Knowl Inf Syst (2011) 28:117–147
DOI 10.1007/s10115-010-0316-x

REGULAR PAPER

Clustering uncertain trajectories

Nikos Pelekis · Ioannis Kopanakis · Evangelos E. Kotsifakos · Elias Frentzos · Yannis Theodoridis

Received: 15 January 2010 / Revised: 22 April 2010 / Accepted: 11 June 2010 /
Published online: 1 July 2010

Abstract Knowledge discovery in Trajectory Databases (TD) is an emerging field which has recently gained great interest. On the other hand, the inherent presence of uncertainty in TD (e.g., due to GPS errors) has not been taken yet into account during the mining process. In this paper, we study the effect of uncertainty in TD clustering and introduce a three-step approach to deal with it. First, we propose an intuitionistic point vector representation of trajectories that encompasses the underlying uncertainty and introduce an effective distance metric to cope with uncertainty. Second, we devise CenTra, a novel algorithm which tackles the problem of discovering the Centroid Trajectory of a group of movements taking into advantage the local similarity between portions of trajectories. Third, we propose a variant of the Fuzzy C-Means (FCM) clustering algorithm, which embodies CenTra at its update procedure. Finally, we relax the vector representation of the Centroid Trajectories by introducing an algorithm that post-processes them, as such providing these mobility patterns to the analyst with a more intuitive representation. The experimental evaluation over synthetic and real world TD demonstrates the efficiency and effectiveness of our approach.

Keywords Centroid trajectory · Intuitionistic fuzzy clustering · Uncertainty · Trajectory similarity · Trajectory clustering

N. Pelekis (✉)
Department of Statistics and Insurance Science, University of Piraeus, Piraeus, Greece
e-mail: npelekis@unipi.gr

E. E. Kotsifakos · E. Frentzos · Y. Theodoridis
Department of Informatics, University of Piraeus, Piraeus, Greece
e-mail: ek@unipi.gr

E. Frentzos
e-mail: efrentzo@unipi.gr

Y. Theodoridis
e-mail: ytheod@unipi.gr

I. Kopanakis
Technological Educational Institute of Crete, Crete, Greece
e-mail: i.kopanakis@emark.teicrete.gr

1 Introduction

With the integration of wireless communications and positioning technologies, TD have become increasingly popular, posing great challenges to the data mining community [20]. On the other hand, since a TD consists of movements of objects, which record their position as it evolves over time, the concept of *uncertainty* appears in various ways; data imprecision due to sampling and/or measurement errors [33], uncertainty in querying and answering [35], fuzziness by purpose during pre-processing for preserving anonymity [1], and so on. The importance of mining uncertain data has been recognized [39], however, though uncertainty is inherent in TD, to the best of our knowledge there is no related work in the spatio-temporal data mining literature that studies its effect in the knowledge discovery process.

Clustering of trajectories into separate collections involves partitioning of a TD into clusters (groups), so that each cluster contains proximate trajectories according to some distance measure. Previous research has mostly focused on clustering of point data that trajectories do not conform to. This means that well-known clustering algorithms (e.g., k-means [28], BIRCH [42], DBSCAN [15], STING [37]) cannot be directly applied. As such the idea of measuring the similarity between two trajectories is an attractive solution that has been utilized as the mean to cluster trajectories. Many approaches have been introduced in the literature that try to quantify the (dis)-similarity between trajectories, dealing with basic trajectory features, [36,40,9,10,31,14]. Although some clustering approaches deal with noise in clustering spatiotemporal data [5], neither of the aforementioned clustering algorithms nor the previously cited approaches for similarity search deal with the inherent uncertainty in TD.

On the other hand, clustering approaches based on fuzzy logic [41], such as FCM [7], consider uncertainty by allowing each data element to belong to different clusters by a certain degree of membership. Considering that input vector values are subject to uncertainty due to imprecise measurements, noise or sampling errors, the distances that determine the membership of a point to a cluster are also subject to uncertainty. Therefore, the possibility of erroneous membership assignments in the clustering process is evident. Moreover, current fuzzy clustering approaches do not utilize any information about uncertainty at the elementary level of the data points, which for the case of trajectories are the spatial locations of the objects recorded in temporal order.

In this paper, we introduce a three-step approach to deal with uncertainty in TD and its effect on trajectory clustering. We initially adopt a symbolic representation and model trajectories as sequences of regions (i.e., wherefrom a moving object passes) accompanied with *intuitionistic fuzzy values*, i.e., elements of an intuitionistic fuzzy set. Intuitionistic fuzzy sets (IFS) [6] are generalized fuzzy sets [41] that can be useful in coping with the *hesitancy* originating from imprecise information. IFS elements are characterized by two values representing, respectively, their *belongingness* and *non-belongingness* to this set. In the case of TD where this set is the region that a trajectory possibly crosses, the above values represent the probabilities of presence and non-presence in the area. In order to exploit this information, we define a novel distance metric especially designed to operate on such intuitionistic fuzzy vectors, having as goal to incorporate it in some variant of the FCM algorithm that will effectively cluster trajectories under uncertainty.

The success of any FCM-variant algorithm depends on the way that cluster centroids are driven towards the correct direction in each iteration of the algorithm. This direction at each step of the algorithm is actually decided with the help of a similarity function. However, in the TD setting where trajectories are complex objects of different lengths, varying sampling rates, different speeds, possible outliers and different scaling factors, even the most efficient

similarity function would most probably fail in different applications. We argue that we can succeed better clustering results if instead of using *global* similarity functions between whole trajectories, we exploit *local* similarity properties between portions of the trajectories. Based on this idea, at the second step of our approach, we propose *CenTra*, a novel density- as well as similarity-based algorithm to tackle the problem of discovering the *Centroid* of a group of trajectories. Among the advantages of *CenTra* in contrast to related work, we distinguish its ability to represent complex time-aware mobility patterns that demonstrate in an intuitive way the *growth* of the pattern in space-time.

At the third step of our approach, we propose a new trajectory clustering algorithm, called *CenTR-I-FCM*, which utilizes *CenTra* in its centroid update step, uses a global uncertainty-supporting similarity function to group trajectories at a higher level, and iteratively refines the results using local similarity between sub-trajectories. This algorithm has the efficiency advantages of partitioning clustering algorithms (in comparison with the higher processing cost of density-based algorithms), whereas produces non-spherical clusters due to the inclusion of *CenTra*, that recognizes representative movements of any shape.

Except for clustering uncertain trajectories, the implicit outcome of the aforesaid approach, namely the centroid trajectories for each of the resulted clusters, has very useful applications in several fields that could gain insight from such mobility patterns. Example domains include traffic engineering, climatology, anthropography and zoology that study, vehicle position data, hurricane track data, pedestrian and animal movement data, respectively. Moreover, consider for example the domain of visual analytics on movement data [3] that supports the analysis of all the above mentioned fields. In this field, it is meaningless to visualize even very small datasets, as the human eye cannot distinguish any movement pattern due to the immense size of the data. As such, more intuitive representations are required. Aiming at supporting even higher level analysis tasks we devise an algorithm, called *TX-CenTra*, that improves the representation of *CenTra* by relaxing its point vector representation. The idea is to identify the maximal time periods wherein the mobility pattern presents uniform behavior.

Summarizing our contributions:

- we propose a time-based segmentation of a TD that leads to an intuitionistic fuzzy vector representation of trajectories, which enables the clustering of trajectories by existing (fuzzy or not) clustering algorithms;
- we define a global distance metric on the previous trajectory representation, which outperforms its competitors proposed in the literature;
- we tackle the problem of identifying the centroid of a bunch of trajectories using density and local similarity properties;
- we propose a novel modification of the FCM algorithm for clustering complex trajectory datasets based on the above distance measure and the idea of the centroid trajectory;
- we devise an algorithm that relaxes the time-based point representation of the centroid trajectories, allowing the modeling of such mobility patterns in a higher level abstraction;
- we conduct a comprehensive set of experiments over several synthetic and a real trajectory dataset, in order to evaluate our approach.

The rest of this paper is structured as follows: Sect. 2 discusses related work. In Sect. 3, we introduce the intuitionistic vector representation of trajectories. The proposed similarity measure is defined in Sect. 4 while in Sect. 5 we describe the CenTra, the CenTR-I-FCM and the TX-CenTra algorithms. In Sect. 6, we conduct an experimental study over synthetic and real TD in order to evaluate our approach. Finally, the conclusions of this study along with ideas for future work are summarized in Sect. 7.

2 Related work

In this section, we review existing works in the domains related with the current work, namely, uncertainty in TD, TD clustering, and intuitionistic fuzzy set theory.

2.1 Representing uncertainty in TD

Probably, the most recognized notion of uncertainty in TD is the uncertainty of the trajectory representation, which means that the location of a moving object stored in a TD will not represent its real location due to a variety of reasons. Although this kind of uncertainty may be inherited by GPS erroneous measurements, its major source in TD is the interpolation method (usually linear) used to capture the complete movement of the moving object and estimate the object's location at timestamps in-between sampled positions. In [33], the authors define the notion of sampling error at a sampled position P_1 at a timestamp t_1 and they study the error behavior across the time axis. They prove the intuitively expected result that by increasing the sampling rate, the sample positions better approximate movement, and the error introduced by sampling is decreased. In [35], a model for uncertain trajectories is proposed that associates an uncertainty threshold ε to the whole trajectory, while a set of uncertainty operators were introduced so as to incorporate uncertainty into user queries. This approach results in trajectories with uncertainty modeled as cylindrical volumes in 3D space. Therefore, each trajectory point (x, y, t) is associated with an ε-uncertainty area which is actually a horizontal disk with radius ε centered at (x, y, t). In order to reduce the complexity of handling this kind of spherical neighborhoods, in [19] square uncertainty areas were introduced.

2.2 TD clustering

Clustering is one of the most popular data mining tasks with numerous applications. As already mentioned, the vast majority of the proposed clustering algorithms, such as *k*-means [28], BIRCH [42], DBSCAN [15], and STING [37] are tailored to work with point data, making thus their application to TD not a straightforward task. During the last decade, several approaches have been proposed in the literature so as to enable well-known algorithms to operate on trajectories. Most of these approaches are inspired by the time series analysis domain, and propose trajectory similarity measures as the vehicle to group trajectories; they usually focus on the movement shape of trajectories, which are usually considered as 2D or 3D time series data [36,40,9,10]. None of the previous approaches considers the underlying uncertainty. On the other hand, clustering approaches based on fuzzy logic [41], such as Fuzzy C-Means (FCM) [7] and its variants are competitive to conventional clustering algorithms, especially for real-world applications [24]. In those algorithms, the degree of membership of a data vector to a cluster is considered as a function of its distance from the cluster centroid or from other representative vectors of the cluster. However, directly mapping these techniques in TD is not straightforward, mainly due to the complex nature of trajectories (a question that arises, for example, is about the nature of the cluster centroid in a group of trajectories).

In the past, [18] and [8] proposed probabilistic algorithms for clustering short trajectories using a regression mixture model. Subsequently, unsupervised learning is carried out by using EM algorithm to determine the cluster memberships in the model. In this approach, the issue of uncertainty is not taken into account, while representation of cluster centroids is out of the scope of these papers. What is more, in our approach we make no assumption

about the size of the trajectories or whether they conform to some regression model, since we are interested in complex, real-world objects following arbitrary movement patterns.

Focusing on applications where the analysis requires thorough examination of the clustering semantics along the time dimension, Nanni and Pedreschi [30], proposed T-OPTICS, an adaptation of the OPTICS [4] density-based clustering algorithm to trajectory data, based on another notion of distance between trajectories. In addition, aiming at exploiting the semantics of the temporal dimension to improve the quality of trajectory clustering, the authors present TF-OPTICS, based on the approach of *temporal focusing*. The idea is to generalize trajectory clustering by focusing on the temporal dimension; basically by enlarging the search space of interesting clusters by considering the restrictions of the source trajectories onto sub-intervals of time. The previously mentioned temporal intervals are given by the user, while TF-OPTICS algorithm essentially consists in reiterated executions of T-OPTICS on segments of trajectories, obtained by properly clipping the original ones. In comparison with our approach, this approach is not applicable in symbolic representations of trajectories that encapsulate uncertainty, while the modeling of aggregated mobility patterns of groups of trajectories is also off the points of interest of this paper.

Recently, [25] proposed TRACLUS, a partition-and-group framework for clustering trajectories which enables the discovery of common sub-trajectories, based on a trajectory partitioning algorithm that uses the minimum description length principle. TRACLUS clusters trajectories as line segments (sub-trajectories) independently of whether the whole trajectories belong to different or the same clusters; for this reason, a variant of DBSCAN for line segments is proposed [25]. Finally, the notion of the *representative trajectory* of a cluster is provided. The fundamental difference of TRACLUS with our approach is that we cluster trajectories as a whole. Furthermore, contrary to our approach, the temporal information is not considered in [25], while the proposed algorithm for identifying the representative trajectory of a cluster primarily supports straight movement patterns and cannot identify complex (e.g. circular) motions, which are usual in real world applications. Moreover, [25] does by no means deal with the uncertainty in TD.

2.3 Intuitionistic fuzzy sets and similarity

Regarding the theoretical foundations of fuzzy and intuitionistic fuzzy sets, these are described in [41,6]. In the following paragraphs, we briefly outline the basic notions used in this paper.

