Clustering Trajectories of Moving Objects in an Uncertain World

Nikos Pelekis[1], Ioannis Kopanakis[2], Evangelos E. Kotsifakos[1], Elias Frentzos[1], Yannis Theodoridis[1]

[1]Dept. of Informatics, Univ. of Piraeus, Greece
{npelekis, ek, efrentzo, ytheod}@unipi.gr

[2]Tech. Educational Institute of Crete,
Greece i.kopanakis@emark.teicrete.gr

Abstract

Mining Trajectory Databases (TD) has recently gained great interest due to the popularity of tracking devices. 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. Third, we propose a variant of the Fuzzy C-Means (FCM) clustering algorithm, which embodies CenTra at its update procedure. The experimental evaluation over real world TD demonstrates the efficiency and effectiveness of our approach.

1. Introduction

With the integration of wireless communications and positioning technologies, TD have become increasingly popular, posing great challenges to the data mining community [12]. 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 [18], uncertainty in querying and answering [19], fuzziness by purpose during pre-processing for preserving anonymity [1], and so on. Although uncertainty is inherent in TD, to the best of our knowledge there is no related work in the database literature that studies its effect in the knowledge discovery process.

For example, clustering of trajectories into separate collections, involves partitioning of a TD into clusters, so that each cluster contains similar trajectories, according to a similarity definition. Several approaches try to quantify the (dis)-similarity between trajectories, dealing with basic trajectory features, [20], [23], [6], [7], [17]. However, neither of the above deals with uncertainty aspects.

On the other hand, clustering approaches based on fuzzy logic [24], such as FCM [4], 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 [3] are generalized fuzzy sets [24] that can be useful in coping with the *hesitancy* originating from imprecise information. The elements of an intuitionistic fuzzy set 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. However, in the TD setting where trajectories are complex objects, 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. Finally, 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 to the higher processing cost of density-based algorithms), whereas produces non-spherical clusters due to the inclusion of *CenTra,* that recognises representative movements of any shape. Summarizing our contributions:

- we propose an intuitionistic fuzzy vector representation of trajectories that 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.

The rest of this paper is structured as follows: Section 2 discusses related work. In Section 3, we introduce the intuitionistic vector representation of trajectories. The proposed similarity measure is defined in Section 4 while in Section 5 we describe the CenTra and the CenTR-I-FCM algorithms. In Section 6 we conduct an experimental study over a real trajectory dataset. Finally, we conclude this study in Section 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.

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 deviates from its real location due to a variety of reasons, which include both the measurement error of the positioning method, and the sampling error due to the interpolation method employed in-between sampled positions. The notion of sampling error and its behavior across the time axis are studied in [18]. In [19], a model for uncertain trajectories is proposed that associates an uncertainty threshold ε to the whole trajectory. This approach results in trajectories with uncertainty modeled as 3D cylindrical volumes. Hence, trajectory points (x, y, t) are associated with an ε-uncertainty area, actually a horizontal disk with radius ε centered at (x, y). In order to reduce the complexity of handling this kind of spherical neighborhoods, square uncertainty areas were introduced in [12].

TD Clustering - The vast majority of the proposed clustering algorithms, such as *k*-means [16], BIRCH [25], DBSCAN [9], and STING [21] 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 [20], [23], [6], [7]. None of the previous approaches considers the underlying uncertainty. On the other hand, clustering approaches based on fuzzy logic [24], such as Fuzzy C-Means (FCM) [4] and its variants are competitive to conventional clustering algorithms, especially for real-world applications. 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).

Among the related works, the ones by Gaffney et al. [11], [5] and Lee et al. [14] are closest to ours. Gaffney et al. [11], [5] 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. Recently, Lee et al. [14] 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 [14]. 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 [14], 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, [14] does by no means deal with the uncertainty in TD.

Intuitionistic Fuzzy Sets and Similarity - Regarding the theoretical foundations of fuzzy and intuitionistic fuzzy sets, these are described in [24], [3]. 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

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

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

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

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

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

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. Recently, Li et al. [15] provided a comprehensive survey and a detailed comparison of those measures.

