far more setup pitches and far fewer knockout pitches. The nomenclature here is not intended to suggest that setup pitches are not thrown with the goal of achieving a favorable outcome (e.g., striking the batter out, recording an out) at that specific moment. Likewise, a knockout or terminal pitch is not an interpretation of a player's most effective pitches with respect to achieving those similarly favorable outcomes.

Table 4 displays the primary setup and knockout pitches that were identified upon inspection of forward dependencies. It is important to note that not all clusters have knockout pitches. Specifically, using the definition above, for these few clusters, there were no pitches that had a large enough positive difference between its inflow and outflow for a specific pitch or had consistently positive differences between inflow and outflow for multiple setup pitches. Instead, for clusters without a knockout pitch, the forward dependencies between pitch-types were approximately equal in magnitude irrespective of the exact direction between the two comparison features *(A, B)* and *(B, A).*

Table 4: *Setup and Knockout Pitches*

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Setup	FF, SI, CU	FF, SI	FC, KC	FF, SL	FF, FT, SI	FF, CU	SI, SL
Knockout	SL	N/A	FF	CH	N/A	N/A	CH
Greatest Diff.	(CU, SL)	(FF, FC)	(FC, FF)	(FF, CH)	(FF, FC)	(FF, SL)	(SI, CH)
Val of Diff.	(5026, 3974)	(1225, 1200)	(4750, 4041)	(3700, 0.05)	(1020, 930)	(816, 800)	(6100, 5070)

	Cluster 7	Cluster 8	Cluster 9	Cluster 10	Cluster 11	Cluster 12	Cluster 13
Setup	FF, SL, CU	FF, SI	FF, FT	FF, FT	FF, FC, CU	FF, FT, CU	FT, CH
Knockout	CH	FS, SL	N/A	SL	FS	CH	CH, FC
Greatest Diff.	(CU, CH)	(SI, SL)	(FF, FT)	(FT, SL)	(FF, FS)	(CU, CH)	(FT, CH)
Val of Diff.	(2120, 0.05)	(7200, 6850)	(2945, 2880)	(13000, 12400)	(3450, 2820)	(1545, 1380)	(5200, 4200)

Note: "Val. of Diff." reports the difference in the sum of forward dependencies for *(A, B)* and the sum of the forward dependencies for *(B, A)* within each cluster. Pitch B is considered a knockout pitch (i.e., local sink) given a relatively large positive difference between these values. Additionally, a pitch B can be considered a knockout pitch if there are multiple pitches A, C, D..., *n* such that the difference between $\sum_{i=A,i\neq B}^{n}(i,B) \gg \sum_{i=A,i\neq B}^{n}(B,i)$ is also true.

The magnitude of the difference between the forward dependencies from one direction to another also indicate that certain knockout pitches are "stronger" than other knockout pitches between clusters. As shown in cluster 3, there is a very extreme distinction between the sum of the forward dependencies in (FF, CH) versus the sum of the forward dependencies in (CH, FF). This implies that sequences that belong to cluster 3 rarely use changeups unless the changeup comes at the end of the sequence. In comparison, the greatest difference between forward dependencies for certain clusters such as cluster 12 is far lower than the difference that is displayed in cluster 3. This finding suggests that the changeup, the knockout pitch in cluster 12, is not exclusively used towards the end of a sequence. Across all sequence clusters, knockout pitches tend to be breaking-balls whereas setup pitches are largely fastballs or fastball variations. While this intuition is assumed as trivial domain knowledge for teams, coaches, and players, this quantitative assessment of setup and knockout pitches within the context of pitch sequences is likely novel.

