Spatio-temporal Mobility Summary using Location Traces

A First Seminar Report
Submitted in partial fulfillment of the requirements for the degree of

Master of Science (by Research)

in

Information Technology

by

Manasa J M

[Roll No. 13IT72P01]

Under the supervision of

Dr. Soumya K. Ghosh(Indian Institute of Technology, Kharagpur)
Dr. Vinay Kolar (IBM Research, India)

School of Information Technology

Indian Institute of Technology, Kharagpur

Kharagpur-721302, India

Contents

Abstract

Mobility data of people is being increasingly recorded by location sensing applications such as GPS traces and Cellular Network Records. Such large-scale location data of people is capable of providing rich mobility context information about how, where and when an individual moves. These insights are useful in several domains such as hyper-targeted advertising, city transportation planning and cellular network planning. While mobility data is capable of providing such interesting information, techniques to summarize a spatio-temporal mobility of a individual is non-existent. We aim at coming up with a framework to do the same.We also propose a novel way to store mobility signatures of a person, which in turn provides a powerful mechanism for solving various use-cases that can rely on regular movement pattern of an individual (such as next-location prediction and anomalous movement detection).

1 Introduction

Advances in GPS-based applications and ubiquitous connectivity have enabled collection of vast amounts of location data describing the movement of humans and animals. Currently, location traces of millions of people are being collected by various applications. In addition, location traces of a vast majority of the population are inherently collected by cellular network operators in the form of Call Detail Records, which continuously log the base-station to which the user is connected. This large-scale location traces of people enables understanding the movement pattern of objects, and opens a plethora of location-enabled applications such as location prediction, mobility-intent identification and anomaly detection.

Location data can be used to analyze user interactions in physical space. Insights from location data enable determining user interests, demographics and who they hangout with based on where, how and when people go to different locations such as stadiums and malls. Hence, location data – similar to social networking data –enables enterprises to create new revenue streams.A primary challenge in enabling new applications is inferring insightful movement patterns from the raw location traces. Existing studies have focused on identifying hangouts of an individual, such as home/work and other frequently visited places. Hangouts provide a spatio-temporal signature of the person in terms of where the person hangs out. However, such algorithms do not indicate the mobility pattern of the user.

1.1 Motivation

Mobility Summary of an individual succinctly describes frequent paths taken by the user. Mobility Summary is the natural abstraction for many higher level applications that utilize "frequent-mobility based queries", where an application can query frequent movement patterns – instead of all the trajectories of the user – to infer some insight. Examples of frequent-mobility based queries include: (1) users who frequently pass through a given place such as a coffee shop, (2) Next-path prediction problem where user's future path is predicted based on current path and a history of user trajectories, and (3) Anomalous Trajectory Detection where outlier trajectories of a user need to filtered. Such queries are efficiently solved by a one-time computation of mobility summary. Applications can then query the summary, rather than each application querying thousands of user trajectories, to provide insights. Hence, mobility summary enables efficient and fast-lookup for many movement-based queries that rely on computing frequent trajectories of an individual. Mobility summary extraction involves working with trajectories. A **trajectory** can be defined as a series of points in the spatial domain , spread over a certain time period.

The aim is to construct the mobility summary for individual users considering only the spatial aspect. A system has to be built that takes as input a time-series of user's location traces and outputs a set of weighted representative summary trajectories of the user, which describe the user mobility pattern.

1.2 Objectives

Based on the motivation behind the work, the broad objectives are listed out below:

Trajectory Preprocessing This involves the cleansing of the raw data and methods to extract meaningful trips or trajectories out of an input chunk, which is a collection of three tuples $\langle$ *latitude, longitude, timestamp* $\rangle$

Trajectory Similarity Analysis Defining a similarity measure between two trajectories is the foundation of any kind of aggregation, or clustering. Various attempts have been made at coming up with the right similarity measure for trajectories in the literature. We discuss the existing measures, and also propose a new scheme or definition for similarity between trajectories . The proposed similarity measure is designed keeping in mind human mobility, as opposed to naturally occurring trajectories such as hurricanes , animal movement, etc.

Trajectory Clustering The next step is to come up with a clustering scheme and the other nuances related to this step. Given an input of trajectories, and a method which defines a similarity between two trajectories, the goal is to devise a clustering algorithm which outputs the final grouping of trajectories such that the final clusters define the mobility summary.

Trajectory Summarization One the clusters are formed, a representative trajectory for each of the cluster needs to be computed and that would form the entire trajectory summarization. We also want to propose a method for storing the trip summaries of a person in a hierarchical fashion using this summarization.

Applications of Mobility Summary As discussed earlier, there are various applications of the mobility summary of individuals. Some of them we aim at exploring are :

- Next Path Prediction: The problem can be defined as, given a query trajectory, can the next path or the destination of the trajectory be predicted using the mobility summary. These applications can use fast lookup on mobility summary to derive insights instead of querying all user trajectories.
- Anomaly Detection Given the mobility summary of a person, can a trajectory be categorized as anomalous.