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.

Table 1. Table of notations

Notation	Description
E={x_1, x_2, ..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 ($1 \le k \le m$ and $1 \le l \le n$), 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
$UnTra(\overline{T}_i)$, $ur_{i,j}$	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_{UnTra} $(= \lvert A - B\rvert_{IFS}^{UnTra})$, D_{IFS}, D_{total}	The distance measure between (a) two *UnTras,* (b) two *I-UnTra*s, and, (c) two trajectories
$mbr(ur)$, $\lvert ur_i - ur_j \rvert_{min}$, $\lvert ur_i - ur_j \rvert_{ext}$	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

3. Intuitionistic fuzzy vector representation of trajectories

Representing trajectories 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 $T_i = <(x_{i,0}, y_{i,0}, t_{i,0}), \ldots, (x_{i,ni}, y_{i,ni}, t_{i,ni})>$, 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 minute periods). Regarding the spatial dimension, we assume a regular grid of equal rectangular cells with user-defined size (e.g., 100×100 m^2); in each cell an identifier is also attached. Given this setting, and inspired by the Piecewise Aggregate Approximation (PAA) technique [13], we propose a method that partitions T_i into $p \ll n_i$ 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:

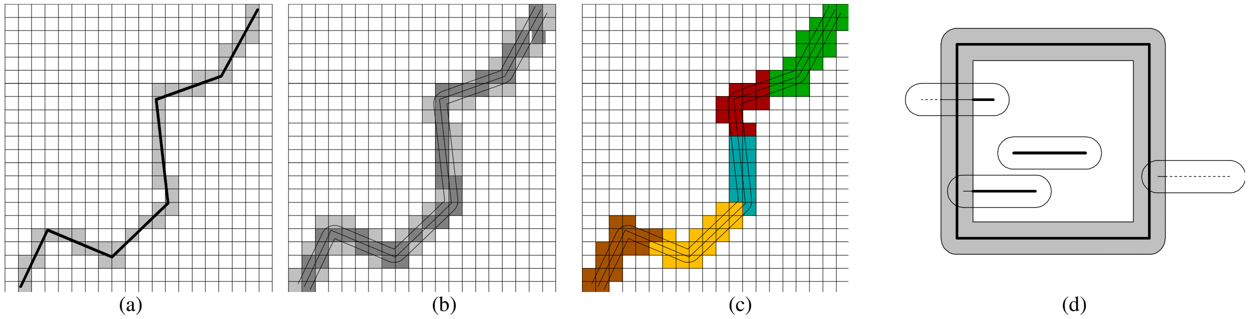


Figure 1 (a) Crossed cells by trajectory, (b) by UnTra with $\varepsilon = 1$, and (c) UnTra with $p = 5$. (d) Representation of membership, non-membership, and hesitancy in the continuous space

Definition 3. Given (i) a regular grid G of granularity $m \times n$ consisting of cells $c_{k,l}$ ($1 \le k \le m$ and $1 \le l \le n$), (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* $\overline{T}_i = \langle r_{i,1}..r_{i,p}\rangle$ 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

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

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, inspired by the approach proposed in [12], we model the *Uncertain Trajectory* (*UnTra*) of $\overline{T}_i$ over G to be $\overline{T}_i$ with its regions $r_{i,j}$ been extended to cover some neighbouring cells, the ones that are touched by the ε-buffer [12] of the initial trajectory T_i. (A similar idea is also found in [19], where each trajectory is modelled as a circular disk evolving in the temporal dimension, thus forming a cylindrical volume.) Formally:

Definition 4. Given an approximate trajectory $\overline{T}_i = \langle r_{i,1}..r_{i,p}\rangle$ and an uncertainty threshold ε, the *Uncertain Trajectory* $UnTra\left(\overline{T}_i\right) = \langle ur_{i,1}..ur_{i,p}\rangle$ of $\overline{T}_i$ 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 crosses during p_j. ■

