K-Means for Semantically Enriched Trajectories

Jana Seep
jana.seep@uni-muenster.de
Westfälische Wilhelms-Universität Münster
Department of Computer Science
Münster, Germany

Jan Vahrenhold
jan.vahrenhold@uni-muenster.de
Westfälische Wilhelms-Universität Münster
Department of Computer Science
Münster, Germany

ABSTRACT

Clustering a set of given objects is a standard component of many data analysis tasks. The well-known *k-means* algorithm is a centroid-based clustering algorithm that optimizes the sum of distances between data objects and their assigned cluster centers. Each centroid then represents all objects assigned to a given cluster.

In this paper, we study the special case of clustering semantically enriched spatio-temporal trajectories, i. e., trajectories where each trace point can be annotated with arbitrary, possibly categorical semantic data in addition to numerical spatio-temporal data. Such trajectories result from, e. g., tracking animals, humans, or weather phenomena and capture semantic contexts analysts may want to be aware of when interpreting the resulting clusters.

Most current clustering algorithms for spatio-temporal categories take into account the numerical spatio-temporal coordinates only; thus, the resulting clusters do not necessarily reflect the characteristics of the additional semantic data. Building upon our earlier work on computing a representative trajectory for a given set of semantically enriched spatio-temporal trajectories, we describe how to implement the *k-means* algorithm to work with such data. In particular, we define a similarity measure called EFSMSim between a trajectory and a graph-based representation of a cluster centroid and show how to use this in the context of the *k-means* algorithm.

We evaluate our EFSMClust approach by comparing it with state-of-the-art clustering algorithms taking into account either spatio-temporal information only or semantic attributes as well. Our experiments show that our algorithm is competitive even with respect to purely geometric performance measure and at the same time returns a representation of the centroids that can be used by domain experts to interpret both spatio-temporal and semantic information as well as to explore their possible relationships.

CCS CONCEPTS

• **Information systems → Clustering**; **Geographic information systems**.

KEYWORDS

semantic clustering, semantic trajectories, k-means

ACM Reference Format:
Jana Seep and Jan Vahrenhold. 2021. K-Means for Semantically Enriched Trajectories. In *1st ACM SIGSPATIAL International Workshop on Animal Movement Ecology and Human Mobility (HANIMOB'21), November 2, 2021, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3486637.3489495

1 INTRODUCTION

Data obtained in the context of capturing and analysing human and animal mobility is usually stored and processed as so-called *spatio-temporal trajectories* [20]. Such trajectories represent the locations of an object at discrete time steps [50] and represented as a time-stamped sequence of *trace points* in the underlying geometric space. The movement of the object between two temporally consecutive trace points then is assumed to follow a linear interpolation.

Recent advances in data capture systems and analysis algorithms have resulted in datasets where the trace points are annotated with additional information, e. g., the mode of transportation, acceleration, or type of underlying route network. As a result, new research questions related to the modeling and analysis of so-called spatio-textual trajectories [23] or semantic trajectories [22] emerged, e. g., in the context of analyzing the behavior of animals [26] or map matching, where the goal is to define a used road by only regarding vehicles' trajectories [32]. As Güting et al. point out, "symbolic trajectories [i. e., time-dependent labels] can be combined with geometric trajectories to obtain annotated [spatio-temporal] trajectories" ([22], p. 1). There is no limitation regarding the type of annotations that can be used in addition to geometric information: annotated spatio-temporal trajectories can be used in a range of application domains [20] including, but not limited to, animal tracking [36], market analysis [24], sports analysis [19], analysis of urban mobility patterns [52], or historical weather data analysis [21].

A common and well-researched first analysis step is to cluster the data; the resulting clusters then can be used to enable further pattern mining steps. Due to the inherent difficulties of matching trajectories to data points in feature space – which would enable well-known data mining methods – clustering of trajectories is usually done based on the underlying spatial, spatio-temporal, or semantic information; see the reviews by Yuan [48] and Zheng [50] and the references therein. In the case of clustering spatio-temporal data, recent approaches have successfully use both the (discrete) Fréchet distance and (continuous) dynamic time warping [8, 10].

The focus of this paper is on *semantic trajectory clustering*. As documented in a recent survey [48], the usual approach is to attempt to derive semantic information, e. g., speed, acceleration, or close points-of-interest, from the given spatio-temporal raw data. While promising results have been obtained using such derived

HANIMOB'21, November 2, 2021, Beijing, China

ACM ISBN 978-1-4503-9122-1/21/11.
https://doi.org/10.1145/3486637.3489495

information, Yuan et al. note the importance and the unused potential of integrating "enriched data, such as geographic information, moving object features of themselves, the environment and state of moving objects when the locations are recorded" [48, p. 136] and conclude that "in the future research, much attention should be paid on the external semantic information" [48, p. 136].

2 RELATED WORK

Semantically Enriched Trajectories. Yuan et al. [48] note that there is a variety of applications where a sole focus on spatio-temporal properties does not unlock the full analysis potential. Instead, one should also contextualize the spatio-temporal data by taking into account semantic annotations and background information [34].

One line of research aims at deriving semantic information from raw spatio-temporal trajectory data [45, 47] and start-/stop-points of the objects [2, 3, 33, 40, 51]. Such *context-aware movement analysis* has been used to derive, e. g., travel modes or static behavior [13, 14, 38]. For more, see the recent survey by Brum-Bastos et al. who state that research in this field should focus on "approaches to extract meaningful information from contextualized data" [9, p. 1]. Other studies focus on *time dependent clustering*: interpreting time stamps not (only) as a spatio-temporal dimension but as a semantic attribute; see the survey by Yuan et al. [48, ch. 4.2].

In contrast to the above approaches, in which "semantic" annotations are derived from – and thus dependent on – spatio-temporal attributes, we focus on semantically enriched (or: *multi-aspect* [35]) trajectories: Each trace point of such a trajectory can be annotated with arbitrary semantic annotations that are independent of its spatio-temporal attributes. From an analysis point-of-view, this seemingly technical distinction is important: If the semantic attributes are independent from, i. e., not derived from spatio-temporal attributes, the analysis can hope to detect correlations or even causal relationships between semantic and spatio-temporal attributes that stem from the underlying object's behavior instead of having been artificially introduced in the derivation process.

A recent example of analyzing semantically enriched trajectories according to this definition is the work of Liu and Guo [27]. The authors mine trajectory patterns using community detection for a network of semantically enriched spatio-temporal trajectories.

Measuring Similarities. Irrespective of whether they use semantic annotations, the algorithms discussed above are similarity-based: They rely on a similarity measure to compare trajectories with each other or, in the case of centroid-based clustering, with representative "centroids" of cluster candidates. Thus, as noted by Zhang et al., "a key issue [for similarity-based clustering of trajectories] is how to measure the similarity between two trajectories" [49, p. 1051].

For *spatial clustering* algorithms, i. e., algorithms working with spatio-temporal information only, researchers have studied intensively (dis)similarity measures such as dynamic time warping [5], discrete and continuous Fréchet distance [1, 15], longest common subsequences [43], or edit distances on real-valued sequences [12]; see a recent survey by Tao et al. [39] for more details about these measures and the algorithms in which they are employed.

Due to our focus on semantically enriched spatio-temporal trajectories, we need a similarity measure (or rather, as we will see in Section 3.2, a distance function) that takes into account "similarities" between values of semantic attributes; these attributes may be numerical but can also be ordinal or even categorical such that traditional approaches using, e. g., averaging may not be possible.

Among the few algorithms taking into account semantic attributes, the algorithm by Lehmann et al. [25] considers "stop" and "move" annotation, i. e., a single binary attribute. As pointed out by Petry et al. [35], trajectories may be enriched by a number of semantic attributes whose implications for the underlying object's behavior may be considerably more complex than the question of whether or not it is moving. Liu and Schneider [28] propose a similarity measure that combines a single spatial dimension (computed as an aggregate of geometric properties such as length and orientation of subtrajectories) and a single semantic dimension (computed based on longest common subsequences of annotation). Since this algorithm on one hand aggregates spatial data and on the other hand requires exact matches among the annotations, it is not well suited to address time-variant behavior or to capture slight differences of possibly multiple semantic attributes.