3.1.3 Pitch-Type Sequencing in Context

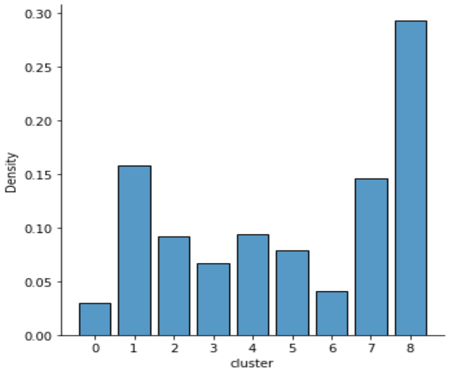
There are several sequencing strategies and patterns that are potentially observable from the model-results. For example, each sequence cluster involves a combination of fastballs and off-speed pitches, as highlighted above**.** It is perhaps significant to note that there are multiple sequences that share pitch-type constitution yet differ in ordering. This suggests that ordering is an important component of pitch-type sequences which alternatives such as simple pitch frequencies and pitch-to-pitch correlation do not otherwise capture. Since it is expected that pitchers throw pitch sequences that belong to different several clusters, it is feasible to analyze how sequence usage varies in multiple in-game contexts and matchups. Figure 5 & 6 (Appendix) displays the relative frequency of each cluster conditional on pitcher-batter handedness and starter/reliever role, as examples.

3.2.1 Zone Sequence Clusters

As with pitch-type sequencing, there are several universal zone sequencing patterns that were identified by model-based clustering. Pitchers tend to vary in how they set-up batters via their zone sequences, but their usage of these clusters is not uniformly distributed across gameplay situations (See Figure 7, Appendix: Figure 8 & 9).

Since players have all zones available to them (i.e., they are not inhibited by pitch arsenal), cluster popularity relates to which location strategies are preferred by MLB pitchers instead of physical or arsenal constraints. It is significant to note that the model-based clustering identifies fewer clusters for zone-based sequences than pitch-type sequences. Additionally, when compared between each other, zone-based clusters reflect the pitcher's desire to not leave pitches in the middle or high in the zone to prevent solid-contact and batted-balls that are hit in the air (Table 5). Instead, each cluster has a high representation of zones that are lower down in the strike zone. This ultimately supports a case that pitch-type sequencing is more complex in aggregate than zone sequencing.

Figure 7: *Zone Sequence Cluster by Relative Frequency/Density*



As with pitch-type sequence clustering, model-based clustering allows us to calculate a probabilistic distribution for each unique at-bat's assignment to a zone sequence cluster. Like pitch-type sequence clusters, the overwhelming majority of at-bat sequences (99.60%) are assigned to clusters with a high probability (greater than or equal to 99% assignment probability). The 0.40% of at-bats that are assigned to a zone cluster with less than an associated 99% probability all have an assignment probability greater than 50% and have a mean assignment probability of 88.99%. As with the pitch-type clusters, here is no explicit pattern as to which zone sequences seem difficult to assign to a cluster. These results indicate that the primary zone cluster assignment can be uniformly used given the high assignment probabilities.

Table 5: *Zone Cluster Membership*

Key
🟥 Upper Location
🟨 Middle Location
🟦 Lower Location

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Zone Freq.	🟦 **9** (0.194)	🟦 **14** (0.229)	🟦 **13** (0.187)	🟦 **13** (0.198)	🟦 **14** (0.248)
Zone Freq.	🟦 **14** (0.155)	🟥 **12** (0.198)	🟦 **14** (0.185)	🟦 **14** (0.128)	🟥 **12** (0.167)
Zone Freq.	🟦 **13** (0.096)	🟦 **13** (0.148)	🟦 **7** (0.174)	🟥 **11** (0.109)	🟨 **6** (0.103)
Zone Freq.	🟥 **11** (0.078)	🟥 **11** (0.137)	🟥 **11** (0.121)	🟨 **6** (0.102)	🟨 **5** (0.095)
Zone Freq.	🟨 **6** (0.074)	🟦 **9** (0.105)	🟥 **12** (0.108)	🟥 **12** (0.086)	🟥 **2** (0.083)

