International Journal of Scientific Research and Modern Education (IJSRME)
Impact Factor: 6.225, ISSN (Online): 2455 – 5630
(www.rdmodernresearch.com) Volume 2, Issue 1, 2017

333A NEW SIMILARITY MEASURE FOR TRAJECTORY DATA CLUSTERING

D. Mabuni* & Dr. S. Aquter Babu**
Assistant Professor, Department of Computer Science, Dravidian University, Kuppam, Chittoor District, Andhra Pradesh

Cite This Article: D. Mabuni & Dr. S. Aquter Babu, "A New Similarity Measure for Trajectory Data Clustering", International Journal of Scientific Research and Modern Education, Volume 2, Issue 1, Page Number 207-214, 2017.

Abstract:

Trajectory data clustering is a very useful technique in clustering user movement sequences of locations. Finding user movement based clusters is useful in many areas including location based services, finding the location of a person, vehicle, animal, friend recommendation, trajectory ranking, and city traffic management etc,. First, details of users' movement sequences of locations are efficiently stored in the form tree data structure. Second, details of users' movement sequences are clustered by the proposed new trajectory clustering algorithm.

Key Words: Clustering, Movement, Trajectory & Location

1. Introduction:

Many state-of-the-art trajectory data based techniques are available for finding locations of users, vehicles, objects, and animals at any time. A trajectory is a collection of sequential locations of users, vehicles and animals etc., Trajectory data based websites are also exist for handling trajectory data based real applications. Trajectory data is particularly useful in many traffic sharing services and user movement based clustered details. A rich trajectory data based training data sets are available and are very useful and convenient in managing and solving many of the trajectory data based problems. Many works have elaborated on discovering user communities from user location history and such a community is thus referred to as a movement-based community [1]. Trajectory data are used for solving traffic sharing based problems, determining paths, path locations, and friend recommendations and so on. As a result of this better efficient and quality traffic information can be provided to many people as and when required. Some applications of movement based community are: Trajectory ranking, Community based traffic sharing services, and friend recommendation; to discover movement-based communities, one should first formulate the similarity of users in terms of their trajectories [1]. Some of the trajectory data based real time applications are:

- ✓ Finding mobile location
- ✓ Finding user location
- ✓ Friend Communication
- ✓ Finding the location of a particular service
- ✓ Finding the location of a vehicle, ship, animal
- ✓ Trajectory ranking
- ✓ Radar based applications
- ✓ Traffic sharing services
- ✓ Path discovery
- ✓ Movement behavior analysis
- ✓ Friend recommendation
- ✓ Location recommendation

In the first step all the details of users are represented in terms of trajectories and then user movement sequential locations are clustered by using similarity measures among the trajectories. Grouping trajectories based on trajectory data similarity measures is called trajectory data clustering and it has got many real time applications. Trajectory pre-processing generates quality trajectory data. Considerable effort has been devoted to discovery of trajectory patterns in data mining and computational geometry [3].

2. Related Works:

Trajectory data clustering is an important task and in particular clustering of user movement sequential locations is more useful in many applications. User's movement details are represented in terms of trajectories. Clustering of user's movements is one type of trajectory data pattern mining. The goal of trajectory data pattern mining is to cluster movement details of sequential locations. It first finds potential locations and then derives sequential relationships among the potential locations. One must consider data uncertainty and fuzziness of potential sequential locations of trajectories before clustering. Wen-Yuan Zhu et al. [1] proposed a new similarity function to model user similarity in terms of their trajectory profiles as tree structures. The increase availability of location acquisition technologies such as GPS set on cars, WLAN networks, and mobile phones carried by people have enabled tracking almost any kind of moving objects, which results in huge volumes of spatio-temporal data in the form of trajectories [2]. Trajectory data is very useful to provide a lot of opportunities for analyzing movement behavior of moving objects. Moving objects are grouped into clusters based on sequential activities of people's behavior. The knowledge obtained from trajectory data mining techniques is very useful in improving quality life of users in areas such as urban cities, finding user or vehicle location dynamically, and path determination of a user and so on. Trajectory data mining is useful for individual persons as well as for groups of people. It is the well known fact that location acquisition technologies generate huge amounts of trajectory data. The recent proliferation of ubiquitous sensing technologies, intelligent transportation systems, and location based services increases the availability of human trajectories [4]. Many trajectory data based computer management information systems are developed for finding different types of movement patterns of users and knowledge. Trajectory data clustering is one part in trajectory data mining. Trajectory data clustering is very useful in the case of moving objects trajectory data mining. Trajectory data is crucial in many applications of moving objects.