To clarify the above 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 (Figure 1(a)). Figure 1(b) illustrates the *UnTra* counterpart of Figure 1(a) with $\varepsilon = 1$. Assuming a target dimension $p = 5$, T_i is approximated by $UnTra(\overline{T}_i)$, which simply consists of five uncertain regions, reflecting the partitioning of the above grey cells in five subsets (i.e. differently colored regions in Figure 1(c)) 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 temporal dimension (i.e., each $ur_{i,j}$ is only subject to spatial uncertainty), Figure 1(d) 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 [19]. 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 $\overline{T}_i$, 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 $UnTra(\overline{T}_i)$, its intuitionistic counterpart, $I\text{-}UnTra(\overline{T}_i)$, is defined as a p-dimensional vector of triplets $\langle(ur_{i,j}, \mu_A(ur_{i,j}), \gamma_A(ur_{i,j})), \ldots, (ur_{i,p}, \mu_A(ur_{i,p}), \gamma_A(ur_{i,p}))\rangle$ 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:

$$\mu_A(ur_{i,j}) = |r_{i,j}| \Big/ \left|UnTra\left(\overline{T}_i\right)\right|, \tag{1}$$

$$\gamma_A(ur_{i,j}) = \left(\left|UnTra\left(\overline{T}_i\right)\right| - |ur_{i,j}|\right) \Big/ \left|UnTra\left(\overline{T}_i\right)\right| \tag{2}$$

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

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:

$$\pi_A(ur_{i_j}) = \left(|ur_{i_j}| - |r_{i_j}|\right) \Big/ \left|UnTra\left(\overline{T}_i\right)\right| \tag{3}$$

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:

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

If we assume that D_{UnTra} and D_{IFS} satisfy the metric space properties, it is straightforward to prove that D_{total} as defined above is a metric. As such, the two steps that are required include the proposals of distance metrics for D_{UnTra} and D_{IFS} (Sections 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) [6]. 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 [20], DTW [23], and EDR [7] do not satisfy the metric space properties [6]. 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 $d \in \{min, ext\}$ between their corresponding *Minimum Bounding Rectangles* (*mbr*):

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

and

$$\left|ur_i - ur_j\right|_{ext} = 1 - \frac{1}{2}\left(\begin{array}{c} \dfrac{ext_x\left(mbr\left(ur_i\right)\right) + ext_x\left(mbr\left(ur_j\right)\right)}{2 \cdot ext_x\left(mbr\left(ur_i \cup ur_j\right)\right)} + \\ \dfrac{ext_y\left(mbr\left(ur_i\right)\right) + ext_y\left(mbr\left(ur_j\right)\right)}{2 \cdot ext_y\left(mbr\left(ur_i \cup ur_j\right)\right)} \end{array}\right), \tag{6}$$

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. $ext_x\left(mbr(ur_i)\right)$ is the extent of the *mbr* of ur_1 along the x axis. ■

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. 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 $UnTra\left(\overline{T}_i\right)$ and $UnTra\left(\overline{T}_j\right)$, is given by:

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

where $Rst\left(UnTra\left(\overline{T}_i\right)\right)$ denotes the remaining regions of $Rst\left(UnTra\left(\overline{T}_i\right)\right)$ 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 [6] 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 Waterman et al. [22]. ■

Theorem 1 The distance measure D_{UnTra} between $UnTra\left(\overline{T}_i\right)$ and $UnTra\left(\overline{T}_j\right)$, 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 $A = \{\langle x, \mu_A(x), \gamma_A(x)\rangle \mid x \in E\}$ 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:

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

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

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

and similarly for Γ_A, Γ_B and Π_A, Π_B. ■

The above 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 $Z(A,B) = (0.4/0.5 + 0.2/0.2 + 0.3/0.4) / 3 = 0.85$, 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:

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

which is proven to be a distance metric.

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

Proof sketch: One can easily verify that isolation, symmetry, and triangular inequality properties hold for D_{IFS}. ■

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 [15], in which other measures fail. For example, the majority of the similarity measures reviewed in [15], fail to result to a valid intuitionistic value for specific cases; some of them result to 0 or 1 suggesting that the compared sets are either totally irrelevant or identical, while it is obvious that this is false in the general case, while others result in a high similarity value for obviously different sets.

