Analysis of distance measures in spatial trajectory data clustering

Sharmila S 1,2 and Sabarish B A 1,2
1Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore
2Amrita Vishwa Vidyapeetham, India
E-mail: cb.en.p2cse19022@cb.students.amrita.edu (Sharmila S)
E-mail: ba_sabarish@cb.amrita.edu (Sabarish B A)

Abstract. Tremendous growth of Location-based technologies resulted in the generation of a huge volume of spatial data, which needs to be analyzed to get potentially important patterns. The spatial patterns extracted can be used to design a better infrastructure ensuring reliable service coverage. Trajectory data is one variant of spatial data that are generated by moving objects travelling across. It is represented as a sequence of spatial coordinates (latitude, longitude) of a location. Trajectory clustering tries to group similar spatial data points to extract the most common movement behaviors. Trajectory data poses major challenges including uncertainty, sampling rate, representation, relationships, spatial autocorrelation, serialization, redundancy, and triviality, which makes it hard to apply traditional clustering algorithms over trajectory data. In this paper, K-Means and DBSCAN (Density-based spatial clustering of applications with noise) clustering algorithms are analyzed using different similarity measures like Euclidean, Hausdorff and Haversine distances with the help of index measures say Adjusted Rand Index (ARI) and Fowlkes-Mallows scores (FMS). Experiment is carried out over two different trajectory datasets and it is proved that usage of Haversine distance for clustering is efficient than Euclidean and Hausdorff distances in terms of spatial trajectory data.

1. Introduction
Due to the rapid technological advancement in various fields using location-based information, data accumulated from multiple resources and services like military, banking, bio-informatics, transportation (e.g., taxi GPS data, logistics tracking), retail, health care units etc. The data collected in huge amount has to be processed to extract useful knowledge or behaviour patterns, which is used to mend the productivity with regard to cost and efficiency. Clustering groups objects together based on similarity to gain knowledge. [1-4]. The process of data mining and clustering of trajectory data is difficult when compared to the clustering of numerical or categorical data by traditional algorithms. Due to the factors like, relationships, spatial autocorrelation, serialization, redundancy and triviality, traditional algorithms fail to make optimum trajectory clusters [5]. In data collection process, data collected are usually of high-dimensions. Due to high dimensional feature space, data becomes sparse and clustering of data into different groups becomes harder to separate one cluster data from others. Because of high-dimensional data, search space and complexity of model increases, dimensional reduction techniques becomes as integral part in trajectory clustering [6-9].The following paper is organised with related works in Section 2, datasets description in Section 3, Analysis of different distance measures on two different trajectory datasets is mentioned with its results in Section 4, the paper is concluded and future scope is mentioned in Section 5 and References.

2. Related Work
As there exists a huge variety of data, the different types of clustering algorithms should be analysed properly before usage. Major groups into which multiple clustering algorithms are aggregated based on: Partition, Density, Hierarchical, Model and Grid [10]. Partition-based or hard clustering algorithm
is K-Means, used in automatic text data categorization and external document plagiarism detection [11, 12]. Based on the type of data like Text, Multimedia, Stream, Uncertain, Time Series, different types of clustering algorithms should be analysed before building a model, otherwise, it will result in loss of efficiency [13]. Near Point with Index Ratio (NPIR) algorithm [14] deals with the searching of a nearest neighbour data point by not considering all the data points at once like K-Nearest neighbour (KNN) algorithm, instead by considering only few neighbour points based on election process using distance vector. DBSCAN clustering is dependent on the type of data. Multiple steps of pre-processing should be made before using DBSCAN on spatial-temporal data. To overcome these multiple pre-processing steps, ST-DBSCAN was proposed in [15]. Due to the reduced number of steps of processing, the efficiency of ST-DBSCAN is high when compared to DBSCAN. Data point may tend to be a member of different clusters based on the type of data. So, a comparative study between soft and hard clustering algorithms are done [16] and concluded that for a generic dataset like iris data [17], it is proved that K-NN outperforms Fuzzy C-Means (FCM) in terms of efficiency as it incorporated huge fuzzy steps. This study proves that the selection of a clustering algorithm based on data type is trivial.