3. Trajectory Data Preprocessing:

Originally collected data sets of trajectories must be processed for obtaining correct and reasonably real and convenient for trajectory data preprocessing. Sequential pattern mining is one way to derive potential data sets of trajectories. A trajectory consists of sequence of time stamped locations over geographical area. During trajectory data preprocessing step, trajectories must be cleaned, modified, segmented and removed with free of uncertainty in the data sets of trajectories. Sometimes trajectories are simplified further for easy processing. Efficient indexing structures are also needed for convenient representation of trajectories in the memory. Efficient indexing structure alleviates many problems during query processing. Various and different types of trajectory data mining techniques are classification, clustering, pattern mining, knowledge based creation, intelligent decision making and so on. A trajectory location is usually represented by longitude and latitude and the point (longitude, latitude) is unique in the geographical space.

4. Trajectory Data Problem Definition:

A trajectory is a collection sequential location detail of a single user or object for one trip. Object may be a vehicle, person, and ship and so on. User details for 'n' number of trips of a single user are represented by 'n' number of sequential trees. Sequential trees are useful for representing and processing trajectory data sets efficiently and effectively. A breadth first algorithm is used for representing group of trajectories of a single person in the form of sequential data structure. By using the same procedure, 'n' groups of trajectories belonging to 'n' distinct number of customers are represented in the form of 'n' sequential data structure trees. During construction process of sequential trees support and conditional probabilities are used for optimal construction of sequential trees. Once all sequential trees are generated then trees are clustered by using newly proposed trajectory data similarity measure. Symmetrical difference is the newly proposed trajectory data similarity measure and it is easy to compute and apply on sequential tree structures.

5. Trajectory Data Clustering:

The goal of trajectory data clustering is to group similar trajectories into clusters in such way that trajectories within the same group contain similar features and trajectories in the different groups have dissimilar features. For better management of trajectory data, trajectory data locations must be equipped with semantic meanings. Most important methods that are used for community based clustering are – Clique method, Hierarchical method, and Betweenness method [1]. Trajectory clustering techniques aim to find groups of moving object trajectories that are close to each other and have similar geometric shapes [2]. Trajectory data sets are collected by various types of moving objects such as people, vehicles, ships, and animals etc. Trajectories are also useful in path relating problems. Most frequent path is better than fastest and shortest trajectory data paths. Finding the desired location and finding the destination location are important applications of trajectory data mining. User's trajectories are useful in finding optimal decisions of applications relating to users. Semantic trajectories are generally created by tagging a location point with some meaningful details such as location value, weight, time, frequency, maximum count and so on. Important points in trajectory data management are:

- ✓ Trajectory data prepossessing,
- ✓ Trajectory data management,
- ✓ Trajectory data based query processing trajectory data mining tasks.

Prepossessing is a step where trajectory data quality is improved. Trajectory data management is required to represent trajectory data efficiently, effectively, and conveniently with scalable manner. Performance of the trajectory data mining tasks decrease as the number of outliers Increases. Two main problems in trajectory data management are: trajectory data uncertainty and storage of very large trajectory data sets. Data uncertainly in the trajectory data sets must be converted into trajectory data certainty. An index structure is needed to store and manage very large amounts of trajectory data sets. Particularly, a tree based multi-way indexing structure is needed for both efficient storage and efficient trajectory data based query management. Query processing is most important task in trajectory data management. Different types of trajectory based queries are pattern queries, nearest neighbor queries, aggregate queries and range queries.

