Spatio-temporal Mobility Summary of Individuals

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

School of Information Technology

Indian Institute of Technology, Kharagpur

Kharagpur-721302, India

Contents

1 Introduction **1**

2 Motivation **3**

2.1 Trajectory Similarity 3

2.2 Trajectory Clustering 4

2.3 Trajectory Summarization 4

2.4 Storing the summaries of a person 4

3 Related Work **4**

3.1 Trajectory Similarity 5

3.2 Trajectory Clustering 5

3.3 Subtrajectory Clustering 5

3.4 Trajectory Preprocessing and Representative Trajectory 6

4 Formulation Overview and Preprocessing **6**

5 Inter-Trajectory Distance **7**

5.1 Curve Distance for Trajectories 7

6 Trajectory Clustering **8**

6.1 Finding optimal clusters 8

6.2 Representative trajectories 10

7 Evaluation and Analysis **10**

7.1 Dataset 10

7.2 Clustering Effectiveness 10

7.3 Individual Movement Summary 11

7.3.1 Proposed Method-OD 12

7.3.2 Variation with DBSCAN 13

7.3.3 Comparisons with DTW 15

7.3.4 Comparison with SWARM 16

7.3.5 TraClus 19

7.3.6 Next Location Prediction 20

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 has 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 [1]. 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 [2]. 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.

Large scale human mobility data enables solving interesting problems and generating new revenue streams for the enterprises. For example, the transportation departments now spend millions of dollars once in a few years in *travel pattern surveys*, which only samples a sub-5% of population, to plan the bus and train networks [24]. Such expensive and exhaustive methods can be easily replaced by analysis of location traces, which can provide real-time and fine-granular data from a significantly larger sample of people. Similarly, 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 [9]. Hangouts provide a spatio-temporal signature of the person in terms of where the person hangs out. However, such algorithms does not indicate the mobility pattern of the user.

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.

Modeling movement summaries of individuals from the location traces has not been studied in the existing literature. Some studies have examined trajectory clustering by extending point or sub-trajectory based clustering mechanisms [21, 19]. Such schemes primarily operate on points (or sub-parts) of trajectories, and finally aggregate the point clusters to compute a trajectory cluster. However, as we show in the paper, such schemes are poor in summarizing individual's trajectories for two main reasons. First, aggregating on cluster of points or sub-trajectories does not consider the similarity between entire trajectories; this often aggregates dissimilar trajectories or fails to identify similar trips within a cluster unless a careful parameter tuning is performed for each individual user. Second, these schemes do not scale to large sets of location traces since they incur high computational time; usually they repeatedly apply clustering to sub-parts on the all the trajectories and later aggregate the results.

We aim at constructing mobility summary for individual users. We have built a system 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.

We make the following contributions:

- We propose abstraction of Mobility Summary to capture important representative paths of an individual; this enables in building spatio-temporal mobility signatures of people and applications that use frequent-mobility based queries.
- We devise an efficient metric, called "Weighted LP-Norm", to compute the distance between a pair of user trajectories. We utilize this metric as a distance metric for clustering trajectories. We show that existing metrics such as Dynamic Time Warping [30] are insufficient as they are non-metrics and computationally expensive.
- We devise algorithms to determine optimal clustering of user trajectories, and determine representative summary trajectories from a set of trajectories.
- We implement Next-Path Prediction algorithm, which uses frequent-mobility query, to demonstrate that one-time computation and storage of user's mobility summary significantly reduces the complexity of applications. These applications can use fast lookup on mobility summary to derive insights instead of querying all user trajectories.

Trajectory Definition A trajectory can be defined as a series of points in the spatial domain , spread over a certain time period. It can be seen as an ordered collection of three tuples, latitude, longitude and the timestamp.

Storing the summaries In addition to finding movement summaries, we also propose a method to store the trajectories, and movement summaries at various levels of granularity. This layered view of the trajectories would help in querying the system for a snapshot of the user's mobility at any zoom level. It would also help in looking at some summaries at a particular abstraction level and diving deeper into other summaries.

2 Motivation

2.1 Trajectory Similarity

Motivation behind giving weight age to OD The intuition behind the similarity measure is that whenever humans move, there is an intent behind the trip. Thus, the origin and destination have an important role to play. If someone is making a trip, the destination has to be of some importance to the person, and the origin should also mean something to him. Following this intuition, we have given more weightage to the points closer to the origin and destination. Another reason behind doing this is that we want to overlook tiny diversions in the route taken from a set pair of origin-destination. For example, for a user, if the trips he make from the office to his home are considered as one summary, and on some day if he takes a tiny diversion in the form of a by-lane rather than the main road, it should still be considered in the same trip summary. Thus, the points closer to the origin and destination are given more importance than those in the middle.

Problems with existing metrics There are various existing metrics for computing similarity between trajectories like Dynamic Time Warping(DTW), Edit Distance on Real Sequences (EDR), Longest Common Sub-sequence(LCSS), etc. But the major problem with most of them is that they are not mathematical metrics and thus, do not follow triangle inequality. This might lead to inconsistent results in various situations, and mainly affect the clustering results.

Problems with defining similarity when both spatial and temporal domain come into the picture Moving into the the domain of defining a spatio-temporal similarity between two trajectories brings in various questions of ambiguity. Are simiarlities between two trajectories which go on the same path 10 mins part same as that of two trajectories which go 1 km apart at the same time? Hence, a hierarchical approach for solving this problem is suggested so as to decouple the spatial and temporal similarities. We first come up with the trip summaries by looking only at the spatial values, and then in the next stage , bring in the temporal (time of the day aspect) aspect to gain further insights.

Denoising over using similarity measures resilient to noise For human trip movement, denoising can be done prior to trajectory processing, instead of making the similarity measures resilient to noise. Such similarity measures are computationally expensive, and do not yield accurate results in all cases.

