Stockholms universitet

Detecting tactical patterns in football with fast search and density peak clustering

Michael Ståhle

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2025:2
Matematisk statistik
Februari 2025

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Mathematical Statistics
Stockholm University
Master Thesis **2025:2**
http://www.math.su.se

Detecting tactical patterns in football with fast search and density peak clustering

Michael Ståhle*

February 2025

Abstract

Sports analytics has seen rapid growth, supported by advancements in data acquisition and computational methodologies. This thesis focuses on clustering football possession chains using on-ball event data to detect and interpret tactical patterns. We adopt the Fast Search and Density Peaks (FSDP) algorithm (a distribution-free, density-based clustering method) to group similar ball trajectories. Two distance measures, the Fréchet distance and the Longest Common Subsequence (LCSS) distance, are assessed based on their theoretical properties and alignment with the assumptions of the FSDP algorithm. Our empirical analysis demonstrates that the Fréchet distance, which is metric-based and sensitive to continuous shape variations, provides more coherent clustering results than the LCSS distance. Using the Fréchet distance, four distinct tactical patterns emerge from the possession data of Arsenal F.C. during the 2017/18 Premier League season. Notably, attacks along the left flank produce nearly twice as many goals as those on the right flank, suggesting a clear advantage when exploiting that side of the pitch. These findings highlight the importance of selecting distance measures that both adhere to metric properties and capture the nuances of the underlying data structure. They also illustrate the potential of density-based methods to uncover meaningful tactical insights, paving the way for further methodological refinements (such as alternative distance measures, fuzzy cluster assignment, and inclusion of player tracking data) that could deepen our understanding of football strategy.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: michael.staahle@gmail.com. Supervisor: Chun-Biu Li.

Contents

1. Introduction

The field of sports analysis has grown significantly in recent years, primarily because of new advancements in video-tracking technologies. [6] Trajectory data of players and the ball are recorded with high precision, enabling spatio-temporal analysis of the game at a large scale. There are typically two types of data in sports analysis: the previously mentioned trajectories and event logs, where event logs consist of sequentially observed events marked with a timestamp and a position. Examples of events include passes or tackles.

This type of data is collected for many sports, but this thesis will focus on football and more specifically possession chains. Football is categorized as an invasion sport, where two opposing teams compete in a constrained area. They score by putting the ball in the goal of the other team while defending their own goal. The winning team is the one who scores the most goals. Football is by far the most popular sport in the world, with more than 3.5 billion viewers during the World Cup of 2018. According to a report by Deloitte, the market size of the European football leagues during the 2015/2016 season was estimated at 24.6 billion euros [8].

There are many other areas that generate both trajectory and event data. Traffic and route planning is a good example of another research area where spatial trajectories are naturally of interest. More generally, sequential events are a very common structure in customer data for many businesses, such as e-commerce companies and subscription businesses. Besides the fact that football engages more than half of the world's population, the many synergies with other research and business areas highly motivate further research in sports analysis.

Despite the large increase in both quality and quantity of data, the unstructured nature of football makes it difficult to analyze. There are usually 11 players in each team simultaneously on the field and all their movements have a significant impact on the game at any point in time. The movements are also highly dependent, both within the team and between teams. There are also a large number of other factors, such as the mental stability of individual players or weather conditions, that have a significant impact on the game. There has been extensive research in the area, and the number of published papers has drastically increased over the last 20 years. Gudmundsson and Horton made a survey [4] in 2016 with the purpose of providing an overview of the research conducted in the area. Figure 1.1 is based on a similar graph presented in [4] showing the number of papers included in the survey by the year they were published.

The following criteria were used when including papers in the survey:

1. Only invasion sports are considered.
2. The models in the research take spatio-temporal data as primary input.
3. The models performs some non-trivial computation on the spatio-temporal data.

Several research papers are presented in [4], especially interesting for this thesis are the papers in section 2.3.4–Identifying Plays and Tactical Group Movement. Borrie et al. [3] subdivide the playing area into zones and identify passes that start and end in the

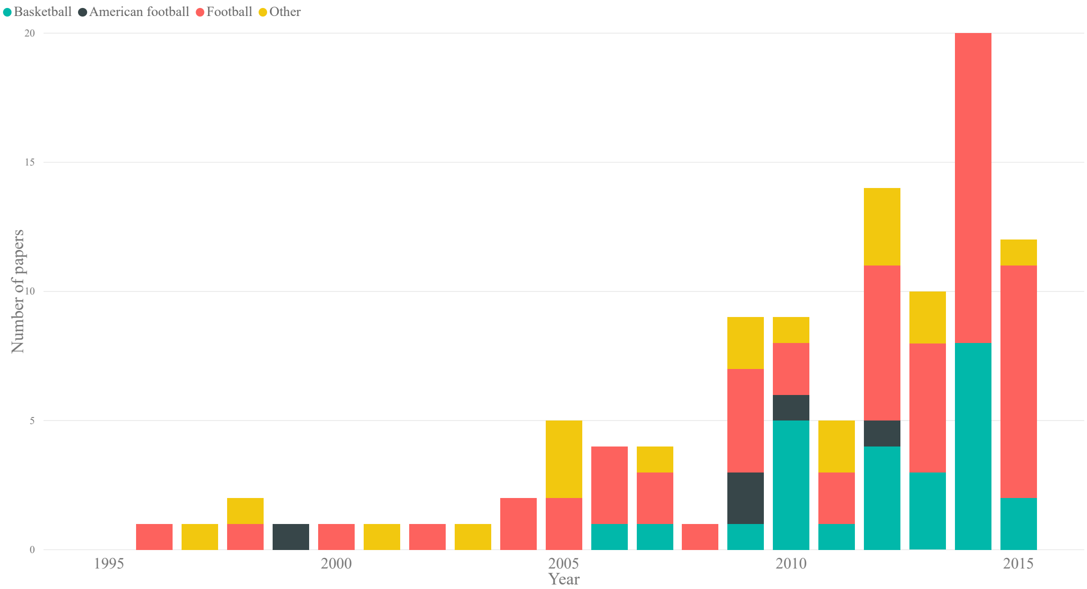


Figure 1.1: The number of papers included in the survey by the year they were published. Colors represent the associated sport, as presented in the legend. Football is represented by the red color.

same zones. They also consider the time elapsed between passes and can thus identify frequently occurring passing sequences.

Wang et al. [14] show an unsupervised approach for identifying tactical plays from passing sequences using event logs from games. They base their model on the same theory used in topic modeling of text corpora. The authors describe their novel model T^3M as an extension of the classical topic model Latent Dirichlet Allocation (LDA) [2]. Similar to Wang et al, this thesis will consider events as ordered tuples of their coordinates. By dividing the data into possession chains and extracting the ball trajectory represented by the sequences of ordered events, this thesis examines two well known distance measures for trajectories and their impact when used as input to the clustering algorithm: Fast Search and Density Peaks by Rodriguez and Laio [9].

The outline of this thesis is as follows: Chapter 2 will cover all methodology used in this thesis, with a focus on cluster analysis in general and Fast Search and Density Peaks clustering specifically. The distance measures will be presented in detail, along with their characteristics and motivation.

Chapter 3 will cover the results, evaluating the resulting clusters with respect to the methods defined in Chapter 2. The final results will be visualized over the football field, and we will infer which tactics yield the most goals.

Chapter 4 will cover the general discussion of the methods and the results. Mathematical proofs, definitions, and implementation details will be available in the appendices.

2. Cluster analysis

Machine learning consists of three primary categories: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, data is labeled, allowing for a distinction between target and input variables; this category typically includes classification and regression tasks. In unsupervised learning, data is unlabeled, with no distinction between target and input variables, often described as a pattern recognition problem. Reinforcement learning, the third category, involves an agent learning to make decisions by interacting with an environment to maximize a cumulative reward, commonly applied in fields such as robotics, game playing, and autonomous systems.

Clustering methodology belongs to the field of unsupervised learning since the goal is to group unlabeled data based on similarities in the input space. There are many clustering methods to choose from, and the performance is heavily dependent on the data and the intended use of the results. Perhaps the most commonly known clustering algorithm is the centroid-based model, K-means.

K-means clustering is a popular choice because of its simplicity and close conceptual relationship to the K-nearest neighbor classifier. Given a set of data $x_1,\dots,x_N$ where x_n is a vector in a D-dimensional space, the goal is to group the data into K clusters such that every element belongs to the cluster with the nearest mean.

Define the set of mean vectors $\mu_1,\dots,\mu_K$ as points residing in the same D-dimensional space as x_n. We now define a binary variable r_{nk} for each x_n, indicating if data point x_n is assigned to cluster k; that is, if x_n is assigned to cluster k, then $r_{nk}=1$ and $r_{nj}=0$ for $j\neq k$. The sum of squared distances from each point to its mean vector is then given by

$$J=\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\left\|x_n-\mu_k\right\|^2 \tag{2.1}$$

The problem is then to minimize J with respect to r_{nk} and μ_k. The process is conducted in two steps. Given a predefined K and randomly selected initial values for μ_k we have

1. Assign each point to its nearest mean-vector, i.e.

$$r_{nk}=\begin{cases}1, & \text{if } k=\arg\min_j \left\|x_n-\mu_j\right\|^2,\\ 0, & \text{otherwise.}\end{cases} \tag{2.2}$$

2. Calculate the new mean-vector for each cluster, i.e.

$$\mu_k=\frac{\sum_n r_{nk}x_n}{\sum_n r_{nk}}$$

By repeating steps 1 and 2 iteratively, the algorithm will converge; however, it does not necessarily find the global minimum, which may produce misleading results. There are several other drawbacks in the standard K-means model. The most notable may be the requirement of a predefined number of clusters K, the assumption that clusters are spherical and of similar size, and the need for the input space to be Euclidean. This Euclidean space requirement limits K-means' effectiveness in cases where the data lies in a non-Euclidean space or requires a more flexible distance measure.

The K-means model is part of the Centroid-based cluster family. Besides Centroid-based models there are mainly two other cluster family's. Distribution-based and Density-based clustering.

2.1 Fast Search and Density Peaks clustering

Fast Search and Density Peaks (FSDP) is part of the density-based clustering family. In contrast, distribution-based models partition the data into a predefined number of k clusters, where the underlying distribution $p(x)$ is assumed to be a mixture of k densities $p_i(x)$ [7]. An example of such a model is the Gaussian Mixture Model (GMM). This model faces similar problems as the K-means model, where the underlying density assumptions result in convex or elliptical clusters and the necessity of a predefined number of clusters k.

Density-based clustering makes no such assumptions about the underlying distribution of the data $p(x)$. Clusters are defined as regions with higher density compared to other regions of the data. It allows for the findings of non-elliptic clusters. In comparison to the more common clustering algorithms such as K-means or mixture of Gaussian (EM algorithm), density-based clustering often has a built-in decision process for choosing the number of cluster centers based on the data. The FSDP algorithm has a unique visualization of such a process, allowing the user to interact with the algorithm for deciding on the number of clusters.

The algorithm is based on the assumption that cluster centers are surrounded by neighboring points with lower local density. Points with higher local density are relatively farther away from these centers; that is, clusters are defined by areas of relatively high density that are distant from other high-density areas. The distance between points is the only input data that the algorithm requires. In Section 2.4, we will see how the choice of an appropriate distance measure is crucial for meaningful clustering, but the following explanation of the algorithm assumes that the distances are known.

2.1.1 Rho and Delta

Given the distances between all the points in the data set, the clustering algorithm is governed by two quantities for each data point i: ρ_i (rho) and δ_i (delta), where ρ_i is the local density of the point, and δ_i is the distance to the closest point with higher density. In the original article [9] by Rodriguez and Laio, ρ_i is defined as

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{2.3}$$

where $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ otherwise and d_c is a cutoff distance. We extend the original definition of ρ_i, and instead of counting the discrete number of points that has a lower distance than the cutoff distance to point i, we introduce a Gaussian kernel as follows

$$\chi(d_{ij}, d_c) = \exp\left(-\left(\frac{d_{ij}}{d_c}\right)^2\right) \tag{2.4}$$

which makes the quantity of ρ_i continues and ensures that no points will have the same density.

The definition of δ_i follows directly from Rodriguez and Laio, i.e. for point i, δ_i is defined as the distance to the closest point j where j has a higher density than i.

$$\delta_i = \min_{j:\rho_j > \rho_i} d_{ij} \tag{2.5}$$

For the point with the highest density, δ_i is set to $max_j(d_{ij})$. This ensures that the point with highest density is a cluster center.

2.1.2 Decision graph

Rho and delta are calculated for each point in the data and then plotted against each other. This is what Rodriguez and Laio [9] calls a decision graph. The decision graph is illustrated in Figure 2.1.

