# Evolving the Automated Search for Clusters of Similar Trajectory Groups

Friedemann Schwenkreis
Business Information Systems
Baden-Wuerttemberg Cooperative State University
Stuttgart, Germany
email: friedemann.schwenkreis@dhbw-stuttgart.de
ORCID: 0000-0003-4072-0582

***Abstract*— The work presented in this paper builds upon a previous approach to automatically detect tactics based on spatiotemporal data in the context of team handball. It will be shown how the availability of additional data allows us to verify the principal approach. However, it will also be shown that the previous approach for choosing parameters of the applied methods was suboptimal, and an application-oriented approach based on heuristics helps to improve the results significantly. Basically, the combination of Shared Nearest Neighbor Clustering and the search for frequent itemsets is used to find clusters of trajectory groups. These basic methods are enhanced by special notions of distance and cluster quality indexes which allows to find optimal parameter settings for the specific application scenario. Furthermore, an approach is presented to use the existing "composite model" to determine the cluster to which a group of trajectories belongs to (application of the composite model).**

***Keywords- trajectory sets; SNN clustering; frequent itemsets; tactics recognition.***

## I. INTRODUCTION

Previous work proposed using data from position tracking systems, such as Vector from Catapult or from Perform LPS by Kinexon to automatically process the position data of players of team ball games [1][2]. Schwenkreis proposed a deep learning-based classification approach to automatically recognize team tactics based on the abovementioned spatiotemporal sensor data [3]. The approach was subsequently modified to avoid the large amount of necessary training data and thus the need for labeling data [4]. The proposed solution was to use clustering based on the Fréchet distance [5] of trajectories combined with a silhouette coefficient-based [6] quality criterion to cope with noise. A subsequent paper enhanced the approach by avoiding the shortcomings of the Fréchet distance and eliminating the need for generating ordered sets [7]. Furthermore, the enhancement avoids the need for a distance criterion of sets of trajectories by introducing a combination of clustering and the search for frequent itemsets.

In his latest paper, Schwenkreis explicitly identified the need to collect additional trajectory data to extract a stable set of groups of trajectory sets. However, there was no discussion regarding how to incrementally improve the model or how to determine a stable state. Furthermore, the actual objective of identifying frequent sets of trajectories is to identify tactical patterns that can later be used to automatically detect them in streams of trajectories. Hence, there needs to be a mechanism to decide whether a set of trajectories belongs to one of the previously identified clusters.

Based on the previously introduced basic mechanisms for extracting a cluster model of trajectory groups from spatiotemporal data, this paper will present the latest developments of the extraction of a cluster model. Furthermore, this paper presents an approach for applying the extracted cluster model to determine whether trajectory groups extracted from a stream of trajectory sets belong to one of the previously identified clusters.

An overview of related work is given in Section II. Then, Section III will introduce the underlying data model in Section III.A and the general clustering approach in Section III.B. An abstraction is introduced that allows the application of the approach in arbitrary situations in which similarity clusters are detected in sets of trajectory groups. In Section IV, the necessary distance functions, the quality criteria for clusters, and the assignment to clusters are discussed. Section V presents the results of an evaluation based on a real-world application of the approach. The paper is concluded with a summary and an outlook on the future application of the approach in Section VI.

## II. RELATED WORK

### A. Pattern Recognition

Pattern recognition in the context of spatiotemporal data has a long history [8], and a significant number of related studies have been published in the area of trajectory clustering. Trajectory clustering provides the foundation for recognizing patterns in sets of team moves. However, since team moves consist of groups of trajectories, the targeted problem of tactic recognition is not identical to the family of problems that is addressed by classical trajectory clustering. Nevertheless, recent trajectory clustering approaches, such as those described in [9], have made significant progress in the area of trajectory clustering, and the work presented in this paper has been influenced by these approaches.

In particular, the flock pattern and later generalizations to the convoy pattern seem to overlap with the problem addressed by this work [10]. Unfortunately, there are significant differences that prevent the direct application of these approaches:

- Current convoy mining approaches assume that clusters are searched in sets of points that have been

collected at the same point in time. This is not the case for the class of problems discussed in this paper. In contrast, the task is to search for clusters of trajectories that have not(!) "happened" at the same point in time. Furthermore, there is no fixed mapping on a common logical clock that would allow us to treat the coordinates of trajectories as if they were collected at the same time.

- The usual approaches to trajectory clustering are based on the notion of density and use a density threshold to determine the clusters. The basic assumption of these approaches is that a single density threshold can be found, which is not possible in the given case. By inspecting the application area, it is known that the density of the trajectories significantly differs across different trajectory clusters (see also Section IV.B).
- Convoy mining approaches assume that points of trajectories belonging to the same cluster also belong to a single cluster. In the given application scenario, this does not need to be the case. In terms of convoy mining approaches, points of the same trajectory may join other convoys and return to the original convoy because trajectories belonging to different clusters may have non-empty intersections from a geometrical point of view because they have identical points.

The field of detecting optimal care pathways in health care [11] has some similarities to detecting team tactics based on an abstract notion of trajectories. Clustering approaches in this area are based on a completely different notion of distance [12]. Furthermore, the requirements regarding the temporal distance of "locations" differ significantly, which leads to methods that are based on the sequence only rather than considering the real distances in time. Hence, the work in that area cannot be applied in the given context.

### B. Sports Analytics

Recently, the analysis of spatiotemporal data in the context of sports has attracted increasing attention. There are several attempts in the area of team sports to exploit the data that are produced by the position sensors carried by players [13]- [15]. Several activities focus particularly on soccer (or football) to extract models that help to explain the mechanics of the game [16]. This is because professional soccer teams can fund analysis projects, and there is still no accurate model for computing appropriate predictions. Some work can be found in the literature that derives patterns from spatiotemporal data, but these approaches use classification to predict, for example, ball losses or scoring probabilities because in these cases, the target value is available in the automatically collected data. This is not the case for tactical labels but essential for the case presented in this paper.