6. Trajectory Data Based Similarity Measures:

Trajectory data based similarity measures are needed to find the differences between two trajectories. Different similarity measures may use different properties to find or extract similarities between two trajectories. An interesting problem on large trajectory data is the similarity search and a trajectory is represented as a sequence of locations each is associated with a corresponding time stamp [5]. Trajectory data reflects human mobility, it can be naturally utilized for location based recommendation applications, including personalized location prediction, group based location recommendation, and user mobility modeling [6]. Various types of trajectory similarity measures are:

- ✓ DTW (Dynamic Time Warping)
- ✓ Edit distance with really penalty (ERP)
- ✓ Edit distance on real sequence (EDR)
- ✓ Cosine similarity
- ✓ HGSM (hierarchical graph based similarity measurement)
- ✓ EDwP (Edit Distance with Projections)
- ✓ DISSIM
- ✓ Model driven Assignment (MA)
- ✓ Euclidian Distance
- ✓ Longest Common Sub Sequence

HGSM is used for comparing the trajectory similarity between two users by using trajectory data. The similarity measure DISSIM finds similarity between trajectories at uniform sampling rates only EDP, ERP and LCSS use thresholds to determine similarity between locations. EDWP trajectory similarity measure is not scalable and the measure EDWP is robust as well as accurate to noise. Trajectories data are useful for people in many applications such as

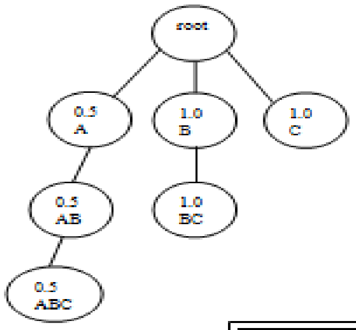
- ✓ User Behavior Analysis
- ✓ Rout Recommendation
- ✓ Traffic Analysis
- ✓ Social Relationship Analysis

1. {A, B, C, AB, BC, ABC}
2. {A, B, C, AB, AC, BC}
3. {B, C, D, BC, BD, CD, BCD}
4. {B, C, D, E, BC, BD, BE, CD, BDE, CDE}
5. {A, B, C, D, AB, BC, CD, ABC}
6. {A, B, C, D, AB, AC, BC, CD}
7. {B, C, D, BC, CD, BCD}
8. {B, C, D, E, BC, BD, CD, BDE, CDE}

Initially, movement sequences of each user are represented by a tree data structure. Tree structure is generated by using a breadth first search (BFS) algorithm with constraints of minimum support and conditional probabilities. Sequential trees are pruned with minimum support and minimum conditional probabilities for obtaining optimal trees. The whole process is divided into two steps. In the first step, movement sequences of 'n' number of users are represented in 'n' sequential trees. In the second step, these tree structures are clustered with the help of newly proposed trajectory data clustering algorithm.

For example, when n value is 8, eight trees are generated corresponding to trajectory data movement sequences of eight users. All these eight trees are shown from FIGURE-1 to FIGURE-8 respectively. Tree nodes are named from top to bottom and left to right. Node of each tree contains two values. The first value indicates support and the second value indicates name of the node. Suffix tree notation is used for naming the nodes.

The proposed trajectory data clustering algorithm is executed on the eight given sample trees. A newly proposed trajectory data similarity measure is used for clustering above said eight initial individual trees. There exist many trajectory data similarity finding measures between trajectories. Trajectory data clustering computational details of these eight individual sequential tree data structures are summarized as follows.