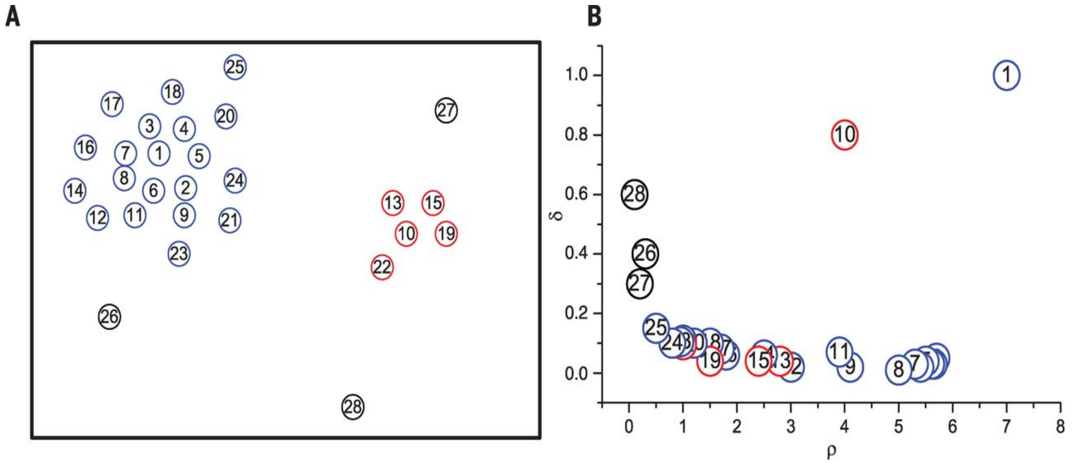


Figure 2.1: Example of a decision graph (B) from a sample of points (A). The picture is from the original article by Rodriguez and Laio.

The example in Figure 2.1 clearly shows how the cluster centers are distinguished in the right plot by their relatively large ρ and δ values. We note that there are many points with similar density as point 10, but what distinguishes it as a cluster center is its relatively large distance to the nearest point with higher density, i.e., its δ value. Since points 7 and 8 have a very short distance to a point with higher density, they should be part of that point's cluster rather than being cluster centers themselves. This intuitive reasoning strengthens the results shown in Figure 2.1.

After the cluster centers has been chosen, the rest of the points are assigned to clusters in descending order, starting with the point with highest density. The following rule is applied during the cluster assignation: If the point x_i is not a cluster center, then x_i is assigned to the same cluster as its nearest neighbor with higher density.

2.1.3 Halo assignment

Notice in Figure 2.1 how the points 26, 27 and 28 does not belong to any cluster. That is what the authors call a halo, i.e. when it's not clear if a point is part of any chosen cluster center. The motive of the halo construction is to infuse some uncertainty in the cluster process. We want to distinguish between points that is more certain to be part of a cluster from points that we are not so certain about.

First a set of boarder points are selected for each cluster. They are defined as the set of points belonging to the cluster but being within a distance d_c of another cluster. Then we find the point with the highest density within the boarder region. The density is called ρ_b and any point with higher density in that cluster will be considered to be a part of the cluster core and all other points will be part of the cluster halo.

2.2 Cluster Evaluation

Our evaluation focuses on utilizing several tools: the decision graph, cluster assignment with consideration of core and halo points, the discriminatory power of evenly distributed clusters, and silhouette analysis. These tools help us assess the quality of the clusters and determine the most appropriate number of clusters and their centers.

2.2.1 Appearance of the Decision Graph

The decision graph is a crucial tool for evaluating clustering results as it visually represents the relationship between the local density (ρ) of data points and their minimum distance (δ) to higher-density points. By plotting δ versus ρ, we can identify potential cluster centers, which are characterized by having both high local density and high minimum distance to other high-density points.

In our analysis, we generate decision graphs for varying values of the cutoff distance d_c, which influences the computation of local densities. By observing how the distribution of ρ and δ values changes with different d_c, we can assess the stability and robustness of potential cluster centers.

The decision graph allows us to:

- Identify distinct points with relatively high ρ and δ, which are potential cluster centers.
- Observe the concentration and distribution of data points in the ρ-δ space, providing insights into the underlying data structure.
- Assess how changes in d_c affect the clustering, helping us choose an appropriate value for further analysis.

By analyzing the decision graphs, we can evaluate whether the data naturally forms clusters and how well-separated these clusters are.

2.2.2 Cluster Assignment

Cluster assignment involves assigning each data point to a cluster based on its proximity and density relative to the identified cluster centers. In our evaluation, we focus on how different selections of cluster centers impact the overall clustering results.

We use the product $\delta \times \rho$ as a criterion to rank potential cluster centers. By plotting $\delta \times \rho$ in descending order, we can detect a clear separation between points with significantly higher values and the rest of the data. This separation helps us decide on the number of clusters by identifying the top points that stand out as cluster centers.

Our approach includes:

- Selecting cluster centers based on the highest $\delta \times \rho$ values.
- Exploring different combinations of cluster centers to see how they affect the clustering outcome.
- Considering permutations of the top k points with the highest $\delta \times \rho$ to find the combination that yields the best clustering performance.

By analyzing the cluster assignments resulting from different sets of cluster centers, we can evaluate the impact on cluster composition, such as the number of core and halo points, and determine which configuration provides the most meaningful clustering of the data.

Core vs. Halo

In our evaluation, we distinguish between core points and halo points within clusters to gain a deeper understanding of the cluster structures and their quality.

Core Points: These are points with high local density and are located close to the cluster center. They represent the most central and densely populated regions of the cluster.

Halo Points: These are points with lower local density, situated towards the outer regions of the cluster. They may be near the borders between clusters and are more susceptible to overlap with other clusters.

By analyzing the proportion of core to halo points in each cluster, we can:

- Assess the compactness and cohesiveness of clusters. A higher proportion of core points indicates a well-defined cluster.
- Identify potential overlaps between clusters, as a high number of halo points suggests that clusters may not be well-separated. Note that some halo points may represent outliers or background noise that do not necessarily indicate cluster overlap; these halo points are characterized by having a low ρ and a relatively high δ, which makes them distinguishable in the decision graph.
- Decide whether to consider only core points in further analysis to improve the reliability of the clustering results.

In our analysis, we focus on core points as successful assignments and may disregard halo points in certain evaluations, such as when calculating silhouette scores, to minimize the impact of overlapping regions and enhance the clarity of the clustering outcome.

Preference for Even Distribution Across Clusters

In our clustering analysis, the selection of cluster centers is a critical step that influences the overall partitioning of possession chains into tactical game plays based on ball trajectories. The algorithm by Rodríguez and Laio [9] involves a degree of subjectivity in choosing cluster centers from the decision graph, and this choice can affect the balance and interpretability of the resulting clusters.

While we do not impose a strict condition that clusters must have equal numbers of points, we consider an even distribution across clusters to be a desirable outcome when selecting among multiple valid clustering options. This preference arises when the data allows for different configurations that are otherwise equivalent in representing the underlying structure.

The rationale for favoring an even distribution is grounded in enhancing the discriminative power of the clustering. Mathematically, the sum of the squares of the cluster sizes, $\sum_{k=1}^{K} n_k^2$, is minimized when clusters are of equal size. Minimizing this sum maximizes the expected number of points in other clusters for any randomly selected data point. This means that each trajectory has more other trajectories to distinguish itself from, leading to more meaningful and interpretable clustering results.

Importantly, this preference does not override the natural tendencies of the data. If the possession chains inherently form imbalanced clusters due to the tactical patterns present in the game, our method accommodates these imbalances. However, when faced with a choice—such as deciding between two or three potential cluster centers where the optimal number is two—we prefer the selection that results in a more even partition of the data, provided that all other factors remain equal.

By incorporating this preference into our iterative evaluation process, which includes tools like the decision graph, core and halo point analysis, and silhouette analysis, we aim to strengthen our analysis and improve the final clustering outcome. This approach helps ensure that the clusters formed are not only statistically sound but also meaningful in the context of tactical game analysis.

See Appendix B.2 for the detailed proof supporting this preference.

2.2.3 Silhouette Analysis

Silhouette analysis is a method used to evaluate the quality of clustering by measuring how well each data point fits within its assigned cluster compared to other clusters. It provides a graphical representation and a quantitative measure called the **silhouette score**, which combines both cohesion (how close a point is to other points in the same cluster) and separation (how far a point is from points in other clusters).

For each data point i, the silhouette score $s(i)$ is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{2.6}$$

where:

- $a(i)$ is the **cohesion**, the average distance between point i and all other points in the same cluster.

- $b(i)$ is the **separation**, the minimum average distance between point i and all points in any other cluster, computed over all other clusters.

$a(i)$ and $b(j)$ can be expressed as follows:

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, j \neq i} d_{ij}, \tag{2.7}$$

$$b(i) = \min_{C \neq C_i} \left(\frac{1}{|C|} \sum_{j \in C} d_{ij} \right), \tag{2.8}$$

where:

- C_i is the cluster containing point i.
- $|C_i|$ is the number of points in cluster C_i.
- C represents each cluster other than C_i.

The silhouette score $s(i)$ ranges from -1 to 1:

- $s(i) \approx 1$: Point i is well matched to its own cluster and is clearly separated from other clusters.
- $s(i) \approx 0$: Point i lies on the boundary between its own cluster and other neighboring clusters.
- $s(i) \approx -1$: Point i may have been assigned to the wrong cluster.

The average silhouette score for all data points provides an overall assessment of the clustering quality:

$$\bar{s} = \frac{1}{N} \sum_{i=1}^{N} s(i), \tag{2.9}$$

where N is the total number of data points.

In our analysis, we use silhouette analysis to determine the optimal number of clusters and to evaluate the quality of the clustering results obtained with different cluster center selections.

Plotting Silhouette Score Against Number of Centers

We compute the average silhouette score $\bar{s}$ for clustering results with varying numbers of cluster centers K. By plotting $\bar{s}$ against K, we identify the number of clusters that yields the highest silhouette score, indicating the most appropriate clustering configuration.

The steps involved are:

1. For each potential number of clusters K, select K cluster centers based on the highest $\delta \times \rho$ values from the decision graph.
2. Assign data points to clusters following Fast Search and Density Peaks algorithm.
3. Calculate the average silhouette score $\bar{s}$ for all data points.
4. Plot $\bar{s}$ against K to visualize how the clustering quality changes with different numbers of clusters.

By analyzing this plot, we can observe how the average silhouette score varies and select the number of clusters that maximizes $\bar{s}$, indicating optimal clustering.

Silhouette Plot

In addition to plotting the average silhouette score against the number of clusters, we generate silhouette plots for specific clustering configurations to gain deeper insights into the clustering structure.

A silhouette plot displays:

- The silhouette score $s(i)$ for each data point, sorted within each cluster.
- The clusters are differentiated by different colors.
- A vertical dashed line indicating the average silhouette score across all data points.

Interpreting the silhouette plot involves:

1. **Assessing individual clusters**: Clusters with higher $s(i)$ (above average) values indicate well-formed clusters with high cohesion and separation.
2. **Visualizing the distribution of points across clusters**: The relative thickness of the silhouettes represents the distribution. If the silhouettes are evenly sized, the points are evenly distributed across the clusters.
3. **Identifying misclassified points**: Data points with negative $s(i)$ values may be misclassified and could belong to a different cluster.

By incorporating silhouette analysis into our methodology, we enhance the robustness of our clustering evaluation as it provides an objective measure to compare different clustering configurations.

Illustrative Example:

To illustrate how to interpret a silhouette plot, consider the example shown in Figure 2.2. In this example, we have generated a synthetic dataset in euclidean space and applied the K-means algorithm with 3 centers.

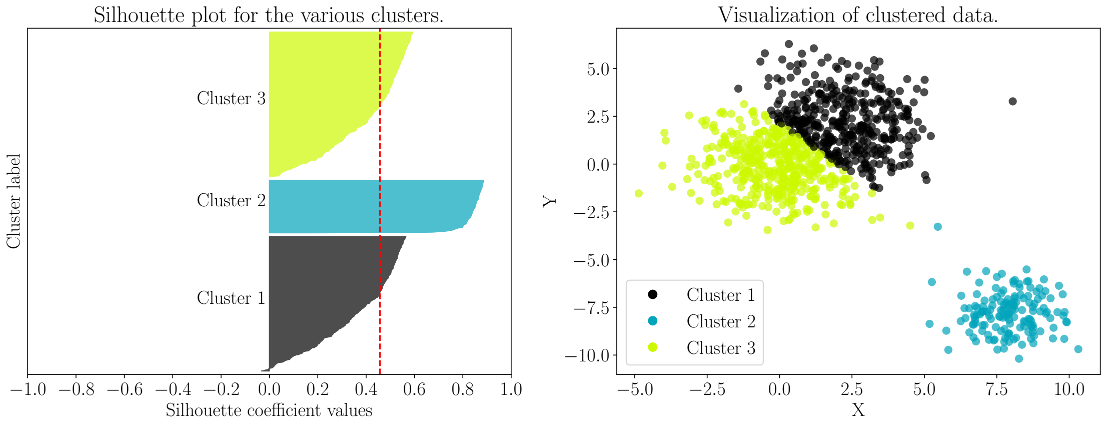


Figure 2.2: Silhouette plot and scatter plot illustrating two larger overlapping clusters (Clusters 1 and 2) and one smaller, well-separated cluster (Cluster 3).

