Mining Spatial Trajectories using Non-Parametric Density Functions

Chun-Sheng Chen, Christoph F. Eick, and
Nouhad J. Rizk

Department of Computer Science, University of Houston,
Houston, TX 77204-3010
{lyons19, vadeerat, ceick, njrizk}@cs.uh.edu

Abstract. Analyzing trajectories is important and has many applications, such as surveillance, analyzing traffic patterns and hurricane path prediction. In this paper, we propose a unique, non-parametric trajectory density estimation approach to obtain trajectory density functions that are used for two purposes. First, a density-based clustering algorithm DENTRAC that operates on such density functions is introduced. Second, unique post-analysis techniques that use the trajectory density function are proposed. Our method is capable of ranking trajectory clusters based on different characteristics of density clusters, and thus has the ability to summarize clusters from different perspectives, such as the compactness of member trajectories or the probability of their occurrence. We evaluate the proposed methods on synthetic traffic and real-world Atlantic hurricane datasets. The results show that our simple, yet effective approach extracts valuable knowledge from trajectories that is difficult to obtain with other approaches.

Keywords: Spatial Data Mining, Non-Parametric Density Function, Mining Spatial Trajectory Datasets, Density-based Trajectory Clustering

1 Introduction

We are living in a digital world where data is generated and collected ubiquitously. One large portion of this data captures motion patterns of objects and events over time. Summarizing and understanding motion patterns is important as it is instrumental to solve many important problems in our society, such as for understanding the spread of the bird flu or for better road planning. Consequently, the application of data mining techniques to trajectory data has gained significant importance in recent years.

This paper centers on clustering spatial trajectories and on the post-analysis of trajectory clusters. Most recent work in this field [9,14,8,11] proposes novel trajectory distance functions and then uses traditional clustering algorithms to cluster the trajectory data. In general, trajectory data can be categorized into spatio-temporal trajectories that contain spatial information and time and spatial trajectories that solely contain spatial information. The work presented in this paper solely focuses on mining spatial trajectories. First, a novel density based clustering algorithm DENTRAC (DENsity based TRAjectory Clustering) is introduced that operates on trajectory density functions that are generated using non-parametric density estimation techniques. The density of arbitrary trajectory in space is computed by summing the influences from all trajectories in the dataset. DENTRAC uses a unique randomized hill climbing procedure that finds local maxima of the density function by exposing trajectories to minor, random changes. Clusters are then formed by grouping trajectories that are associated with the same local maximum. Second, unique post-analysis for trajectory clusters are proposed, that compute representative trajectories from clusters, which estimate the probability of a trajectory belonging to a particular cluster, and which characterize clusters based on the shape of the density function. The main contributions of this paper include:

1. A non-parametric density estimation technique for trajectories is proposed.
2. A novel, density-based trajectory clustering algorithm named DENTRAC is introduced. To the best of our knowledge, DENTRAC is the only trajectory clustering algorithm that operates on an explicit trajectory density function.
3. Unique post-analysis techniques are proposed that use the trajectory density function to extract valuable knowledge to characterize spatial clusters.
4. The proposed methods are evaluated on synthetic traffic and real-world Atlantic hurricane datasets.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 introduces the trajectory density function, our trajectory clustering framework and the proposed methods for the post-analysis of trajectory clusters. Finally, section 4 evaluates the proposed trajectory mining techniques and section 5 summarizes our findings.

2 Related work

Distance-based clustering algorithms can be used for trajectory clustering once a proper distance/similarity measure is defined. For instance, Nanni and Pedreschi [10] proposed a distance function for trajectories and used k-means and hierarchical agglomerative clustering techniques to cluster trajectories. Lee *et al.* [8] proposed an approach that clusters line-segments first; then, from those clusters sub-trajectories are generated. Pelekis *et al.*[13] introduced a distance metric for trajectories using fuzzy sets to model uncertainty of trajectories and to compute representative, grid-based trajectories for clusters. Morris and Trivedi [9] evaluated the trajectory clustering performance of seven clustering algorithms with six different distance functions on six trajectory datasets.