2.2 Trajectory Clustering

Why hierarchical: We really dont know the number of clusters. We need to iterate over each k and then find out the optimal k. The time complexity of running k-means for 100's of ks and then finding out the optimal k is much more expensive than doing 1-shot hierarchical clustering and finding a good point to cut.

2.3 Trajectory Summarization

Use cases for trajectory summarization

Computing the trajectory or trip summaries of a person can answer various queries about the person. Some of them include

- Customer Profile: Give summary of a person's trips in a region – both in space or time
- Next-Location Profile: What are the most probable trajectories to find a person between time x to time y (optional: given that the person is at location L at time t)
- Alerts: Alert when a customer is moving in an anomalous way

Trip summaries can also be used for

- Insurance Usecase: Give summary of a person's trips that have good or bad speed profiles – immaterial of the space or time

2.4 Storing the summaries of a person

We also propose a method of storing the trip summaries of a person in a hierarchical fashion. By doing this, we can get an idea about how the person moves at various levels of granularity. We can also set the number of clusters to a particular value, and query his movement pattern for the top k prominent trips. This kind of storage also supports zooming into a particular summary and breaking it down further for other analytics.

3 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.

3.1 Trajectory Similarity

The most commonly used similarity measure for any kind of time series similarity is the LP- norm similarity. In [3], 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 [3],[29],[13],[11],[17] . 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.* [4] 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.* [27] applied Longest Ccommon Subsequence(LCSS) measure, and Chen *et al.*[8] 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.*[8] introduced Edit Distance with Real Penalty (ERP) which handled local time shifting and is a metric. In [28], 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.* [25] have presented an extensive experimental study comparing the various similarities and also proposing a similarity of their own for trajectories.

3.2 Trajectory Clustering

In [12], Gaffney *et al.* talk about trajectory clustering using the Expectation Maximization algorithm. Nanni *et al.* [23] 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 [34].In [26], the authors propose a system to generate text summaries of the trips made by people using a partition and summarization approach. [[15],[22],[10]] 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.

3.3 Subtrajectory Clustering

Li *et al.*[21] talk about moving clusters and detecting closed swarms for trajectories. Moving clusters in trajectory clustering is talked about in [14],[5],[16] as well. Lee *et al.* [19] 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.

3.4 Trajectory Preprocessing and Representative Trajectory

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 [20, 39] respectively. In [35], Zheng gives a complete overview on trajectory data mining which also contains a section on trajectory data preprocessing. Buchin *et al.* talk about various ways to find the representative or median trajectories for a group or cluster of trajectories in [6].

4 Formulation Overview and Preprocessing

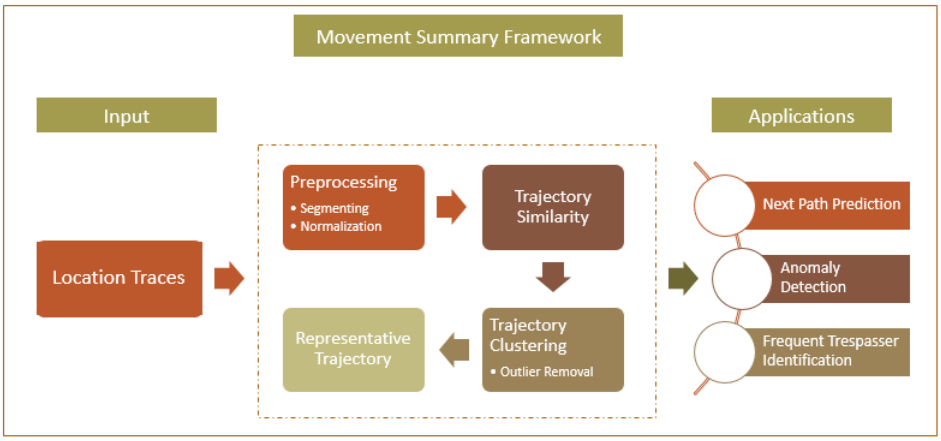


Figure 1: Movement Summary Flow

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. We now describe the components of the framework in detail

We first identify meaningful trips of a user using trajectory segmentation approaches [37]. Here we compute the distance, velocity and time-gap between consecutive set of sample points to identify if the user is mobile. We use the well-known representation of trajectory to denote a meaningful trip; it is an ordered set of 3-tuples $\langle \text{latitude}, \text{longitude}, \text{time} \rangle$.

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

$$\text{lat}_i = \frac{\text{rlat}_i - \min_{\text{lat}}}{\max_{\text{lat}} - \min_{\text{lat}}} \tag{1}$$

Sim Measure	Is Metric	Type	Sen. to sample noise	OD Cognizant	Computational Cost
LP Norm	Yes	Sampling Sensitive	No	No	O(N)
DTW/LCSS/EDW/EDW With real sequences	No	Sampling Sensitive	Yes	No	O(n^2)
LP Norm with Interpolation	Yes	Shape Sensitive	No	No	O(Num samples)
ODSim (Ours)	Yes	Shape sensitive	No	No	O(Num samples)

Table 1: Taxonomy of Similarity Measures

where $\min_{\text{lat}} = \min(\text{rlat}_j, \forall j)$ is the minimum latitude observed, and $\max_{\text{lat}} = \max(\text{rlat}_j, \forall j)$. We similarly normalize raw longitude and time values into normalized longitude and time values.

5 Inter-Trajectory Distance

Computing the distance between a pair of trajectories is crucial to Trip Compactionbecause of higher level clustering algorithms require such a distance measure. A large number of trajectory similarity metrics, which give an estimate of the trajectory distance, have been proposed for various types of applications. Standard LP-Norm and ERP have been shown to be metrics [7], where as a large number of other heuristic measures (such as LCSS, DTW, EDR) are non-metrics [27, 30, 8]. A taxonomy of trajectory similarity metrics is discussed in Table 1.

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 [32, 36]. Hence, we use modified standard LP-Norm functions for distance computation, and avoid time consuming non-metric algorithms.

