Mobility Patterns Based Clustering: A Novel Approach

Linh Hoang Tran and Loc Hoang Tran

***Abstract*—Clustering is the basic technique in data mining research field. However, there are just few mobility patterns based clustering techniques which are hierarchical clustering and k-means clustering. Moreover, these two techniques suffer from the so-called "curse of dimensionality". Hence in this paper, the spectral clustering methods and the novel power symmetric normalized spectral clustering method are proposed and these three methods are used to solve the mobility pattern based clustering problem. First, the novel similarity among mobility patterns is defined in the trajectory dataset. From this novel similarity, a similarity graph can be constructed. Finally, the three proposed clustering methods are applied to this graph. Experimental results show that the clustering results of the power symmetric normalized clustering method are more well-balanced than the clustering results of the un-normalized and symmetric normalized spectral clustering methods. Moreover, the time complexity of the power symmetric normalized clustering method is also lower than the time complexity of the two spectral clustering methods.**

***Index Terms*—Spectral clustering, graph Laplacian, similarity matrix, mobility patterns, power method.**

I. Introduction

Clustering is the data mining technique separating objects into groups [1], [2]. In the recent years, it's used to partitioning mobility patterns into different groups [3]. However, in our literature review, there are just few mobility patterns based clustering techniques have been proposed. In [4], the authors used hierarchical clustering technique to partitioning a set of mobility patterns into different groups. In [5], [6], the authors proposed the extended k-mean clustering technique. In details, in [5], [6], the authors try to combine both spatial similarity and temporal similarity to form a new similarity between two mobility patterns and then apply their extended k-mean clustering method to the trajectory dataset. We can easily see that the hierarchical clustering technique and the k-mean clustering technique suffer from the so-called "curse of dimensionality". It means that these methods are very computational expensive when they are applied to datasets that have high dimensions.

In this paper, a novel mobility pattern based clustering method will be developed. This method is the extended spectral clustering method. To the best of my knowledge, this work has not been investigated.

In specific, first, the new spatial similarity between two mobility patterns will be defined. It's a function of the Longest Common Subsequence (i.e. LCS) [7], [8] between two mobility patterns. Applications of LCS algorithm are huge (but old idea), specifically in bio-informatics research area. It's often used to measure the similarity between two DNA strings [9]. In the field mobility pattern mining, it's also used to measure the spatial similarity between two mobility patterns [10]. However, in [10], the authors do not try to combine the spatial similarity and the temporal similarity between two mobility patterns in order to solve the mobility pattern based clustering problem. In this paper, the LCS spatial similarity and our new defined temporal similarity of two mobility patterns will be combined to solve the clustering problem. This is the novel idea. We also point out that since the spatial similarity and the temporal similarity has the symmetric property, this will lead to the reduction of the time complexity of procedure computing the combined weighted similarity matrix. Second, the un-normalized spectral clustering method (i.e. the current state of the art graph based clustering method) [11], [12] and the symmetric normalized spectral clustering method [11], [13] will be applied to the weighted similarity matrix. These two methods will be served as the baseline methods. However, the time complexities of these two methods are high due to we need to compute the whole set of eigenvectors of the graph Laplacian. This is not practical for big data problems. Thus, finally, instead of computing the whole set of eigenvectors of the symmetric normalized graph Laplacian, the power method will be used to compute the largest "pseudo-eigenvector" of the symmetric normalized weighted similarity matrix. Then the k-mean clustering method can be applied to this largest "pseudo-eigenvector" of the symmetric normalized weighted similarity matrix. Information about the largest "pseudo-eigenvector" can be found in [14], [15] and in Section IV. This will lead to the significant reduction of the time complexity of the symmetric normalized spectral clustering method. This novel method will be called "the power symmetric normalized clustering method".

We will organize the paper as follows. Section II will present the preliminary definitions and notations used in this paper. Section III will present the un-normalized and symmetric normalized spectral clustering methods. Section IV will present the power symmetric normalized clustering method. Section V will show the experimental results of the un-normalized spectral clustering method, the symmetric normalized spectral clustering method, and the power symmetric normalized clustering method. Section VI will conclude this chapter and the future directions of researches will be discussed.

Manuscript received May 2, 2018; revised July 9, 2018.

Linh Hoang Tran is with Thu Dau Mot University, Vietnam (e-mail: linhtran.ut@gmail.com).

Loc Hoang Tran is with John von Neumann Institute, Vietnam (e-mail: tran0398@umn.edu).