Unfortunately, the mentioned work does not focus on detecting patterns of moves of groups of players. The reason is that soccer and other team sports significantly differ from team handball in terms of the speed of attacks. In the case of team handball, it is crucial that the individual moves (and resulting positions) of teammates are known upfront by the other players because, in most cases, the determination of the location of other team members by an explicit visual observation is too slow. Thus, the coordination of the players' moves is trained based on explicitly communicated movement patterns (called tactics). Ice hockey has some similarities to team handball because the players' speed is even greater than that of team handball players. However, tactics are focused on the movement patterns of individual players rather than on the coordinated patterns of a whole team, which has also been reflected in recent work that targets the analysis of spatiotemporal data in the context of ice hockey [17].

## III. BASICS

### A. Underlying Data Model

#### 1) Individual trajectories

As proposed in the aforementioned previous work, the 2D coordinates delivered by tracking sensors are “normalized” to avoid differences due to changing directions of play [4]. Given an observation interval of $t_2$-$t_1$ seconds and a position sampling rate $f$, the individual trajectory $T_s(t_1,t_2)$ of a sensor is defined as the timely ordered set of $r=(t_2-t_1)*f$ coordinate pairs $p$: $T_s(t_1,t_2)=(p_1, p_2, \dots, p_r)$. It is assumed that all the observed sensors generate samples at the same rate in the given application context. Furthermore, there is a mapping for individual trajectories $T_s$ that assigns a team $O$ to the individual trajectory: $team(T_s)$: $T_s \rightarrow O$.

#### 2) Team Moves: Sets of trajectory class identifiers

The notion of a team position has been defined by Schwenkreis as a vector of positions of the contained sensors (team members), which is similar to the coordinate of a team in a $2n$-dimensional space, where $n$ denotes the number of sensors [3]. The challenge of this approach is to associate a specific sensor with a well-defined position in the vector. This approach is challenging because the collection of sensors that comprises a team is not constant and might change due to the substitutions of team members. Originally, all possible permutations of a vector were generated when training a model (which is rather costly) [3]. Generating permutations is avoided in subsequent work by introducing a so-called canonical sort order of sensor positions contained in a team position [4]. The sorting order is based on additional information regarding the individuals who are carrying a sensor. As a result, there is a unique mapping of sensors to positions in a team vector.

Alternatively, to the approach above, a core assumption regarding individual trajectories can be exploited. Individual trajectories are not randomly distributed across the feature space. There are rather clusters of similar trajectories that are intentionally followed. Hence, there exist only a limited number of trajectory classes, each representing an *intended trajectory* given a certain context. In application terms, this can be called the *intended individual contribution* given an intended team tactical move. Based on this assumption, individual trajectories $T_s(t_1,t_2)$ can be mapped onto identifiers of intended trajectory classes: $T \rightarrow c$, $c \in \mathbb{N}$. There might be cases where individual trajectories do not match with any intended trajectory class. In these cases, the individual trajectory is mapped onto the “noise” class identifier

represented by a value of *-1*. As a result, a team tactical move *M* of a time interval can be defined as a tuple of trajectory class identifiers of the time interval $M(t_1,t_2)=(c_1, c_2, \ldots, c_n)$ with *n* in the range of one to the number of sensors belonging to the observed collection, also called the team or group size. The sorting order of the class identifiers contained in *M* is irrelevant. Thus, we simply assume that the class identifiers are given in descending order: $\forall c_i, c_k \in M : i<k \rightarrow c_i >= c_k$.

### B. Clustering Aspects

#### 1) Two-Step Approach

The automated detection of tactics based on clustering was proposed in [4] to avoid the need for labeling data. The approach did not explicitly select a clustering technique but introduced a quality criterion to compare different techniques. The mentioned approach uses spatiotemporal data that comprise groups of trajectories of team members to search for similar team moves by clustering. In a given example application scenario (team handball), this results in records of 1.760 attributes (880 pairs of 2-d coordinates) per team move [4].

This number of attributes is rather high, and a number of clustering approaches have been described in the literature to reduce the dimensionality of the data, particularly in the context of time series data (such as trajectories) and [18]. A special group of these approaches is the set of multilevel or multistep clustering methods, which (from a high-level perspective) follow a stepwise approach to reduce the dimensionality of the data by clustering a "sub-aspect" first. The sub-clusters are then clustered again on the next level. For example, Aghabozorgi et al. introduced a two-step clustering approach for time series data by starting with a fine-granularity cluster search, which is followed by a subsequent clustering step to merge similar clusters using a different criterion that is specific for the next level [19].

The basic idea of two-step clustering is adopted for the case of this paper to address the dimensionality challenge. Rather than directly trying to find clusters in a set of trajectory groups, it is proposed to search for clusters at the trajectory level first, given a trajectory-specific distance criterion. Subsequently, a search for similarity clusters of trajectory group clusters (denoted as team tactics) is performed. Thus, rather than having to find clusters based on *2nk* attributes (when *n* is the number of positions of the trajectories and *k* is the number of trajectories per group), the clustering of the first step has to find clusters in records with only *2n* attributes. The subsequent step has to handle records consisting of only *k* attributes. Projected on the application case of [4], there are only 220 rather than 1.540 attributes on level one and 7 attributes on level two (each representing a the individual move of player of a team).

The two-step approach is particularly promising for trajectory groups because it usually provides a meaningful explanation on the application level. The first clustering searches for patterns of individual contributions, while the second search focuses on patterns of combinations of individual efforts. In application terms from team ball games: what are the intended moves of players (or player types), and how are team tactics composed of these individual moves?

#### 2) Clustering of trajectories

There is always "noise" in the trajectory data in the given context because there will always be player moves that are not intended moves in the sense of a contribution to some tactics. Thus, only clustering techniques can be used that can cope explicitly with noise, which excludes, for instance, basic spectral clustering [20]. Even later enhancements of spectral clustering called robust spectral clustering can only handle a low number of noise points compared to the number of non-noise points [21]. Furthermore, no assumption regarding the cluster shape can be made. Trajectory clusters might have concave boundaries, which excludes clustering techniques, such as k-means clustering. Based on this, only two clustering concepts have been further investigated: agglomerative hierarchical clustering [22] and density-based spatial clustering of applications with noise (a.k.a. DBSCAN) [23]. However, agglomerative hierarchical clustering can be simulated using particular parameter settings of DBSCAN. Thus, this paper focuses on DBSCAN only.