Researchers also investigated the use of density-based approaches for trajectory clustering. Several DBSCAN[5]-style density-based clustering algorithms for trajectory clustering have been proposed in the literature. Palma [12] used DBSCAN to discover interesting places within trajectory datasets. Nanni and Pedreschi[11] developed a trajectory clustering algorithm called TF-OPTICS which supports interactive search to find the best clustering. BSNTC [14] defines the density of trajectories using k-nearest neighbor queries. Different from DBSCAN, DENCLUE [6] uses a non-parametric based density function to cluster the objects in space. However, none of the presented density-based trajectory clustering algorithms employs non-parametric density functions to enhance the clustering performance; to fill this gap, this paper proposes a methodology for trajectory clustering that operates on non-parametric density functions and it also proposes a trajectory clustering algorithm called DENTRAC which operates on the proposed density functions.

3 Trajectory Mining with Density Functions

3.1 Trajectory Density Estimation

In the this section, we introduce a density estimation approach that generates a density function ψ^{TRDS} from a trajectory dataset TRDS. The density of a trajectory is determined by the influence from its neighboring trajectories. The influence of an object on the density of another object can be modeled by a Gaussian kernel function that gives more weight to the nearby objects and less weight to objects that are far away. Assume a trajectory TR_i in a trajectory dataset $TRDS = \{TR_1, TR_2 ..., TR_n\}$, the influence of a trajectory $TR_i \in TRDS$ on another trajectory TR is defined as:

$$f_{influence}(TR, TR_i) = e^{-\frac{d(TR,TR_i)^2}{2*\sigma^2}} \tag{1}$$

In the above equation, d is the distance function that measures the distance between trajectories TR and TR_i; different trajectory distance functions can be used in this generic influence function such as the Fréchet distance [2] and the Hausdorff distance [1].

The parameter σ determines the width of the Gaussian kernel; it determines how quickly the influence of a point on other points decreases with distance. In our non-parametric density estimation approach, the density $\psi^{TRDS}(TR)$ of a trajectory TR is computed by summing the overall influences of all trajectories in the trajectory dataset $TRDS$; that is:

$$\psi^{TRDS}(TR) = \sum_{i=1}^{n} f_{influence}(TR, TR_i) \tag{2}$$

3.2 The DENTRAC Trajectory Clustering Algorithm

In this section, a novel clustering algorithm DENTRAC is proposed to cluster the objects in $TRDS$ based on ψ^{TRDS}. As we will see later, different hills of the density landscape correspond to different clusters and objects that are on the same hill belong to the same cluster. DENTRAC uses a randomized hill climbing to identify local maxima of ψ^{TRDS}; the hill climbing procedure is applied separately to each object in the dataset to be clustered, creating pairs (u, v) where

u is an object in the dataset and v is the result of the hill climbing process; v will be called density attractor in the following. Fig. 1 gives an example of a one-dimensional density function that has been defined for a single continuous attribute whose values range between x1 and x4; as can be seen the density function has 3 local maxima and DENTRAC will identify 3 clusters that occupy regions [x1, x2], [x2, x3] and [x3, x4], respectively in the one-dimensional attribute space. The hill climbing procedure is applied to the objects of the dataset and objects that are associated with the same density attractor are put into the same cluster; for example, the objects having round shape form Cluster2. Finally, it is interesting to compute the volume of a hill, which can be viewed as a proxy for the likelihood of a cluster; e.g. the depicted volume of Cluster3 can be computed as follows:

$$\int_{x3}^{x4} \psi^{TRDS}(x)dx \tag{3}$$

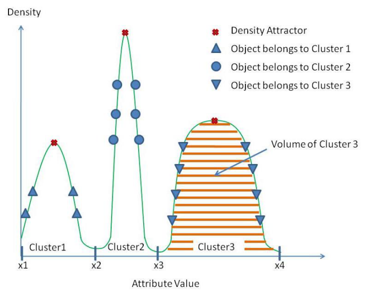

Fig. 1. A one-dimensional density function

Ideally, if we have two objects u and u' located on the same hill, the hill climbing procedure should terminate for each object with the same density attractor. Unfortunately due to rounding errors and other complications, it is very unlikely that the hill climbing will compute exactly the same density attractor. To cope with this problem DENTRAC uses a hierarchical, agglomerative clustering algorithm that generates clusters based on the distances between density attractors v from set of pairs (u, v) that were generated by the hill climbing procedure, merging clusters that are in close proximity of each other. "close proximity" is approximated by a density attractor distance threshold named dth, preventing clusters to be merged whose density attractor distance is above dth. In summary, the agglomerative clustering algorithm merges objects whose density attractors are in close proximity of each other. Algorithm 1 gives the pseudo-code for DENTRAC.