5.1 Curve Distance for Trajectories

Instead of treating trajectory as a set of sample points, we approximate the trajectories as curves on n-dimensional vector space. This approximation is reasonable if the location traces contains finely sampled GPS points (as in our case)

For trajectory representation, the main idea is to represent trajectory as a curve in two independent dimensions latitude and longitude (in $\mathbb{R}^2$) [18]. A natural extension is to represent in three independent dimensions including time. However, in this paper, we propose to look at latitude and longitude dimensions

Let $f_i[0, 1] \rightarrow \mathbb{R}^2$ 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

$\|f_i\| = \left[\int_0^1 f_i(x)^2 \, \mathrm{d}x\right]^{\frac{1}{2}}$.

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

Numerically, we solve this by first resampling the trajectory at 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). We now propose a metric 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

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

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.

6 Trajectory Clustering

We use the distance metrics to aggregate similar trajectories into one cluster. We use hierarchical agglomerative clustering 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 us 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.

We use "average" link clustering 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).

6.1 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. We went against determining a static similarity level to cut dendrogram 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; we demonstrate this in Section . Hence, we design an algorithm 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**

$$SSW(Clus_i) = \sum_{j=1}^{|Trajs(Clus_i)|} Sim(Traj_j, Mean(Clus_i)) \tag{4}$$

$$SSW(Level_k) = \sum_{j=1}^{k} SSW(Cluster_j) \tag{5}$$

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)

We finally represent the "Mobility Summary" of a user as the clusters with number of trajectories greater than a certain threshold (5% of user's trajectories)

6.2 Representative trajectories

We now find a representative trajectory for each of the cluster that can be used as a proxy for a summary cluster. One approach is to consider the mean trajectory curve $\bar{t}_i$ as a representative for the cluster S_i. However, since the mean of multiple curves may not fall on any of the individual trajectories (and hence the roads on which the user went), it is not recommended as a representative. Hence, we follow a well-known method of computing piece-wise median trajectories [6]. Here, at an interval of δ number of points on interpolated mean trajectory, one point on any of the trajectories in the cluster closest to the mean is added to the representative trajectory.

7 Evaluation and Analysis

7.1 Dataset

- Microsoft GeoLife Dataset GeoLife Dataset published by Microsoft Research [41],[38],[40]. This is a GPS trajectory dataset with GPS traces of 182 users over a period of three years (from April 2007 to August 2012).
- Microsoft T Drive Taxicab Dataset This is a trajectory dataset that contains one-week trajectories of 10,357 taxis published by Microsoft Research. [31] ,[33]

7.2 Clustering Effectiveness

We have shown various comparisons and result, but the main measure of clustering effectiveness that we use 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

$$s(x) = \frac{b(x) - a(x)}{max(a(x), b(x))} \tag{6}$$

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

$$SC = \frac{1}{N} \sum_{i=1}^{N} s(x) \tag{7}$$

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

7.3 Individual Movement Summary

In this section, we talk about the experiments made on the Microsoft GeoLife Dataset. These are GPS traces of around 182 users collected over a period of three years. From this data, we aim at finding the movement summary of the person which will give us insights into how that person moves and which patters appear repeatedly. In the first section we show the working of our proposed method with supporting visuals at every step explaining the rationale behind it. Further, we implemented and modified some other works from the literature to suit the problem and have discussed the results. A brief description of the methods we have compared with are given below.

Dynamic Time Warping This is a comparison made in the choice of the similarity metric that we use. We show the effects of using DTW in place of our similarity measure, keeping everything else in the algorithm exactly as it is.The DTW similarity between A and B is defined below

$$D_{dtw}(A,B) = \begin{cases} 0 & \text{if both A and B are empty} \\ \infty & \text{if one of A or B is empty} \\ \phi_d(head(A), head(B)) + & \\ \quad min \begin{cases} D_{dtw}(A, rest(B)), \leftarrow \textit{Stretch A} \\ D_{dtw}(rest(A), B), \leftarrow \textit{Stretch B} \\ D_{dtw}(rest(A), rest(B)) \end{cases} & \\ otherwise & \end{cases} \tag{8}$$

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

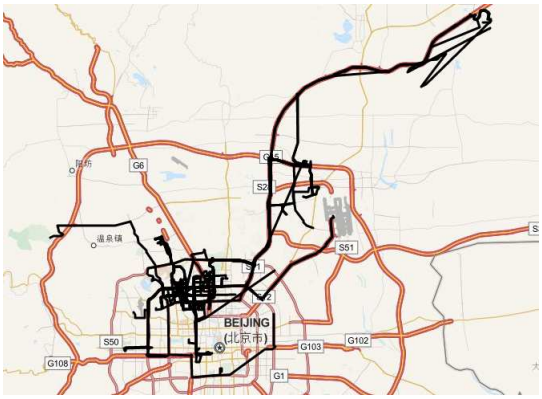


Figure 2: A snapshot of all the trajectories of the test user (363 trajectories in total)

SWARM SWARM is a moving objects clustering algorithm proposed by Li *et al.*[21] In this algorithm they consider trajectories to be of the same cluster if they are together in similar clusters for a specific number of (not necessarily consecutive) timestamps.

TRACLUS TRACLUS is an algorithm proposed by Lee *et al.* in [19]. This algorithm partitions the trajectories, clusters the partitions and then comes up with a final representative cluster.

7.3.1 Proposed Method-OD

In this section we show each stage of the proposed method and the corresponding visualizations at that stage for a test user. Fig. 2 shows all the trajectories of the user. We compute the similarity matrix using the similarity defined earlier and run hierarchical clustering 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 4 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 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 3 shows the elbow curve with two lines indicating the elbow point, and the optimal cluster number. We see that the

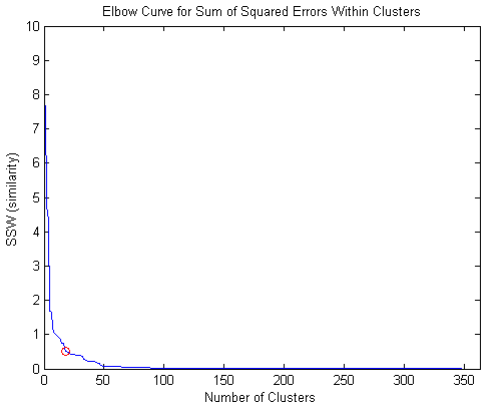


Figure 3: Elbow curve - Plot of the SSW vs number of Clusters; Elbow point 18 clusters

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 and (elbow point +5) number of clusters in Fig. 6.

