Probability Trees for Sequential Patterns of user Details

K. Dileep Kumar
M.Tech Student, Department of CSE
KMM Institute of Technology and Sciences
Tirupati, India

C. Sudarsana Reddy
Assistant Professor, Department of CSE
KMM Institute of Technology and Sciences
Tirupati, India

***Abstract*–Clustering is an important technique in data mining. Movement based communities (MBC) of users is one important clustering technique. Finding movement based communities of users are very useful in many real life applications such as location based services, tracking churning behaviors of cell-phone users, outlier detection, and trajectory devices etc. Important steps in movement based communities are creating trajectory profiles of users, finding similarity between trajectory profiles, and finding movement based communities using a clustering technique.**

A new multi-way tree structure is designed to represent trajectory profiles of users. This tree structure represents transition probabilities in addition to the sequential patterns of user profiles. Tree is constructed using breadth first or depth first algorithm.

Authors have proposed a new clustering technique to cluster all the movement based communities based on the similarity values derived from trajectory profiles of users. The proposed greedy algorithm Geo-cluster effectively derives movement based communities. Experiments conducted on real data sets reveal that the experimental results are effective in finding movement based user community cluster.

***Keywords*–Movement based communities, community cluster, Sequential probability tree.**

I. INTRODUCTION

In many real life applications finding location details of users is very useful for decision making. Many portable devices are available to find positions of users. A trajectory is a sequence of collected positions of a particular user. Each trajectory corresponds to the actual movement of a particular user in his/her real life. A community is a group of trajectories user communities are determined from user locations or positions collected during different time intervals. These communities are called movement based communities. User locations (positions) are called regions.

With the help of movement based community it is possible to find similar movement behaviors of groups of people.

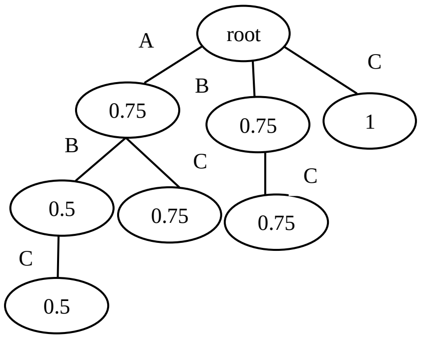
Each trajectory is represented by a collection of hot regions. More frequently visiting regions are called hot regions. Movement based communities are very useful in managing many applications such as GPS (Global Positioning System), finding the specific tower to forward a phone call, Identifying the location of a specific object, Trajectory ranking, Finding a service centre at a particular place and at a particular time.