Algorithm 1

```
Input: $TRDS$, trajectory distance threshold $dth$

1. Generate $\psi^{TRDS}$
2. $D := \emptyset$
3. FOR EACH $u \in TRDS$ DO
   (a) Apply hill climbing procedure to $u$ that terminates with $v$;
   (b) Add $(u, v)$ to $D$
4. Apply Agglomerative clustering to $D$ merging clusters whose density attractor
   distance is below $dth$.
5. Return the result of the agglomerative clustering algorithm;
```

Algorithm 2

```
Input: trajectory $t$, radius $r$ "determines neighborhood size"

1. $current := t$;
2. RUN FOREVER
   (a) Create $p$ trajectories TSET in the neighborhood of $current$;
   (b) Let $p'$ the trajectory in TSET with the highest density;
   (c) IF $\psi(current) \geq \psi(p')$ THEN EXIT RETURNING $current$ ELSE $current :=$
       $p'$;
```

3.3 Hill Climbing Procedure

The goal of the hill climbing procedure is to associate density attractors with the objects in the dataset. DENCLUE's hill climbing procedure [6] computes density attractors from point objects by determining the density function maximum gradient, and then moves a predetermined step-width in the direction of the maximum gradient. However, this approach is not feasible for trajectories, because it is computationally impossible to compute derivatives of the trajectory density function; in other words, it is infeasible to compute gradients due to the complexity of trajectories and their associated distance functions. Consequently, DENTRAC relies on an iterative, randomized hill climbing procedure that generates p trajectories in the neighborhood of the current trajectory, and continues this process as long as the trajectory density increases. Algorithm 2 gives the pseudo-code of the randomized hill climbing procedure.

The hill climbing procedure generates trajectories in the neighborhood of the current trajectory by conducting small random changes on the current trajectory. This is implemented by randomly inserting, deleting, or changing points in the trajectory, with new points been selected at random within a radius r of the modified point. r is an input parameter that determines the granularity of trajectory changes. Fig. 2 illustrates the three types of change the hill climbing procedure uses to alter the current trajectory. In particular:

- In the case of replacement a point on the trajectory is picked at random and replaced with a randomly selected point within the radius r of the picked point.
- In the case of deletion a randomly selected point is deleted from the trajectory
- In the case of insertion, a insertion point is selected at random and a new point with a radius r of the selected point is inserted into the trajectory.

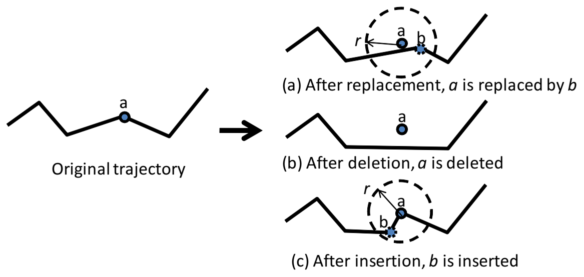
a
Original trajectory
r
a
b
(a) After replacement, a is replaced by b
a
(b) After deletion, a is deleted
r
a
b
(c) After insertion, b is inserted

Fig. 2. Three types of change to a trajectory by the hill climbing procedure. (a) replacement, (b) deletion and (c) insertion

Fig. 3 demonstrates how the randomized hill climbing procedure works for two trajectories R and B that end up to be in the same cluster. In this example, trajectory R and B are assumed to be on the same hill of a density function and the local maxima of that hill is between the two trajectories. Trajectory R moved to R' after several iterations of the hill climbing procedure and eventually stopped on the top of the density hill denoted by trajectory R". Similarly, the trajectory B moved to B' and finally stopped at trajectory B" on the top of the density hill. If the distance between trajectories R" and B" is less than a predefined density attractor distance threshold dth, trajectories R and B will be in the same cluster.