	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Zone Freq.	🟦 **13** (0.168)	🟥 **11** (0.269)	🟥 **11** (0.302)	🟦 **14** (0.347)
Zone Freq.	🟦 **14** (0.152)	🟦 **14** (0.220)	🟦 **13** (0.234)	🟦 **13** (0.256)
Zone Freq.	🟥 **12** (0.137)	🟥 **2** (0.095)	🟦 **14** (0.129)	🟦 **8** (0.061)
Zone Freq.	🟥 **11** (0.136)	🟥 **12** (0.083)	🟨 **4** (0.099)	🟨 **5** (0.048)
Zone Freq.	🟥 **1** (0.093)	🟥 **3** (0.072)	🟦 **8** (0.059)	🟦 **9** (0.044)

3.2.2 Zone Sequencing in Context

If zone sequencing strategies are relatively non-complex, then it is expected that the distribution of zone sequence clusters to be relatively consistent between pitchers and across matchups. In particular, split summary statistics can easily describe how the usage of zone sequence clusters differs against various right- and left-handed batters and between starters and relievers.

The two zone-based sequence clusters (clusters 4 and 6) that have a higher frequency of pitches up or in the middle of the strike zone tend to be thrown in similar contexts, as shown in Figure 8 & 9 (Appendix). Both cluster 4 and cluster 6 are thrown more by relievers than by starters in comparison to the other zone clusters, though the absolute difference in usage between starters and relievers is far less drastic than differences that are observed in usage of pitch-type clusters between starters and relievers. Additionally, both clusters 4 and 6 are thrown predominantly when pitchers share handedness with their batters.

3.3 Relationship between Pitch-Type and Zone Sequencing

The Chi-Square test of independence (*df* = 104) finds ample statistical evidence to suggest that the pitch-type and zone-clusters are not independent; that is, the value of one cluster variables implies a change in the expected probability distribution of the other ($p < 0.001$). Given that certain pitch-types coincide with location (e.g., breaking-balls are designed to be in lower zones), it is expected that clusters with high frequencies of these types of pitches tend to be associated with certain zone sequences. Thus, future analysis and work can attempt to re-cluster pitch labels (before sequence analysis) by including location as a pitch-level characteristic. In aggregate, players do not exhibit strong differences in zone sequencing strategies above what is expected by differences in their pitch arsenals. However, it is noteworthy *when* pitchers decide to deviate from their otherwise strong tendencies. This analytical framework allows teams to evaluate game- and matchup-level circumstances when they may anticipate shifts in pitch sequencing behavior.

3.4 Example Player Sequencing Comparisons

The clustering framework enables player comparison on the basis of pitch sequencing. While it is valuable to know what sequences a player is accustomed to throwing, it is increasingly important to have an understanding of how a player's sequencing behavior changes in various game contexts. Teams can achieve an edge in scouting, player development, and matchup preparation by using these tools to decrypt pitching patterns and investigate motivations behind these decisions.

Figure 10 displays side-to-side comparison of the pitch-type sequencing strategies used by 4 prominent MLB pitchers against left-handed and right-handed batters. There are several key takeaways that are observable from this small sample and are largely reflective of baseball intuition. Each pitcher relies on various clusters or pitch-type sequencing patterns. Pitchers are largely incentivized to remain unpredictable and variance in pitch-type sequencing is one component of their ability to deceive a batter. There are also commonalities among pitchers who show similarities in sequence pattern distribution. For example, starters (e.g., Blake Snell) tend to show more variance and matchup versatility in their pitch-type sequencing strategies than Chapman and Hader, both of whom are high-leverage relievers.