II. Preliminary Definitions and Notations

Consider we are given a set of Points of Interests (i.e. PoIs).

doi: 10.18178/ijmlc.2018.8.4.717

Then the mobility of users can be represented by the un-directed graph. The set of PoIs is the set of vertices of the graph. Each vertex (i.e. PoI) of the graph is connected to all of its adjacent vertices by edges. The example of this undirected graph is shown is Fig. 1.

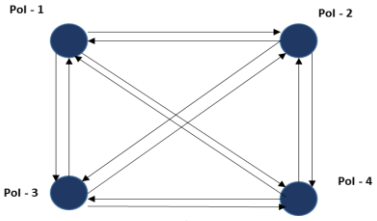

Fig. 1. Example of un-directed graph.

Let *c* be the ID number of the POI where the user is located at the predefined timestamp *t*. Hence *C* and *T* will be the set of IDs of PoIs and the set of timestamps respectively.

Let the point $p=(c,t)$ s.t. $c \in C$ and $t \in T$. Then the trajectory of the user can be defined as a finite sequence of points $< p_1, p_2, ..., p_n >$, where $p_j = (c_j, t_j)$ s.t. $c_j \in C, t_j \in T, 1 \le j \le n$.

Assume that we are given two mobility patterns $P_a =< p_1, p_2, ..., p_n >$ and $P_b =< p_{1'}, p_{2'}, ..., p_{n'} >$ with $n \ne n', p_i = (c_i, t_i), p_{i'} = (c_{i'}, t_{i'})$. Our first objective in this section is to define the spatial similarity and the temporal similarity between these two mobility patterns P_a and P_b.

Definition 1: Spatial Similarity

The spatial similarity between P_a and P_b can be defined as the following

$$w_{\text{space}}(P_a, P_b) = e^{-\frac{1}{LCS(P_a,P_b)+\delta}}, \quad (1)$$

where $LCS(P_a, P_b)$ is the longest common subsequence (i.e. LCS) between P_a and P_b and δ is any small positive number (i.e. to avoid the case $LCS(P_a, P_b) = 0$). The case $LCS(P_a, P_b) = 0$ means two mobility patterns P_a and P_b do not have any common PoI.

Next, the LCS algorithm will be presented as follows:

Algorithm 1: Longest common subsequence (LCS) between P_a and P_b

Input: P_a and P_b

Output: The length of the longest common subsequence between P_a and P_b

1. Initialize the integer array *C* with *n* rows and *n'* columns.
2. If $P_a.c_1 = P_b.c_1$
   C[1,1] = 1
   Else
   C[1,1] = 0
3. Let all the elements of the leftmost column of *C* be *C*[1,1].
4. Let all the elements of the top row of *C* be *C*[1,1].
5. For *i*=2:*n*
   For *j*=2:*n*'
   If $P_a.c_i = P_b.c_j$
   C[*i*,*j*] = *C*[*i*-1, *j*-1] + 1
   Else
   C[*i*, *j*] = max(*C*[*i*-1,*j*], C[*i*,*j*-1])
6. Return *C*[*n*,*n*']

Please note that $0 \le e^{-\frac{1}{LCS(P_a,P_b)+\delta}} \le 1$ and $LCS(P_a, P_b) = LCS(P_b, P_a)$.

Next, [5] have pointed out that the temporal similarity between P_a and P_b can be defined as the following

$$w_{\text{time}}(P_a, P_b) = \frac{1}{k} \sum_{i=1, j=1}^{n,n'} \frac{|P_a.t_i - P_b.t_j|}{\max(P_a.t_i, P_b.t_j)}, \quad (2)$$

where $P_a.c_i = P_b.c_j$ and *k* is the number of common PoI between P_a and P_b.

In this paper, we propose the novel temporal similarity between P_a and P_b as follows:

Definition 2: Temporal similarity

The spatial similarity between P_a and P_b P_b can be defined as the following

$$w_{time}(P_a, P_b) = \frac{1}{k} \sum_{i=1, j=1}^{n,n'} \frac{\min(P_a.t_i, P_b.t_j)}{\max(P_a.t_i, P_b.t_j)}, \quad (3)$$

where $P_a.c_i = P_b.c_j$ and *k* is the number of common PoI between P_a and P_b.

Please note that $0 \le w_{\text{time}}(P_a, P_b) \le 1$ and $w_{\text{time}}(P_a, P_b) = w_{\text{time}}(P_b, P_a)$.