3.4 Complexity of DENTRAC

The time complexity of DENTRAC depends on the time complexity three factors:the non-parametric density estimation function, the hill climbing procedure and the agglomerative clustering algorithm that merges the density attractors. The density function estimates the density of a trajectory by summing all influences from other trajectories in the dataset. Thus the complexity for the density estimation function is $O(n \cdot O(D))$ where n is the number of trajectories and $O(D)$ is the time complexity of the trajectory distance function. An iteration in the hill climbing procedure creates p trajectories randomly in the neighborhood of the current trajectory and their density are compared with that of the current trajectory; If, at an average, it takes k iterations for this procedure to converge

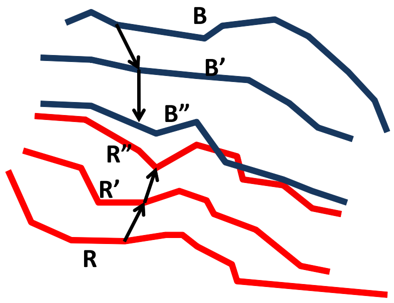

Fig. 3. An example of applying the randomized hill climbing procedure to two trajectories to find density attractors.

to its density attractor, the density function will be evaluated $p \cdot k$ times. Since the hill climbing procedure will be performed on all trajectories in the dataset, the overall time complexity is $O(p \cdot k \cdot n^2 \cdot O(D))$. After the density attractors are computed, a distance matrix between all density attractors can be pre-computed, the complexity for generating the distance matrix is $O(n^2 \cdot O(D))$.Finally, the single-link agglomerative clustering algorithm merges pairs of clusters in each iteration and has a complexity of $O(n^2)$—merge-candidates are determined by accessing the pre-computed distance matrix. Putting all things together, the time complexity for ihe DENTRAC algorithm is $O(p \cdot k \cdot n^2 \cdot O(D) + n^2 \cdot O(D) + n^2)$.

3.5 Post Analysis for Trajectory Clusters

After clusters have been obtained by DENTRAC, the question arises how those clusters can be utilized to extract useful knowledge for domain experts. In this subsection, we will introduce computational procedures, measures and displays that create useful summaries of clustering results. In Section 5 we will demonstrate how these post-analysis techniques can be used to create valuable background knowledge for hurricane and traffic trajectories.

An issue—that has been excessively explored by past research—is to generate summary trajectories for the objects belonging to a cluster. In general, clusters partition a given attribute space into disjoint regions. In our approach, the density attractor is a unique choice as a cluster summary as it represents the most likely trajectory for the region of the trajectory space that is occupied by the cluster. Moreover, density attractors can be effectively computed from a density function without the need to introduce complicated averaging procedures, as the sweeping procedure that has been proposed by Lee *et al.* [8]. Another useful choice as a summary is the cluster medoid—the member in a cluster that has the smallest distance to the other objects in the cluster. A cluster medoid identifies a representative of the most frequent group in the cluster. However medoids lack the "averaging" capabilities that are typically associated with cluster summaries. So far, our analysis ignored the density of the objects in the cluster that itself provides valuable information; a higher density indicates a higher likelihood of the trajectory to occur. From individual densities, we can compute average densities of the objects that belong to a cluster and it is also worthwhile analyzing the relationship between cluster average density and the density of the density attractors, which indicates how close the objects in the clusters are to the local maximum associated with the cluster. As our clustering approach creates clusters that solely include objects on a single hill of the density functions, high differences between density attractor density and average density indicate that most objects in the cluster are on the foot of the hill, whereas small differences indicate that objects in the cluster are near the top of a hill.

Due to the fact that density functions assess likelihoods, other more unique summaries can be created from clusters that are generated from density functions. For example, we might be interested to rank hurricane trajectory clusters based on the likelihood that hurricanes in a particular group occur in the future based on past experience; this capability would come in very handy for predicting future hurricanes. We claim that such a ranking can be performed, although there are multiple approaches to do that. The first and most natural choice is to rank clusters based on the density of their members and density attractors. However, just using the elevation ignores other characteristics of a particular density function, most importantly the density volume of the hill; e.g. we could have a very steep hill that occupies a very small area of the attribute space and a slightly lower maximum density very big hill that occupies a large portion of the attribute space; shouldn't the latter cluster that is associated with a very big hill be considered more important to characterize frequently occurring hurricanes? This raises the question, how do we measure the density volume of the hill? The challenge is to come with a size measure for the portion of the attribute space a cluster occupies and to deal with the fact that the value of the density function changes as we move in the spatial region that is occupied by the cluster. In general, as clusters occupy a larger regions of the attribute space their members have larger intra attribute distances; therefore, we propose the following measure to assess the volume of their density function hill:

Definition 1. *Let C be a cluster, $d_{intra}(C)$ is the average intra cluster distance of C and $\psi(C)$ is the average object density of cluster C, the density volume of cluster C is defined as:*

$$DensityVolume(C) = d_{intra}(C) * \psi(C) \tag{4}$$

4 Experimental evaluations

In this section, we apply the methodologies described in section 3 to an artificial traffic dataset and a hurricane dataset. We used Hausdorff distance[1] as the distance function in the following experiments. Unfortunately, there is no well-accepted measure for the quality of trajectory clustering. The distance-based measures used for traditional clusters are not suitable for evaluating trajectory clustering because they are biased to the trajectory distance function being used.

To evaluate our approach, we use a K-Medoids [7] style trajectory clustering algorithm taking the trajectory distance measures as an input as the baseline algorithm. We visually compare the DENTRAC clustering results with the baseline results to illustrate the improvement of DENTRAC over the baseline algorithm; moreover, some results will be evaluated using visual inspection

4.1 Datasets

We use one synthetic dataset and one real-world dataset. The Oldenburg traffic dataset is a synthetic dataset generated by a network-based moving objects generator [4] which is downloaded from the Internet[1]. The dataset contains 501 trajectories and 14,807 points. This dataset simulates the traces of 501 vehicles moving in the street network of city Oldenburg, Germany. We configured the dataset generator to create heavy traffic on the highway system and streets near the center of the city. Statistics of the dataset is given in Table 1.

Table 1. Statistics of the Oldenburg dataset

No. of Trajectories	501
Hausdorff Distance Between Trajectories	
Maximum	26292.95
Minimum	206.46
Average	10133.92
Standard Deviation	4291.36

The real-world dataset consists of tracks of historical tropical storms and hurricanes in the north Atlantic basin[2] from year 1950 to 2008. A trajectory is a sequence of center locations of a storm that were recorded every 6 hour (0000, 0600, 1200, 1800 UTC). This dataset has 638 trajectories and 19,788 points.

The experiments were run on a PC equipped with the Intel i7 920 2.67GHz quad-core CPU and 12GB memory. Programs are implemented in Java using the open source Cougar^2 data mining and machine learning development framework[3].

4.2 Results for the Oldenburg Traffic Data

We rank the DENTRAC clusters by the average intra-cluster distance to find clusters of similar traffic routes. The parameter setting for the DENTRAC clustering algorithm, the hill climbing procedure parameter σ is selected based on the average k-nearest neighbor distances of the dataset and the K is from 1 to 5

[1] The generator and data files are publicly available at http://www.fh-oow.de/institute/iapg/personen/brinkhoff/generator/

[2] Available under "Atlantic Tracks File 1851-2008" at http://www.nhc.noaa.gov/pastall.shtml#hurdat

(σ = 1400, 1574, 2000, 2244 and, 2446). We also used four different trajectory density threshold (dth = 0.5σ, 1.0σ, 1.5$\sigma$and 2.0σ) to get different clustering results and only the best result (σ = 1574 and dth = 1σ) is presented here due to the limited space. For the baseline algorithm, we run the k-medoid algorithm with k=25 for 30 times, each time starts with a different random seed, and the 3 clusters, of the same run, having the lowest average intra-cluster distance are reported

Fig. 4 is a comparison between clusters generated by the baseline and the DENTRAC algorithm. Clusters 1, 2 and 3 in the figure are the lowest intra-cluster distance clusters generate by each algorithm (intra-cluster distance: cluster 1 < cluster 2 < cluster 3). Table 2 listed the size and the average intra-cluster distance of the clusters. The trajectories picked up by the same DENTRAC cluster are more similar and closer in distance to each other than trajectories of the baseline cluster. Furthermore, most trajectories in the DENTRAC clusters share a part of the section with other trajectories in the same cluster which is not the case for the baseline clusters. The main reason is that the DENTRAC groups trajectories into one cluster if their density attractors are close. Trajectories sharing the same path are more likely to have the same density attractor because the overlapped section of the trajectories is normally a local maximum in a density function. Consequently, they will be grouped into the same cluster by DENTRAC.