In the silhouette plot (left side of Figure 2.2), each color represents a different cluster, and the silhouette scores $s(i)$ for all data points are displayed. The vertical dashed line indicates the average silhouette score across all data points.

In this example:

- **Assessing individual clusters**: Cluster 3 exhibits higher silhouette scores, indicating it is a well-defined cluster with high cohesion and separation from the other clusters. Clusters 1 and 2 have lower silhouette scores due to overlap, suggesting they are less separated.
- **Visualizing the distribution of points across clusters**: The thickness of the silhouettes corresponds to the number of data points in each cluster. Clusters 1 and 2 have thicker silhouettes, reflecting their larger sizes compared to Cluster 3, which has fewer data points and thus a thinner silhouette.
- **Identifying misclassified points**: In Clusters 1 and 2, some data points have negative silhouette scores, extending to the left of zero on the plot. These points may be misclassified due to the overlap between Clusters 1 and 2, indicating that they are closer to the neighboring cluster than to their own.

The accompanying scatter plot (right side of Figure 2.2) visualizes the clustered data points in the euclidean space, with colors corresponding to cluster assignments. Clusters 1 and 2 overlap in the central region, which explains the lower silhouette scores and the presence of misclassified points in these clusters. Cluster 3 is well-separated from the others, consistent with its higher silhouette scores.

This example demonstrates how the silhouette plot provides valuable insights into the clustering structure.

2.3 Possession Chains

As with most clustering algorithms, the Fast Search and Density Peaks (FSDP) algorithm depends on a distance measure between the points in the dataset. The choice of an appropriate distance measure is heavily dependent on the characteristics of the data. Therefore, we introduce the general structure of our data in this section.

The data consists of on-ball events, recorded both live at the stadium and from video. For a given match, the data is presented as an ordered list of events. Every event has X,Y coordinates and a time stamp, along with some descriptors of the event. This thesis will mainly focus on the X,Y coordinates of subsequent events, namely possession chains.

We define a possession chain as a series of events during which one team (or none) is in possession of the ball. A match can then be divided into a sequence of possession chains P_i^{team}.

$$P_1^{team}, P_2^{team}, \dots, P_N^{team} \tag{2.10}$$

Here, $team$ indicates which team P_i belongs to, or none if neither team is in possession. Since this thesis focuses on detecting tactical patterns that are assumed to differ among teams, we are interested in a single team at a time. Therefore, $team$ becomes a constant representing the team under analysis, and we can omit the $team$ notation, Thus, we have

$$P_1, P_2, \dots, P_N \tag{2.11}$$

where P_i is a sequence of events e_j. The subscript m in Equation 2.12 represents the position of the last event in the sequence, and consequently its length.

$$P_i = e_1, e_2, \dots, e_m \tag{2.12}$$

For our cluster analysis, the order of P_i in a match is not important, only the order of events e_j within a possession chain matters. We extend the data over several matches to obtain a collection of possession chains for a given team. Since team dynamics and tactics may change over time, we restrict our analysis to matches from a single season. Thus, N will be the number of possession chains owned by a given team during that season.

2.3.1 Ball Trajectories

An event e_j may contain one or two geographical coordinates on the field, i.e. where the event starts and where it ends. For example, a pass event will have the (X_1, Y_1) coordinates from where the pass starts and the (X_2, Y_2) coordinates where the pass ends. In contrast, an interception event will only have one pair of coordinates (X_1, Y_1).

The sequence of events is mapped into a series of X_j, Y_j coordinates that forms the ball trajectory during the possession chain.

$$T_i = (X_1, Y_1), (X_2, Y_2), \dots, (X_m, Y_m) \tag{2.13}$$

As m is dependent on the number and type of events in the possession chain, we will have ball trajectories of different length as the data points in our cluster analysis.

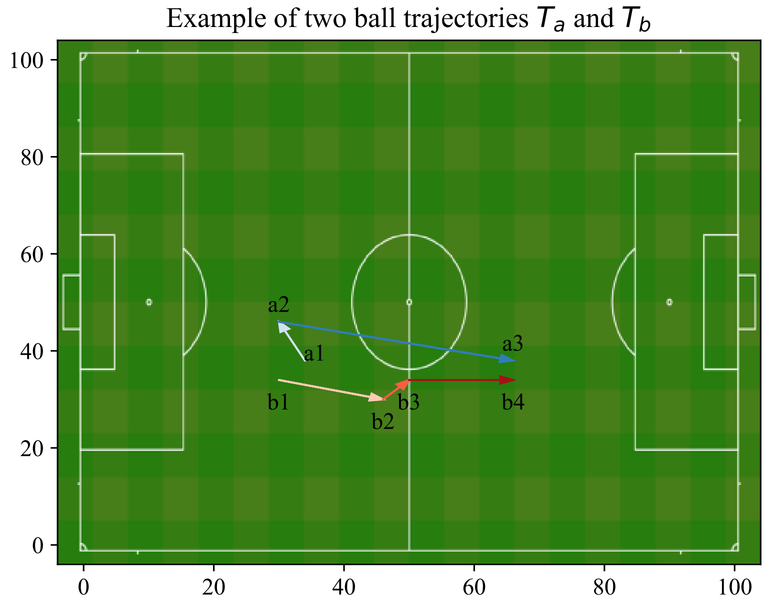


Figure 2.3: Example of two different ball trajectories T_a and T_b. The trajectories are visualized by arrows going between the points, where point $a_i = (X_i, Y_i)$ for trajectory T_a and $b_i = (X_i, Y_i)$ for T_b. The trajectories are differentiated by a red and a blue color schema where the lighter color represent the beginning of the trajectory, while the color darkens as it progresses to the end.

Figure 2.3 shows two trajectories of different length where $m = 3$ for T_a and $m = 4$ for T_b. The trajectories are visualized as arrows so that the observer can understand the direction of the ball. The team in possession has always the oppositions goal to the right and their own goal to the left.

The trajectories in Figure 2.3 appears similar from a visual inspection. Quantifying exactly how similar they are is one of the key problems studied in this thesis. In the next section, we discuss the distance between trajectories

2.4 Distance Between Trajectories

We denote the distance between two trajectories T_i and T_j as $d(T_i, T_j) = d_{ij}$. This distance quantifies the dissimilarity between the trajectories, reflecting the geometric differences between their respective paths across the football field. Since trajectories may vary in length due to the number and nature of events in a possession chain, the distance measure must be robust to such variations.

The primary challenge in this thesis is to identify a suitable distance measure that effectively captures the variations in ball trajectories corresponding to different tactical patterns of the game. The distance measure should account for the nuances of football game-play, including variable trajectory lengths, noise, and the dynamic nature of ball movement.

2.4.1 Overview of Trajectory Distance Measures

Advancements in sensor technology have led to extensive research on distance measures suitable for trajectory analysis. Su et al. [10] provide a comprehensive comparative survey of 15 distance measures, evaluating their performance across multiple dimensions. They categorize these measures based on two main criteria:

1. **Data Representation**: Whether the trajectory data are treated as discrete points or continuous curves.
2. **Temporal Information**: Whether the distance measure considers the timing of trajectory points.

The survey assesses the effectiveness of these distance measures in handling real-world transformations such as noise, point shifts, and trajectory shifts, as well as their computational complexity.

An important aspect discussed in the literature is whether a distance measure satisfies the properties of a metric space. One critical property is the **triangle inequality**, which states that for any three trajectories T_1, T_2, and T_3:

$$d(T_1, T_3) \leq d(T_1, T_2) + d(T_2, T_3). \tag{2.14}$$

The triangle inequality ensures consistent and meaningful distance relationships among trajectories, which is also stated as an assumption for the FSDP algorithm in the paper by Rodriguez and Laio [9].

2.4.2 Contrasting Perspectives on Distance Measures

Based on Su et al. [10] experimental results, they identify the Longest Common Subsequence (LCSS) distance as one of the top five most effective measures, citing its robustness to noise and flexibility in matching trajectories of varying lengths. Conversely, they rank the Fréchet distance among the bottom five, noting challenges with computational complexity and sensitivity to noise.

In contrast, Tao et al. [11] find that the Fréchet distance tends to align more closely with human perceptions of similarity in trajectory data. They observe that the Fréchet distance effectively distinguishes between trajectories arising from different activities, while the LCSS distance may fail to differentiate trajectories that are contextually dissimilar but share common subsequences.

These conflicting findings highlight that the selection of a distance measure cannot rely solely on theoretical attributes or general performance rankings. The effectiveness of a distance measure is highly dependent on the specific application and characteristics of the data-set. Therefore, an empirical evaluation is appropriate to determine the most suitable distance measure for clustering football trajectories.

2.4.3 Approach of This Thesis

Given the importance of choosing an appropriate distance measure, this thesis employs two different measures with distinct theoretical properties:

- **Fréchet Distance**: Emphasizes the exact spatial shape and sequential ordering of trajectories, making it suitable for capturing precise movement patterns.
- **Longest Common Subsequence (LCSS) Distance**: Offers flexibility by focusing on common subsequences within a spatial threshold, accommodating variations and partial matching.

By incorporating both measures, we aim to evaluate their effectiveness in capturing tactical patterns within the data. An empirical evaluation will be conducted to assess how well each distance measure performs in the clustering analysis. This approach acknowledges that theoretical attributes provide valuable guidance but must be validated against practical outcomes in the intended application.

The subsequent sections present the Fréchet distance and the LCSS distance in detail, including their mathematical definitions and theoretical properties.

2.4.4 Fréchet Distance

The Fréchet distance is a measure used to quantify the dissimilarity between two trajectories by capturing the minimum "leash length" required to traverse the trajectories while respecting their sequential ordering. Unlike measures that compare trajectories based solely on point-wise distances, the Fréchet distance considers the location and ordering of points along the curves, providing a comprehensive assessment of their similarity.

Intuitively, the Fréchet distance can be visualized using the analogy of a person walking a dog. The person and the dog each walk along their respective trajectories, T_i and T_j, without backtracking. The Fréchet distance represents the minimal length of a leash that allows both to move from the start to the end of their paths while remaining connected.

This analogy emphasizes the importance of both spatial proximity and the alignment of sequence ordering in the trajectories.

Formal Definition

Consider two continuous trajectories $T_i(t)$ and $T_j(t)$ parameterized over the interval $t \in [0,1]$. The Fréchet distance $d_F(T_i,T_j)$ between T_i and T_j is defined by Alt and Godau [1]:

$$d_F(T_i,T_j) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \left\|T_i(\alpha(t)) - T_j(\beta(t))\right\|, \tag{2.15}$$

where:

- $\alpha(t)$ and $\beta(t)$ are continuous, non-decreasing reparameterizations of the interval $[0,1]$, satisfying $\alpha(0) = \beta(0) = 0$ and $\alpha(1) = \beta(1) = 1$.
- $\|\cdot\|$ denotes the Euclidean distance between points.

These reparameterizations allow for the synchronization of the trajectories as they are traversed, ensuring that the comparison accounts for the inherent temporal progression along each path.

Free Space Diagram and Decision Problem

To compute the Fréchet distance, we utilize the concept of a *free space diagram* [1]. The free space diagram is a two-dimensional representation where:

- The horizontal axis corresponds to the parameterization of trajectory T_i from $t = 0$ to $t = 1$.
- The vertical axis corresponds to the parameterization of trajectory T_j from $t = 0$ to $t = 1$.

Each point in the diagram represents a pair of positions $(T_i(s),T_j(t))$ along the trajectories. The diagram is divided into cells, each corresponding to a pair of segments from T_i and T_j.

Within this diagram, the *free space* for a given threshold ε is defined as the set of all points (s,t) satisfying:

$$\left\|T_i(s) - T_j(t)\right\| \le \varepsilon. \tag{2.16}$$

The **decision problem** involves determining whether there exists a continuous, monotonically increasing path from the bottom-left corner $(0,0)$ to the top-right corner $(1,1)$ of the free space diagram that lies entirely within the free space. If such a path exists for a given ε, it implies that the Fréchet distance between T_i and T_j is less than or equal to ε.

By solving the decision problem for different values of ε, we can perform a binary search to find the minimal ε for which a valid path exists. This minimal value is the Fréchet distance between the two trajectories.

The implementation of the Fréchet distance computation, including the construction of the free space diagram and the solution of the decision problem, is described in detail in Appendix A.1.

The time complexity of computing the Fréchet distance between T_i and T_j where $\text{lenght}(T_i) = n$ and $\text{lenght}(T_j) = m$ is derived by the complexity of its 3 components:

- Free Space Computation: $O(nm)$
- Decision Problem: $O(nm)$, due to the reachability analysis over the grid.
- Binary Search: $O(\log k)$, where k is the number of critical values.

where the overall complexity is $O(nm \log k)$.

