IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 30, 2019, accepted July 25, 2019, date of publication July 30, 2019, date of current version August 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931937

Social Relationships and Temp-Spatial Behaviors Based Community Discovery to Improve Cyber Security Practices

JIUXIN CAO[1], WEIJIA LIU[1], BIWEI CAO[2], PAN WANG[3], SHANCANG LI[4], (Senior Member, IEEE), BO LIU[5], AND MUDDESAR IQBAL[6]

[1]School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China
[2]Department of Computer Science, The Australia National University, Canberra, ACT 0200, Australia
[3]Southeast University-Monash University Suzhou Joint Graduate School, Suzhou 215123, China
[4]Department of Computer Science and Creative Technologies, University of the West of England, Bristol BS16 1QY, U.K.
[5]School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
[6]School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K.

Corresponding author: Shancang Li (shancang.li@uwe.ac.uk)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772133 and Grant 61472081, in part by the National Social Science Foundation of China under Grant 19@ZH014, in part by the Jiangsu Provincial Key Project under Grant BE2018706, in part by the Jiangsu Provincial Key Laboratory of Computer Networking Technology, in part by the Jiangsu Provincial Key Laboratory of Network and Information Security under Grant BM2003201, and in part by the Key Laboratory of Computer Network and Information Integration, Ministry of Education, China, under Grant No.93k-9.

ABSTRACT Cyber security significantly relies on the dynamic communities in social networks. The location-based social network (LBSN) is a new type of social system that has sprung up recently that. It turns traditional social networks into heterogeneous networks by incorporating location information, which is used as the medium between the real world and the online social networks, thus bringing new challenges to the community discovery problems. This paper proposes a LBSN homogeneous network model (LSHNM) based on the user social relations and temp-spatial behaviors to calculate the user similarity relations in multi-dimensional features and construct LBSN isomorphism network topology, which can be used to improve cyber security practices. After that non-negative matrix decomposition (NMF) is used to find communities from above isomorphism network topology. The experimental results show that the LSHNM can find more satisfactory community structures.

INDEX TERMS Location-based service, social network, homogeneous social network, community discovery, cyber security.

I. INTRODUCTION

The Location-Based social network (LBSN) is a new type of social system that has sprung up recently. It connects the real world and online social network closely by integrating location information into traditional social network, and provides users a brand new social service. The research methods in traditional social network are no longer meet the current demands under the impact of the new forms of social data. Hence, researchers are beginning to delve deeper into LBSN. Community discovery is one of the basic research problems on social network. How to use user social, temporal, spatial and behavioral information contained in new forms of social data to make comprehensive analysis of user characteristics and find potential user communities is an extremely valuable problem. For this reason, this paper considers the constitute form of communities in LBSN from many aspects and defines them as: the users in same community are closely connected, and the geographic distance of their access areas are close, and their behavior patterns are consistent. It is beneficial to excavate user groups with similar social relations, temp-spatial distribution and behavioral patterns by detecting communities on LBSN, which could provide support for many applications with economic significance and social significance such as friend recommendation, direct-marketing and behavior prediction [1]–[3].

The LBSN is a heterogeneous network composed of two different types of vertices, user and location [1], [4], [5]. As a result, the community discovery algorithms in traditional social network cannot be applied directly to LBSN. There are

The associate editor coordinating the review of this manuscript and approving it for publication was Kim-Kwang Raymond Choo.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

three main solutions to the problem of community discovery in LBSN. Wang *et al.* [6] proposed an edge clustering algorithm which disconnected social links of users in LBSN and makes them as an attribute of user vertex. After that, the clustering can be achieved by measuring the similarity between user check-in edges. Another popular solution is to use LDA (Latent Dirichlet Allocation) model to detecting communities in LBSN. Joseph *et al.* [7] applied LDA model in LBSN. In their paper, the users were regarded as the documents and the categories of check-in locations were regarded as the words that constitute the documents. After solving LDA model, clusters are obtained according to the topic distribution. Based on the above research, Li *et al.* [4] spent more effort on figuring out the major factors of community formation in LBSN. He established a more complex LDA model to discover multi-dimension user communities of LBSN by taking into account both social relations and behavioral patterns. Brown *et al.* [8] came up with another classic clustering idea. They re-annotated the users' social topology by using the information that friends visit the same location. Consequently, location information can be used as weight integrating into social topology. High weight represented the important role of location in the formation of users' friendship. Therefore, it was possible to use the difference of weights to find user communities with similar social relations and location. Lim *et al.* [9] considered the influence of time factor on the formation of user community in LBSN. Therefore, they added time constraints to users' common check-in behaviors and created new connections between users from three perspectives: social relationships, location and time [10]–[13]. After that, LBSN communities were discovered based on above new connections by using traditional community discovery algorithms.

Furthermore, the existing solutions to the community discovery in LBSN are facing various problems: 1. The edge clustering algorithm will consume a lot of computer memory and spend a lot of computing time; 2. The clustering algorithm based on LDA model requires many parameters and the selection of these parameters is difficult; 3. The homogeneous network model with traditional clustering algorithm often fails to excavate the hidden community relations between users without social connections and creates some isolated nodes.

This paper studies the community discovery technology in LBSN from the perspective of heterogeneous network. It is required to find reasonable community structure that users in same community should have strong social relationships, similar geographical distribution in same time slice and similar behavior patterns. Hence, this paper proposes a LBSN homogeneous network model (LSHNM) based on users' social, temp-spatial and behavioural information [14].

II. PROPOSED SOLUTIONS AND FRAMEWORK

In order to solve community discovery problem in LBSN, one intuitive idea is to re-factor the LBSN heterogeneous network. More specifically, it first deletes location vertices and all edges from original LBSN topology, and then creates new links between users based on user characteristics in LBSN community, finally community discovery algorithm can be directly applied to above new network topology. The overall framework is shown in Figure 1. According to the Figure 1, the first step is to analyze and clean the acquired LBSN data which is provided to other modules as data source. The next step is to quantify the similarity between users in LBSN from three characteristics: social relations, temp-spatial distribution and behavioural patterns. Then, LBSN homogeneous network topology is built based on above three characteristics similarity. Finally, high-speed and stable community discovery algorithm will be used to find multi-attribute community in LBSN with the help of above new topology. The combination of all above steps constitutes the LSHNM which is proposed in this paper.

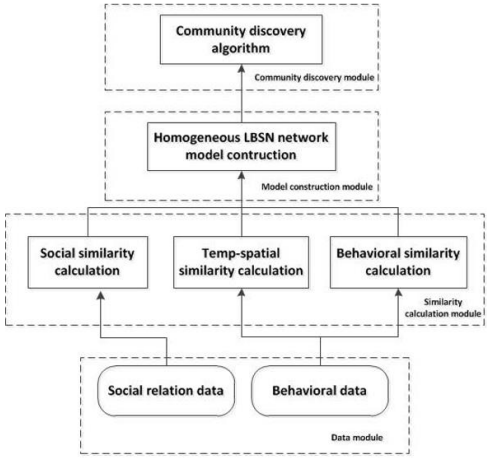

FIGURE 1. Community discovery framework of LBSN.

A. USER CHARACTERISTIC ANALYSIS

With the emergence of LBSN, a large number of new forms of social data have been generated. In these social networks, the user community is no longer just satisfied the close bonding in social relations but also the similarity of user characteristics shown in the new data. According to the above social data, this paper focus on analyzing user characteristics from the aspects of social relationship-geographic space and interests.

1) SOCIAL RELATIONSHIPS

In most traditional social networks, the community discovery algorithms are based on social relations, that is, the user's friend relationship. Thus, the traditional user communities are made up of close friends. In other words, friendship is an important feature of community formation in social networks.