2 Related Work

Trajectory similarity and trajectory clustering has been studied widely over the past few years due to the growing availability of data to experiment with. There have been various papers proposing new similarity measures and many others which propose an end to end framework for clustering trajectories. More than individual movement summary, people have mainly considered hurricane data, and animal movement pattern. The problem of of finding individual moment summary has not yet been addressed by anyone to the best of our knowledge. We divide the literature review into various section and discuss the papers published in each of the sections respectively.

2.1 Trajectory Preprocessing

Li *at al.* and Zheng *et al.* have a proposed a heuristic to cut meaningful trajectories from a stream of $< latitude, longitude, timestamp >$ three tuples in [12, 28] respectively. In [24], Zheng gives a complete overview on trajectory data mining which also contains a section on trajectory data preprocessing.

2.2 Trajectory Similarity

The most commonly used similarity measure for any kind of time series similarity is the LP- norm similarity. In [1], the authors talk about using Euclidean distance to measure similarity, coupled with Discrete fourier transform to reduce the dimensionality and R-tree was used for indexing. Various Fast and efficient methods for indexing and retrieval of similar series , when the similarity is defined by the Euclidean similarity are proposed in [1],[20]. The main problem with Euclidean distane is its inability to handle noise and local time shifting. In order to overcome this, exploration in other techniques began. Berndt *et al.* [2] introduced Dynamic Time warping which allowed a time series to be stretched so as to match the query series. In variations to DTW, Vlachos *et al.* [18] applied Longest Ccommon Subsequence(LCSS) measure, and Chen *et al.*[6] applied Edit Distances on Real Sequences(EDR). But all these measures, i.e., DTW, LCSS , and EDR are not metrics and don not follow triangle inequality. As an improvement, Chen *et al.*[6] introduced Edit Distance with Real Penalty (ERP) which handled local time shifting and is a metric. In [19], Wang *et al.* compare six widely used similarity measures including measures such as Euclidean distance, DTW, ERP, EDR and LCSS and their performances as trajectory similarity measures. They use a taxi dataset to evaluate and compare the results. Sankararaman *et al.* [16] have presented an extensive experimental study comparing the various similarities and also proposing a similarity of their own for trajectories.

2.3 Trajectory Clustering

In [7], Gaffney *et al.* talk about trajectory clustering using the Expectation Maximization algorithm. Nanni *et al.* [15] came up with an adaptation of a density-based clustering algorithm for trajectories.Zhang *at al.* came up with a kernel density estimation based approach for the clustering of spatio-temporal trajectories in [23].In [17], the authors propose a system to generate text summaries of the trips made by people using a partition and summarization approach. [[9],[14]] talk about trajectory clustering after snapping the trajectories on some base map. Map matching constraints the trajectories to a specific network, and various other avenues open up in terms of defining similarity.

2.4 Sub-trajectory Clustering

Li *et al.*[13] talk about moving clusters and detecting closed swarms for trajectories. Moving clusters in trajectory clustering is talked about in [8],[3] as well. Lee *et al.* [11] propose a partition and grouping framework for clustering trajectories. They try to cluster trajectories which have a common course for most of the path, or some sub trajectory similarity is high, but then start to diverge. Lee *et al.* talk about a classification algorithm for trajectories based on two methods - hierarchical region based and trajectory clustering based classification.

2.5 Representative Trajectory

Buchin *et al.* talk about various ways to find the representative or median trajectories for a group or cluster of trajectories in [4].

3 Work Done

The flow of the system and the various components are discussed, which is followed by the description of work done.

