Measuring the Distance of Moving Objects from Big Trajectory Data

Khaing Phyo Wai [1] *, Nwe Nwe [2]

[1]*University of Computer Studies, Mandalay,*
Patheingyi
Mandalay 1001, Myanmar
khaingphyowai.mm@gmail.com

[2]*University of Computer Studies, Mandalay,*
Patheingyi
Mandalay 1001, Myanmar
nwenwemdy08@gmail.com

Abstract

Location-based services have become important in social networking, mobile applications, advertising, traffic monitoring, and many other domains. The growth of location sensing devices has led to the vast generation of dynamic spatial-temporal data in the form of moving object trajectories which can be characterized as big trajectory data. Big trajectory data enables the opportunities such as analyzing the groups of moving objects. To obtain such facilities, the issue of this work is to find a distance measurement method that respects the geographic distance and the semantic similarity for each trajectory. Measurement of similarity between moving objects is a difficult task because not only their position changes but also their semantic features vary. In this research, a method to measure trajectory similarity based on both geographical features and semantic features of motion is proposed. Finally, the proposed methods are practically evaluated by using real trajectory dataset.

Keywords: Big Trajectory Data, Moving Objects, Geographic Distance, Semantic Similarity.

1. Introduction

Today, there have been many technologies to provide location services such as Global Positioning Systems (GPS), location estimate smart phone sensors, GSM beacons, infrared or ultrasound systems etc. The advances in mobile computing techniques have also generated massive spatial trajectory data, which represent the mobility of a diversity of moving objects, such as people, vehicles, and animals. Consequently, it becomes easier to build large data trajectory tracking of moving objects.

Many techniques have been proposed for processing, managing, and mining trajectory data in the past decade, fostering a broad range of applications. The popularity of these applications in turn requires the systematic research for new computer technologies for the discovery of knowledge from the trajectory data. In these circumstances, exploration of trajectory data has become an ever more important research topic, interesting the attention from many areas, including computer science, sociology and geography. Moreover, spatiotemporal data management and query processing have been an active topic of research in the past decades, covering a wide range of areas.

A trajectory is the path that a moving object follows through space as a function of time. It is a record of the path of a variety of moving objects including human beings, animals, vehicles, and even natural phenomena. People have been recording their movements of the real

* Corresponding author.

world in the form of spatial trajectories, passively and actively, for a long time. A large number of vehicles equipped with GPS such as taxis, buses, ships and planes have appeared in our daily lives. Many taxis in large cities have been equipped with a GPS sensor, which allows them to report a time-stamped location with a certain frequency. These reports formulate a large number of spatial trajectories. Biologists have been gathering the moving trajectories of animals such as tigers and birds, to study the migratory traces of animals, their behavior and their life situations. Meteorologists, ecologists, climatologists and oceanographers are demanding to collect the trajectories of certain natural phenomena, such as hurricanes, tornadoes and ocean currents. These trajectories capture the change of the environment and climate, serving scientists cope with natural disasters and protect the natural environment in which we live.

A trajectory can be described as a trace formed by a moving object on geographic location, usually represented by a series of chronologically ordered points: for example, $p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_n$, where each point is made geospatial coordinates set a time stamp, as p = (x, y, t). Such trajectory information is a database on a large scale, it can be called big data. Big trajectory data consists of a wealth of information about when and where a specific object is moving. It can offer a unique opportunity to discover the physical patterns of users. Therefore, the ever-increasing volumes of big trajectory data needs for new models and computationally efficient algorithms for efficient storage, processing, analysis and visualization of advanced data-intensive systems and applications.[1]

Some moving objects in the real world share the same moving patterns, which is an important feature and can help people solve real problems. The spatial trajectory data record the history of moving objects in geographic space. Intuitively, if two trajectories can be considered as similar to each other, they should satisfy some particular requirements, for example, they should be close enough to each other in the Euclidean space, and further they should have similar directions. Then a challenging problem is: how to measure the closeness?

To find representative paths or common trends shared by different moving objects, it is usually needed to group similar trajectories into clusters. A general clustering approach is to represent a trajectory with a feature vector, denoting the similarity between two trajectories by the distance between their feature vectors. However, it is not easy to generate a feature vector with a uniform length for different trajectories, as different trajectories contain different and complex properties, such as length, shape, sampling rate, number of points, and their orders. In addition, it is difficult to encode the sequential and spatial properties of points in a trajectory into its feature vector.

However, it is difficult to control process and mine trajectory data. There is a number of challenges to mine the large-scale trajectory data. First, there is a nontrivial task to store big volume of data tracks that are rapidly accumulating. Secondly, it is elusive to define the similarity measure for comparing the trajectory, because the trajectories are likely to be built with different sampling techniques or different sampling frequencies. Thirdly, query processing in bulk volume trajectory data is very difficult in terms of space time or complexity.[2] To address these issues, by taking into account an object's trajectory sequence, an approach is proposed to measure the distance between trajectories from a large number of trajectories data by considering both geographic and semantic similarity score. These methods are evaluated with a large-scale real-world GPS dataset. As a result, our method outperforms the related work using the various clustering algorithms.