Choosing perfect Intra-cluster and Inter-Cluster similarity measure and Rand Index for analysing the quality of clusters is considerable. The analysis of multiple such similarity measures and the Rand Index are made with varying dimensions of data as inputs [18]. This study comes out with major findings like the Pearson correlation coefficient is not suitable for datasets which are low dimensional and centroid dependent, instead, it is suggested to be used on high-dimensional data which uses hierarchical approaches. Finally, the same study reveals an important finding of doing ANOVA (Analysis of Variance) test for proving the correctness of the similarity measure used in a model. The selection of similarity measure should also be compatible with data types. [19,20] used semantic similarity measures for documents to detect plagiarism. The common distance measures like Euclidean, Manhattan, Cosine distances demands the same length of trajectories. Since trajectory length varies, Haversian and Hausdorff distance measures are studied against trajectory data [21]. Hierarchical trajectory clustering algorithm is proposed in [22] uses Dynamic Time Warping (DTW) to identify the resemblance between trajectories. Multiple trajectory mining algorithms for future location prediction like HMM (Hidden Markov Model), String matching algorithms, Semantic-based prediction methods, Pattern matching algorithm-based prediction methods are analysed in [23]. The Index measures are used for cluster validation as it measures the agreement between two partitions. Adjusted rand index and Fowlkes-Mallows score are verified in [24,25] and concluded that the measures are suitable for measuring the quality of trajectory clustering.

The trajectory data is collected every single second and the amount of data and its dimensionality is huge. The general dimensionality reduction techniques like PCA, FA, LDA are not relevant in this case. Sampling of data should be done to reduce its size by retaining its feature and underlying information. Vertex sub-sampling algorithms which are used to select certain vertices in a line or trajectory are Douglas-Peucker, Visvalingam–Whyatt, Reumann–Witkam, Opheim simplification algorithms [26-29]. These algorithms aim to reduce a curve or line with huge data points to fewer data points without affecting its shape. These algorithms are used for data reduction in many data-type specific applications. In particular, the Douglas-Peucker algorithm is applied on ship trajectory data in [30] to reduce trajectory data and proposed a modified DBSCAN algorithm for marine traffic pattern recognition.

3. Datasets
TN291 and Microsoft T-Drive datasets are used in this analysis

3.1 TN291
It contains the Latitude and Longitude of a locality as data points. It is collected from 9 different districts of Tamil Nadu, India and contains about 2910 data points of 291 trajectories. It is already pre-processed using the Douglas-Peucker algorithm.
3.2 Microsoft T-Drive [31]
It is collected by taxis in Beijing, China for Research purposes. It contains the Latitude and Longitude of a locality as data points along with taxi id and time stamp. One-week trajectories of 10,357 taxis are collected. The data set contains about 15 million data points and it covers the distance of about 9 million kilometres.

4. Analysis of Distance Measures on Trajectory clustering
The popular clustering algorithms K-Means and DBSCAN will be applied on two trajectory datasets with Euclidean, Hausdorff and Haversine distance measures. The clustering validity is calculated with the Adjusted Rand Index (ARI) and Fowlkes-Mallows scores (FMS).

4.1 Data Reduction using Ramer-Douglas-Peucker (RDP) algorithm
TN291 is already pre-processed to have only 2910 data points. First 500 taxi data from Microsoft T-Drive trajectory dataset is chosen and RDP is applied with $\varepsilon = 0.0005$, such that 7,89,543 data points are reduced to 5,86,512 latitude longitude pairs. Figure 1 visualizes the spread of datapoints of both TN291 and T-Drive datasets.

![Figure 1. Latitude and longitude pair of a) TN291 dataset and b) T-Drive dataset.](image)

4.2 Analysis of hyper-parameter for clustering
K-Means clustering algorithm faces a problem of identifying number of clusters initially. Elbow method is used to define optimum $K$ by calculating Within Cluster Sum of Squared Error (WSS) for different values of $K$ and by choosing the $K$ for which WSS value becomes the first start to diminish as in Figure 2.

![Figure 2. Distortion is plotted against different K values for a) TN291 dataset and b) T-Drive dataset](image)
It is clear from the above Fig 2(a) that, optimum number of centres (i.e) K value for TN291 dataset is either 3 or 4 and from Fig 2(b), optimum K value is either 2 or 3. In this case, further analysis is done with K=3 for both TN291 and T-Drive dataset.