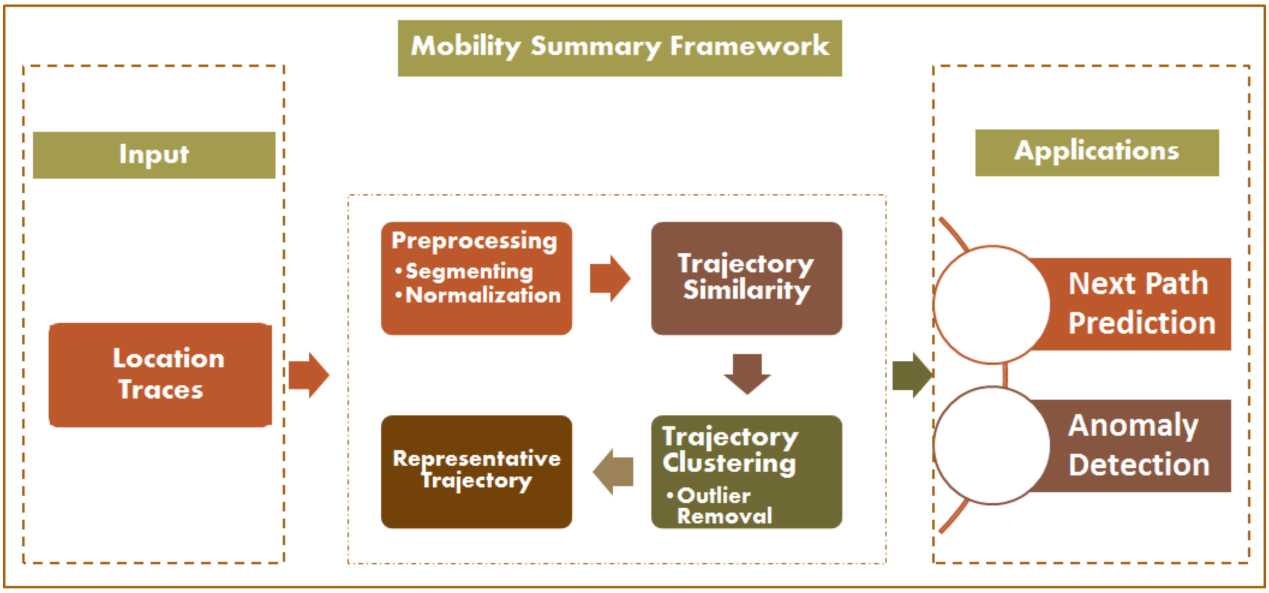


Figure 1: Overall Flow of Mobility Summary Framework

Figure 1 provides the different components for computing the Mobility Summary, which in turn can be used a variety of applications that rely on frequent-mobility based queries.The various components of the flow are described briefly below:

Input : The input in the form of a raw collection of three tuples, which are the location traces of the person or user.

Preprocessing : Preprocessing consists of the steps segmenting which is the process of making meaningful trips out of the raw traces. The second component is normalization.

Trajectory Similarity : This is the module which defines and computes the similarity between all the pairs of trajectories in the dataset. This is a very crucial step as the clustering depends on the goodness of the similarity measure.

Trajectory Clustering : This module takes as input the similarity matrix formed in the earlier step, and runs a clustering algorithm on it and outputs a final cluster of trajectories. This step also encompasses outlier or anomaly detection.

Representative Trajectory : Once we have all the final clusters, we have to come up with on trajectory per cluster which can act as the representative trajectory for that cluster. This task is taken up in this section.

Applications : The applications of Movement Summary include next path prediction and anomaly detection.

3.1 Trajectory Preprocesing

Segmentation : The first step in pre-processing is to identify meaningful trips of a user using trajectory segmentation approaches [26]. The input to the module is a raw file of three tuples containing the latitude and longitude of the user sampled at various time instances. This does not necessarily consist of moving trajectories, it might also contain tuples where the user is static for a long time. The aim is to eliminate such tuples, and extract only the meaningful trajectories where the user is in motion or has made a trip. Here the distance, velocity and time-gap between consecutive set of sample points are computed to identify if the user is mobile. The well-known representation of trajectory to denote a meaningful trip is used; it is an ordered set of 3-tuples $\langle \mathrm{latitude}, \mathrm{longitude}, \mathrm{time} \rangle$.

Normalization : The next step is to normalize the user locations since small changes in latitude and longitude can result in large distances on earth. Each raw latitude rlat_i in the sample is normalized into a normalized latitude lat_i in the range $[0, 1]$ as:

$$\mathrm{lat}_i = \frac{\mathrm{rlat}_i - \mathrm{min}_{\mathrm{lat}}}{\mathrm{max}_{\mathrm{lat}} - \mathrm{min}_{\mathrm{lat}}} \tag{1}$$

where $\min_{\text{lat}} = \min(\text{rlat}_j, \forall j)$ is the minimum latitude observed, and $\max_{\text{lat}} = \max(\text{rlat}_j, \forall j)$.

Similarly, the raw longitude rlon_i in converted to a normalized longitude lon_i in the range $[0, 1]$ as:

$$\text{lon}_i = \frac{\text{rlon}_i - \min_{\text{lon}}}{\max_{\text{lon}} - \min_{\text{lon}}} \tag{2}$$

where $\min_{\text{lon}} = \min(\text{rlon}_j, \forall j)$ is the minimum longitude observed, and $\max_{\text{lon}} = \max(\text{rlon}_j, \forall j)$.

3.2 Trajectory Similarity Analysis

Non-metric distance functions are clearly inappropriate in clustering. In addition, we show that blindly using the existing non-metric distance functions not only results in sub-optimal clusters but also incurs significant high cost in terms of computational time. A primary reason for the most non-metric functions (LCSS, DTW, EDR, ERP) are designed to suppress noise using techniques such as dynamic programming. However, for GPS trajectory, de-noising can be done as a preprocessing step by removing or resampling the outlier points [22, 25]. Hence, the modified standard LP-Norm functions are used for distance computation, and avoid time consuming non-metric algorithms.