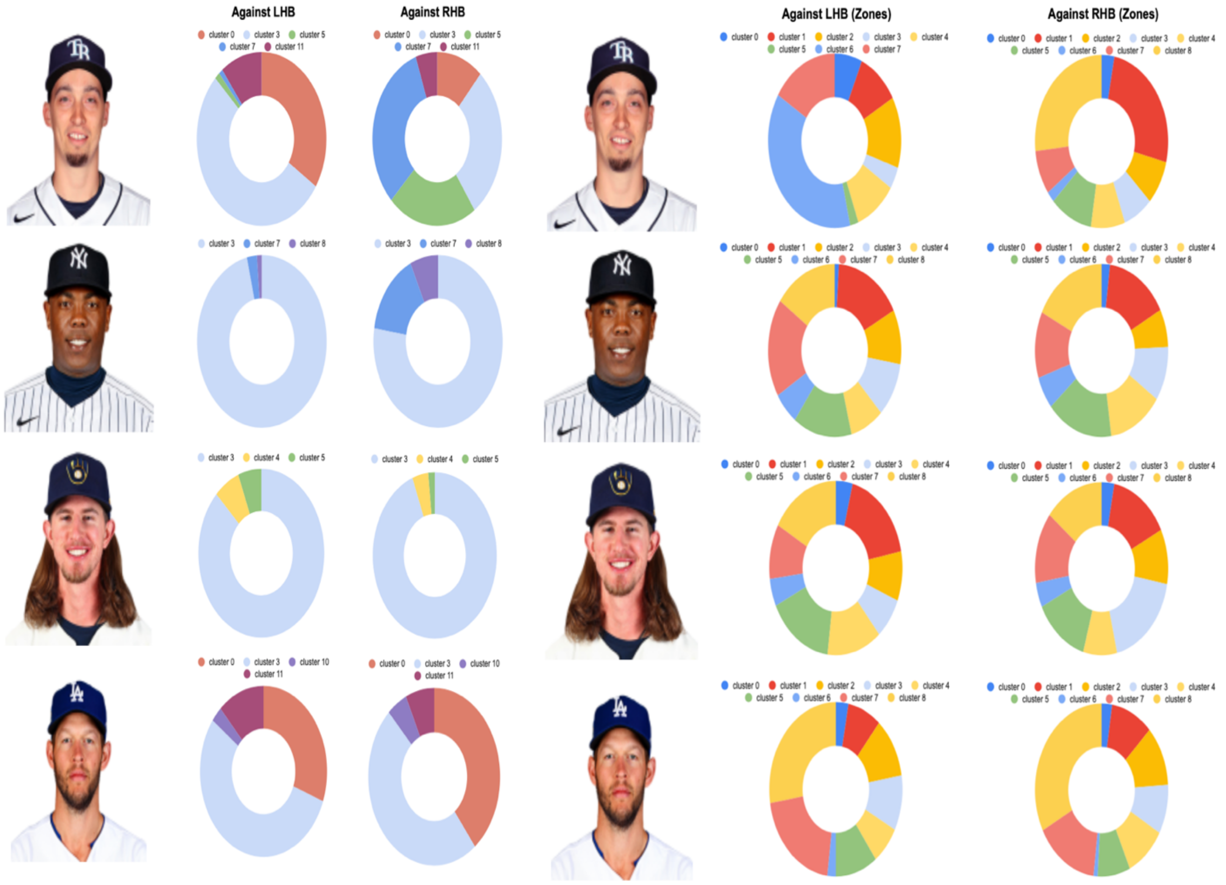
Using a pitcher's cluster distribution as a proxy the overall diversity of their sequencing strategies, a strong negative correlation between sequencing complexity and fastball velocity is detected. Intuitively, this suggests that players who throw hard can rely on their "stuff" more-so than players who do not have that advantage. Conversely, pitchers that cannot throw in high velocities need to take more diverse and complex approaches to their sequencing decisions. This finding is consistent with prior research on effective velocity and pitch sequencing strategy [11].

Players seem to pursue universal sequencing strategies with respect to zone selection as shown in Figure 11. Their zone sequence decision-making is highly correlated with their pitch arsenal. Unlike pitch-type selection where players need to draw from pitch-types in their pitch-arsenal, any zone is available to a pitcher at a given time, assuming that a pitcher has a general command over their pitches. Accordingly, each pitcher in this sample throws at least one sequence that belongs to each cluster.