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, Bezdek et al. [4] 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}:

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

Given this, the FCM objective function has the form:

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

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, Lee et al. [14] 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, 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 they diverge would result in non representative motions patterns that can not 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 as the number of distinct trajectories traversing the cell), a region density threshold δ, a similarity threshold σ and a set S of p-dimensional $UnTra\left(\overline{T_i}\right)$, 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*), 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* is $Sim(L\_Centra, ur_{i,j}) \geq \sigma$, and (b) adopted regions $AR_{i,j}$ have average density $avg\_density(AR_{i,j}) \geq \delta$. ■

Figure 2 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 σ and dense over δ. The seed region is determined as the one with the minimum average distance from the rest candidate regions. Subsequently, the growing process begins (line 4) and the algorithm tries to find the next region to extend (lines 5-6) among the *k* Most Similar Trajectories (*k*-MST) [10], as someone would expect to find the *best region* in one of these *k* regions. Note that searching for the *k*-MST introduces only a small overhead in the algorithm's execution since the corresponding results are kept in a priority queue 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 has the maximum similarity, or, 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* to *CenTra* (line 12).

```
Algorithm CenTra(set of I-UnTra $S$, Grid $G$, Real
$\delta$, Real $\sigma$, Integer $k$)
01.  $CenTra = \varnothing$ ;
02.  forall temporal periods $p_j$
03.    $L\_CenTra$ = Init_Local_Centra($p_j$);
04.    repeat
05.      forall regions $ur_{i,j}$ in $k\text{-}MST$
06.        $AR_{i,j}$ = $L\_CenTra$ extended with $ur_{i,j}$;
07.        $AR$ ={$ur_{i,j}$|Sim($L\_CenTra$,$ur_{i,j}$)$\geq\sigma$
                      and avg_density($AR_{i,j}$)$\geq\delta$};
08.      if $AR \neq \emptyset$
09.        $ur_{i,j}$=argmax$_{reg \in AR}$(Sim(L_CenTra,$AR_{reg}$),
                               avg_density($AR_{reg}$));
10.        $L\_CenTra = AR_{i,j}$;
11.    until $AR \neq \emptyset$;
12.    $CenTra = CenTra \cup L\_CenTra$;
13.  return $CenTra$;
```

Figure 2: CenTra Algorithm

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 can not utilize intrinsically such qualitative information, we propose a different perspective by substituting the distance function with the distance metric D_{total} introduced in Section 4. As such, the fuzzy c-means objective function takes the form:

$$J_m^{CenTR-I-FCM}(U,V)=\sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik})^m\left|x_k-v_i\right|_{IFS}^{UnTra} \tag{13}$$

Theorem 2. Given a ($p\times c$)-dimensional matrix *V* storing *c* centroids trajectories *I-UnTra* of dimensionality *p*, a distance $\left|x_k-v_i\right|_{IFS}^{UnTra}$ 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 :

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

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

$$\underset{\substack{1\le i\le c\\ i\le k\le N}}{\forall} u_{ik}=\begin{cases} \left(\left|x_k-v_i\right|_{IFS}^{UnTra}\right)^{\frac{1}{1-m}}\bigg/\sum_{j=1}^{c}\left(\left|x_k-v_j\right|_{IFS}^{UnTra}\right)^{\frac{1}{1-m}},\ I_k=\varnothing, \\ \begin{cases} 0, & i\notin I_k \\ \sum_{i\in I_k}u_{ik}=1, & i\in I_k \end{cases},\quad I_k\neq\varnothing, \end{cases} \tag{14}$$

and

$$\underset{1\le i\le c}{\forall}\quad v_i=\sum_{k=1}^{N}(u_{ik})^m x_k\bigg/\sum_{k=1}^{N}(u_{ik})^m. \tag{15}$$

■

Proof sketch: Eqs. (14) and (15) follow from straightforward mathematical operations. ■

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. (15) 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