Definition 1 Let a set E of elements be fixed. A fuzzy set $\tilde{A}$ on E is an object of the form

$$\tilde{A} = \left\{ \langle x, \mu_{\tilde{A}}(x) \rangle \middle| \, x \in E \right\}$$

where $\mu_{\tilde{A}} : \mathrm{E} \to [0, 1]$ defines the degree of membership of element $x \in E$ to set $\tilde{A} \subset E$. For every element $x \in E$, $0 \leq \mu_{\tilde{A}}(x) \leq 1$.

Definition 2 An intuitionistic fuzzy set A on E is an object of the form

$$A = \{ \langle x, \mu_A(x), \gamma_A(x) \rangle | \, x \in E \}$$

where $\mu_A : E \to [0, 1]$ and $\gamma_A : E \to [0, 1]$ define the degree of membership and non-membership, respectively, of element $x \in E$ to set $A \subset E$. For every element $x \in E$ it holds that $0 \leq \mu_A(x) \leq 1, 0 \leq \gamma_A(x) \leq 1$ and $0 \leq \mu_A(x) + \gamma_A(x) \leq 1$. For every $x \in E$, if $\gamma_A(x) = 1 - \mu_A(x)$, A represents a fuzzy set. The function $\pi_A(x) = 1 - \gamma_A(x) - \mu_A(x)$ represents the degree of *hesitancy* of element $x \in E$ to set $A \subset E$.

Table 1 Table of notations

Notation	Description
$E = \{x_1, x_2, \ldots, x_n\}$	A finite space of n elements x_i
$\mu_A(x), \gamma_A(x), \pi_A(x)$	The membership, non-membership, and hesitancy of $x \in E$ in an intuitionistic fuzzy set A
D, ls, T_i, n_i, ls_i	A trajectory database, its lifespan, a single trajectory, its number of segments and its lifespan
$G, c_{k,l}, gap$	A regular grid used to approximate trajectories, a single cell ($1 \leq k \leq m$ and $1 \leq l \leq n$), and cell $c_{1,1}$
$\overline{T}_i, r_{i,j}$	The approximation of trajectory T_i over G and its j-th approximated region
$\text{UnTra}(\overline{T}_i), ur_{i,j}$	The approximated uncertain trajectory T_i over G, and its j-th approximated uncertain region
$I - \text{UnTra}(\overline{T}_i)$	The intuitionistic approximated uncertain trajectory T_i over G
$D_{\text{total}} (= \lvert A - B \rvert_{\text{IFS}}^{\text{UnTra}}), D_{\text{IFS}}, D_{\text{UnTra}}$	The distance measure between (a) two *I-UnTra*s, (b) two IFS, and, (c) two *UnTra*s
$mbr(ur), \lvert ur_i - ur_j \rvert_{\min}, \lvert ur_i - ur_j \rvert_{ext}$	The minimum bounding rectangle of uncertain region ur, and the minimum and external distances between the $mbr(ur_i)$ and $mbr(ur_j)$
M_A, Γ_A, Π_A	The sets containing the values of membership, non-membership and hesitancy for every member of the fuzzy set A
U, c, N	A $(c \times N)$-dimensional matrix of reals $u_{ik} \in [0, 1]$, the number of clusters, the cardinality of the data vectors

The plethora and importance of the potential applications of intuitionistic fuzzy sets have drawn the attention of many researchers that have proposed various kinds of similarity measures between intuitionistic fuzzy sets. Example applications include identification of functional dependency relationships between concepts in data mining systems, approximate reasoning, pattern recognition and others. A variety of similarity measures between intuitionistic fuzzy sets have been proposed, including S_C [11,12], S_H [21], S_L [16], S_O by [27], S_{DC} [13], S_{HB} [29], S_e^p, S_s^p and S_h^p [43], S_{HY}^1, S_{HY}^2 and S_{HY}^3 [22]. Recently, [26] provided a comprehensive survey and a detailed comparison of those measures, pointing out the weaknesses of each one.

In the following sections, we will present in detail our approach for TD clustering that takes uncertainty into consideration. The notation used in the rest of the paper is summarized in Table 1.

3 Intuitionistic fuzzy vector representation of trajectories

Representing trajectory data stored in TD by means of intuitionistic fuzzy sets is challenging. Formally, let $D = \{T_1, T_2, \ldots, T_N\}$ be a TD consisting of N trajectories. Assuming linear interpolation between consecutive time-stamped positions, a trajectory $T_i = \langle (x_{i,0}, y_{i,0}, t_{i,0}), \ldots, (x_{i,ni}, y_{i,ni}, t_{i,ni}) \rangle$, consists of a sequence of $n_i > 0$ line segments in 3D space, where the j-th segment interpolates positions sampled at time $t_{i,j-1}$ and $t_{i,j}$.

A basic requirement for applying existing clustering algorithms (usually designed for point vector data) into TD, is to transform trajectories in a space where each T_i is represented

as p-dimensional point. We therefore propose an approximation technique and define the dimensionality of trajectories by dividing the lifespan of each trajectory in p sub-intervals (e.g., 1 min periods). Regarding the spatial dimension, we assume a regular grid of equal rectangular cells with user-defined size (e.g., $100 \times 100\,\mathrm{m}^2$); in each cell an identifier is also attached. Given this setting, and inspired by the Piecewise Aggregate Approximation (PAA) technique [23], we propose a method that partitions T_i into $p << n_i$ equi-sized temporal periods and substitutes the trajectory 3D line segments of each period with the set of the grid cells that T_i crosses during this period. More formally:

Definition 3 Given (i) a regular grid G of granularity $m \times n$ consisting of cells $c_{k,l}$ ($1 \leq k \leq m$ and $1 \leq l \leq n$), (ii) a trajectory T_i as a sequence of n_i line segments, the lifespan ls of all trajectories in the trajectory database D, and (iii) a target dimension $p << n_i$, the *approximate trajectory* $\bar{T}_i = \langle r_{i,1}..r_{i,p}\rangle$of trajectory T_i is the one resulted by T_i when all trajectory triplets $(x_{i,j}, y_{i,j}, t_{i,j})$ of T_i found inside a temporal period

$$p_j = \left[\frac{ls \cdot (j-1)}{p}, \frac{ls \cdot j}{p}\right], \quad 1 \leq j \leq p$$

are replaced by a region $r_{i,j}$, which is composed by the set of cells $c_{k,l}$ crossed by T_i during p_j .

The advantage of this technique is that it allows us to view and store *all* trajectories in D as vectors in the *same* user-defined dimensionality p, where each value of the vector corresponds to a dynamic time-ordered list of cells crossed by the trajectory. Note that depending on the choice of the spatial and temporal granularity a trajectory may introduce *gaps* (i.e., regions with empty set of cells due to the fact that there is no motion during the particular period of time).

Next, following the approach proposed in [19], we model the *Uncertain Trajectory* (UnTra) of $\bar{T}_i$ over G to be $\bar{T}_i$ with its regions $r_{i,j}$ been extended to cover some neighboring cells, the ones that are touched by the ε-buffer [19] of the initial trajectory T_i. (A similar idea is also found in, where each trajectory is modeled as a circular disk evolving in the temporal dimension, thus forming a cylindrical volume.) Formally:

Definition 4 Given an approximate trajectory $\bar{T}_i = \langle r_{i,1}..r_{i,p}\rangle$ and an uncertainty threshold ε, the *Uncertain Trajectory* $\mathrm{UnTra}\left(\bar{T}_i\right) = \langle ur_{i,1}..ur_{i,p}\rangle$ of $\bar{T}_i$ over G is obtained by replacing each region $r_{i,j}$with an uncertain region $ur_{i,j}$ consisting of the set of cells $c_{k,l}$that the ε-buffer of T_i touches/crosses during p_j.

To clarify the aforementioned definitions through an example, assume a simple trajectory T_i consisting of 6 (i.e. $n_i = 6$) line segments, which, when it is overlaid on a grid, it crosses some of its cells (Fig. 1a). Figure 1b illustrates the UnTra counterpart of Fig. 1a with $\varepsilon = 1$. Assuming a target dimension $p = 5$, T_i is approximated by $\mathrm{UnTra}(\bar{T}_i)$, which simply consists of five uncertain regions, reflecting the partitioning of the above grey cells in five subsets (i.e. differently colored regions in Fig. 1c) with respect to the lifespan of T_i. Without loss of generality, in the rest of the paper, we assume that all trajectories in D have the same uncertainty threshold ε.

Based on the above representation, in the following we propose an intuitionistic fuzzy vector representation of a trajectory. The idea is to model each region $ur_{i,j}$ of an UnTra as an intuitionistic fuzzy set $A \subset E$ of the regions universe E that belongs to A by a degree $\mu_A(ur_{i,j})$ and does not belong to A by a degree $\gamma_A(ur_{i,j})$ (recall Definition 2). Let us, for the moment, assume that we work in the continuous space. Assuming no uncertainty in the

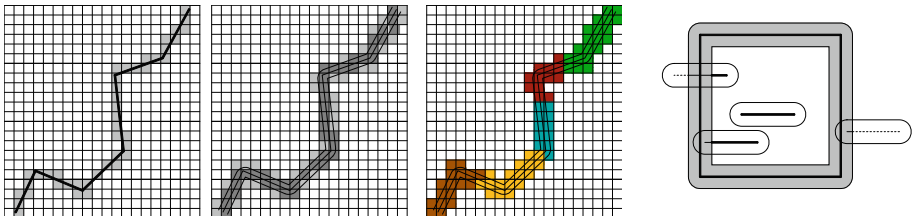

Fig. 1 **a** Crossed cells by trajectory, **b** by UnTra with $\varepsilon = 1$, and **c** UnTra with $p = 5$. **d** Representation of membership, non-membership, and hesitancy in the continuous space

temporal dimension (i.e., each $ur_{i,j}$ is only subject to spatial uncertainty), Fig. 1d depicts one cell $c_{k,l}$ and two auxiliary buffers in grey color, one exterior and one interior, in distance ε from the cell; these buffers are formed, respectively, as the *Minkowski sum* $(c_{k,l} \oplus \varepsilon)$ and *Minkowski difference* $(c_{k,l} \ominus \varepsilon)$ of $c_{k,l}$ with ε [35]. There are also the projections of four segments along with their corresponding buffers (also in ε distance from the interpolated segment). The thick portion of these segments implies the part of the segment that lies *inside* the cell with 100% probability. The dashed portion implies the part of the segment that lies *outside* the cell with 100% probability, while the solid thin portions are the parts of the segments that we do not know whether they lie inside or outside the cell. So, the ratio of the length of the thick portion over the total trajectory length corresponds to the *membership* of the segment to the cell. Similarly, the dashed and the solid thin fractions result to its *non-membership* and *hesitancy*, respectively. Technically speaking, the thick portion is the result of the intersection of $(c_{k,l} \ominus \varepsilon)$ with the segment, while the dashed portion is the topological difference of the segment with $(c_{k,l} \oplus \varepsilon)$.

Let us return to our discretized world; as we assume that, after the initial preprocessing, we handle $\bar{T}_i$, i.e., the set of $c_{k,l}$ that are definitely crossed by T_i, we can approximate the previous probabilities by counting the number of cells of $r_{i,j}$ and $ur_{i,j}$. Formally, given the membership $\mu_A(ur_{i,j})$ and non-membership $\gamma_A(ur_{i,j})$ of an uncertain region $ur_{i,j}$ to the fuzzy set A containing the trajectories that have or have not, respectively, traversed this region with 100% probability, we provide the following notion of *Intuitionistic Uncertain Trajectory*:

Definition 5 Given an uncertain trajectory $\text{UnTra}(\bar{T}_i)$, its intuitionistic counterpart, $I - \text{UnTra}(\bar{T}_i)$, is defined as a p-dimensional vector of triplets $\langle (ur_{i,1}, \mu_A(ur_{i,1}), \gamma_A(ur_{i,1})), \ldots, (ur_{i,p}, \mu_A(ur_{i,p}), \gamma_A(ur_{i,p}))\rangle$ where each triplet consists of an uncertain region $ur_{i,j}$, its membership $\mu_A(ur_{i,j})$, and its non-membership $\gamma_A(ur_{i,j}))$, with the latter two being defined as:

$$\mu_A(ur_{i,j}) = \frac{\left|r_{i,j}\right|}{\left|\text{UnTra}\left(\bar{T}_i\right)\right|}, \tag{1}$$

$$\gamma_A(ur_{i,j}) = \frac{\left(\left|\text{UnTra}\left(\bar{T}_i\right)\right| - \left|ur_{i,j}\right|\right)}{\left|\text{UnTra}\left(\bar{T}_i\right)\right|} \tag{2}$$

and $|..|$ notating the number of cells of $\text{UnTra}\left(\bar{T}_i\right)$.

Similarly, the hesitancy $\pi_A(ur_{i,j})$, namely, the degree that it is not certain whether the trajectory has passed or not from $ur_{i,j}$, is given by the following equation:

$$\pi_A(ur_{i_j}) = \frac{\left(\left|ur_{i_j}\right| - \left|r_{i_j}\right|\right)}{\left|\text{UnTra}\left(\bar{T}_i\right)\right|} \tag{3}$$

Note that it is a straightforward task to prove the intuitionistic property that $\pi_A(ur_{i,j}) = 1 - \mu_A(ur_{i,j}) - \gamma_A(ur_{i,j})$.