As with pitch-type sequencing, each pitcher displays a unique assortment of patterns and behaviors in their zone sequencing strategies. Blake Snell (LHP), for example, takes divergent approaches when he pitches to left-handed and right-handed batters; specifically, he relies much more on sequence cluster 6 (characterized by pitches up in the zone) against left-handed batters and more on sequence cluster 1 (mixed locations) and sequence cluster 8 (low locations). This shift is not particularly observable in other pitchers and is visually apparent in Figure 6. Other pitchers, such as Aroldis Chapman, Josh Hader, and Clayton Kershaw, depend on very similar zone sequencing strategies against left-handed and right-handed pitchers. So, while zone sequencing strategies are less complex than pitch-type sequencing strategies in aggregate (See 3.2.1), players still exhibit significant differences in how they form strategies for locating their pitches.

Given that both pitch-type and zone sequences are clustered on an at-bat level, player-specific development and evolution across various time-spans (e.g., within a game or season) is observable. While some players' sequencing behaviors are relatively stable across the seasons they participated in, other players show evidence of shifts in how they approach sequencing overtime. This can be attributed to changes in matchup strategy, growth and development of pitch arsenals, or even injury mitigation and recovery processes. Conversely, pitchers do not seem to exhibit significant time-dependence in their zone sequencing strategies.

Figure 10 & 11: *Pitch-Type & Zone Sequencing Tendencies – Player Examples*



3.5 Aggregate Sequence-Based Player Similarity

After aggregating pitch-level data, pitch sequences are analyzed at the at-bat and player-career level. The at-bat embeddings are used to perform analysis on game-varying qualities of pitch sequencing; an example analysis that this paper explores is what pattern clusters are identifiable across all pitchers. The aggregate pitcher sequence embeddings will form the basis for player similarity search.

Player similarities are immensely useful to teams, scouts, and fans, and have a long history in baseball analytics dating back to Bill James' similarity scores [11]. Since player similarity enhances our contextual understanding of a player's tendencies and behavior, it is valuable to determine player similarity as measured through sequencing decisions that players have made during in-game situations. In the same way that a DNA sequence defines the biological makeup of an organism, a pitch sequence—when aggregated across a player's career—summarizes the cumulative decisions that the pitcher has previously made. After embedding each player-aggregate sequence, player similarities are computed with the dot product of the query embedding with other embeddings in our player sequence database [5]. For each player sequence query, the player sequencing embedding that produces the largest dot product with the query will be taken as the most similar sequence. When using raw embedding outputs, this similarity method takes into account diversity in pitch arsenals since pitch-types that are not seen in a specific players sequence will have a forward dependency that is initialized with a zero-value. Using the dot product, pitchers with diverse pitch arsenals (i.e., many pitch-types they can throw) tend to appear frequently as similarity matches since they have the last number of forward dependencies initialized as zero. Thus, it is important to note that while these similarity matches are correlated with player pitch arsenals, sequence-based similarity also takes into the order in which pitches are thrown.