III. Spectral Clustering

In this section, we will present the spectral clustering methods to partition a set of mobility patterns into groups. Assume we are given a set of mobility patterns $\{P_1, P_2, ..., P_m\}$. First, we need to construct the weighted similarity matrix $W \in R^{m*m}$. We have that

$$W = \alpha W_{\text{space}} + (1-\alpha) W_{\text{time}}, \quad (3)$$

where $w(i, j) = \alpha w_{\text{space}}(P_i, P_j) + (1-\alpha) w_{\text{time}}(P_i, P_j)$ with $w(i, j) \in W$ and $1 \le i, j \le m$.

Please also note that $0 \le \alpha \le 1$.

Because of the symmetric property of spatial similarity of temporal similarity, we just need to compute the upper diagonal part of the matrix *W* since this matrix *W* is symmetric. Then we can obtain the lower diagonal part of matrix *W* easily. This will reduce the time complexity of the procedure computing *W*.

Let *D* be diagonal matrix containing the degrees of vertices (of *W*) in its diagonal entries. Please note that *D* is the R^{m*m} matrix. Then

$$d(i) = \sum_{j=1}^{m} w(i, j)$$

Next, the un-normalized spectral clustering will be presented as follows:

Algorithm 2: Un-normalized spectral clustering algorithm

Input: The weighted similarity matrix *W* and the diagonal matrix *D*

Output: Clusters $C_1, C_2, ..., C_k$

1. Compute the un-normalized graph Laplacian $L = D - W$
2. Compute all eigenvalues and eigenvectors of *L* and sort all eigenvalues and their corresponding eigenvector in ascending order. Pick the first *k* eigenvectors $v_2, v_3, ..., v_{k+1}$ of *L* in the sorted list. *k* can be determined in the following two ways:
   a. *k* is the number of connected components of *L* [11]
   b. *k* is the number such that $\frac{\lambda_{k+2}}{\lambda_{k+1}}$ or $\lambda_{k+2} - \lambda_{k+1}$ is largest for all $2 \le k \le n$
3. Let $V \in R^{m*k}$ be the matrix containing the vectors $v_2, v_3, ..., v_{k+1}$ as columns
4. For $i = 1, 2, ..., m$, let $y_i \in R^k$ be the vector corresponding to the *i-th* row of *V*
5. Use k-mean clustering algorithm to cluster the points $(y_i)_{i=1,2,...,m}$ into clusters $C_1, C_2, ..., C_k$

Finally, the symmetric normalized spectral clustering will be presented as follows:

Algorithm 3: Symmetric normalized spectral clustering algorithm

Input: The weighted similarity matrix *W* and the diagonal matrix *D*

Output: Clusters $C_1, C_2, ..., C_k$

1. Compute the symmetric normalized graph Laplacian $L_{sym} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$
2. Compute all eigenvalues and eigenvectors of L_{sym} and sort all eigenvalues and their corresponding eigenvector in ascending order. Pick the first *k* eigenvectors $v_2, v_3, ..., v_{k+1}$ of L_{sym} in the sorted list. *k* can be determined in the following two ways:
   a. *k* is the number of connected components of L_{sym} [11]
   b. *k* is the number such that $\frac{\lambda_{k+2}}{\lambda_{k+1}}$ or $\lambda_{k+2} - \lambda_{k+1}$ is largest for all $2 \le k \le n$
3. Let $V \in R^{m*k}$ be the matrix containing the vectors $v_2, v_3, ..., v_{k+1}$ as columns
4. Form the matrix $T \in R^{m*k}$ from *V* by normalizing the rows to norm 1, that is set $T(i,:) = \frac{V(i,:)}{(\sum_{j=1}^{k} v_{ij}^2)^{\frac{1}{2}}}$
5. For $i = 1, 2, ..., m$, let $y_i \in R^k$ be the vector corresponding to the *i-th* row of *T*
6. Use k-mean clustering algorithm to cluster the points $(y_i)_{i=1,2,...,m}$ into cluster $C_1, C_2, ..., C_k$

At the end of this section, we will present how to choose the centroids for clusters that we obtained in the spectral clustering algorithm. This work has been shown clearly in [5]. Assume that we have cluster $X = (P_1, P_2, ..., P_p)$. For each mobility pattern in *X*, set $O_i = 0$ for $1 \le i \le p$. Next, we will compute $O_i = \sum_{j \ne i}^{p} w(i, j)$. Choose P_i such that O_i is maximized. Set P_i be the centroid of cluster *X*. In the future research, this centroid will be used for recommendation system for new users.