3.2.1 Curve Distance for Trajectories

Instead of treating trajectory as a set of sample points, the trajectories can be approximated as curves on n-dimensional vector space. This approximation is reasonable if the location traces contain finely sampled GPS points (as in this case)

For trajectory representation, the main idea is to represent trajectory as a curve in two independent dimensions latitude and longitude (in $\mathbb{R}^2$) [10]. A natural extension is to represent in three independent dimensions including time. However, in this work, only latitude and longitude dimensions are looked into.

Let $f_i[0, 1] \to \mathbb{R}^2$ be the curve for the i-th trajectory trajectory, which maps a number between 0 to 1 to the (latitude, longitude) pairs of the trajectory. The standard $\mathbb{L}^2$ norm for this curve is given $\|f_i\| = \left[\int_0^1 f_i(x)^2 \, \mathrm{d}x\right]^{\frac{1}{2}}$.

$$\text{CD}(t_i, t_j) = \left[\int_0^1 (f_i(x) - f_j(x))^2 \, \mathrm{d}x\right]^{\frac{1}{2}}. \tag{3}$$

Numerically, this is solved by first resampling the trajectory to a large number of points for each trajectory (100 samples in our case) and then using Trapezoidal Rule to find the curve distance.

Weighted Curve Distance For human mobility, a user's meaningful trip has an associated intention (such as commuting to work or grocery shop visit). More often, each frequent trajectory are between end-points that are important to the user (such as home and work). A metric is hence proposed that emphasizes origin and destination of the trajectories, while computing the distance.

We first generalize the Curve Distance to Weighted Curve Distance, where all trajectories have different weights at each point. It can be shown the Weighted Curve Distance (WCD) between two trajectories t_i and t_j is given by

$$\mathrm{WCD}(t_i, t_j) = \left[\int_0^1 w(x)\,(f_i(x) - f_j(x))^2 \,\mathrm{d}x \right]^{\frac{1}{2}}. \tag{4}$$

where $w(x) \rightarrow [0, 1]$ is a weighting function. In the case of providing higher weights to origin and destination an Origin-Destination (OD) weighing function should be constructed such that the weights are high at the ends than at the center. We use Beta function $\mathrm{B}(\alpha, \alpha)$, where α in $[0, 1]$. This provides a bimodal curve where weights at the ends are higher than the weights at the intermediate points in th curve; lower values of alpha provide very high values at ends than at the intermediate points.

Since CD is a metric and WCD is a weighted combination of CD (with positive weights), it can be shown that WCD is also a metric.

3.3 Trajectory Clustering

The distance matrices are used to aggregate similar trajectories into one cluster. Hierarchical agglomerative clustering is used since it provides the flexibility of analyzing the entire merge history of user's trajectories, and then cutting the dendrogram at the right level. This enables to personalize and automate clusters for different users. Each user has different motion patterns and – apriori – we do not have information of how often/densely a user travels along different paths. Our system automatically recognizes the right number of clusters by analyzing the dendrogram.

Average link clustering is used to measure similarity between intermediate clusters. This is to avoid bias for clustering trajectories that are associatively near (using single-link) or by concentrating on the extreme points of the merge (using complete-link).

3.3.1 Finding optimal clusters

In this step, we cut the dendrogram at a level that defines meaningful mobility clusters. The main idea is to determine the optimal number of clusters for each person, and to use that knowledge to cut the dendrogram. A static similarity level to cut dendrogram is not chosen since different people have different forms of mobility. A user whose main travel pattern is long distance commute to two office locations has different distance thresholds than a student who is commuting mainly commuting from dormitory to classes.

Existing well-known methods, such as the elbow method, do not cut the dendogram at appropriate level to provide good movement summaries; which is demonstrated in Section . Hence, an algorithm is designed that cuts the dendogram at a level where trajectories in a cluster are between nearby origin and destinations, which signify similar meaningful trips of a person.

```
Input: Dendrogram with cluster information at each level
Output: Optimal Clusters (Trip Summaries)
for k=1 to N do
    $SSW(Clus_i) = \sum_{j=1}^{|Trajs(Clus_i)|} Sim(Traj_j, Mean(Clus_i)) \quad (5)$
    $SSW(Level_k) = \sum_{j=1}^{k} SSW(Cluster_j) \quad (6)$
end
Find the elbow point from the SSW Plot over all levels
Set all trajectories as unmarked for k=elbowPoint+1 to N do
    for Each non anomalous cluster do
        if Trajs(Cluster) are unmarked && isSummary($Cluster_i$) then
            Report Cluster as a final Cluster
            Mark all Trajs in Cluster
        end
    end
end
isSummary($Cluster_i$)
for All pairs of trajectories in Cluster do
    if Maximum Pointwise Distance$\leq \delta$ (kms) then
        return True
    end
end
return False
```

Algorithm 1: Algorithm for reporting final clusters from Dendrogram

The algorithm iterates down the dendrogram starting at the root; the number of clusters at this stage being $k = 1$. As we proceed down each level to cut the dendrogram, the number of clusters k increases by one. We compute the possible summary clusters of the user for each k. Let $\mathbf{S}_k = \{S_{ki}, \forall i\}$ be the set of clusters for the user at level k. Let t_{kij} be the j-th trajectory in S_{kij}. Let the mean trajectory for S_{ki} be $\bar{t}_{ki}$. We examine the tightness of clustering at this level by computing the Sum of Squares Within cluster (SSW). SSW at level k is defined as $SSW_k = \sum_{S_{ki} \in \mathbf{S}_k} \sum_{t_{kij} \in S_{ki}} \mathrm{WCD}(t_{kij}, \bar{t}_{ki})^2$.

