Clustering based on user Movement Similarity Measures

R. Muni Latha
M.Tech Student, Department of CSE
KMM Institute of Technology and Sciences
Tirupati, India

Dr. K. Venkata Ramana
Associate Professor, HOD, Department of CSE
KMM Institute of Technology and Sciences
Tirupati, India

***Abstract* – In real life different users are performing different varieties of tasks. In this paper, users are clustered based on user tasks details. Multi-way sequential probability trees are constructed for representing user movement pattern details. For each user, tasks details are represented in one sequential probability tree data structure. Once all the tree data structures are constructed, then users are clustered based on the similarity measurement details among the users. We propose a new technique to cluster user tasks details. Proposed framework for clustering tasks details of users is very easy and very simple. A special and very simple similarity measurement technique is proposed and used in measuring the similarity details among the users. Proposed clustering based techniques are very useful in many real life applications. Example application are- trends of studies, bank loan issuing patterns, location based services, share market trend change details.**

Keywords – Similarity measures, clustering, machine learning, normalized similarity measure

I. INTRODUCTION

Clustering is one of the best techniques in both machine learning and data mining. Sequential location movement details of vehicles, cell phones, users etc are nowadays are most import and must be needed and useful in many real life applications. To solve many of such problems we have proposed new similarity measurement techniques among the users, vehicles, cell phones, studies trends etc and then this new technique is used to cluster the user movement details based on similarity measures. In many real life applications finding sequential location pattern details of users is very useful for decision making.

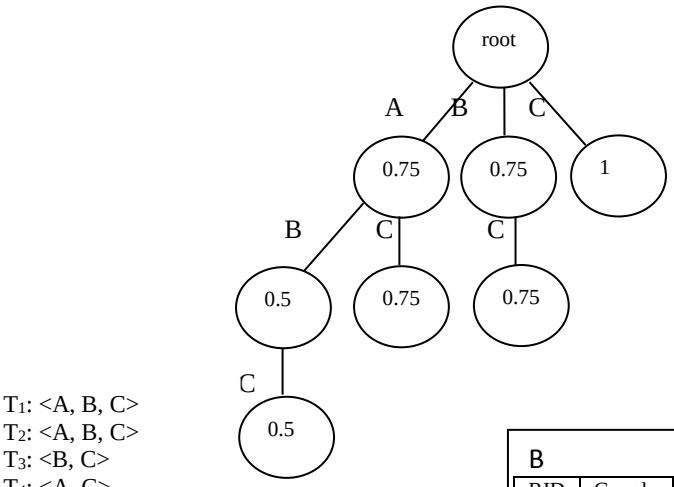
Assume that n users are there. For each user a sequential probability tree data structure is constructed based on the trajectory profile of each user. Time complexity of tree data structure is O(logn). After constructing n tree data structures all user details are clustered based on the new similarity measurement technique,

Some applications of movement-based communities are given below:

Trajectory ranking, Community-based traffic sharing services, and Friend communication [1]. In retrieving user trajectories trajectory ranking is useful. Traffic sharing services have become very useful. For example, users can download the mobile application to share traffic information by using navigational services [1]. Users are grouped into clustered based on the similarity behaviors of users. To capture movement behaviors we can use sequential pattern methods [1].

Movement based communities are very useful in managing many applications such as GPS (Global Positioning System), finding the specific tower to forward a phone call, Identifying the location of a specific object, Trajectory ranking, Finding a service centre at a particular place and at a particular time.