#### 3) Finding similarity groups of team moves

As described in Section III.A, team moves are represented by ordered *k-tuples* of trajectory cluster identifiers. To find groups of similar team moves, another clustering step can be used, but we lack a meaningful notion of distance for team moves. A straightforward approach would be to use the Hamming distance (based on [24]), as in the case of distances of words, which has no meaningful interpretation in the context of team moves.

Alternatively, the search for similarity groups can be performed based on the method of searching for *frequent itemsets,* as is done in the case of association rule mining [25]. With this approach, all *frequently occurring* combinations of previously extracted cluster identifier combinations will be found without the need for a distance or similarity criterion. However, not every previously identified trajectory cluster is relevant when identifying team tactics. For instance, there are clusters that represent player trajectories in which the players (almost) do not move at all. These trajectory clusters can be seen as *passive* contributions to a team tactic rather than *active* contributions. Consequently, the trajectory clusters must be *weighted* based on the distance covered by the contained trajectories to reflect the contribution to a team tactic.

When weighting trajectory clusters, the search for frequently occurring clusters needs to take weights into account, which is comparable to the process of searching weighted itemsets. In previous work in the area of weighted itemsets, the weights became somewhat part of the notion of frequency [26]. That is, the low weight of an itemset can be "compensated" by high support to still have a frequent itemset and vice versa. In the given application scenario, this is not the case. A trajectory cluster with a low weight is considered to be of low relevance regardless from its frequency. Even a high support of the cluster will not "make it more relevant". In application terms, if a player does not move, there is no relevance of the trajectory with respect to a team tactic, no matter how often this occurs.

The weight of a trajectory cluster is defined as the length of the trajectory (sum of the Euclidean distances of the contained points) representing the containing cluster $t_r^c$. The

representative trajectory of a cluster is defined as the trajectory with the minimal distance to all other trajectories of the same cluster: $t_r^c = t_i | \forall\, t_i, t_k \in c\colon\ D_i^c \leq D_k^c$ and $D_i$ is the sum of all distances of a trajectory of a cluster to any other trajectory of the same cluster $D_i^c = \sum dist(t_i, t_k) \mid t_i, t_k \in c$.

The *relevance coefficient* $r_i$ is assigned to the tuples $t_i$ representing team moves: $t_i \rightarrow r_i,\ r_i \in \mathbb{N}_0$. The relevance coefficient represents the number of contained trajectory cluster identifiers that identify a cluster whose representative trajectory $t_r^c$ has a length greater than a specified threshold. The search for *relevant frequent itemsets* identifies the sets of trajectory cluster identifiers that have a support $s_i$ greater than a given minimum support and a relevance greater than a given threshold: $\{c_i\} \mid s_i > s_{min} \wedge r_i > r_{min}$.

The Apriori approach to finding frequent itemsets [25] can be easily extended to cover cases with a relevance coefficient. The basic idea of Apriori is that the support of an itemset containing a certain number of items cannot be greater than the support of any subset containing fewer items. The relevance coefficients of team moves do not have this property in general because the relevance coefficient of an itemset cannot exceed the number of contained items. Hence, the straightforward approach is to use regular Apriori to generate the frequent itemsets, which are subsequently checked for their relevance based on the assigned relevance coefficient of the contained trajectory clusters and the specified minimum relevance. After the identification of the relevant frequent itemsets, itemsets containing non-relevant items can be eliminated to focus on team moves with relevant trajectories only.

The straight-forward approach can be improved by using the relevance as a sort criterion of the items contained in an itemset (the itemset becomes a tuple). As introduced in [26], itemsets can be treated as sorted sets (tuples) based on the decreasing relevance of the contained items. The candidate generation then combines only trajectory cluster identifiers that represent a cluster whose representing trajectory has a length greater than the specified threshold (which means a contribution to the relevance coefficient greater than zero), and candidate itemsets with nonrelevant items are *pruned*. Finally, the resulting itemsets need to be checked for the minimum relevance limit and support.

*4) Assigning team moves to itemsets*

To group the team moves, a mapping of each team move $t_i$ onto one identified relevant frequent itemset $f_k$ is needed: $t_i \rightarrow f_k$. A naïve approach would be to directly assign an itemset to any team move that supports the itemset. Unfortunately, this simple association is ambiguous because itemsets can have a subset relationship, and a single team move might even support multiple itemsets not having a subset/superset relationship. The latter case is an indication of not having enough data to be able to identify the "missing" superset of the union of the supported itemsets as relevant and frequent. The association of a team move to any of the itemsets can be chosen arbitrarily in this case. The case of nested itemsets is rather simple. A team move should be associated with the relevant frequent itemset that consists of the maximum number of items that is supported by the team move. It represents the specialization of another tactical move—the subset.

## IV. Distances, Similarity, and Quality Indicators

### A. Trajectory Distance

A distance or similarity function for individual trajectories is needed to be able to find clusters of similar trajectories in the absence of a labeling attribute (the ground truth) in the data. In previous work, the discrete Fréchet distance [5] was used as the distance between two trajectories without an in-depth discussion of alternatives. In a more recently published comparison of trajectory distances, it was shown that the discrete Fréchet distance is sensitive to outliers and to timely shifts in trajectories [27]. It is also shown that dynamic time warping [28] outperforms the Fréchet distance in scenarios that are similar to the scenario addressed by this paper. However, dynamic time warping is not a metric (missing the triangle inequality property), which limits its applicability. Fortunately, the used approaches do not rely on the triangle inequality property because of their independence from path-length based criteria.