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

7.3.2 Variation with DBSCAN

In order to make an attempt at doing away with the heuristic, we tried another approach. In this approach we perform two stages of clustering followed by a DBSCAN based on the OD similarity measure we proposed. This method is described below :

- In the first stage, we cluster the trajectories based on the Origin to origin distance. We run hierarchical clustering, and then plot the elbow curve for the SSW values and cluster it at the elbow point.
- In the next stage, we further cluster each cluster obtained from above based on the destination distances and performing hierarchical clustering individually on each of the clusters.
- Once we have performed the origin and destination clusters, we now have trajectories clustered on their OD values. Now, on each of these individual clusters, we run DBSCAN based on the similarity proposed earlier. The values for minLns and epsilon are chosen using the heuristic mentioned in the paper .

By using this variant, we dont use the heuristic at any step, and are still successful in getting optimum results. The visualizations of the top 4 clusters by using this method is shown in Fig. 8

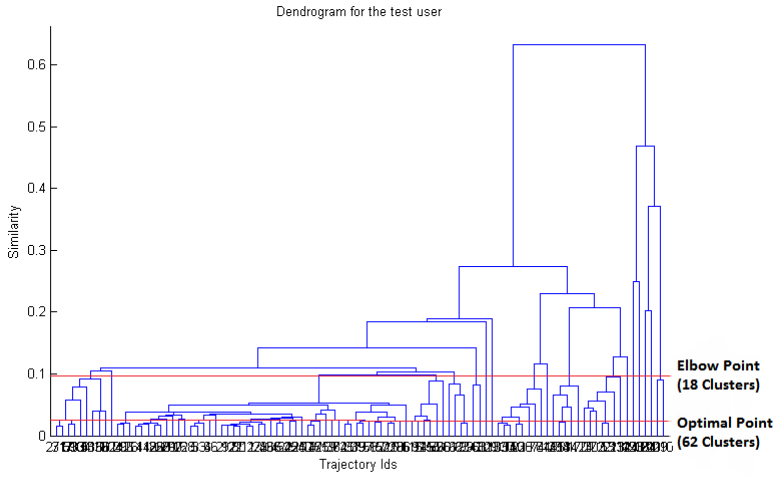


Figure 4: Dendrogram of the all the trajectories (Shows the hierarchy of similarity) The elbow point is at 18 clusters, whereas the optimal number of clusters is 62

(a) Cluster 1 (32 trajectories)

(b) Cluster 2(31 trajectories)

(c) Cluster 3(31 trajectories)

(d) Cluster 4(31 trajectories)

Figure 5: Visualizations of the top 4 clusters at Elbow Point

(a) Cluster 1 (31 trajectories)

(b) Cluster 2(31 trajectories)

(c) Cluster 3(24 trajectories)

(d) Cluster 4(20 trajectories)

Figure 6: Visualizations of the top 4 clusters at Elbow Point+5 point- Clusters not tight enough,need to go down the dendrogram further

(a) Cluster 1 (20 trajectories)

(b) Cluster 2(15 trajectories)

(c) Cluster 3(11 trajectories)

(d) Cluster 4(9 trajectories)

Figure 7: Visualizations of the top 4 final optimal clusters

(a) Cluster 1 (19 trajectories)

(b) Cluster 2(16 trajectories)

(c) Cluster 3(11 trajectories)

(d) Cluster 4(9 trajectories)

Figure 8: Visualizations of the top 4 final optimal clusters using the DBSCAN variation

7.3.3 Comparisons with DTW

The first comparison is made using Dynamic Time Warping as the similarity metric, and clustering based on that matrix.WE use the same framework as that proposed in our method, the only difference being that we plug in DTW similarity in place of our OD based similarity defined earlier. DTW is very close to our 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, our 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.

Problems 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). Figure 9 shows an example of the violation of triangle inequality using DTW similarity.
- 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 resampled and DTW similarity is computed, it would reduce to the same as pointwise Eucledian distance, and would still be computationally more expensive. Fig 10 shows the computation time differences over all the users for clustering using DTW and OD similarity measures over all the