II. SEQUENTIAL PATTERNS OF USERS



T_1: <A, B, C>
T_2: <A, B, C>
T_3: <B, C>
T_4: <A, C>
User U_1's trajectories and SP-tree (SPT_1)

root

RID	C.prob
A	0.3
B	0.3
C	0.4

A

RID	C.prob
B	0.67
C	0.33

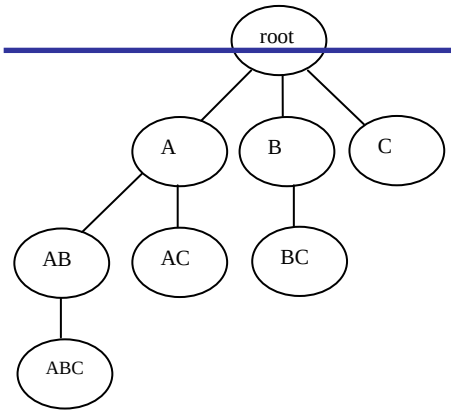
B

RID	C.prob
C	1

AB

RID	C.prob
C	1

Fig. 1. User U_1's trajectories and SP-tree SPT_1



Nodes of SPTree$_1$are named as shown above

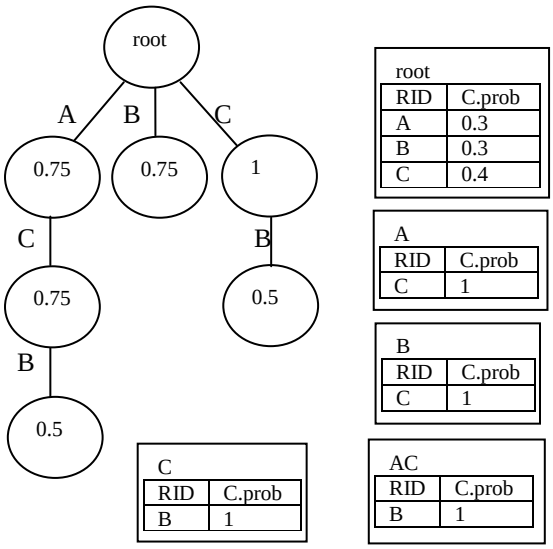


Fig. 2. User U_2's trajectories and SP-tree SPT_2

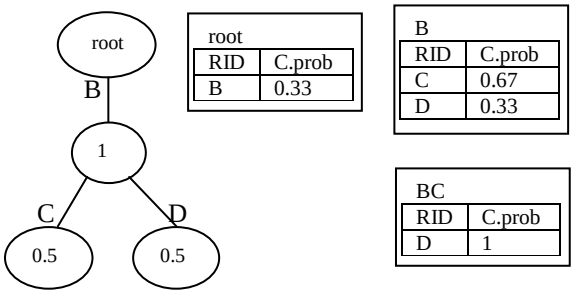


Fig. 3. User U_3's trajectories and SP-tree SPT_3

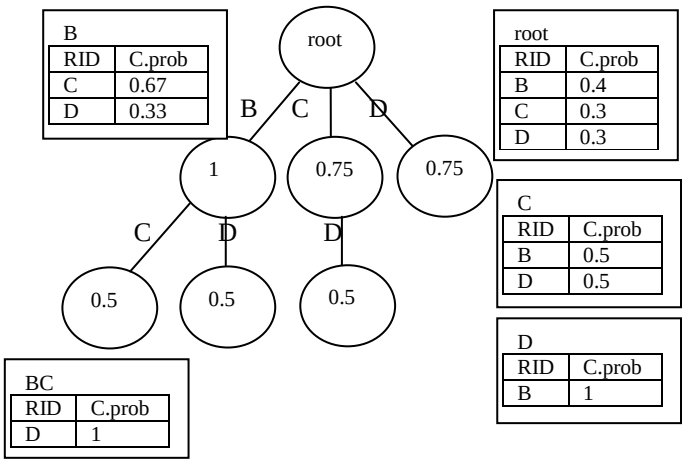


Fig. 4. User U_4's trajectories and SP-tree SPT_4

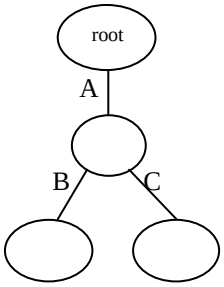


Fig.5 SPT_5

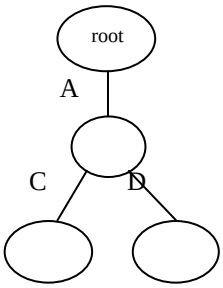


Fig 6 SPT_6

Movement behaviors are characterized using movement sequential patterns and transition probabilities [1]. We propose a special and new similarity measure finding algorithm for grouping users into clusters. This new similarity uses a simple normalized measure for comparing two users. Based on the minimum threshold value clusters are created. Clustering process stops when all users are checked.