2.4.5 Longest Common Subsequence (LCSS) Distance

The Longest Common Subsequence (LCSS) distance is a measure used to quantify the similarity between two trajectories by identifying the longest subsequence of points that are common to both trajectories within a specified spatial threshold. Unlike measures that require exact point-to-point correspondence, LCSS allows for some elements to be unmatched, providing robustness against noise, outliers, and variations in trajectory lengths.

Intuitively, the LCSS distance can be thought of as a flexible matching process that aligns similar portions of trajectories while ignoring dissimilar or noisy segments. This flexibility is particularly beneficial in the context of football trajectories, where possession chains can vary significantly in length and may include intermittent variations that are of less importance for the tactical game-play.

Formal Definition

Consider two trajectories $T_a = a_1, a_2, \ldots, a_n$ and $T_b = b_1, b_2, \ldots, b_m$, where point $a_k = (X_k, Y_k)$ for trajectory T_a and $b_l = (X_l, Y_l)$ for T_b. The LCSS distance focuses solely on the spatial component, without considering temporal constraints [13].

The length of the longest common subsequence $\text{LCSS}(k,l)$ between the prefixes $[a_1, \ldots, a_k]$ and $[b_1, \ldots, b_l]$ is computed recursively:

$$\text{LCSS}(k,l) = \begin{cases} 0, & \text{if } k = 0 \text{ or } l = 0, \\ \text{LCSS}(k-1, l-1) + 1, & \text{if } \text{match}(a_k, b_l), \\ \max\{\text{LCSS}(k-1, l), \text{LCSS}(k, l-1)\}, & \text{otherwise}, \end{cases} \tag{2.17}$$

where the function $\text{match}(a_k, b_l)$ determines if points a_k and b_l are considered a match under the spatial constraint:

$$\text{match}(a_k, b_l) = \begin{cases} \text{True}, & \text{if } \|a_k - b_l\| \leq \varepsilon, \\ \text{False}, & \text{otherwise}. \end{cases} \tag{2.18}$$

Here, $\|\cdot\|$ denotes the Euclidean distance between points, and ε is the spatial threshold controlling the maximum allowable spatial deviation between matched points.

The LCSS similarity measure between T_a and T_b is then defined as:

$$\text{Sim}_{\text{LCSS}}(T_a, T_b) = \frac{\text{LCSS}(n,m)}{\min(n,m)}. \tag{2.19}$$

The LCSS distance $d_{\text{LCSS}}(T_a, T_b)$ is derived from the similarity measure:

$$d_{\text{LCSS}}(T_a, T_b) = 1 - \text{Sim}_{\text{LCSS}}(T_a, T_b) = 1 - \frac{\text{LCSS}(n,m)}{\min(n,m)}. \tag{2.20}$$

This distance ranges between 0 and 1:

- $d_{\text{LCSS}}(T_a, T_b) = 0$ indicates that the trajectories are identical within the specified spatial threshold.
- $d_{\text{LCSS}}(T_a, T_b) = 1$ implies that there is no common subsequence satisfying the spatial threshold.

This thesis utilizes the implementation of the LCSS distance in the `traj_dist` library [5] that directly follows the formal definition provided above. It employs a dynamic programming approach to compute the $\text{LCSS}(k,l)$ table, with a computational complexity of $O(nm)$.

2.4.6 Comparing Fréchet and LCSS Distances

Table 2.1 presents the theoretical properties that are of most interest for our cluster analysis.

	Fréchet Distance	**LCSS Distance**
Robust to Outliers	No	Yes
Metric	Yes	No
Parameter free	Yes	No
Computational Complexity	$O(nm\log k)$	$O(nm)$
Type	Continuous	Discrete
Family	Lp-norm	Edit distance

Table 2.1: *Comparison of theoretical properties between Fréchet Distance and LCSS Distance*

Fréchet Distance

The Fréchet distance is continuous, meaning it considers both the sample points and the movement in-between. It evaluates the shape of trajectories in a strict manner by assessing all points along the paths, including the continuous transitions between them. This comprehensive consideration can make it sensitive to minor deviations and outliers, as any variation along the trajectory may influence the overall distance calculation.

It is parameter free. However, the higher computational complexity of the Fréchet distance may present challenges when processing large datasets. As the size of our data set is naturally bounded by the season and team. This is not an issue in our analysis.

LCSS Distance

The LCSS distance reflects the proportion of two trajectories that are close to each other based on a matching condition where the Euclidean distance between points is less than a threshold ε. By allowing unmatched segments and focusing on matching points within the threshold, it offers flexibility and robustness against noise and outliers.

In FSDP clustering, the robustness to outliers can be advantageous, potentially leading to more stable local density estimates and clusters that are less affected by noise in the data. However, a significant drawback of the LCSS distance is that it is not metric; specifically, it violates the triangle inequality. This violation means that the direct distance between two trajectories may be greater than the sum of the distances when passing through a third trajectory. Such inconsistencies can affect the reliability of distance measurements and may impact the effectiveness of clustering algorithms like FSDP.

Furthermore, since LCSS belongs to the Edit distance family, the distribution of distance values is discrete, as it counts the number of matching points rather than evaluating continuous differences. This can lead to less granularity in the distance measurements, potentially causing different trajectory pairs to have identical or similar distance values. In the context of FSDP clustering, this lack of granularity might affect the algorithm's ability to distinguish between trajectories based on subtle differences. It may impact the identification of cluster centers and the overall clustering structure, as the local density estimation could be less precise.

Illustrative Example

To illustrate how the Fréchet and LCSS distances capture different aspects of trajectory similarity, consider the trajectories depicted in Figure 2.4. The two ball trajectories, T_a and T_b, are shown, where T_a consists of points $a_i = (X_i, Y_i)$ and T_b consists of points $b_j = (X_j, Y_j)$. These trajectories are the same as in Fig 2.3 when plotted on top of the football field.

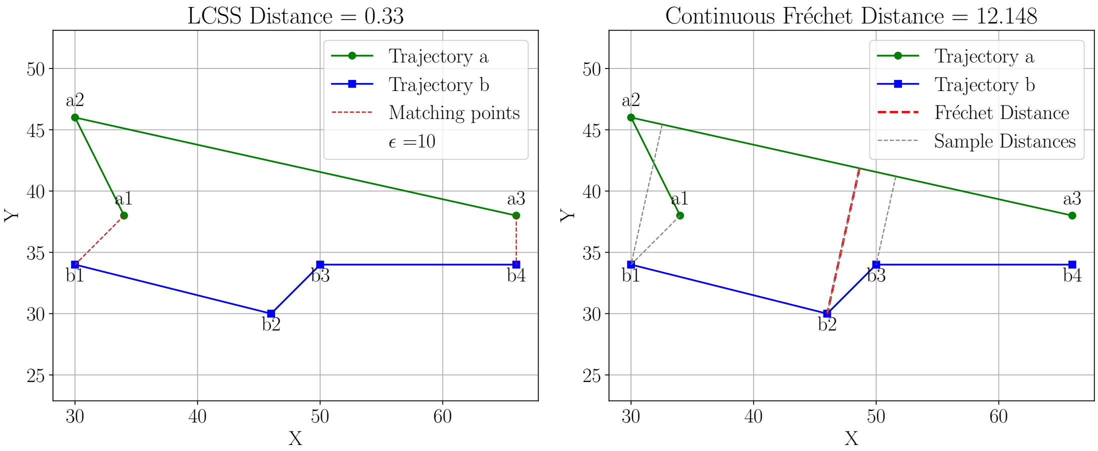


Figure 2.4: Example of two different ball trajectories T_a and T_b. Trajectory T_a is shown in green with points a_1, a_2, a_3, and trajectory T_b is shown in blue with points b_1, b_2, b_3, b_4. The left plot shows LCSS matching points within the threshold ε, connected with red dashed lines. The right plot illustrates the Fréchet distance represented by the red dashed line.

The trajectories are defined as:

$$
\begin{aligned}
T_a: &\quad a_1 = (34,38), \quad a_2 = (30,46), \quad a_3 = (66,38), \\
T_b: &\quad b_1 = (30,34), \quad b_2 = (46,30), \quad b_3 = (50,34), \quad b_4 = (66,34).
\end{aligned}
$$

Fréchet Distance

The Fréchet distance between T_a and T_b is approximately 12.148, computed by considering the continuous movement along the trajectories. It measures the minimal leash length required to traverse both trajectories from start to end, strictly considering the shape and sequence of the paths. In the right plot of Figure 2.4, the red dashed line represents the Fréchet distance.

This distance highlights the differences in the overall shape and sequence of the trajectories. The Fréchet distance is sensitive to deviations, as it evaluates both the sample points and the movement between them, reflecting its continuous nature and strict adherence to the trajectory shape.

LCSS Distance

Using a spatial threshold $\varepsilon = 10$, the LCSS distance focuses on matching points between T_a and T_b. The matching pairs are:

$$(a_1, b_1), \quad (a_3, b_4).$$

These points are connected with red dashed lines in the left plot of Figure 2.4. The LCSS similarity measure is approximately 0.667, resulting in an LCSS distance of 0.333.

The LCSS distance captures similarities despite variations, emphasizing shared subsequences without requiring strict alignment. It is robust to outliers and variations in trajectory shape, as it allows for unmatched segments and focuses on the proportion of matching points.

This example demonstrates that:

- The **Fréchet distance** emphasizes the maximum deviation between trajectories, highlighting differences in shape and sequence due to its continuous evaluation and strict consideration of the entire path.

- The **LCSS distance** captures common patterns by identifying matching points within a threshold, reflecting its discrete nature and flexibility in matching. It emphasizes similarities even when the overall trajectory shapes differ.

These visual differences relate directly to the properties of each distance measure. The Fréchet distance's sensitivity to shape deviations and strict alignment contrasts with the LCSS distance's robustness to variations and focus on shared segments.

Both the Fréchet and LCSS distances offer valuable perspectives for clustering football trajectories. The choice between them depends on the specific requirements of the analysis, the characteristics of the data, and the intended application. By empirically evaluating both measures, this study aims to identify the most effective approach for uncovering meaningful clusters that represent different tactical patterns in football.

3. Results

This chapter presents the findings of our trajectory clustering analysis, organized around key methodological steps. We begin with a descriptive analysis of the dataset to outline its primary characteristics. Next, we evaluate the clustering results for each distance measure — Longest Common Subsequence (LCSS) and Fréchet distance.

To visualize the resulting clusters, we display the trajectories of core points on a football field, where the core points are characterized by high local density and are located close to the cluster center. This visual representation allows for an intuitive understanding of the spatial patterns and movements associated with each cluster.

Finally, we infer which clusters are most strongly associated with goal-scoring, offering insights into the tactical implications of different trajectory patterns in football. This analysis provides practical conclusions that could lead to improved coaching strategies.

3.1 Data

The data was originally generated by a company that collects on-ball events during the game. The data used in this thesis has been processed by the company Twelve Football, that has partitioned the events into possession chains.

This thesis uses the possession chains of the team Arsenal F.C. during the Premier League season of 2017/2018. As covered in Section 2.3.1, the ball trajectories are extracted from the possession chains and stored in a list $T_1, T_2, \ldots, T_n$.

It turns out that some of the possession chains are very short, containing only one or two events. Since this thesis is focused on finding tactical gameplay patterns based on the ball trajectory, we remove any trajectories with $length < 3$. The characteristics of the dataset studied in this thesis are presented in Table 3.1.

Team	Arsenal
Season	17/18
League	Premier League
Number of possession chains	2880
Number of filtered possession chains	2497
Median length of possession chains	14

Table 3.1: *Possession chain statistics for Arsenal in the 17/18 Premier league season*

The lengths of the trajectories range between 3 and 105. The distribution is presented in Figure 3.1.