2) TIME

In addition to the above social relation characteristics, users in LBSN have some time characteristics which related to their check-in behaviors. Due to the sparseness of the

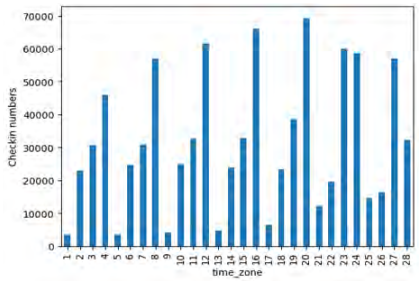

FIGURE 2. User check-in time distribution diagram (28ts).

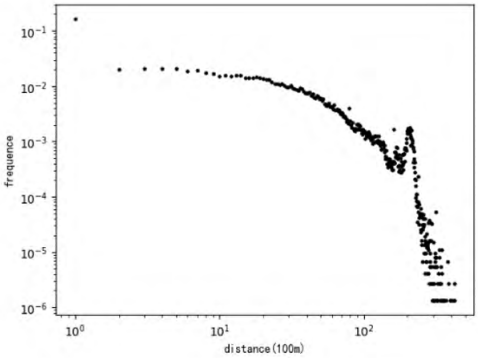

FIGURE 3. The distance distribution map of the check-in location from the check-in center.

check-in data, this paper divides the day into $\{0-6, 6-12, 12-18, 18-24\}$ time slices, and 7 days per week. Figure 2 shows the LBSN users total check-ins in 4×7 time slices. According to above figures, it is found that user check-in behavior has strong time regularity. The number of user check-ins vary from time to time, and the check-in characteristic during the working day is almost the same, while it is slightly different in weekends.

3) GEOGRAPHIC SPACE

Noulas *et al.* [15] do some research on continuous check-in behavior in LBSN which shows that 80% of continuous check-in behaviors occur within a distance of 10 kilometres. In addition, this paper also makes an analysis on the distance between all check-in locations of users and their central check-in points, the statistical results are shown as Figure 3.

The above figure indicates that most users tend to stay in a small region of the city. Moreover, this paper randomly selects two users from Four square New York data set and visualizes their check-in Figure 4 with the help of Google map. From the observation of the above images, the user's check-in behavior has a strong geographical preference, and different users have different frequently-visited areas.

4) INTERESTS

The check-in data in LBSN could not only show the time, geographic space characteristics of users, but also find their interests from check-in category data. For example, Figure 5

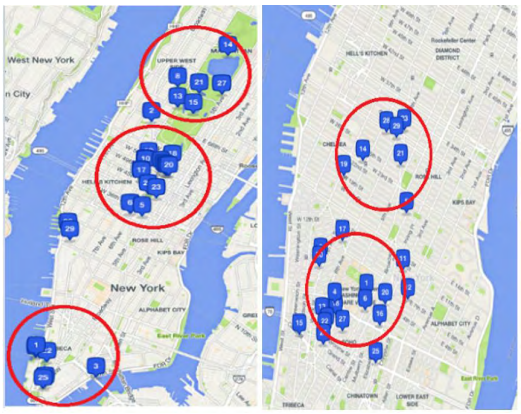

FIGURE 4. User A and B check-in distribution map.

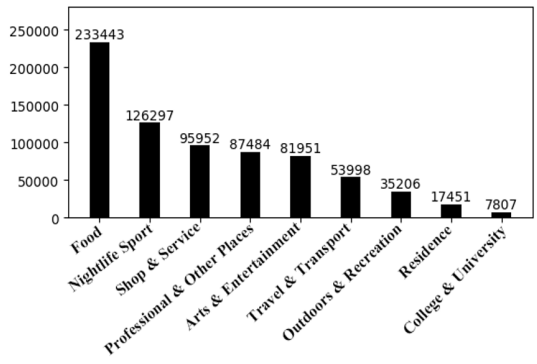

FIGURE 5. All users check-in number distribution under different categories.

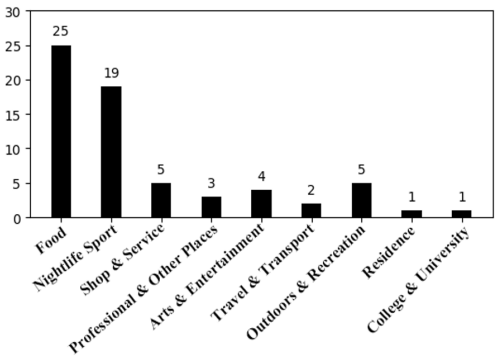

FIGURE 6. User A check-in number distribution.

shows the statistical analysis of the check-in number of users in the entire data set under each category. As can be seen from the above figure, the user's interests are different and the number of check-ins in category ''Food'' and category ''Nightlife sport'' accounts for the most. In order to highlight the differences of users' interests more clearly, this paper also randomly selects two users from data set and counts their check-in categories. The specific statistical results are shown in Figure 6 and 7: In combination with the user's social relations and rules of time, geographic space and interests contained in user check-ins, this paper proposes social relation, check-in temp-spatial distribution and behavioral pattern characteristics of user in LBSN community. Social

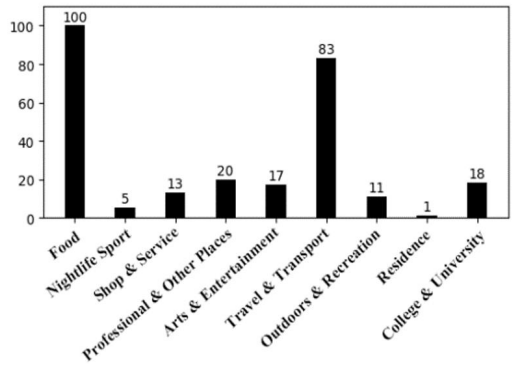

FIGURE 7. User B check-in number distribution.

relations characteristic refers to the connection features in the social topology, and check-in temp-spatial distribution characteristic indicates the geographical location distribution of the user's check-in within the same time slice and behavioral pattern characteristic means the category distribution of the user's check-in location within the same time slice. It makes sense because the user behaviours are stable in working day.

B. USER SIMILARITY CALCULATION

In this paper, the similarities of users in social relations, temp-spatial distribution and behavioral patterns are calculated by using social relation data and check-in data from LBSN. The detailed calculation scheme are described as follows.

1) SOCIAL SIMILARITY

The vertices with high correlation in the network usually have following two features: (1) There are many paths between two related vertices. (2) The length of paths between two related vertices are relatively shorter. Based on the above features, this paper propose two social similarity calculation methods which are fast Katz and random walk algorithms.

2) THE BASIC CONCEPT OF KATZ

This calculation method is an amplification of shortest distance [16]. It not only considers the shortest path between two nodes in the network, but also considers the number of different length paths. At the same time, this method adds β_l which is the power of a decay factor and path length to control the contribution of different length paths to social similarity. In general, the path length between two points is inversely proportional to the weight of social similarity. The specific calculation formula is as

$$Sim_s(x, y) = \sum_{i=1}^{n} \beta^i \cdot |paths_{(x,y)}^{<i>}| \quad (1)$$

in which $paths_{(x,y)}^{<l>}$ is the path set with path length l between vertex x and y. β is a constant decay factor less than 1, is usually is 0.05 or 0.005, and n denotes the given max path length.

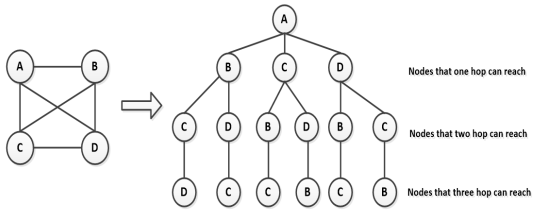

FIGURE 8. All users check-in number distribution under different categories.

3) THE CALCULATION OF KATZ

According to the definition of Eq.(1), the main goal of Katz algorithm is to find the number of different length paths between any two nodes in the network, and then set different weights to calculate similarity between users based on different path length. In order to achieve the above goal, a fast Katz algorithm based on spanning tree is proposed.