This paper is organized as follows. Section 1 introduces what trajectory data and trajectory data mining are. Section 2 offers related work about this research. Section 3 covers the compression big trajectory data. Section 4 presents the important components of measuring the geographic distance of trajectories. Section 5 covers the semantic similarity measurement for each trajectory. Section 6 discusses about the calculating the total trajectory distance based on both geographic and semantic score. Next, performance evaluation of the proposed method is discussed in Section 7. Section 8 discusses about the validation of the proposed method. Finally, Section 9 concludes this work.

2. Related Work

People have been registering their movements in the real world as spatial trajectories for a long time. A large number of vehicles equipped with GPS such as taxis, buses, ships and airplanes have appeared in our daily lives. Basically, a time-stamped geographic data can be coordinated every second for a moving object. But these big trajectory data costs a lot of the excessive costs for communication, computing and data storage. In addition, many applications do not really require such a locating accuracy. To solve this problem, compression strategies were proposed to reduce the size of a trajectory without affecting accuracy in its new data representation.[3] Some are the compression that reduces the size of the path after the path has been completely created. The others are, on-line compression, compressing a web immediately like a travel object.

A number of approaches measuring the similarity of the moving object trajectories are cluster-based methods. See Refs, 4 and 5 for more details. A partition-and-group algorithm to cluster similar trajectories is discussed in Ref. 6. It defines three measurements, i.e., perpendicular distance, parallel distance and angle distance as the measurements of the similarity. An approach which mines the similarity of people's trajectories based on location histories is proposed in Ref. 7.

A method of determining the optimal number of clusters has been proposed by dividing the multiple transformations for the purpose of the efficient processing of query against the results of applying the transformations to time series. See Ref. 8 for more details. In this paper, the moving average is used as a transformation for simplicity. The model of query time to the number of clusters is constructed for determining the optimal number of clusters. As the query time could be represented with the concave function of the number of clusters, it is shown that the optimal number of clusters for the best query time can be obtained. The verification experiment confirms the validity of the model constructed. It is revealed that the optimal number of clusters could be determined by the times obtained from a single query execution.

The problem of continuous reverse k nearest neighbors (RkNN) in directed road network is studied in this research[9], where a road segment can have a particular orientation. A RNN query returns a set of data objects that take query point as their nearest neighbor. Although, much research has been done for RNN in Euclidean and undirected network space, very less attention has been paid to directed road network, where network distances are not symmetric. In this paper, pruning rules which are used to minimize the network expansion has been provided while searching for the result of a RNN query. Based on these pruning rules an algorithm named SWIFT for answering RNN queries in continuous directed road network has been proposed.

While the above models find the trajectory similarity based on geographic features, some recent approaches introduce the semantic tags to enhance the accuracy of the measurement.[10] The authors consider trajectories as a set of stops and moves. A recent approach which detects similarity in semantic trajectories is proposed. It defines a stay point as the place where a user stays for a while, and it carries a particular semantic meaning. The approach uses the longest common subsequence (LCSS) algorithm to find the similarity mainly on the semantics. A continuous work adds geographic feature to measure the similarity and make prediction. It introduces two measurements, i.e. Semantic Score and Geographic Score. However, this approach first filters the trajectories by semantic similarity and then detect the geographic similarity, therefore two trajectories which are far from each other might have a very high similarity score, if their semantic similarity is high. The difference of our approach is that we compare geographic similarity at the beginning and then measure the semantic similarity.

A key issue in performance evaluation of tracking results is the distance metric that determines the similarity of the trajectories. Any additional analysis, such as action recognition, event detection, etc., highly depends on the accuracy of the similarity assessment. To address these issue, an approach to measure the similarity between moving object trajectories is proposed in this research. The method is based on both geographic features as well as semantic properties of trajectories. Then we combine both similarity measurements together and define an efficient distance function.

3. Compressing Big Trajectory Data

Recent advances in object tracking made it possible to obtain spatiotemporal motion trajectories for further analysis of concealed information. Trajectory data grow intensively as the time goes by, this is due primarily to the large availability of mobile sensors, such as GPS-enabled smart phones. Basically, a time-stamped geographical coordinate can be recorded every second for a moving object. But, this costs a lot of battery power and the overhead for communication, computing, and data storage. Mining massive trajectories is very time consuming, as it is need to access different samples of the trajectories or different parts of a trajectory many times. In addition, many applications do not really need such a precision of location. Accordingly, there are a huge number of data stored in database, thereby the necessity of trajectory data compression which plays an essential task of big trajectory reconstruction.