IV. Power Symmetric Normalized Clustering

Although the spectral clustering method is very popular in data mining research field, it's rarely used in practical big data problems. It will normally take $O(m^3)$ time to compute the whole set of eigenvectors of the graph Laplacian, where *m* is the number of data points in the dataset. In order to solve this difficulty, in [14], the authors used the power method to compute the largest "pseudo-eigenvector" of the random walk normalized similarity matrix $D^{-1}W$. The time complexity of the power method is $O(m^2)$.

In the other words, suppose that the set of eigenvalues and the set of their associated eigenvectors of $D^{-1}W$ are $\{\lambda_1, \lambda_2, ..., \lambda_m\}$ and $\{u_1, u_2, ..., u_m\}$ respectively. Please note that all the eigenvalues are sorted in descending order. The first and largest eigenvalue λ_1 is 1 and its associated eigenvector u_1 is a constant vector since $D^{-1}W$ is a stochastic matrix.

Let $v^{(0)}$ be a random vector. The power method will compute the largest "pseudo-eigenvector" like the following

$$v^{(t)} = (D^{-1}W)^1 v^{(t-1)} = (D^{-1}W)^2 v^{(t-2)} = ... = (D^{-1}W)^t v^{(0)}$$
$$= c_1 (D^{-1}W)^t u_1 + c_2 (D^{-1}W)^t u_2 + ... + c_m (D^{-1}W)^t u_m$$
$$= c_1 \lambda_1^t u_1 + c_2 \lambda_2^t u_2 + ... + c_m \lambda_m^t u_m$$

Hence we have

$$\frac{v^{(t)}}{c_1 \lambda_1^t} = u_1 + \frac{c_2}{c_1} \frac{\lambda_2^t}{\lambda_1^t} u_2 + ... + \frac{c_m}{c_1} \frac{\lambda_m^t}{\lambda_1^t} u_m$$

Power method will finally converge to the largest eigenvector u_1 which is useless for clustering since it's a constant vector. To solve this problem, in [14], the authors define the velocity at *t*+1 be the vector $\delta^{(t+1)} = | v^{(t+1)} - v^{(t)} |$ and the acceleration be the number $\varepsilon^{(t+1)} = \| \delta^{(t+1)} - \delta^{(t)} \|$. The authors will let the algorithm stop as soon as $\varepsilon^{(t+1)}$ is below some small positive threshold but before the algorithm converges to the largest eigenvector u_1 (i.e. the constant vector). The computed vector by this algorithm is called the largest "pseudo-eigenvector". One interesting thing is this largest "pseudo-eigenvector" is a linear combination of all eigenvectors of $D^{-1}W$.

Following the idea from [14], the power method is used to develop a novel clustering method. Instead of computing the largest "pseudo-eigenvector" of the random walk normalized similarity matrix $D^{-1}W$, the largest "pseudo-eigenvector" of the symmetric normalized similarity matrix $D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ is computed. Next, the following theorem will be proved

Theorem 1

λ is an eigenvalue of $D^{-1}W$ with eigenvector u if and only if λ is an eigenvalue of $D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ with eigenvector $w = D^{\frac{1}{2}} u$.

Proof: The theorem 1 can be proved easily by solving

$$D^{-\frac{1}{2}} W D^{-\frac{1}{2}} w = \lambda w \Leftrightarrow D^{-\frac{1}{2}} D^{-\frac{1}{2}} W D^{-\frac{1}{2}} = \lambda D^{-\frac{1}{2}} w$$
$$\Leftrightarrow D^{-1} W D^{-\frac{1}{2}} w = \lambda D^{-\frac{1}{2}} w$$

Let $u = D^{-\frac{1}{2}} w$, (in the other words, $w = D^{\frac{1}{2}} u$), we have

$$D^{-1} W u = \lambda u$$

This completes the proof.

From the above theorem 1, we see easily that $D^{-1}W$ and $D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$ have the same set of eigenvalues but the eigenvectors of $D^{-1}W$ are different from the eigenvectors of $D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$. This will lead to distinct clustering results when we apply the power method to the symmetric normalized similarity matrix $D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$. This work, to the best of my knowledge, has not been investigated.