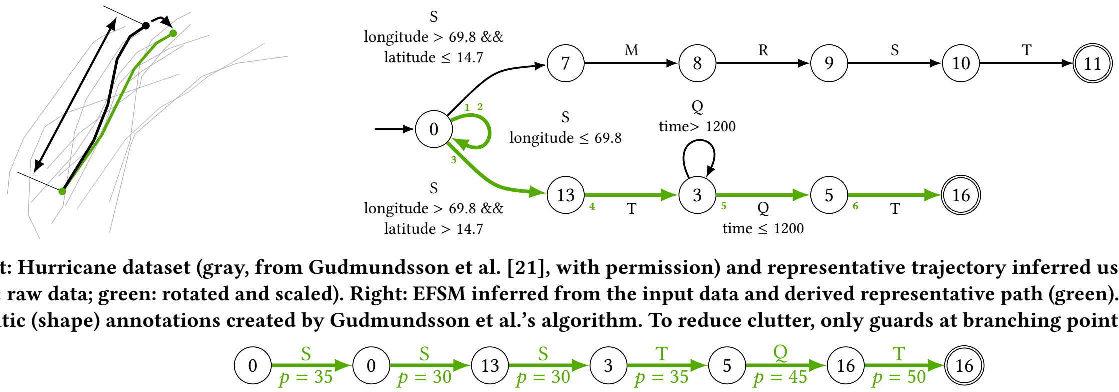
The algorithm of Xiao et al. [46] is hand-tuned to work with a fixed set of semantic attributes, e. g., a location's name or the number of visits, and thus is restricted to few use cases; moreover, this approach does not take into account a temporal dimension.

The similarity measure most relevant to our paper is the MUITAS similarity defined by Petry et al. [35] to compute the similarity between two multi-aspect trajectories. This measure can be parameterized to take into account not only independent semantic attributes but also possible relations between them. We will come back to MUITAS as part of our empirical evaluation. Their study also shows that MUITAS is superior to another similarity measure previously proposed by Furtado et al. [16]; we thus do not discuss Furtado et al.'s measure or any of its predecessors in more detail.

In conclusion, there is a need for more research on clustering semantically enriched spatio-temporal trajectories that takes into account external semantic information as a first-order contributor to a similarity measure. We contribute to narrowing this research gap by showing how to implement the well-known *k-means* algorithm [29] for semantically enriched spatio-temporal trajectories. For this, we use an approach inspired from reverse software engineering which allows us to consider both spatio-temporal and semantic attributes independent of each other. The resulting centroids are represented both geometrically and as an extended finite state machine in which states and transitions are annotated with characteristic attributes and guards. This representation allows domain experts to sanity-check the clustering results from both a geometric and a semantic perspective and can serve as a basis for forming and stating hypotheses with respect to possible interactions between spatio-temporal and semantic attributes.

3 EFSMCLUST ALGORITHM OVERVIEW

Our EFSMClust approach uses the classic *k-means* algorithm by Lloyd [29]. In the basic version of this centroid-based clustering algorithm, the input consists of a set of points and a parameter k. The algorithm then first selects an initial set of k candidate centroids and proceeds in multiple rounds until a termination criterion is fulfilled. In each round, the algorithm iterates over all points and



(a) Left: Hurricane dataset (gray, from Gudmundsson et al. [21], with permission) and representative trajectory inferred using our approach (black: raw data; green: rotated and scaled). Right: EFSM inferred from the input data and derived representative path (green). Letters represent semantic (shape) annotations created by Gudmundsson et al.'s algorithm. To reduce clutter, only guards at branching points are shown.

(b) Unrolled representative path; transitions are annotated with shape labels (above) and air pressure (below).

Figure 1: Input data, inferred EFSM, and derived representative path for the hurricane dataset.

assigns each point to the nearest centroid. After each point has been assigned to a centroid, all centroids are updated to an aggregate, usually the mean, of all points assigned to them. Upon termination, the centroids along with the points assigned to them are returned.

There has been a wealth of research on *k-means* and its analysis; we refer the reader to the survey of Blömer et al. [7] and the references therein. Also, there is a line of research on a variant called *k-means++* [4] whose sampling of the initial centroids has been found to be highly effective in practice and to also provide provable approximation results; see [6] for more details.

For this paper, we focus on Lloyd's basic variant as it simplifies the exposition and since the sampling conditions of *k-means++* can be layered on top of our algorithm. Given this setup, we identify two components that need to be updated when applying the basic algorithm to a set of semantically annotated trajectories: The algorithm needs to employ a function to compute, for each input trajectory, the distance to its nearest centroid, and the algorithm needs to be able to update a centroid to an aggregate of all input trajectories assigned to it. We address both using a machinery we have introduced previously: Inferring so-called *extended finite state machines* from semantically enriched spatio-temporal trajectories [37].

3.1 Inferring Extended Finite State Machines

Our approach to model semantically enriched spatio-temporal trajectories builds on the following assumption [37]: The behavior of the objects whose traces we wish to model is determined by an underlying function such that, when moving along the time dimension, the next trace element depends on the spatial and non-spatial attributes of the current trace elements. The goal then is to recover the program computing this function given a set of traces.

Based on this assumption, we interpret a sequence of trace elements as a sequence of program states described by $(key, value)$-pairs of all variables. We then use the algorithm of Walkinshaw et al. for inferring an extended finite state machine from a set of traces representing this assumed function [44]. The outcome of this algorithm is an extended finite state machine (EFSM), i. e., a finite state machine in which the transitions are annotated with guards indicating whether or not they may be taken given a specific program state. By construction, this EFSM models the common behavior of the set of traces for which it was inferred. Figure 1a shows an example for hurricane traces each of which models time and air pressure attributes in addition to the spatio-temporal information.

In our previous work, we used the EFSM approach to construct a representative trajectory for a set of traces that were presented to the algorithm with the underlying assumption that they represent similar behavior [37]. Our algorithm scans the resulting EFSM to compute a so-called *representative path* (shown in green in Figure 1): a sequence of states and transitions in the EFSM that captures the behavior of a majority of the traces for which the EFSM was inferred; see [37]. This representative path then is used to obtain a spatio-temporal representation of a representative trajectory.

We use our EFSM-based algorithm in the context of implementing *k-means* for semantically enriched spatio-temporal trajectories to update the centroids. At the end of each iteration, we infer an EFSM for all trajectories assigned to the same cluster and use the resulting representative trajectory as this cluster's new centroid.

Next we discuss how to leverage the EFSM inferred for a tentative cluster to implement another building block of *k-means*: finding the closest centroid for a given trajectory. For this, we need a representation for each centroid representing a cluster.

As mentioned above, the *k-means* algorithm proceeds in rounds and, in each round, assigns each object to be clustered to its closest centroid. While there is a variety of purely geometry-based distance metrics for spatio-temporal trajectories, none of these takes into account non-spatial, possibly categorical, attributes.[1]

In contrast, our EFSM construction computes for each tentative cluster a representative path, i. e., an unrolled sequence of states and, possibly guarded, transitions between them. Each state and transition is labeled with $(key, value)$-pairs of all variables that characterize the representative path; see Figure 1b for the representative path inferred from the EFSM shown in Figure 1a (right).

It remains to describe how to find the nearest centroid for a given semantically enriched spatio-temporal trajectory. For this, we follow standard procedure and first describe a similarity measure which we then use to compute distances.

[1] The MUITAS approach [35] is a notable exception but cannot be used to compute a representative trajectory needed to update the centroid at the end of each iteration.

3.2 An Asymmetric Similarity Measure

In principle, we could interpret the trajectory as its own representative path and then compute some distance between two representative paths. In this case, however, we would need to know which attributes might introduce transition guards and, in the case of numerical attributes, which threshold values to use. While we discuss below that we can deal with this by considering the means and standard deviations of attributes, a closer look at the *k-means* algorithm reveals that we only need to compute the distance from a trajectory to a centroid and not vice versa. In particular, the algorithm does not require the similarity measure from which we seek to obtain a distance to be symmetric. This allows us to define a similarity measure that captures how well an EFSM (representing a cluster) is suited to "generate" a given semantically enriched spatio-temporal trajectory. For this, we compare the semantically enriched spatio-temporal trajectory against each cluster's representative path.