Fast Katz algorithm does not cycle calculate the number of different length paths between any two nodes directly. Instead, from each node, this algorithm use depth search algorithm to traverse all the nodes it can reach with different hop counts and generate a tree structure. Figure 8 is a simple example of the fast Katz algorithm.

Starting from node A in the above figure, the nodes that it can reach after one hop are B,C,D, so B,C,D appear in the first layer of the tree. Then proceed from B, C, D to find the nodes that node A can reach through 2 hops and put it into the second layer of the tree. Follow above steps until algorithm have created a complete tree. After completing the spanning tree construction of the node A, the number of paths of all different lengths between node A and other nodes in the figure can be obtained by simply traversing each layer of the tree. Finally, by adding weight to each path found by the algorithm, the social similarity between node A and other nodes can be quickly calculated. The concrete algorithm framework is shown as in Algorithm 1

Function ''createAdlist'' is used to build the adjacency list for each node, but it is not described in detail here. Function ''Generate_tree'' is used to build a spanning tree based on the given node v and its concrete algorithm framework is shown as follows

The basic concept of random walk. In addition to the two features mentioned in Section 2.1, there is a third feature for nodes with high correlation—-The paths between two related vertices less pass through big degree vertices. The nodes with large degree are popular nodes in the network, which connect a great quantity of other nodes, so the path through such nodes will contribute less to social similarity. However, most current measurement methods [17], such as shortest distance, common neighbor, Jaccard's coefficient, Adamic/Adar and Katz, have not fully consider above features especially the third feature. Therefore, Fouss *et al.* [18] proposed a random walk algorithm based on Markov chain to solve the above problems, and achieved good experimental results.

According to above research results, this paper proposes a user similarity calculation method based on random walk.

Algorithm 1 Fast Katz Algorithm

Input : Node-set V, edge-set E, decay factor β
Output: Social similarity between users

Build the adjacency list for each node;
foreach *node v of the V* **do** CreateAdlist(V, E);
Build an empty similarity dictionary similarity $= \{\}$;
Build an empty tree structure tree $= \{\}$;
Build a spanning tree based on the given node v;
for $v \in V$ **do**
  $tree[v] = Generate\_tree(v)$;
end
*T*raversing each layer of the node v spanning tree and overlaying the similarity weights between v and nodes in each layer;
for $v \in V$ **do**
  for *level* $l \in tree[v]$ **do**
    $weight = \beta^{level}$; **for** *level* $l \in tree[v]$ **do**
      $similarity[v][node] = S_{social}(v, node)$;
    end
  end
end
*R*eturn similarity for each $v \in V$;

This method gives a transfer probability to each vertex in the network of reaching its neighbour node. The value of this probability is equal to the reciprocal of degree of the vertex. Then, each vertex will travel to other vertices according to this probability. If two vertices meet the above three characteristics, the probability of mutual visits between them is higher, namely, the social similarity between the two vertices is higher. The calculation formula of the access probability between vertices is shown as

$$Prob(x|y) = \sum_{l=1}^{n} Chain_l^{(x,y)} \tag{2}$$

where $Chain_l^{(x,y)}$ is the access probability of the lth path which is started with vertex x and end with vertex y. n represents the total number of paths between x and y. The calculation method of $Chain_l^{(x,y)}$ is shown as

$$Chain_l^{(x,y)} = \Pi_{i=1}^{k} pnode_i \tag{3}$$

where $pnode_i$ is the transfer probability of vertex i which is the ith vertex in the path that is started with vertex x and end with vertex y. k represents the total number of vertices in this path. The calculation method of $pnode_i$ is shown as

$$pnode_i = \frac{1}{degree_i} \tag{4}$$

in which $degree_i$ is the degree of vertex i. Since the access probability between two vertices is not symmetrical, the similarity calculation formula between vertices is as follows:

$$Sim_s = \sqrt{prob(x|y) + prob(y|x)} \tag{5}$$

Algorithm 2 Generate_tree Algorithm

Input : Node v, adjacency list *node_neighbors*[v]
Output: the spanning tree of v

Create empty tree structure $node = v$, $level = 1$, $tree = \{\}$;
Create depth search access token visit $= \{\}$;
Define the depth search function dfs($node$, $level$);
If the search depth has reached the upper limit, the call is ended and upper limit here is the maximum path length;
if $level \geq limit$ **then**
  return v;
end
Set the current node access flag as $visited[node] = True$;
Traverse the node's adjacency list and find the unvisited node for recursive access;
for $n \in node\_neighbors[node]$ **do**
  if $visited[n] == False$ **then**
    tree[level].append(n);
    dfs(n, level++);
    visited[n] = False;
  end
end
Call the depth search algorithm dfs(v);
return tree v;

The calculation of random walk. Similar to fast Katz algorithm, this algorithm also need to find all paths between any two vertices in the graph. However, the random walk algorithm does not need to count the number of paths but needs to calculate the access probability of each path between nodes. Therefore, a slight modification of the fast Katz algorithm can achieve the above goal. In simple terms, each layer of a node spanning tree stores not only the reachable nodes ID but also the path access probability to reach these nodes. Moreover, the concrete algorithm framework is similar to fast Katz algorithm, which will not be described in detail.

Parameter setting. The social relationship topology in LBSN is an unweighted and undirected graph. Therefore, the path number involved in fast Katz and random walk social similarity calculation algorithms increases as the path length increases, which results in a large amount of computational resource loss and the production of meaningless results. This paper adopts the ''three-degree influence criterion'' proposed by Christakis *et al.* [19] in 2011 to only consider the strong connections within three degrees of the social topology. In addition, the value of decay factor β in fast Katz algorithm is 0.05.

4) TEMP-SPATIAL DISTRIBUTION SIMILARITY

Considering the temporal and spatial factors analyzed above, the goal of this section is to calculate the temp-spatial

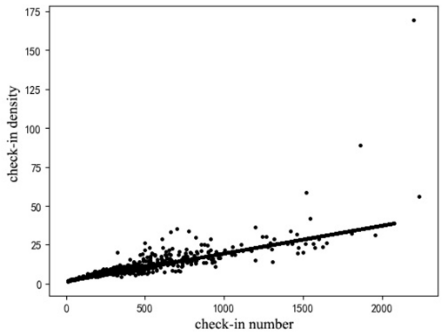

FIGURE 9. Statistical graph of average check-in density of grid under different check-in scales.

distribution similarity between LBSN users, namely, the distribution similarity of the locations that users visited in same time slices. In order to achieve above goal, this paper proposes adaptive density clique algorithm and temp-spatial distribution map convolution algorithm (TSDMC).

The basic concept of adaptive density clique. According to the geospatial analysis of users' check-ins in chapter 2, it is found that users' check-in locations are significantly clustered and they have frequently visited areas. Therefore, the temp-spatial distribution similarity of users can be roughly expressed by comparing the distance between the frequently visited areas of users in the same time slice. In order to mine above areas, this paper uses adaptive density clique algorithm to cluster the user's check-in locations.