The power symmetric normalized clustering method will be presented as follows:

Algorithm 4: The power symmetric normalized clustering

Input: The weighted similarity matrix W and the diagonal matrix D
Output: Clusters $C_1, C_2, ..., C_k$

1. Compute the symmetric normalized similarity matrix $D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$
2. Pick an initial random vector $v^{(0)}$
3. Set $t = 0$
4. Set $\delta^{(t)} = [0, 0, ..., 0]^T$
5. Set $\varepsilon = 1$
6. While $\varepsilon > 10^{-6}$

   Compute $v^{(t+1)} = \frac{D^{-\frac{1}{2}}WD^{-\frac{1}{2}}v^{(t)}}{\| D^{-\frac{1}{2}}WD^{-\frac{1}{2}}v^{(t)} \|}$

   Compute $\delta^{(t+1)} = | v^{(t+1)} - v^{(t)} |$

   Compute $\varepsilon = \| \delta^{(t+1)} - \delta^{(t)} \|$

   $t = t + 1$
7. End
8. Apply the k-mean clustering method to the largest "pseudo-eigenvector" $v^{(t+1)}$ to get cluster $C_1, C_2, ..., C_k$

In the end, we can also compute the centroids of the clusters. These centroids will be used for recommender system for users or will be used later for streaming power symmetric normalized clustering technique which will be our future researches.

V. Experiments and Results

In this paper, the trajectory dataset that is available from [16] is used. This dataset was derived from Yahoo Flickr Creative Commons 100M dataset [17]. For our clustering purpose, we just use two .csv files in the trajectory dataset. These two files are poi-Melb-all.csv and traj-noloop-all-Melb.csv.

The poi-Melb-all.csv file contains all the information about the Point of Interest (i.e. PoI). The information of PoIs contained in the poi-Melb-all.csv is:

- poiID: POI unique ID
- poiName: POI Name
- poiTheme: POI Categoy
- poiLat: POI Latitude
- poiLon: POI Longitude
- poiURL: URL of Wikipedia webpage that describes this POI
- poiPopularity: The popularity of POI, i.e. the number of distinct users that visited the POI

The file poi-Melb-all.csv file is shown in Table I.

From the above Table I, we see that there are total 88 PoIs in the dataset.

TABLE I: Sample of Poi-Melb-All.Csv File

poiID	poiName	poiTheme	poiLat	poiLon
0	Arts Precinct	City precincts	-37.82167	144.96778
1	Docklands	City precincts	-37.817	144.946
…				
87	Yarra River	Transport	-37.85194	144.90833

The traj-noloop-all-Melb.csv file contains all the information about users' trajectories. The information contained in the tra-noloop-all-Melb.csv is:

- userID: User ID
- poiID: PoI ID
- startTime: When a user starts to visit the PoI, approximated by the time the first photo taken by the user at that PoI
- endTime: When a user leaves the PoI, approximated by the time the last photo taken by the user at that PoI
- #photo: Number of photos taken at the PoI by the user
- poiDuration: Visit duration (in seconds) at the PoI

The file traj-noloop-all-Melb.csv file is shown in Table II.

TABLE II: Sample of Traj-Noloop-All-Melb.Csv File

userID	poiID	startTime	endTime
10058801@N06	25	1226726126	1226726126
10087938@N02	58	1205332532	1205332541
…			
99804259@N00	21	1396489522	1396489522

By a little programming, from the above Table II, we see that there are total 1000 users in the trajectory dataset. Our main task is to partitioning 1000 users into different groups based on their mobility patterns.

First, two arrays of lists are constructed. This first array has 1000 elements (i.e. 1000 users). Each element contains a list of PoIs that each user visits. The second array also has 1000 elements. Each element contains a list of timestamps when each user visits PoIs. Next the spatial similarity matrix and the temporal similarity matrix can be constructed. Finally, we combine these two matrices to form the weighted similarity matrix W.

Second, the un-normalized spectral clustering method, the symmetric normalized spectral clustering method, and the power symmetric normalized method are applied to this weighted similarity matrix to get the clustering results. We test these three methods with k=3, 4, 5 where k is the number of clusters. In this paper, the clustering accuracy performance measures are not measured.

TABLE III: Centroids for Three Clusters as Lists of poiIDs

Cluster 0	[26 66 28 58]
Cluster 1	[81]
Cluster 2	[6 50 42 84 40 ... 14]

Next, the clustering results of the un-normalized spectral clustering method will be shown. Please note that *idx* is the clustering indices of 1000 users.

For k=3, we have the clustering results as follows

$$idx = [2, 0, 2, 2, ..., 2, 2]$$

The centroids for these three clusters are shown in the following Table III and Table IV.

TABLE IV. CENTROIDS FOR THREE CLUSTERS AS LISTS OF NAMES of PoIs

Cluster 0	Luna Park Albert Park and Lake Melbourne Zoo Melbourne Grand Prix Circuit
Cluster 1	Capital City Trail
Cluster 2	Sports and Entertainment Precinct St Paul's Cathedral Supreme Court of Victoria Melbourne Central station Parliament House … Collins Street

For k=4, we have the clustering results as follows

$$idx = [0,1,0,0,...,0,0]$$

The centroids for these four clusters are shown in the following Table V and Table VI.