A well-known metric is to accept the elbow point of the SSW_k vs. k curve (say, at $k = k_e$) as the optimal number of clusters. However, as we show in Section 7, the trajectories in clusters at the elbow point k_e may contain multiple type of short trips in a small region; such trips can be split further into different cluster. Hence, we iterate from the elbow point towards greater k, and each k we examine the resulting clusters to see if the trajectories from different types of meaningful trips are contained in a single cluster. We declare that different type of meaningful trips are present in a cluster if the physical distance between any point of any pair of trajectory curves in the cluster is greater than a intra-cluster separation distance T_{intra} (which is 1.5 km in our case)

Finally, the “Mobility Summary” of a user is represented as the clusters with number of trajectories greater than a certain threshold (5% of user’s trajectories)

4 Evaluation and Analysis

4.1 Dataset

- Microsoft GeoLife Dataset GeoLife Dataset published by Microsoft Research [30],[27],[29]. This is a GPS trajectory dataset with GPS traces of 182 users over a period of three years (from April 2007 to August 2012).

4.2 Clustering Effectiveness

The main measure of clustering effectiveness that is used is the Silhouette Coefficient(SC). SC is a standard metric that shows the effectiveness of clustering. SC is based on the cohesion and the separation of clusters formed. The cohesion (*a(x)*) is defined as the average distance of x to all other vectors in the same cluster. The separation (*b(x)*) is defined as the minimum of the average distances of x to the vectors in other clusters. Further, the silhouette coefficient of a data point is defined as

$$s(x) = \frac{b(x) - a(x)}{max(a(x), b(x))} \tag{7}$$

The total silhouette coefficient of the dataset is the average over all the points given by

$$SC = \frac{1}{N} \sum_{i=1}^{N} s(x) \tag{8}$$

Ideally, SC is between [-1,1], where values closer to 1 representing better formed clusters.

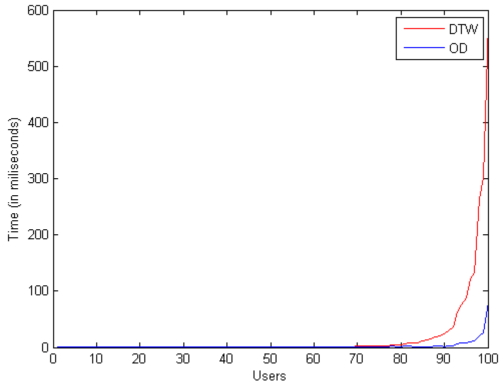


Figure 2: Computation time comparison of DTW and OD

4.3 Comparison with Dynamic Time Warping (DTW)

This is a comparison made in the choice of the similarity metric that is used. The effects of using DTW in place of the proposed similarity measure, keeping everything else in the algorithm exactly as it is, is shown.The DTW similarity between A and B is defined below

$$D_{dtw}(A,B) = \begin{cases} 0 & \text{if both A and B are empty} \\ \infty & \text{if one of A or B is empty} \\ \phi_d(head(A), head(B)) + & \\ \quad min \begin{cases} D_{dtw}(A, rest(B)), \leftarrow \textit{Stretch A} \\ D_{dtw}(rest(A), B), \leftarrow \textit{Stretch B} \\ D_{dtw}(rest(A), rest(B)) \end{cases} & \\ otherwise & \end{cases} \tag{9}$$

where $\phi_d(p1, p2) = L_2 - dist(p1, p2)$

The same framework as that proposed in our method is used for comparison, the only difference being that the DTW similarity is plugged in place of the OD based similarity defined earlier. DTW is very close to the proposed method considering the clustering effectiveness, but there are cases where it misses out trajectories that are a part of a meaningful trip summary. On the basis of computation time, proposed approach is way faster than DTW, because DTW heavily depends on the number of sample points. As the number of sample points increase, the time starts to blow up.