4 A distance metric for I-UnTra

In this section, we propose a novel distance metric modeling the dis-similarity between two *I*-UnTra instantiations. The key observation is that such a metric can be decomposed in two parts, one measuring the distance between the sequences of regions of the two trajectories (D_{UnTra}), and the other measuring the distance between intuitionistic fuzzy sets, based only on the corresponding membership and non-membership values (D_{IFS}); then, we can combine them into a single one using an aggregate function g(•), e.g., the average (or the weighted sum) of the two components. As an example, the total distance D_{total} between two *I*-UnTra A and B can be expressed as follows:

$$D_{\text{total}}(A, B) = |A - B|_{\text{IFS}}^{\text{UnTra}} = \frac{(D_{\text{UnTra}}(A, B) + D_{\text{IFS}}(A, B))}{2} \tag{4}$$

If we assume that D_{UnTra} and D_{IFS} satisfy the metric space properties, it is straightforward to prove that D_{total} as defined earlier is a metric. As such, the two steps that are required include the proposals of distance metrics for D_{UnTra} and D_{IFS} (Sects. 4.1 and 4.2, respectively).

4.1 A distance metric for sequences of regions

In order to measure the distance D_{UnTra} between two UnTra, we propose an appropriate modification of the Edit distance with Real Penalty (ERP) [9]. Among several proposals in the literature, we chose to modify ERP, given that the Euclidean distance has poor performance at the presence of noise and local time shift, while LCSS [36], DTW [40], and EDR [10] do not satisfy the metric space properties. Below we give the definition of the distance between two regions (i.e., sets of cells) that is the building element of the D_{UnTra} definition.

Definition 6 Given two uncertain regions ur_i and ur_j, their distance $|ur_i - ur_j|_d$ is defined in two different versions using two different distances d ∈ {min, ext} between their corresponding *Minimum Bounding Rectangles* (*mbr*):

$$\left|ur_i - ur_j\right|_{\text{min}} = MinDist\left(mbr\left(ur_i\right) - mbr\left(ur_j\right)\right)/MaxCellDist \tag{5}$$

and

$$\left|ur_i - ur_j\right|_{ext} = 1 - \frac{1}{2}\left(\frac{\text{ext}_x\left(mbr\left(ur_i\right)\right) + \text{ext}_x\left(mbr\left(ur_j\right)\right)}{2 \cdot \text{ext}_x\left(mbr\left(ur_i \cup ur_j\right)\right)} + \frac{\text{ext}_y\left(mbr\left(ur_i\right)\right) + \text{ext}_y\left(mbr\left(ur_j\right)\right)}{2 \cdot \text{ext}_y\left(mbr\left(ur_i \cup ur_j\right)\right)}\right), \tag{6}$$

where the former represents the minimum Euclidean distance between the MBRs of ur_i and ur_j, and the latter exploits on the extent of MBRs in the two axes; e.g. $\text{ext}_x\left(mbr(ur_i)\right)$ is the extent of the *mbr* of ur_i along the x axis.

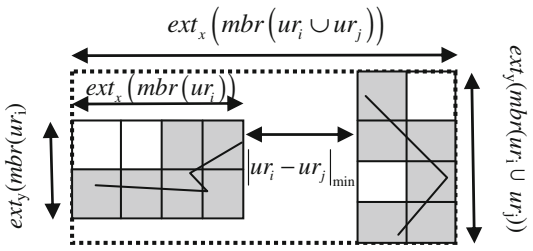

Fig. 2 The $|ur_i - ur_j|_{\text{ext}}$ and $|ur_i - ur_j|_{\min}$ distance functions

It is self-evident that $|ur_i - ur_j|_{ext}$ always results into [0,1]. Intuitively, $|ur_i - ur_j|_{ext}$ takes into account both the Euclidean distance between two regions and their extents, while it produces non-zero results in the case of overlapping regions; in the latter case, $|ur_i - ur_j|_{\min}$ yields zero (see Fig. 2). Therefore, one may choose $|ur_i - ur_j|_{ext}$ instead of $|ur_i - ur_j|_{\min}$ when refinement into the details of the ur_i, ur_j is desired. Finally, in order for $|ur_i - ur_j|_{\min}$ to be normalized in [0,1] it should be divided by the maximum possible distance of two regions, called *MaxCellDist* in Eq. (5), i.e., the distance between the two diagonal cells (i.e. the bottom left and the upper right) of the grid.

Now, the distance D_{UnTra} between two UnTra() is defined as follows:

Definition 7 Given a regular grid G of cells $c_{k,l}$, the distance D_{UnTra} between two uncertain trajectories $\text{UnTra}\left(\bar{T}_i\right)$ and $\text{UnTra}\left(\bar{T}_j\right)$, is given by:

$$D_{\text{UnTra}}\left(\text{UnTra}\left(\bar{T}_i\right), \text{UnTra}\left(\bar{T}_j\right)\right) = \min \left\{ \begin{array}{l} D_{\text{UnTra}}\left(Rst\left(\text{UnTra}\left(\bar{T}_i\right)\right), Rst\left(\text{UnTra}\left(\bar{T}_j\right)\right)\right) + \left|ur_{i,1} - ur_{j,1}\right|_d, \\ D_{\text{UnTra}}\left(Rst\left(\text{UnTra}\left(\bar{T}_i\right)\right), \text{UnTra}\left(\bar{T}_j\right)\right) + \left|ur_{i,1} - gap\right|_d, \\ D_{\text{UnTra}}\left(\text{UnTra}\left(\bar{T}_i\right), Rst\left(\text{UnTra}\left(\bar{T}_j\right)\right)\right) + \left|gap - ur_{j,1}\right|_d \end{array} \right\} \quad (7)$$

where $Rst\left(\text{UnTra}\left(\bar{T}_i\right)\right)$ denotes the remaining regions of $Rst\left(\text{UnTra}\left(\bar{T}_i\right)\right)$ after removing $ur_{i,1}$, and *gap* is the region containing the first cell of our grid (i.e., cell $c_{1,1}$).

The value of the *gap* element is given in a way similar with [9] where it is determined as the first value of the time scale for the time series (i.e., typically $gap = 0$). Note that as all UnTra have the same dimensionality p, *gap* regions may be introduced not due to difference in lengths rather than the lack of motion of an individual trajectory during this particular period. Next, we present Lemma 1, required by Theorem 1 that proves that D_{UnTra} is a metric.

Lemma 1 *For any three regions ur_q, ur_i, ur_j, any of which may be a gap region, it is always true that $\left|ur_q - ur_j\right|_d \leq \left|ur_q - ur_i\right|_d + \left|ur_i - ur_j\right|_d$.*

Proof It has been proven by [38]. □

Theorem 1 *The distance measure D_{UnTra} between UnTra $\left(\bar{T}_i\right)$ and UnTra $\left(\bar{T}_j\right)$, is a metric.*

Proof It is straightforward that isolation and symmetry properties hold for D_{UnTra}. Due to Lemma 1, the triangular inequality property also holds for D_{UnTra}. □

4.2 A distance metric for intuitionistic fuzzy sets

Given a finite universe $E = \{x_1, x_2, \ldots, x_n\}$ and an intuitionistic A = $\{\langle x, \mu_A(x), \gamma_A(x)\rangle | x \in E\}$ fuzzy set, we define three fuzzy sets $M_A = \{\mu_A(x)\}$, $\Gamma_A = \{\gamma_A(x)\}$, $\Pi_A = \{\pi_A(x)\}$,

containing the values of membership, non-membership, and hesitancy, respectively, for every $x \in A$. Under this connection, A can be also described by the triplet (M_A, Γ_A, Π_A). Exploiting the aforementioned description of a fuzzy set A, we devise a method for measuring the similarity between intuitionistic fuzzy sets, based on the membership, non-membership, and hesitancy values of their elements.

Definition 8 Considering a finite universe $E = \{x_1, x_2, \ldots, x_n\}$ and two intuitionistic fuzzy sets on it, $A = (M_A, \Gamma_A, \Pi_A)$ and $B = (M_B, \Gamma_B, \Pi_B)$, with the same cardinality n, the similarity measure Z between A and B is given by the following equation:

$$Z(A, B) = \frac{1}{3}\left(z(M_A, M_B) + z(\Gamma_A, \Gamma_B) + z(\Pi_A, \Pi_B)\right) \tag{8}$$

where $z(A', B')$ for fuzzy sets A' and B' (e.g. for M_A, M_B) is defined as:

$$z\left(A', B'\right) = \begin{cases} \frac{\sum_{i=1}^{n} \min\left(\mu_{A'}(x_i), \mu_{B'}(x_i)\right)}{\sum_{i=1}^{n} \max\left(\mu_{A'}(x_i), \mu_{B'}(x_i)\right)}, & A' \cap B' \neq \emptyset \\ 1, & A' \cap B' = \emptyset \end{cases} \tag{9}$$

and similarly for $\Gamma A, \Gamma B$ and $\Pi A, \Pi B$.

The aforesaid definitions can be demonstrated by the following simple numeric example: Assuming three intuitionistic fuzzy sets A, B, C with $A = \{x, 0.4, 0.2\}, B = \{x, 0.5, 0.3\}, C = \{x, 0.5, 0.2\}$ we want to find whether B or C is more similar to A. Using the equations of Definition 8 we compute the similarity of B and C to set A : $Z(A, B) = (0.4/0.5 + 0.2/0.3 + 0.2/0.4)/3 = 0.65$, and $Z(A, C) = (0.4/0.5 + 0.2/0.2 + 0.3/0.4)/3 = 0.85$, concluding that C is more similar to A than B.

Finally, the intuitionistic fuzzy set distance D_{IFS} between two I-UnTra$A$ and B, can be expressed as:

$$D_{\text{IFS}}(A, B) = 1 - Z(A, B) \tag{10}$$

which is proven to be a distance metric.

Lemma 2 *The intuitionistic fuzzy set distance D_{IFS} between two I-UnTraA and B is a metric.*

Proof sketch One can easily verify that isolation, symmetry, and triangular inequality properties hold for D_{IFS}. (Nevertheless, a complete proof of Lemma 2 appears in the appendix for the convenience of the reviewers.) □

The proposed intuitionistic similarity measure uses the aggregation of the minimum and maximum membership, non-membership, and hesitancy values. It is simple to calculate, sensitive to small value variations, and deals well with all the counter-intuitive cases, reported in [26], in which other measures fail. For a qualitative evaluation of the proposed distance metric in comparison with other approaches the reader is referred to Sect. 6.2.

5 A novel trajectory clustering algorithm

The majority of the proposed clustering methods so far assume that each vector belongs to one cluster only, a reasonable assumption when vectors reside in dense and well-separated clusters. However, in real-world applications where complex input data may form overlapping clusters, the degree of membership of a vector x_k to the i-th cluster u_{ik} is a value in the

interval [0, 1]. Based on this observation, [7] introduced the FCM algorithm which uses a weighted exponent on the fuzzy memberships. FCM iteratively discovers cluster centroids that minimize a criterion function measuring the quality of a fuzzy partition. A fuzzy partition is denoted by a $(c \times N)$-dimensional matrix U of reals $u_{ik} \in [0, 1]$, with $1 \leq i \leq c$ and $1 \leq k \leq N$, where c and N are the number of clusters and the cardinality of the data vectors, respectively. The following constraint is imposed upon u_{ik}:

$$\sum_{i=1}^{c} u_{ik} = 1, \quad 0 < \sum_{k=1}^{N} u_{ik} < N \tag{11}$$

Given this, the FCM objective function has the form:

$$J_m\,(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik})^m\, \mathrm{d}_{ik}^2 \tag{12}$$

where V is a $(p \times c)$-dimensional matrix storing c centroids, p is the dimensionality of the data, d_{ik} is an A-norm measuring the distance between data vector x_k and cluster centroid v_i, and $m \in [1, \infty)$ is a weighting exponent. The parameter m controls the fuzziness of the clusters. When m approximates 1, FCM performs a hard partitioning as the k-means algorithm does, while as m converges to infinity the partitioning is as fuzzy as possible. There is no analytical methodology for the optimal choice of m. By iteratively updating the cluster centroids and the membership degrees for each feature vectors, FCM iteratively moves the cluster centroids to the "correct" location within the data set.

Regarding the centroid calculation, [25] presented a first approach to solve this problem in the context of TD, providing the notion of *representative trajectory*. Assuming that movement patterns are more or less straight lines, they introduce an averaging technique between segments that works well when trajectories are dense and follow such a linear regression model. However, in this paper, we claim that real-world applications involve trajectories that often follow circular movement patterns or present large agility. Moreover, trajectories that follow similar routes for only a portion of their lifespan and then diverge would result in non-representative motions patterns that cannot be described by conventional averaging techniques. In order to overpass these obstacles and support real-world requirements, we argue that a better representation can be succeeded if we utilize local criteria (contrary to global criteria via generic distance functions) to decide whether a sub-trajectory is part of the movement pattern. For this reason, next, we provide a method that enables this calculation exploiting local trajectory matches.