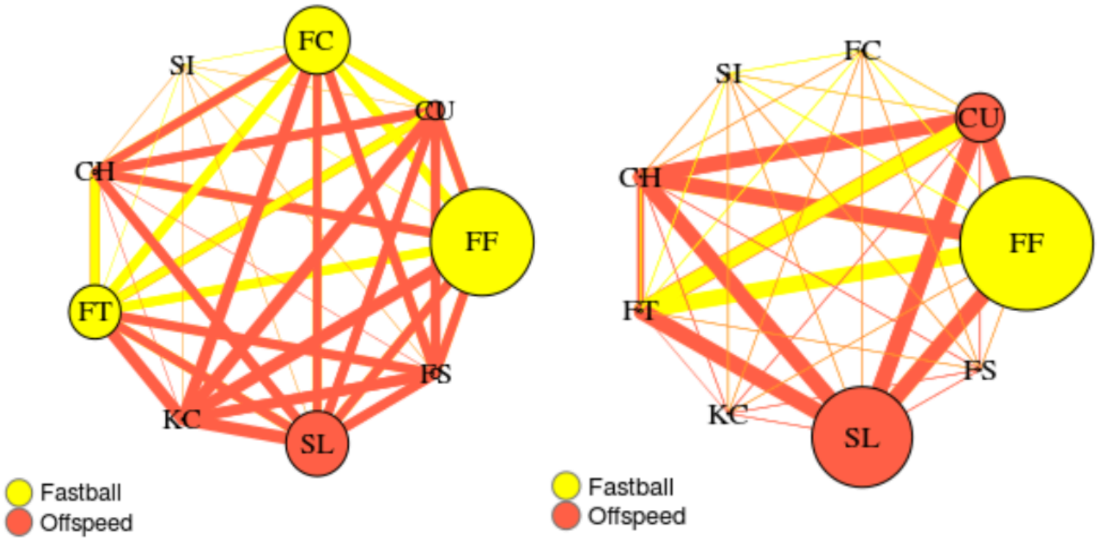
Sequence-based player similarities are used to begin to explore the relationship between player value and sequencing. Drawing from a player similarity function that outputs the closest player for each player input, it is evident that similarity in pitch-type sequencing does not correlate to either simple (e.g., K%, ERA, H/9) or advanced metrics of player performance (e.g., FIP, xFIP, tERA) above what is achieved from random assignment. While there are more extensive statistical models that can test this hypothesis further, players who manifest similar sequencing strategies seem to vary quite drastically in their performance. This signals that sequencing alone does not predict a player's performance or compensate for a lack of pitching "stuff."

It is also important to acknowledge that the approach described in this paper should not be interpreted as a means to assign value to pitch sequences or sequencing decisions. In order to define sequencing skill precisely, researchers should strive to separate the effect and value of individual pitches from the added benefit of a specific ordering. Instead, this research is instrumental in providing a framework to decode the sequencing strategies that pitchers consistently utilize for the purposes of in-game strategy. Instead of applying pitch sequencing similarities for player value predictions, player sequencing similarities can be utilized to assist teams in matchup preparation and pitchers in their developmental goals.

From this graph embedding approach, a pitcher's aggregate sequencing decisions can be represented in total by a directed graph or network. Each graph representation can easily be presented as a visualization (See Figure 12 & 13) that characterizes a pitcher's sequencing patterns across multiple seasons (2016-2019). This visualization can capture the general complexity of each pitcher's pitching style for a variety of research and development purposes. The relative size of each node corresponds to the relative frequency of each pitch-type whereas the relative thickness (i.e., weight) of each edge corresponds to the forward dependency between two pitch-types.

In order to maintain visual consistency across all graphs, each major pitch-type is included in a pitcher's graph visualization irrespective of whether that pitcher has that pitch in their arsenal. Each link is colored according to the direct of the originating node. The stronger association is laid over the weaker association directionally. The relative size of each node denotes the existence of that pitch in a specific pitcher's arsenal. This consistency in the visualization format allows for a qualitative side-to-side comparison of player network graphs, though the design/aesthetic features are customizable by the user. The data that underlies each network visualization can be limited to certain in-game scenarios or matchups to illustrate situational decision-making.

Figure 12 & 13: *Sequence Graph Networks for Blake Snell and Clayton Kershaw (2016-2019)*



4 Related Applications & Future Work

In addition to developing a novel analytical framework to study pitch sequencing strategies, this paper presents multiple tools that MLB teams can use in matchup preparation, scouting, and player development. Since pitch sequencing is otherwise difficult to encode for using traditional statistical methods, teams can take immediate steps in applying this graph embedding approach to study pitcher styles and behavior. There are many questions to be asked about the motivations surrounding usage of certain pitch sequences. This framework enables researchers to reproduce this approach in their analysis of pitcher behavior. Furthermore, graph network visualizations (as shown in Figure 12 & 13) that are based upon player-aggregate sequence data can be helpful in communicating a pitcher's sequencing profile in an interpretable manner. When mapped in a time-series, teams can observe how sequence networks change overtime and how players are developing and modifying their existing strategies. This tool can supplement advance scouting preparation against other teams, especially after conditionalizing player data based on certain in-game and matchup scenarios. Additionally, teams can consider how a pitcher's sequencing changes in response to their previous outcomes in an at-bat or game. There is a strong strategic component to this work as well—future work may consider testing whether pitch sequence clusters are predictive of player decisions, which would imply that pitchers are victims to information leakage at the sequence level.