Issues with DTW

- DTW is not a metric as it violates triangle inequality. This can lead to issues during clustering.Any distance metric d follows triangle inequality if, for any three points, x,y,and z: d(x, z) d(x, y) + d(y, z).
- The biggest concern about DTW is the computation time. When the sample points are very large, it can get to as much as 400 times slower than the proposed approach. If the points are

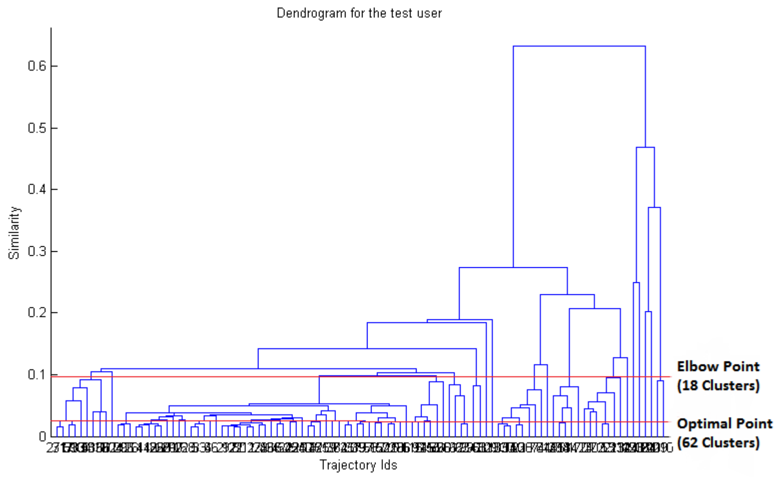


Figure 3: Dendrogram of the all the trajectories (Shows the hierarchy of similarity) The elbow point is at 18 clusters, whereas the optimal number of clusters is 62

resampled and DTW similarity is computed, it would reduce to the same as pointwise Eucledian distance, and would still be computationally more expensive. Fig 2 shows the computation time differences over all the users for clustering using DTW and OD similarity measures over all the users

- As the number of sample points is decreased, the effectiveness of both the proposed method and DTW go down. But the goodness of the clusters returned by DTW decreases more than that of the proposed method. The number of sample points is reduced in each of the trajectories to 90%,95%, and 97% and the silhouette coefficient values using DTW,LP and OD were plotted, and compared.

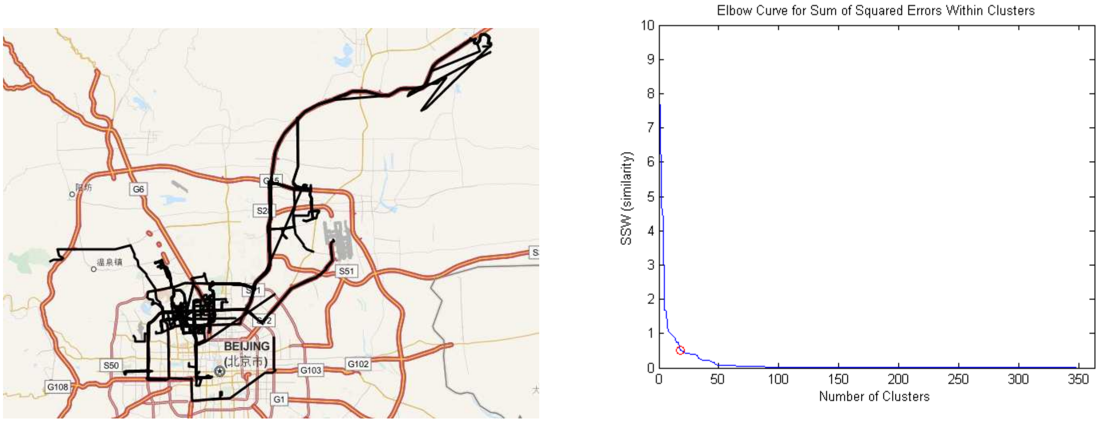
The working of the proposed method with supporting visuals at every step explaining the rationale behind it is shown hence.

4.4 Analysis of Proposed Origin-Destination(OD) Method

In this section, each stage of the proposed method is shown with the help of the corresponding visualizations at that stage for a test user. Fig. 4a shows all the trajectories of the user. The similarity matrix is computed using the similarity defined earlier and hierarchical clustering is run on it.

The dendrogram is a pictoral representation of how similar the trajectories are among each other. The ones more similar to each other are paired closer to the bottom as compared to the ones higher in the tree. Fig 3 shows the dendrogram of the trajectories of the test user. The user had 363 trajectories in total, so the dendrogram has 363 leaves.