5.1 The centroid trajectory algorithm

We base our proposal for the *Centroid Trajectory* (*CenTra*) estimation on the definition of *I*-UnTra. Our methodology not only overpasses the previously mentioned obstacles, but also, it may be used to represent the *thickness* of the centroid, so as to model the amount of trajectories that contribute to its formation. Towards this goal, we firstly adopt some local similarity function to identify common sub-trajectories (concurrent existence in space-time), and secondly we follow a *region growing* approach so as to represent this local cluster. The idea is to form *CenTra* similar to an UnTra, requiring at the same time to satisfy some similarity and density constraints. Formally:

Definition 9 Given a regular grid G of granularity $m \times n$ consisting of cells $c_{k,l}$ $(1 \leq k \leq m$ and $1 \leq l \leq n)$, each of which has cell density $G(k, l)$ (where *cell density* is defined

```
Algorithm CenTra(set of I-UnTra S, Grid G, Real δ, Real d, Real ΔG)
01.  CenTra=∅;
02.  forall temporal periods p_j, j in [1,P]
03.    L_CenTra_j = Init_Local_Centra(p_j);
04.    repeat
05.      forall regions ur_{i,j} in PQ that Growth(L_CenTra_j)-Growth(L_CenTra_j ∪ ur_{i,j})<ΔG
06.        AR_{i,j} = L_CenTra_j extended with ur_{i,j};
07.      AR ={ur_{i,j}|Sim(L_CenTra_j,ur_{i,j})≥d and avg_density(AR_{i,j})≥δ};
08.      if AR ≠ ∅
09.        ur_{i,j}=argmax_{reg∈AR}(avg_density(AR_reg));
10.        L_CenTra_j=AR_{i,j};
11.    until AR ≠ ∅;
12.    CenTra=CenTra ∪ L_CenTra_j;
13.  return CenTra;
```

Fig. 3 CenTra algorithm

as the number of distinct trajectories traversing the cell), a region density threshold δ, a similarity (or distance) threshold d and a set S of p-dimensional UnTra $\left(\bar{T}_i\right)$, we define the *CenTra* of S as an UnTra whose regions at each period $p_j, 1 \leq j \leq P$, correspond to a *Local CenTra* (L_CenTra_j), which is an *Augmented Region* (AR) of a seed region that has been extended "towards" other regions (i.e. sub-trajectories) if and only if (a) the similarity between $ur_{i,j}$ (under examination) regions and L_CenTra_j is $Sim(L\_Centra_j, ur_{i,j}) \geq d$, and (b) adopted regions $AR_{i,j}$ have average density $avg\_density(AR_{i,j}) \geq \delta$.

Figure 3 illustrates the developed *CenTra* algorithm used to calculate the centroid trajectories based on Definition 9. The background idea is to perform some kind of time-focused local clustering using a region growing technique under similarity and density constraints. The algorithm for each time period (line 2), determines an initial seed region, (via the *Init_Local_CenTra* (line 3)) and searches for the maximum region that is composed of all sub-trajectories that are similar over d and dense over δ. The seed region is determined as the one with the minimum average distance from the rest candidate regions, and which is also dense. Subsequently, the growing process begins (line 4) and the algorithm tries to find the next region to extend (lines 5–6) among the Most Similar Trajectories (MST) [17], as someone would expect to find the *best region* in one of these regions. Note that searching in-between the MST introduces only a small overhead in the algorithm's execution, since the corresponding results are kept in a priority queue PQ that has been fed during the initialization of the seed region (line 3). Then the algorithm searches among the candidates regions, i.e., those that satisfy the similarity and density constraints (line 7), in order to find the best, i.e., the one that maximizes the average density after growing (lines 9–10). The whole process continues until no more growing can be applied (line 11), appending in each repetition the temporally local centroid L_CenTra_j to *CenTra* (line 12).

At this point it is crucial to enlighten the part of the algorithm where it searches among the MST candidates (line 5), the best ones that will be used for growing. In [32], the approach was to search among the k-MST, where k was a user parameter. This approach implies that no growing occurs if $k = 1$ (only the seed, if there is one according to the similarity threshold d, takes place in the formation of the corresponding L_CenTra_j), while the growing is the maximum possible when k is equal to TD cardinality. Obviously, if a small k is selected growing will stop early and will possible leave some similar trajectories out of the process, while if a large k is selected then the algorithm will grow up to a point, but further unnecessary iterations (i.e. unsuccessful growing attempts) will take place. More importantly, the most intuitive growing will probably happen for different values of k in different time periods. Therefore, even a good selection of such a parameter will not behave well in all time periods.

The previous discussion implies that the selection of k is not a straightforward task and that another technique is necessary that will enable the user to control the growing process in an intuitive and efficient way. The idea at this point is to allow searching for more MST to grow, as long as the difference of the variance of the distances between the already grown L_CenTra_j and the L_CenTra_j augmented with the subsequent candidate region (in the ordering according to the average distance), is less than a predefined threshold. Note that the variance (i.e. square standard deviation σ^2) of the distances of the regions participating in the formation of each Local Centroid of a particular time period is a measure that intuitively describes the spatial expansion or differently the growth of the *L_CenTra*. Formally:

Definition 10 The *Growth* of a L_CenTra_j that corresponds to a time period p_j, $1 \leq j \leq P$, which has been formed by the growing of $|LC_j|$ regions, whose distances from the L_CenTra_j before growing by itself is $\mathrm{d}_1, \ldots, \mathrm{d}_{|LC_j|-1}$, respectively, is defined as the variance of these distances, namely:

$$Growth\left(L\_CenTra_j\right) = \frac{1}{|LC_j| - 1} \sum_{i=1}^{|LC_j|-1} \left(\mathrm{d}_i - \bar{d}\right)^2 \tag{13}$$

which is simplified with straightforward mathematical operations that allow its incremental calculation as: (e.g. see http://en.wikipedia.org/wiki/Standard_deviation)

$$Growth\left(L\_CenTra_j\right) = \frac{1}{|LC_j| - 1} \sum_{i=1}^{|LC_j|-1} (\mathrm{d}_i)^2 - \left(\bar{d}\right)^2 \tag{14}$$

where $\bar{\mathrm{d}} = \sum_{i=1}^{|LC_j|-1} \mathrm{d}_i \Big/ |LC_j| - 1$ is the average (mean) distance.

The previous definition allows us to stop searching for further growing when the difference of the *Growth* of the L_CenTra_j before and after the growing with a new region $ur_{i,j}$, i.e. $|Growth(L\_CenTra_j) - Growth(L\_CenTra_j \cup ur_{i,j})|$ is over a given threshold ΔG. This is reflected in Algorithm *CenTra* by ceasing the loop over the candidate regions as these are dequeued by *PQ* once the previous condition does not hold (line 5). Note that, by keeping the summation of the distances d_i and the square distances d_i^2, we succeed calculating the *Growth* incrementally in constant time.

Theorem 2 *The time complexity of the CenTra algorithm is* $O(|S|^2)$.

Proof In the appendix. □

5.2 The cenTR-I-FCM algorithm for I-Untra

Continuing our discussion regarding FCM, it must be mentioned that its direct employment in the context of TD would result to an inefficient scheme: during the process of transforming trajectories to data points, initial trajectories should be interpolated at all time instances every other trajectory sampled its position, something that would prohibitively increase the dimensionality of the problem. More importantly, using an A-norm as the mean to measure the distance between trajectories, it is expected to encounter all the well-known problems being present when measuring the similarity in time series data, such as the presence of outliers, different speeds, local shifts, different baselines and scales. Furthermore, FCM tries to partition the dataset simply by looking at the vector values ignoring the fact that these

vectors may be accompanied by qualitative information (i.e., the uncertainty) which may be given per dimension.

Contrary to these shortcomings, we take advantage of our intuitionistic trajectory representation *I*-UnTra, i.e., the *p*-dimensional vectors of triplets$(ur_{i,j}, \mu_A(ur_{i,j}), \gamma_A(ur_{i,j}))$. While it is evident that the FCM algorithm cannot utilize intrinsically such qualitative information, we propose a different perspective by substituting the distance function with the distance metric D_{total} introduced in Sect. 4. Using the proposed distance function, the fuzzy c-means objective function takes the form:

$$J_m^{CenTR}\text{–}I - FCM\,(U, V) = \sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik})^m\,|x_k - v_i|_{\text{IFS}}^{\text{UnTra}} \tag{15}$$

Theorem 3 *Given a* $(p \times c)$*-dimensional matrix V storing c centroids trajectories I-UnTra of dimensionality p, a distance* $|x_k - v_i|_{\text{IFS}}^{\text{UnTra}}$ *between trajectory* x_k *and cluster centroid* v_i*, a weighting exponent* $m \in [1, \infty)$*, and sets* $I_k, \tilde{I}_k$ *defined as:*

$$\forall\, 1 \le k \le N,\ \begin{cases} I_k = \left\{\, i \mid 1 \le i \le c;\ \ |x_k - v_i|_{\text{IFS}}^{\text{UnTra}} = 0\right\}, \\ \tilde{I}_k = \{\,1, 2, \ldots, c\} \setminus I_k, \end{cases}$$

then $J_m^{CenTR-I-FCM}\,(U, V)$ *may be minimized if and only if:*

$$\underset{\substack{1\le i\le c\\ i\le k\le N}}{\forall} u_{ik} = \begin{cases} \left(|x_k - v_i|_{\text{IFS}}^{\text{UnTra}}\right)^{\frac{1}{1-m}} \Big/ \sum_{j=1}^{c}\left(|x_k - v_j|_{\text{IFS}}^{\text{UnTra}}\right)^{\frac{1}{1-m}}, & I_k = \emptyset, \\ \begin{cases} 0, & i \notin I_k \\ \sum_{i\in I_k} u_{ik} = 1, & i \in I_k, \end{cases} & I_k \neq \emptyset, \end{cases} \tag{16}$$

and

$$\underset{1\le i\le c}{\forall}\ v_i = \sum_{k=1}^{N}(u_{ik})^m x_k \Big/ \sum_{k=1}^{N}(u_{ik})^m. \tag{17}$$

Proof sketch Equations (16) and (17) follow from straightforward mathematical operations. Nevertheless, a complete proof of Theorem 3 appears in the appendix □

Note that u_{ik} corresponds to the membership of the *k*-th *I*-UnTra to the *i*-th cluster and it is different from the internal intuitionistic fuzzy memberships of each *I*-UnTra. Moreover, after the centroids' computation using Eq. (17) and before the next iteration, where the memberships u_{ik} to the new clusters are updated, we calculate the memberships and non-memberships of the new (virtual) centroid trajectories. At each iteration and for every centroid, we extract the membership degree μ_{i_j} (non-membership degrees γ_{i_j}) of centroid v_i as the average of the memberships (non-memberships, respectively) of all *I*-UnTra that belong to cluster *i*. More formally, if C_i is a set defined as

$$\underset{1\le i\le c}{\forall}\quad C_i = \left\{k \,|\, 1 \le k \le N;\ |x_k - v_i|_{\text{IFS}}^{\text{UnTra}} < |x_k - v_r|_{\text{IFS}}^{\text{UnTra}},\quad \forall 1 \le r \le c \wedge r \neq i\right\}$$

we obtain that:

$$\underset{1\le \text{j}\le \text{p}}{\forall}\quad \mu_{i_j} = \frac{\sum_{\forall k\in C_i}\mu_{k_j}}{|C_i|},\quad \nu_{i_j} = \frac{\sum_{\forall k\in C_i}\gamma_{k_j}}{|C_i|} \tag{18}$$

Using the update procedure of Eq. (17) in the TD setting, we would share the same problems with FCM and k-means. Since we are especially interested in the representation of

```
Algorithm CenTR-I-FCM (set of I-UnTra S, Real ε, Int c)
01. V(0) = c random I-UnTra; j=1;
02. repeat
03.   Calculate membership matrix U(j) // use Eq.(16)
04.   Update the centroids' matrix V(j) using CenTra;
05.   Compute membership and non-membership degrees of V(j)  // use Eq.(18)
06. Until ||U(j+1)-U(j)||F≤ε; j=j+1;
```

Fig. 4 CenTR-I-FCM algorithm for clustering I-UnTra

real movement patterns, we could use the centroid trajectory derived by the density-based *CenTra* algorithm instead of this weighted averaging technique; we argue that the adoption of *CenTra* as the update centroid methodology of the product of Theorem 3, will result to more meaningful trajectory clustering. The idea is that the algorithm implied by Theorem 3 iteratively tries to diminish the intra-cluster variance using some global, approximate distance metric, and *CenTra* comes at each iteration to push (i.e., grow) the centroid (only the sub-trajectories and not the whole trajectory) towards *interesting* places, where interestingness in our case means high density and similarity. The incorporation of *CenTra* into FCM (named *Centroid TRajectory Intuitionistic FCM* (CenTR-I-FCM)) is a straightforward task and only takes place at line 4 of the algorithm in Fig. 4 with the invocation of *CenTra*.