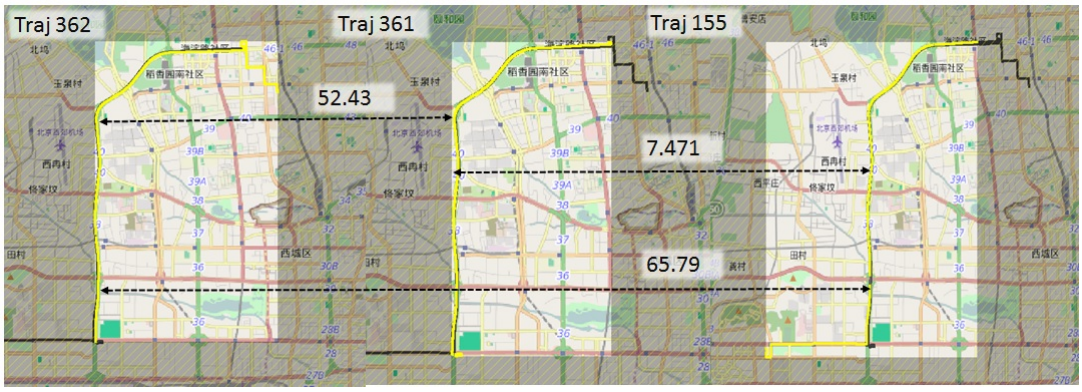


Figure 9: Example of Triangle Inequality Violation using DTW

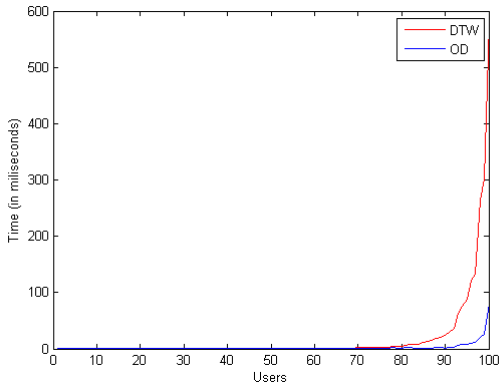


Figure 10: Computation time comparison of DTW and OD

users

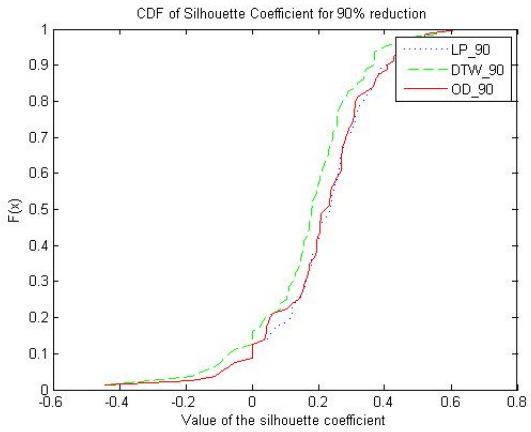
- As we decrease the number of sample points, the effectiveness of both our proposed method and DTW go down. But the goodness of the clusters returned by DTW decreases more than that of the proposed method. We reduced the number of sample points in each of the trajectories to 90%,95%, and 97% and plotted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

7.3.4 Comparison with SWARM

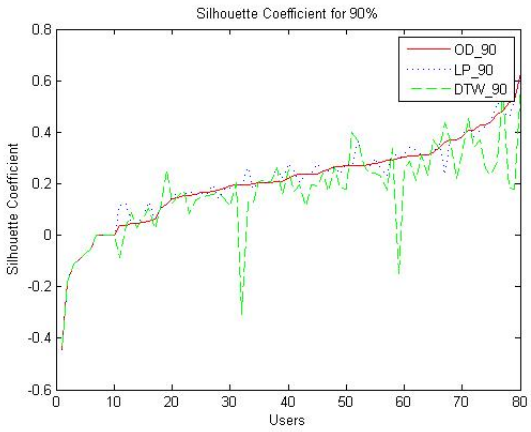
The major problems with SWARM are as follows:

- Does not report all the big clusters
- Dependent a lot on the sample points. If we resample, it would be computationally more expensive than our approach.

We show that our method performs better than SWARM over all the users using the values of the Silhouette Coefficient of the resulting clusters in Fig 14a. We also show the CDF of the SSW of the resulting clusters of SWARM vs our method in Fig 14b.