For each sequential probability tree the nodes are named as follows:

Node names of SPT_1 = {root, A, B, C, AB, AC, BC, ABC}
Node names of SPT_2 = {root, A, B, C, AC, CB, ACB}
Node names of SPT_3 = {root, B, BC, BD}
Node names of SPT_4 = {root, B, C, D, BC, BD, CD}
Node names of SPT_5 = {root, A, AB, AC}
Node names of SPT_6 = {root, A, AC, AD}

$SPT_1 \cap SPT_2$= {root, A, B, C, AC}
$SPT_1 \cup SPT_2$= {root, A, B, C, AB, AC, BC, ABC, CB, ACB}

Normalized similarity measure between trees SPT_1 and $SPT_2 = \frac{SPT_1 \cap SPT_2}{SPT_1 \cup SPT_2} = \frac{5}{10} = 0.5$

$SPT_1 \cap SPT_3$= {root, B, BC}
$SPT_1 \cup SPT_3$= {root, A, B, C, AB, AC, BC, ABC, BD}
Normalized similarity measure between trees SPT_1 and $SPT_3 = \frac{SPT_1 \cap SPT_3}{SPT_1 \cup SPT_3} = \frac{3}{9} = 0.33$

$SPT_1 \cap SPT_4$= {root, B, C, BC}
$SPT_1 \cup SPT_4$= {root, A, B, C, AB, AC, BC, ABC, D, BD, CD}
Normalized similarity measure between trees SPT_1 and $SPT_4 = \frac{SPT_1 \cap SPT_4}{SPT_1 \cup SPT_4} = \frac{4}{11} = 0.363$

$SPT_1 \cap SPT_5$= {root, A, AB, AC}
$SPT_1 \cup SPT_5$= {root, A, B, C, AB, AC, BC, ABC}
Normalized similarity measure between trees SPT_1 and $SPT_5 = \frac{SPT_1 \cap SPT_5}{SPT_1 \cup SPT_5} = \frac{4}{8} = 0.5$

$SPT_1 \cap SPT_6$= {root, A, AC}
$SPT_1 \cup SPT_6$= {root, A, B, C, AB, AC, BC, ABC, AD}
Normalized similarity measure between trees SPT_1 and $SPT_6 = \frac{SPT_1 \cap SPT_6}{SPT_1 \cup SPT_6} = \frac{3}{9} = 0.33$

$SPT_2 \cap SPT_3$= {root, B}
$SPT_2 \cup SPT_3$= {root, A, B, C, AC, CB, ACB}
Normalized similarity measure between trees SPT_2 and $SPT_3 = \frac{SPT_2 \cap SPT_3}{SPT_2 \cup SPT_3} = \frac{2}{7}$ =0.285

$SPT_2 \cap SPT_4$= {root, B, C}
$SPT_2 \cup SPT_4$= {root, A, B, C, AC, CB, ACB, D, BC, BD, CD}
Normalized similarity measure between trees SPT_2 and $SPT_4 = \frac{SPT_2 \cap SPT_4}{SPT_2 \cup SPT_4} = \frac{3}{11}$ =0.272

$SPT_2 \cap SPT_5$= {root, A, AC}
$SPT_2 \cup SPT_5$= {root, A, B, C, AC, CB, ACB, D, BC, BD, CD}
Normalized similarity measure between trees SPT_2 and $SPT_4 = \frac{SPT_2 \cap SPT_5}{SPT_2 \cup SPT_5} = \frac{3}{11}$ = 0.272

$SPT_2 \cap SPT_6$ = {root, A, AC}
$SPT_2 \cup SPT_6$ = {root, A, B, C, AC, CB, ACB, AD}
Normalized similarity measure between trees SPT_1 and $SPT_2 = \frac{SPT_2 \cap SPT_6}{SPT_2 \cup SPT_6} = \frac{3}{8}$ =0.37