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

we obtain that:

$$\underset{1\le j\le p}{\forall}\quad \mu_{i_j}=\sum_{\forall k\in C_i}\mu_{k_j}\bigg/\left|C_i\right|,\quad \nu_{i_j}=\sum_{\forall k\in C_i}\gamma_{k_j}\bigg/\left|C_i\right| \tag{16}$$

Using the update procedure of Eq. (15) in the TD setting we would share the same problems with FCM and k-means. Since we are especially interested in the representation of 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 2, will result to more meaningful trajectory clustering. The idea is that the algorithm implied by Theorem 2 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 Figure 3 with the invocation of *CenTra*.

```
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.(14)
04.   Update the centroids' matrix V^(j)
        using CenTra;
05.   Compute membership and non-membership
        degrees of V(j)  // use Eq.(16)
06. Until ||U^{j+1}-U^j||_F≤ε; j=j+1;
```

Figure 3 CenTR-I-FCM algorithm for clustering I-UnTra

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

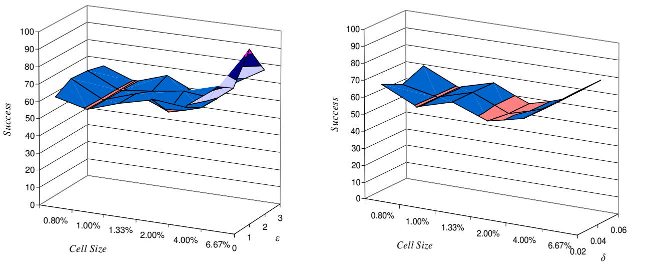
To the best of our knowledge in the TD domain there is no available real dataset already clustered by a domain expert in order to be used as ground truth for benchmarking. Nevertheless, in this paper, we have used a real dataset from which we extracted real clusters. The initial dataset consists of the 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). There are 112,300 position records consisting of the truck identifiers, dates and times, and geographical coordinates. The temporal spacing is regular and equals 30 seconds. From these raw data, we produced 1100 trajectories by splitting the recordings of a truck in subsets if there was a temporal gap between two consecutive recordings larger than 15 minutes (a gap that indicates a stop not due to traffic or traffic lights). Subsequently, we used the CommonGIS visual analytics tool [2] to manually identify real clusters, thus producing four identifiable clusters. More specifically, we 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 directions of the trips of the two clusters differ (we call these two clusters *"round trips"*). Likewise, we also discovered two clusters, moving E → W and W → E, respectively, (we call these two clusters *"linear trips"*).

6.2 Experiments

We implemented a variation of the classic FCM algorithm appropriately modified for our needs. In order to be as fair as possible, this algorithm, named TR-FCM, uses our point vector representation of trajectories, along with the minimum distance between MBRs so as to calculate the distance between the cluster's centroid and each candidate trajectory. In our first experiment we employed only the two "linear trips" clusters. We then used our CenTR-I-FCM and TR-FCM algorithms varying the grid's *cell size* and ε, and we measured the algorithm's success as the percentage of the correctly classified trajectories. The corresponding results regarding CenTR-I-FCM are illustrated in Figure 4; note that *cell size* in Figure 4(a) and (b) is demonstrated as percentage of the size of the total space. Regarding the other experiment's parameters, in Figure 4(a) we fix the value for the density threshold δ to 2% (of the total number of trajectories), while in Figure 4(b), we set ε to 1. In all cases we fix parameters σ to 0.5 and k to the number of trajectories in each cluster.

Clearly, Figure 4 demonstrates that CenTR-I-FCM achieves very good results, with a typical rate above 70%, while it reaches 90% when the cell size is set to its maximum value, regardless of the value of δ and ε, as clustering is performed at a higher granularity level where specific movement details are vanishing. On the other hand, when using the same experimental settings over TR-FCM, it produces rather poor results, with an average success of about 53% regardless of the experimental settings. We also performed the same experiments on the other two clusters (i.e., "round trips"); the respective figures are omitted due to space constraints. Nevertheless, the general observation obtained from this study, is that the CenTR-I-FCM outperforms TR-FCM regardless of the experimental setting, verifying that it produces very good clustering results, with a typical rate above 65%.