The objectives of trajectory compression are to reduce the size of the dataset, and to allow computations with low complexity, and also to support low deviation between the reduced trajectory and the original one, without warping the trend of the trajectory or distorting data. The data compression is a method that reduces the size of data to cut down the memory space and improve the efficiency of transmission, storage, and processing without losing information or reorganizes data to reduce the redundancy and memory space according to certain algorithms. Data compression can be classified into two categories, namely, lossless and lossy compression. Moving object trajectory compression aims to reduce the size and memory space of a trajectory on the premise that the information contained in trajectory data is reserved; that is to say, in order to cut down the size of data, it removes redundant location points while ensuring the accuracy of the trajectory.

Enormous amounts of GPS trajectories, which record users' spatial and temporal information, are collected by geo-positioning mobile phones in recent years. The massive volumes of trajectory data bring about heavy burdens for both network transmission and data storage. For example, if data is collected at 10 second intervals, a calculation shows that without any compression, 100 Mb of storage capacity is required to store just 400 objects for a single day. To overcome these difficulties, a number of compression algorithms have been proposed.

Moving object data are normally available as sample points in the form (t_{id}, x, y, t), where t_{id} is an object identifier and x, y and t are respectively spatial coordinates and a time stamp. To compress big trajectory data, movement status (such as position, direction, and speed etc.) need to be retained. In our study, the developed methodologies, namely KiT is used to reduce the size of big trajectory dataset.[11] KiT algorithm detects key points in a raw trajectory. It denoted a trajectory as: P= {Pt_0, Pt_1, Pt_2, Pt_3...Pt_n} where, Pt_n ={x,y,t_n} is referred as a data point. The output of KiT is to find a reduced subset of P denoted as P' and P'= {Pt_0, Pt_1… Pt_i…Pt_j} 0<i<j.

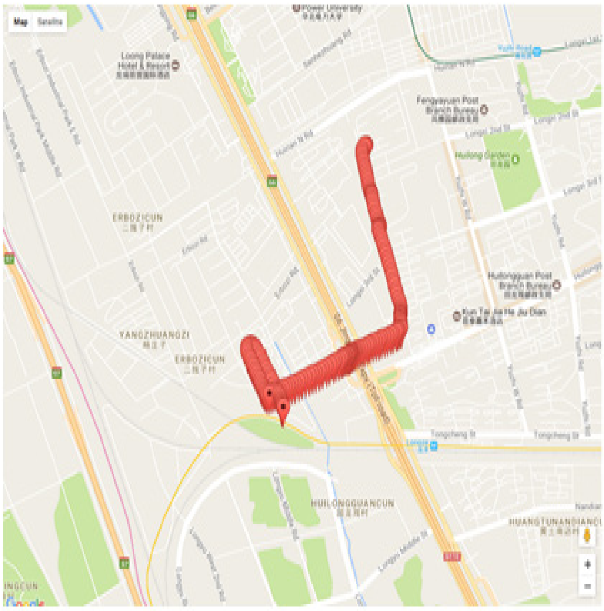

Fig. 1. The original trajectory.

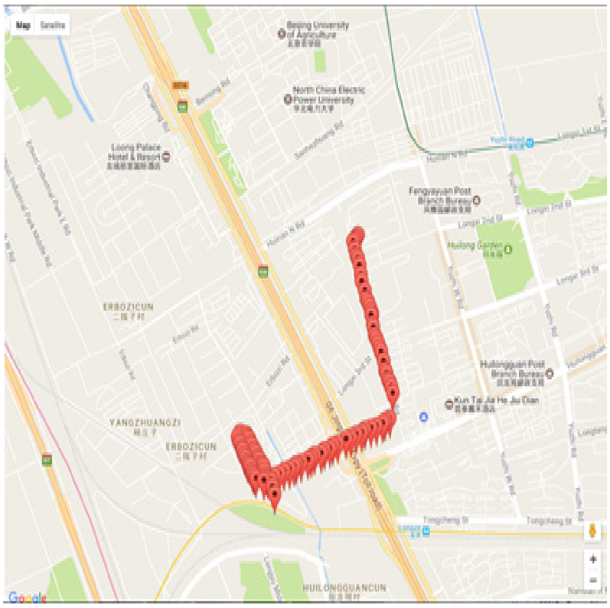
Map
Satellite
Beijing University of Agriculture
North China Electric Power University
Loong Palace Hotel & Resort
ERBOZICUN
YANGZHUANGZI
ERBOZICUN
HUILONGGUANCUN
Fengyayuan Post Branch Bureau
Huilong Garden
Huilongguan Post Branch Bureau
Kun Tai Jia He Jiu Dian
Google

Fig. 2. The compressed trajectory.

Given a spatial trajectory, the storage cost of original trajectory relies on the sampling rate. The higher the sampling rate, the more the points contains with higher storage cost. However, the sampling rate does not affect KiT compression that much. KiT can compress under different sampling rate. Via changing the sampling rate from 5 second/point to 30 seconds/point, the KiT compression on average can achieve a fair compression ratio which is the median sampling rate of the trajectory dataset used.