The adaptive clique algorithm is an improvement of the clique algorithm [20]. It also belongs to a grid-based clustering algorithm and is mainly used for the discovery of high-density clusters. The algorithm does not need to specify the number of communities in advance, and only requires two parameters, grid step size and density threshold. The grid step size is mainly used for the grid division of the target space, while the density threshold is used to distinguish the high-density grids. Furthermore, the user check-in density refers to the ratio between the user total check-in number and the number of grids with check-ins, and high density refers to the check-in density that exceeds the check-in density of current check-in quantity scales. Since the number of check-in for each user is different, the density threshold should also be different for each user. Aiming at the difference of density threshold among users, this paper proposes an adaptive density threshold method. This method firstly counts the average check-in density of users under different check-in quantity scales, and then uses a straight line to fit the average density change with the number of check-ins. Finally, the adaptive clique algorithm can directly determine the user density threshold based on the fitting line and the number of user check-ins. The result is shown in Figure 9. By using the adaptive density clique algorithm to obtain the users' frequently visited areas under each time slice, an appropriate similarity metric is needed to compare the temp-spatial distribution similarity between users. Therefore, this paper proposes a suitable temp-spatial distribution

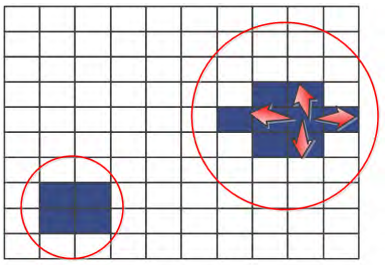

FIGURE 10. Clustering process.

similarity measurement method which is based on the method proposed in literature [21].

$$Sim_s(x, y) = \frac{1}{T}\sum_{t=1}^{T}\frac{\rho_{xt}\rho_{yt}}{M_t \cdot N_t}\sum_{m=1}^{M_t}\sum_{n=1}^{N_t} dis(l_{xm}, l_{yn}) \quad (6)$$

In the above equation, M_t and N_t represent the number of frequently visited areas by user x and y under time slice t, and ρ_{xt} and ρ_{yt} represent the ratio of the number of check-ins from user x and y in the tth time slice and the total number check-ins from them in all time slices. l_{xm} and l_{yn} represent the coordinates of the center point of the area. dis represents the spherical distance between the center points of the two areas and T refers to the total time slices. The physical meaning of the Eq.(6) is the average distance between frequently visited areas within T time slices, and is used to represent the temp-spatial distribution similarity between users.

The calculation of adaptive density clique. First, this algorithm divides the study area such as New York City into several grids. Second, mapping all check-in locations of a single user to the grids and calculating check-in density per grid. Finally, clustering can be achieved by merging adjacent areas with high check-in density. The diagram of clustering process is shown in Figure 10, where the dark square represents the area with high check-in density. Since this paper needs to calculate the temp-spatial distribution similarity of users, it is necessary to divide users' check-ins by time and calculate the similarity of the check-in location distribution of users in each time slice. However, the user data obtained by the project team is relatively sparse. If time division is carried out, it may not be possible to cluster the user's check-in at a certain time slice. Therefore, this paper clustering user's check-in at all time slices first, and then allocate the clustering results to each time slice.

After using the adaptive density clique algorithm to obtain the center points of all users' frequently visited areas, the temp-spatial distribution similarity between users can be calculated according to Eq.(6).

The basic concept of temp-spatial distribution map convolution algorithm. Although the adaptive density clique algorithm can calculate the similarity of temp-spatial distribution among users, the use of the center point to represent the user's frequently visited area is slightly rough, and may not fully express the temp-spatial distribution of the user's check-in. Therefore, this paper proposes another temp-spatial

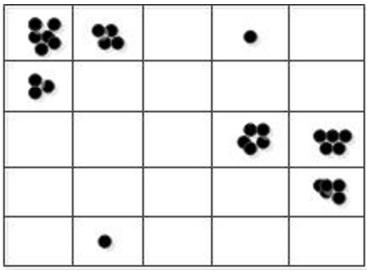

FIGURE 11. Check-in temp-spatial distribution diagram.

distribution similarity calculation method: Temp-Spatial Distribution Map Convolution algorithm (TSDMC).

The main idea of the TSDMC algorithm is to project user's check-ins of one time slice into a grey-scale geographical map which is already divided into grids. After above conversion, the check-in location distribution of one user in same time slice can be represented by a grey-scale map. Then, this paper take the idea of convolution which is widely used in convolution neural network to extract local statistical features of this grey-scale map. Finally, by using Euclidian distance to compare similarity of above local statistical features from each user, the similarity of user's temp-spatial distribution is also obtained. The geographical map is shown in Figure 11.

The final temp-spatial distribution similarity calculation formula is shown as

$$Sim_{spatio}(x, y) = \frac{1}{n \cdot T} \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{r=1}^{m} (fmap_i^x[r] - fmap_i^y[r])^2 \tag{7}$$

in which $fmap_i^x[r]$ is the value of the rth feature in ith feature map of vertex x. n is the number of feature maps. m represents the number of features in a feature map. T indicates the number of time slices.

The calculation of temp-spatial distribution map convolution algorithm. The concrete algorithm framework is shown as follows, where the convolve2d function is the convolution calculation function in the scipy package of Python [22]:

This paper uses different sizes convolution cores to carry out convolution operations for each user under different time slices, and obtain their feature maps. Finally, the similarity of temp-spatial distribution among users is calculated according to Eq.(7). However, it is a very time-consuming process to calculate the similarity of feature maps by Euclidean distance. In order to solve the above difficulties, this paper proposes a matrix based algorithm for calculating Euclidean distance between any two users. The specific calculation process is shown as follows.

Suppose there are two groups of vectors a and b, and two three-dimensional vectors in each group. For example, the group a is composed of a_1, a_2, and the group b is composed of b_1, b_2, as shown in the following

$$a = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}, b = \begin{pmatrix} b_{11} & b_{12} \\ b_{12} & b_{22} \\ b_{13} & b23 \end{pmatrix} \tag{8}$$

Algorithm 3 TSDMC Algorithm

Input : User's check-in data under time slice $t = \{c_1, c_2, \ldots, c_M\}$, grid number n, convolution kernel size k

Output: Feature map (fmap)

Initialize the map and divide the study area into $n \times n$ grids.;
for $m \leftarrow 1$ **to** M **do**
    $r = calcRegion(c_m)$;
    $img[r] + +$;
end
Convert the number of check-ins in each grid to the check-in ratio;
for $r \leftarrow 1$ **to** $n \times n$ **do**
    $img[r] = img[r]/sum(img)$;
end
$fmap = convolve2d(img, k)$;
return feature map fmap;

In order to calculate the Euclidean distance between any two vectors of groups a and b, the square sum of elements in each row of the group a is first calculated and a square sum matrix is formed by extending square sum results horizontally, as shown

$$Asq = \begin{pmatrix} a_{11}^2 + a_{12}^2 + a_{13}^2 & a_{11}^2 + a_{12}^2 + a_{13}^2 \\ a_{21}^2 + a_{22}^2 + a_{23}^2 & a_{21}^2 + a_{22}^2 + a_{23}^2 \end{pmatrix} \tag{9}$$

The sum of elements in each column of the b group is then summed up and expanded vertically according to the number of columns to form a square sum matrix as

$$Bsq = \begin{pmatrix} b_{11}^2 + b_{12}^2 + b_{13}^2 & b_{11}^2 + b_{12}^2 + b_{13}^2 \\ b_{21}^2 + b_{22}^2 + b_{23}^2 & b_{21}^2 + b_{22}^2 + b_{23}^2 \end{pmatrix} \tag{10}$$

Finally, the Euclidean distance is $Asq + Bsq - 2ab$.

Because of the above calculation method, the multi-layer cycle in the original algorithm is replaced by the matrix operation, so the calculation time can be greatly reduced by the acceleration of matrix operation. In addition, his paper regards a as the matrix of all users' fmaps, and regards b as transpose of a.

Parameter setting. In the above two algorithms, the size and the number of grids should be carefully selected. There are no strict rules for the selection of grid size. The smaller the grid is, the more accurate the results are. But it also leads to increase time complexity. Chao *et al.* [23] set the grid size to $5km \times 5km$ in the grid division of New York City. Based on reference and above analysis, in order to obtain more accurate experimental results, this paper decides to sacrifice some computation time and set grid size to $1km \times 1km$. In addition, the selection of grid number is constrained by the scope of study region and the size of grid (n = 46 in this paper). As for TSDMC algorithm, there are also no strict rules in selecting convolution kernel size. However, different

convolution kernel size would extract different local statistical features of images. So, this paper choose $1km \times 1km$, $2km \times 2km, \ldots, 5km \times 5km$ these five different kernel size to carry out five repeated experiments. The final result is the mean of above five experimental results. As for the selection of values in the convolution kernel, all 1 is used which can get the local accumulation characteristics of images.