$SPT_3 \cap SPT_4$ = {root, B, BC, BD}
$SPT_3 \cup SPT_4$ = { root, B, C, D, BC, BD, CD }
Normalized similarity measure between trees SPT_1 and $SPT_2 = \frac{SPT_3 \cap SPT_4}{SPT_3 \cup SPT_4} = \frac{4}{7}$ =0.57

$SPT_3 \cap SPT_5$ = {root}
$SPT_3 \cup SPT_5$ = { root, B, BC, BD, A, AB, AC }
Normalized similarity measure between trees SPT_1 and $SPT_2 = \frac{SPT_3 \cap SPT_5}{SPT_3 \cup SPT_5} = \frac{1}{7}$ =0.15

$SPT_3 \cap SPT_6$ = {root}
$SPT_3 \cup SPT_6$ = { root, B, BC, BD, A, AC, AD }
Normalized similarity measure between trees SPT_1 and $SPT_2 = \frac{SPT_3 \cap SPT_6}{SPT_3 \cup SPT_6} = \frac{1}{7}$ = 0.15

$SPT_4 \cap SPT_5$ = {root}
$SPT_4 \cup SPT_5$ = { root, B, C, D, BC, BD, CD, A, AB, AC}
Normalized similarity measure between trees SPT_1 and $SPT_2 = \frac{SPT_4 \cap SPT_5}{SPT_4 \cup SPT_5} = \frac{1}{10}$ =0.1

$SPT_4 \cap SPT_6$ = {root}
$SPT_4 \cup SPT_6$ = { root, B, C, D, BC, BD, CD, A, AC, AD }
Normalized similarity measure between trees SPT_1 and $SPT_2 = \frac{SPT_4 \cap SPT_6}{SPT_4 \cup SPT_6} = \frac{1}{10}$ =0.1

$SPT_5 \cap SPT_6$ = {root, A, AC}
$SPT_5 \cup SPT_6$ = { root, A, AB, AC, AD}

Normalized similarity measure between trees SPT_1 and $SPT_2 = \frac{SPT_5 \cap SPT_6}{SPT_5 \cup SPT_6} = \frac{3}{5}$ = 0.6

Based on the highest similarity measure values sequential probability tree T_5 are clustered, and the SPT_3 and SPT_4 are finally SPT_1 and SPT_2 are clustered. Among all these formal clusters we repeat the same process to construct larger clusters.

III. PROBLEM DEFINITION

Trajectory of the user is represented by a sequence < l_1, l_2, l_3, …….. , l_n >, where l_i denotes the locations, $1 \leq I \leq n$ [1]. Initially trajectory contains many raw data locations. Processing of raw data locations is closely in terms of computations. Trajectories containing raw data locations are converted into trajectories containing hot regions. Frequently visiting raw data locations are called hot regions. Many methods exist to find hot regions from raw data locations. Grid-based method is one of the best techniques to find hot regions [1].

In the modified new procedure each trajectory is represented as a sequence of hot regions. Authors have proposed a new procedure to develop a data list of trajectory profiles of users to capture movement based behaviors of user communities. In this new procedure trajectory profiles of users are represented as a set of sequential patterns. Sequential pattern contains frequent sequences of hot regions. Number of sequential patters in many real life applications is very large and also very difficult to capture all the transition probabilities of hot regions. Transitions probabilities associated with profiles of user trajectories are very large and may not be possible to capture from a set of sequential patterns. Authors have proposed a new tree structure called sequential probability tree (SP-tree) to represent trajectory profile of a user.

A sequential probability tree (sp-tree) is a special prefix multi-way tree that contains one root node (root) and a set of tree nodes. Each hot region is represented by an edge of sp-tree. Each node of a sp-tree is labeled by a special systematic string s_1, s_2, s_3 ….. , s_k labeling starts from root to all other tree nodes. K is the length of the string from root to any other node s. $s_i \in \sum$ where $\sum$ represents the set of hot regions. T represents the set of user trajectory profiles. The K-length trajectory is represented by a string s_1, s_2, s_3 ….. , s_k. Each nodes stores support(s) and a conditional table. Support(s) represents proportion of its present count from a set of total trajectories and its value is support(s) $\in$ [0, 1].

$$\text{Support(s)} = \frac{\text{Number of trajectories that contian string in s as subsequencies}}{\text{Total number of trajectories}}$$

The conditional table contains two attributes-Row id, C.Prob conditional probability table at a node ‘s’ represents user next move probability corresponding to the traversal sequences from the node root to s.