Given a representative path and a semantically enriched trajectory, we wish to know how well this representative path represents the trajectory. Put differently, we wish to assess whether the trajectory could be generated from the underlying EFSM by traversing a path without skipping "too many" transitions. For this, we use the well-known sequence alignment algorithm by Needleman and Wunsch [31] for computing the cost of an optimal alignment of two character sequences $C = c_1 \dots c_m$ and $\Gamma = \gamma_1 \dots \gamma_n$. In a nutshell, their algorithm is a dynamic programming algorithm that optimizes the cost by comparing, for each pair (c_i, γ_j) of characters considered, whether the cost $\alpha(c_i, \gamma_j)$ of aligning these characters is preferable to the cost δ of "skipping" one of these characters.

3.2.1 *Initializing a Cost Matrix.*

The algorithm of Needleman and Wunsch [31] is parameterized over the cost function $\alpha(\cdot,\cdot)$ and the skipping cost δ. While $\alpha(\cdot,\cdot)$ usually is defined as a symmetric function, the algorithm itself does not rely on this property.[2] We thus can safely define $\alpha(\sigma, \tau) \in [0 \dots 1]$ as the cost of aligning a pair (σ, τ) where σ is a segment of a given semantically enriched spatio-temporal trajectory $\mathcal{T} = (\sigma_1, \dots, \sigma_m)$ and τ is a transition taken from a given representative path $\mathcal{P} = (\tau_1, \dots, \tau_n)$. These cost values are stored in a *cost matrix* $A_{\mathcal{T},\mathcal{P}} = (\alpha(\sigma_i, \tau_j))_{i,j} \in [0 \dots 1]^{m \times n}$ so that they can be accessed readily.

Keeping in mind that we want to use the algorithm of Needleman and Wunsch to compute a *similarity* measure, we define the cost matrix such that $\alpha(\sigma, \tau)$ is high if the segment σ is similar to the transition τ in the sense that τ is suited to generate the segment σ.

Assume that we are given a semantically enriched spatio-temporal trajectory $\mathcal{T} = (\sigma_1, \dots, \sigma_m)$ and a representative path $\mathcal{P} = (\tau_1, \dots, \tau_n)$ derived from an EFSM $\mathcal{E}$. To compute $\alpha(\sigma_i, \tau_j)$, we first inspect the transition τ_i. Recall that $\mathcal{E}$ was inferred from a non-empty set S of semantically enriched spatial trajectories. By construction of $\mathcal{E}$, each transition in $\mathcal{E}$, and thus in the derived representative path $\mathcal{P}$ as well, is induced by a non-empty set of segments of the trajectories in S; the algorithm ensures that each transition is annotated with these segments. This implies that τ_j is annotated with a non-empty set of segments; let r_j be the number of these segments.

The key insight is that in the context of our implementation of the *k-means* algorithm the set S of semantically enriched spatial trajectories inducing $\mathcal{E}$ and $\mathcal{P}$ is a subset of the same dataset from which $\mathcal{T}$ is taken. Thus, all segments τ_j are annotated with the same set $\mathcal{A} = \{a_1, \dots, a_r\}$ of (spatio-temporal and non-spatio-temporal) attributes as the segments in $\mathcal{T}$. We thus can simply check, for each attribute $a_\ell \in \mathcal{A}$ if there is at least one of the r_j segments, τ_j is annotated with, that stores a value for a_ℓ that is similar to the value of a_ℓ σ_i stores; let $0 \le \rho_j \le r$ be the number of such attributes.

For ordinal or categorical attributes, we define similarity to occur iff there is an exact match. For a real-valued attribute a_ℓ, we compute the mean M_ℓ and standard deviation SD_ℓ of all pairwise distances of the values a_ℓ assumed in the input, and consider two a_ℓ-values to be similar iff their absolute difference is less than $M_\ell - SD_\ell$.

The similarity $\alpha(\sigma_i, \tau_j)$ of σ_i and τ_j then is defined as $\rho_j / r_j \in [0 \dots 1]$, i.e., σ_i and τ_j are considered to be similar if there are "many" attributes in $\mathcal{A}$ for which there is at least one of the segments τ_j is annotated with that has a similar attribute value. We repeat this for all (*segment,transition*)-pairs to obtain the full matrix $A_{\mathcal{P},\mathcal{T}}$.

Since clusters and, thus, the representative paths inferred can change after each iteration of the *k-means* algorithm, we need to compute $A_{\mathcal{T},\mathcal{P}}$ for all pairs $(\mathcal{P}, \mathcal{T})$ at the start of each iteration.[3]

3.2.2 *Computing an Alignment.*

Having the cost matrix $A_{\mathcal{T},\mathcal{P}}$ at our disposal, we could simply run the algorithm by Needleman and Wunsch [31] to compute the score for an optimal alignment by maximizing (recall that we want to compute a similarity score) for $OPT(\sigma_m, \tau_n)$ using the following recursive definition:

$$\begin{aligned} OPT(\sigma_i, \tau_j) = \max\{&\alpha(\sigma_i, \tau_j) + OPT(\sigma_{i-1}, \tau_{j-1}), \\ &\delta + OPT(\sigma_{i-1}, \tau_j), \delta + OPT(\sigma_i, \tau_{j-1})\}. \end{aligned}$$

In contrast to the default use cases of the Needleman and Wunsch algorithm, we have the situation that one single object (the trajectory $\mathcal{T}$) has to be aligned with k different representative paths $\mathcal{P}$, one for each tentative cluster, individually; then, the path for which the best score was obtained, is chosen. The k paths, however, can have quite different numbers of transitions each. Thus, the penalty for skipping one transition in the alignment process can be very different depending on the number of transitions, potentially resulting in skewed similarity scores when using the above formula.

To ensure that the k similarity scores are comparable when selecting their maximum, we adjust the cost for skipping a segment or a transition according to its influence, i. e., relative to the length of $\mathcal{T} = (\sigma_1, \dots, \sigma_m)$ and $\mathcal{P} = (\tau_1, \dots, \tau_n)$. We dynamically scale each cost value by the size of the alignment gap introduced by skipping one or more transitions in $\mathcal{P}$ relative to n or by the size of the alignment gap introduced by skipping one or more segments in $\mathcal{T}$ relative to m. More precisely, assume that the last alignment was made between $\sigma_{i'}$ and $\tau_{j'}$ and that we are currently in the process of assessing the cost of aligning σ_i and τ_j. We define

$$\ell(\mathcal{T}, i, i', \mathcal{P}, j, j') := \min\left\{\frac{(i'-i)-1}{m}, \frac{(j'-j)-1}{n}\right\}$$

to be the minimum (normalized) gap size that would be induced by aligning $\sigma_{i'}$ and $\tau_{j'}$ as well as σ_i and τ_j. If we plug in $i = i'$ or $j = j'$, the respective term evaluates to zero which models the "cost" of not skipping a segment or a transition. Consequently, the

[2] In fact, the running example used in Needleman and Wunsch's paper uses an asymmetric cost function [31, p. 445].

[3] While this is computationally expensive, it is also embarrassingly parallel and thus can be sped up using straightforward parallelization techniques.

"penalty" term δ, which would be zero as well, is captured in the above definition as well. This results in the following adjusted recursive definition (again, with appropriate recursion base cases):

$$OPT(\sigma_i, \tau_j) = \max\{\ell(\mathcal{T}, i, i', \mathcal{P}, j, j') \cdot \alpha(\sigma_i, \tau_j) + OPT(\sigma_{i-1}, \tau_{j-1}), \\ OPT(\sigma_{i-1}, \tau_j), OPT(\sigma_i, \tau_{j-1})\}.$$

Computing the above function, i. e., running the algorithm of Needleman and Wunsch for our cost function, we obtain the similarity score for an alignment of $\mathcal{T}$ and $\mathcal{P}$ that is optimal given our cost function. To make these scores comparable across different paths $\mathcal{P}$ given a fixed trajectory $\mathcal{T}$, we normalize this score by the length of $\mathcal{P}$ and obtain EFSMSim($\mathcal{T}, \mathcal{P}$) which can be quickly verified to fall into the interval $[0 \dots 1]$.

We now define $d(\mathcal{T}, \mathcal{P}) = 1 - \text{EFSMSim}(\mathcal{T}, \mathcal{P})$ and use this function as the distance function used in Lloyd's algorithm. Since Lloyd's algorithm only computes the minimum of a set of distances, it does not require this distance function to be a metric; in particular, it is not required that this function realizes the triangle inequality.