Trajectory Data
ABC
ABC
BC
BC

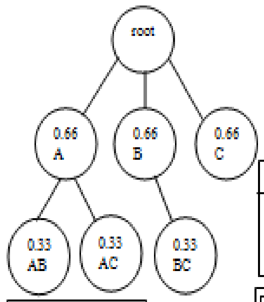
root	
NODE	PROB
A	0.2
B	0.4
C	0.4

A	
NODE	PROB
B	1.0

B	
NODE	PROB
C	1.0

AB	
NODE	PROB
C	1.0

FIG-1 Support and conditional probabilities of user_1



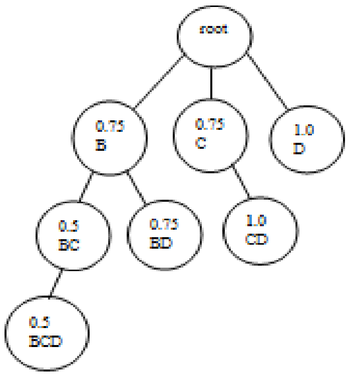
Trajectory Data
AB
AC
BC

A	
NODE	PROB
B	0.5
C	0.5

root	
NODE	PROB
A	0.33
B	0.33
C	0.33

B	
NODE	PROB
C	1.0

FIG-2 Support and conditional probabilities of user_2



Trajectory Data
BCD
BCD
BD
CD

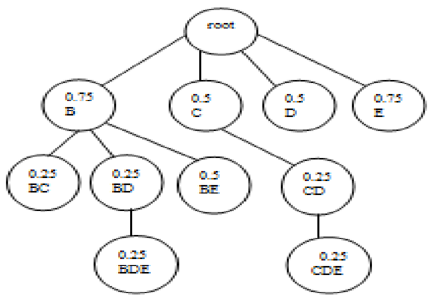
root	
NODE	PROB
B	0.3
C	0.3
D	0.4

B	
NODE	PROB
C	0.666
D	0.333

C	
NODE	PROB
D	1.0

BC	
NODE	PROB
D	1.0

FIG-3 Support and conditional probabilities of user_3



Trajectory Data
BC
BDE
BE
CDE

root	
NODE	PROB
B	0.3
C	0.2
D	0.2
E	0.3

B	
NODE	PROB
C	0.33
D	0.33
E	0.33

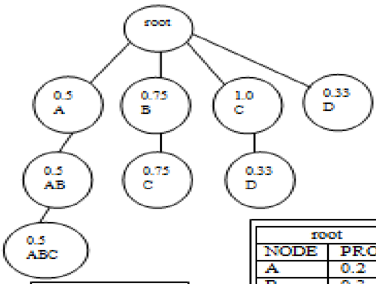
C	
NODE	PROB
D	1.0

AB	
NODE	PROB
C	1.0

BD	
NODE	PROB
E	1.0

CD	
NODE	PROB
E	1.0

FIG-4 Support and conditional probabilities of user_4



Trajectory Data
ABC
ABC
BC
CD

root	
NODE	PROB
A	0.2
B	0.3
C	0.4
D	0.1

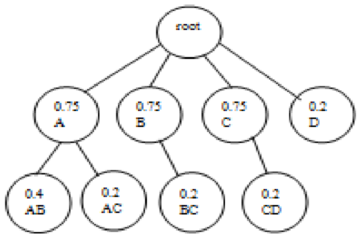
A	
NODE	PROB
B	1.0

B	
NODE	PROB
C	1.0

C	
NODE	PROB
D	1.0

AB	
NODE	PROB
C	1.0

FIG-5 Support and conditional probabilities of user_5



Trajectory Data
AB
AB
AC
BC
CD

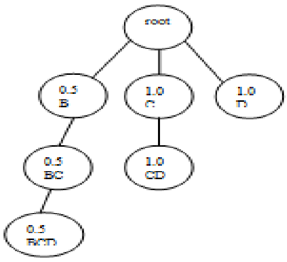
root	
NODE	PROB
A	0.3
B	0.3
C	0.3
D	0.1

A	
NODE	PROB
B	0.666
C	0.333

B	
NODE	PROB
C	1.0

C	
NODE	PROB
D	1.0

FIG-6 Support and conditional probabilities of user_6



Trajectory Data
BCD
BCD
CD
CD

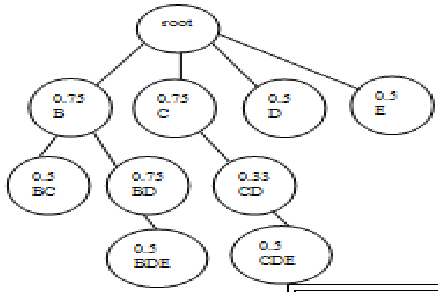
root	
NODE	PROB
B	0.2
C	0.4
D	0.4

C	
NODE	PROB
D	1.0

BC	
NODE	PROB
D	1.0

B	
NODE	PROB
C	1.0

FIG-7 Support and conditional probabilities of user_7