5.3 Temporal relaxation of centroid trajectories

As delineated from the earlier discussion, except for trajectory clustering, the implicit outcome of the CenTR-I-FCM algorithm is a set of *I*-UnTra which correspond to the Centroid Trajectories for each of the resulted clusters. Such aggregated mobility patterns are very useful in several fields such as traffic engineering, zoology studying animals' migration, anthropography studying humans' behavior, and, more generally, in designing mobility-aware systems and services. Being *I*-UnTra the dimensionality of the centroid trajectories is *P*. Recall that *P* is a user parameter that allows us to represent every trajectory under the same dimensionality, which further allows us to cluster trajectories with vector-based algorithms. Also, this parameter is selected to operate on the time dimension, upon which we assume no uncertainty, in accordance with the literature. Viewing dimensionality from a different perspective, one may consider it as a segmentation of the trajectories in time axis. Although intuitive and necessary for our purposes, this segmentation is static and predefined for the whole TD. However, there is no reason for the dimensionality to remain fixed also after the end of the clustering process and for the centroid trajectories which are aggregated patterns. We argue that a more intuitive representation that certainly has added value for an analyst would be to provide these mobility patterns also aggregated along the temporal dimension. Such a representation would model each centroid trajectory with a (eventually) different number of time periods (less than *P*) of varying longer duration. Each such period is the result of merging successive periods (and the respective regions) of the initial representation, aiming to identify the maximal temporal periods during which some uniformity criterions hold. Figure 5 presents an algorithm, called *Time-relaXed CenTra* (*TX-CenTra*) which is devised to transform a centroid trajectory according to the previous discussion. The idea of the algorithm is, starting from an initial time period, to follow a greedy time-expansion technique which will concatenate a set of successive periods, as long as the density constraint is satisfied, but also the *Growth* of the candidate (for expansion) regions remains more or less the same. More specifically, after initialization (lines 1–2), the algorithm starts an iterative procedure until all time periods are used (lines 3–15). During each iteration, a local centroid *L_CenTra* is appended to the transformed *CenTra* (line 14). The new local

```
Algorithm TX-CenTra(I-UnTra centra, Grid G, Real δ, Real ΔG)
01. TXCenTra=∅ ; j=1; tx=1;
02. forall i in [j,P] used(p_i)=false;
03. repeat
04.   L_CenTra_tx = centra(p_j);
05.   repeat
06.     forall periods p_k, k in [j+1,P]
07.       TE_j,k = L_CenTra_tx expanded with centra(p_k);
08.     TE ={TE_j,k| avg_density(TE_j,k)≥δ and Growth(L_CenTra_tx)-Growth(TE_j,k)<ΔG};
09.     if TE ≠ ∅
10.       L_CenTra_tx = arg max_{TE_j,k∈TE} (avg_density(TE_j,k));
11.       forall i in [j,k] used(p_i)=true;
12.       j=k+1;
13.   until TE ≠ ∅;
14.   TXCenTra=TXCenTra ∪ L_CenTra_tx;tx=tx+1;
15. until used(p_P)==true;
16. return TXCenTra;
```

Fig. 5 TX-CenTra algorithm

centroid is initialized with the corresponding old local centroid of the first time period not used so far (line 4). Subsequently, using the previous region as seed, the temporal expansion (TE) begins (line 5) by searching the best among the candidates periods (lines 6–8), which, if concatenated to local centroid, will result in a region which will be dense over δ, while the difference in its *Growth* (i.e., the difference of the variance of the distances of the regions participating in the formation of the Local Centroids) before and after the merging will not be significant (i.e., below a threshold ΔG). The best period to stop expansion is the one that maximizes the average density after growing (lines 9–12). The whole process continues until no more expansion can be applied (line 13).

Theorem 4 *The time complexity of the TX-CenTra algorithm is* $O(P^2)$.

Proof In the appendix. □

6 Experimental evaluation

In this section, we present an experimental study in order to evaluate our approach. The experiments were run on a PC with Intel Core Duo at 2.53 GHz, 4 GB RAM and 240 GB hard disk. We implemented the proposed algorithms using C++.

6.1 Datasets

In order to evaluate the accuracy of our clustering algorithms, we have used synthetic datasets generated by a custom generator based on the GSTD data generator [34]. Specifically, this generator produces trajectory datasets following complex mobility patterns based on a given distribution of *spatio-temporal focal points*, to be visited by each trajectory in a specific order. Each generated dataset then forms a natural cluster, since all trajectories follow more or less the same mobility pattern.

The general idea behind this generator is to use the focal points so as to attract each trajectory's movement; when a particular trajectory has reached the area around a focal point, having at the same time completed the respective temporal predicate, the generation algorithm changes the attracting point to the next focal point in the list, and so on, until no focal points are left unvisited. For example, Fig. 6a illustrates the 2D projection of a dataset generated with the above generator, using points 1 to 5 as focal points. It is clear that all

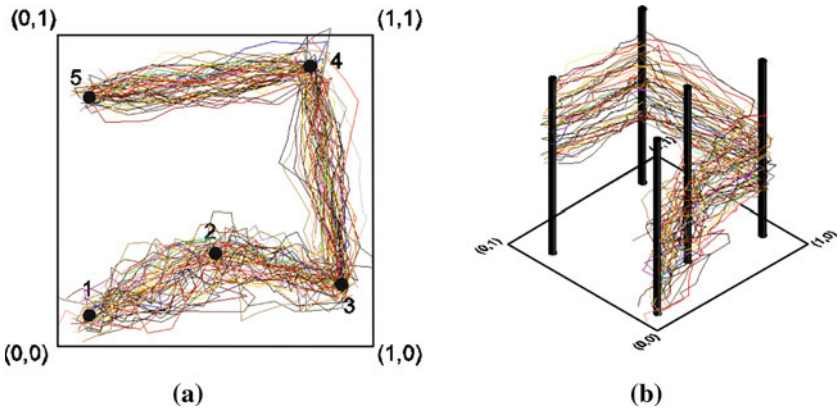

Fig. 6 Synthetic clusters of trajectories in (**a**) 2D space, (**b**) 3D space

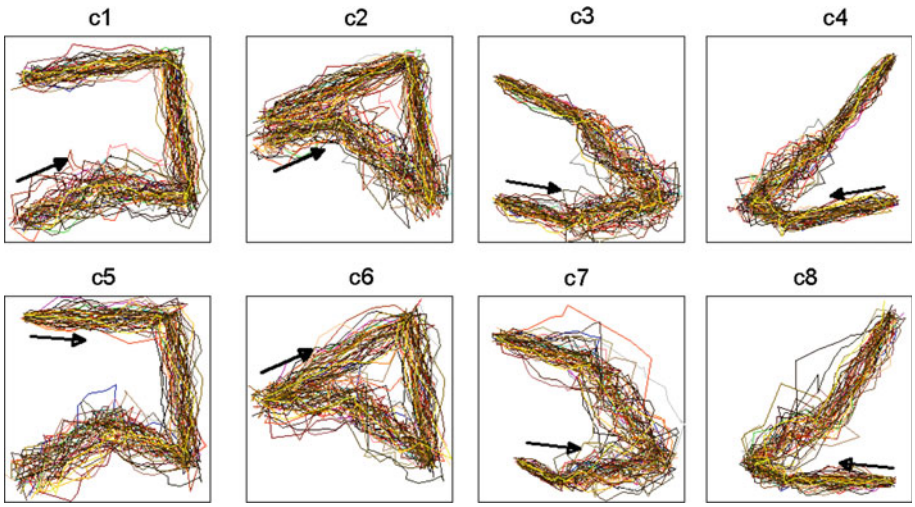

Fig. 7 The eight trajectory clusters

trajectories follow the same trend, visiting the areas around points 1, 2, . . ., 5, always with the same order. The generator has also the ability to control the lifespan of each trajectory, thus allowing us to distribute the trajectories along the temporal axis. For example, Fig. 6b illustrates the previous dataset in 3D space, with the lifespan of each trajectory set to 0.5. As a consequence, the movement of each point begins in a random time instance between 0 and 0.5. This allows us to simulate realistic movements of objects, as objects moving on the same spatial regions rarely start their movement at the same time instance. Regarding, the rest properties of the generator, one may also adjust the velocity of each moving point (which may follow either random or normal distribution), as well as the temporal periods between sampled points (i.e., temporal gap between two sampled positions).

Using the generator, we produced eight synthetic clusters of trajectories illustrated in Fig. 7. Each cluster contains 50 trajectories, covering the spatial data space, with their lifetime set to 0.50 of the temporal space. The datasets used in our experiments were produced by mixing the clusters illustrated in Fig. 7, as described in each respective experiment that follows.

Note that by tuning the area around the focal points that a trajectory must pass, we produce more-or-less linear sub-patterns of varying extend (i.e., *Growth*). As such, generated clusters may have several noisy or compact parts (e.g., the part of the trajectories in Fig. 6a between points 1 and 2, vs. those between 4 and 5). Furthermore, these sub-patterns are intentionally mixed up in several parts of their lifetime (e.g., the part between points 3 and 4 in Fig. 6a is common for both clusters c1 and c2 (Fig. 7), while they show only minor differentiations in other parts.

To the best of our knowledge, in the TD domain there is no available real dataset already clustered by a domain expert in order to be used as ground truth for benchmarking. Nevertheless, in this paper, we have used a real dataset in order to evaluate the efficiency of our approach as well as the effectiveness of our algorithms in discovering complex mobility patterns. The initial dataset consists of the GPS-tracked positions of 50 trucks transporting concrete in the area of Athens between August and September 2002 (the dataset is publicly available at http://www.rtreeportal.org). The dataset contains 1,12,300 position records consisting of the truck identifiers, dates and times, and geographical coordinates. The temporal spacing is regular and equals 30 s. From these raw data, we produced 1,100 trajectories by splitting the recordings of a truck in subsets if there was a temporal gap between two consecutive recordings larger than 15 min (a gap that indicates a stop not due to traffic or traffic lights). Subsequently, we used the CommonGIS visual analytics tool [3] to manually identify real clusters.

More specifically, we manually discovered two clusters of trajectories where the start and end locations almost coincide, i.e. each truck returned to its original location after performing a round trip; the clusters are illustrated in Fig. 8a and b, while the directions of their trips differ (we call these two clusters *"round trips"*). This set of clusters is a typical dataset that, as will be shown in Sect. 6.6, existing algorithms discovering the representative trajectories [25] of the clusters cannot operate in such cases.

6.2 Qualitative evaluation of Z similarity metric

Before we experiment with the clustering accuracy of our algorithms and their ability to extract useful mobility patterns, in this section we present a qualitative comparison of our intuitionistic similarity metric in comparison with other approaches proposed in the literature. The majority of the intuitionistic similarity measures reviewed in Sect. 2 fail

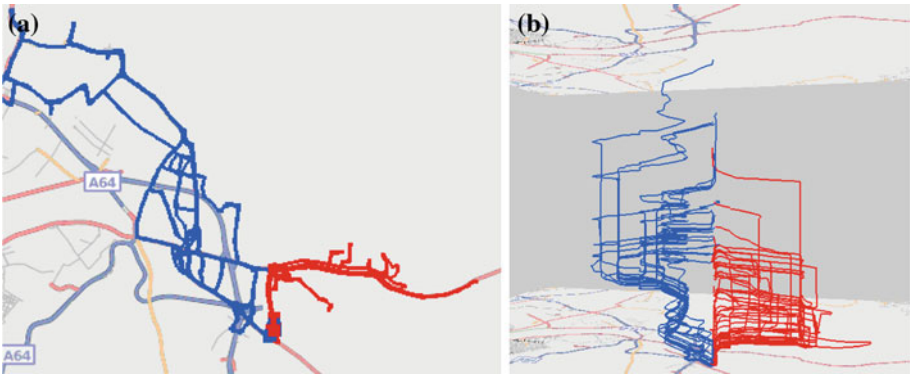

Fig. 8 Real clusters: The two "Round trips" clusters in (**a**) 2D space, (**b**) 3D space

Table 2 Qualitative evaluation between proposed and other similarity measures with counter-intuitive cases

No	Measure	Counter-intuitive cases	Measure values	Proposed measure value
I.	S_C, S_{DC}	$A=\{(x, 0, 0)\}$, $B=\{(x, 0.5, 0.5)\}$,	$S_C(A, B)=S_{DC}(A, B)=1$	$Z(A, B)=0$
II.	S_H, S_{HB}, S_e^p	$A=\{(x, 0.3, 0.3)\}$, $B=\{(x, 0.4, 0.4)\}$,	$S_H(A, B)=S_{HB}(A, B)=S_e^p(A, B)=0.9$	$Z(A, B)=0.66$
		$C=\{(x, 0.3, 0.4)\}$, $D=\{(x, 0.4, 0.3)\}$	$S_H(C, D)=S_{HB}(C, D)=S_e^p(C, D)=0.9$	$Z(C, D)=0.83$
III.	S_H, S_{HB}, S_e^p	$A=\{(x, 1, 0)\}$, $B=\{(x, 0, 0)\}$,	$S_H(A, B)=S_{HB}(A, B)=S_e^p(A, B)=0.5$	$Z(A, B)=$
		$C=\{(x, 0.5, 0.5)\}$	$S_H(B, C)=S_{HB}(B, C)=S_e^p(B, C)=0.5$	$Z(B, C)=0$
IV.	S_L and S_S^p	$A=\{(x, 0.4, 0.2)\}$, $B=\{(x, 0.5, 0.3)\}$,	$S_L(A, B)=S_S^p(A, B)=0.95$	$Z(A, B)=0.65$
		$C=\{(x, 0.5, 0.2)\}$	$S_L(A, C)=S_S^p(A, C)=0.95$	$Z(A, C)=0.85$
V.	$S_{HY}^1, S_{HY}^2, S_{HY}^3$	$A=\{(x, 1, 0)\}$, $B=\{(x, 0, 0)\}$	$S_{HY}^1(A, B)=S_{HY}^2(A, B)=S_{HY}^3(A, B)=0$	$Z(A, B)=0$
			$S_{HY}^1(A, B)=S_{HY}^1(C, D)=0.9$	
VI.	$S_{HY}^1, S_{HY}^2, S_{HY}^3$	Same as II	$S_{HY}^2(A, B)=S_{HY}^2(C, D)=0.85$	Same as II
			$S_{HY}^3(A, B)=S_{HY}^3(C, D)=0.82$	
			$S_{HY}^1(A, B)=S_{HY}^1(A, C)=0.9$	
VII.	$S_{HY}^1, S_{HY}^2, S_{HY}^3$	Same as IV	$S_{HY}^2(A, B)=S_{HY}^2(A, C)=0.85$	Same as IV
			$S_{HY}^3(A, B)=S_{HY}^3(A, C)=0.82$	

to result to a valid intuitionistic value for specific cases; some of them result to either 0 or 1, suggesting that the compared sets are either totally irrelevant or identical, while it is obvious that this is false, while others result to a high similarity value for obviously different sets. Table 2 presents all the counter-intuitive cases defined in [26], along with the measure calculation for those cases.