Continuous dynamic time warping was not covered by the comparison but was identified in later work as the most flexible distance criterion for trajectory distances in [29]. Continuous dynamic time warping was derived from dynamic time warping to cover cases in which the discretizations of trajectories are not uniform. However, in the cases addressed in this paper, uniform discretization can be guaranteed because the individual trajectories consist of the same number of coordinates distributed evenly over time. Thus, continuous dynamic time warping has no advantage compared to the original dynamic time warping approach in the given scenario. Paparrizos et al. argue against the use of dynamic time warping when clustering time series data [30]. However, this process is performed based on a generalized case without considering specific cases, such as a time series of two-dimensional position data. The concerns raised in the paper do not hold in this specific case.

The advantage of dynamic time warping compared to the discrete Fréchet distance is the concept of "warping", which tolerates time shifts between the coordinates of two trajectories. Furthermore, the discrete Fréchet distance focuses on the maximum distance between pairs of coordinates, while the dynamic time warping distance is the sum of all distances between matching pairs, thus smoothing the effect of outliers. In conclusion, dynamic time warping is the optimal method for determining trajectory distances in the context of this work. However, the dynamic time warping distance has been slightly adapted to avoid a dependency on the number of points a trajectory consists of. Rather than the sum of all distances, the average distance is used.

The optimal warping window size is highly application dependent. It depends on the accuracy of the frequency of the position detection technology as well as the absolute speed of the sensors. Furthermore, the notion of similarity of a given context limits the time gap that is tolerated when two trajectories are compared. In case of team handball moves, the time gap must be in the sub-second range. Non-

comprehensive experiments with a tolerable time gap of up to half a second have shown acceptable results.

### B. Shared Nearest Neighbor Trajectory Similarity

Distance-based clustering algorithms, such as DBSCAN, look for dense areas based on a single distance-based threshold. Consequently, algorithms cannot cope well with areas with varying densities. In the case of varying densities, clusters with lower density are not found if the distance threshold is set too low. On the other hand, if the distance threshold is too high, then multiple clusters with high density might be merged, and additional details may be lost.

Unfortunately, the application scenario of this work must explicitly handle varying densities because the running distances of different player roles (positions in a team) and thus the DTW distances differ significantly. The so-called *shared nearest neighbor similarity* is an approach for handling varying densities while still using the original notion of distance as the underlying criterion [31]. Shared nearest neighbor similarity uses a notion of similarity that depends only indirectly on distance. Conceptually, the approach computes a list of nearest neighbors for each record based on the chosen notion of distance (the dynamic time warping distance in the case of this paper). When the similarity of two records is computed, prefixes of length $l$ (a user-specified limit) of the records' lists of nearest neighbors are compared. The number of nearest neighbors contained in both lists is the value for the similarity of the two records. Then, a DBSCAN-like clustering approach searches for clusters based on similarity values.

The notion of similarity has a significant limitation: the similarity value depends on the number of points that are compared. Thus, the notion of similarity has been adapted as in the case of the dynamic time warping distance. The Jaccard coefficient is an alternative notion of similarity that "normalizes" similarity with the number of points considered: $J(X,Y) = |X \cap Y|/|X \cup Y|$ [32]. Hence, its value is in the interval of $[0,1]$ independent of the size of the compared sets. Furthermore, it can be easily converted into a distance by subtracting it from 1, which allows us to use a "standard" subsequent DBSCAN approach to search for clusters.

Interestingly, the basic concept of shared nearest neighbor similarity is analogous to the "sparsifying" approach used by Laplacians for robust spectral clustering [21]. Since computing the nearest neighbor similarity also "reduces" the noise in the original data, it might be interesting to compare the results of a subsequent DBSCAN with the results of a subsequent robust spectral clustering.

### C. Quality indicators

#### 1) Generalized Dunn index

There are a multitude of quality coefficients for clustering [33]. These parameters are particularly important for finding the optimal parameter settings for clustering methods when no *ground truths are* available. In the context of the work presented in this paper, there is no upfront knowledge regarding similarity groups of team moves. Clustering is explicitly used to find representatives of groups that will be used by experts to label the tactics used in application terms.

At least two aspects need to be considered when selecting a quality indicator for the extracted clustering model in the context of the presented work. Clustering methods that handle noise explicitly, such as DBSCAN, assign noise records to a separate noise cluster that must be excluded from the calculation of a quality indicator value. As a result, there are two extreme cases. In the first case, the parameter settings are chosen such that all the non-noise points are assigned to a single cluster or very few clusters. This is, for instance, the case when the search radius for similar points of DBSCAN is too large. The second extreme is the case when the search radius is rather small, such that most of the found clusters consist of only a single data point and are thus treated as noise. Consequently, there are only a few but well-separated clusters. The first case is indicated by a low number of noise points and clusters with a large distance between the contained points, while the latter case has only a relatively small number of non-noise points and a very small distance between the contained points.

Originally, Dunn introduced the idea of using the ratio of the *diameter* of a cluster to the distance to the closest neighboring cluster as a quality indicator for a clustering model [34]. This idea was later generalized by Bezdek and Pal, who introduced several notions of diameter (intra-cluster distance) and distance (inter-cluster distance) denoted as the Generalized Dunn Index (GDI) [35]. Since the chosen clustering approach does not compute any centroids, centroid-based variants have not been considered. To avoid oversensitivity to outliers, the average distances of the intra-cluster distances and the maximum distances of the inter-cluster distances were chosen as the underlying values to compute the GDI, which is also denoted as *GDI 2-2*. The Generalized Dunn Index is always positive, and the higher the value is, the better the clustering.

#### 2) The side effect of excluding noise

The described clustering approach based on nearest neighbor similarity has 3 main parameters that can be varied to find an optimal clustering for a set of trajectories:

- The number of neighbors used to determine the similarity: The smaller the considered number of neighbors, the smaller the set of neighbors with a Jaccard coefficient greater than epsilon. The number of points treated as noise increases.
- The ε limits the ability to find "close" points: A small epsilon of DBSCAN results in small sets of close points that can be assigned to the same cluster. Consequently, the number of points treated as noise increases.
- The minimum number of points needed to form an initial cluster: A small number of close points results in many identified clusters. As a result, the number of points treated as noise increases because the cluster size decreases.