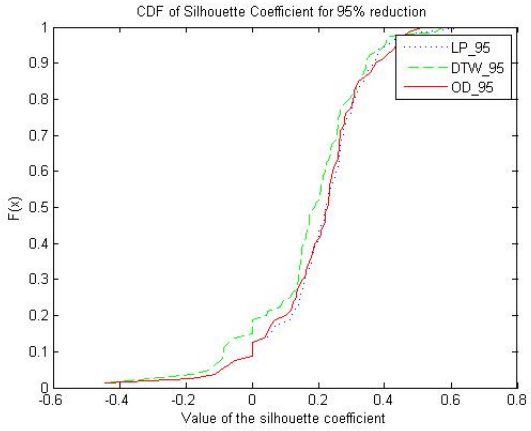


(a) CDF of the silhouette coefficient for 90% reduction of sample points

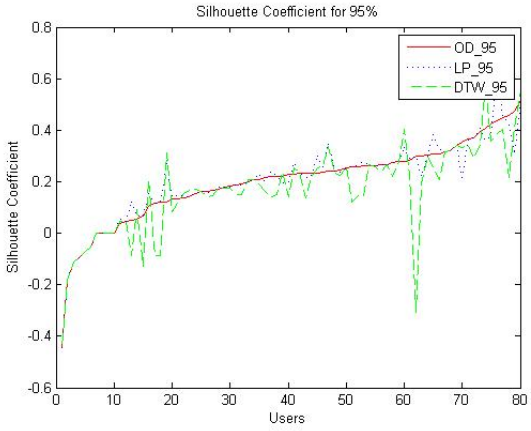


(b) Plot of the silhouette coefficient for 90% reduction of sample points

Figure 11: 90% reduction in sample points- Comparison of silhouette graphs

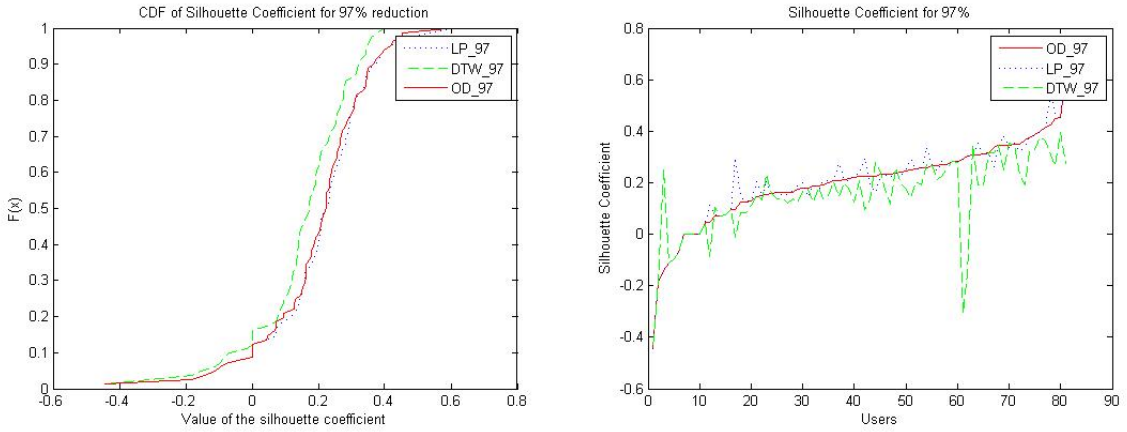


(a) CDF of the silhouette coefficient for 95% reduction of sample points



(b) Plot of the silhouette coefficient for 95% reduction of sample points

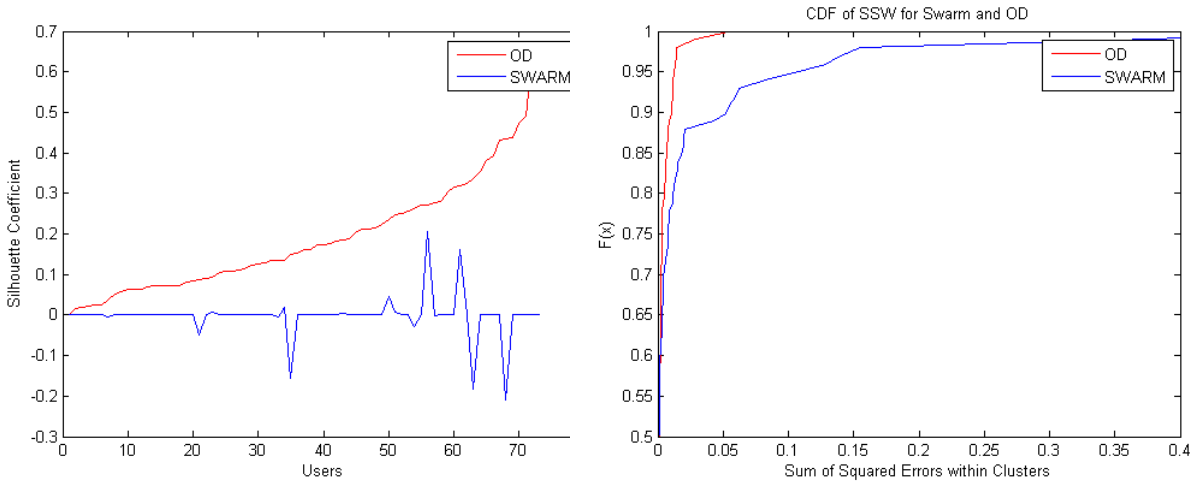
Figure 12: 95% reduction in sample points- Comparison of silhouette graphs



(a) CDF of the silhouette coefficient for 97% reduction of sample points

(b) Plot of the silhouette coefficient for 97% reduction of sample points

Figure 13: 97% reduction in sample points- Comparison of silhouette graphs

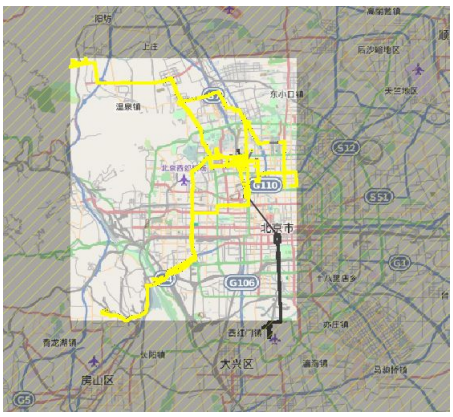


(a) Silhouette Values of the resulting clusters for SWARM and OD- We perform significantly better overand OD- Our SSW error is lesser than SWARM over all all users users

(b) SSW Values of the resulting clusters for SWARM

Figure 14: SWARM comparisons

(a) All the trajectories of an example user



(b) Cluster reported by TraClus

Figure 15: TraClus Final cluster reported

7.3.5 TraClus

TraClus looks at the sub-trajectory level, and clusters trajectories on the basis of the similarities between those sub-trajectories. In the first phase, it partitions the trajectories into segments, and in the second phase, it clusters the partitions together using a DBSCAN-like technique.The problems with TrajClus are

- The similarity measure between any two partitions is defined as a weighted sum of the perpendicular distance, parallel distance and the angular distance between the partitions. Here, three quantities with different units are being merged together, so when two partitions are similar, it is difficult to say which distance contributed to the similarity. Also, this creates a problem in arriving at the neighbourhood parameters.
- There are two parameters used in this algorithm, *epsilon* and *minLns*. *epsilon* defines the neighourhood reach of each of the partition, and minLns is the minimum number of Partitions required in the neighbourhood for it to be considered as a cluster. The authors suggest a simulated annealing technique to arrive at the value of epsilon , and from that value, further calculate the value of minLns. But, because the algorithm is highly associative, nearly all the partitions end up in one cluster, thus not identifying the correct movement summaries.
- TrajClus does not give enough weightage to the direction of the line segment. In cases like animal movement or hurricane movement, this makes sense, because there wont be many cases of trajectories in different directions in a flock or cluster. But when it comes to human movement pattern, directions play a very important role in determining the movement summaries of a person. TrajClus overlooks it and clusters two trajectories in different directions in the same cluster.

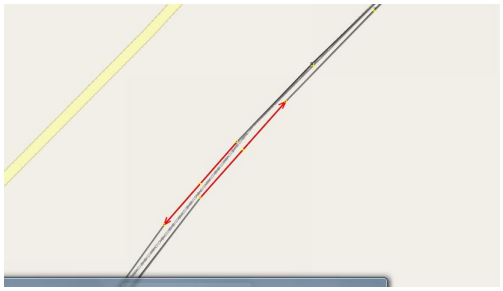


Figure 16: Trajectories with different directions clustered together by TrajClus

7.3.6 Next Location Prediction

Another way to test the summarization of the movement patterns is to test a query trajectory and plot its predicted next location/destination as predicted by all the methods.