4 ALGORITHMIC BUILDING BLOCKS

The *k-means* algorithm [29] our EFSMClust approach relies on can be stated generically as a function partitioning a set O of objects into k clusters using four building blocks; see Algorithm 1. At the beginning of the algorithm, the function initializeCentroids(O, k) computes a set $\mathcal{Z}$ of k centroids, one for each cluster. The algorithm then iterates until a termination criterion (terminate($\mathcal{Z}, C, k, O$)) is fulfilled. In each iteration, the algorithm assigns each object to its nearest centroid (assignToClusters($\mathcal{Z}, O$)). It then updates the centroids using updateCentroids(C).

Algorithm 1 Generic version of the *k-means* algorithm [29].

```
function clusterObjects(int $k$, objects $O$)
    $\mathcal{Z}[1 \dots k] \leftarrow$ initializeCentroids($O, k$);        ▷ Centroids.
    $C\ [1 \dots k] \leftarrow \emptyset^k$;        ▷ Objects assigned to each cluster.
    while not terminate($\mathcal{Z}, C, k, O$) do
        $C \leftarrow$ assignToClusters($\mathcal{Z}, O$);
        $\mathcal{Z} \leftarrow$ updateCentroids($C$);
    return ($C, \mathcal{Z}$);
```

We now discuss how to implement each of the building blocks in the context of partitioning a set O of semantically enriched spatio-temporal trajectories into k clusters.

initializeCentroids(O, k). It is well known that "the algorithm's sensitivity to the initial selection of the cluster centers [centroids] remains to be its most serious drawback" [11, p. 79]. The original *k-means* algorithm randomly samples all k initial centroids from the input set O. In contrast, the current state-of-the-art variant of *k-means*, the *k-means++* algorithm [4] samples only the first centroid and then adaptively samples the remaining centroids taking into account each point's cost in the current assignment [4, 6, 7].

For the evaluation of our approach, we need to compare the EFSMSim similarity with the MUITAS similarity. To ensure that we start with the same set of centroids, we implement a deterministic initialization procedure.[4] The *maximin*-approach by Gonzalez [18], which can be seen as a deterministic predecessor to *k-means++*, starts from a distinct seed centroid. This seed is computed as the centroid of a virtual cluster containing all input data. The algorithm then selects the remaining $(k-1)$ initial centroids by iteratively choosing an object whose distance to its neareast centroid is maximal. This way, the initialization is agnostic of the nature of the distance function used. At the same time, it is compatible with the way distances are computed during the iterations of the `while`-loop. Thus, it can be used in an experimental comparison of our EFSMSim similarity and the MUITAS similarity without advantaging either method. To implement the MUITAS approach we used the same notion of attribute similarity as described for EFSMSim: we implement exact match comparisons for ordinal and categorical attributes and used $M_\ell - SD_\ell$ as MUITAS' internal threshold (see [35] for more details) for numerical attributes; all attributes were weighted uniformly.

Before entering the first iteration of Algorithm 1, we infer a (trivial) EFSM from each of k clusters consisting of the k trajectories just selected and store the k derived representative paths in $\mathcal{Z}$.

terminate($\mathcal{Z}, C, O, k$). As *k-means* is known to require exponentially many iterations until convergence is reached, even for the basic case of two-dimensional input points [42], implementations of *k-means* usually resort to a heuristic for deciding when to terminate the `while`-loop. In our case, we stop once the average radius of a cluster (maximum distance between the cluster's centroid and one of its assigned trajectories) does not decrease from one iteration to the next. Again, this procedure only relies on the existence of a distance function and thus can be used in a comparative study.

assignToClusters($\mathcal{Z}, O$). This function iterates over each input trajectory and assigns it to the cluster corresponding to its nearest centroid. As discussed in Section 3.2, this requires to check each trajectory $\mathcal{T}$ against each representative path $\mathcal{P}$; for each such check, we need to compute the cost matrix $A_{\mathcal{T},\mathcal{P}}$ based upon which we can compare the similarity score EFSMSim($\mathcal{T}, \mathcal{P}$). Instead of minimizing the distance $d(\mathcal{T}, \mathcal{P}) = 1 - \text{EFSMSim}(\mathcal{T}, \mathcal{P})$, the algorithm shown below (Algorithm 2) simply maximizes EFSMSim($\mathcal{T}, \mathcal{P}$).

Algorithm 2 Assign each trajectory to its nearest centroid.

```
function assignToClusters(centroids $\mathcal{Z}$, trajectories $O$)
    $C[1 \dots k] \leftarrow \emptyset^k$;        ▷ Objects assigned to each cluster.
    for $\mathcal{T} \in O$ do        ▷ Iterate over all trajectories.
        $maxScore = -1$;
        for $\mathcal{P} \in \mathcal{Z}$ do        ▷ Iterate over all representative paths.
            $A_{\mathcal{T},\mathcal{P}}$ = computeMatrix($\mathcal{T}, \mathcal{P}$);        ▷ Section 3.2.1.
            EFSMSim($\mathcal{T}, \mathcal{P}$) = computeSimilarity($\mathcal{T}, \mathcal{P}, A_{\mathcal{T},\mathcal{P}}$);
                                                        ▷ Section 3.2.2.
            if EFSMSim($\mathcal{T}, \mathcal{P}$) > $maxScore$ then
                $maxScore$ = EFSMSim($\mathcal{T}, \mathcal{P}$); $clusterID = \mathcal{P}.id$;
        $C[clusterID] = C[clusterID] \cup \mathcal{T}$;
    return $C$;
```

[4] This choice is for the sake of a fair comparison only. We can compute the cost of any input trajectory by means of the distance to its nearest centroid, and thus implement the probabilistic *k-means++* initialization procedure in our setting as well.

updateCentroids(C). The final building block ensures that each centroid is updated to reflect the objects that have been assigned to it during the current iteration. In the basic *k-means* algorithm for point data, this is done by computing the mean point of all points in the cluster. In our case, we proceed as follows: For each cluster, we infer an EFSM from all trajectories currently assigned to it (see Section 3.1). We then derive the representative path of each EFSM and use it as the corresponding cluster's new centroid. Since this derivation of a new centroid is unique to our EFSMClust approach and – to the best of our knowledge – the only approach known for deriving a representative of a set of semantically enriched spatio-temporal trajectories (see [37] for more details), we used this step also for the MUITAS-based benchmark implementation.

5 EVALUATION

To assess the clustering quality of our EFSMClust approach, we implemented the building blocks based on the software by Walkinshaw et al. [44] and Seep and Vahrenhold [37].

Algorithms. We used two different state-of-the-art benchmarks to compare against: To assess how well our similarity measure EFSMSim captures the underlying semantic information, we also implemented EFSMClust using the MUITAS similarity measure of Petry et al. [35]. While we argued above that external semantic information should be treated as a first-order component of measuring similarity, we acknowledge that purely spatio-temporal attributes are important parts of (visual) data analytics. Thus, we also assessed the purely spatio-temporal clustering quality by comparing EFSMClust (using either EFSMSim and MUITAS) against the recent (k, l)-clustering algorithm of Buchin et al. [10] which also resembles the *k-means* approach. For both MUITAS and (k, l)-clustering, we used their authors' implementations [10, 35].

Benchmark Criteria. The MUITAS similarity was designed for and evaluated in an information retrieval context [35]. For this, Petry et al. followed González et al. [17] in that trajectories of humans are more similar if "generated" by the same person and defined the ground truth to be based on whether or not the trajectory under investigation was generated by the same individual. This favors spatio-temporal coherence strictly over the similarity of semantic attributes, an imbalance we did not want to introduce.

We thus decided to follow Buchin et al. [10] and to quantitatively assess the resulting clusters using their maximum or minimum diameter and average radius. When using a distance function, such as the Fréchet distance, a smaller maximum diameter and average radius is better, when using a similarity measure, such as MUITAS or EFSMSim, a larger minimum diameter and average radius is better. By plugging in the distance function or similarity measure for which each of the three competitors was designed, we obtain three values for each of the two measurements:

FD The discrete Fréchet distance [15] used in the design and evaluation of the (k, l)-clustering algorithm [10].

MUITAS The MUITAS similarity used in [35].

EFSMSim The EFSMSim similarity introduced in Section 3.2.