II. EXAMPLES

T_1: <A, B, C> T_2: <A, B, C>

T_3: <B, C> T_4: <A, C>

(a) User U_2's trajectories



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

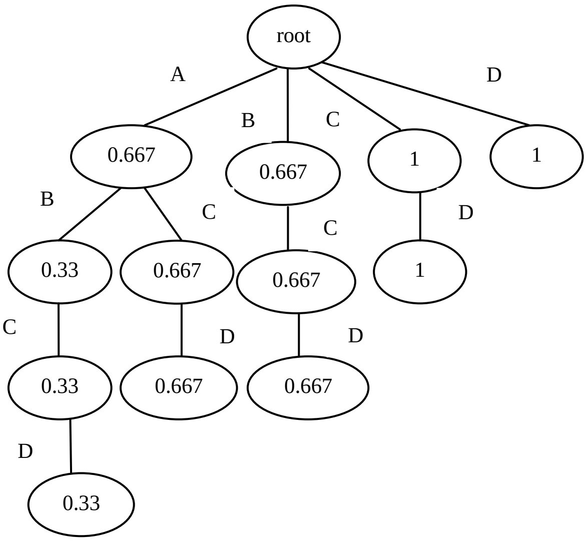
AB

RID	C.Prob
C	1

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

T_1: <A, B, C, D> T_2: <A, C, D>
T_3: <B, C, D>

(a) User U_2's trajectories



root

RID	C.Prob
A	0.2
B	0.2
C	0.3
D	0.3

A

RID	C.Prob
B	0.5
C	0.5

B

RID	C.Prob
C	1

C

RID	C.Prob
D	1

AB

RID	C.Prob
C	1

AC

RID	C.Prob
D	1

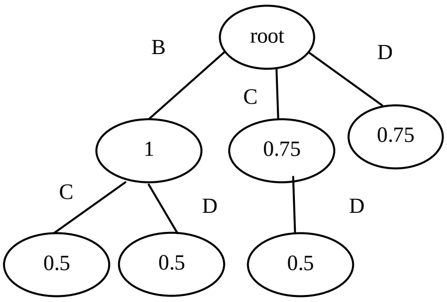
BC

RID	C.Prob
D	1

ABC

RID	C.Prob
D	1

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



root

RID	C.Prob
B	0.4
C	0.3
D	0.3

B

RID	C.Prob
C	0.67
D	0.33

C

RID	C.Prob
B	0.5
D	0.5

D

RID	C.Prob
B	1

BC

RID	C.Prob
D	1

Fig. 3. User SP-tree SPT_3

Trajectory of the user is represented by a sequence < l_1, l_2, l_3, …….. , l_n>, where l_i denotes the locations, 1 ≤ I ≤n. 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.

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 €∑ where ∑ 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) € [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 ith 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_1^1s 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 = $\frac{\text{Num ber of times hot region C appears in the trajectories of user U1}}{\text{Total number of trajectories U1}}$ = 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. 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) $\text{Similarity}_N(N_i^s, \quad N_j^s) \quad =$

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

otherwise (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, $\text{Similarity}_N(N_1^{BC}, N_3^{BC}) = (1-|0.75-0.5|)*\frac{0.75+0.5}{2} = 0.47$. 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) =

$$1- \frac{|\sum_{s\in C_i^s \cup C_j^s} |\Pr\left(C_i^s(\rho)\right)-\Pr(C_j^s(\rho))|}{|C_i^s \cup C_j^s|} \qquad (2)$$

IV. ALGORITHMS

Breadth First Algorithm for constructing Sequential Probability Tree:

Algorithm: 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.

```
Root =null      //A root node with null entry
So ={root}      //At the very beginning only root is there
K=0
While ($S_k \neq 0$) do     //While the set $S_k$ contains elements do
    $S_{k+1}$=0      // $S_{k+1}$ is to store node names in the next level
    For each node s in the set $S_k$ do
        Find frequent hot regions and then create conditional table of node s
        For each $\sigma$ in frequent hot regions do
            If $\sigma$ is in conditional table of node is s then
                Create a new trajectory set of s $\sigma$
                S$\sigma$ is a child of s , so add node s$\sigma$ into $S_{k+1}$
            End if
        End for
    End  for
    k  = k + 1
End while
```

Depth First Algorithm for constructing sequential probability tree:

Algorithm for 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.

```
Find frequent hot regions at the root node, s
Create conditional probability tale of root node, s
For each $\sigma$ in frequent hot regions do
    If $\sigma$ is in conditional probability table of a nodes s then
        s$\sigma$ is a child of s
        Create transformed trajectory set $T^1$
        Depth First Method ($T^1$ , s$\sigma$, minimum probability, minimum support)
    End if
End for
```

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

Algorithm 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.

```
Construct a new connection graph G = (V, E) by using SP-tree$_1$, SP-tree$_2$, SP-tree$_3$,……, SP-tree$_n$ and $\delta$
Initially consider all nodes of U as separate clusters.
Previous cost= Total cost of cluster formation
Test=True
While (Test=True) do
    Test =False
    Initialize two clusters $X_i$, $X_j$ each to empty
    Minimum cost= Previous cost
    For each cluster $C_i$, $C_j$ in the set c do
        Present cost= previous cost+| $C_i$ * $C_j$| - 2* Intercost ($C_i$, $C_j$)
        If (present cost $\leq$ minimum cost) then
            Test =True
            Store $C_i$ in $X_i$ and $C_j$ in $X_j$
        End if
    End for
    If (Test=True) then
        Combine $X_i$ and $X_j$ into a single cluster
        Previous cost = minimum cost
    End if
End while
```

V. SIMILARITY MEASURES

Time complexity details to compute the similarity, Similarity$_{SP}$ (SPT$_i$, SPT$_j$) are discussed below. First, computer the similarity score of common nodes and their conditional probability tales first. Let us suppose that N_i^s and N_j^s are the common nodes in two different sequential probability trees SPT$_i$ and SPT$_j$. The similarity score of the two nodes similarity$_N$(N_i^s, N_j^s) is O(1). O(1) is a constant time complexity because this similarity is computed using support values. Time complexity for finding similarity score between two conditional probability tables of two distinct sequential probability trees is |∑| because in the worst case the conditional probability table contains at most |∑| entries. Assume that, the set of common tree nodes between two different sequential probability trees is $S_{i,\ j}$.