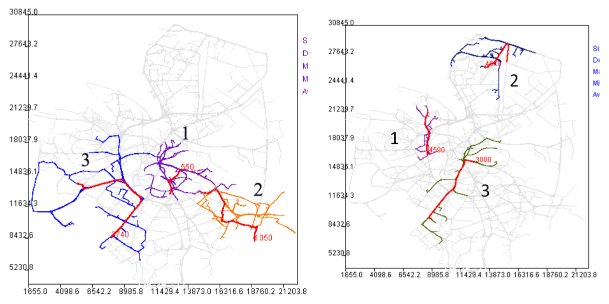

Fig. 4. Visualizations of clusters for the baseline (left) and DENTRAC (right) on the Oldenburg dataset. (Red lines are medoids in the left figure and density attractors in the right figure)

4.3 Post analysis by cluster average density and the density of density attractors

In this experiment, we perform the post-analysis on the DENTRAC clusters created in the previous section by ranking the clusters based on the density of density attractor to find clusters passing through the busiest sections of streets in the Oldenburg traffic dataset.

Table 2. The size and average intra-cluster distance (Hausdorff Distance) for clusters in Fig. 4.

Cluster	size	Average Intra-cluster distance
baseline-1	14	2910
baseline-2	15	3609
baseline-3	13	4882
DENTRAC-1	7	1462
DENTRAC-2	16	2706
DENTRAC-3	9	2881

The trajectories passing the high traffics areas in our synthetic Oldenburg dataset are identified by the clusters of high density attractor densities as shown in Fig. 5. Fig. 5 visualizes the top 3 clusters ranked by the density attractor density. Blue trajectories are cluster members and the red trajectory is the density attractor of the cluster, the rest of the dataset are plotted in light gray color as a background. The first cluster consists of trajectories using the highway on the west-side of the city; the second and third clusters are trajectories passing the major roads around the center of the city Oldenburg. Moreover, the density attractors shown in the figure further distinguish the difference between the second and the third cluster. The density attractor of the second cluster is on the road of the east-side of the downtown whereas the density attractor of the third cluster is located on the street of the southern-east of the downtown. This difference clearly points out that the trajectories of the second and third clusters are actually grouped by different dense traffic areas.

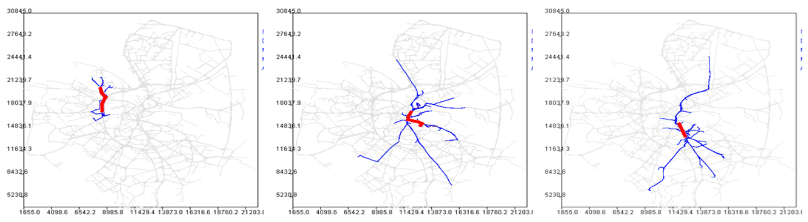

Fig. 5. Member trajectories (thin blue) and density attractors(thick red) of the top 3 clusters (left to right) ranked by the density attractor density for the Oldenburg dataset.

The x-axis in the Fig. 6 is the average intra-cluster distance and the y-axis is the density difference between the density of density attractor and the average cluster density (normalized by the maximum density attractor density). It shows that the density difference is positive correlated to the average intra-cluster distance and the correlation coefficient is 0.757. The difference between the density attractor density and the average cluster density of a cluster is a good indicator for the level of stretched out of the member trajectories in a cluster. It is because the smaller the difference is, the less changes are needed for a trajectory to reach its density attractor during the hill climbing procedure. In other words, these trajectories are closer to the top of a hill of a density function and intuitively the average distance between them should be therefore closer to the density of the density attractor.

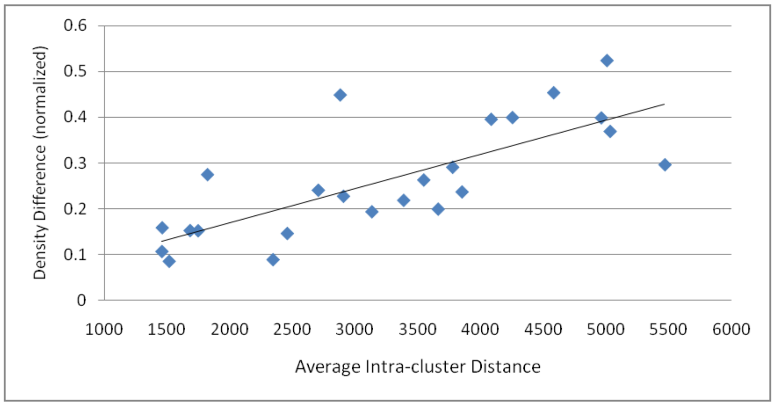

Fig. 6. Positive linear relationship between the average intra-cluster distance and the density difference between the density attractor and the average cluster density.