All three parameters have a direct impact on the clustering model and the quality indicators not only by resulting in differing numbers of clusters and sets of contained points but also because the number of points treated as noise is directly impacted. This is also reflected by the quality indicators. If a single parameter is varied from low to high, we obtain the

same "behavior" for the quality indicators. Similarly, the silhouette coefficient and GDI 2-2 increase with an increasing parameter value, while the Davies–Bouldin index decreases with increasing parameter value (and the inverse Davies–Bouldin coefficient increases as well).

No indication of an optimal parameter setting can be derived from the course of the indicators' graphs. This is caused by the overlapping effect of a changing number of clusters and an increasing number of points considered noise; thus, these clusters are excluded from the quality indicator values. Hence, it is necessary to take the number of points considered valid into account as well as the number of clusters by weighting the clustering quality indicator. The basic concept of quality indicator weighting was introduced in [4].

A straightforward approach for weighting the quality indicator value is based on two simple observations (heuristics). Given that two clustering models have the same base quality indicator value, a clustering model consisting of more non-noise points is preferable because it represents more information of the input data. Second, if two clustering models have the same quality indicator values and the same number of non-noise points, then a model consisting of more clusters is considered preferable because it potentially allows for better differentiation of cases.

Simple weighting with the number of non-noise (or valid) points $N_v = |\{t_i^c\}|$ results in a weighted quality indicator whose value depends on the sample size. Thus, the relative number of non-noise points based on the size of the input sample $N=|\{t_i\}|$ is used rather than the absolute input size: $n_v = N_v/N$. Using the absolute number of identified trajectory clusters $G^C=|\{c_j\}|$ as an additional weight would overemphasize the importance of the number of clusters. Using the maximum number of clusters to normalize $G^C$ would require knowing this number upfront. Thus, the ratio of the number of clusters to the sample size is used as the weight that represents the number of clusters: $g^C=|\{c_j\}|/N$.

A weighted clustering indicator value $q_i^w$ can now be defined as the product of the original indicator value $q_i$ and the two weights introduced in the previous section: $q_i^w = q_i\, n_v\, g^C$.

## V. Applicability Study

### A. Complexity and Runtimes

The concepts presented in this paper have been used in a real-world scenario of sports. In collaboration with a first league team and the first German Handball Bundesliga, *HBL*, the position data of all matches of the selected team in 2022, 2023, and first half of 2024 were collected as a basis for identifying the offensive tactics they played. The future objective is to be able to detect the played tactics of a team, which can then be used to automatically determine the performance of played tactics for teams and players.

The data consisted of 82 matches, from which a total of 12,366 team moves were extracted before a scoring attempt. A total of 3,020 moves of the 12,366 were offensive moves of the selected team, from which 23,674 trajectories were extracted. Since so-called fast break attacks are not of interest in the context of tactic recognition and some trajectories contain erroneous data, the data for the analysis were reduced to 2,089 team moves with 16,277 trajectories.

The practical evaluation was performed using MathWorks MATLAB™ R2024b version 24.2.0.2712019 running on Ubuntu 22.04.5 using a virtual machine with 16 vCPUs equipped with 32 GB of main memory.

The runtimes of the different computation steps of a complete single run with the "optimal" parameter settings (see Section V.B) are listed in TABLE I. The most time-consuming step of the data preparation is the calculation of the dynamic time warping distances for each pair of trajectories, which is of complexity $n^2$-$n$ or $O(n^2)$ when $n$ denotes the number of trajectories. The distance calculation itself is implemented using the classical dynamic programming approach with a complexity of $mw$ with respect to $m$ as the number of points contained in the trajectories and $w$ as the window size, which results in an overall complexity of $n^2mw$ (if $w$ is defined as a ratio of $m$, then this results in $O(n^2m^2)$).

The computation of dynamic time warping distances is the most time-consuming step of a clustering run, even when searching for optimal parameter settings consisting of repeating subsequent steps. Thus, parallelizing the computation of the DTW distances helps to reduce the overall computation time significantly. Furthermore, it is advisable to use only the computed distances rather than the original trajectory data in the subsequent steps.

The second most time-consuming step after the computation of the dynamic time warping distances is the computation of the shared nearest neighbor similarity. Finding the $k$ nearest neighbors ($k \ll n$) based on the previously computed distances is a set of $nk$ simple searches of $n$-$1$ distances when $n$ denotes the number of trajectories. However, computing a full matrix of nearest neighbors might be advantageous when variations in $k$ need to be computed (see the following Section V.B). This is particularly needed when varying the optimal number of trajectories used to compute the shared neighbor similarity (see Section IV.C.2). The subsequent calculation of the Jaccard similarity coefficients to determine the value of the shared neighbor similarity is of complexity $nk^2$ and can be easily parallelized.

The runtime of the subsequent search for relevant frequent itemsets is negligible given the runtimes of the previous steps. However, this heavily depends on the number of relevant frequent items found.

TABLE I. Measured runtimes of computation steps

| Computation Step | Runtime in seconds |
|---|---|
| Reading and filtering data | 17 |
| Computation of DTW distances | 6684 |
| Computation of similarities | 926 |
| Single DBSCAN clustering | 24 |
| Search for itemsets | 3 |

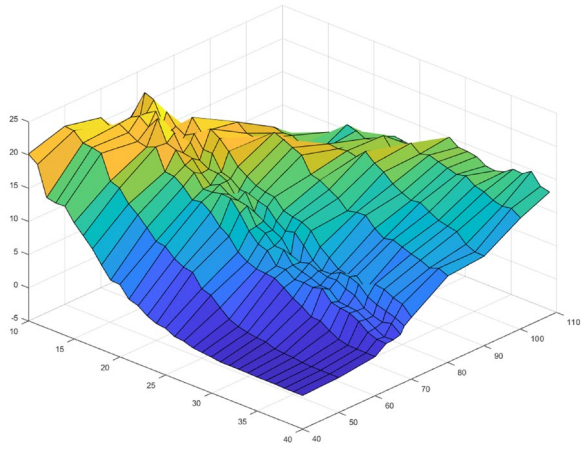

Figure 1. Weighted GDI 2-2 values with variations of the similarity window and the minimum number of points needed to be core

### B. Optimal Parameters and Results

There are three parameters that can be varied while searching for trajectory clusters:

- The number of nearest neighbors that are checked to determine similar trajectories (*similarity window*).
- The minimum number of points that need to be close to be considered the core.
- The limit of the Jaccard coefficient that is used to determine "close" trajectories.