We observe that shorter trajectories are over-represented in the data, with the median length being 14. The distribution has a long, thin tail with a maximum value of 105. The shorter trajectories especially affect the LCSS distance, causing the distance distribution to become more discrete. In the extreme case where the $length = 3$ for any of the trajec-

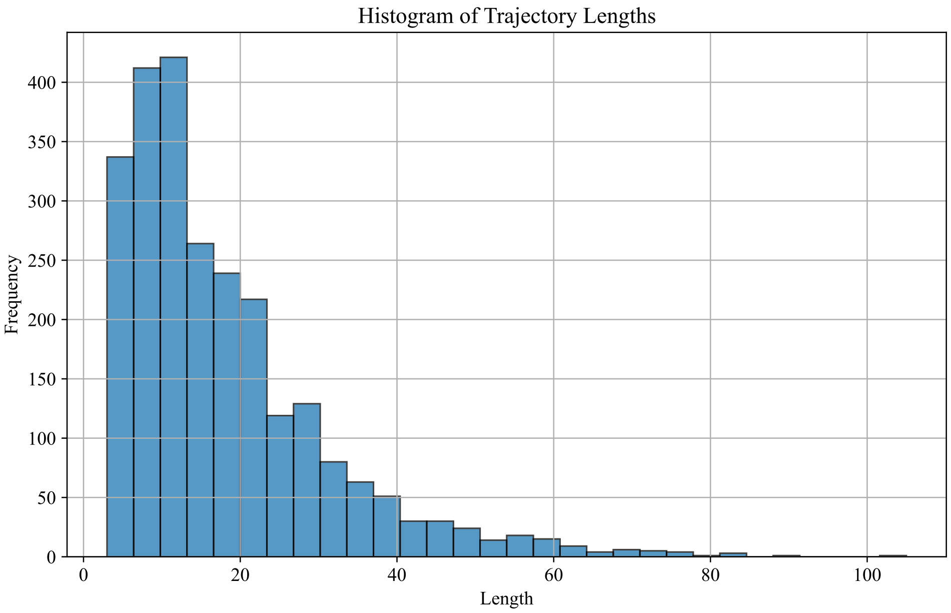


Figure 3.1: Histogram of the trajectory length distribution, where length refers to the number of points or (X_i, Y_i) coordinates in a trajectory.

tories T_i or T_j, the distance measure $d_{\text{LCSS}}(T_i, T_j)$ will only have four possible values, i.e., $d_{lcss}(T_i, T_j) \in [0, \frac{1}{3}, \frac{2}{3}, 1]$.

After extracting and filtering the trajectories, we calculate the pairwise distances between them using the chosen distance measure. For easier comparison, we normalize the Fréchet distance to the interval $[0, 1]$ by dividing $d_F(T_i, T_j)$ by its maximum value as seen in equation 3.1.

$$d_F(T_i, T_j) = \frac{d_F(T_i, T_j)}{\max(d_F(T_i, T_j))} \tag{3.1}$$

3.2 Cluster Evaluation

3.2.1 Fréchet Distance

Choosing the Cutoff Distance d_c

As we increase the cutoff distance d_c, the neighborhood considered for computing the local density ρ around each point becomes larger. This broader neighborhood includes more data points, leading to higher ρ values and a wider spread in the density distribution. Consequently, points that were previously considered to have low density at smaller d_c values may now exhibit higher densities due to the inclusion of more distant neighbors.

Similarly, the minimum distance δ to a point with higher density is influenced by the choice of d_c. With a larger d_c, there is a greater chance of finding points with comparable or higher densities at varying distances. This increases the variability in δ values because the distances to these higher-density points can differ significantly across the dataset. The overall effect is a larger spread in both ρ and δ values at higher d_c levels, reflecting the underlying structure of the data where clusters may be more diffuse or overlapping.

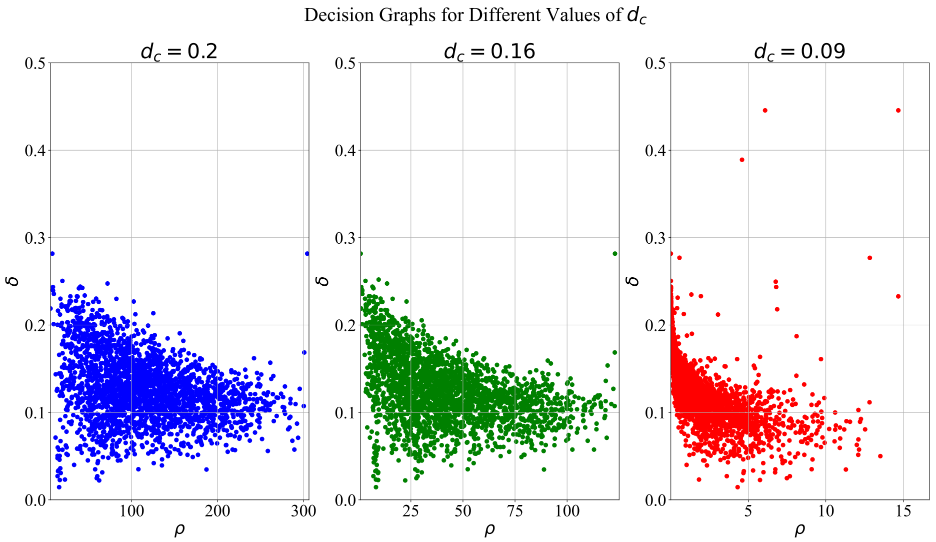


Figure 3.2: Decision graphs (ρ vs δ plots) based on the Fréchet distance for different values of d_c. Starting with $d_c = 0.2$ in the blue scatter plot, $d_c = 0.16$ for the green and $d_c = 0.09$ for the red scatter plot.

As d_c decreases, the neighborhood used to compute the density ρ for each point becomes smaller. This means that for any given point, the contribution in density from other points will drop much faster as their distance from each other increases. As we see in Figure 3.2, this will push the concentration of points to the left. There is, however, still a substantial set of points with the same relatively large density, making them potential candidates for cluster centers.

We also observe δ values appear more squeezed together as d_c decreases. This is most apparent in the region where $1 < \rho < 3$. It is explained by the more compressed distribution of ρ increasing the likelihood of finding a closer point with a slightly higher density.

From the red scatter plot in Figure 3.2, we observe some distinct points emerge with relatively high ρ and δ. These points will be our candidates for cluster centers.

Choosing the cluster centers

In Figure 3.3, by plotting $\delta \times \rho$ in descending order we can observe a clear separation of points with significant higher $\delta \times \rho$ for the top four highest.

However, from Figure 3.4, we also observe a severe decline in the silhouette score when the number of cluster centers increases beyond two.

The low silhouette score for a higher number of centers suggests that the appropriate number of clusters is two. Generating the clusters based on the 2 points with the highest $\delta \times \rho$ yields the cluster assignment presented in Table 3.2.

Cluster	Center Point Index	Number of Core Points	Number of Halo Points
1	1516	81	1128
2	2140	61	1227

Table 3.2: *Formed clusters based on the 2 center points with the highest $\delta \times \rho$ for $d_c = 0.09$.*

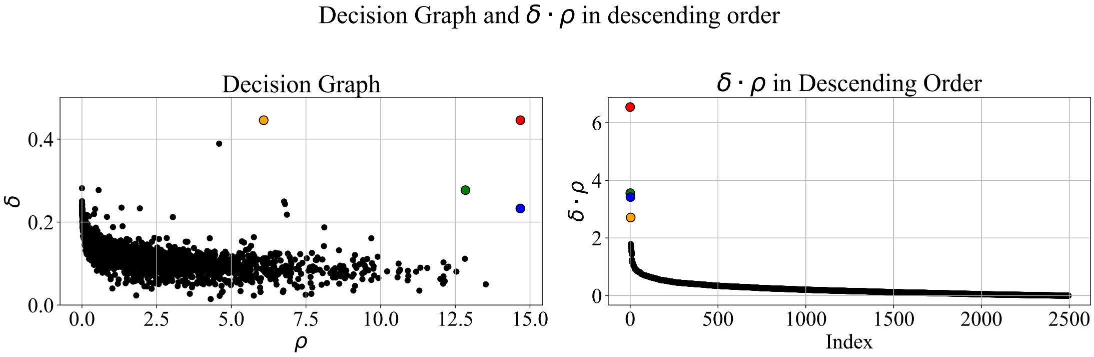


Figure 3.3: Decision graph for $d_c = 0.09$ in the left plot, where the points associated with the highest $\delta \times \rho$ are marked in colors. The plot to the right shows $\delta \times \rho$ in descending order where the corresponding points is marked in the same color. Each point corresponds to a trajectory where the marked ones are potential cluster centers.

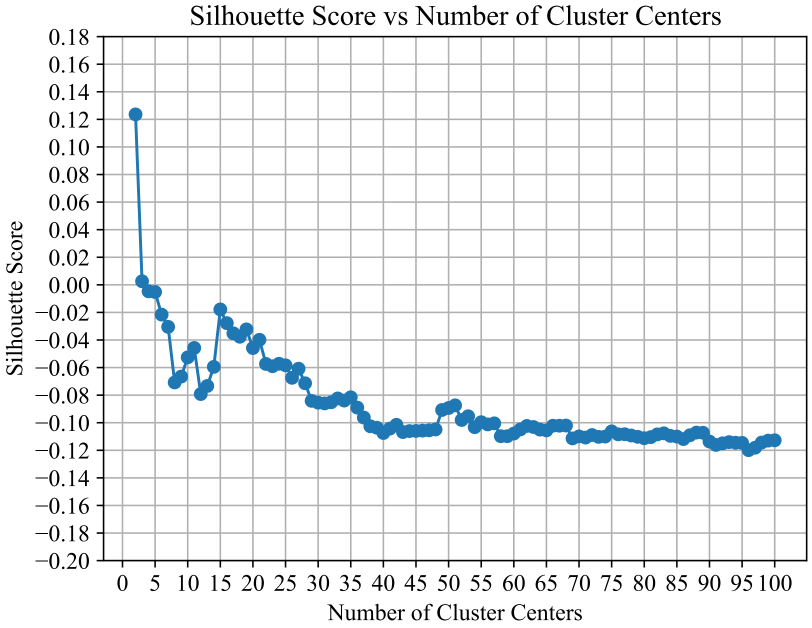


Figure 3.4: Silhouette score vs number of centers. The centers corresponds to the points with the highest $\delta \times \rho$.

We observe an even distribution of points between the clusters which is desirable as it enhances the discriminative power discussed in Section 2.2.2. There is a large proportion of halo points present in both clusters, with core points accounting for only 7.2% in cluster 1 and 5% in cluster 2.

The low proportion of core points is expected, given the sparse number of points in the lower right region of the decision graph presented in Figure 3.3. These points have high density (ρ) and low distance to the nearest point with higher density (δ), indicating that they are not centers themselves but are close to a center. These points represent the core of the cluster.

The high concentration of points in the lower left region of the decision graph in 3.3 rather suggests that most points are part of areas with lower density further away from the center. This increases the likelihood for clusters to overlap, where one cluster boarder may come closer to another clusters more dense area. This will inflate the boarder density ρ_b in those clusters, causing higher amount of halo points.

Note that this interpretation of halo points differs from that of outliers, which are represented by points with very low ρ and relatively high δ. While outliers are also present in the decision graph in Figure 3.3, they are likely not the reason for the large proportion of halo points in the formed clusters in Table 3.2.

As we shall see later in our analysis, we might consider disregarding the halo points, as they are likely overlapping with the dense areas of other clusters.

Exploring permutations of the top k points with highest $\delta \times \rho$

The plot in Figure 3.4 does not account for different permutations of potential cluster centers, as it cumulatively increases the number of centers by including the subsequent point with the highest $\delta \times \rho$ in descending order.

To account for potentially missed center candidates, we plot the silhouette score calculated from combinations of centers chosen from the points with the 12 highest $\delta \times \rho$.

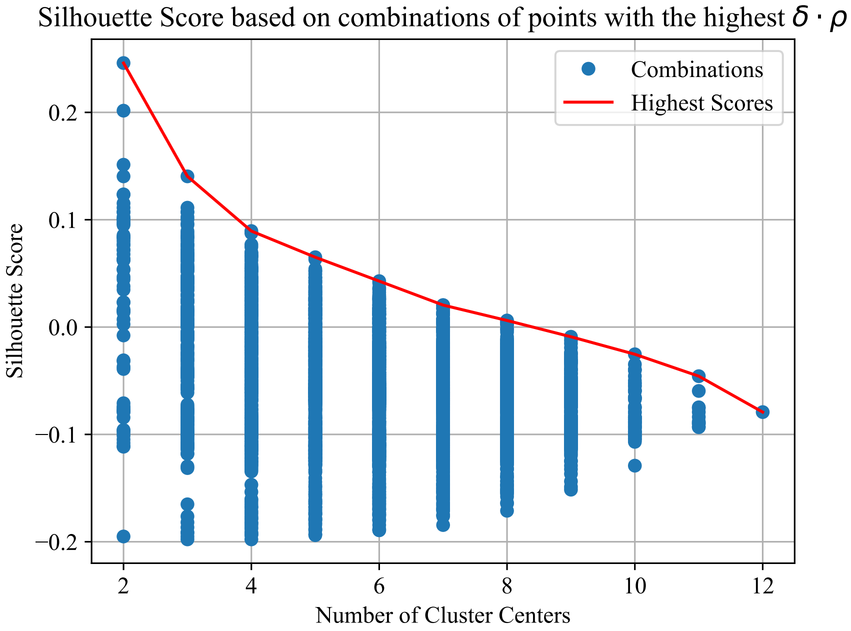


Figure 3.5: Silhouette score vs number of centers. The centers corresponds to any combinations of points among the top 12 highest $\delta \times \rho$.

The highest score is represented by the red line in Figure 3.5. We observe a significantly higher score than previously seen in Figure 3.4. The outcome remains the same, with the highest score achieved using two cluster centers. The notable change is that one of the center points has changed, the trajectory with index 186 replaces the one with index 2140. The sets of points yielding the top three highest scores are presented in Table 3.3 and the cluster assignments are presented in Table 3.4.