5) BEHAVIORAL PATTERN SIMILARITY

Behavioral pattern similarity refers to the similarity of users' check-in category vectors in the same time slice. However, due to the sparsity of the check-in data, the number of users' check-ins in each time slice is relatively small. Therefore, this paper aims to alleviate the sparsity by combing some similar approximation categories such as airport and airport lounge. In the end, this paper uses the cosine similarity and generalized Jaccard similarity to calculate the similarity of the user's check-in categories in the same time slice.

Cosine similarity. This method is to measure the similarity between two vectors by calculating the cosine value of the angle they formed. The closer the cosine value is to 1, the smaller the angle between them is, and the more similar they are. Based on the above concepts, this paper proposes a behavioral pattern similarity based on cosine similarity. The specific formula is shown below

$$Sim_b(x, y) = \sum_{t=1}^{T} \frac{Vec_x \cdot Vec_y}{\|Vec_x\| + \|Vec_y\|} \tag{11}$$

where Vec_x, Vec_y are two 140 dimensional vectors, which represent the ratio user x and user y checked in 140 different categories at same time slice. T indicates the number of time slices.

Generalized Jaccard similarity. This method is also called Tanimoto coefficient [24], which is mainly used to calculate the similarity between sets. Compared with the narrow sense of Jaccard, the value of elements in generalized Jaccard similarity can be real number. So it is often used to calculate text similarity. The calculation formula is as follows

$$Sim_b(x, y) = \sum_{t=1}^{T} \frac{Vec_x \cdot Vec_y}{\|Vec_x\|^2 + \|Vec_y\|^2 - 2Vec_x \cdot Vec_y} \tag{12}$$

in which the meaning of Vec_x, Vec_y and T is same as Eq.(12) and the $\cdot$ denotes the vector inner production.

6) COMBINATION OF COMPUTATIONAL METHODS

In above three sections, two user similarity calculation methods are proposed for each user characteristic. Although each calculation method has the same goal, its idea and calculation process are different. Therefore, this paper intends to produce $2 \times 2 \times 2 = 8$ user similarity calculation method combinations under three characteristics and determine the best combination by many contrast experiments. The specific experimental results refer to the next Section.

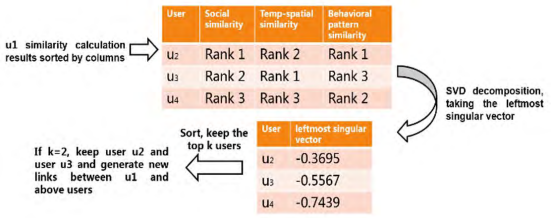

FIGURE 12. Unified graph construction algorithm schematic.

III. HOMOGENEOUS NETWORK CONSTRUCTION AND COMMUNITY DISCOVERY

A. HOMOGENEOUS NETWORK CONSTRUCTION

This chapter intends to reconstruct the original network by creating virtual links based on the similarity of social relations, temp-spatial distribution and behavioral patterns, thus forming the LBSN homogeneous network topology. In order to make each link of LBSN homogeneous network topology better reflect the similarity of users in the above three aspects, a unified graph construction algorithm is proposed.

Unified graph construction algorithm uses SVD(Singularly Valuable Decomposition) [25] to find main feature that can represent the relationship between above three similarity measurement results in new feature space. Figure 12 is a simple example.

Firstly, the similarity calculation results of a random user u_1 in the LBSN is represented in a similarity matrix, and each column of above matrix is sorted in descending order, and the highest similarity value ranks first. As shown in Figure 12, the user u_1 has the highest similarity with user u_2 and lowest similarity with user u_4 in social similarity. Then, the SVD decomposition is performed on the sorted similarity matrix, leaving the leftmost singular value vector, that is, the main feature vector in the new feature space. Finally, this algorithm selects k users that are most similar to user u_1 in social relationships, temp-spatial distribution, and behavior patterns based on the values of the main feature vector. As shown in Figure 12, when $k = 2$, the unified graph construction algorithm will keep user u_2 and user u_3 and generate new links between u_1 and above users. The algorithm framework is described in Algorithm 4: Fast Katz Algorithm.

By using above unified graph construction algorithm, the new topology structure of each user in LBSN is generated. Thus, a LBSN homogeneous network based on new topology links is constructed.

B. COMMUNITY DISCOVERY

This subsection intends to explore the community structure of LBSN homogeneous network constructed in the previous section by using traditional community discovery algorithm. NMF (Nonnegative Matrix Factorization) [26] is a widely used community discovery algorithm, which not only has strong interpretability but also can find overlapping communities structure. For this reason, this paper users NMF

Algorithm 4 Fast Katz Algorithm

Input : The similarity measurement results Sim_{social}, $Sim_{spatial}$, $Sim_{behavior}$ of user u_1 in LBSN. Number of remaining neighbors k.

Output: k new neighbor topology links of u_1

1 *Create an $N \times 3$ user matrix A. Where N represents the total number of users in LBSN except u_1 is the number of similarity measurement features. $A[i][j]$ indicates the value of feature j between u_1 and u_i.*;

2 *For each column in A, it is sorted by values. The highest similarity value is Rank*1*, and so on. If the same value is encountered, the same Rank value is given. After above processing, the similarity value in each column is replaced by the Rank value, but does not change the original sequence of A.*;

3 *Implements SVD decomposition on matrix $A = U\Sigma V^T$: Where U is a $m \times m$ unitary matrix, Σ is a semi-positive $m \times n$ diagonal matrix, V is a $n \times n$ unitary matrix.*;

4 *Take out the first column U_1 from U. It is the main feature in new feature space which represents the comprehensive similarity comparison results between u_1 and other users based on Sim_{social}, $Sim_{spatial}$ and $Sim_{behavior}$. The larger the value in U_1, the more similar to u_1 in above three features. Next, sorting the value of U_1 from largest to smallest, and retain top-k values. Finally, take out the user id corresponding to the top-k values, and generate new topology links of u_1.*;

5 *Output k new topology links of u_1.*

algorithm to explore community structure in LBSN homogeneous network.

The idea of non-negative matrix decomposition is to project the original data into a new feature space, and reconstruct the original data with the projection result and spatial information. Given a non-negative matrix $X_{m\times n}$, and decomposes it into two low-rank non-negative matrices $W = [W_i c] \times R_+^{(m\times k)}$ and $H = [H_{jc}] \times R_+^{(n\times k)} (k \ll n, m)$ to make their product as close to original matrix X as possible, namely, $X \approx WH$. Formally, the NMF algorithm can be considered as the following optimization problem

$$\min D(X, WH) \quad s.t.\ W \geq 0, H \geq 0 \tag{13}$$

where the loss function $D(A, B)$ is used to measure the difference between A and B. The squared error function and KL divergence are two commonly used loss functions. In this paper, square loss function is used, and it's specific formula is as

$$D_{LSE}(X, WH) = \|X - WH\|_F^2 \tag{14}$$

The data matrix X used here is the adjacency matrix of LBSN homogeneous network. If there is an edge between vertex i and vertex j in the network, then $X_{ij} = 1$, otherwise $X_{ij} = 0$. After determine data matrix X and loss function, this paper use iterative method to solve Eq.(13). The specific solution steps are shown below: (1) Build Lagrange function

$$\begin{aligned} l &= \|X - WH\|^2 - \partial W - \beta H \\ &= tr[(X - WH)(X - WH)^T] - \partial W - \beta H \\ &= tr(XX^T - 2WHX^T + WHH^T W^T) - \partial W - \beta H \end{aligned} \tag{15}$$