Breadth-First algorithm for constructing sequential probability tree:

Breadth first sequential probability tree (BFSPT) accepts three inputs: A set of trajectories of a single user, minimum support and minimum probability. Each sequential probability tree (SPT) stores and manages trajectory details of one particular user profiles. Sequential

probability tree is constructed in a level by level manner in breadth first approach.

At the very beginning sequential probability tree contains only one root node with the conditional probability table (CPT). CPT stores the probabilities of hot regions such that the probability of each hot region is greater than a minimum probability threshold (Minimum-Probability). Minimum-support and minimum probabilities are context dependent and are specified by experts. In the second level a new node is created corresponding to the node whose support value of a hot region at the root. After all the nodes in the second are created same process is repeated to create all possible nodes in the third level, fourth level, etc. In each level of the sequential probability three first find frequent hot regions and construct conditional probability table for each node. The find and create child nodes in the $(i+1)^{th}$ level corresponding to each node in the i^{th} level whose minimum support is greater than the minimum-support.

Sequential probability tree construction procedure for a single user, u_1 in the fig-1 with minimum-support is 0.4 and minimum-probability is 0.3. Initially, in the very beginning, the root node contains empty sequential patterns. It is represented as S_0 = {root}. Conditional probability table of root represents three frequent hot regions A, B and C of U_1^1 s trajectories. Support values of hot regions A, B and C are 3/4, 3/4 and 4/4 respectively.

Number of trajectories in which A present

Support of A = $\frac{\text{Number of trajectories in which A presents}}{\text{Total number of trajectories}}$ =3/4 = 0.75

Support of B = $\frac{\text{Number of trajectories in which B presents}}{\text{Total number of trajectories}}$ =3/4 = 0.75

Support of C = $\frac{\text{Number of trajectories in which C presents}}{\text{Total number of trajectories}}$ = 4/4 = 1.0

Conditional probability of root node are calculated as follows: U_1

Total number of hot regions of U_1^1 s trajectories at the root node = 10

Conditional probability of hot region A= $\frac{\text{Number of times hot region A appears in all the trajectories of U1}}{\text{Total number of trajectories of user U1}}$ = 3/10 = 0.3

Conditional probability of hot region B = $\frac{\text{Number of times hot region B appears in the trajectories of user U1}}{\text{Total number of trajectories U1}}$ = 3/10 = 0.3

Conditional probability of hot region C = $\frac{\text{Number of times hot region C appears in the trajectories of user U1}}{\text{Total number of trajectories U1}}$ = 4/10 = 0.4

Conditional probability values of hot regions conditional probability table of root are {0.3, 0.3, 0.4}.

Conditional probability table of root node contains three hot regions (A, B and C) because conditional probabilities of A, B and C are greater than minimum-probability. Root is in the first level and we must create three child nodes to root node. That is, for each frequent hot region, text whether the frequent hot region is in the conditional probability table of root or not. Conditional probability table of root nodes has three hot regions, whose conditional probability table of root node has three hot regions, whose conditional probability is greater than the minimum-probability value, hence three children of the root node will be created. That is S_1 = {A, B, C}. S_2 will be created from S_1 and S_3 will be created from S_2 and so on.

The frequent hot region of node A are B and C and the corresponding projected trajectory profile data sets are {(B, C), (B, C), (C)}. Conditional probability table of node A contains 2 entries because conditional probability of both B and C is greater than minimum conditional probability is (0.3). Two children for node A will be created because support values of both B and C are also greater than minimum-support (0.4). Children of A are labeled as AB and AC. Hence S_3 = {AB, AC} corresponding to A.