Number of Clusters	Silhouette Score	Cluster Centers
2	0.25	(1516, 186)
3	0.14	(1516, 186, 2131)
4	0.09	(1516, 2140, 186, 2131)

Table 3.3: *The combination of centers yielding the highest silhouette score for 2, 3, and 4 clusters. The cluster centers are represented by their index number in the data. Note that the 4 different center points are among the top 5 highest $\delta \times \rho$.*

Cluster label	Center	Elements	Core	Halo
First Set of Clusters				
1	186	10	10	0
2	1516	2487	2487	0
Second Set of Clusters				
1	186	10	10	0
2	1516	2456	2456	0
3	2131	31	31	0
Third Set of Clusters				
1	186	10	10	0
2	1516	1178	81	1097
3	2131	31	31	0
4	2140	1278	61	1217

Table 3.4: *Cluster results for three different sets of clusters. The cluster label is presented in the first column, followed by the center index, number of total elements in the cluster, and the number of core and halo points in the last two columns.*

Notably, the clusters generated by indices 186 and 2131 are small, with only 10 and 31 points, respectively. Both are composed entirely of core points, indicating strong separation from each other and from the larger cluster centered at 1516.

Introducing a fourth cluster at index 2140 (previously considered a cluster center in Table 3.2) effectively splits the larger cluster into two, preserving the same number of core points observed in Table 3.2. Although this increases the discriminative power by creating a more balanced partition, it also introduces a substantial number of halo points due to poorer separation between the two larger clusters, resulting in a lower overall silhouette score of 0.09.

By focusing only on core points, we can exploit the improved discriminative power offered by splitting the large cluster, while mitigating the confounding effects of points on the ambiguous cluster boundaries. Figure 3.6 shows the silhouette plot of these four clusters (with centers at indices 1516, 2140, 186, and 2131), computed using only core points.

We observe an average silhouette score of 0.42. This indicates relatively high cohesion within clusters and clear separation between clusters. The smaller clusters represented by

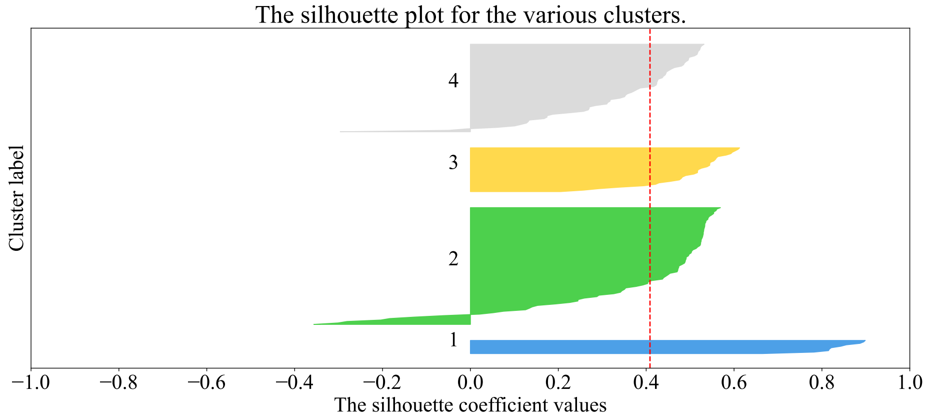


Figure 3.6: Silhouette plot of the cluster result, using 4 centers represented by the trajectories with index: (1516, 2140, 186, 2131). Only core points were considered. Each silhouette is marked with the corresponding cluster label from Table 3.4 and differentiated by colors. The dashed red line shows the average silhouette score over all points and equals 0.42.

points 186 and 2131 show the best results, with a majority of individual scores above the average.

The two larger clusters, represented in gray and green, have less decisive results. The gray cluster represented by point 2140 has the fewest individual scores above the average, and the green cluster represented by point 1516 has the highest number of negative scores, indicating a potential misassignment of those points.

This observation is consistent with the originally large proportion of halo points in these two clusters (see Table 3.4), which signals weaker separation. Nevertheless, the overall silhouette plot and mean score of 0.42 highlight the merits of focusing on core points to capture the fundamental structure of the data.

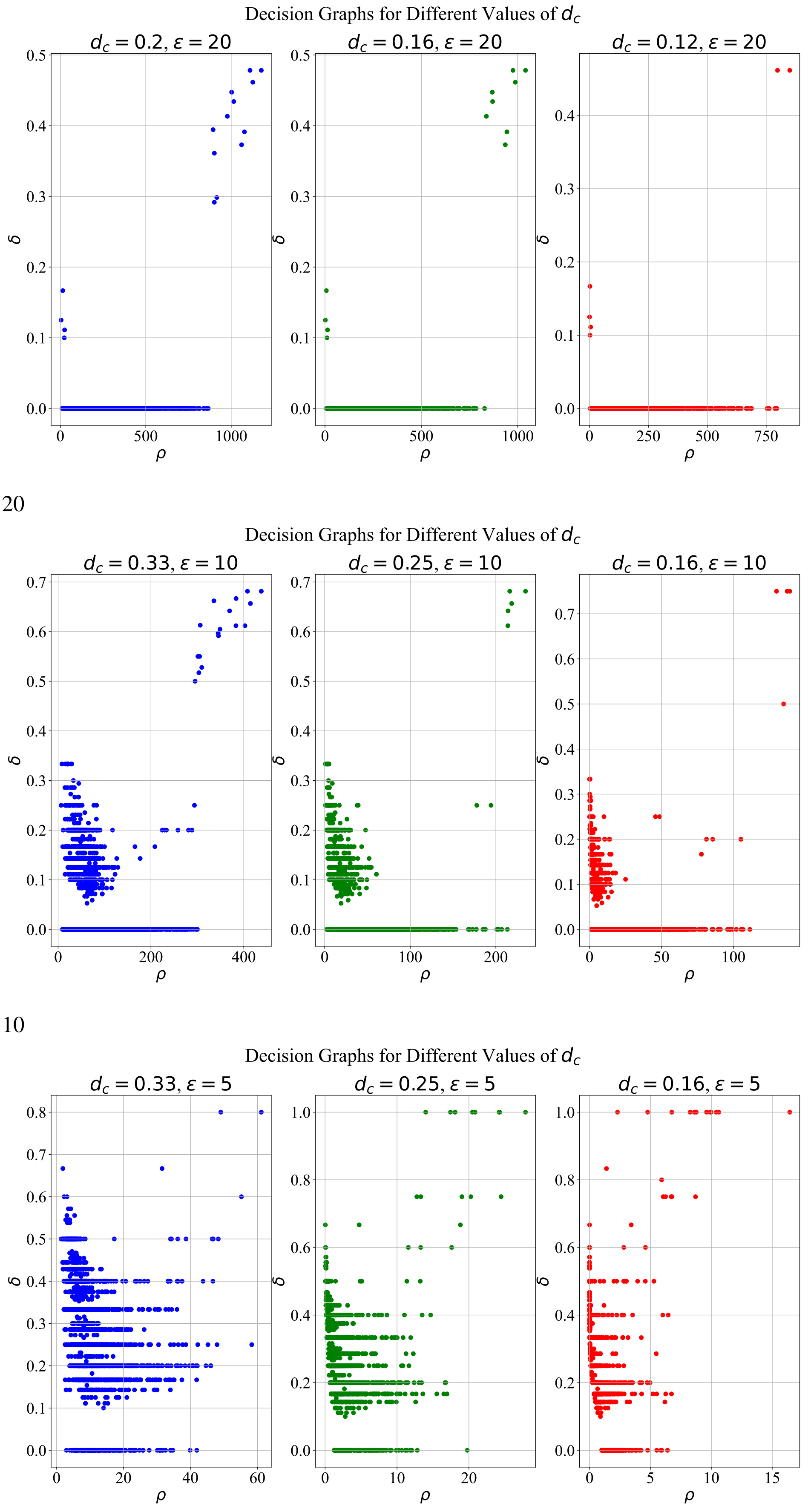
3.2.2 LCSS distance

In contrast to the Fréchet distance, the LCSS distance is not parameter-free. It requires us to set the spatial threshold ε. As a consequence, we have two parameters that govern the shape of the decision graphs presented in Figure 3.7.

We observe very different results compared to the decision graphs based on the Fréchet distance in Figure 3.2. A notable difference is that the δ values are now discrete, as seen in Figures 3.7b and 3.7c. This is expected due to the discrete nature of the LCSS distance.

We notice that the number of points with $\delta = 0$ decrease as ε decreases. In fact, we start seeing more points with $\delta = 1$. Since ε is the maximum distance allowed between two points to be considered a match, lowering ε shifts the trajectory distances towards one, and as a direct consequence, the δ values are shifted upwards. The effect of decreasing ε on the trajectory distance distribution is presented in Figure 3.8.

The large set of points with $\delta = 0$ for $\varepsilon = 20$ in Figure 3.7a is partially explained by the higher share of trajectories with zero distance between them. However, the number of points with $\delta = 0$ is not proportional to the number of trajectories with zero-distance's as it represents less than 4% of all distances, while in Figure 3.7a except for a few points



(a) $\varepsilon = 20$

(b) $\varepsilon = 10$

(c) $\varepsilon = 5$

Figure 3.7: Decision graphs based on the LCSS distance for different values of d_c and ε.

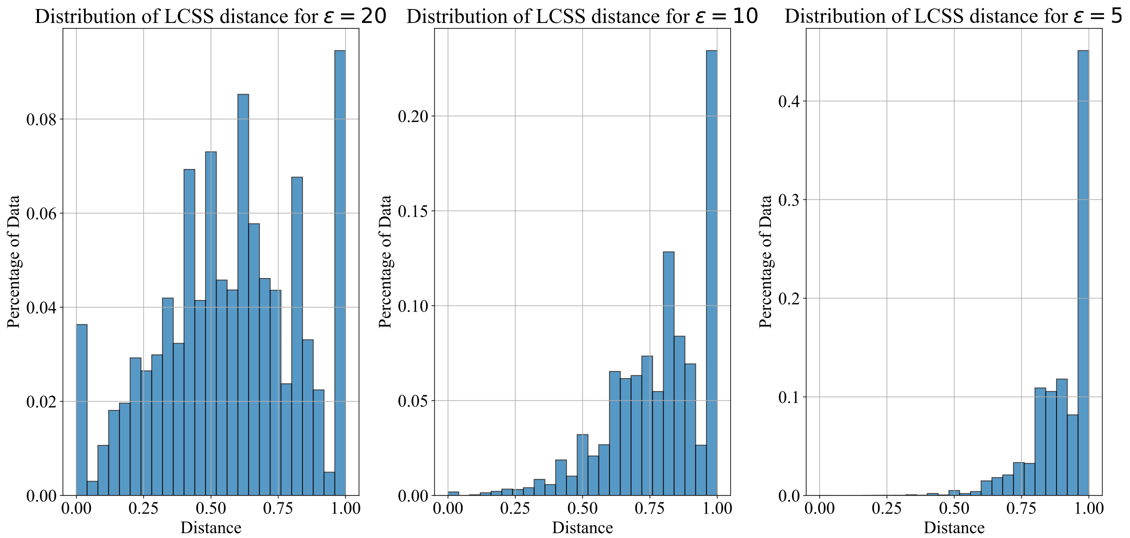


Figure 3.8: Distribution of the LCSS distance for different values of the spatial threshold ε. The distances on the X-axis ranges from 0 to 1. For $\varepsilon = 20$ in the left plot, we have almost a bell curve forming, except the spikes in both tails where the distance's are 0 and 1. As epsilon decreases, the distribution shifts and the percentage of data for length 1 increases drastically while there is diminishing lengths of zero left. The right plot, showing the lowest ε of 5 has around 45% of ones and no visible distances of zero.

all have $\delta = 0$. This suggests that even with a relatively small share of distances equal to zero, a large share of δ values will equal zero.

Consider a point with many distances equal to zero. By definition, this point will have a high density. Since δ is defined by the minimum distance to a point with higher density and since zero will always be the minimum, all points with zero distance to the high density point will have $\delta = 0$.

Since a point is assigned to the same cluster as its nearest neighbor with higher density, the distance between the points will be zero when $\delta = 0$. In the case of $d_c = 0.12$ in Figure 3.7a, the decision graph clearly shows two center points with the rest of the points having $\delta = 0$ (except for four outliers). It also appears that the centers are the two points with the highest ρ. Given the two cluster centers, the point with the third highest ρ will be assigned to either one of the two cluster centers. We know that the distance to whichever center point is closest equals zero since $\delta = 0$. Then the fourth point will be assigned to a cluster based on the distance to its nearest neighbor, which we also know is zero as $\delta = 0$.

Given that $\delta = 0$ for the rest of the points (except for the four outliers), all points will be assigned to a cluster center with the distance to their nearest neighbor equal to zero. Intuitively, this would suggest that the formed clusters will all have points on top of each other, i.e., with zero distance between them. This is, however, contradictory to the distance distribution in Figure 3.8, which suggests a wide spread of the distances with a small portion ($< 4\%$) of distances equal to zero.