4.4 Results of Atlantic Hurricane Tracks Data

We applied DENTRAC to the hurricane tracks dataset to find the most likely hurricane zones and compare our results with the clustering result generated by TRACLUS, as well as the "ground truth" from the National Hurricane Center (NHC). According to the NHC, the peak hurricane season for the Atlantic Basin is from mid-August to late October. Historically, September is the month that has more hurricanes than any other month. The upper-left figure[3] in Fig. 7 was obtained from the website of NHC, and is the most likely zones and typical hurricane tracks in September. The upper-right figure is the frequent sub-trajectory clustering of the same dataset created by TRACLUS (taken directly from the TRACLUS paper [8][4]). The figures on the second row are visualizations of the top three clusters created by DENTRAC ($\sigma = 4.0$, $dth = 4.0$, $lsdt = 0.5$) ranked by the density volume defined by formula 4. The input dataset for DENTRAC were trajectories formed by connecting high density line-segments (top 50% in

[3] Climatological Areas of Origin and Typical Hurricane Tracks by Month from the National Hurricane Center, http://www.nhc.noaa.gov/pastprofile.shtml#ori

[4] We were not able to obtain the TRACLUS source code from the authors, thus we use the figure they published in their paper for comparison.

this example) that were obtained by applying an additional pre-processing step to the original dataset.

Comparing the clusters of DENTRAC with the figure downloaded from NHC, we can see that there is a good match between the 3 clusters and the most likely zones (orange zones) in NHC's figure. In particular, the trajectories in the first cluster are away from the coast and the trajectories in the third cluster are closer to the coast. The same pattern of hurricane tracks can be found in the NHC's figure too. The comparison above indicates that we can assess the likelihood of trajectories by ranking clusters based on their density volume.

The red lines in the upper-right figure are representative trajectories of clusters found by TRACLUS. There are seven frequent sub-trajectories clusters presented in the TRACLUS's figure and their representative trajectories are somewhat similar to the prevailing hurricane tracks depicted by the NHC's figure (white arrow-lines). However, we would like to point out that the results obtained by DENTRAC with the proposed post processing technique can provide more information than the cluster representatives created by TRACLUS:

1. The density volume of the DENTRAC clusters can be used to rank the probability of the hurricane clusters and this ranking cannot be provided by TRACLUS.
2. The member trajectories of the DENTRAC cluster are on the same hill of the density function so the space they occupied depicts the region covered by the high density trajectories. In the hurricane experiment, they represent the most likely zone of hurricanes. The TRACLUS's cluster representatives are not able to capture the concept of a zone that is covered by the trajectories of multiple hurricanes.

5 Conclusion and future works

In this paper, we proposed a novel density-based trajectory clustering algorithm DENTRAC that operates on a non-parametric trajectory density function. DENTRAC uses a unique, randomized hill climbing algorithm that exposes trajectories to small random changes to find local maxima of the density function. To the best of our knowledge, DENTRAC is the only trajectory clustering algorithm, operating on an explicit trajectory density function. Moreover, post-analysis techniques that extract meaningful summaries from density clusters and the underlying density function were proposed and evaluated. Using our non-parametric density based approach, we are able to use density attractors as the representative trajectory for trajectory clusters. We also demonstrated how characteristics of the density function, such as the density volume of a local maximum of the density function, can be used to obtain probability rankings of clusters. Finally, the experimental evaluation showed that by using our approach meaningful clusters and summary data can be obtained for trajectories and they provide valuable background knowledge with respect to the trajectories analyzed.