Trajectory Data
BDE
CDE
BC
BC

root	
NODE	PROB
B	0.3
C	0.3
D	0.2
E	0.2

B	
NODE	PROB
C	0.666
D	0.333

C	
NODE	PROB
D	1.0

BD	
NODE	PROB
E	1.0

CD	
NODE	PROB
E	1.0

FIG-8 Support and conditional probabilities of user_8

Sequential symmetric difference of trajectory data similarity measure between Trees-1 and Trees-2 is defined as follows:

Tree-1 Δ Tree-2 = {Tree-1 – Tree2} U {Tree2 – Tree1}

For simplicity purpose, Tree-1 and Tree-2 are denoted by T-1 and T-2 respectively. Initially 8 clusters are there and 8/2 = 4 is selected as head cluster. Now cluster 4 is compared with all the remaining clusters using symmetrical difference trajectory data similarity measure. Symmetrical difference trajectory data similarity measures are computed as follows:

T-4 Δ T-1 = {T-4 – T-1} U {T-1 – T-4} = {DE,BD,BE,CD} U {A, AB, ABC}
= {A,AB,ABC,BE,BD,CD,DE} and |{A,AB,ABC,BE,BD,CD,DE} | = 7

T-4 Δ T-2 = {T-4 – T-2} U {T-2 – T-4} = {D,E,BD,BE,BDE,CD,CDE} U {A,AB,AC}
= {A,AB,AC,BD,BE,BDE,CD,CDE,D} and |{A,AB,AC,BD,BE,BDE,CD,CDE,D}| = 10

T-4 Δ T-3 = {T-4 – T-3} U {T-3 – T-4} = {E,BE,BDE,CDE} U {BCD}
= {E,BE,BDE,CDE,BCD} and |{E,BE,BDE,CDE,BCD} | = 5

T-4 Δ T-5 = {T-4 – T-5} U {T-5 – T-4} = {E,BD,BE,BDE,CDE} U {A,AB,ABC}
= { E,BD,BE,BDE,CDE,A,AB,ABC } and |{ E,BD,BE,BDE,CDE,A,AB,ABC } | = 8
T-4 Δ T-6 = {T-4 – T-6} U {T-6 – T-4} = {E,BD,BE,BDE,CDE} U {A,AB,AC}
= { E,BD,BE,BDE,CDE,A,AB,AC } and |{ E,BD,BE,BDE,CDE,A,AB,AC } | = 8
T-4 Δ T-7 = {T-4 – T-7} U {T-7 – T-4} = {E,BD,BE,BDE,CDE} U {BCD}
= { E,BD,BE,BDE,CDE,BCD } and |{ E,BD,BE,BDE,CDE,BCD } | = 6
T-4 Δ T-8 = {T-4 – T-8} U {T-8 – T-4} = {BE} U {} = {BE} and |{ BE } | = 1
Here, 4 and 8 are clustered