Since the (k, l)-clustering algorithm is not designed to deal with semantically enriched data, we will use purely spatio-temporal data for comparing all three approaches; as described in our previous work [37], we derived semantic labels from this data such that the resulting data can be fed into our EFSM-based approach as well.

When dealing with semantically enriched data, i. e., when comparing the EFSMSim-based and the MUITAS-based variant of EFSMClust, we will also qualitatively assess the resulting clusters by studying the semantic annotations.

We remind the reader that a core feature of centroid-based clustering in general and, thus, of the *k-means* algorithm in particular lies in updating the centroids to represent the current clusters. While we can change the distance function used for assigning trajectories to clusters as part of EFSMClust (and do so when considering both MUITAS and EFSMSim), updating the centroids is an independent task. For this, we will use the *Fréchet Centering* approach in the case of (k, l)-clustering [10] or our EFSM-based approach [37] discussed in Section 3.1 for EFSMClust.

5.1 Spatio-Temporal Trajectories

We start by discussing the performance on two basic benchmark data sets. Both datasets consist of two-dimensional piecewise linear curves, i. e., purely spatial data. These datasets resemble the most basic type of input that can be given to a clustering algorithm for (possibly semantically enriched) spatio-temporal trajectories.

As noted above, our algorithm for inferring an EFSM for semantically enriched spatio-temporal data can process such basic data by first running an annotation step in which semantic annotations describing the local shape of the curve are derived; see [37] for more details. We use this annotated data as the input for both instantiations of EFSMClust, i. e., for the version using the MUITAS similarity and the version using the EFSMSim similarity.

"Lake Walks" Dataset. The first, tiny dataset is the so-called "lake walks" dataset [41] which consists of seven trajectories, presumably representing the paths taken by walking "above" and "below" a lake (see Figure 2, from [41], with permission). This dataset consists of two clusters and can be used as a benchmark for algorithms computing representative paths that are realistic in the sense that a simple averaging would lead to a path right through the water.

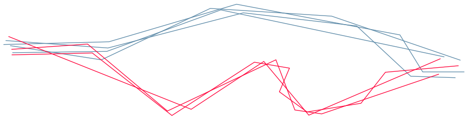
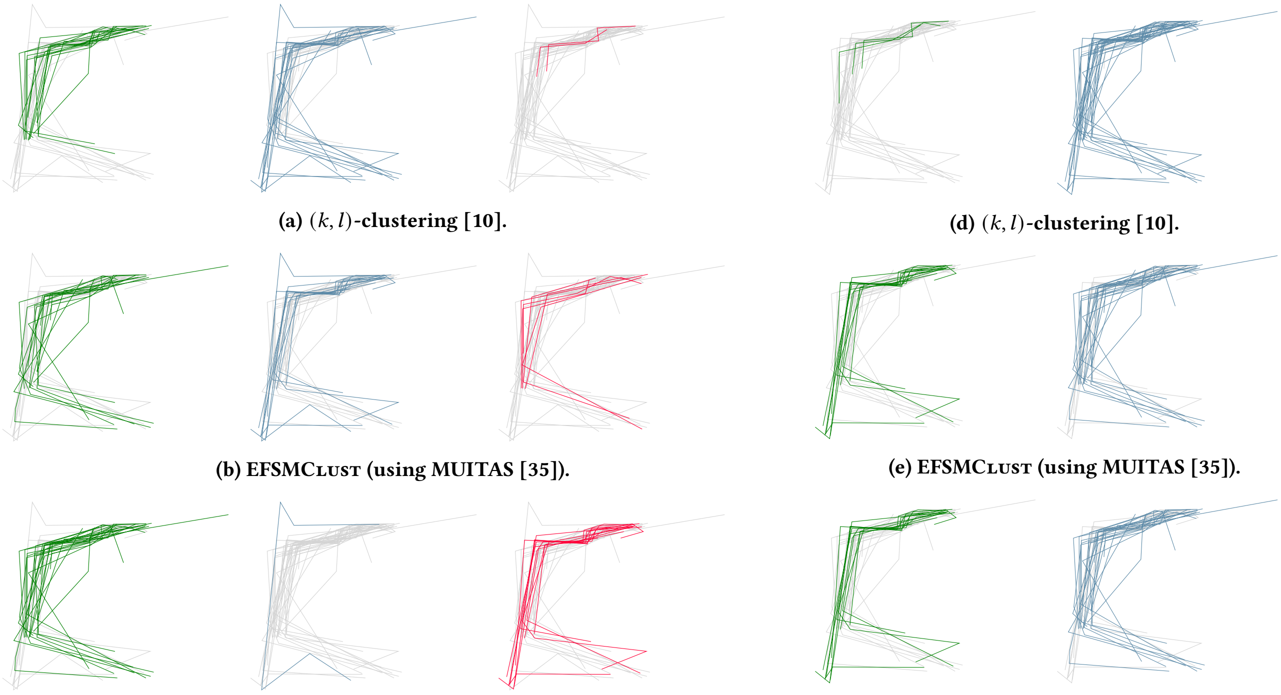


Figure 2: The two "lake walks" clusters. Image credits: CC BY M. van Kreveld, M. Löffler, F. Staals [41, fig. 1(b)].

All three algorithms were able to reconstruct the exact same $k = 2$ clusters. Thus, no quantitative follow-up analysis was needed.

CVRR Trajectory Clustering Dataset. The *CVRR Trajectory Clustering Dataset* [30] is a collection of simulated and recorded groups of trajectories that can be used for benchmarking trajectory clustering algorithms. We used all trajectories from $k = 3$ clusters taken from the CROSS dataset which models turns through traffic of a four-way intersection (see Figure 4).

Both the (k, l)-clustering algorithm and the EFSMClust implementation based on the EFSMSim similarity were able to faithfully



(a) (k, l)-clustering [10].

(d) (k, l)-clustering [10].

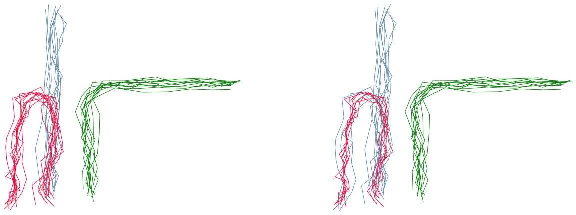
(b) EFSMClust (using MUITAS [35]).

(e) EFSMClust (using MUITAS [35]).

(c) EFSMClust (using EFSMSim).

(f) EFSMClust (using EFSMSim).

Figure 3: Results for clustering data from GeoLife user #20 [52] (left: $k = 3$ clusters, right: $k = 2$ clusters).



(a) (k, l)-clustering and EFSMClust (using EFSMSim).

(b) EFSMClust (using MUITAS).

Figure 4: Data from *CVRR Trajectory Clustering Dataset* [30].

reconstruct the "ground truth" clusters as described in the benchmark; see Figure 4a. In contrast, when using the MUITAS similarity, our approach failed to construct the three expected clusters; see Figure 4b. A closer inspection revealed that the MUITAS similarity, when applied to the representative paths derived from the EFSM inferred of the clusters, resulted in values too small for some of the trajectories to be assigned to their correct cluster. At least for this benchmark dataset, using the EFSMSim similarity which is designed to work with the representative paths inferred from the EFSM is advantageous over using MUITAS which was designed for a different purpose, namely similarity in information retrieval.

5.2 Semantic Trajectories: GeoLife

In the next set of experiments, we used trajectories of the GeoLife dataset [52]. This dataset contains spatio-temporal trajectories capturing human mobility data in Bejing, China. Each trace has been annotated manually with, e. g., the mode of transportation, day of the week, and season and thus contains numerical, ordinal, and categorical attributes. For our experiment, we used traces recorded by "user #20" who, according to their annotations, traveled parts of their way by bike, by bus, or on foot. These traces were recorded at the same time of the day on varying days during different seasons.

Figure 3 (left) shows the results of running the three algorithms under consideration on this data set with the intent of deriving $k = 3$ clusters. The three algorithms resulted in three rather different clusters each. As per its design goals, the (k, l)-clustering algorithm visibly optimizes for minimizing the Fréchet distance within a cluster: This can be seen from the leftmost cluster in Figure 3a in which the only two "short" trajectories are kept together.