Finally, steps are taken to develop a novel open-source aggregation database that allows independent researchers and MLB teams to perform pitch sequence analysis based on this research. Functionality of this toolkit includes searching for specific pitch sequences that have occurred over a multi-year span, ability to visualize networks and download pitch sequencing graph embeddings for specific pitchers, and access to drop-down comparisons and player-similarity matching.

References

[1] Bock, J. (2015). Pitch Sequence Complexity and Long-Term Pitcher Performance. *Sports*, 40-55.

[2] Sharpe, S. (2020). MLB Pitch Classification. https://technology.mlblogs.com/mlb-pitch-classification-64a1e32ee079.

[3] Glenn, H., & Zhao S. (2017). Using Pitchf/x to model the dependence of strikeout rate on the predictability of pitch sequences. https://content.iospress.com/articles/journal-of-sports-analytics/jsa103.

[4] Zhan, J., Gerstner, L., & Polimeni, J. (2020). Measuring the Impact of Robotic Umpires. https://global-uploads.webflow.com/5f1af76ed86d6771ad48324b/5f6a65851d1ac98081a707f0_Zhan_MeasurinM-the-impact-of-robotic-umpires.pdf.

[5] Ranjan, C., Ebrahimi S., & Paynabar, K. (2016). Sequence Graph Transform (SGT): A Feature Embedding Function for Data Mining. arXiv:1608.03533v13.

[6] Manning, C., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. ISBN: 0521865719.

[7] Srihari, S. (2019). Mixtures of Gaussians. https://cedar.buffalo.edu/~srihari/CSE574/Chap9/Ch9.2-MixturesofGaussians.pdf.

[8] Carrasco, O. (2019). Gaussian Mixture Models Explained. Towards Data Science. towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95.

[9] Brossard, E. (2010). Graph Theory: Network Flow. University of Washington. https://sites.math.washington.edu/~morrow/336_10/papers/elliott.pdf.

[10] Agarwal, B., & Khan, M. H. (2013). Analysis of Rank Sink Problem in PageRank Algorithm. International Journal of Scientific & Engineering Research. https://www.ijser.org/researchpaper/Analysis-of-Rank-Sink-Problem-in-PageRank-Algorithm.pdf.

[11] Driveline Baseball. (2019). Calling the Right Pitch: Investigating Effective Velocity at the MLB Level. https://www.drivelinebaseball.com/2019/05/calling-right-pitch-investigating-effective-velocity-mlb-level/.

[12] Baseball Reference. (2020). Similarity Scores. https://www.baseball-reference.com/about/similarity.shtml.

Appendix

Table 2: *Order Comparison by Cluster*

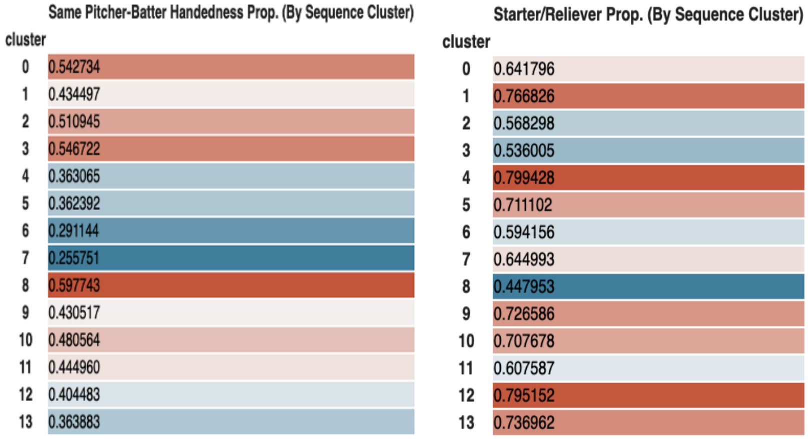
	Cluster 3	Cluster 7	Cluster 10
Ex. 1	SL, SL, FF, CH	FF, FF, CH, FF, CH, CH	FF, FF, FT, FT, CH, FT
Ex. 2	SL, FF, SL, FF, CH	FF, FF, CH, CH, FF	FT, FF, SL, FT
Ex. 3	FF, FF, CH, CH	FF, FF, CH, CU, FF	SL, FT, FF, SL, FT
Ex. 4	SL, FF, CH, CH	FF, SL, FF, FF, CH	FT, FF, FT, FT, SL
Ex. 5	FF, SL, CH, CH	FF, CU, FF, CH	FT, FF, FF, SL, SL