In the same fashion conditional probability table of B in level 1 contains only one hot region in (C), whose conditional probability and support are greater than minimum conditional probability and minimum support projected trajectory data set for B are {(C), (C), (C)}. A new node, BC, will be created for the node B at level 1. Nodes in the level 3 are S_2 = {AB, AC, and BC}. At level 1, projected trajectory for node C is empty as there are no hot regions starting from C.

In the level 2 only AB contains one hot region with both conditional probability and support greater than minimum support and minimum conditional probability values hence a new node {A, B, C} will be created as a child node for AB, and S_3 ={A, B, C}.

For the given example-1 the sequential probability is shown in Fig-2. Nodes of the sp-tree are represented as root, A, B, C, AB, AC, BC and ABC. Final tree has four levels S_0= {root}, S_1 = {A, B, C}, S_2 = {AB, AC, BC} and S_3= {ABC}.

Clustering of users based on the movement based community details:

Users are clustered based on their trajectory profiles. Trajectory profiles of each user is represented by one sequential probability tree [1]. A trajectory profile of n-users is represented by n sequential probability trees. Once all sequential probability trees are constructed, then the next goal is to cluster all these users into groups such that users in the same group have certain types of similarity movement features and users between the two different groups have certain types of (many) dissimilar features. Verities of similarity measure details are presented and these similarity measures are used in clustering users.

Different types of similarity functions considered in sequential probability trees:

Movement details of users are represented in sequential probability trees. Each sequential probability tree represents on user profile. That is, each sequential probability tree stores and maintaining sequential patterns as well as transition probabilities. In framing different types of similarity measures of users, many structured information details of sequential probability trees are considered. Most important similarity measures that are considered are:

1. *Number of common tree nodes in sequential probability trees:*

   Every node in a sequential probability tree represents a full or partial frequent sequence of hot regions of the user's trajectories. The similarity measure strength increases as the number of common nodes between two sequential probability trees increases and it results the increase in movement behavior of the two respective users.

2. *The number of total nodes of sequential probability trees :*

   Total number of nodes in the sequential probability trees is also considered as one of the important similarity measure between two movement details of users. Similarity measure between two parts of the two different sequential probability trees increases as the total number of nodes with the same sequence of patterns increase. The length of sequential patterns increases as the total number of nodes in the sequential probability trees increases.

3. *Number of distinct nodes of sequential probability trees:*

   Every node in the sequential probability tree from top to bottom represents a specific sequential movement pattern of a particular user. Similarity function can also be taken as a function of different nodes of sequential probability trees. Also note that two users may have the same common nodes but with different movement coverage ranges. Assume that the user U_1 may have a set of different movement coverage ranges with larger path length. Also, assume that the user U_2 also have other set of different movement coverage ranges with small path lengths. Due to the difference in many path lengths, movement coverage ranges also differ between two different users.

4. *Support values of sequential probability trees :*

   Every node in the sequential probability tree is assigned a support value, which indicates the frequency of frequent sequential pattern that appears in the trajectory profile of the user. Movement behaviour of the user changes as the support values of nodes changes. The movement behaviours of the two different users come close to near when the corresponding support values of two common nodes approach very close to each other. Support value is directly proportional to the sequential pattern. The length of sequential pattern increases as the support value increases. The importance of sequential pattern increases when the support value increases. Support is the most important similarity measure and it is popularity used in many applications of movement based related tasks. Support is the most important and most frequently used similarity measure in many movement based applications that are mainly based on trajectory profiles of users. We can formulate many similarity measurements in many number of ways using support values. Different support usage formulas will give different ways of measuring similarity strength between users.

5. *Conditional probability details of nodes in the sequential probability trees :*

   Each node in the sequential probability tree is associated with a conditional set of probability table. The conditional probability table represent s set of probability such that each probability indicates the frequency of the movement from the current node to one of its childrens. These probabilities are called transition probabilities. Transitional probabilities explain how the next movement occurs from the current node. Transitional probabilities represent more detailed movement based details of users. It may be possible that the same type of two sequential patterns may have different sequential probability table are one of the most important and frequently used popular similarity measure used to compare two different user profiles or trajectories. Comparing two sequential probability trees gives more accurate result when conditional probabilities in the conditional probability tables are used.