In case (I) of Table 2 $S_C(A, B)$ and $S_{DC}(A, B)$ imply that A and B are totally similar. The proposed measure in the contrary, suggests that A and B are totally different. In cases (II) and (IV) other measures result in a rather big similarity value while our measure is not that optimistic. In case (II), $Z(A, B) = 0.66$ as the hesitancy in set A is a rather large quantity, while $Z(C, D) = 0.83$ as hesitancy is constant at 0.3. Moreover, in case (IV) it is obvious that set A is more similar to C than to B (A and C have the same non-membership value), something that other measures do not take into account. In (III), while A, B and C are totally different, all other measures give a similarity value of 0.5. In (VI) they do not recognize that C is more similar to D than A is to B, due to the same hesitancy value of C and D, and in (VII) they do not recognize that A is more similar to C than to B, due to the same non-membership value of A and C.

Table 2 indicates the intuitiveness of the proposed measure; it does not fail in cases where other measures do. Furthermore, it is easy to be calculated without requiring exponents or other time-consuming functions computations.

6.3 Metrics

The quality of a clustering algorithm can be evaluated using several validity indices proposed in the literature. In particular, in our experimentation we have adopted the clustering validity indices used in [30], namely the average *purity* of the clusters (i.e., the percentage of objects in the cluster that were assigned to their real cluster) and the average *coverage* (i.e., the percentage of objects of the real cluster that appear in the cluster found). Based on these indices, we also compute the average *accuracy* and the average *F-measure* of the clustering. More specifically, given the apriori knowledge of the correct clustering and the clustering obtained as the output of an algorithm, we encounter the following four measures for each cluster c_i:

- *true positives* (tp_i), i.e., the number of trajectories that belong to c_i and they were correctly assigned to c_i;
- *false positives* (fp_i), i.e., the number of trajectories that do not belong to c_i but they were incorrectly assigned to c_i;
- *false negatives* (fn_i), i.e., the number of trajectories that belong to c_i but they were incorrectly assigned to a cluster different from c_i;
- *true negatives* (tn_i), i.e., the number of trajectories that do not belong to c_i and they were correctly assigned to a cluster different from c_i.

Given the above four values, the *Purity* P_i, the $CoverageC_i$, the $AccuracyA_i$, and the weighted harmonic mean of P_i and C_i, i.e. the $F\text{-}measureF_i$ of a cluster c_i are defined, respectively, according to the following formulae:

$$P_i = \frac{tp_i}{tp_i + fp_i} \tag{19}$$

$$C_i = \frac{tp_i}{tp_i + fn_i} \tag{20}$$

$$A_i = \frac{tp_i + tn_i}{tp_i + tn_i + fp_i + fn_i} \tag{21}$$

$$F_i = 2 \cdot \frac{P_i \cdot C_i}{P_i + C_i} \tag{22}$$

The averages of these measures from all clusters define the corresponding *Purity P*, *Coverage C*, *Accuracy A*, and *F-measure F* of a clustering produced by a clustering algorithm. In general, the larger these values are, the better the clustering is.

6.4 Clustering accuracy experiments

Comparing our approach with other clustering algorithms is not a straightforward task since, as we have discussed in the related work section, existing trajectory clustering algorithms, e.g. [30,25], do not operate on approximate trajectories. The single class of algorithms that is directly applicable to uncertain trajectories is that of hierarchical algorithms by using an appropriate distance function. To this effect, we have compared our approach with three well-known variations (namely, single-link, average-link, and complete-link) of the hierarchical agglomerative algorithm, using the distance function proposed in [2]. The results of the complete linkage hierarchical algorithm (called, Com-L-Hier in the charts) turned out to be slightly better than those of the other two variations; therefore, we report only these results.

We also implemented a variation of the classic FCM algorithm appropriately modified for trajectories, called TR-FCM. In order to provide an as fair as possible comparison,

TR-FCM uses our point vector representation of trajectories, along with the previous discussed distance function [2] between MBRs in order to calculate the distance between the cluster's centroid and each candidate trajectory.

Furthermore, each experiment performed with a given set of parameters for TR-FCM and CenTR-I-FCM, was repeated 3 times, each time with different initialization. In this way, we study the effect of the random initializations, wherefrom FCM-based techniques suffer, and as such results are more statistically confident.

The quality of the algorithms is measured in terms of validity index A (the experiments measuring P, C, and F are not reported in the paper because they presented identical behavior with the one illustrated here).

In the first set of experiments, we employed all six pairs of the first 4 synthetic clusters (i.e., clusters 1 and 2, clusters 1 and 3, and so on). We mixed-up each pair and then used the CenTR-I-FCM, TR-FCM and Com-L-Hier algorithms varying grid's *cell size*, dimensionality p, density threshold δ, and uncertainty ε, in order to measure the effectiveness of each algorithm in terms of the metrics defined in Sect. 6.3. The *cell size* is provided as percentage of the size of the total space, taking values in the range {0.8, 1, 1,33, 2, 4, 6.67%}; dimensionality p ranges in {5, 15, 30, 50}; density threshold δ is set to {2, 4, 6%} of the total number of trajectories; uncertainty ε ranges in {0, 1, 2, 3}. As such, for CenTR-I-FCM we run 6*4*3*4 = 288 experiments per cluster pair and initialization, i.e. in total 288*6*3 = 5184 experiments. In all cases, we have fixed parameters $d = 0.5$ and $\Delta G = 1$. Obviously, the experiments on TR-FCM and Com-L-Hier do not take the density threshold δ into account. In Fig. 9, consisting of four plots (one for each parameter under experimentation), we present the results of the three algorithms. The reported results in a given plot are the average measures of all experiments when fixing one of the parameters (i.e. the one corresponding to the figure). For instance, Fig. 9a depicts the average accuracy of the algorithms from all experiments by fixing the grid's cell size with the six different given values. Each curve of each plot also includes the standard deviation from all experiments.

Clearly, Fig. 9 demonstrates that the proposed CenTR-I-FCM algorithm achieves very good quality, regardless of the experimental setting, with an average *Accuracy* around 84%, in contrast to 67% of TR-FCM and 54% of Com-L-Hier. Moreover, even the largest achieved rates from TR-FCM and Com-L-Hier are significantly lower than the average rate of CenTR-I-FCM. We further note that the deviation in the accuracy for CenTR-I-FCM and TR-FCM algorithms is bigger that the one of Com-L-Hier, the explanation of which is hidden on the effect of the random initialization of these approaches.

In order to study the behavior of the algorithms in the presence of more than two clusters, we performed a second set of experiments with similar parameters' setting using the synthetic clusters. More specifically, we incrementally added clusters c3, c4, c5, c6, c7, and c8 to the dataset containing clusters c1 and c2, as such producing 7 datasets with 2, 3, 4, 5, 6, 7, and 8 clusters, respectively. The results of this experiment are illustrated in Fig. 10, where for each curve we also include the standard deviation from all relevant experiments. Once again, the quality of the CenTR-I-FCM result is clearly higher than that of its competitors when scaling the number of clusters. Actually, for number of clusters higher than 3, the lower rate of CenTR-I-FCM is bigger than the higher rate of its competitors in most of the cases. On the other hand, it is important to note that the quality decreases (for all three algorithms) as the number of requested clusters increases. This behavior is normal considering the increasing complexity of the clustering problem setting, which in each step adds highly overlapping and complex clusters, making the separation of the clusters a really hard benchmark. Taking into consideration that we only use approximate trajectories, the problem gets even more difficult.

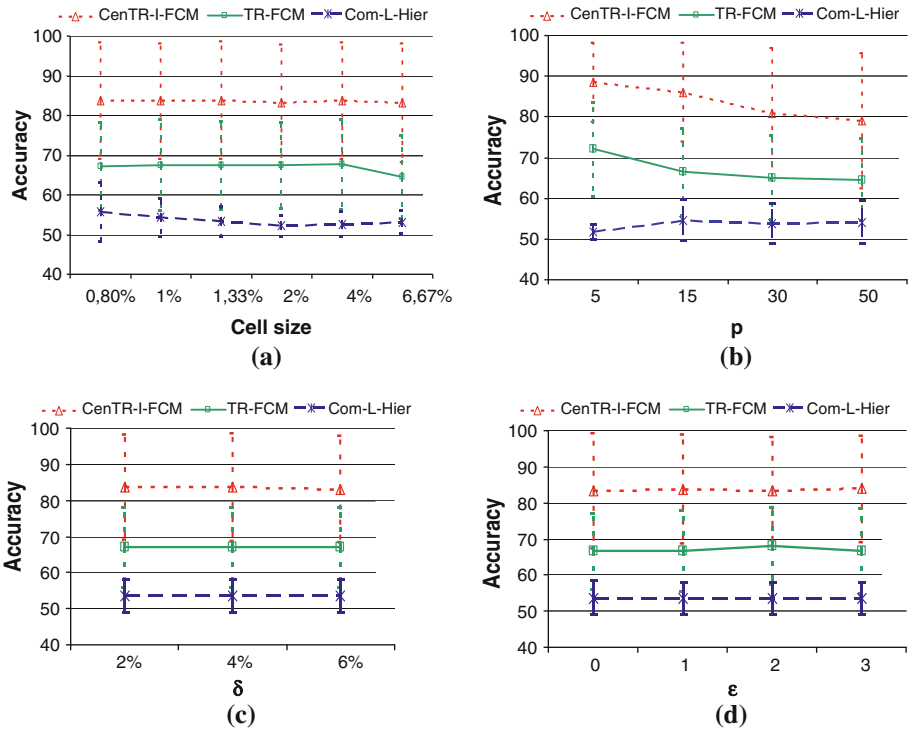

Fig. 9 Comparing clustering quality, measured in accuracy, scaling (**a**) *cell size*, (**b**) dimensionality p, (**c**) density threshold δ, (**d**) uncertainty ε

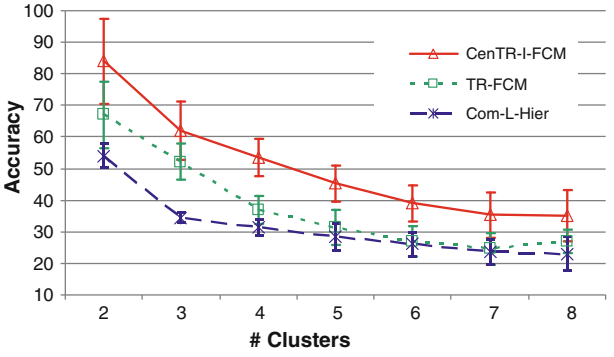

Fig. 10 Comparing clustering quality, measured in accuracy, scaling the number of clusters

6.5 Performance evaluation

Regarding the performance of the proposed CenTR-I-FCM algorithm, it was evaluated using the whole "trucks" dataset by increasing the trajectory cardinality; we also set cell size = 4% of the data space and $\varepsilon = 2$. The results of the respective experiments are illustrated in Fig. 11, which demonstrates the efficiency of CenTR-I-FCM for various numbers of clusters requested. It is clear that the execution time of the algorithm is not affected by the number

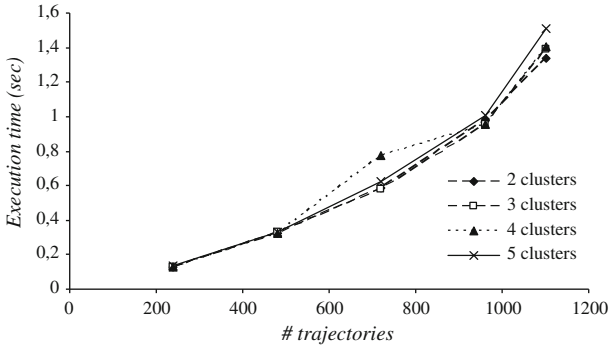

Fig. 11 CenTR-I-FCM performance, scaling the dataset cardinality

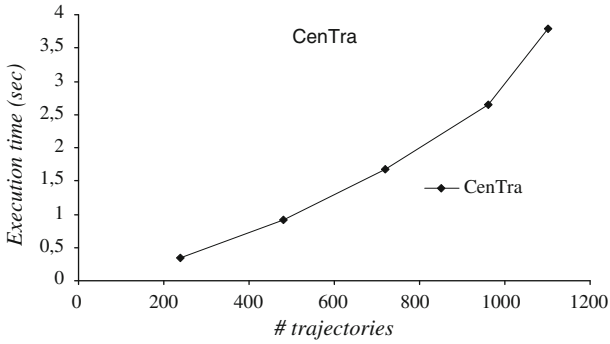

Fig. 12 CenTra performance, scaling the dataset cardinality

of clusters requested; it is also clear from that the algorithm is super-linear with the dataset cardinality. Similar results are exposed when setting the rest of the parameters to a variety of other values, e.g. setting $\varepsilon = 1$ or cell size = 1% does not affect the trends exposed by Fig. 11.