Also by design, the EFSMClust approach considers the semantic attributes as well. Comparing Figures 3b and 3c, we see that the EFSMClust approach when used with EFSMSim creates one cluster with a single trajectory in it. This cluster is not present when EFSMClust is used with MUITAS. Instead the trajectories in the third cluster of Figure 3c are distributed among two clusters.

A closer look at the underlying data reveals that the EFSMClust approach using MUITAS detects one cluster corresponding to travel on foot or using a bus and two clusters with bike rides; these clusters differ by the days of the week and seasons. In contrast, the EFSMClust approach using EFSMSim detects the only trajectory in the dataset in which user #20 exclusively traveled by bus and assigns this trajectory alone to the second cluster; see also Figure 5. The first cluster contains all rides by bike and the third cluster contains trajectories in which user #20 walked and, at times, rode

Segment	Label	Day of week	Season
s_0	bike	Tuesday	Autumn
s_1	bike	Wednesday	Autumn
s_2	bike	Tuesday	Autumn

Segment	Label	Day of week	Season
s_0	bus	Sunday	Summer
s_1	bus	Sunday	Summer
s_2	bus	Sunday	Summer
s_3	bus	Sunday	Summer
s_4	bus	Sunday	Summer
s_5	bus	Sunday	Summer

Segment	Label	Day of week	Season
s_0	walk	Tuesday	Summer
s_1	walk	Tuesday	Summer
s_2	walk	Tuesday	Summer
s_3	walk	Tuesday	Summer

Segment	Label	Day of week	Season
s_0	bike	Tuesday	Autumn
s_1	bike	Tuesday	Autumn
s_2	bike	Tuesday	Autumn

Segment	Label	Day of week	Season
s_0	walk	Tuesday	Summer
s_1	walk	Tuesday	Summer
s_2	walk	Tuesday	Summer
s_3	walk	Tuesday	Summer
s_4	walk	Tuesday	Summer

Segment	Label	Day of week	Season
s_0	bike	Wednesday	Winter
s_1	bike	Wednesday	Winter
s_2	bike	Wednesday	Winter
s_3	bike	Monday	Winter

Figure 5: Semantic representative paths created by EFSMClust when used with EFSMSim (left) and MUITAS (right).

Algorithm	FD		MUITAS		EFSMSim	
Variant	max. diameter	avg. radius	min. diameter	avg. radius	min. diameter	avg. radius
(k, l)-clustering [10]	**138.740**	**71.172**	1.110	N/A[5]	0.104	N/A[5]
EFSMClust (using MUITAS [35])	199.747	108.070	**2.375**	3.944	**0.219**	0.619
EFSMClust (using EFSMSim)	201.027	73.767	**2.375**	**4.388**	**0.219**	**0.750**

Table 1: Quantitative comparison of clustering results (Figure 3, left) according to different measures. When using the Fréchet distance (FD), smaller values are better, when using the MUITAS and EFSMSim similarity, larger values are better.

by bus. While the assessment of this clustering from a domain perspective is beyond the scope of this paper, we point out that only the EFSMClust approach using EFSMSim was able to identify the single semantic outlier: the only trajectory in the dataset whose (categorical) mode of transportation constantly was "bus".

We sanity-checked this result by removing the single outlier and re-running all algorithms with $k = 2$. As previously, the (k, l)-clustering still grouped together the "short" trajectories (Figure 3d). When run with $k = 2$, both variants of EFSMClust returned the same clusters: one cluster containing all bike rides and one cluster containing all trajectories containing at least some paths on foot.

We assessed the clusterings shown in Figure 3 (left) using the quantitative measures discussed at the beginning of this section.

Table 1 confirms that (k, l)-clustering optimizes for the purely geometric distance measure it was designed for: The Fréchet distances between the trajectories inducing the maximum diameter of any cluster or inducing the average radius are considerable smaller than for either implementation of EFSMClust. When implemented with EFSMSim, EFSMClust returns clusters with a competitive average radius even though it was not optimized for this measure; this indicates a desirable geometric similarity between the centroids created by (k, l)-clustering and derived from the representative paths of each cluster's EFSM (see [37] for more details).

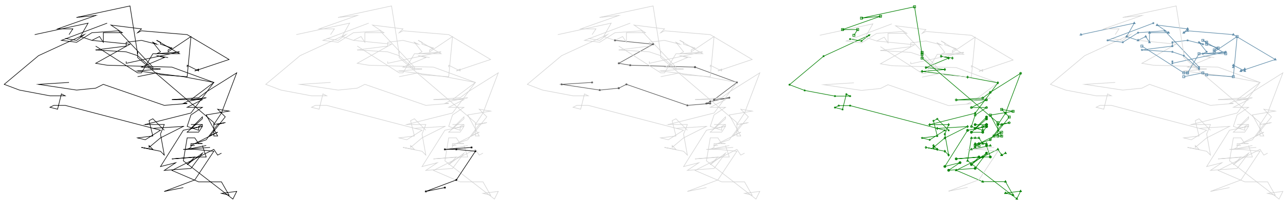
The next two columns of Table 1 list the quantitative data obtained from comparing the resulting clusters based upon the MUITAS and EFSMSim similarity measures. The clusters obtained from running EFSMClust have twice as high similarity measures for the least similar cluster (as measured by the diameter) than the clusters obtained from (k, l)-clustering. Under the MUITAS and EFSMSim similarity measures, the average radius, measured as the smallest similarity between a centroid and any trajectory assigned to this centroid, is higher – thus better – for the clusters generated by EFSMClust using the EFSMSim similarity.

We conclude that, unsurprisingly, the clusters generated by the purely geometric (k, l)-clustering algorithm are superior when evaluated by a purely geometric measure, in our case, the Fréchet distance. Nonetheless, the clusters generated by EFSMClust using the EFSMSim similarity exhibit a competitive average radius. Conversely, again also unsurprisingly, the EFSMClust approach, which was designed to take into account semantic attributes as well, outperforms (k, l)-clustering when the clusters generates are evaluated by either the MUITAS and EFSMSim similarity. A notable insight from this set of experiments is that EFSMClust using EFSMSim was able to identify a single semantic outlier which went undetected for both (k, l)-clustering and EFSMClust using MUITAS.

5.3 Semantic Trajectories: Starkey

The final experiment was run on data from the *Starkey* project [36]. This dataset models deer and elk movement in the Starkey Experimental Forest and Range in northeast Oregon. The time-stamped

[5] By design, the centroids created by (k, l)-clustering do not carry any semantic information. As both MUITAS and EFSMSim are based upon similarities for semantic information, computing a similarity between a trajectory and a centroid while ignoring semantic annotations in not meaningful in the context of this comparison.



(a) All traces. **(b) Cluster 1.** **(c) Cluster 2.** **(d) Cluster 3.** **(e) Cluster 4.**

Figure 6: Traces for animal #880109D01 from the Starkey dataset [36]. Results for partitioning into $k = 4$ clusters.

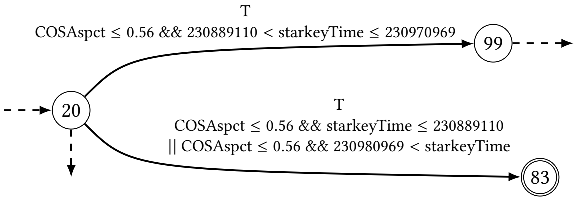

T
COSAspct ≤ 0.56 && 230889110 < starkeyTime ≤ 230970969
99
20
T
COSAspct ≤ 0.56 && starkeyTime ≤ 230889110
|| COSAspct ≤ 0.56 && 230980969 < starkeyTime
83

Figure 7: Excerpt from the EFSM inferred for cluster 3.

data from each animal tracked were matched to a fine grid in which each cell is labeled with background information such as distance to the nearest road, distance to the nearest water source within a pasture, or slope of the terrain; see [36] for more details.

For our experiment, we extracted 11 traces induced by one specific deer (id: 880109D01) over several days. As can be seen from Figure 6a, these traces are not readily seen to form geometry-induced clusters. To investigate whether the semantic information led to more insight, we labeled each tracepoint with the background information obtained from the containing grid cell. In addition, we labeled each tracepoint with semantic annotations regarding the local shape and curvature derived from the underlying spatial information using the algorithm described in our previous work [37].