Detailed explanation of computing similarity measurements:

Similarity measurement values are computed for each pair of common tree nodes of two different sequential probability trees based on their support and conditional probability values. Most popular and most important two similarity measurement functions or techniques are similarity_N and similarity_T. The similarity function similarity_{SP} (.) is obtained by summarizing the similarity scores and then normalizing the similarity scores based on the number of nodes of respective sequential probability trees. To make the all comparisons very easy between any two sequential all comparisons very easy between any two sequential probability trees all the similarity measurement scores are normalized and this normalization helps us to manipulate and maintain all the desired results in a systematic and in a convenient way.

Let us consider SPT_i and SPT_j are two sequential probability trees of two different users. Nodes in the i^{th} tree with the sequential pattern s are represented by N_i^s. Similarity the nodes in j^{th} sequential probability tree are represented by N_j^s. For any given two nodes N_i^s and N_j^s in the different sequential probability trees, the similarity measure is defined by the equation (1) Similarity_N (N_i^s, N_j^s) =

$$\begin{cases} 1, & if\ s = root \\ \left(1 - \left|support(N_i^s) - (N_j^s)\right|\right) * \frac{support(N_i^s) + support(N_i^s)}{2}, & \text{otherwise} \end{cases} \quad (1)$$

For two distinct sequential probability trees consider the two distinct nodes each node taken from a separate sequential probability tree and also consider the common sequential pattern. The first term explains the closeness of their support values. If support values of these nodes are very close to each other, then the difference between them becomes very smaller. When the first term is very large it indicates that the two support values are very close to each other. The second term in the equation (1) represents the weights of their support values. Clearly, the nodes with the

large support values are very important and they are considered first in calculating the similarity measurements. Also, this particular measure is very useful and very important it actually tells us how important the average support of these selected nodes. Usually, the nodes with the larger support values are more important in finding similarity scores.

Consider the two nodes $N_i^s=N_1^{BC}$ and $N_j^s=N_3^{BC}$ in SPT_1 and SPT_3 sequential probability trees. Similarity measure, $Similarity_N(N_1^{BC}, N_3^{BC}) = (1-|0.75-0.5|)*\frac{0.75+0.5}{2}$ = 0.47. The conditional probability table of a node N_i^s is used to store all the probability corresponding to the next movements from the current node to its children nodes where s is the sequential pattern of the current node. When two common node are considered from two different sequential probability trees and when the differences between these two conditional probability tables is very less then it is the indication that these two common nodes are more similar in terms similarity measurement scores and hence corresponding user movement are similar. Next movement details of children nodes are provided only to the nodes whose minimum probability condition is satisfied. Each conditional probability table satisfies probability distribution rules. Probability distribution rules or formulas applicable to evaluate the similarity measurement score values between two conditional tables. We assume that the conditional probability table of a node N_i^s is C_i^s and the conditional probability table of a node N_j^s is C_j^s .The similarity measurement score of these two conditional probability tables is defined as:

$$Similarity_T(C_i^s, C_j^s) = 1 - \frac{|\sum_{s\in C_i^s \cup C_j^s} |Pr(C_i^s(\rho)) - Pr(C_j^s(\rho))|}{|C_i^s \cup C_j^s|} \quad (2)$$

IV. ALGORITHMS

Breadth First Algorithm for constructing Sequential Probability Tree:

Algorithm 1: Breadth First Method (BFM)

Input:

1. A set of user profiles, T, represented as transformed trajectories.
2. A minimum conditional Probability threshold (minimum probability).
3. A minimum support threshold (minimum support).

Output:

A sequential probability tree that represents profile details of a single user.