TABLE V: CENTROIDS FOR FOUR CLUSTERS AS LISTS OF POIIDS

Cluster 0	[6 50 42 84 40 … 14]
Cluster 1	[26 7]
Cluster 2	[49]
Cluster 3	[40 22 72]

TABLE VI: CENTROIDS FOR FOUR CLUSTERS AS LISTS OF NAMES of PoIs

Cluster 0	Sports and Entertainment Precinct St Paul's Cathedral Supreme Court of Victoria Melbourne Central station Parliament House … Collins Street
Cluster 1	Luna Park University of Melbourne
Cluster 2	Shrine of Remembrance
Cluster 3	Parliament House Queen Victoria Village Fitzroy Gardens

For k=5, we have the clustering results as follows

$$idx = [4,2,4,4,...,4,4]$$

The centroids for these five clusters are shown in the following Table VII and Table VIII.

TABLE VII: CENTROIDS FOR FIVE CLUSTERS AS LISTS of poiIDs

Cluster 0	[0]
Cluster 1	[4]
Cluster 2	[26 66 28 58]
Cluster 3	[22 41 41]
Cluster 4	[6 50 42 84 40 … 14]

TABLE VIII: CENTROIDS FOR FIVE CLUSTERS AS LISTS OF NAMES of PoIs

Cluster 0	Arts Precinct
Cluster 1	RMIT City
Cluster 2	Luna Park Albert Park and Lake Melbourne Zoo Melbourne Grand Prix Circuit
Cluster 3	Queen Victoria Village State Library of Victoria State Library of Victoria
Cluster 4	Sports and Entertainment Precinct St Paul's Cathedral Supreme Court of Victoria Melbourne Central station Parliament House … Collins Street

Next, the clustering results of the symmetric normalized spectral clustering method will be shown.

For k=3, we have the clustering results as follows

$$idx = [1,1,2,2,...,2,1]$$

The centroids for these three clusters are shown in the following Table IX and Table X.

TABLE IX: CENTROIDS FOR THREE CLUSTERS AS LISTS of poiIDs

Cluster 0	[50 70 71]
Cluster 1	[6 50 42 84 40 … 14]
Cluster 2	[23 32 85 45 48 26 35 71 57]

TABLE X: CENTROIDS FOR THREE CLUSTERS AS LISTS OF NAMES of PoIs

Cluster 0	St Paul's Cathedral City Square Federation Square
Cluster 1	Sports and Entertainment Precinct St Paul's Cathedral Supreme Court of Victoria Melbourne Central station Parliament House … Collins Street
Cluster 2	Royal Arcade General Post Office Southern Cross station Eureka Tower Royal Exhibition Building Luna Park Melbourne Town Hall Federation Square Melbourne Cricket Ground (MCG)

For k=4, we have the clustering results as follows

$$idx = [0,0,0,3,...,0,0]$$

The centroids for these four clusters are shown in the following Table XI and Table XII.

TABLE XI: CENTROIDS FOR FOUR CLUSTERS AS LISTS of poiIDs

Cluster 0	[6 50 42 84 40 … 14]
Cluster 1	[56 71 4 24 56]
Cluster 2	[26 14 35]
Cluster 3	[23 32 85 45 48 26 35 71 57]

TABLE XII: CENTROIDS FOR FOUR CLUSTERS AS LISTS OF NAMES of PoIs

Cluster 0	Sports and Entertainment Precinct St Paul's Cathedral Supreme Court of Victoria Melbourne Central station Parliament House … Collins Street
Cluster 1	Melbourne Multi Purpose Venue (Hisense Arena) Federation Square RMIT City Swanston Street Melbourne Multi Purpose Venue (Hisense Arena)
Cluster 2	Luna Park Collins Street Melbourne Town Hall
Cluster 3	Royal Arcade General Post Office Southern Cross station Eureka Tower Royal Exhibition Building Luna Park Melbourne Town Hall Federation Square Melbourne Cricket Ground (MCG)

For k=5, we have the clustering results as follows

$$idx = [1,1,1,1,...,1,1]$$

The centroids for these five clusters are shown in the following Table XIII and Table XIV.