∴ Time complexity for finding similarity score by using supports two nodes N_i^s and N_j^s = O(1) and time complexity for finding similarity measurement score by using conditional probability tables between two different sequential probability trees (Sim(C_i^s, C_j^s)) is |∑|.

∴ Time complexity for computing Similarity_{SP} (SPT_i, SPT_j) = $|S_{i,\ j}|$ * O (1)*O (|∑|) = O($|S_{ij}|$|∑|).

The space complexity in computing the similarity score of each common node needs O(1) space. That is, space complexity in finding similarity score, Similarity_{SP} is O(1). Procedure for clustering different types of users based on sequential probability tree similarity measurement scores.

Many similarity functions are available to compute similarity measurement scores. Based on these similarity measurement scores authors have proposed a graph structures in order to represent movement similarity relationships of users. Authors have proposed a new algorithm for clustering all the users. A new objective function is formulated to evaluate the quality of clustering procedure. Two different users have a geo-connection relationship if the similarity measurement score of the corresponding trajectory profiles must be greater than a pre-specified threshold, . We can also say that δ is the minimum threshold similarity bound for movement based communities. Based on the geo-connection relationships among the users a graph called geo-connection graph is constructed. This geo-connection graph is represented as G=(V,E),where V={ ϑ_1 , ϑ_2, ϑ_3 ……, ϑ_n }.V represents set of users and each edge represents similarity score between two different users U_i and U_j. That is, E={($\vartheta_{i,}$ ϑ_j)|Similarity_{SP} (SPT_i, SPT_j)>δ}.

After constructing a geo-connection graph, the next task is to divide the users into components (clusters) where each component represents a set of nodes having a certain degree of movement similarity relationship. Within the geo-connection graph different objective functions are considered in order to evaluate the quality of the cluster results. Authors have used a special new technique to transform the graph into a set of perfect communities. These set of perfect communities are called cliques. Here objective function consists of two parts. First part is called intra cost score and the second part is called inter cost score. Intra-cost score represents the minimum number of edges added in order to make the component a clique. A clique is the largest complete sub-graph of a graph. Thus, intra cost of a community or component C_i=(V_i, E_i) is defined as Cost_{intra}(C_i) = $|K_{|Vi|}|$ - $|E_i|$, – (4) , Where $K_{|Vi|}$ is a $|V_i|$ -clique.

In the geo-connection graph terminology, the inter cost represents the minimum number of component edges removed from the geo-connection graph to make the component dissointed from each other. One way to formulate inter cost score between two components C_i = (V_i, E_i) and C_j = (V_i, E_i) is as follows.

Cost_{inter} (C_i, C_j) = |{ϑ_i, $\vartheta_{j)}$| $\vartheta_{i\ \epsilon}$ v_i, $\vartheta_{j\ \epsilon}$ v_j}| −(5)

In order to combine intra-cost and inter-cost scores for a given set of users C ={c_1, c_2, c_3…, c_n } the formula is derived as follows:

Cost_{total} (C) $= \sum_{C_i \in C} cost_{intra}\ (C_i) + \sum_{C_{i,}C_j \in C} cost_{inter}\ (C_i, C_j)$ − (6)

Authors have proposed a new algorithm called Geo-cluster for clustering all the users based on their similarity measurement scores. At the very beginning each vertex is viewed as a community. Total cost scores between two communities is calculated and repeat the same process for all the nodes two at a time. Candidate communities are selected and merged as long as the total cost is reduced. This algorithm continues until the total cost is not possible to reduce or none of the communities are expanded.

Evaluating the performance of proposed algorithm:

Data sets Description Details:

Authors have used Kstaxi data set for experimental verification. Also another data set that is used for location based social services is every trial. For the Kstaxi data set minimum support and minimum conditional probability details are 0.1 and 0.1 respectively and for every trail data set minimum support and minimum conditional probabilities are 0.2 and 0.2 respectively.