The next location prediction is done by the following algorithm:

Explanation of the algo For any query trajectory, resample, and compute similarity with the median trajectories of all summary clusters. Report the one with the maximum similarity.

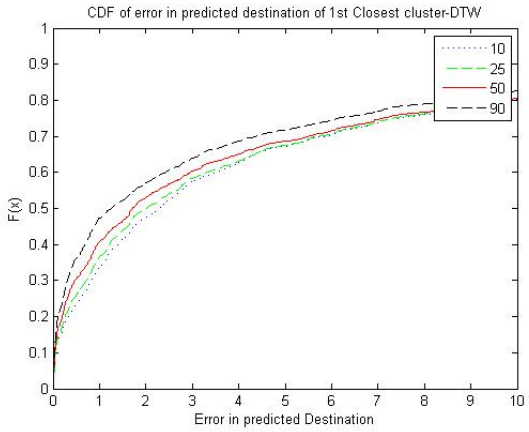
Let $g(i)$ be the probability of the summary i. Given an input traj t_{in}, compute the distance (in meters or so) $d(i, t_{\text{in}})$ between summary i and t_{in}. Now the probability that this sub-trajectory lies within summary i is given by

$$p(i, t_{\text{in}}) = \frac{1}{\sqrt{2\pi}\sigma_t} \mathrm{e}^{-0.5\left(\frac{d(i,t_{\text{in}})}{\sigma_t}\right)} \tag{9}$$

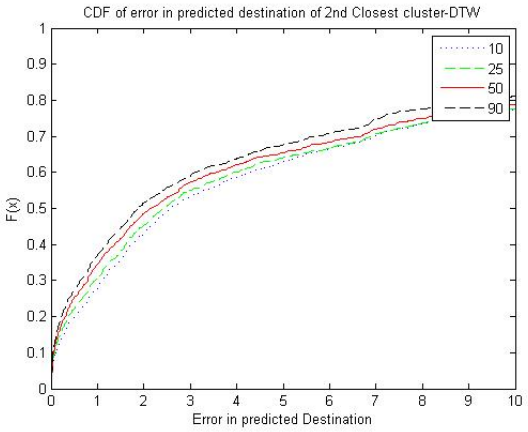
Here we assume that the input trajectory is a noisy input from GPS samples. σ_t is the standard deviation of the sub-trajectory distance. For now take, $\sigma_t = \sigma_p$, where σ_p is the standard deviation of the GPS sampling a location (value is 15.61, which is the 95-th percentile of GPS considering 30 m error). It should ideally be standard deviation introduced when we compute distance between 100 points of a path

For each of the methods compared, we plot the CDF of the error of the predicted destination for the Top-3 Closest clusters to each of the query trajectories. Each of the graphs contain 4 plots, each one varying the number of sample points given to the query trajectory. We have plotted the errors for query trajectories with 10%, 25 %, 50% and 90% of the sample points.

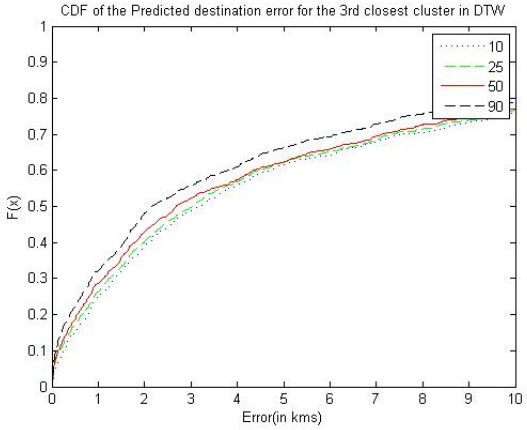
Some of the observations from the next location prediction results are as follows:
Our Approach and LP predict the destination of *82%* of the trajectories with less than 10km error for closest cluster. DTW also is very close with prediction *80%* of the trajectories, but SWARM fares very badly with just *5%* of the trajectories' destination predicted. SWARM cannot detect the third closest cluster for any of the input trajectories.



(a) CDF of error in predicted destination for 1st closest cluster using DTW

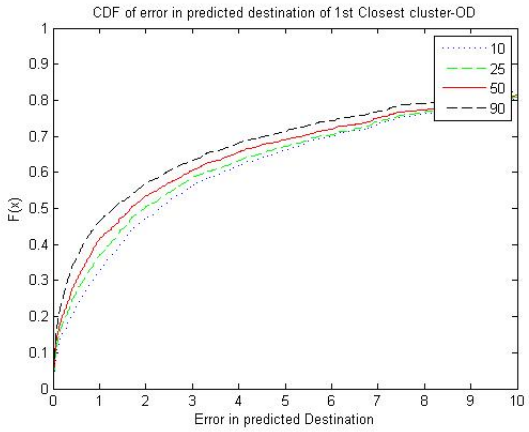


(b) CDF of error in predicted destination for 2nd closest cluster using DTW

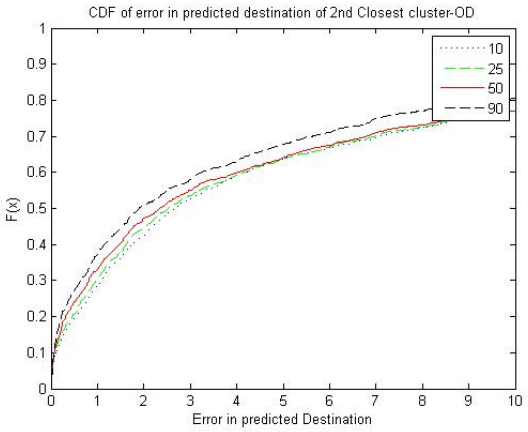


(c) CDF of error in predicted destination for 3rd closest cluster using DTW

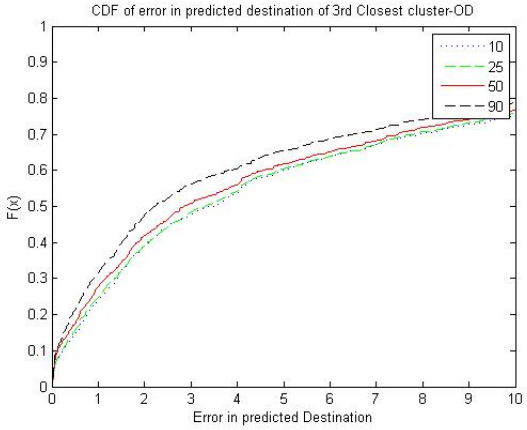
Figure 17: CDF of the error in predicted destinations for the top 1,2,3 closest clusters for *DTW*