Fig. 1, shows the original trajectory of the Geolife dataset which is used in this research. A GPS trajectory of this dataset is represented by a sequence of time-stamped points, each of which contains the information of latitude, longitude and altitude. The first, second and third columns represent latitude, longitude and altitude respectively. In this research, two-dimensional operation will be performed in each trajectory including latitude and longitude. A line in a file represents a point and a file represents a trajectory. Fig. 2, presents the compressed trajectory of this dataset using KiT algorithm. KiT algorithms is able to reduce the size of the trajectory data by removing non-crucial data points. It uses geometric principles to identify important positional data points. It shows that a trajectory can be reduced without distorting the original data.

4. Geographic Distance between Two Trajectories

The trajectory of a moving object is typically modelled as a time-stamped sequence of consecutive locations in a multidimensional (generally two or three dimensional) space. Such type of data has offered unprecedented information to help understand the behavior of moving objects, and resulted in growing interest of data analysis in such data. An important problem in such analysis is designing techniques for identifying trajectories that are similar. Such techniques can be used by many data analysis tasks including trajectory clustering, classification, and k nearest neighbor search, which have a broad range of real applications.

When measuring the similarity between two trajectories, an intuitive way is to measure how close in distance they are to each other. In order to analyze trajectory data points, one has to define a metric such that the value of that metric between two points accurately represents the degree of dissimilarity or similarity between them. Dissimilarity is numerical measure of how different two data objects are. It is large when instances are very different and is small when they are close. Distance d (p, q) between two trajectory p and q is a dissimilarity measure if it satisfies:

$$d\,(p, q) \geq 0 \tag{1}$$

$$d\,(p, q) = 0 \tag{2}$$

$$d\,(p, q) = d\,(q, p) \tag{3}$$

$$d\,(p, r) \leq d\,(p, q) + d\,(q, r) \tag{4}$$

In this research, very general procedures, capable of processing all trajectory data, are necessary without prior knowledge of the geographical location where they are collected. Beyond the notion of mathematical distance, many functions can be used to qualify this dissimilarity. Actually, the desired distance should have the properties including comparing the distances as a whole and the trajectories of different lengths, the time indexing can be very different from one trajectory to another.

The usual distance metric used for this purpose is the Euclidean distance, Cosine distance, Manhattan distance and so on. However, these measurements need to have the same number of points in a trajectory. Therefore, these gaps cannot be used directly for human trajectory since these always have GPS different number of points. Another alternative is to find the center of gravity of each trajectory, and then use the Euclidean distance to measure the distance between the two centroids but this also loses a lot of information. To address these issues, Hausdorff distance for trajectory comparison is used by looking at each of the trajectories as an unordered set of spatial coordinates.[12]

4.1. Hausdorff Distance

Hausdorff metric measures how far two subsets of a metric space are from each other. It turns the set of non-empty compact subsets of a metric space into a metric space in its own right. It is used to measure the dissimilarity of two sets of points in a metric space. It is defined as the maximum distance from the first set to the point closest to the second set. The Hausdorff distance is the longest distance that an opponent can be forced to choose a point in one of the two sets, from where you must then go to the other set. In other words, it is the greatest of all distances from one point in a set to the next point in the other set. That is, given two sets of points T_1= {a_1, a_2... a_n} and T_2= {b_1, b_2... b_m}, the directed Hausdorff distance from T_1 to T_2 is given by:

$$h\ (T_1, T_2) = \max a\ €T_1\ \{\min b€T_2\ \{d\ (a, b)\}\} \quad (5)$$

where d (a, b) is the distance between points “a” and “b” under any distance metric of our choice. For our application, Haversine formula has been chosen to calculate the distances between two data points using their reported GPS coordinates.

The distance function h (T_1, T_2) is not symmetric and therefore, h (T_1, T_2) is generally not equal to h (T_2, T_1). There are multiple ways of combining the directed Hausdorff distances to obtain an undirected distance measure. In many instances, undirected Hausdorff distance is obtained by taking the maximum of the two directed measures.

$$H(T_1, T_2)=H\ (T_2, T_1) = \max\ \{h\ (T_1, T_2), h\ (T_2, T_1)\} \quad (6)$$

In the remaining parts of this paper, the distance metric given by (6) is stated as the Hausdorff distance for simplicity.