Rather than assuming that all of the mentioned parameters can be chosen arbitrarily, as indicated in [7], this paper follows a different approach. It is assumed that the ability of the Jaccard coefficient to identify close points is application dependent. In the given application context of trajectories of team handball players, we assume that at least 50% of the neighbors of two trajectories need to be "shared" in terms of the shared nearest neighbor approach to be considered "close trajectories". This translates into a minimum Jaccard coefficient of 0.40 (DBSCAN ε of 0.60).

Unfortunately, the similarity matrix directly depends on the number of nearest neighbors that are checked to determine similar trajectories. Hence, for each value of the similarity window, a similarity matrix needs to be computed, which is fairly time-consuming, as described in the previous Section V.A. However, to determine the optimal number of nodes in the similarity window, we compared 12 different cases.

The last parameter that was varied was the minimum number of similar trajectories for being core in terms of DBSCAN. This parameter was varied in the range of [10, 40] with the assumption that at least 10 close trajectories are needed to be considered core. Figure 1 depicts the weighted GDI 2-2 values when varying the similarity window and the limit for the minimum number of points to be considered core points in the sense of the DBSCAN algorithm. The colors indicate ranges of similar index values. Yellow is the color for the highest range, while blue is the color associated with the lowest range of index values. An interesting observation is that the diagonal direction is the same for the same levels of index values. It seems that a decreasing lower limit for core points can compensate for the effect of an increasing similarity window to some extent.

The global maximum of the weighted GDI 2-2 value is at a similarity window of 67 trajectories, and the minimum is 12 necessary points for a core. The value of the weighted GDI 2-2 peaks at 23.27 (based on a GDI 2-2 value of 1.05). The nearest neighbor similarity clustering identified 39 trajectory clusters that represented approximately 58% of the input trajectories. Approximately 42% of the trajectories are identified as noise. After coding the 1,757 team moves using trajectory cluster identifiers, 1,625 team moves with two or more non-noise trajectory cluster identifiers remained. A team move that contains fewer than two non-noise trajectory cluster identifiers is considered an individual move rather than a team tactical move.

The search for relevant frequent itemsets was performed using the 1,625 team moves of the previous step. The lower length limit was set to 3.0, which means that a trajectory is relevant only if the length of the trajectory is greater than 3.0 meters. This is an application-dependent limit and might differ between application scenarios. The absolute minimum support was set to 10, which means that an itemset is frequent if ten team moves support it. The search for frequent itemsets identified 143 itemsets of length 2, 41 itemsets of length 3, and 7 itemsets of length 4.

While 6,790 of the 16,277 trajectories were associated with a trajectory cluster, 1,562 team moves of 2,089 (approx. 75%) were associated with a team move group (or cluster of trajectory sets). 185 of 191 frequent itemsets were used to identify team move groups. The 6 "unused" frequent itemsets result from the criterion that was used to assign a frequent itemset with a team move (see Section III.B.4).

### C. Application-level evaluation

To evaluate the results on an application level, the representative trajectories of each associated itemset were extracted. The contained trajectories were then visualized in a video presenting the tactical view and shown to team handball experts (coaches of the first league teams) to decide whether an actual team tactic was detected by the system and how the identified team tactic could be named. Figure 2 shows a snapshot of three example animations that have been presented to the coaches with their associated names. The seven offense players are depicted as green diamonds. Their trajectories are depicted as "sequences" of green diamonds. To depict the time aspect, the color of the diamonds starts with dark green and ends in bright green. Since the data have been transformed, such that attacks always occur from left to right, only the right half of the field is depicted.

The coaches were able to confirm that the extracted similarity groups actually represent team tactics rather than an arbitrary collection of team moves. The evaluation of the trajectory clusters was successful as well. The trajectory clusters clearly represented the different positions (roles) of the players whose trajectories were contained in a cluster. Some of the detected tactics were considered similar when

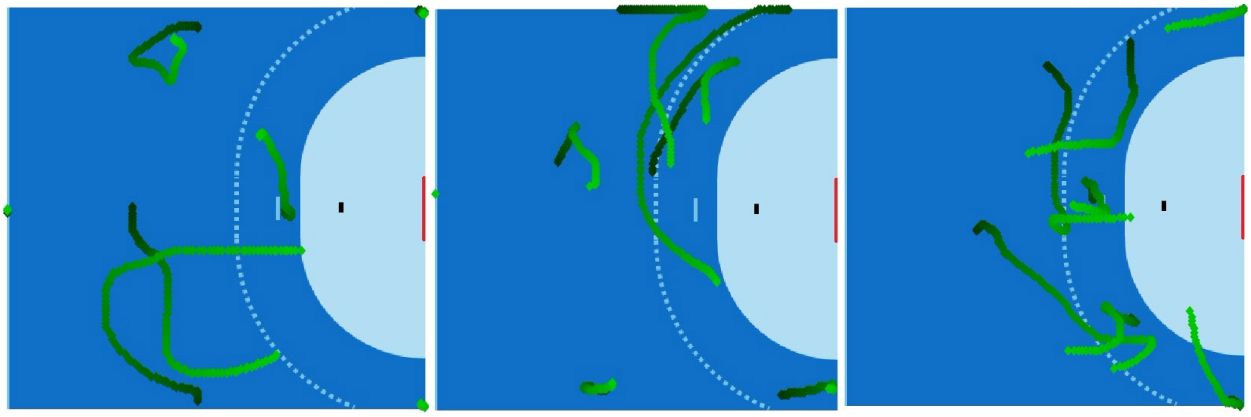

Figure 2. Three snapshots of animation videos of team move clusters: “empty crossing right”, “runner from position 1”, and “7 versus 6”

evaluated by human experts. A detailed analysis of the significant properties of these clusters needs to be performed to determine the key differences between the clusters. If the differences are correlated with the success or nonsuccess of the attacks, the analysis will help to identify the critical aspects of the realized tactics.