Comparison between two algorithms (Depth First sequential probability and Breadth first sequential probability tree):

Breadth First (BF) SP-Trees and Depth First (DF) SP-Trees are two different ways of constructing sequential probability trees. Breadth first method generally saves the memory storage for constructing the SP-Tree and it is very much useful for long and coherent trajectories. On the other hand depth first method was developed to construct the sequential probability trees corresponding to the users such that the trajectories are short and sparse. In general memory usage of BF is smaller than DF.

Comparing similarity measurements scores:

Two important metrics called entropy and purity are used to measure the quality of the classes are distributed in the resulting class in the resulting communities. There exists inversely proportional relationship between the entropy and the derived communities. When entropy is zero then communities are very good. Similarity derived communities are good when purity value is very high. Minimum support is a special measurement score that tells minimum number of sequential patterns appearing in trajectory profiles. Height of the sequential probability tree is inversely proportional to the minimum support.

If the minimum support value is large then the corresponding nodes of two different users in two different SP-Trees are placed in lower levels of SP-Trees which indicate larger similarity between users. In that case it may not be possible to find distinction between two different users and hence it is cumbersome to find communities which are distinct.

There exist various types of methods for finding or clustering community movement behaviours. Each method has its own advantages and disadvantages. Clique use movement based method is a simple and useful method for grouping users community wise based on social relationships. Hierarchical clustering method initially assumes each user as one community, and then gradually combines two communities. Agglomerative hierarchical clustering community behaviours. Initially at the very beginning it starts each user as a separate single community. After that it merges two communities based on the largest similarity measurement score. Between clustering method follows a different approach in forming community based clusters. Generally clique method gives good and reasonable results in many cases. Geo-cluster method creates less number of clusters than clique method. Clique method does not consider when the difference of two clusters is very small. All clustering methods need to know the number of clusters at the beginning where as Geo-cluster method need not know initially the number clusters.

VI. PROPOSED ALGORITHM

A new method is proposed for clustering user details which are represented by sequential probability trees. Initially each user is represented as a single cluster. Based on the similarity measurements which are represented as connection among the nodes where a node is a single user cluster. Based on the total degree of the nodes are clustered. We assume minimum total degree to combine clusters. The process of combining clusters is repeated until the minimum total degree.

```
Algorithm: Degree cluster
Input: Set of sequential probability trees of  'n' users.
Output: Set of clusters.
1.  Create n-sequential probability trees for n-users
2.  While minimum total degree condition true do
3.  For each of given sequential probability trees
4.          Compare total degrees of a pair of trees
5.          If the minimum total degree condition is satisfied
6.          Combine those two clusters.
7.          End If
8.  End For
9.  End While
10. Foreach cluster
11.         Display cluster details
12. End For
```

VII. CONCLUSION

A new procedure or method is formulated 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.
- Based on the stored details of users in the sequential probability trees different similarity measurement scores are calculated.
- Based on the calculated similarity measurement scores movement-based user details are clustered.

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. J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, third ed. Morgan Kaufmann, 2011.
2. W. Zhu, W. C. 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.
3. C.-C. Hung, C.-W. Chang, and W.-C. Peng, "Mining Trajectory Profiles for Discovering User Communities," Proc. Int'l Workshop Location Based Social Networks, Nov. 2009.
4. H. Cao, N. Mamoulis, and D.W. Cheung, "Mining Frequent Spatio- Temporal Sequential Patterns," Proc. Fifth IEEE Int'l Conf. Data Mining, Nov. 2005.
5. X. Cao, G. Cong, and C.S. Jensen, "Mining Significant Semantic Locations from GPS Data," Proc. VLDB Endowment, vol. 3, no. 1, pp. 1009-1020, Sept. 2010.
6. J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory Clustering: A Partition and Group Framework," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '07), June 2007.
7. I.X. Leung, P. Hui, P. Li, and J. Crowcroft, "Towards Real-Time Community Detection in Large Network," Physical Review E, vol. 79, no. 6, 066107, June 2009.
8. Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma, "Mining User Similarity Based on Location History," Proc. 16th ACM SIGSPATIAL Int'l Conf. Advances in Geographic Information Systems (GIS '08), Nov. 2008.
9. L. Liu, C. Andris, and C. Ratti, "Uncovering Cabdrivers' Behavior Patterns from their Digital Traces," Computers, Environment and Urban Systems, vol. 34, no. 6, pp. 541-548, Nov. 2010.