----

1. {A, B, C, AB, BC, ABC}
2. {A, B, C, AB, AC, BC}
3. {B, C, D, BC, BD, CD, BCD}
4. {B, C, D, E, BC, BD, BE, CD, BDE, CDE}
5. {A, B, C, D, AB, BC, CD, ABC}
6. {A, B, C, D, AB, AC, BC, CD}
7. {B, C, D, BC, CD, BCD}

7/2 = 3 is taken as head cluster and similarity measures are computed with 3 and all the remaining clusters.
T-3 Δ T-1 = {T-3 – T-1} U {T-1 – T-3} = {D,BD,CD,BCD} U {A, AB, ABC}
= { D,BD,CD,BCD,A, AB, ABC } and |{ D,BD,CD,BCD,A, AB, ABC } | = 7
T-3 Δ T-2 = {T-3 – T-2} U {T-2 – T-3} = {D,BD,CD,BCD} U {A,AB,AC}
= { D,BD,CD,BCD,A,AB,AC } and |{ D,BD,CD,BCD,A,AB,AC }| = 7
T-3 Δ T-4 = {T-3 – T-4} U {T-4 – T-3} = {BCD} U {E,BE,BDE,CDE}
= { BCD,E,BE,BDE,CDE } and |{ BCD,E,BE,BDE,CDE } | = 5
T-3 Δ T-5 = {T-3 – T-5} U {T-5 – T-3} = {BD,BCD} U {A,AB,ABC}
= {A,AB,ABC, BCD, BD } and |{ A,AB,ABC, BCD, BD } | = 5
T-3 Δ T-6 = {T-3 – T-6} U {T-6 – T-3} = {BD,BCD} U {A,AB,CD,ABC}
= { BD,BCD,A,AB,CD,ABC } and |{ BD,BCD,A,AB,CD,ABC } | = 6
T-3 Δ T-7 = {T-3 – T-7} U {T-7 – T-3} = {BD} U {} = {BD} and |BD\ = 1
Here, 3 and 7 are clustered

----

1. {A, B, C, AB, BC, ABC}
2. {A, B, C, AB, AC, BC}
3. {B, C, D, BC, BD, CD, BCD}
4. {B, C, D, E, BC, BD, BE, CD, BDE, CDE}
5. {A, B, C, D, AB, BC, CD, ABC}
6. {A, B, C, D, AB, AC, BC, CD}

6/2 = 3 Hence, cluster 3 is taken as head cluster.

T-3 Δ T-1 = {T-3 – T-1} U {T-1 – T-3} = {BCD, BD,CD,D} U {A, AB, ABC}
= {A, AB, ABC, BCD, BD,CD,D } and |{ A, AB, ABC ,BCD, BD,CD,D } | = 7
T-3 Δ T-2 = {T-3 – T-2} U {T-2 – T-3} = {BCD, BD, CD, D} U {A, AB, ABC}
= {A, AB, ABC, BCD, BD, CD, D } and |{ A, AB, ABC ,BCD, BD,CD,D } | = 7
T-3 Δ T-4 = {T-3 – T-4} U {T-4 – T-3} = {BCD} U {E, BE, BDE, CDE}
= {BCD, E, BE, BDE, CDE } and |{ BCD, E, BE, BDE, CDE } | = 5
T-3 Δ T-5 = {T-3 – T-5} U {T-5 – T-3} = {BD, BCD} U { A, AB, ABC }
= {A, AB, ABC, BD, BCD } and |{A, AB, ABC, BD, BCD} | = 5
T-3 Δ T-6 = {T-3 – T-6} U {T-6 – T-3} = {BC, BD, BCD} U { A, AB, ABC }
= {A, AB, ABC, BC, BD, BCD} and |{ A, AB, ABC, BC, BD, BCD} | = 6
Here, 3 and 5 are clustered

----

1. {A, B, C, AB, BC, ABC}
2. {A, B, C, AB, AC, BC}
3. {A, B, C, D, BC, BD, CD, BCD, AB, ABC}
4. {B, C, D, E, BC, BD, BE, CD, BDE, CDE}
5. {A, B, C, D, AB, AC, BC, CD}

5/2 = 2. Hence, cluster 2 is taken as head cluster.