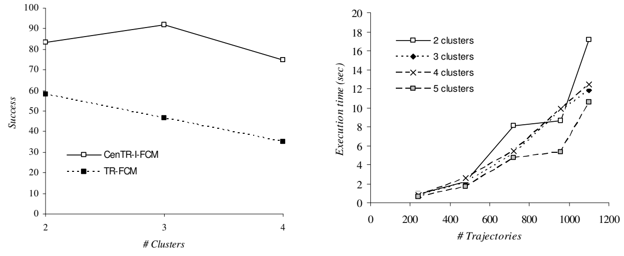
In order to study the algorithms' behaviour in cases where more than two clusters are present, we performed another experiment using different portions of the trucks dataset containing three (i.e., the two "round trips" clusters, and one of the "linear trips" clusters), and four clusters. The results of this experiment are illustrated in Figure 5(a); again, CenTR-I-FCM clearly outperforms its competitor. On the other hand, the performance of both algorithms evidently downgrades as the number of requested clusters increases; however the performance of our proposal decreases with a smaller ratio, always being above 75%.



(a) (b)

Figure 4: Clustering accuracy scaling (a) cell size, ε and (b) density threshold, δ

Regarding the performance of the CenTR-I-FCM algorithm, it was evaluated using the whole "trucks" dataset by increasing the trajectory cardinality. The results illustrated in Figure 5(b) demonstrate the efficiency of the proposed algorithm for various numbers of clusters requested. It is clear that the performance of the algorithm is not affected by the number of clusters, while all curves illustrated in Figure 5(b) imply that the algorithm has super-linear behaviour regarding the dataset cardinality.



(a) (b)

Figure 5: (a) Clustering accuracy scaling the number of clusters (b) TR-I-FCM performance scaling the dataset cardinality

To complete our experimental study, we evaluate the quality of the CenTra algorithm. Although starting from different base lines and focusing on different applications, we compare it with the representative trajectory produced by the state-of-the-art TRACLUS algorithm [14]. The result of the comparison is illustrated in Figure 6. In particular, Figure 6(a) 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, Figure 6(b) and Figure 6(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), it catches turns, and it becomes thicker in portions where something interesting (i.e. dense-similar subtrajectories) happens.

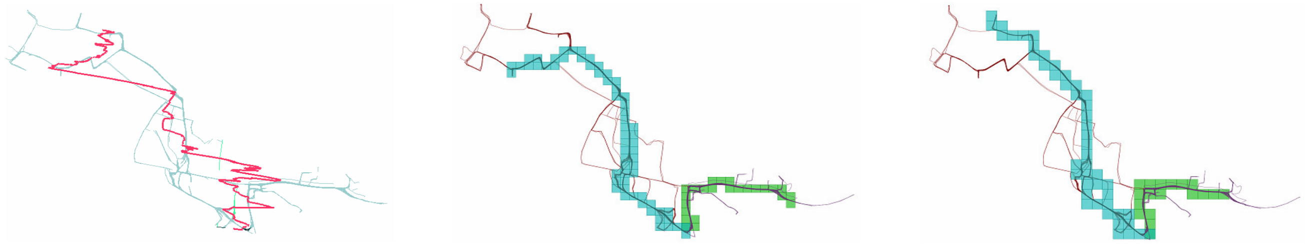

(a) (b) (c)

Figure 6: Representative Trajectories (TRACLUS) (a) and Centroid Trajectories (CenTra) ((b) with *cell size*=1.3%, ε=0 and δ=0.09, and (c) with *cell size*=2.8%, ε=0 and δ=0.02) in "round trips" dataset

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 that are inherently subject to uncertainty. Based on our 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, and used it to devise the so-called CenTR-I-FCM algorithm for clustering trajectories under uncertainty, which also includes a novel technique for discovering the centroid of a bundle of trajectories (called CenTra). The effectiveness and efficiency of our approach has been experimentally shown on a real trajectory dataset.

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.