1.Root = null //A root node with null entry
2.So ={root} //At the very beginning only root is there
3.K=0
4.While ($S_k \neq 0$) do //While the set S_k contains elements do
5. $S_{k+1}=0$ // S_{k+1} is to store node names in the next level
6. For each node s in the set S_k do
7. Find frequent hot regions and then create conditional table of node s
8. For each σ in frequent hot regions do
9. If σ is in conditional table of node is s then
10. Create a new trajectory set of s σ
11. Sσ is a child of s , so add node sσ into S_{k+1}
12. End if
13. End for
14. End for
15. k = k + 1
16. End while

Depth First Algorithm for constructing sequential probability tree:

Algorithm 2: Depth First Method (DFM)

Input:

1. A set of user profiles, T, represented as transformed trajectories.
2. A minimum conditional Probability threshold (minimum probability)
3. A minimum support threshold (minimum support)

Output: A sequential probability tree that stores all the profile details of a single user.

1. Find frequent hot regions at the root node, s
2.Create conditional probability tale of root node, s
3.For each σ in frequent hot regions do
4. If σ is in conditional probability table of a nodes s then
5. sσ is a child of s
6. Create transformed trajectory set T^1
7. Depth First Method (T^1 , sσ, minimum probability, minimum support)
8. End if
9. End for

This algorithm calls itself recursively at each level and at each node in that level.

Algorithm 3: New-Clustering :

Input:

1.A set of sequential probability trees {SP-tree$_1$, SPtree$_2$, SP-tree$_3$……, SP-tree$_n$}
2.Minimum similarity bound,δ

Output: A set of clusters representing user communities.

1.Construct a new connection graph G = (V, E) by using SP-tree$_1$, SP-tree$_2$, SP-tree$_3$,……, SP-tree$_n$ and δ
2.Initially consider all nodes of U as separate clusters.
3.Previous cost= Total cost of cluster formation
4.Test=True
5.While (Test=True) do
6. Test =False
7. Initialize two clusters X_i, X_j each to empty
8. Minimum cost= Previous cost
9. For each cluster C_i, C_j in the set c do
10. Present cost= previous cost+| C_i * C_j| - 2* Intercost (C_i, C_j)
11. If (present cost ≤ minimum cost) then
12. Test =True
13. Store C_i in X_i and C_j in X_j
14. End if
15. End for
16. If (Test=True) then
17. Combine X_i and X_j into a single cluster

18. Previous cost = minimum cost
19. End if
20. End while

Algorithm 4: Proposed-Clustering

Input:

1. A set of n sequential probability trees
2. Minimum threshold to combine clusters

Output :

A set of clusters.

1.Initially traverse a set of n sequential probability trees.
2.Find all the names of nodes of each tree and store node names
of each tree separately.
3.Initialize each tree as one cluster.
4.While for each cluster id
5. For each cluster j = (i+1) to n do
6. Calculate similarity measures of clusters i and j
and then store all these measures
7. End for j
8. End for i
9. While (similarity measure is > minimum threshold) do
Combine two clusters whose similarity measure is maximum
10. End of while
11. End of while

VII. CONCLUSION

A new algorithm is proposed to cluster movement based users based on trajectory profiles of different users. Movement based clustering procedure or method mainly consists of three steps.

- Constructing sequential probability trees to store all the details of user trajectory profiles [1].
- Based on the stored details of users in the sequential probability trees different similarity measurement scores are calculated [1].
- Based on the calculated similarity measurement scores movement-based user details are clustered [1].

First sequential probability trees are constructed to store all the movement based details of users. Two special methods are used to construct sequential probability trees. The first method is called breadth first sequential probability tree construction and the second method is called depth first sequential probability tree construction. Different types of measurement scores such as support, conditional probability values, number of similar nodes and number of distinct nodes etc are used. Finally a new clustering method called Geo-cluster is used to cluster movement based details of users.

REFERENCES

[1] When-Yuan Zhu,, Wen-Chih Peng , Member, IEEE, C. C. Hung, P.R. Lei, and L.J. Chen, Senior Member, IEEE "Exploring Sequential Probability Tree for Movement-Based Community Discovery" IEEE Transactions on Knowledge and Data Engineering, Nov. 2014.

[2] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, third ed. Morgan Kaufmann, 2011.