T-2 Δ T-1 = {T-2 – T-1} U {T-1 – T-2}
{AC} U {ABC} = {AC, ABC} and |{AC, ABC}| = 2
T-2 Δ T-3 = {T-2 – T-3} U {T-3 – T-2}
{AC} U {D, BD, CD, BCD, ABC} = {AC, D, BD, CD, BCD, ABC } and |{ AC, D, BD, CD, BCD, ABC }| = 6

T-2 Δ T-4 = {T-2 – T-4} U {T-4 – T-2} = {A, AB, AC} U {D, E, BD, BE, CD, BDE, CDE}
= {A, AB, AC, D, E, BD, BE, CD, BDE, CDE} and |{ A, AB, AC, D, E, BD, BE, CD, BDE, CDE } | = 10
T-2 Δ T-5 = {T-2 – T-5} U {T-5 – T-2} = {} U {D, CD} = {D, CD} and |{ D, CD } | = 2
Here, 2 and 5 are clustered.

----

1. {A, B, C, AB, BC, ABC}
2. {A, B, C, AB, AC, BC, D, CD}
3. {A, B, C, D, BC, BD, CD, BCD, AB, ABC}
4. {B, C, D, E, BC, BD, BE, CD, BDE, CDE}

4/2 = 2 is taken as head cluster
T-2 Δ T-1 = {T-2 – T-1} U {T-1 – T-2} = {AC, D, CD} U {ABC}
= { AC, D, CD, ABC } and |{ AC, D, CD, ABC } | = 4
T-2 Δ T-3 = {T-2 – T-3} U {T-3 – T-2} = {AC} U {BD, BCD, ABC}
= { AC, BD, BCD, ABC } and |{ AC, BD, BCD, ABC } | = 4
T-2 Δ T-4 = {T-2 – T-4} U {T-4 – T-2} = {A, AB, AC} U {E, BD, BE, BDE, CDE}
= {A, AB, AC, E, BD, BE, BDE, CDE} and |{A, AB, AC, E, BD, BE, BDE, CDE}| = 8
Here, 2 and 3 are clustered and Final resultant clusters are shown below:
1) {A, B, C, AB, BC, ABC}------- (1)
2) {A, B, C, D, AB, AC, ABC, BC, BD, BCD, CD, D} ----------four clusters, (2, 3, 5, 6) are clustered
3) {B, C, D, E, BC, BD, BE, CD, BDE, CDE} --------------------------------- 2 clusters (4, 8) are clustered

Proposed New Trajectory Data Similarity Measure Algorithm for Trajectory Data Clustering

---

1. input minimum support and minimum conditional probability threshold values
2. input trajectory profiles of 'n' users
3. create 'n' number of tree structures corresponding to 'n' number of users trajectory profiles
4. clusterCount ← n
5. threshold ← n*(30/100)
6. k ← n/2
7. while (clusterCount > threshold) do
    7.1 presentCluster ← k
    7.2 for i =1 to clusterCount do
        determine pair wise symmetric difference measure
        values for all the clusters with the present cluster.
    7.3 combine the pair with minimum symmetrical
        difference value.
    7.4 clusterCount = clusterCount – 1
    7.5 k ← n/2
  End-of-while
8. display all the final clusters

Proposed New Trajectory Data Similarity Measure for Trajectory Data Clustering Algorithm Explanation
Algorithm working procedure is explained below:

At the beginning trajectory profiles of user's are pre-processed and cleaned trajectory data sets are created. Here, 'n' number of sequential tree data structures are created for 'n' number of user profiles. Here, cluster Count Variable initially contains 'n' number of individual starting clusters. Head cluster is a cluster such that it is used to find symmetrical differences with all the remaining clusters. Initially n/2 is taken as head cluster. For example,
8/2 = 4 is the head cluster in the first iteration.
7/2 = 3 is the head cluster in the first iteration.
6/2 = 3 is the head cluster in the first iteration.
5/2 = 2 is the head cluster in the first iteration.
4/2 = 2 is the head cluster in the first iteration.
In each iteration of the while loop head cluster is computed using the formula:
Head cluster no = total no of present clusters count / 2
Each iteration of while loop, pair wise symmetrical differences are computed with head cluster and all the remaining clusters. Two clusters with minimum symmetric difference value are combined and then clusterCount is reduced by one. In the second iteration cluster-2 is selected as head cluster, in the third iteration cluster-3 is selected as head cluster and so on. In the first iteration of while loop, cluster1 (Tree-1) is taken as head cluster and all other clusters are taken as subordinate clusters. Pair wise symmetric difference is computed for all these clusters.

In the first iteration of the while loop cluster pairs (1, 2), (1, 3), (1, 4),(1, 5), (1, 6)...(1, n) are considered for finding symmetrical difference according to pairs wise. Here, cluster-1 is taken as head cluster. Similarly in the second iteration of the while loop clustered pairs (2, 1), (2, 3), (2, 4), (2, 5) … (2, n-1) are considered. Similarly third, fourth iteration pair wise comparisons are

(3, 1), (3, 2), (3, 4), (3, 5), (3, 6) ... (3, n-2)

(4, 1), (4, 2), (4, 3), (4, 5), (4, 6) ... (4, n-3) and so on

While loop terminates when cluster count is less than the threshold value. Threshold value dictates the final number of output resultant clusters. Threshold is taken approximately thirty percent of the initial number of clusters.

7. Conclusions:

Trajectory data are very useful in many day to day applications of present scenario. Trajectory data clustering is gaining its importance in diversified fields. Present paper proposes an algorithm for clustering sequential locations movements of users called movement clustering. There exists many similarity measures to compare any two clusters with similar domain. In the present paper a new trajectory data similarity measure called symmetrical difference similarity measures is used to similarity between two clusters. In the future, there is a possibility to extend this work for clustering trajectory data using more advanced trajectory data similarity measures and efficient clustering algorithms.

8. References:

1. Wen-Yuan Zhu, Wen-Chih Peng, Chih-Chieh Hung, Po-Ruey Lei, and Ling-Jyh Chen, "Exploring Sequential Probability Tree for Movement-Based Community Discovery," IEEE Trans on Knowledge and Data Engineering, Vol. 26, No. 11, November 2014, Pages 2717-2730
2. Kai Zheng, Yu Zheng, Nicholas J. Yuan, Shuo Shang, and Xiaofang Zhou "Online Discovery of Gathering patterns over Trajectories", IEEE Trans On Knowledge and Data Engineering, Vol.26,No.8,August 2014, pages 1974-1988
3. Jae-Gil Lee, Jiawei Han and XiaoleiLi, "A Unifying Framework of Mining Trajectory Patterns of Various Temporal Tightness", IEEE Trans on Knowledge and Data Engineering, Vol.27, No.6, June 2015, pages 1478-1490.
4. Nicholas Jing Yuan, Yu Zeng, Xing Xie, Yingzi Wang, and Hui Xiong, "Discovering Urban Functional Zones Using Latent Activity Trajectories", IEEE Trans on Knowledge and Data Engineering, Vol.27, No.3, March 2015, pages 712-725.
5. Chunyang Ma, Hua Lu, Lidan Shou, and Gang Chen, "KSQ: Top-k Similarity Query on Unceratin Trajectories", IEEE Trans on Knowledge and Data Engineering, Vol.25, No.9, September 2013, pages 2049-2062.
6. NingNan Zhou, Wayne Xin Zhao, Xiao Zhang, Ji-Rong Wen, and Shan Wang, "A General Multi-Context Embedding Model for Mining Human Trajectory Data", IEEE Trans On Knowledge and Data Engineering, Vol.28, No.8, November 2016, pages 1945-1958.