### D. Development of a stable cluster model

Originally, 32 matches were used to evaluate the clustering approach, and it has been observed that the dataset was too small to extract a comprehensive set of trajectory clusters [7]. Thus, the approach is continuously evaluated with larger data sets. The results presented in this paper are based on the data of 82 matches. The originally identified trajectory clusters have been confirmed to a vast extent. However, some smaller clusters have been merged, which can be explained by the availability of additional data points that “bridge” the distance between the small clusters.

We assume that the set of identified trajectory clusters will eventually reach a stable state when enough trajectories can be used to extract the cluster model. Since we still observe significant changes in terms of clusters and the number of clustered trajectories, this stable state has not yet been reached. When a stable state has been reached, the number of team moves can be used as a “team move window size” to continuously extract models from the current data. These models can then be compared with previous models to detect significant changes in the set of clusters indicating significant changes in the applied tactics.

### E. Using the model to automatically detect tactics

#### 1) Concept for applying the model

The trajectory cluster model can be used to determine whether a detected trajectory belongs to one of the clusters based on the criteria that were used to extract the cluster model. Trajectories are assigned to a cluster if they belong to the core points of a cluster or if they are directly reachable from a core point in the sense of DBSCAN. Consequently, a trajectory is considered to belong to a previously identified trajectory cluster if it is directly reachable from any of the core points of that cluster.

To evaluate the criteria above, the similarity-based distances to all the core trajectories that are part of the clustering model need to be computed. If the distance to any of the core trajectories is less than the ε that was used to compute the clustering model, the trajectory belongs to the cluster of that core trajectory. If no core trajectory is within the ε distance, the trajectory that is checked cannot be assigned to any of the clusters and is treated as noise.

With the assignment of trajectories to clusters, team moves can be “recoded”, as described in Section III.A.2) Then, it is determined whether the team move supports any of the previously identified items. In this case, the team move contains the tactic that is represented by the supported frequent itemset.

#### 2) Performance aspects

Given the current set of data used to compute the clustering model, 3,912 core trajectories were identified. The SNN-based distance that is derived from the DTW distances needs to be computed for each trajectory contained in a team move to check the ε limit of direct reachability. A rough estimation using the measured time for calculating the distances and similarities of the trajectories to derive the cluster model is based on the total number of distances that have been calculated so far. In total, approximately 132 million distances and similarities were calculated, which took about 7,610 seconds. For the application case, approximately 27 thousand distances and similarities (3,912 times 7) need to be calculated, which can be estimated with an elapsed time of half a second. The encoding and the search for supported itemsets are in the millisecond range.

Overall, we can assume an upper limit of one second for automatically determining the tactics based on a stream of trajectories, which is fast enough for the given application case because the data of the team moves consist of 5.5 seconds of position data before an attempt happens, i.e., after an attempt, there is a minimum time gap of 5.5 seconds until we might have another set of trajectories of an attempt.

## VI. Conclusion and Future Work

A clustering approach has been proposed to find similarity groups of team moves without the need for the upfront

assignment of class labels. Using a two-step approach based on the concept of shared similarity and the dynamic time warping distance addresses multiple shortcomings of the original approach in finding similarity groups. In particular, the need for manual collection of data in addition to spatiotemporal data is avoided.

The results of the clustering of trajectories (the first step) can be verified by evaluating representatives of the identified clusters. From an application perspective, the representative trajectories should correspond with the intended individual moves of certain player types at the application level. With this application-level mapping, end-users are more likely to establish trust in the approach.

The second step of the search for similarity groups involves searching for relevant frequent itemsets that deviate from the usual approaches that try to solve the task via a clustering approach. Using the search for relevant frequent itemsets avoids the need for an explicit distance criterion, which is difficult to define and difficult to explain in the application context. Furthermore, the concept of relevance is important when combining individual trajectories with team tactical moves. It has been shown that the search for frequent itemsets can be efficiently combined with the concept of relevance of trajectories.

Rather than arbitrarily varying the parameters of the approach, application-level decisions have been made to limit the number of cases that need to be considered when looking for optimal parameters. With this approach, the quality of the trajectory cluster model and the number of "represented" trajectories increase significantly. Furthermore, the assignment ratio of team moves to team move similarity groups increased as well.

The results of the applicability study show that the approach works in a real-world scenario. Previously recognized problems due to the low amount of available data cannot be observed anymore. The long-term objective is to derive a stable model that allows us to assign a label to team moves during matches by using the previously extracted cluster model. Hence, we have a basis for a novel approach to take individual contributions to a team tactic into account when evaluating players' performance in the future.

## ACKNOWLEDGMENT

The work presented in this paper was supported by the German Handball Federation (DHB) and the German Handball-Bundesliga GmbH (HBL). Furthermore, it was particularly supported by the first league handball teams, TVB Stuttgart and Frisch Auf! Göppingen.

## REFERENCES

[1] Catapult, "Vector T7". [retrieved: January 2025]. Available: https://www.catapult.com/blog/vector-t7-white-paper

[2] Kinexon, "Perform LPS". [retrieved: January 2025]. Available: https://kinexon-sports.com/products/perform-lps/

[3] F. Schwenkreis, "An Approach to use Deep Learning to Automatically Recognize Team Tactics in Team Ball Games", in *Proceedings of the 7th Conference on Data Science, Technology and Applications.*, Porto: Scitepress, Jul. 2018, pp. 157–162.