(a) CDF of error in predicted destination for 1st closest cluster using OD

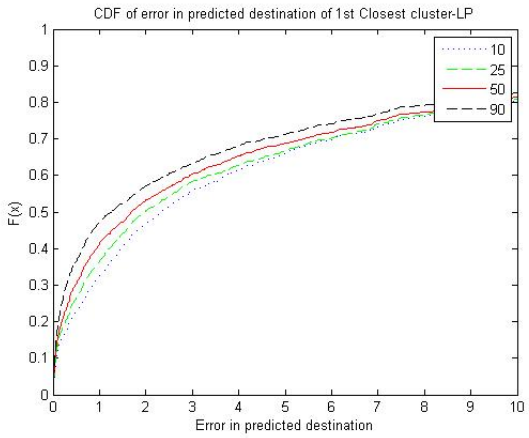


(b) CDF of error in predicted destination for 2nd closest cluster using OD

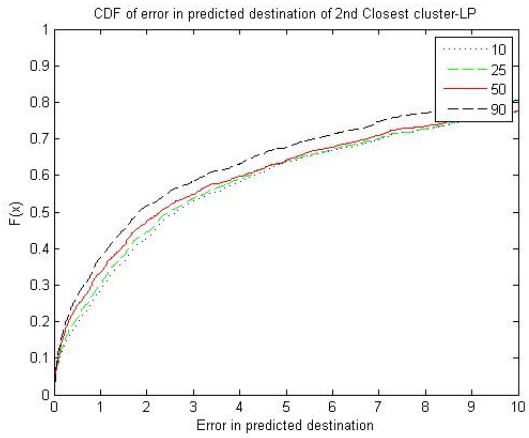


(c) CDF of error in predicted destination for 3rd closest cluster using OD

Figure 18: CDF of the error in predicted destinations for the top 1,2,3 closest clusters for *OD*

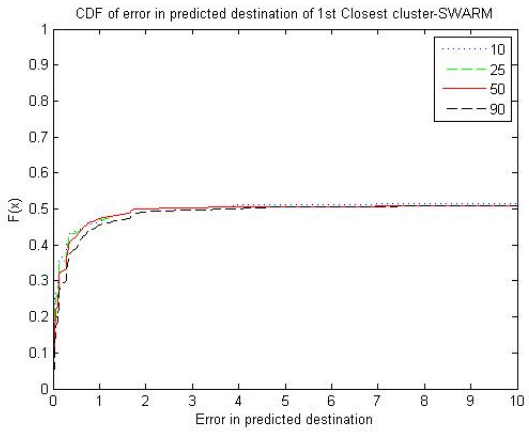


(a) CDF of error in predicted destination for 1st closest cluster using LP

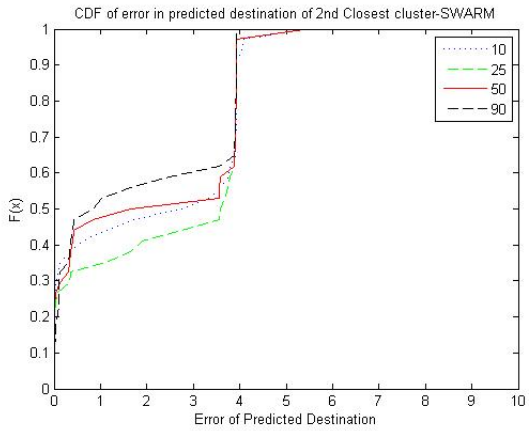


(b) CDF of error in predicted destination for 2nd closest cluster using LP

Figure 19: CDF of the error in predicted destinations for the top 1,2 closest clusters for *LP*



(a) CDF of error in predicted destination for 1st closest cluster using SWARM



(b) CDF of error in predicted destination for 2nd closest cluster using SWARM

Figure 20: CDF of the error in predicted destinations for the top 1,2 closest clusters for *SWARM*

References

[1] Waze. https://www.waze.com/.

[2] 3GPP TS 25.413. UTRAN Iu interface RANAP signalling. http://www.3gpp.org/DynaReport/25413.htm.

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

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] T. M. T. Do and D. Gatica-Perez. The places of our lives: Visiting patterns and automatic labeling from longitudinal smartphone data. *Mobile Computing, IEEE Transactions on*, 13(3):638–648, March 2014.

[10] M. K. El Mahrsi and F. Rossi. Graph-based approaches to clustering network-constrained trajectory data. In *NFMCP*, pages 124–137. Springer, 2012.

[11] C. Faloutsos and K.-I. Lin. *FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets*, volume 24. ACM, 1995.

[12] 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.

[13] A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 1999.

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

[15] 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.

[16] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen. Discovery of convoys in trajectory databases. *Proceedings of the VLDB Endowment*, 1(1):1068–1080, 2008.

[17] T. Kahveci and A. Singh. Variable length queries for time series data. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 273–282. IEEE, 2001.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

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

[24] A. J. Richardson, E. S. Ampt, and A. H. Meyburg. *Survey Methods for Transport Planning*. Eucalyptus Press, 1995.

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

[26] 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.

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

[28] 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.

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

[30] 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.

[31] J. Yuan, Y. Zheng, X. Xie, and G. Sun. Driving with knowledge from the physical world. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 316–324. ACM, 2011.

[32] 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.

[33] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, pages 99–108. ACM, 2010.

[34] 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.

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

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.