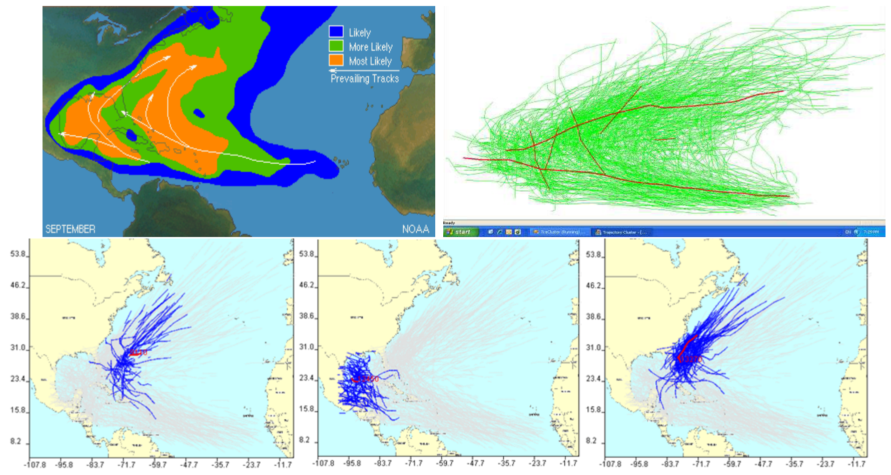

Fig. 7. NHC Visualizations of the most likely zones and typical hurricane tracks for September (upper-left), the TRACLUS [8] frequent sub-trajectory clustering result (upper-right), and the top 3 frequent sub-trajectory clusters ranked by the density volume generated by DENTRAC (2nd row).

Future work of this research is to embed other distance functions such as the Fréchet distance function, into our approach. More works can also be done in the area of trajectory and sub-trajectory likelihood assessment and prediction. A more long term plan is to investigate techniques that increase the speed of the hill climbing procedure employed; particularly, if trajectories consist of a large number of poly-lines as the current hill climbing procedure is quite slow in such a case.

References

1. Alt, H., Behrends, B., and Blömer, J. Approximate matching of polygonal shapes. *Annals of Mathematics and Artificial Intelligence 13* (1995), 251–265.
2. Alt, H., Knauer, C., and Wenk, C. Matching polygonal curves with respect to the fréchet distance. In *Proc. the 18th Annual Symposium on Theoretical Aspects of Computer Science STACS '01* (London, UK, 2001), Springer-Verlag, pp. 63–74.
3. Bagherjeiran, A., Celepcikay, O. U., Jiamthapthaksin, R., Chen, C.-S., Rinsurongkawong, V., Lee, S., Thomas, J., and Eick, C. F. Cougar^2: An open source machine learning and data mining development framework. In *Proc. Open Source Data Mining Workshop OSDM '09* (April 2009).
4. Brinkhoff, T., and Str, O. A framework for generating network-based moving objects. *Geoinformatica 6* (2002), 2002.
5. Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. AAAI Press, pp. 226–231.

6. Hinneburg, E., and Gabriel, H.-h. Denclue 2.0: Fast clustering based on kernel density estimation. In *Proc. The 7th International Symposium on Intelligent Data Analysis* (2007), pp. 70–80.
7. Kaufman, L., and Rousseeuw, P. J. *Finding Groups in Data: an Introduction to Cluster Analysis.* John Wiley & Sons, 1990.
8. Lee, J.-G., Han, J., and Whang, K.-Y. Trajectory clustering: a partition-and-group framework. In *Proc. the 2007 ACM SIGMOD international conference on Management of data SIGMOD '07:* (2007), pp. 593–604.
9. Morris, B., and Trivedi, M. Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '09* (2009), pp. 312–319.
10. Nanni, M. *Clustering methods for spatio-temporal data.* Phd thesis, CS Department, University of Pisa, Italy, 2002.
11. Nanni, M., and Pedreschi, D. Time-focused clustering of trajectories of moving objects. *J. Intell. Inf. Syst. 27*, 3 (2006), 267–289.
12. Palma, A. T., Bogorny, V., Kuijpers, B., and Alvares, L. O. A clustering-based approach for discovering interesting places in trajectories. In *Proc. the 2008 ACM symposium on Applied computing SAC '08* (2008), pp. 863–868.
13. Pelekis, N., Kopanakis, I., Kotsifakos, E., Frentzos, E., and Theodoridis, Y. Clustering trajectories of moving objects in an uncertain world. In *Proc. Ninth IEEE International Conference on Data Mining ICDM '09* (2009), pp. 417–427.
14. Zhang, Y., and Pi, D. A trajectory clustering algorithm based on symmetric neighborhood. In *Proc. WRI World Congress on Computer Science and Information Engineering* (2009), vol. 3, pp. 640–645.