[4] F. Schwenkreis, "Using the Silhouette Coefficient for Representative Search of Team Tactics in Noisy Data", in *Proceedings of the 11th Conference on Data Science, Technology and Applications*, Lisbon, Portugal: Scitepress, Jul. 2022, pp. 193–202.

[5] B. Aronov, S. Har-Peled, C. Knauer, Y. Wang, and C. Wenk, "Fréchet distance for curves, revisited", in *European symposium on algorithms*, Berlin, Heidelberg: Springer, 2006, pp. 52–63.

[6] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis", *Computational and Applied Mathematics*, no. 20, pp. 53–65, 1987.

[7] F. Schwenkreis, "Automated Detection of Trajectory Groups Based on SNN-Clustering and Relevant Frequent Itemsets", presented at the IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA), Thessaloniki, Greece: IEEE, Oct. 2023, pp. 1–10.

[8] B. Kerner, Ed., "The Physics of Traffic", Springer, 2004.

[9] S. Wang, Z. Bao, J. S. Culpepper, T. Sellis, and X. Qin, "Fast large-scale trajectory clustering", *Proc. VLDB Endow.*, vol. 13, no. 1, pp. 29–42, Sep. 2019.

[10] Y. Liu et al., "ECMA: An Efficient Convoy Mining Algorithm for Moving Objects", in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, Virtual Event Queensland Australia: ACM, Oct. 2021, pp. 1089–1098.

[11] G. Schrijvers, A. van Hoorn, and N. Huiskes, "The care pathway: concepts and theories: an introduction", *International journal of integrated care*, 2012. [retrieved: January 2025] Available https://doi.org/10.5334/ijic.812.

[12] V. Vogt, S. M. Scholz, and L. Sundmacher, "Applying sequence clustering techniques to explore practice-based ambulatory care pathways in insurance claims data", *European Journal of Public Health*, vol. 28, no. 2, pp. 214–219, Apr. 2018.

[13] U. Brefeld, J. Davis, J. V. Haaren, and A. Zimmermann, Eds., *Machine Learning and Data Mining for Sports Analytics*. Ghent, Belgium: Springer, 2020.

[14] U. Brefeld, J. Davis, J. Van Haaren, and A. Zimmermann, *Machine Learning and Data Mining for Sports Analytics*, Springer, 2021.

[15] U. Brefeld, J. Davis, J. Van Haaren, and A. Zimmermann, *Machine Learning and Data Mining for Sports Analytics*, Springer, 2022.

[16] P. Bauer and G. Anzer, "Data-driven detection of counterpressing in professional football: A supervised machine learning task based on synchronized positional and event data with expert-based feature extraction", *Data Min Knowl Disc*, vol. 35, no. 5, pp. 2009–2049, Sep. 2021.

[17] Y. Jiang and C. Bao, "Human-centered artificial intelligence-based ice hockey sports classification system with web 4.0", *Journal of Intelligent Systems*, vol. 31, pp. 1211–1228, 2022.

[18] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, "Time-series clustering – A decade review", *Information Systems*, vol. 53, pp. 16–38, Oct. 2015.

[19] S. Aghabozorgi, T. Y. Wah, T. Herawan, and H. A. Jalab, "A Hybrid Algorithm for Clustering of Time Series Data Based on Affinity Search Technique", *The Scientific World Journal*, no. 3, 2014. [retrieved: January 2025] Available: https://doi.org/10.1155/2014/562194

[20] U. von Luxburg, "A tutorial on spectral clustering", *Statistics and Computing*, vol. 14, no. 4, pp. 395–416, 2007.

[21] A. Bojchevski, Y. Matkovic, and S. Günnemann, "Robust Spectral Clustering for Noisy Data: Modeling Sparse Corruptions Improves Latent Embeddings", in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax NS Canada: ACM, Aug. 2017, pp. 737–746.

[22] M. Roux, "A Comparative Study of Divisive and Agglomerative Hierarchical Clustering Algorithms", *Journal of Classification*, no. 35, pp. 345–366, 2018, [retrieved: January 2025]. Available: https://doi.org/10.1007/s00357-018-9259-9.

[23] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, Portland Oregon: AAAI Press, 1996, pp. 226–231.

[24] R. W. Hamming, “Error detecting and error correcting codes”, *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[25] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules”, in *Proceedings of 20th International Conference on Very Large Databases*, Santiago de Chile, Chile: Morgan Kaufmann, 1994, pp. 487–499.

[26] X. Zhao, S. C. Xinhui Zhang Pan Wang, and Z. Sun, “A weighted frequent itemset mining algorithm for intelligent decision in smart systems”, *IEEE Access*, vol. 6, pp. 29271–29282, 2018.

[27] Y. Tao *et al.*, “A comparative analysis of trajectory similarity measures”, *GIScience & Remote Sensing*, vol. 58, no. 5, pp. 643–669, Jul. 2021.

[28] R. Bellman and R. Kalaba, “On adaptive control processes”, *IRE Transactions on Automatic Control*, vol. 4, no. 2, pp. 1–9, 1959.

[29] M. Brankovic et al., “(k, l)-Medians Clustering of Trajectories Using Continuous Dynamic Time Warping”, in *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, Nov. 2020, pp. 99–110.

[30] J. Paparrizos, C. Liu, A. J. Elmore, and M. J. Franklin, “Debunking Four Long-Standing Misconceptions of Time-Series Distance Measures”, in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, Portland OR USA: ACM, Jun. 2020, pp. 1887–1905.

[31] L. Ertöz, M. Steinbach, and V. Kumar, “Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data”, in *Proceedings of the 2003 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, May 2003, pp. 47–58.

[32] P. Jaccard, “The Distribution of the Flora in the Alpine Zone”, *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[33] Bernard Desgraupes, “Clustering Indices.” [retrieved: January 2025] Available: https://de.scribd.com/document/268911122/Cluster-Criteria

[34] J. C. Dunn, “Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters”, *Journal on Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[35] J. C. Bezdek and N. R. Pal, “Some new indexes of cluster validity”, *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, no. 3, pp. 301–315, Jun. 1998.