This contradiction is explained by the fact that the LCSS distance violates the triangle inequality (defined in Equation 2.14). As it no longer holds true, we have that even if $d_{ik} = d_{kj} = 0$, it does not imply that $d_{ij} = 0$.

Disregarding the LCSS distance from further analysis

To counteract the inflated number of points with $\delta = 0$, we decrease ε. For $\varepsilon = 10$ in Figure 3.7b, a concentration of points is visible for low ρ and relatively low δ. This is more inline with what we see in our results for the Fréchet distance, indicating sparse clusters with many points in low density areas. However, the proportion of points with $\delta = 0$ remains relatively high at around 70% of the data.

When $\varepsilon = 5$ in Figure 3.7c, the proportion of points with $\delta = 0$ accounts for only around 7% of the data. The wider spread of δ implies that points are more sparse and that clusters are less likely to form. This is also confirmed by the highest silhouette score achieved being 0.003 for two centers at $d_c = 0.33$.

Given the results of inflated number of $\delta = 0$ for higher ε and the poor shape of the decision graph and low silhouette score for lower ε, we will disregard the LCSS distance from further analysis.

3.3 Cluster Inference

3.3.1 Clustered Trajectories

The trajectories in the four clusters detected using the Fréchet distance are plotted on the football field in Figure 3.9.

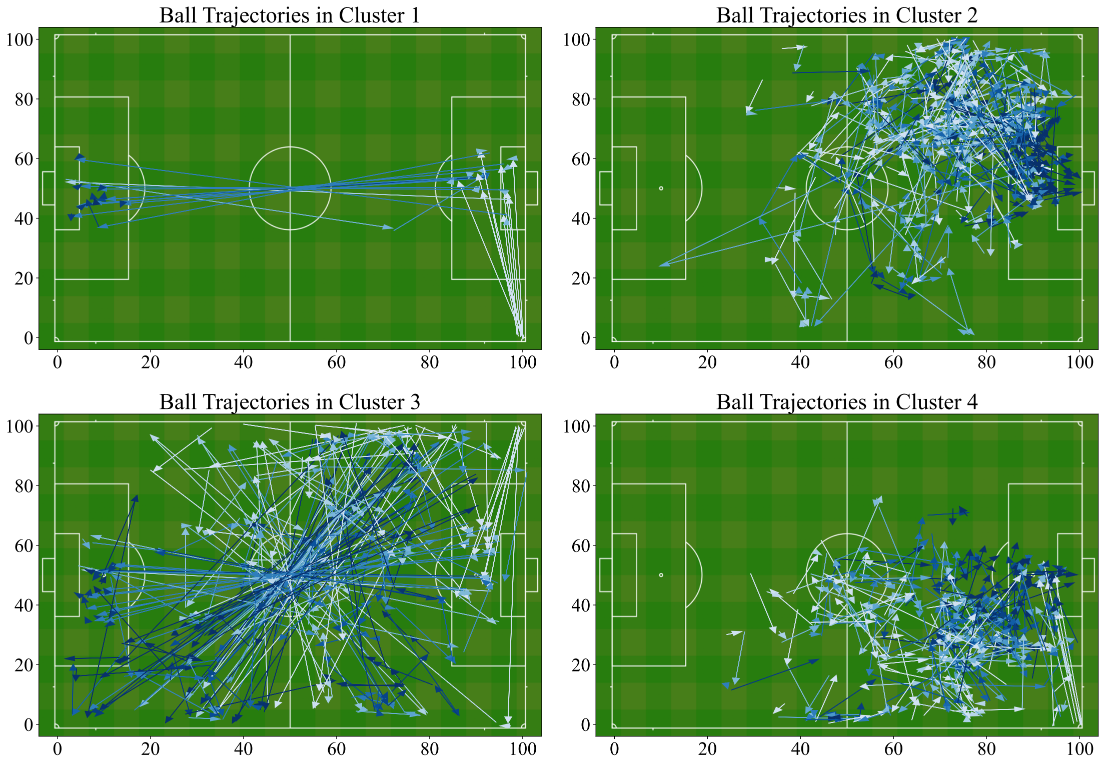


Figure 3.9: Ball trajectories from possession chains in the four clusters generated by the Fréchet distance, using the core points and the centers presented in Table 3.4. The trajectories are visualized by arrows connecting the points and a blue color scheme where the lighter color represents the beginning of the trajectory, while the color darkens as it progresses to the end.

There is a clear visual distinction between the clusters. Cluster 1 also shows a high level of cohesion, as the trajectories in the cluster almost follow the exact same pattern. In terms of football, Cluster 1 represents an unsuccessful corner kick that results in a long pass or clearance back to the team's own goal area.

Clusters 2 and 4 show similar patterns of tactical gameplay but on opposite sides of the football field. The possession seems to start between the center of the field and the edge of the goal area. The ball then travels along the left side for Cluster 2 or the right side for Cluster 4 in a somewhat irregular path. The possession ends somewhere in the goal area of the opposing team, where they potentially score a goal. Notably, Cluster 4 also contains some possession chains starting with a corner kick from the right side.

In terms of the start and end positions, Cluster 3 has the most irregular patterns. The cluster is characterized by long distances between the events in a possession chain, where many of them tend to pass through the center circle. We also see that some of the possession chains start with a corner kick from the left side.

3.3.2 Cluster Characteristics

Scoring a goal is the most important event in football. The number of goals scored per cluster is presented in Table 3.5.

Cluster	**Goals**
1	0
2	6
3	0
4	3

Table 3.5: *The number of goals scored within each cluster. The cluster labels correspond to those introduced in Table 3.4.*

As expected from Figure 3.9, Clusters 1 and 3 do not contain any goals. Cluster 2 has double the number of goals as Cluster 4, i.e., 6 compared to 3. Normalizing for the number of possession chains in the clusters shows that the team is approximately 50% more efficient in scoring a goal when attacking via the left flank.

4. Discussion

In this study, we applied the Fast Search and Density Peaks (FSDP) clustering algorithm [9] to analyze football possession chains based on ball trajectories. Our primary objective was to uncover underlying tactical patterns by grouping similar ball trajectories. The evaluation involved several analytical tools, including decision graphs, cluster assignation with core and halo distinctions, considerations of cluster distribution, and silhouette analysis. We explored two distance measures for trajectory similarity: the Fréchet distance and the Longest Common Subsequence (LCSS) distance. This discussion elaborates on the implications of our findings, the effectiveness of the methodologies employed, the insights gained into football tactics and future improvements.

4.1 Summary of the Results

The choice of an appropriate distance measure is critical in clustering trajectory data. Our analysis compared the Fréchet distance and the LCSS distance, each with distinct theoretical properties. The Fréchet distance considers the continuous shape and sequential ordering of trajectories, making it sensitive to even minor deviations in paths. In contrast, the LCSS distance allows for flexibility by focusing on common subsequences within a spatial threshold, offering robustness to noise.

Our empirical results indicated that the Fréchet distance was more suitable for clustering football trajectories in this context. The Fréchet distance provided a continuous and metric-based measure that respected the triangle inequality, which is an essential assumption in the FSDP algorithm [9]. The LCSS distance, while robust to outliers, violated the triangle inequality and resulted in inflated numbers of zero distances, complicating the clustering process. Consequently, we disregarded the LCSS distance from further analysis.

The application of the FSDP algorithm using the Fréchet distance yielded meaningful clusters that corresponded to distinct tactical patterns. The decision graphs revealed clear potential cluster centers characterized by high local density (ρ) and high minimum distance (δ) to higher-density points. By varying the cutoff distance d_c, we observed how the density and distance distributions changed, allowing us to assess the stability and robustness of potential cluster centers along with the underlying structure of the data.

The cluster assignment process involved selecting centers based on the highest $\delta \times \rho$ values. We found that permutations of the top k points with the highest $\delta \times \rho$ improved the clustering results.

While choosing two cluster centers originally yielded the highest average silhouette score, further analysis revealed 4 cluster centers to be the optimal choice. By utilizing core and halo points, and only consider core points as successful assignments, we could partition parts of the data into 4 distinct clusters. The final result yielded a highest average silhouette score of 0.42. However, the core points only accounts for 7.3% of the data.

The presence of a large proportion of halo points suggested overlapping clusters and lower separation, which is consistent with the observations from the decision graphs. A remedy to this problem is so called Fuzzy Clustering which is discussed in the next section 4.2.2.

Visualizing the clustered trajectories on the football field revealed clear distinctions between clusters. Cluster 1 consisted of unsuccessful corner kicks resulting in long passes or clearances back toward the team's own goal area. Clusters 2 and 4 represented attacking patterns along the left and right flanks, respectively, culminating near the opposition's goal area. Cluster 3 displayed more irregular patterns, with trajectories often passing through the center circle.

Analyzing the clusters in terms of goal-scoring provided practical tactical insights. Cluster 2, representing attacks along the left flank, was associated with double the number of goals compared to Cluster 4, which represented right-flank attacks. This suggests that the team was more effective when attacking via the left side. Such insights could inform coaching strategies, emphasizing the development of plays that exploit the team's strengths on the left flank.

4.2 Limitations and Future Work

Based on the findings from our analysis, several avenues emerge for improving the clustering model to achieve more nuanced and informative results.

4.2.1 Exploring Alternative Distance Measures

Our study highlighted the limitations of the LCSS distance in the context of the FSDP algorithm due to its violation of the triangle inequality and the resulting complications in clustering. While the Fréchet distance proved to be somewhat useful, it only succeeded in partitioning 7.3% of the data into meaningful clusters.

To enhance the clustering analysis, we could explore other distance measures that both adhere to metric properties, offer robustness to noise and better capture variations in trajectory data. Potential candidates include:

- **Time-Weighted Distance Measures**: Modifying the distance calculations to account for the timing of events within possession chains. Incorporating temporal information could enhance the model's ability to distinguish between similar spatial patterns that occur at different paces or under different phases of the game.
- **Graph Distance**: In recent work by Tas Kiper [12], the graph distance: Commute Time Distance (CTD) is used in density peak clustering. They show its enhanced ability to detect clusters as it reduces within-cluster distances and amplifies between-cluster distances compared to conventional distance measures (like the euclidean distance). CTD is also robust to noise as it is based on the average of links connecting two nodes in the graph.
- **Contextual Features**: Including additional variables such as the event type (e.g., pass or a shot) or player involvement to provide a richer data set that captures more variation of the tactical game play.

Empirically evaluating and possibly combining these distance measures could reveal a more suitable measure that better captures the variation in tactical patterns on the football field.

4.2.2 Refining cluster assignment and Handling of Halo Points

The presence of a large proportion of halo points in the clusters indicates potential overlaps and lower separation between clusters. To address this issue, this thesis suggests a fuzzy clustering approach.

In contrast to the original FSDP algorithm applied in this thesis, fuzzy clustering consider each point to have a probability of membership to each cluster. This is a generalization of the special case of hard clustering (applied in this thesis) where the probability of membership is either 0 or 1 for each point. Tas Kiper [12] present an extension of the original FSDP algorithm for fuzzy clustering.

Applying the fuzzy clustering approach will give further insights into the halo points and could potentially increase the coverage of assigned points, I.e. instead of disregarding all halo points as unassigned, they can be included as members in multiple clusters.

4.2.3 Future Data Enhancements

Lastly, improving the quality and breadth of the data could significantly enhance the clustering model:

- **Validating results with multiple teams and seasons**: By performing the analysis on multiple teams and seasons we can evaluate the robustness and generalization properties of the method.
- **Player Tracking Data**: Incorporating detailed player movement data could offer deeper insights into tactical patterns beyond ball trajectories. It would also increase the granularity of the trajectories, I.e. the number of coordinate pairs would significantly increase in each trajectory.

By addressing these areas for improvement, the clustering model can be refined to provide more accurate and tactically relevant insights, ultimately contributing to better analytical tools in football strategy analysis.

4.3 Final words

Our study demonstrates that clustering football possession chains using the FSDP algorithm with the Fréchet distance can effectively uncover tactical patterns between possession chains. The methodological framework, combining decision graphs, cluster assignment, core and halo distinctions, and silhouette analysis, provided a robust approach to evaluating clustering results.

The findings highlight the team's greater effectiveness when attacking via the left flank, offering actionable insights for coaching strategies. The study underscores the importance of selecting appropriate distance measures that align with the theoretical assumptions of the clustering algorithm and the characteristics of the data.

By revealing underlying patterns in possession chains, this analysis contributes to the broader field of sports analytics, providing tools and methodologies that can aid in tactical decision-making and performance improvement.

A. Appendix: Implementation Details

A.1 Implementation of the Fréchet Distance

The Fréchet distance between two trajectories is computed using the algorithm proposed by Alt and Godau [1], as implemented in the `traj_dist` library [5]. This method constructs a free space diagram for a set of critical values ε and solves the decision problem of reachability within the free space.