TABLE XIII: CENTROIDS FOR FIVE CLUSTERS AS LISTS of poiIDs

Cluster 0	[26 25]
Cluster 1	[6 50 42 84 40 … 14]
Cluster 2	[51 40 51 51 29 31 44 76 9]
Cluster 3	[2 50 71 3]
Cluster 4	[50 50 50 35 50 … 71]

TABLE XIV: CENTROIDS FOR FIVE CLUSTERS AS LISTS OF NAMES of PoIs

Cluster 0	Luna Park Crown Casino and Entertainment Complex
Cluster 1	Sports and Entertainment Precinct St Paul's Cathedral Supreme Court of Victoria Melbourne Central station Parliament House … Collins Street
Cluster 2	St Patrick's Cathedral Parliament House St Patrick's Cathedral St Patrick's Cathedral Australian Centre for Contemporary Art NGV International Arts Centre Royal Botanic Gardens Bourke Street
Cluster 3	Government Precinct St Paul's Cathedral Federation Square Little Italy
Cluster 4	St Paul's Cathedral St Paul's Cathedral St Paul's Cathedral Melbourne Town Hall St Paul's Cathedral … Federation Square

Next, the clustering results of the power symmetric normalized clustering method will be shown.

For k=3, we have the clustering results as follows

$$idx = [0, 0, 0, 1, ..., 1, 2]$$

The centroids for these three clusters are shown in the following Table XV and Table XVI.

TABLE XV: CENTROIDS FOR THREE CLUSTERS AS LISTS of poiIDs

Cluster 0	[26 66 28 58]
Cluster 1	[25 27 42 48 69 32 18 23 2]
Cluster 2	[6 50 42 84 40 … 14]

TABLE XVI: CENTROIDS FOR THREE CLUSTERS AS LISTS OF NAMES of PoIs

Cluster 0	Luna Park Albert Park and Lake Melbourne Zoo Melbourne Grand Prix Circuit
Cluster 1	Crown Casino and Entertainment Complex Melbourne Aquarium Supreme Court of Victoria Royal Exhibition Building Carlton Gardens General Post Office Little Collins Street Royal Arcade Government Precinct
Cluster 2	Sports and Entertainment Precinct St Paul's Cathedral Supreme Court of Victoria Melbourne Central station Parliament House … Collins Street

For k=4, we have the clustering results as follows

$$idx = [3, 3, 2, 1, ..., 2, 0]$$

The centroids for these four clusters are shown in the following Table XVII and Table XVIII.

TABLE XVII: CENTROIDS FOR FOUR CLUSTERS AS LISTS of poiIDs

Cluster 0	[6 50 42 84 40 … 14]
Cluster 1	[36 41 50 55 71 81 50]
Cluster 2	[32 35 78]
Cluster 3	[26 7]

TABLE XVIII: CENTROIDS FOR FOUR CLUSTERS AS LISTS OF NAMES of PoIs

Cluster 0	Sports and Entertainment Precinct St Paul's Cathedral Supreme Court of Victoria Melbourne Central station Parliament House … Collins Street
Cluster 1	Old Melbourne Gaol State Library of Victoria St Paul's Cathedral Margaret Court Arena Federation Square Capital City Trail St Paul's Cathedral
Cluster 2	General Post Office Melbourne Town Hall Treasury Gardens
Cluster 3	Luna Park University of Melbourne

For k=5, we have the clustering results as follows

$$idx = [2, 2, 4, 1, ..., 3, 0]$$

The centroids for these five clusters are shown in the following Table XIX and Table XX.

TABLE XIX: CENTROIDS FOR FIVE CLUSTERS AS LISTS of poiIDs

Cluster 0	[6 50 42 84 40 … 14]
Cluster 1	[9 21 36 45 71 82]
Cluster 2	[49]
Cluster 3	[9 32 42 41]
Cluster 4	[26 66 28 58]

TABLE XX: CENTROIDS FOR FIVE CLUSTERS AS LISTS OF NAMES of PoIs

Cluster 0	Sports and Entertainment Precinct St Paul's Cathedral Supreme Court of Victoria Melbourne Central station Parliament House … Collins Street
Cluster 1	Bourke Street Queen Victoria Market Old Melbourne Gaol Eureka Tower Federation Square Flinders Street station
Cluster 2	Shrine of Remembrance
Cluster 3	Bourke Street General Post Office Supreme Court of Victoria State Library of Victoria
Cluster 4	Luna Park Albert Park and Lake Melbourne Zoo Melbourne Grand Prix Circuit

From the above table, we easily see that the clustering results of the power symmetric normalized clustering method

are more well-balanced than the clustering results of the un-normalized and symmetric normalized spectral clustering methods. In the other words, for the un-normalized and symmetric normalized spectral clustering methods, there exist some clusters that have only one element. This is not the case for the clustering results of the power symmetric normalized clustering method.