(2) Take the derivative with respect to W and H

$$\frac{\partial l}{\partial W} = -2HT^T + 2WHH^T - \alpha \tag{16}$$

$$\frac{\partial l}{\partial H} = -2W^T X + 2W^T WH - \beta \tag{17}$$

(3) $\alpha W_{ir} = 0$ and $\beta H_{rj} = 0$ according to the KKT conditions and restrictions ($W \geq 0, H \geq 0$), and

$$[-2HT^T + 2WHH^T]_{ir} W_{ir} - \alpha W_{ir} = 0 \tag{18}$$

$$[-2W^T X + 2W^T WH]_{rj} H_{rj} - \beta H_{rj} = 0 \tag{19}$$

where r is the number of communities. Fix other variables and update W according to

$$W_{ij} \leftarrow W_{ir} \frac{(XH^T)_{ir}}{(HH^T)_{rj}} \tag{20}$$

Fix other variables and update H according to Eq.(19)

$$H_{rj} \leftarrow H_{rj} \frac{(W^T X)_{rj}}{(W^T WH)_{rj}} \tag{21}$$

The maximum number of iterations or convergence thresholds is set up in above iterative method to get membership matrix W. Each row of the matrix W represents the probability that the node belongs to the community.

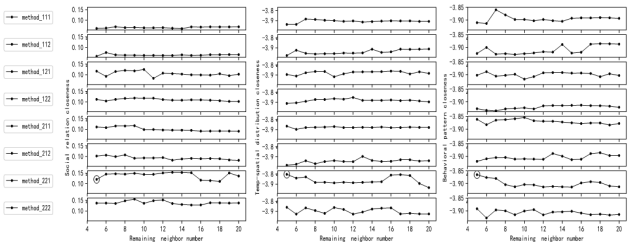

FIGURE 13. Internal experimental results ($c = 5$, $\rho = 0.1$).

IV. EXPERIMENTS AND RESULT ANALYSIS

A. EXPERIMENTAL DATA AND EVALUATION INDEXES

This experiment selects the New York City data collected by the project team during 2013, including user social relation data and user check-in data. After simple data cleaning, some abnormal data and outliers are filtered. Finally, It gets 6,141 users, 16,947 locations, 739,589 check-in records and 116,778 social edges. In this work, we made all experiments on a Macbook Pro with 2.8GHz Intel Core i7 CPU and 8GRAM.

In this paper, the quality of community discovery is evaluated from three aspects: social relations, temp-spatial distribution and behavioral patterns. Social closeness, using classic overlapping modularity [27] to evaluate the results of overlapping community discovery, the specific formula is shown below

$$Q = \frac{1}{2m}\sum_{c}\sum_{i,j\in c}[A_{i,j} - \frac{k_i k_j}{2m}]\frac{1}{O_i O_j} \tag{22}$$

where m is the number of social edges in one community. A is the adjacent matrix and k_i, k_j represent the degree of vertex i and vertex j. O_i, O_j represent the probability of vertex i and vertex j that belong to this community.

Temp-spatial distribution closeness, using entropy method to measure dispersion degree of users' check-in locations in the same community at same time slice. The specific calculation formula is as follows

$$Q = -\sum_{t=1}^{T}\sum_{j=1}^{M}\frac{|C_j|}{TC}\sum_{n=1}^{N}\frac{W_{c_j,z_n}}{|C_j|}log\frac{W_{c_j,z_n}}{|C_j|} \tag{23}$$

where T is the number of time slices and M is the number of communities. TC represents the total check-in numbers and $|C_j|$ indicates the number of check-ins in j community. N is the number of grids in New York City and $W_(C_j, Z_n)$ represents the check-in number in grid Z_n of community C_j. Smaller Spatial H represents a large number of check-in locations of users in same community and same time slice is gathered in a small number of areas.

Behavioral pattern closeness which uses the same calculation method as temp-spatial distribution. But the user check-in grid is replaced by user check-in location category. The specific formula is as follows

$$BhaviorH = -\sum_{t=1}^{T}\sum_{j=1}^{M}\frac{|C_j|}{TC}\sum_{n=1}^{N}\frac{W_{c_j,Cat_n}}{|C_j|}log\frac{W_{c_j,Cat_n}}{|C_j|} \tag{24}$$

where $W(C_j, Cat_n)$ represents the total check-ins of category Cat_n in community C_j.

B. RESULTS AND ANALYSIS

The proposed LSHNM needs to set 4 parameters, they are the combination of similarity calculation methods, the number of remaining neighbors k in unified graph algorithm, the number of community c, and the threshold ρ of community membership. In addition, the first two parameters only exist in LSHNM. Therefore, the experiment first needs to find the best value of the above two parameters in different situations through the internal experiment, and then the superiority of the LSHNM is proved by the contrast of the external experiments with other algorithms.

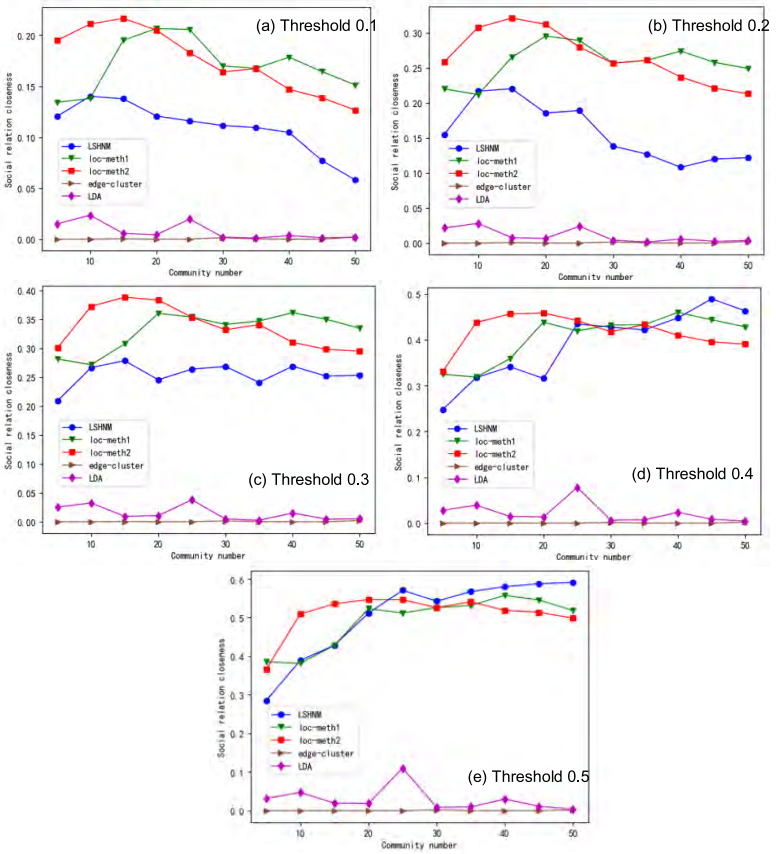

FIGURE 14. Social closeness contrast result.