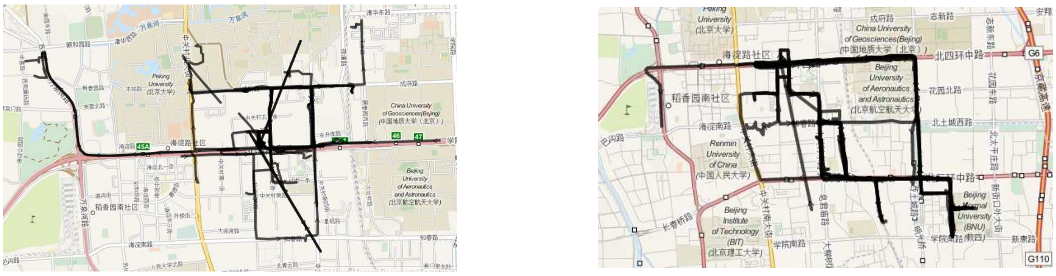
To obtain a certain cluster of the trajectories, we have to cut the dendrogram at a certain level. The height at which we cut the dendrogram decides the number of clusters we will get. Finding the right



(a) All the trajectories of the test user(363 trajs)

(b) Elbow curve - Plot of the SSW vs number of Clusters; Elbow point 18 clusters

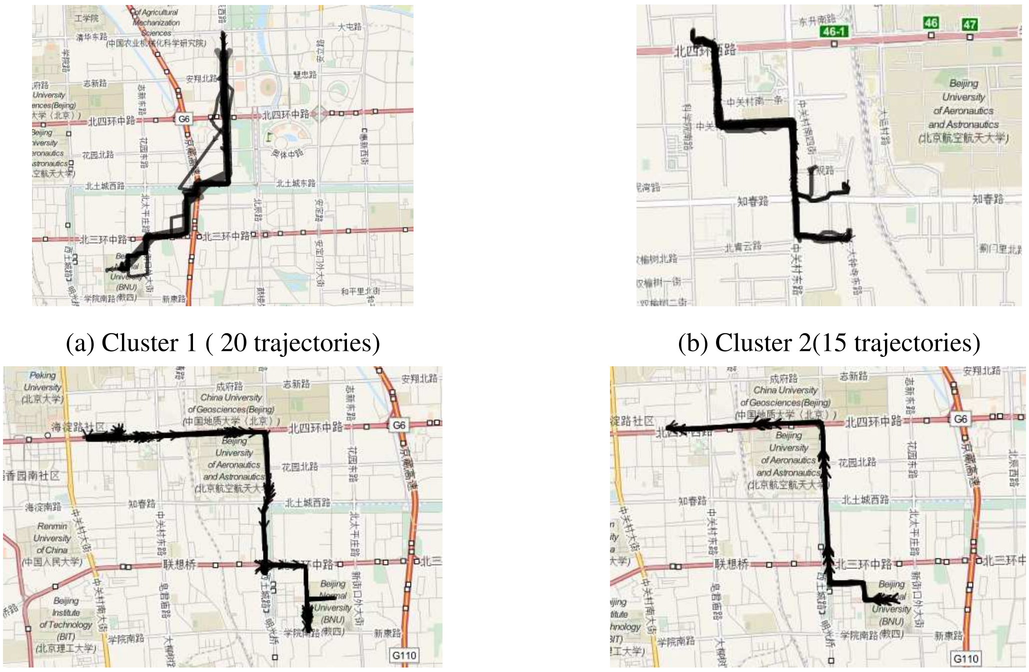
Figure 4: All Trajectories and Elbow Curve



(a) Cluster 1 (32 trajectories)

(b) Cluster 2(31 trajectories)

Figure 5: Visualizations of the top 2 clusters at Elbow Point


G6
Beijing University of Aeronautics and Astronautics
46-1
46
47
Peking University
China University of Geosciences(Beijing)
Renmin University of China
Beijing Institute of Technology (BIT)
G110

(a) Cluster 1 (20 trajectories)

(b) Cluster 2(15 trajectories)

(c) Cluster 3(11 trajectories)

(d) Cluster 4(9 trajectories)

Figure 6: Visualizations of the top 4 final optimal clusters

height at which we should cut the dendrogram is one of the biggest challenges in coming up with an accurate summary for a user. One method which is widely used in literature is to plot the cumulative Sum of Squared Errors within Cluster as the cluster number varies from 1 to N. This is called the elbow curve or the scree plot and the elbow point in this curve should give us the right number of clusters. The rationale behind this is that we want to find the saturation point beyond which even on increasing the number of clusters, the error within the clusters doesnt decrease significantly. But, in our case, we found that the elbow point doesn't get us anywhere close to the actual optimal point of clustering. The validation of the clusters formed at each value of number of clusters was done by visualizing the results and looking at the general tightness of the clusters. Fig 4b shows the elbow curve with two lines indicating the elbow point, and the optimal cluster number. We see that the actual optimal point is way beyond the elbow point, and show the visualizations of the top 4 clusters at both the elbow point, in Fig. 5.

Thus, we see that the elbow point does not get us anywhere close to the optimal number of clusters, but can help as a starting point for finding the optimal number. One insight that the elbow point gives is that the optimal point cant be before the elbow point, and hence we can prune the search space till the elbow point. The heuristic that we follow to obtain the optimal number of clusters is to traverse down the dendrogram , starting from the elbow point, and at any stage if we hit a cluster where the maximum pointwise distance between any pair of trajectories is less than δ, we report it as a final cluster.

Fig.6 shows the visualizations of the top-4 clusters at the optimal level.