VI. Conclusions and Future Works

There are three main contributions in this paper. First, the novel similarity among mobility patterns in the trajectory dataset is defined. From this novel similarity, a similarity graph can be constructed. Second, the un-normalized and symmetric normalized spectral clustering methods are applied successfully to this graph constructed from the trajectory dataset. Finally, the novel power symmetric normalized clustering method is developed successfully. This work, to the best of our knowledge, has not been investigated. From the experimental results, we recognize that the clustering results of the power symmetric normalized clustering method are more well-balanced than the clustering results of the un-normalized and symmetric normalized clustering methods.

In the future, the p-Laplacian spectral clustering method (i.e. the very high time complexity method) can also be applied to the trajectory dataset. The p-Laplacian spectral clustering method is worth investigated because of its hard nature and because this method has never been applied to the mobility pattern based clustering problem. The Alternating Direction Methods of Multipliers (i.e. the ADMM) method will be employed to solve the p-Laplacian based clustering method. Moreover, the streaming spectral clustering method will also be developed and will be applied to the streaming trajectory dataset. This work, to the best of our knowledge, has not been investigated up to now.

References

[1] P. Berkhin, "A survey of clustering data mining techniques," *Grouping Multidimensional Data*, Springer Berlin Heidelberg, pp. 25-71, 2006.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.

[3] T. V. T. Duong and D. Q. Tran, "A fusion of data mining techniques for predicting movement of mobile users," *Journal of Communications and Networks*, vol. 17, no. 6, pp. 568-581, 2015.

[4] S. Sung, Y. Seo, and Y. Shin, "Hierarchical clustering algorithm based on mobility in mobile ad hoc networks," in *Proc. International Conference on Computational Science and Its Applications*, Springer Berlin Heidelberg, 2006, pp. 954-963.

[5] T. V. T Duong and D. Q. Tran, "Clustering mobility patterns in wireless networks with a spatiotemporal similarity measure," *International Journal of Innovative Computing, Information and Control (IJICIC)*, vol. 9, no. 11, pp. 4263-4284, 2013.

[6] T. V. T. Duong and D. Q. Tran, "Mobility prediction based on collective movement behaviors in public WLANs," in *Proc. Science and Information Conference*, 2015, pp. 1003-1010.

[7] M. Paterson and V. Dančík, "Longest common subsequences," in *Proc. International Symposium on Mathematical Foundations of Computer Science*, Springer Berlin Heidelberg, pp. 127-142, 1994.

[8] J. W. Hunt and T. G. Szymanski, "A fast algorithm for computing longest common subsequences," *Communications of the ACM*, vol. 20, no. 5, pp. 350-353, 1977.

[9] D. Gusfield, *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*, Cambridge University Press, 1997.

[10] D. Zeinalipour-Yazti, S. Lin, and D. Gunopulos, "Distributed spatio-temporal similarity search," in *Proc. the 15th ACM International Conference on Information and KNOWLEDGE management*, ACM, 2006, pp. 14-23.

[11] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395-416, 2007.

[12] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 9, pp. 1074-1085, 1992.

[13] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in Neural Information Processing Systems*, vol. 2, pp. 849-856, 2002.

[14] F. Lin and W. W. Cohen, "Power iteration clustering," in *Proc. the 27th International Conference on Machine Learning*, 2010, pp. 655-662.

[15] N. D. Thang, Y. K. Lee, and S. Lee, "Deflation-based power iteration clustering," *Applied Intelligence*, vol. 39, no. 2, pp. 367-385, 2013.

[16] dongwookim-ml/flickr-photo. [Online]. Available: https://github.com/arongdari/flickr-photo

[17] B. Thomee *et al.*, "YFCC100M: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64-73, 2016.

Linh H. Tran received the B.S. degree in electrical and computer engineering from University of Illinois, Urbana – Champaign in 2005, M.S. and PhD. in computer Engineering from Portland State University. Currently, he is working as a lecturer at the Faculty of Electrical-Electronics Engineering, Ho Chi Minh City University of Technology. His research interests include low power, high speed integrated circuit design, quantum/reversible circuit and data-mining.

Loc H. Tran completed his bachelor of science and master of science in computer science at University of Minnesota in 2003 and 2012 respectively. Currently, he's a researcher at John von Neumann Institute, Vietnam and Thu Dau Mot University.