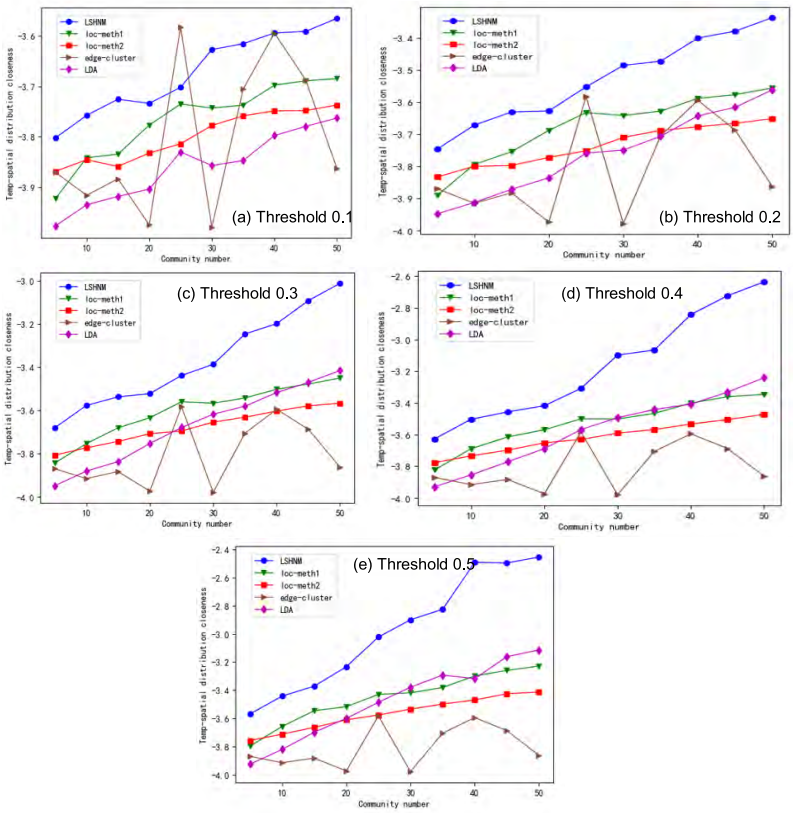

FIGURE 15. Temp-spatial distribution closeness contrast result.

Internal experiments. Figure 13 is the result of an internal experiment with whose community number is 5 and community membership threshold is 0.1.

The above figure is composed of three subgraphs, representing the experimental results under the three evaluation indexes, and each subgraph is composed of eight curves which represent the experimental results of the eight user similarity calculation method combinations vary with the change of the number of remaining neighbors k. According to the above three kinds of evaluation indexes, ''method_221, $k = 5$'' (the circle point in the diagram) has obtained a relatively better experimental results. ''method_221'' represents the combination of second method of calculating social similarity, namely, random walk algorithm, and second similarity calculation method in temp-spatial distribution similarity, namely, TSDMC algorithm and the first calculation method of behavior pattern similarity, that is, cosine similarity algorithm. However, because of too many internal experiments, this paper does not analyse the experimental results of other situations in detail, but gives the final results directly, as shown in Table 1.

In above table, c represents the number of communities, and ρ represents the threshold of community membership. k the number of remaining neighbors, and the number string represents different combinations of similarity computation methods. The results in the above table show the two combinations ''221'' and ''121'' get better performance under three evaluation indexes, and we select combination ''221'' and $k = 17$ in the next experiments to do comparisons with other typical LBSN community discovery algorithms.

TABLE 1. Comparison of experimental results.

	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.4$	$\rho = 0.5$
c=5	221, k=5	221, k=17	221, k=17	221, k=17	221, k=17
c=10	221, k=17	221, k=17	221, k=17	221, k=17	221, k=17
c=15	221, k=16	221, k=16	221, k=16	221, k=16	221, k=16
c=20	221, k=11	221, k=11	221, k=11	221, k=10	221, k=11
c=25	221, k=17	221, k=17	221, k=17	121, k=20	121, k=17
c=30	221, k=8	121, k=8	121, k=18	121, k=20	121, k=17
c=35	221, k=8	121, k=20	121, k=20	121, k=20	121, k=19
c=40	221, k=8	121, k=8	121, k=20	121, k=20	121, k=19
c=45	121, k=7	121, k=20	121, k=20	121, k=17	121, k=19
c=50	121, k=12	121, k=20	121, k=20	121, k=17	121, k=17

External experiments. The comparison algorithm selected by the experiment is derived from four LBSN clustering algorithms in literature [4], [6], [8], respectively, edge-cluster, loc-focus method1, loc-focus method2 and LDA. Figure 14 shows the algorithm experimental comparison results of the social closeness with different membership threshold.

The user community membership matrix is obtained due to the community discovery algorithm adopted in this paper. Therefore, it is possible to control the overlap degree of the community by setting the threshold value of community membership. The lower the threshold sets, the greater the overlap gets. As can be seen from the above 5 images,

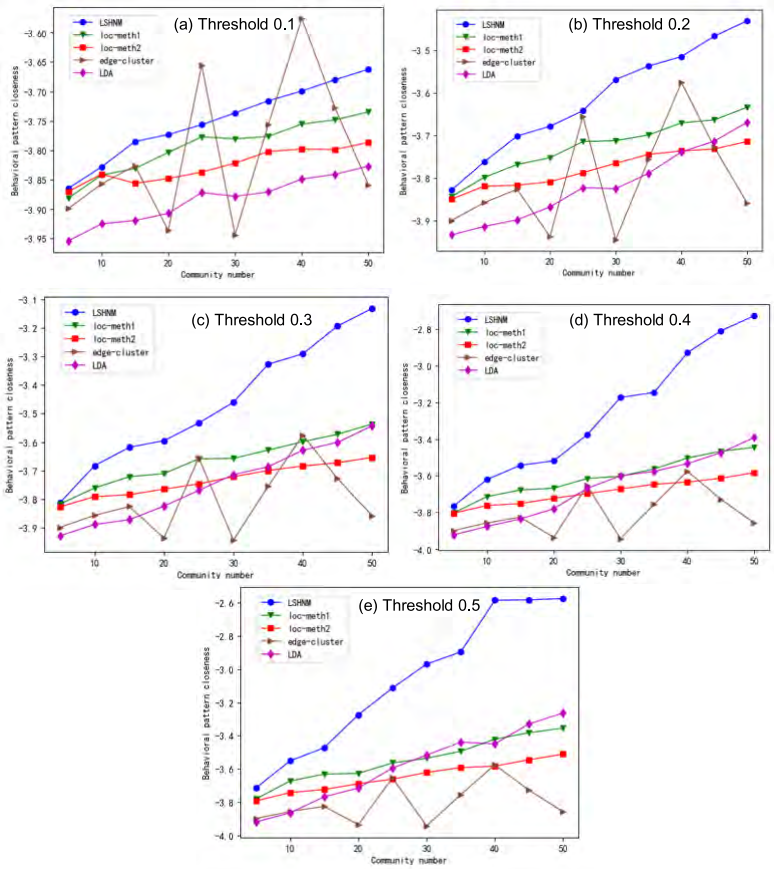

FIGURE 16. Behavioral pattern closeness contrast result.

the effect of the LSHNM algorithm in this paper is much higher than the edge-cluster and LDA algorithm in the social closeness evaluation standard, but it is inferior to the loc-focus method 1 and loc-focus method2 algorithm under the low degree of membership threshold. This is because the loc-focus series algorithms give weights directly on social edges and delete low weight edges which will result in generating more isolated nodes. Hence, this two algorithms focus more on social relationships and find smaller user communities. However, with the increase of membership threshold, the number of nodes in each community is reduced, so the communities discovered by the LSHNM have only a small number of core nodes. While the core nodes in the same community have strong correlations under social relationships, temp-spatial distribution, and behavioral patterns, which will weak the focus of different algorithms. So the LSHNM algorithm can also achieve the matching community discovery effect of the loc-focus series algorithm in high community membership threshold. Figure 15 shows the algorithm experimental comparison results of the temp-spatial distribution closeness with different membership threshold.

According to Figure 15, LSHNM algorithm has significant advantages on temp-spatial distribution closeness evaluation standard over four kinds of comparison algorithms, and with the increase of membership threshold the effect is gradually enhanced. Moreover, the experimental results of the edge-cluster algorithm fluctuates violently and the stability is poor. This is because the implementation of the edge-cluster algorithm uses the K-means clustering algorithm. The K-means algorithm randomly selects k points as the cluster center points at the beginning of the algorithm. If the selected center point is a noise point or an outlier, it will have a great impact on subsequent algorithm steps. Therefore, the K-means algorithm is very sensitive to the selection of the initial center points of the clusters, which results in a large fluctuation of the experimental results. Figure 15 (a-e) are the algorithm experimental comparison results of the behavioral pattern closeness with different membership threshold:

According to the above five figures, LSHNM algorithm has significant advantages on behavioral pattern closeness evaluation standard over four kinds of comparison algorithms, and with the increase of membership threshold the effect is gradually enhanced. The results of the edge-cluster algorithm still fluctuate drastically, and the reason has been explained in the above experiments. Finally, this experiment also compares the operating efficiency of each algorithm. The following table is the experimental result

It can be seen from the above table that the LSHNM model and loc-focus series algorithms have high operating efficiency and are suitable for application in large-scale social

TABLE 2. Comparison of experimental results.