**K-means clustering algorithm**

| STEP 1: Initialization | K initial centroids are generated in random. |
|------------------------|------------------------------------------|
| STEP 2: Assignment     | Distance between data points and centroid will be calculated and K clusters are formed by associating each data points with the nearest centroid. |
| STEP 3: Update         | The cluster centroid are updated with the mean of each cluster data points. |
| STEP 4: Steps 2 to 3   | will be repeated until the new assignment is equal to the previous assignment. |

**DBSCAN clustering algorithm**

| STEP 1: Initially the algorithm starts with selection of data point (not visited) |
|--------------------------------|-------------------------------------------------|
| STEP 2: Distance between points is calculated and the neighbourhood of a data point is extracted using epsilon (\(\varepsilon\)). |
| STEP 3: If there exist minPoints of neighbour points around a point, then clustering starts by marking that point as visited, else it will be marked as noise (temporarily, until it gets allotted to a cluster). |
| STEP 4: If a point is noted to be a part of a cluster, then its \(\varepsilon\) neighbourhood points is also a part of the cluster. Steps 2 to 3 will be repeated until all the points in a cluster is determined. |
| STEP 5: New unvisited data point will get retrieved and processed, resulting in the discovery of cluster point or noise. |
| STEP 6: Steps 2 to 5 will get repeated until all the data points are marked as visited. |

K- Means and DBSCAN with different distance measures like Euclidean, Hausdorff and Haversine are implemented and analysed as follows.

**4.3 Euclidean distance**

Let \(D_i\) represents the spatial co-ordinates, where \(D_1\) is Latitude and \(D_2\) is the Longitude of a locality. Consider the randomly generated centroid as \(C_i\) where \(C_1\) is centroid latitude and \(C_2\) is centroid longitude. The general formula for finding Euclidean distance \(d(D, C)\) between latitude longitude pair and centroid is represented in Equation 1

\[
d(D, C) = \sqrt{\sum_{i=1}^{2} (C_i - D_i)^2} \quad (1)
\]

**4.3.1 K-means clustering using Euclidean distance**

| STEP 1: Generate random centroids |
|--------------------------------|---------------------------------|
| STEP 2: Assignment and update stages are repeated until the cluster reassignment is not possible using the Euclidean distance measure. Figure 4 represents final clusters visually differentiated with different colours. |
Figure 3. represents the initial indication of randomly generated K Centroids in a) TN291, b) T-Drive

Figure 4. Final clusters assignment a) TN291 dataset with 3 clusters and b) T-Drive dataset with 3 clusters

4.3.2 DBSCAN clustering using Euclidean distance

STEP 1: Optimal epsilon (ε) value can be found using K-Distant Graph. Figure 5 represents the K-Distant graph on TN291 dataset, the optimum Epsilon value where the maximum curvature resides is at 0.05. So, from this graph ε = 0.05 for TN291 dataset
STEP 2: There exists no standard method to choose optimum minPoints. So, minPoints=40 is chosen using the trial and error method. Figure 6 represents final clustered data points using the DBSCAN algorithm with $\varepsilon = 0.05$ and minPoints = 40.

Figure 6. Clustered using DBSCAN on a) TN291 dataset with 13 clusters and b) T-Drive dataset with 4 clusters

4.4 Hausdorff distance
The Hausdorff distance calculates dissimilarity amongst two different groups of data points. In other words, it returns the longest distance from one group of points closest to the next group. Let $D_i$ represents the latitude, longitude pair of a locality. Consider centroid as $C_i$. The general formula to find Hausdorff distance $H(D,C)$ between spatial co-ordinates and centroid is represented in Equation 2

$$H(D, C) = H(C, D) = \max \{ h(D, C), h(C, D) \} \quad (2)$$

where $h(C, D) = \max_{c \in C} \{ \min_{d \in D} \{ d(c, d) \} \}$, $h(D, C) = \max_{d \in D} \{ \min_{c \in C} \{ d(d, c) \} \}$ and $d(d,c)$ and $d(c,d)$ is distance between $d,c$ and $c,d$ using any distance measure.
Figure 7. represents Clustering using K-Means and Hausdorff distance measure on a) TN291 dataset with 3 clusters and b) T-Drive dataset with 3 clusters.

4.4.1 DBSCAN clustering using Hausdorff distance
The final clustered partition with DBSCAN and Hausdorff distance measure on both TN291 and T-Drive datasets is mentioned in Figure 8.

Figure 8. represents Clustering using DBSCAN and Hausdorff distance measure on a) TN291 dataset with 11 clusters and b) T-Drive dataset with 6 clusters.