A.1.1 Computing Free Space on Cell Boundaries

The algorithm constructs the free space diagram by subdividing each trajectory into segments between consecutive points. This segmentation lets us pair each segment from one trajectory with each segment from the other, forming a "cell" in the diagram. The algorithm then checks the *left* and *bottom* boundaries of these cells to determine which regions are reachable.

Free Space Intervals

Suppose trajectory T_a has points $a_0, a_1, \ldots, a_m$, and trajectory T_b has points $b_0, b_1, \ldots, b_n$. Each cell in the free space diagram then corresponds to a pair of segments:

$$[a_i,\, a_{i+1}] \text{ from } T_a, \quad \text{and} \quad [b_j,\, b_{j+1}] \text{ from } T_b,$$

where $0 \leq i < m$ and $0 \leq j < n$. Within this cell, we define two local parameters:

$$\sigma \in [0,1], \quad \tau \in [0,1].$$

We let $T_a(\sigma)$ be the linear interpolation from a_i to a_{i+1} (i.e. $T_a(\sigma) = (1-\sigma)\,a_i + \sigma\,a_{i+1}$), and $T_b(\tau)$ be the linear interpolation from b_j to b_{j+1}.

Left Free Space (LF).

The left boundary of this cell in the free space diagram corresponds to a fixed value of τ (for example, $\tau = 0$ if we are literally at the left edge in the (σ,τ)-plane. We then solve for all $\sigma \in [0,1]$ such that

$$\left\| T_a(\sigma) - T_b(\tau_{\text{fixed}}) \right\| \leq \varepsilon.$$

If such values of σ exist, they form an interval along $[a_i, a_{i+1}]$ indicating where the distance to $T_b(\tau_{\text{fixed}})$ stays at most ε.

Bottom Free Space (BF).

The bottom boundary of the cell corresponds to a fixed value of σ (e.g. $\sigma = 0$ if at the bottom edge). We then solve for all $\tau \in [0,1]$ such that

$$\left\| T_a(\sigma_{\text{fixed}}) - T_b(\tau) \right\| \leq \varepsilon.$$

This yields an interval along $[b_j, b_{j+1}]$ where the distance to $T_a(\sigma_{\text{fixed}})$ remains within ε.

Here, $\|\cdot\|$ is the Euclidean distance. These intervals computed by the function `free_line` in the `traj_dist` library [5], specify exactly which portions of each boundary are “open” (i.e. within ε).

A.1.2 Reachability Analysis

After computing free space intervals in every cell, the `LR_BR` (Left/Bottom Reachability) function determines whether one can reach the right/top boundaries of each cell. In particular:

- **Left Reachability (LR):** Checks whether the left boundary of a cell is connectable from adjacent cells.
- **Bottom Reachability (BR):** Checks whether the bottom boundary is connectable similarly.

By iterating through these cells and updating their reachability status, we test whether there is a continuous path of “free” points from the diagram’s lower-left corner were $(\sigma,\tau) = (0,0)$ to its upper-right corner were $(\sigma,\tau) = (1,1)$. If such a path exists at threshold ε, then the distance ε is feasible.

A.1.3 Decision Problem and Binary Search

To decide whether a particular ε is valid, we invoke this reachability procedure on the free space diagram. Then, to find the smallest such ε:

1. We collect a set of critical values, e.g., distances between points and opposing segments.
2. We perform a binary search over these critical values. For each candidate ε, we test whether the free space allows a path from $(0,0)$ to $(1,1)$.
3. Once we pinpoint the minimal ε for which a path exists, that ε is the continuous Fréchet distance between the two trajectories.

A.1.4 Illustrative Example

Consider the following trajectories:

$$\begin{aligned} T_a: &\quad a_1 = (34,38),\ a_2 = (30,46),\ a_3 = (66,38), \\ T_b: &\quad b_1 = (30,34),\ b_2 = (46,30),\ b_3 = (50,34),\ b_4 = (66,34). \end{aligned}$$

A.1 shows both the trajectories (left) and the free space diagram (right). In the diagram, each axis nominally runs from 0 to 1 in a formal sense. In practice, each trajectory is indexed by its consecutive segments, producing cells in the diagram that correspond to $[a_i, a_{i+1}]$ vs. $[b_j, b_{j+1}]$. For a threshold $\varepsilon = 12.15$, there is precisely a valid path through the free space from $(0,0)$ to $(1,1)$, so this value equals the Fréchet distance for these trajectories.

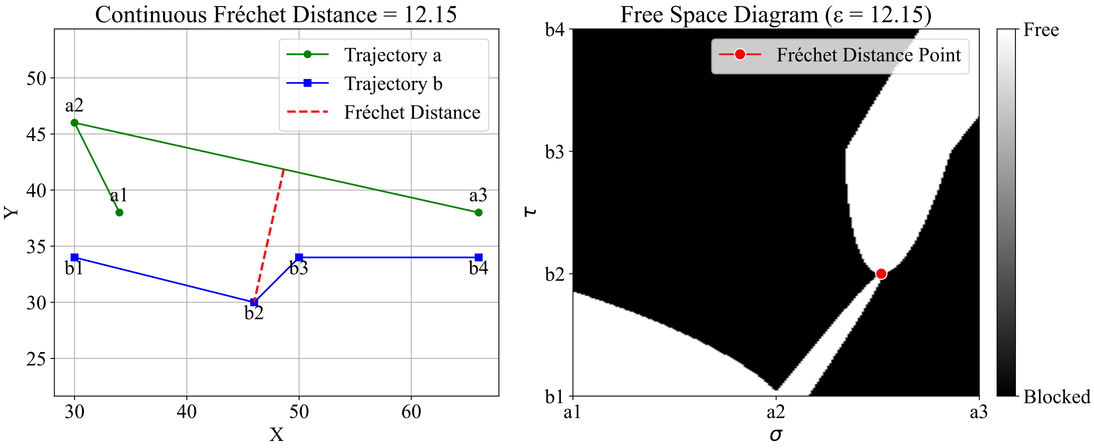


Figure A.1: Example of two distinct ball trajectories T_a (green) with points a_1, a_2, a_3 and T_b (blue) with points b_1, b_2, b_3, b_4. The left plot illustrates the Fréchet distance by the red dashed line between the two trajectories. The right plot shows the corresponding free space diagram for $\varepsilon = 12.15$, which is the smallest ε admitting a valid monotonic path through the white "free" cells—thereby matching the Fréchet distance. The red dot in the free space diagram indicates the parameter location at which this maximum separation (the Fréchet distance) occurs.

B. Appendix: Definitions and Proofs

B.1 Metric distance measure

A metric distance as defined in [10].

Definition B.1. A distance measure d is metric if for all T_1, T_2, and T_3 of a trajectory dataset, the following conditions are satisfied:

1. **Nonnegativity**: $d(T_1, T_2) \geq 0$.
2. **Identity of Indiscernibles**: $d(T_1, T_2) = 0 \iff T_1 = T_2$.
3. **Symmetry**: $d(T_1, T_2) = d(T_2, T_1)$.
4. **Triangle Inequality**: $d(T_1, T_3) \leq d(T_1, T_2) + d(T_2, T_3)$.

B.2 Proof that Even Cluster Distribution Maximizes the Expected Number of Points in Other Clusters

We will show that an even distribution of data points across clusters minimizes $\sum_{k=1}^{K} n_k^2$, and thereby maximizing the expected number of points in other clusters for a randomly selected data point.

Let:

- N be the total number of data points.
- K be the total number of clusters.
- n_k be the number of points in cluster k, where $k = 1, 2, \ldots, K$, with $n_k \in \mathbb{N}$ and $n_k \geq 1$.

The total number of data points is the sum of the cluster sizes:

$$\sum_{k=1}^{K} n_k = N.$$

To compute the expected number of points in other clusters for a randomly selected data point, we have that:

- The probability that a randomly selected data point belongs to cluster k is $\frac{n_k}{N}$.
- Given that the data point is in cluster k, the number of points in other clusters is $N - n_k$.

Therefore, the expected number of points in other clusters is:

$$E[N_{\text{other}}] = \sum_{k=1}^{K} \left(\frac{n_k}{N} \times (N - n_k) \right) = N - \frac{1}{N} \sum_{k=1}^{K} n_k^2.$$

Our goal is to maximize $E[N_{\text{other}}]$, which requires minimizing $\sum_{k=1}^{K} n_k^2$.

B.2.1 Using the Cauchy-Schwarz Inequality

Under the constraint $\sum_{k=1}^{K} n_k = N$, we use the Cauchy-Schwarz inequality to find a lower bound for $\sum_{k=1}^{K} n_k^2$.

The Cauchy-Schwarz inequality states that for any real numbers a_k and b_k:

$$\left(\sum_{k=1}^{K} a_k b_k\right)^2 \leq \left(\sum_{k=1}^{K} a_k^2\right)\left(\sum_{k=1}^{K} b_k^2\right).$$

Let $a_k = 1$ for all k and $b_k = n_k$. Then:

$$\left(\sum_{k=1}^{K} a_k b_k\right)^2 = \left(\sum_{k=1}^{K} n_k\right)^2 = N^2,$$

$$\sum_{k=1}^{K} a_k^2 = \sum_{k=1}^{K} 1^2 = K,$$

$$\sum_{k=1}^{K} b_k^2 = \sum_{k=1}^{K} n_k^2.$$

Applying the inequality:

$$N^2 \leq K \sum_{k=1}^{K} n_k^2,$$

which implies:

$$\sum_{k=1}^{K} n_k^2 \geq \frac{N^2}{K}.$$

Equality holds when there exists a constant c such that $a_k = cb_k$ for all k. Since $a_k = 1$, equality occurs when all n_k are equal:

$$n_k = \frac{N}{K}, \quad \text{for all } k.$$

B.2.2 Conclusion

Therefore, the sum $\sum_{k=1}^{K} n_k^2$ is minimized when all clusters are of equal size:

$$\sum_{k=1}^{K} n_k^2 = K\left(\frac{N}{K}\right)^2 = \frac{N^2}{K}.$$

Substituting back into the expression for $E[N_{\text{other}}]$:

$$E[N_{\text{other}}]_{\text{max}} = N - \left(\frac{1}{N} \times \frac{N^2}{K}\right) = N - \frac{N}{K} = N\left(1 - \frac{1}{K}\right).$$

This demonstrates that an even distribution of data points across clusters maximizes the expected number of points in other clusters. When clusters are unevenly distributed, $\sum_{k=1}^{K} n_k^2$ increases, leading to a decrease in $E[N_{\text{other}}]$. By maximizing $E[N_{\text{other}}]$, we enhance the discriminative power of the clustering because each data point has more points in other clusters to distinguish itself from. In our analysis, this leads to more meaningful and interpretable clustering results.

C. References

[1] Helmut Alt and Michael Godau. Computing the fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(1-2):75–91, 1995.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[3] Andrew Borrie, Gudberg K. Jonsson, and Magnus S. Magnusson. Temporal pattern analysis and its applicability in sport: an explanation and exemplar data. *Journal of Sports Sciences*, 20(10):845–852, 2002. PMID: 12363299.

[4] Joachim Gudmundsson and Michael Horton. Spatio-temporal analysis of team sports - A survey. *CoRR*, abs/1602.06994, 2016.

[5] Benjamin Guillouet. Trajdist: A python library for computing distances between trajectories. urlhttps://github.com/bguillouet/traj-dist, 2021. Version 1.0.3, Python library for computing various trajectory distances.

[6] Michael Horton. *Algorithms for the Analysis of Spatio-Temporal Data from Team Sports*. University of Sydney School of Information Technologies, Faculty of Engineering and Information Technologies, 2018. `http://hdl.handle.net/2123/17755`.

[7] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.

[8] James Savage Michael Barnard, Calum Ross and Christopher Winn. *Deloitte Annual Review of Football Finance*. Deloitte, 26th edition, 2017.

[9] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

[10] Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. A survey of trajectory distance measures and performance evaluation. *The VLDB Journal*, 29(1):3–32, oct 2019.

[11] Yaguan Tao, Alan Both, Rodrigo I. Silveira, Kevin Buchin, Stef Sijben, Ross S. Purves, Patrick Laube, Dongliang Peng, Kevin Toohey, and Matt Duckham. A comparative analysis of trajectory similarity measures. *GIScience & Remote Sensing*, 58(5):643–669, 2021.

[12] Busra Tas Kiper. *Contemporary developments and applications of unsupervised machine learning methods*. PhD thesis, Stockholm University, Department of Mathematics, 2024.

[13] Michalis Vlachos, George Kollios, and Dimitrios Gunopulos. Discovering similar multidimensional trajectories. In *Proceedings of the 18th International Conference on Data Engineering*, pages 673–684. IEEE, 2002.

[14] Qing Wang, Hengshu Zhu, Wei Hu, Zhiyong Shen, and Yuan Yao. Discerning tactical patterns for professional soccer teams: An enhanced topic model with applications. 08 2015.