Running EFSMClust using EFSMSim on this semantically enriched dataset we observe that there are two (spatial) outliers which are isolated into one cluster each; see the Figures 6b and 6c. Thus, when exploring the dataset, at least $k = 4$ clusters are needed.

Figures 6b through 6e visualize the four clusters obtained from running EFSMClust using EFSMSim. The first two clusters contain the (spatial) outliers, the third cluster contains six trajectories, and the fourth cluster contains three trajectories.

The semantic analysis and validation of these clusters is domain-specific to animal behavior; thus, we cannot comment on the semantic quality of our clustering. What our algorithm offers for both the analysis and validation, though, is the EFSM inferred from each cluster. Recall that the EFSM models all paths in a cluster and presents annotations (guards) for each branch. For example, the excerpt shown in Figure 7 lists threshold values for all attributes (see [36] for their semantics) the EFSM construction algorithm considered relevant to determine where to branch to from state 20.

The EFSMClust algorithm thus provides domain experts not only with the results of a clustering process. It also supplements the assignments by documenting, for each cluster, at which point the clustered trajectories behave similarly and where they differ; something purely spatial clustering is unable to provide.

6 CONCLUSIONS

We have presented EFSMClust, a *k-means*-variant to cluster semantically enriched spatio-temporal trajectories. Our approach builds upon the concept of inferring extended finite state machines from a set of semantically enriched trajectories. To incorporate the semantic information into the centroid-based clustering process of *k-means*, we have defined the EFSMSim similarity measure and used the representative paths obtained from the EFSM construction algorithm to generate centroids for each cluster. This information can be used to assess the semantic quality and validity of a clustering. Conversely, switching semantic labels on or off can be used to test hypotheses regarding the influence of semantic attributes.

Our evaluation against a state-of-the-art spatial clustering algorithm and a state-of-the-art semantic similarity function show that EFSMClust is superior for clustering semantically enriched spatio-temporal trajectories. Depending on the quality measure used, it is also competitive for purely spatial trajectories.

What distinguishes our approach from previous work is that we generate a model and a path for each cluster with a human-readable description. This enables analysts to gain deeper insights into the given data and thus opens opportunities for a better interpretability of movement and behavior of the object creating the trajectories.

REFERENCES

[1] H. Alt and M. Godau. 1995. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.* 5 (1995), 75–91. https://doi.org/10.1142/S0218195995000064

[2] L. O. Alvares, V. Bogorny, J. A. F. de Macêdo, B. Moelans, and S. Spaccapietra. 2007. Dynamic Modeling of Trajectory Patterns using Data Mining and Reverse Engineering. In *Challenges in Conceptual Modelling. Tutorials, posters, panels and industrial contributions at the 26th International Conference on Conceptual Modeling - ER 2007 (CRPIT)*, Vol. 83. Australian Computer Society, Auckland, New Zealand, 149–154. http://crpit.scem.westernsydney.edu.au/abstracts/CRPITV83Alvares.html

[3] L. O. Alvares, V. Bogorny, B. Kuijpers, J. A. Fernandes de Macêdo, B. Moelans, and A. A. Vaisman. 2007. A model for enriching trajectories with semantic geographical information. In *15th ACM International Symposium on Geographic Information Systems, ACM-GIS 2007*. ACM, New York, NY, 22. https://doi.org/10.1145/1341012.1341041

[4] D. Arthur and S. Vassilvitskii. 2007. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007*. SIAM, Philadelphia, PA, 1027–1035. http://dl.acm.org/citation.cfm?id=1283383.1283494

[5] D. J. Berndt and J. Clifford. 1994. Using Dynamic Time Warping to Find Patterns in Time Series. In *Knowledge Discovery in Databases: Papers from the 1994 AAAI Workshop. Technical Report WS-94-03*. AAAI Press, Seattle, WA, 359–370.

[6] A. Bhattacharya, J. Eube, H. Röglin, and M. Schmidt. 2020. Noisy, Greedy and Not so Greedy k-Means++. In *28th Annual European Symposium on Algorithms (ESA 2020) (Leibniz International Proceedings in Informatics (LIPIcs))*, Vol. 173. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 18:1–18:21. https://doi.org/10.4230/LIPIcs.ESA.2020.18

[7] J. Blömer, C. Lammersen, M. Schmidt, and C. Sohler. 2016. Theoretical Analysis of the k-Means Algorithm - A Survey. In *Algorithm Engineering - Selected Results and Surveys*. Lecture Notes in Computer Science, Vol. 9220. Springer, Cham,

81–116. https://doi.org/10.1007/978-3-319-49487-6_3

[8] M. Brankovic, K. Buchin, K. Klaren, A. Nusser, A. Popov, and S. Wong. 2020. (k, l)-Medians Clustering of Trajectories Using Continuous Dynamic Time Warping. In *SIGSPATIAL '20: 28th International Conference on Advances in Geographic Information Systems*. ACM, New York, NY, 99–110. https://doi.org/10.1145/3397536.3422245

[9] V. Brum-Bastos, M. Łoś, J. A. Long, T. Nelson, and U. Demšar. 2021. Context-aware movement analysis in ecology: a systematic review. *International Journal of Geographical Information Science* (2021). https://doi.org/10.1080/13658816.2021.1962528 In press.

[10] K. Buchin, A. Driemel, N. van de L'Isle, and A. Nusser. 2019. klcluster: Center-based Clustering of Trajectories. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2019*. ACM, New York, NY, 496–499. https://doi.org/10.1145/3347146.3359111

[11] M. E. Celebi and H. A. Kingravi. 2015. Linear, deterministic, and order-invariant initialization methods for the k-means clustering algorithm. In *Partitional clustering algorithms*. Springer, Berlin, 79–98. https://doi.org/10.1007/978-3-319-09259-1_3

[12] L. Chen, M. T. Özsu, and V. Oria. 2005. Robust and Fast Similarity Search for Moving Object Trajectories. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, New York, NY, 491–502. https://doi.org/10.1145/1066157.1066213

[13] J. De Groeve, N. Van de Weghe, N. Ranc, T. Neutens, L. Ometto, O. Rota-Stabelli, and F. Cagnacci. 2016. Extracting spatio-temporal patterns in animal trajectories: An ecological application of sequence analysis methods. *Methods in Ecology and Evolution* 7, 3 (2016), 369–379. https://doi.org/10.1111/2041-210X.12453

[14] S. Dodge, G. Bohrer, R. Weinzierl, S. C. Davidson, R. Kays, D. Douglas, S. Cruz, J. Han, D. Brandes, and M. Wikelski. 2013. The environmental-data automated track annotation (Env-DATA) system: linking animal tracks with environmental data. *Movement Ecology* 1, 1 (2013), 1–14. https://doi.org/10.1186/2051-3933-1-3

[15] T. Eiter and H. Mannila. 1994. *Computing discrete Fréchet distance.* Technical Report CD-TR 94/64. Technical University of Vienna.

[16] A. S. Furtado, D. Kopanaki, L. O. Alvares, and V. Bogorny. 2016. Multidimensional Similarity Measuring for Semantic Trajectories. *Trans. GIS* 20, 2 (2016), 280–298. https://doi.org/10.1111/tgis.12156

[17] M. C. González, C. A. Hidalgo, and A. Barabási. 2008. Understanding individual human mobility patterns. *CoRR* abs/0806.1256, 7196 (2008), 779–782.