4.5 Haversine distance
The haversine distance is accurate in terms of measuring distance in spherical object say the Earth. The equation to calculate distance \( d(D,C) \) between the data point and a centroid is mentioned in Equation 3

\[
d(D, C) = 2 \gamma \sin^{-1} \sqrt{\sin^2\left(\frac{D_1-C_1}{2}\right) + \cos(D_1)\cos(C_1)\sin^2\left(\frac{D_2-C_2}{2}\right)}
\]  

where \( D_1 \) and \( D_2 \) are data point, \( C_1 \) and \( C_2 \) are centroid. \( \gamma \) is the approximate radius of Earth, \( \gamma = 6371 \) km.
4.5.1 **K-MEANS clustering using Haversine distance**

The final clustered partition on both TN291 and T-Drive datasets with K-Means and Haversine distance measure is mentioned in Figure 9.

![Figure 9](image)

**Figure 9.** represents Clustering using K-Means and Haversine distance measure on a) TN291 dataset with 3 clusters and b) T-Drive dataset with 3 clusters.

4.5.2 **DBSCAN clustering using Haversine distance**

The final clustered partition with DBSCAN and Haversine distance measure on both TN291 and T-Drive datasets is mentioned in Figure 10.

![Figure 10](image)

**Figure 10.** represents Clustering using DBSCAN and Haversine distance measure on a) TN291 dataset with 11 clusters and b) T-Drive dataset with 3 clusters.

4.6 **Performance Analysis and Findings**

Adjusted Rand Index (ARI) and Fowlkes-Mallows scores (FMS) are the popular cluster validity measurement standard as both measures agreement and similarity between two clusters. These measures are used for validation with reference to [18,19], as these two index measures are suitable for trajectory data. Table 1 statistically represents both ARI and FMS values for the applied K-Means and DBSCAN algorithms with different distance measures on two different datasets.
Table 1. ARI and FMS values for K-Means and DBSCAN algorithms with Euclidean, Hausdorff and Haversine distances on TN291 and T-Drive datasets

| Algorithms | Dataset | Euclidean | | | | Hausdorff | | | | Haversine | | |
|------------|---------|-----------|---|---|---|---|---|---|---|---|---|---|---|
| K-Means | TN291 | 0.55 | 0.610 | 0.50227 | 0.61042 | 0.67782 | 0.68313 |
| | T-Drive | 0.545112 | 0.57443 | 0.45019 | 0.42143 | 0.72389 | 0.72626 |
| DBSCAN | TN291 | 0.66553 | 0.27095 | 0.62294 | 0.32520 | 0.74662 | 0.78872 |
| | T-Drive | 0.52419 | 0.43951 | 0.67752 | 0.69067 | 0.82511 | 0.79801 |

From Table 1, the distance measure which gives high values for both ARI and FMS can be concluded as the optimal distance measure for Trajectory data. It can be seen that the Haversine distance reported having comparatively high ARI and FMS values that other distance measures. That is, ARI = 0.67782 and FMS 0.68313 for K-Means algorithm on TN291 dataset, ARI = 0.72389 and FMS = 0.72626 for K-Means algorithm on T-Drive dataset, ARI = 0.74662 and FMS = 0.78872 for DBSCAN on TN291 and ARI =0.82511 and FMS = 0.79801 for DBSCAN on T-Drive dataset.

5. Conclusion and Future Work

The different distance measures like Euclidean, Hausdorff and Haversine distances are applied and validated with K-Means and DBSCAN algorithms on both TN291 and Microsoft T-Drive datasets. Thus, the Haversine distance measure outperforms both Euclidean and Hausdorff distances and it ensures to have better validity and accuracy results with respect to moving objects Trajectory data. The Haversine distance resulted to be optimal because it is specifically for measuring distance on spherical objects say Earth.

Further, it is noted that when K-Means and DBSCAN algorithms are used for clustering Trajectory data, both ARI and FMS values are at most 82.0 which is lesser than even 90. This can be improved with the concept of fuzzy clustering as there exist chances for one data point to belong to multiple clusters in trajectory data and with a better way to predict the optimum number of clusters before building the model. Because, it is obviously seen from TN291 dataset that it contains 9 clusters visibly, but the popular elbow method reports a lesser number of clusters than it actually contains, which may have resulted in lesser ARI and FMS scores. The future work is to propose a fuzzy concept for clustering trajectory data with an improved way to foresee the optimum number of clusters beforehand.