8. REFERENCES

[1] O. Abul, F. Bonchi, M. Nanni, 'Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases', In *Proc. of ICDE*, 2008.

[2] G. Andrienko, N. Andrienko, and S. Wrobel, 'Visual Analytics Tools for Analysis of Movement Data', *ACM SIGKDD Explorations, 9 (2)*, 2007.

[3] K.T. Atanassov, 'Intuitionistic Fuzzy Sets: Theory and Applications', *Studies in Fuzziness and Soft Computing*, 35, 1999.

[4] J.C. Bezdek, R. Ehrlich, and W. Full, 'FCM: the Fuzzy c-Means clustering algorithm', *Computers and Geosciences*, 10, 1984.

[5] I. V. Cadez, S. Gaffney, and P. Smyth, 'A general probabilistic framework for clustering individuals and objects', In *Proc. of SIGKDD*, 2000.

[6] L. Chen and R. Ng, 'On the marriage of edit distance and Lp norms', In *Proc. of VLDB*, 2004.

[7] L. Chen, M. Tamer Özsu, and V. Oria, 'Robust and Fast Similarity Search for Moving Object Trajectories', In *Proc. of SIGMOD*, 2005.

[8] L. Dengfeng, C. Chuntian, 'New similarity measure of intuitionistic fuzzy sets and application to pattern recognitions', *Pattern Recognition Letters*, 23, 2002.

[9] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, 'A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise', In *Proc. of KDD,* 1996.

[10] E. Frentzos, K. Gratsias, Y. Theodoridis. 'Index-based Most Similar Trajectory Search', In *Proc. of ICDE*, 2007.

[11] S. Gaffney, and P. Smyth, 'Trajectory Clustering with Mixtures of Regression Models', In *Proc. of SIGKDD*, 1999.

[12] F. Giannotti, M. Nanni, D. Pedreschi, F. Pinelli, 'Trajectory Pattern Mining', In *Proc. of SIGKDD*, 2007.

[13] E. J. Keogh and M. J. Pazzani. 'A simple dimensionality reduction technique for fast similarity search in large time series databases'. In *Proc. of PAKDD*, 2000.

[14] J.-G. Lee, J. Han, and K.-Y. Whang, 'Trajectory clustering: a partition-and-group framework'. In *Proc. of SIGMOD*, 2007.

[15] Y. Li, D.L. Olson, Z. Qin, 'Similarity measures between vague sets: A comparative analysis', *Pattern Recognition Letters*, 28, 2007.

[16] S. Lloyd, 'Least Squares Quantization in PCM', *IEEE Trans. Information Theory*, 28(2), 1982.

[17] N. Pelekis, I. Kopanakis, I. Ntoutsi, G. Marketos, G. Andrienko and Y. Theodoridis. 'Similarity Search in Trajectory Databases'. In *Proc. of TIME*, 2007.

[18] D. Pfoser, and C. S. Jensen, 'Capturing the Uncertainty of Moving-Object Representations'. In *Proc. of SSD*, 1999.

[19] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain, 'Managing uncertainty in moving objects databases', *ACM TODS*. 29(3), 2004.

[20] M. Vlachos, G. Kollios, and D. Gunopulos, 'Discovering Similar Multidimensional Trajectories', In *Proc. of ICDE*, 2002.

[21] W. Wang, J. Yang, and R. R. Muntz, 'STING: A Statistical Information Grid Approach to Spatial Data Mining', In *Proc. of VLDB*, 1997.

[22] M.S. Waterman, T.F. Smith, and W.A. Beyer, 'Some biological sequence metrics', Advances in Mathematics, 20(4), 1976.

[23] B-K Yi, H. Jagadish, and C. Faloutsos, 'Efficient Retrieval of Similar Time Sequences under Time Warping'. In *Proc. of ICDE*, 1998.

[24] L.A. Zadeh, 'Fuzzy sets', *Information Control*, 8, 1965.

[25] T. Zhang, R. Ramakrishnan, and M. Livny, 'BIRCH: An Efficient Data Clustering Method for Very Large Databases', In *Proc. of SIGMOD*, 1996.