Regarding the performance of the CenTra algorithm, we used the real trajectory dataset so as to measure its execution time scaling the number of trajectories used. The corresponding results are illustrated in Fig. 12 which demonstrates that CenTra also presents super-linear behavior with the dataset cardinality. Actually this is an expected result given the outcome of our analysis (see Theorem 2).

Finally, the performance of TX-CenTra is evaluated on the Centroid Trajectory of the synthetic cluster C1 by calculating the processing time of 1,000 experiments with various cell sizes (those used in the previous experiments) and different *Growth* values ($\Delta G = 1, 10, 20, 30, 40, 50$), also scaling dimensionality p. The main conclusion is that TX-CenTra turns out to be extremely fast, since only a few milliseconds are required for each run (Fig. 13).

6.6 Evaluation of cenTra and TX-cenTra algorithms

To complete our experimental study, we evaluate CenTra and TX-CenTra with respect to the quality of their output. Although starting from different base lines and focusing on different applications, we compare their output with the representative trajectory produced by

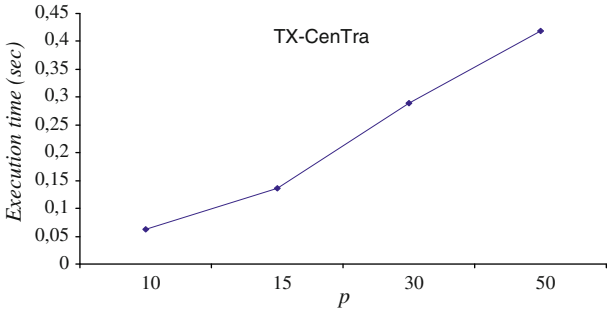

Fig. 13 TX-CenTra performance, scaling dimensionality p

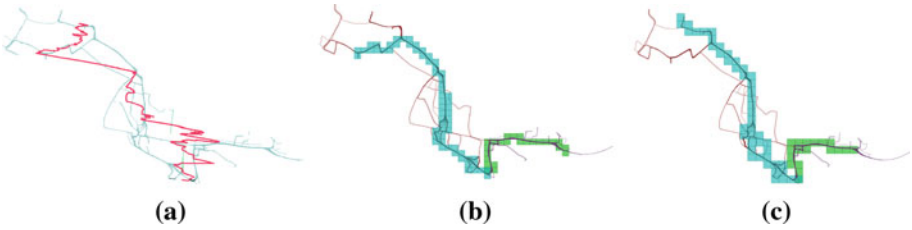

Fig. 14 Visually comparing (**a**) Representative Trajectories (produced by TRACLUS) with Centroid Trajectories (produced by CenTra) using (**b**) $cell\ size = 1.3\%$, $\varepsilon = 0$ and $\delta = 0.09$, (**c**) $cell\ size = 2.8\%$, $\varepsilon = 0$ and $\delta = 0.02$, in "round trips" dataset

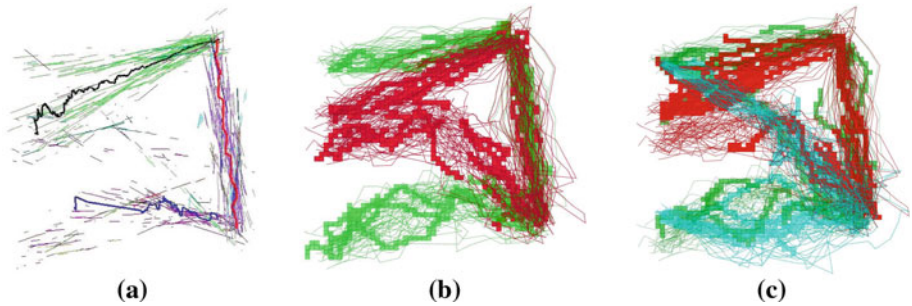

Fig. 15 Visually comparing (**a**) Representative Trajectories (produced by TRACLUS) with Centroid Trajectories (produced by CenTra) over (**b**) synthetic clusters C1 and C2 with $p = 10$, and (**c**) synthetic clusters C1, C2 and C3 with $p = 15$; all with $cell\ size = 1.67\%$, $\Delta G = 1$ and $\delta = 0.02$

the state-of-the-art TRACLUS algorithm [25]. The result of the comparison regarding the "round trips" clusters is illustrated in Fig. 14. In particular, Fig. 14a illustrates the outcome of TRACLUS. Evidently, the cluster representative (red line) does not fit the real movement, mainly due to its averaging technique. Recall, at this point, that TRACLUS clusters segments rather than whole trajectories (even considering this, the algorithm does not compass the turn occurring at the bottom of the figure). On the other hand, Fig. 14b, c illustrate CenTra, produced with variable *cell size*, ε and density δ. It turns out that CenTra not only resides on the data traces, but also vanishes the non-interesting movement details (the 'noisy' infrequent parts are not part of the centroid), catches turns, and becomes thicker in portions where something interesting (i.e. dense-similar sub-trajectories) happens (Fig. 14).

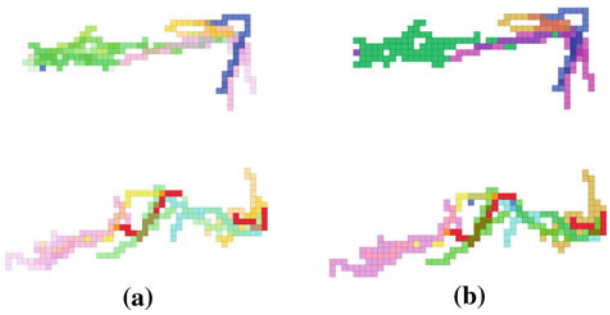

Fig. 16 Visually comparing (**a**) Centroid Trajectories (produced by CenTra) over synthetic cluster C1, with $p = 20$, $cell\ size = 2\%$, $\delta = 0.02$ and (**b**) Time-RelaXed Centroid Trajectory (produced by TX-CenTra), with $\Delta G = 60 (p = 13)$

The same conclusions can be drawn also by synthetic clusters. Figure 15a, b presents the TRACLUS's representatives (i.e., the three thick-colored lines) and the CenTR-I-FCM's centroid trajectories (i.e., the two differently colored sets of cells), respectively. It is evident that TRACLUS discovers three linear sub-clusters, which are compositions of segments (sub-trajectories) with their corresponding whole trajectories belonging to different clusters. On the other hand, CenTR-I-FCM catches the overall complex mobility patterns. Note that the black representative in Fig. 15a is outside movements' space, as this cluster (i.e., green segments) consists of the parts of both synthetic clusters which slightly deviate. Figure 15c illustrates the centroid trajectories when joining synthetic cluster 3 to the previous setting.

Finally, in order to demonstrate TX-CenTra we apply CenTra to the first synthetic cluster and then pass the resulted centroid trajectory (see Fig. 16a) as input to TX-CenTra. Depending on the *Growth* of each of the p local centroids, the outcome of TX-CenTra is a centroid trajectory covering exactly the same regions (cells), with some of these regions (corresponding to different time periods) being merged (see Fig. 16b). Different colors in Fig. 16a, b correspond to different time periods (regions). Merged regions in Fig. 16b are depicted by more solid, darker colors than the corresponding colors of the unified regions in Fig. 16a, which are colored in slight lighter and different variations of the same color. Same color of a region in both figures implies no merging for this region. The visual effect is that 20 different regions in Fig. 16a are transformed to 13 more compact regions in Fig. 16b. Note also that in this case the centroid trajectory presents a gap in its development.

7 Conclusion and future work

In this paper, we proposed a three-step approach for clustering trajectories of moving objects, motivated by the observation that clustering and representation issues in TD are inherently subject to uncertainty. Based on our novel intuitionistic fuzzy vector representation of trajectories, we defined a distance metric consisting of two components, a metric for sequences of regions D_{UnTra}, and a metric for intuitionistic fuzzy sets D_{IFS}, respectively, all used to devise the so-called CenTR-I-FCM algorithm for clustering trajectories under uncertainty. Notably, the proposed algorithm includes a novel technique for discovering the centroid of a bundle of trajectories (called, CenTra). The representation of such mobility patterns is further improved by devising an algorithm (called TX-CenTra) that relaxes their time-based point

modeling. The effectiveness and efficiency of our approach has been experimentally shown on synthetic and real trajectory datasets.

Clear future work objectives arise from our proposal: we plan to adopt some clever sampling technique for multi-dimensional data so as to diminish the effect of initialization in our algorithms, while a second direction includes the development of an index-based version for efficiency purposes and the performance of an extensive experimental evaluation using large trajectory datasets.

Acknowledgments This paper substantially improves and extends [32]. Research partially supported by the FP7 ICT/FET Project MODAP (Mobility, Data Mining, and Privacy) funded by the European Union. URL: www.modap.org. Elias Frentzos is supported by the Greek State Scholarships Foundation.

8 Appendix

Proof of Lemma 2 In order for D_{IFS} to be a metric, Z must me a metric. By $A' \subseteq B' \subseteq C'$ we have that $A'(x_i) \leq B'(x_i) \leq C'(x_i)$and

$$z(A', C') = \frac{\sum_{i=1}^{n} \min(A'(x_i), C'(x_i))}{\sum_{i=1}^{n} \max(A'(x_i), C'(x_i))} = \frac{\sum_{i=1}^{n} A'(x_i)}{\sum_{i=1}^{n} C'(x_i)},$$
$$z(A', B') = \frac{\sum_{i=1}^{n} \min(A'(x_i), B'(x_i))}{\sum_{i=1}^{n} \max(A'(x_i), B'(x_i))} = \frac{\sum_{i=1}^{n} A'(x_i)}{\sum_{i=1}^{n} B'(x_i)},$$
$$z(B', C') = \frac{\sum_{i=1}^{n} \min(B'(x_i), C'(x_i))}{\sum_{i=1}^{n} \max(B'(x_i), C'(x_i))} = \frac{\sum_{i=1}^{n} B'(x_i)}{\sum_{i=1}^{n} C'(x_i)}.$$

Thus, $\frac{\sum_{i=1}^{n} A'(x_i)}{\sum_{i=1}^{n} C'(x_i)} \leq \frac{\sum_{i=1}^{n} A'(x_i)}{\sum_{i=1}^{n} B'(x_i)}$, $\frac{\sum_{i=1}^{n} A'(x_i)}{\sum_{i=1}^{n} C'(x_i)} \leq \frac{\sum_{i=1}^{n} B'(x_i)}{\sum_{i=1}^{n} C'(x_i)}$ hence, $z(A', C') \leq z(A', B')$ and $z(A', C') \leq z(B', C')$. Isolation and symmetry properties of z are also straightforward. As such, for $A, B, C \in IFSs(E)$ and $A \subseteq B \subseteq C$ we have $\mu_A(x) \leq \mu_B(x) \leq \mu_C(x)$ and $\gamma_A(x) \geq \gamma_B(x) \geq \gamma_C(x) \quad \forall x_i \in E, i = 1, 2, \ldots, n$ therefore, $z(M_A, M_B)$, $z(\Gamma_A, \Gamma_B)$and $z(\Pi_A, \Pi_B)$ satisfy all metric properties, also resulting that Z satisfies these properties. Thus, Z is a metric □

Proof of Theorem 2 The region growing process is repeated P times (line 2). Each time the complexity of the initialization step (line 3) is $O(|S|^2 + \log |S|)$. We first compute distances in $O(|S|^2)$. While doing this we also maintain the average distances and afterwards we create the priority queue PQ of the MST in $O(\log|S|)$. Additionally, the initial filtering of the candidate regions to grow (lines 5–6) is performed in at most $|S|$ iterations (i.e. $|LC_j| = |S|$, all regions will not result in an excessive and sudden growing, namely all are close to each other), each of which has constant cost (recall that the *Growth* test is calculated incrementally). The aggregated calculations of the secondary filtering (line 7) and the actual growing (lines 8–10) can also be calculated incrementally for all iterations of the growing process, which in the worst case will take place $|S|$ times (line 4). So the cost of the growing (lines 4–11) is at most $O(|S|^2)$. Summing up the various costs the overall complexity is $O(P(|S|^2 + \log|S| + |S|^2))$. As $P << |S|$ we can safely argue that CenTra runs in $O(|S|^2)$. □