References

[1] Gupta M K and Chandra P International journal of Information Technology 12 1243-1257.
[2] Benabdellah A C, Benghabrit A, Bouhaddou I 2019 Procedia Computer Science 148 291-302.
[3] Rokach L 2009 A survey of Clustering Algorithms, Data Mining and Knowledge Discovery Handbook (Boston, MA:Springer)
[4] Rui Xu and Wunsch D 2005 IEEE Transactions on Neural Networks 16 645-678
[5] Yuan G, Sun P and Zhao J 2017 Artificial Intelligence Review 47 123–144
[6] Dou W, Dai G, Xu C and Zhang Z 2010 Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2010 6321
[7] Sivasankar E, Selvi C and Mala C 2017 Advances in Intelligent Systems and Computing 556
[8] Kavilkrue M and Boonma P 2015 Advances in Intelligent Systems and Computing 361
[9] Assent I 2012 Clustering high dimensional data URL 10.1002/widm.1062; https://doi.org/10.1002/widm.1062
[10] Mehta V, Bawa S and Singh J 2020 Artificial Intelligence Review 53 5995-6023
[11] Vani K and Gupta D 2014 Using K-means cluster based techniques in external plagiarism detection International Conference on Contemporary Computing and Informatics (IC3I) pp 1268-1273.
[12] Unnikrishnan S, Sreelakshmi S and Deepa G 2016 International Journal of Control Theory and Applications 9 7619-7626
[13] Oyelade J 2019 International Conference on Computational Science and Its Applications (ICCSA) pp 71-81

[14] Qaddoura R, Faris H and Aljarah I 2020 International Journal of Machine Learning & Cybernetics 11 675–714 URL 10.1007/s13042-019-01027-z;https://dx.doi.org/10.1007/s13042-019-01027-z

[15] Birant D and Kut A 2007 Data & Knowledge Engineering 60 208-221 URL 10.1016/j.datak.2006.01.013;https://dx.doi.org/10.1016/j.datak.2006.01.013

[16] Bora D J, Gupta D and Kumar A 2014 International Journal of Computer Trends and Technology 10 108-113

[17] Fisher R URL https://archive.ics.uci.edu/ml/datasets/iris

[18] Shirkhorshidi A S, Aghabozorgi S and Wah T Y 2015 PLOS ONE 10 e0144059 URL 10.1371/journal.pone.0144059; https://dx.doi.org/10.1371/journal.pone.0144059

[19] Nedungadi P, Raj H 2014 Lecture Notes in Computational Vision and Biomechanics 28 215-226

[20] Sabarish B A, Karthi R and Gireesh K T 2018 Lecture Notes in Computational Vision and Biomechanics 28 215-226

[21] Sabarish B A, Karthi R and Gireesh K T 2014 A Survey of Location Prediction Using Trajectory Mining, Artificial Intelligence and Evolutionary Algorithms in Engineering Systems Proceedings of ICAEES 2014 vol 1 (Springer) pp 119–127

[22] Lin Y, Huang H, Simon G E and Liu S 2016 Mathematical Biosciences 282 191-203 URL 10.1016/j.mbs.2016.10.008;https://dx.doi.org/10.1016/j.mbs.2016.10.008

[23] Yu Q, Luo Y, Chen C and Chen S 2019 Applied Intelligence 49 2315–2338 URL 10.1007/s10489-018-1385-x;https://dx.doi.org/10.1007/s10489-018-1385-x

[24] Saalfeld A 1999 Topologically Consistent Line Simplification with the Douglas-Peucker Algorithm Cartography and Geographic Information Science 26 7-18 URL 10.1559/15230409782424901;https://dx.doi.org/10.1559/15230409782424901

[25] Visvalingam M and Whyatt J D 1993 Line generalisation by repeated elimination of points The Cartographic Journal 30 46-51 URL 10.1179/caj.1993.30.1.46;https://dx.doi.org/10.1179/caj.1993.30.1.46

[26] Reumann K and Witkam A P Optimizing curve segmentation in computer graphics

[27] Opheim H Smoothing a digitized curve by data reduction methods

[28] Zhao L and Shi G 2019 A trajectory clustering method based on Douglas-Peucker compression and density for marine traffic pattern recognition URL 10.1016/j.oceaneng.2018.12.019;https://dx.doi.org/10.1016/j.oceaneng.2018.12.019

[29] Zheng Y 2011 URL https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/