4.2. Haversine Formula

The haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes and the shortest distance over the earth’s surface. It is an equation important in navigation and a special case of a more general formula in spherical trigonometry, the law of haversines that relates the sides and angles of spherical triangles. Haversine formula denoted by d (a, b) is used to calculate the distance between the individuals points of set T_1 and set T_2 in order to compare the GPS trajectories. Using the Haversine formula, the approximate distance between two points, “a” and “b”, on the surface of the Earth can be calculated as follows:

$$d(a,b)=2\gamma\text{arcsine}*\sqrt{\sin^2\left(\frac{\varphi 2-\varphi 1}{2}\right)+\cos(\varphi 1)\cos(\varphi 2)\sin^2\left(\frac{\delta 2-\delta 1}{2}\right)} \quad (7)$$

here, φ_2 and φ_1 are latitudes, δ_2 and δ_1 are longitudes of points a and b respectively and $\gamma = 6371$ km is the approximate radius of the Earth assuming Earth is a perfect sphere. Actually, the value obtained from the above equations is not metric. So, it is need to standardize the value to the range of [0, 1]. Here, normalization method has been used to be a distance metric from the results of Hausdorff distance function.

5. Semantic Similarity between Two Trajectories

In this section, we discuss the semantic similarity between trajectories. Movement behavior is a particularly complex process, since several factors can affect the movement itself, including the nature of the moving object, its motivation and the geographic environment where the object moves through. Similarity measure of how close to each other two objects are. The closer the objects are to each other, the larger is the similarity value. It falls in the range of [0, 1]. Two completely similar, that is identical, objects give the maximum similarity usually “1”, whereas the least similar pairs reach the minimum value “0”. Similarities, also have some well-known properties.

$$s\ (p, q) = 1 \text{ if } p = q \quad (8)$$

$$s\ (p, q) = s\ (q, p) \text{ for all } p \text{ and } q \quad (9)$$

where, s (p, q) is the similarity between points (data objects), p and q.

After compressing the trajectories, each trajectory is converted into semantic trajectory. In this phase, we use online reverse geocoding tool to transform GPS trajectories into semantic trajectories.[13] Reverse geocoding is the process if back coding of a point location (latitude, longitude) to a readable address or place name. This permits the identification of nearby street addresses, places and/or subdivisions such as state or country.

After transforming each geographic trajectory to a semantic trajectory, each geographic trajectory set is transformed as a semantic trajectory dataset. The semantic trajectory of a moving object may be quite diverse since the object movements may change time to time. However, the main location of an object may be fixed and thus can be discovered.

Hence, to identify user frequent movement location, we perform sequential pattern mining algorithm Prefix-span on each moving object’s semantic trajectory dataset to mine the frequent semantic trajectories by using SPMF.[14] SPMF is an open source data mining library specialized in pattern mining.[15] It offers 120 data mining algorithms. Suppose that, we set the minimum support of Prefix-span algorithm in SPMF as 60%, the pattern and all of its subsequences will be mined. However, it is clear to observe that the longer the pattern we mine the more subsequences will be generated due to the downward closure property. Therefore, the maximal patterns, named maximal semantic trajectory pattern was only maintained for representing user frequent movement places.

Now we discuss the semantic similarity between trajectories. Given two maximal semantic trajectory pattern, the argument is that a longer pattern provides more information about moving object than a shorter pattern. They are more similar when they have more common parts. In this phase, we follow Equal Average of Ref. 16 approach to find the semantic similarity score of the two trajectories. In this approach, the Longest Common Sequence (LCS) algorithm of these two patterns is used to find their longest common part.[17] Longest common subsequence (LCSS) is one of most popular measurements, which is used for string similarity as trajectory similarity measure. For detecting sample points matching like string's characters matching, a threshold is used, if two points' distance less than threshold, they are considered to match. The basic idea of LCSS is that it allows some sample points unmatched to match some sequences in trajectories. LCSS is good for processing with low quality trajectory data (i.e. noisy trajectory data), which can figure out similarity trajectories in high accuracy. The similarity of two trajectory patterns is calculated by averaging the participation ratio of their common part of them. Given the two trajectories T_1 and T_2, a simple approach is to directly compute the average of the two ratios to T_1 and T_2 as shown in the following equation:

$$s\,(T_i,\,T_j) = \frac{\mathrm{ratio}(LCS(T_i,T_j),T_i)+\mathrm{ratio}\,(LCS(T_i,T_j),T_j)}{2} \qquad (10)$$

6. Total Distance between Two Trajectories

Now we need to combine both measurements together. If we take a look at Equation 6, we may find that this is a measurement of the length. We need to first normalize the Hausdorff distance measure to the range [0, 1]. However, when we look into Equation 10, we observe that it is a ratio between [0, 1]. An idea is to take the second measurement as a distance. However, the former is a distance function, but the latter shows a higher score when two trajectories are similar. Thus, we should convert the second measurement to the distance. That is, distance is the complement of the similarity measured in the range of [0, 1]. Obviously, a similarity cannot be metric. In general, any monotonically decreasing transformation can be applied to convert similarity measures into dissimilarity measures. Therefore, we give the following definition, the total distance of two trajectories T_1 and T_2 measures the similarity between them considering both geographic and semantic features, and is defined as:

$$\mathrm{TotalD}\,(T_i,\,T_j) = \frac{d(T_i,T_j)+\sqrt{(1-s(T_i,T_j))}}{2} \qquad (11)$$

7. Experimental Evaluation

In this section, an experimental study obtained from the above-mentioned distance function is illustrated. Geolife Trajectory dataset is used to evaluate the performance of the proposed method. In this dataset, the GPS data is collected and recorded by 182 users over a three years' period of time and contains more than 17,000 trajectories. To evaluate our approach, the proposed method was experimented using Hierarchical Clustering Explorer (HCE) tool. The experiments were run on a PC with Intel Core i7 at 3.40 GHz, 4 GB RAM and 1000 GB hard disk.