[18] T. F. Gonzalez. 1985. Clustering to Minimize the Maximum Intercluster Distance. *Theor. Comput. Sci.* 38 (1985), 293–306. https://doi.org/10.1016/0304-3975(85)90224-5

[19] J. Gudmundsson and M. Horton. 2017. Spatio-Temporal Analysis of Team Sports. *ACM Comput. Surv.* 50, 2 (2017), 22:1–22:34. https://doi.org/10.1145/3054132

[20] J. Gudmundsson, P. Laube, and T. Wolle. 2012. Computational Movement Analysis. In *Springer Handbook of Geographic Information*. Springer, Berlin, 423–438. https://doi.org/10.1007/978-3-540-72680-7_22

[21] J. Gudmundsson, A. Thom, and J. Vahrenhold. 2012. Of motifs and goals: mining trajectory data. In *Proc. 20th Intl. Conf. on Advances in Geographic Information Systems - SIGSPATIAL '12*. ACM, New York, NY, 129–138. https://doi.org/10.1145/2424321.2424339

[22] R. H. Güting, F. Valdés, and M. L. Damiani. 2015. Symbolic Trajectories. *ACM Trans. Spatial Algorithms Syst.* 1, 2 (2015), 7:1–7:51. https://doi.org/10.1145/2786756

[23] H. Issa and M. L. Damiani. 2016. Efficient Access to Temporally Overlaying Spatial and Textual Trajectories. In *IEEE 17th International Conference on Mobile Data Management, MDM 2016*. IEEE Computer Society, Los Alamitos, CA, 262–271. https://doi.org/10.1109/MDM.2016.47

[24] J. S. Larson, E. T. Bradlow, and P. S. Fader. 2005. An exploratory look at supermarket shopping paths. *International Journal of Research in Marketing* 22, 4 (2005), 395–414. https://doi.org/10.1016/j.ijresmar.2005.09.005

[25] A. L. Lehmann, L. O. Alvares, and V. Bogorny. 2019. SMSM: a similarity measure for trajectory stops and moves. *Int. J. Geogr. Inf. Sci.* 33, 9 (2019), 1847–1872. https://doi.org/10.1080/13658816.2019.1605074

[26] Z. Li, J. Han, M. Ji, L. Tang, Y. Yu, B. Ding, J. Lee, and R. Kays. 2011. MoveMine: Mining moving object data for discovery of animal movement patterns. *ACM Trans. Intell. Syst. Technol.* 2, 4 (2011), 37:1–37:32. https://doi.org/10.1145/1989734.1989741

[27] C. Liu and C. Guo. 2020. STCCD: Semantic trajectory clustering based on community detection in networks. *Expert Syst. Appl.* 162 (2020), 113689. https://doi.org/10.1016/j.eswa.2020.113689

[28] H. Liu and M. Schneider. 2012. Similarity measurement of moving object trajectories. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS@SIGSPATIAL 2012*. ACM, New York, NY, 19–22. https://doi.org/10.1145/2442968.2442971

[29] S. P. Lloyd. 1982. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* 28, 2 (1982), 129–136. https://doi.org/10.1109/TIT.1982.1056489

[30] B. Morris and M. M. Trivedi. 2009. Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*. IEEE Computer Society, Piscataway, NJ, 312–319. https://doi.org/10.1109/CVPR.2009.5206559

[31] S. B. Needleman and C. D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48, 3 (1970), 443–453. https://doi.org/10.1016/0022-2836(70)90057-4

[32] P. Newson and J. Krumm. 2009. Hidden Markov map matching through noise and sparseness. In *17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2009*. ACM, New York, NY, 336–343. https://doi.org/10.1145/1653771.1653818

[33] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares. 2008. A clustering-based approach for discovering interesting places in trajectories. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), March 16-20, 2008*. ACM, New York, NY, 863–868. https://doi.org/10.1145/1363686.1363886

[34] C. Parent, S. Spaccapietra, C. Renso, G. L. Andrienko, N. V. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. A. F. de Macêdo, N. Pelekis, Y. Theodoridis, and Z. Yan. 2013. Semantic trajectories modeling and analysis. *ACM Comput. Surv.* 45, 4 (2013), 42:1–42:32. https://doi.org/10.1145/2501654.2501656

[35] L. M. Petry, C. A. Ferrero, L. O. Alvares, C. Renso, and V. Bogorny. 2019. Towards semantic-aware multiple-aspect trajectory similarity measuring. *Trans. GIS* 23, 5 (2019), 960–975. https://doi.org/10.1111/tgis.12542

[36] M. M. Rowland. 1997. *The Starkey Project: History Facilities, and Data Collection Methods for Ungulate Research*. Vol. 396. US Department of Agriculture, Forest Service, Pacific Northwest Research Station.

[37] J. Seep and J. Vahrenhold. 2019. Inferring Semantically Enriched Representative Trajectories. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Computing with Multifaceted Movement Data, MOVE++@SIGSPATIAL 2019*. ACM, New York, NY, 3:1–3:4. https://doi.org/10.1145/3356392.3365220

[38] K. Sila-Nowicka, J. Vandrol, T. Oshan, J. A. Long, U. Demsar, and A. S. Fotheringham. 2016. Analysis of human mobility patterns from GPS trajectories and contextual information. *Int. J. Geogr. Inf. Sci.* 30, 5 (2016), 881–906. https://doi.org/10.1080/13658816.2015.1100731

[39] Yaguang Tao, Alan Both, Rodrigo I. Silveira, Kevin Buchin, Stef Sijben, Ross S. Purves, Patrick Laube, Dongliang Peng, Kevin Toohey, and Matt Duckham. 2021. A comparative analysis of trajectory similarity measures. *GIScience & Remote Sensing* 58, 5 (2021), 643–669. https://doi.org/10.1080/15481603.2021.1908927

[40] S. Tasnim, J. Caldas, N. Pissinou, S. S. Iyengar, and Z. Ding. 2018. Semantic-Aware Clustering-based Approach of Trajectory Data Stream Mining. In *2018 Intl. Conf. Computing, Networking and Communications*. IEEE Computer Society, Piscataway, NJ, 88–92. https://doi.org/10.1109/ICCNC.2018.8390371

[41] M. J. van Kreveld, M. Löffler, and F. Staals. 2017. Central trajectories. *J. Comput. Geom.* 8, 1 (2017), 366–386. https://doi.org/10.20382/jocg.v8i1a14

[42] A. Vattani. 2011. *k*-means Requires Exponentially Many Iterations Even in the Plane. *Discret. Comput. Geom.* 45, 4 (2011), 596–616. https://doi.org/10.1007/s00454-011-9340-1

[43] M. Vlachos, D. Gunopulos, and G. Kollios. 2002. Discovering Similar Multidimensional Trajectories. In *Proceedings of the 18th International Conference on Data Engineering*. IEEE Computer Society, Piscataway, NJ, 673–684. https://doi.org/10.1109/ICDE.2002.994784

[44] N. Walkinshaw, R. Taylor, and J. Derrick. 2013. Inferring Extended Finite State Machine models from software executions. In *20th Working Conference on Reverse Engineering, WCRE 2013, October 14-17, 2013*. IEEE Computer Society, Piscataway, NJ, 301–310. https://doi.org/10.1109/WCRE.2013.6671305

[45] X. Wang, G. Li, G. Jiang, and Z. Shi. 2013. Semantic trajectory-based event detection and event pattern mining. *Knowl. Inf. Syst.* 37, 2 (2013), 305–329. https://doi.org/10.1007/s10115-011-0471-8

[46] X. Xiao, Y. Zheng, Q. Luo, and X. Xie. 2014. Inferring social ties between users with human location history. *J. Ambient Intell. Humaniz. Comput.* 5, 1 (2014), 3–19. https://doi.org/10.1007/s12652-012-0117-z

[47] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer. 2013. Semantic trajectories: Mobility data computation and annotation. *ACM Trans. Intell. Syst. Technol.* 4, 3 (2013), 49:1–49:38. https://doi.org/10.1145/2483669.2483682

[48] G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang. 2017. A review of moving object trajectory clustering algorithms. *Artificial Intelligence Review* 47, 1 (2017), 123–144. https://doi.org/10.1007/s10462-016-9477-7

[49] Z. Zhang, K. Huang, and T. Tan. 2006. Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes. In *18th International Conference on Pattern Recognition (ICPR 2006)*. IEEE Computer Society, Piscataway, NJ, 1135–1138. https://doi.org/10.1109/ICPR.2006.392

[50] Y. Zheng. 2015. Trajectory Data Mining: An Overview. *ACM Trans. Intell. Syst. Technol.* 6, 3 (2015), 29:1–29:41. https://doi.org/10.1145/2743025

[51] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W. Ma. 2008. Understanding mobility based on GPS data. In *UbiComp 2008: Ubiquitous Computing, 10th International Conference*. ACM, New York, NY, 312–321. https://doi.org/10.1145/1409635.1409677

[52] Y. Zheng, L. Wang, R. Zhang, X. Xie, and W. Ma. 2008. GeoLife: Managing and Understanding Your Past Life over Maps. In *9th International Conference on Mobile Data Management (MDM 2008)*. IEEE Computer Society, Piscataway, NJ, 211–212. https://doi.org/10.1109/MDM.2008.20