Table 3: *Cluster Average Entropy*

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Entropy	0.77	1.05	0.59	0.25	1.18	0.98	0.78

	Cluster 7	Cluster 8	Cluster 9	Cluster 10	Cluster 11	Cluster 12	Cluster 13
Entropy	0.74	0.51	1.04	0.88	0.66	1.25	0.52

Figure 5 & 6: *Pitch-Type Sequence Cluster Splits by Matchup Type & Role*



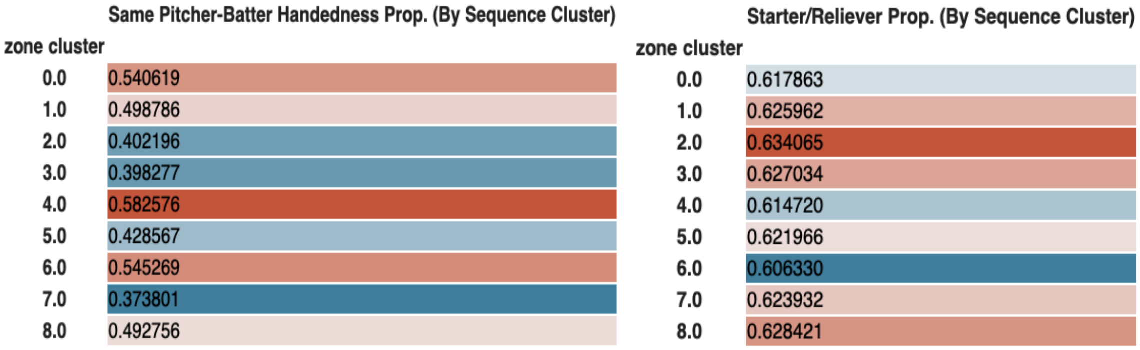
Same Pitcher-Batter Handedness Prop. (By Sequence Cluster)

cluster	
0	0.542734
1	0.434497
2	0.510945
3	0.546722
4	0.363065
5	0.362392
6	0.291144
7	0.255751
8	0.597743
9	0.430517
10	0.480564
11	0.444960
12	0.404483
13	0.363883

Starter/Reliever Prop. (By Sequence Cluster)

cluster	
0	0.641796
1	0.766826
2	0.568298
3	0.536005
4	0.799428
5	0.711102
6	0.594156
7	0.644993
8	0.447953
9	0.726586
10	0.707678
11	0.607587
12	0.795152
13	0.736962

Figure 8 & 9: *Zone Sequence Cluster by Relative Frequency/Density*



Same Pitcher-Batter Handedness Prop. (By Sequence Cluster)

zone cluster	
0.0	0.540619
1.0	0.498786
2.0	0.402196
3.0	0.398277
4.0	0.582576
5.0	0.428567
6.0	0.545269
7.0	0.373801
8.0	0.492756

Starter/Reliever Prop. (By Sequence Cluster)

zone cluster	
0.0	0.617863
1.0	0.625962
2.0	0.634065
3.0	0.627034
4.0	0.614720
5.0	0.621966
6.0	0.606330
7.0	0.623932
8.0	0.628421