Hierarchical clustering approach has been chosen to experiment with total distance function because it can support fully unsupervised learning. Moving objects trajectories may have different number of lengths. It cannot compute the average distance from the different number of trajectory points. Therefore, partitional clustering approach is not suitable for trajectory data mining. A set of experiments is conducted with hierarchical clustering algorithm, which do not require the calculation of the means or centroids of the pairs of trajectories.

Hierarchical clustering does not require a prespecified number of clusters. The hierarchy is needed to be cut at some point. A number of criteria can be used to determine the cutting point. Fig. 3 shows sample result for trajectory clustering obtained by our methods with minimum distance 0.295 of 50 trajectories in the GeoLife dataset. Cutting the dendrogram at distance score 0.295 yields 3 clusters. In this experiment, there are 50 trajectories of three continents including Asia, Europe and North America. It is found that the proposed distance can exactly cluster the trajectories according to their respective continents.

As in flat clustering, the number of clusters k can also specify and select the cutting point that produces k clusters. The corresponding results are shown in Fig. 4 that describes the sample results for trajectory clustering obtained by our methods with k = 7 for 100 trajectories of six countries in the GeoLife dataset including China, Cambodia, USA, Italy, France and Spain. There are six countries but we set the number of clusters as k=7. The effect of combining the semantic score was found in this experiment because we found the next cluster in Cambodia. The proposed method can cluster the moving object trajectories accurately based on their distance score and can capture the hierarchy of clusters of trajectories.

The next experiment was done with 200 trajectories of two regions including Asia and Europe. Setting the number of clusters k=2 yields correctly divide these two regions. The results are proved in Fig. 5.

8. Validation

This section discusses the validation of the results obtained from our proposed methods. Clustering validation is a technique to find a set of clusters that best fits natural partitions (number of clusters) without any class information. Evaluation of clustering results sometimes is referred to as cluster validation. There have been several suggestions for a measure of similarity between two clustering. Such a measure can be used to compare how well different data clustering algorithms perform on a set of data. These measures are usually tied to the type of criterion being considered in assessing the quality of a clustering method. There are two types of criteria to define the cluster quality-external and internal measures. In external evaluation, clustering results are evaluated based on data that was not used for clustering, such as known class labels and external benchmarks. Such benchmarks consist of a set of pre-classified items, and these sets are often created by (expert) humans. Thus, the benchmark sets can be thought of as a gold standard for evaluation. These types of evaluation methods measure how close the clustering is to the predetermined benchmark classes. When a clustering result is evaluated based on the data that was clustered itself, this is called internal evaluation. These methods usually assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters.

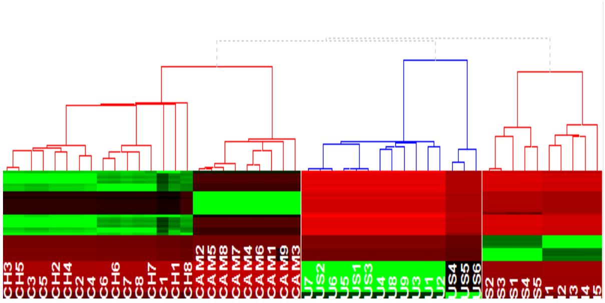

Fig. 3. Clustered result which is extracted by proposed method with minimum distance score (0.295) of 50 trajectories.

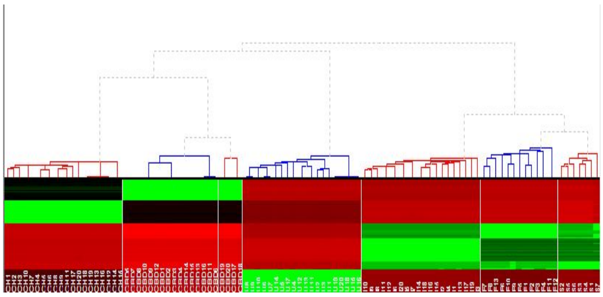

Fig. 4. Clustered result which is extracted by proposed method with k=7 of 100 Trajectories.