Proof of Theorem 3 The minimization of Eq. (15) can be achieved term by term:

$$J_m^{CenTR-I-FCM}(U, V) = \sum_{k=1}^{N} \varphi_k(U) \tag{23}$$

where

$$\underset{1\le k\le N}{\forall} \quad \varphi_k(U) = \sum_{i=1}^{c} (u_{ik})^m \, |x_k - v_i|_{\text{IFS}}^{\text{UnTra}} \tag{24}$$

The Lagrangian of (24) with constraints from (11) is:

$$\underset{1\le k\le N}{\forall} \quad \Phi_k(U,\lambda) = \sum_{i=1}^{c} (u_{ik})^m \, |x_k - v_i|_{\text{IFS}}^{\text{UnTra}} - \lambda\left(\sum_{i=1}^{c} u_{ik} - 1\right) \tag{25}$$

where λ is the Lagrange multiplier. Setting the partial derivatives of $\Phi_k(U,\lambda)$ to zero we obtain:

$$\underset{1\le k\le N}{\forall} \quad \frac{\partial \Phi_k(U,\lambda)}{\partial \lambda} = \sum_{i=1}^{c} u_{ik} - 1 = 0 \tag{26}$$

and

$$\underset{\substack{1\le z\le c\\ 1\le k\le N}}{\forall} \quad \frac{\partial \Phi_k(U,\lambda)}{\partial u_{zk}} = m\,(u_{zk})^{m-1} \, |x_k - v_i|_{\text{IFS}}^{\text{UnTra}} - \lambda = 0 \tag{27}$$

Solving (27) for u_{zk} we get:

$$u_{zk} = \left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}} \left(|x_k - v_z|_{\text{IFS}}^{\text{UnTra}}\right)^{\frac{1}{1-m}} \tag{28}$$

From (26) and (28) we obtain:

$$\left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}} = \frac{1}{\sum_{j=1}^{c} \left(|x_k - v_j|_{\text{IFS}}^{\text{UnTra}}\right)^{\frac{1}{1-m}}} \tag{29}$$

The combination of (28) and (29) yields:

$$\underset{\substack{1\le z\le c\\ 1\le k\le N}}{\forall} \quad u_{zk} = \frac{\left(|x_k - v_z|_{\text{IFS}}^{\text{UnTra}}\right)^{\frac{1}{1-m}}}{\sum_{j=1}^{c} \left(|x_k - v_j|_{\text{IFS}}^{\text{UnTra}}\right)^{\frac{1}{1-m}}} \tag{30}$$

Equation (17) follows with the same way as in [7]. □

Proof of Theorem 4 There are two loops that prescribe the complexity of the algorithm (lines 3 and 5). The loop (line 3) that revokes the time expansion process (line 5) may be repeated at most P times. This will happen if no actual expansion occurs. This means that in this case the internal loop (line 5) will be revoked only once for each period (i.e. TE=Ø) and as such its cost is only the cost of the loop through all (the remaining) time periods (lines 6–7), which is P; plus the cost of the filtering (line 8), which is also P, as the aggregated calculations are computed incrementally. Consequently, in this case the complexity of the algorithm is $O(P(P+P))$, namely $O(P^2)$. In the other extreme case, the loop (line 3) that revokes the time expansion process (line 5) is revoked only once, implying that all periods are merged in one. In this case, the cost of the algorithm is dominated by the internal loop (line 5), which in the worst case will be revoked P times. Again the cost of each iteration is $O(P+P)$ as delineated in the previous paragraph, even if TE=Ø in this case, as the actual expansion (line 10) can also be calculated incrementally for all iterations of the time expansion process. So, the overall complexity is again $O(P^2)$. □

References

1. Abul O, Bonchi F, Nanni M (2008) Never walk alone: uncertainty for anonymity in moving objects databases. In: Proceedings of ICDE
2. Anagnostopoulos A, Vlachos M, Hadjieleftheriou M, Keogh E, Yu PS (2006) Global distance-based segmentation of trajectories. In: Proceedings of KDD
3. Andrienko G, Andrienko N, Wrobel S (2007) Visual analytics tools for analysis of movement data. ACM SIGKDD Explor 9(2):38–46
4. Ankerst M, Breunig MM, Kriegel H-P, Sander J (1999) OPTICS: ordering points to identify the clustering structure. In: Proceedings of SIGMOD
5. Assent I, Krieger R, Glavic B, Seidl T (2008) Clustering multidimensional sequences in spatial and temporal databases. Knowl Inf Sys 16(1):29–51
6. Atanassov KT (1999) Intuitionistic fuzzy sets: theory and applications. Studies in fuzziness and soft computing, p 35
7. Bezdek JC, Ehrlich R, Full W (1984) FCM: the fuzzy *c*-means clustering algorithm. Comput Geosci 10(2-3):191–203
8. Cadez IV, Gaffney S, Smyth P (2000) A general probabilistic framework for clustering individuals and objects. In: Proceedings of SIGKDD
9. Chen L, Ng R (2004) On the marriage of edit distance and Lp norms. In: Proceedings of VLDB
10. Chen L, Tamer Özsu M, Oria V (2005) Robust and fast similarity search for moving object trajectories. In: Proceedings of SIGMOD
11. Chen SM (1995) Measures of similarity between vague sets. Fuzzy Sets Sys 74(2):217–223
12. Chen SM (1997) Similarity measures between vague sets and between elements. IEEE TSMC 27(1):153–158
13. Dengfeng L, Chuntian C (2002) New similarity measure of intuitionistic fuzzy sets and application to pattern recognitions. Pattern Recogn Lett 23(1–3):221–225
14. Denton AM, Besemann CA, Dorr DH (2009) Pattern-based time-series subsequence clustering using radial distribution functions. Knowl Inf Sys 18(1):1–27
15. Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of KDD
16. Fan L, Zhangyan X (2001) Similarity measures between vague sets. J Softw 12(6):922–927
17. Frentzos E, Gratsias K, Theodoridis Y (2007) Index-based most similar trajectory search. In: Proceedings of ICDE
18. Gaffney S, Smyth P (1999) Trajectory clustering with mixtures of regression models. In: Proceedings of SIGKDD
19. Giannotti F, Nanni M, Pedreschi D. Pinelli F (2007) Trajectory Pattern Mining. In: Proceedings of SIGKDD
20. Giannotti F, Pedreschi D (eds) (2008) Mobility, data mining and privacy, geographic knowledge discovery. Springer, UK
21. Hong DH, Kim C (1999) A note on similarity measures between vague sets and between elements. Inf Sci 115(1–4):83–96
22. Hung W-L, Yang M-S (2004) Similarity measures of intuitionistic fuzzy sets based on Hausdorff distance. Pattern Recogn Lett 25(14):1603–1611
23. Keogh EJ, Pazzani MJ (2000) A simple dimensionality reduction technique for fast similarity search in large time series databases. In: Proceedings of PAKDD
24. Kianmehr K, Alshalalfa M, Alhajj R (2009) Fuzzy clustering-based discretization for gene expression classification. Knowl Inf Sys, pp 0219–3116 (Online)
25. Lee J-G, Han J, Whang K-Y (2007) Trajectory clustering: a partition-and-group framework. In: Proceedings of SIGMOD
26. Li Y, Olson DL, Qin Z (2007) Similarity measures between vague sets: a comparative analysis. Pattern Recogn Lett 28(2):278–285
27. Li Y, Zhongxian C, Degin Y (2002) Similarity measures between vague sets and vague entropy. J Comput Sci 29(12):129–132
28. Lloyd S (1982) Least squares quantization in PCM. IEEE Trans Inf Theor 28(2):129–137
29. Mitchell HB (2003) On the Dengfeng–Chuntian similarity measure and its application to pattern recognition. Pattern Recogn Lett 24(16):3101–3104
30. Nanni M, Pedreschi D (2006) Time-focused clustering of trajectories of moving objects. J Intell Inf Sys 27(3):267–289
31. Pelekis N, Kopanakis I, Ntoutsi I, Marketos G, Andrienko G, Theodoridis Y (2007) Similarity Search in Trajectory Databases. In: Proceedings of TIME

32. Pelekis N, Kopanakis I, Kotsifakos EE, Frentzos E, Theodoridis Y (2009) Clustering trajectories of moving objects in an uncertain world. In: Proceedings of ICDM
33. Pfoser D, Jensen CS (1999) Capturing the uncertainty of moving-object representations. In: Proceedings of SSD
34. Theodoridis Y, Silva JRO, Nascimento MA (1999) On the generation of spatiotemporal datasets. In: Proceedings of the 6th int'l symposium on spatial databases
35. Trajcevski G, Wolfson O, Hinrichs K, Chamberlain S (2004) Managing uncertainty in moving objects databases. ACM TODS 29(3):463–507
36. Vlachos M, Kollios G, Gunopulos D (2002) Discovering similar multidimensional trajectories. In: Proceedings of ICDE 2002
37. Wang W, Yang J, Muntz RR (1997) STING: A statistical information grid approach to spatial data mining. In: Proceedings of VLDB
38. Waterman MS, Smith TF, Beyer WA (1976) Some biological sequence metrics. Adv Math 20(4):367–387
39. Weng C-H, Chen Y-L (2009) Mining fuzzy association rules from uncertain data. Knowl Inf Sys, pp 0219–3116 (Online)
40. Yi B-K, Jagadish H, Faloutsos C (1998) Efficient retrieval of similar time sequences under time warping. In: Proceedings of ICDE
41. Zadeh LA (1965) Fuzzy sets. Inf Control 8(3):338–353
42. Zhang T, Ramakrishnan R, Livny M (1996) BIRCH: An efficient data clustering method for very large databases. In: Proceedings of SIGMOD
43. Zhizhen L, Pengfei S (2003) Similarity measures on intuitionistic fuzzy sets. Pattern Recogn Lett 24(15):2687–2693

Author Biographies

Dr. Nikos Pelekis is a Lecturer at the Department of Statistics and Insurance Science, University of Piraeus. Born in 1975, he received his B.Sc. degree from the Computer Science Department of the University of Crete (1998). He has subsequently joined the Department of Computation in the University of Manchester (former UMIST) to pursue his M.Sc. in Information Systems Engineering (1999) and his Ph.D. in Moving Object Databases (2002). He has been working for almost ten years in the field of data management and data mining. He has co-authored more than 40 research papers and book chapters and he is a reviewer in many international journals and conferences. His research interests include knowledge discovery, data mining, machine learning and spatiotemporal databases.

Dr. Ioannis Kopanakis is an Assistant Professor and Head of the Dept. of Commerce and Marketing at the Technological Educational Institute of Crete. He holds a Diploma in computer science from the University of Crete (1998), Greece, an M.Sc. in information technology (1999), and a Ph.D. in computation (2003), both from UMIST, UK. He is the scientific Director of the e-Business Intelligence Lab (http://www.e-bilab.com). His research interests include data mining, visual data mining, and business intelligence. He has been involved in fifteen Hellenic and in four European research programs. He has published more than thirty papers in journals and refereed conferences, and he has demonstrated the results of his work in Europe and the US. His latest distinction was the "Best Application Paper" award at the IEEE International Conference on Data Mining (ICDM09-Miami) for his paper "Clustering Trajectories of Moving Objects in an Uncertain World".

Evangelos E. Kotsifakos received his Ph.D. (2010) in Pattern Management and Data Mining from the Department of Informatics, University of Piraeus, Greece. He was born in 1978 in Athens and he received his Bachelor (2001) and Master (2003) degree in information systems from the department of Informatics of Athens University of Economics and Business. His research interests include pattern management, data mining and scientific databases. He has participated in various database conferences and has over 13 publications in conferences and journals of that field. He also has professional experience in software engineering and IT consultanting.

Dr. Elias Frentzos received his Diploma in Civil Engineering and M.Sc. in Geoinformatics, both from NTUA. He also holds a Ph.D. from the Department of Informatics of the University of Piraeus where he is currently a Postdoc researcher, scholar of the Greek State Scholarships Foundation. He has published more than 20 papers in scientific journals and conferences such as IEEE TKDE, ACM SIGMOD and IEEE ICDE. He has participated in several national and European research projects, and also involved in the development of several commercial GIS-related applications and projects. His research interests include spatial and spatiotemporal databases, location-based services and geographical information systems.

Yannis Theodoridis is Associate Professor with the Department of Informatics, University of Piraeus (UniPi), where he currently leads the Information Systems Laboratory (http://infolab.cs.unipi.gr). Born in 1967, he received his Diploma (1990) and Ph.D. (1996) in Electrical and Computer Engineering, both from the National Technical University of Athens, Greece. His research interests cover spatial and spatio-temporal database management, mobility analysis and knowledge discovery. In the above topics, he currently participates in several national and European projects, including MODAP (FP7/ICT, 2009-12) and MOVE (COST, 2009-13). He has served or is serving as general co-chair for SSTD'03 and ECML/PKDD'11, vice PC chair for IEEE ICDM'08, member of the editorial board of IJDWM, and member of the SSTD Endowment. He has co-authored three monographs and over 80 refereed articles in scientific journals and conferences with more than 600 citations in his work. He is member of ACM and IEEE.