5 Conclusion and Future Work

Two objectives have been worked on, and three other remain. As mentioned earlier, the next steps of work are listed as follows :

- Comparisons for trajectory Clustering : Implement other state of the art techniques existing in literature like TraClus and SWARM, and comapare the results.
- Devise a better heuristic to find the optimal number of clusters.
- Trajectory Summarization : An algorithm has to be devised to find the representative trajectory for a cluster. An efficient way of storing the clusters is also an area to be explored.
- Applications: Once the mobility summary of an individual is obtained, work has to be done to predict the next location and find anomalies.

References

[1] R. Agrawal, C. Faloutsos, and A. Swami. E cient similarity search in sequence databases. 1994.

[2] D. J. Berndt and J. Clifford. Finding patterns in time series, advances in knowledge discovery and data mining, aaai, 1996.

[3] K. Buchin, M. Buchin, J. Gudmundsson, M. Löffler, and J. Luo. Detecting commuting patterns by clustering subtrajectories. *International Journal of Computational Geometry & Applications*, 21(03):253–282, 2011.

[4] K. Buchin, M. Buchin, M. Van Kreveld, M. Löffler, R. I. Silveira, C. Wenk, and L. Wiratma. Median trajectories. *Algorithmica*, 66(3):595–614, 2013.

[5] L. Chen and R. Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 792–803. VLDB Endowment, 2004.

[6] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pages 491–502, New York, NY, USA, 2005. ACM.

[7] S. Gaffney and P. Smyth. Trajectory clustering with mixtures of regression models. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 63–72. ACM, 1999.

[8] J. Gudmundsson, M. van Kreveld, and B. Speckmann. Efficient detection of patterns in 2d trajectories of moving points. *Geoinformatica*, 11(2):195–215, 2007.

[9] B. Han, L. Liu, and E. Omiecinski. Neat: Road network aware trajectory clustering. In *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, pages 142–151. IEEE, 2012.

[10] S. Kurtek, A. Srivastava, E. Klassen, and Z. Ding. Statistical modeling of curves using shapes and related features. *Journal of the American Statistical Association*, 107(499):1152–1165, 2012.

[11] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: A partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, pages 593–604, New York, NY, USA, 2007. ACM.

[12] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, page 34. ACM, 2008.

[13] Z. Li, B. Ding, J. Han, and R. Kays. Swarm: Mining relaxed temporal moving object clusters. *Proc. VLDB Endow.*, 3(1-2):723–734, Sept. 2010.

[14] M. K. E. Mahrsi and F. Rossi. Modularity-based clustering for network-constrained trajectories. *arXiv preprint arXiv:1205.2172*, 2012.

[15] M. Nanni and D. Pedreschi. Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, 27(3):267–289, 2006.

[16] S. Sankararaman, P. K. Agarwal, T. Mølhave, and A. P. Boedihardjo. Computing similarity between a pair of trajectories. *arXiv preprint arXiv:1303.1585*, 2013.

[17] H. Su, K. Zheng, K. Zeng, J. Huang, S. Sadiq, N. J. Yuan, and X. Zhou. Making sense of trajectory data: A partition-and-summarization approach. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pages 963–974. IEEE, 2015.

[18] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 673–684, 2002.

[19] H. Wang, H. Su, K. Zheng, S. Sadiq, and X. Zhou. An effectiveness study on trajectory similarity measures. In *Proceedings of the Twenty-Fourth Australasian Database Conference-Volume 137*, pages 13–22. Australian Computer Society, Inc., 2013.

[20] B.-K. Yi and C. Faloutsos. Fast time sequence indexing for arbitrary lp norms. VLDB, 2000.

[21] B.-K. Yi, H. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pages 201–208, Feb 1998.

[22] J. Yuan, Y. Zheng, X. Xie, and G. Sun. T-drive: Enhancing driving directions with taxi drivers' intelligence. *Knowledge and Data Engineering, IEEE Transactions on*, 25(1):220–232, Jan 2013.

[23] P. Zhang, M. Deng, and N. Van de Weghe. Clustering spatio-temporal trajectories based on kernel density estimation. In *Computational Science and Its Applications–ICCSA 2014*, pages 298–311. Springer, 2014.

[24] Y. Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):29, 2015.

[25] Y. Zheng, Y. Chen, X. Xie, and W.-Y. Ma. Geolife2.0: A location-based social networking service. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09. Tenth International Conference on*, pages 357–358, May 2009.

[26] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma. Understanding mobility based on gps data. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, pages 312–321, New York, NY, USA, 2008. ACM.

[27] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma. Understanding mobility based on gps data. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 312–321. ACM, 2008.

[28] Y. Zheng and X. Xie. Learning travel recommendations from user-generated gps traces. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1):2, 2011.

[29] Y. Zheng, X. Xie, and W.-Y. Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, 33(2):32–39, 2010.

[30] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM, 2009.