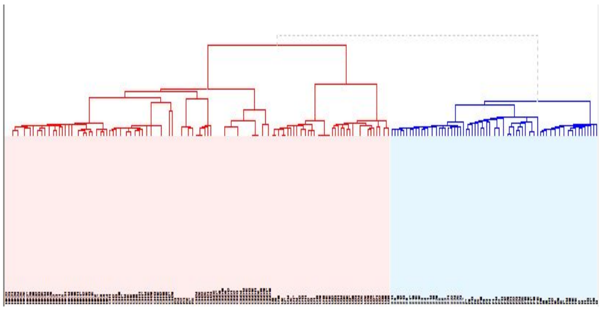

Fig. 5. Clustered result which is extracted by proposed method with k=2 of 100 Trajectories.

The total distance function measuring the similarity between two trajectories in (11) is symmetric because we have proved that the geographic distance and the semantic ratio are both symmetric, therefore the same value will be gotten no matter which trajectory comes first. To evaluate the cluster results with proposed distance function, we the internal criteria namely Silhouette Index of clustering validity approaches is used.[18]

8.1. Silhouette

Silhouette refers to a method of interpretation and validation of consistency within clusters of data. The technique provides a succinct graphical representation of how well each object lies within its cluster. The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters. The silhouette can be calculated with any distance metric.

Silhouette value can be defined as:

$$S(i) = \frac{1-a(i)}{b(i)} \quad \text{if } a(i) < b(i) \tag{12}$$

$$S(i) = 0 \quad \text{if } a(i) = b(i) \tag{13}$$

$$S(i) = \frac{b(i)}{a(i)-1} \quad \text{if } a(i) > b(i) \tag{14}$$

An S(i) close to one means that the data is appropriately clustered. If S(i) is close to negative one, then by the same logic, i would be more appropriate if it was clustered in its neighboring cluster. An S(i) near zero means that the datum is on the border of two natural clusters. The average S(i) over all data of a cluster is a measure of how tightly grouped all the data in the cluster are. Thus, the average S(i) over all data of the entire dataset is a measure of how appropriately the data have been clustered. If there are too many or too few clusters, as may occur when a poor choice of k is used in the clustering algorithm (e.g. k-means), some of the clusters will typically display much narrower silhouettes than the rest. Thus, silhouette plots and averages may be used to determine the natural number of clusters within a dataset. One can also increase the likelihood of the silhouette being maximized at the correct number of clusters by re-scaling the data using feature weights that are cluster specific.

First, the distance between each of trajectories is computed. The average distance to fellow cluster members is then compared to the average distance to members of the neighboring cluster. The silhouette score for an entire cluster is calculated as the average of the silhouette scores of its members. This is a measure of how appropriately the data has been clustered.

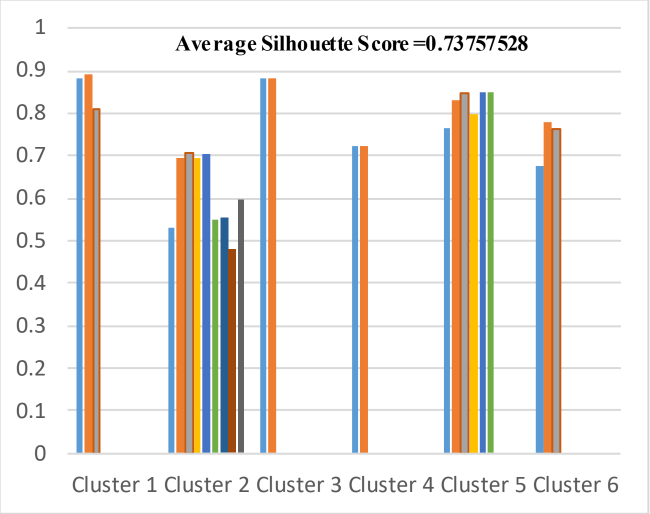

Fig. 6. Average Silhouette score for clustering 25 trajectories with k=6.

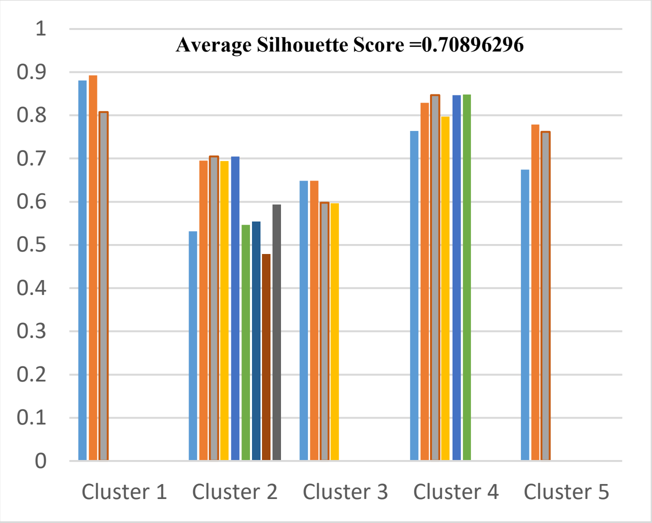

Fig. 7. Average Silhouette score for clustering 25 trajectories with k=5.

Clearly, Fig. 8 demonstrates that our algorithm achieves very good results, with a typical silhouette score about 70% when the cluster size (k) is set to appropriate value. The more accurate the algorithm cluster, the more increase the score. Our proposed method can capture the inherent characteristics of the trajectories.

9. Conclusion

In this research, a distance measurement method is performed to measure the similarity of trajectories. Big trajectory data is compressed by using efficient KiT algorithm in order to handle the size of the dataset easily, and to allow computations with low complexity. Hausdorff distance proves suitable to measure the similarity between trajectories of different lengths. Semantic similarity score of each trajectory is measured based on MSTP method. Then, the total distance measurement method is proposed based on both geographic distance and semantic scores for each trajectory. The proposed method is experimented and evaluated with hierarchical clustering algorithm. The clustering results obtained from the proposed method are analyzed for their effectiveness based on internal evaluation criteria. It appears that hierarchical clustering with total distance provides good performance. Additionally, the proposed approach is completely unsupervised and does not require any special domain knowledge.

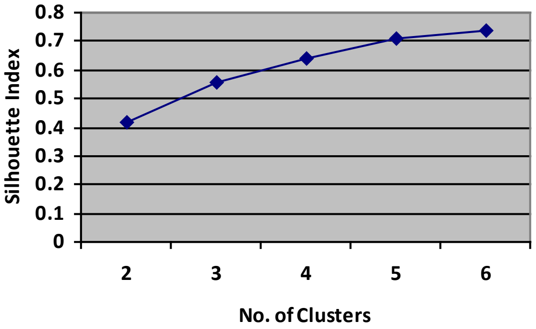

Fig. 8. Average Silhouette score for clustering 25 trajectories with different number of clusters.

References

1. J.D. Mazimpaka, and S.Timpf, Trajectory data mining - A review of methods and applications, J. Spatial Information Science (2015).
2. Y. Zheng, Trajectory Data Mining: An Overview, ACM Transaction on Intelligent Systems and Technology, (2015), Article No 29.
3. P. Sun, S. Xia, G.Yuan, and D.Li, An Overview of Moving Object Trajectory Compression Algorithms, Mathematical Problems in Engineering (2016).
4. HS. Khaing and T.Thein, An Efficient Clustering Algorithm for Moving Object Trajectories, in Proc. 3rd Int. Conf. on Computational Techniques and Artificial Intelligence, ICCTAI '2014, (Feb. 11-12,Singapore, 2014).
5. L.Zhenhui, D. Bolin, H.Jiawei and K.Roland, Swarm: Mining Relaxed Temporal Moving Object Clusters, in Proc. of the VLDB Endowment (2010).
6. J.G. Lee, J.Han, and K.Y.Whang, Trajectory clustering: a partition-and-group framework, In ACM SIGMOD, (2007) pages 593–604.
7. H. Liu and M. Schneider, Similarity measurement of moving object trajectories, In Proc. 3rd ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS '12, pages 19–22, 2012.
8. T. Iwashita, T. Hochin and H. Nomiya, Optimal Number of Clusters for Fast Similarity Search Considering Transformations of Time Varying Data, Int. J. of Networked and Distributed Computing, IJNDC' 2015, Vol. 3, No. 2 (April 2015), pp 79-88.
9. R.Qamar, M.Attique and T.Chung, A Pruning Algorithm for Reverse Nearest Neighbors in Directed Road Networks, Int. J. of Networked and Distributed Computing, IJNDC'2015, Vol. 3, No. 4 (November 2015), pp 261-272.
10. S. Chakri, S.Raghay and S.E.Hadaj, Modeling, Mining, and Analyzing Semantic Trajectories: The Process to Extract Meaningful Behaviors of Moving Objects, In Int. J. of Computer Applications (0975 – 8887),(August 2015), Volume 124 – No.8.
11. Wang And Ting, Sometimes Too Big: Compressing Trajectory Data, In Proc. of PACIS'2014, Paper 370, (2014).
12. S. Nutanong, E. H. Jacox, and H. Samet, An incremental Hausdorff distance calculation algorithm, In Proc. of the VLDB Endowment, vol. 4, no. 8, (2011), pp. 506–517.
13. https://www.doogal.co.uk
14. J. Pei, J. Han, B.Mortazavi-Asl and H.Pinto, PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth, In Proc. of the 17th International Conference on Data Engineering, ICDE '01 ,(2001).
15. http://www.philippe-fournier-viger.com/spmf/
16. J. JYing, E.H. Lu, W.Lee, T.Weng, and V.S.Tseng, Mining user similarity from semantic trajectories, In LBSN, (2010), pages 19–26.
17. https://en.wikipedia.org/wiki/Longest_common_subsequence_problem
18. E. Rendón, I. Abundez, A.Arizmendi and E.M.Quiroz, Internal versus External cluster validation indexes, Int. J. of Computers and Communications, Issue 1, Volume 5, (2011).