Algorithms	LSHNM	Loc_1	Loc_2	$Edge_c$	LDA
Times	5s	5s	5s	$\geq 12hr$	16 min

networks. However, the performance of other algorithms is far inferior to the above two algorithms, especially edge-cluster algorithm. This is because the edge-cluster algorithm needs to continuously compare the similarities of any two edges in the network. However, the number of edges in the network is very large, far exceeding the number of nodes, which causes the algorithm time complexity to rise sharply.

The results of LSHNM algorithm on the social relation closeness experiment are lower than that of loc-focus series algorithms under the low community membership threshold, as shown in Figure 16, but it can also achieve the matching community discovery effect of the loc-focus series algorithms with the increase of the community membership threshold. And the LSHNM algorithm is superior to the above algorithms in the experimental results of temp-spatial distribution closeness and behavioral pattern closeness. Comparing the experimental results of the LSHNM algorithm and the edge-cluster algorithm, the performance of the edge-cluster algorithm is not as good as that of the LSHNM algorithm under all evaluation indicators. In addition, edge-cluster algorithm has poor stability and high time complexity. As for LDA algorithm, it does not have any outstanding performance. According to the above three comparison experiments, this paper shows that the LSHNM algorithm can find satisfied community structure in LBSN.

V. CONCLUSION

This paper focus on community discovery problem in LBSN from user social relations, temp-spatial distribution and behavioral patterns characteristics. Based on the massive new form of social data, an effective method LSHNM is proposed which reconstruct LBSN topology by integrating above user characteristics and find suitable communities. After above experimental comparison, LSHNM has a great advantage over other algorithms. In the future, the user's trajectory data can be analyzed in detail to find more realistic communities.

REFERENCES

[1] S. Xu, J. Cao, P. Legg, B. Liu, and S. Li, "Venue2Vec: An efficient embedding model for fine-grained user location prediction in Geo-social networks," *IEEE Syst. J.*, to be published.

[2] Z. Liu, B. Xiang, W. Guo, Y. Chen, K. Guo, and J. Zheng, "Overlapping community detection algorithm based on coarsening and local overlapping modularity," *IEEE Access*, vol. 7, pp. 57943–57955, 2019.

[3] J. Obregon, M. Song, and J. Jung, "InfoFlow: Mining information flow based on user community in social networking services," *IEEE Access*, vol. 7, pp. 48024–48036, 2019.

[4] S. Li, S. Zhao, Y. Yuan, Q. Sun, and K. Zhang, "Dynamic security risk evaluation via hybrid Bayesian risk graph in cyber-physical social systems," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 4, pp. 1133–1141, Dec. 2018.

[5] N. Alduaiji, A. Datta, and J. Li, "Influence propagation model for clique-based community detection in social networks," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 2, pp. 563–575, Jun. 2018.

[6] Z. Wang, D. Zhang, X. Zhou, D. Yang, Z. Yu, and Z. Yu, "Discovering and profiling overlapping communities in location-based social networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 44, no. 4, pp. 499–509, Apr. 2014.

[7] K. Joseph, C. H. Tan, and K. M. Carley, "Beyond 'local', 'categories' and 'friends': clustering foursquare users with latent 'topics'," in *Proc. ACM Conf. Ubiquitous Comput.*. New York, NY, USA: ACM, Sep. 2012, pp. 919–926. doi: 10.1145/2370216.2370422.

[8] C. Brown, V. Nicosia, S. Scellato, A. Noulas, and C. Mascolo, "The importance of being placefriends: Discovering location-focused online communities," in *Proc. ACM Workshop Workshop Online Social Netw.*. New York, NY, USA: ACM, Aug. 2012, pp. 31–36. doi: 10.1145/2342549.2342557.

[9] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera, "Detecting location-centric communities using social-spatial links with temporal constraints," in *Advances in Information Retrieval*, A. Hanbury, G. Kazai, A. Rauber, and N. Fuhr, Eds. Cham, Switzerland: Springer, 2015, pp. 489–494.

[10] X. Cheng, Y. Wu, G. Min, and A. Y. Zomaya, "Network function virtualization in dynamic networks: A stochastic perspective," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2218–2232, Oct. 2018.

[11] W. Miao, G. Min, Y. Wu, H. Huang, Z. Zhao, H. Wang, and C. Luo, "Stochastic performance analysis of network function virtualization in future Internet," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 613–626, Mar. 2019.

[12] K. Berahmand, A. Bouyer, and M. Vasighi, "Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 4, pp. 1021–1033, Dec. 2018.

[13] E. C. Hall, G. Raskutti, and R. M. Willett, "Learning high-dimensional generalized linear autoregressive models," *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2401–2422, Apr. 2019.

[14] L. Qi, Q. He, F. Chen, W. Dou, S. Wan, X. Zhang, and X. Xu, "Finding all you need: Web APIs recommendation in Web of things through keywords search," *IEEE Trans. Comput. Social Syst.*, to be published.

[15] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil, "An empirical study of geographic user activity patterns in foursquare," in *Proc. 5th Int. AAAI Conf. Weblogs Social Media*, Jul. 2011, pp. 1–5.

[16] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[17] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," in *Proc. 12th Int. Conf. Inf. Knowl. Manage.*. New York, NY, USA: ACM, May 2007, pp. 556–559. doi: 10.1145/956863.956972.

[18] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 3, pp. 355–369, Mar. 2007.

[19] S. Li, S. Zhao, P. Yang, P. Andriotis, L. Xu, and Q. Sun, "Distributed consensus algorithm for events detection in cyber-physical systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2299–2308, Apr. 2019.

[20] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," *ACM SIGMOD Rec.*, vol. 27, no. 2, pp. 94–105, Jun. 1998,

[21] S. Cheng, B. Zhang, G. Zou, M. Huang, and Z. Zhang, "Friend recommendation in social networks based on multi-source information fusion," *Int. J. Mach. Learn.*, vol. 10, no. 5, pp. 1003–1024, Feb. 2018.

[22] SciPy.org. (2019). *Scipy Source Package*. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/index.html

[23] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: ACM, Sep. 2011, pp. 1082–1090.

[24] P. Laffort and A. Dravnieks, "An approach to a physico-chemical model of olfactory stimulation in vertebrates by single compounds," *J. Theor. Biol.*, vol. 38, no. 2, pp. 335–345, Feb. 1973.

[25] D. Kalman, "A singularly valuable decomposition: The SVD of a matrix," *College Math. J.*, vol. 27, no. 1, pp. 2–23, 1996.

[26] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, "Community discovery using nonnegative matrix factorization," *Data Mining Knowl. Discovery*, vol. 22, no. 3, pp. 493–521, 2011.

[27] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, "Detect overlapping and hierarchical community structure in networks," *Phys. A, Stat. Mech. Appl.*, vol. 388, no. 8, pp. 1706–1712, Apr. 2009.

JIUXIN CAO is currently with the School of Cyber Science and Engineering, Southeast University, and the Director of the Jiangsu Provincial Key Lab of Computer Networking Technology. His research interests include computer network technology and application, data analysis and privacy protection, and cyber physical system science.

WEIJIA LIU is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Southeast University. She has participated in the Interdisciplinary Contest in Modeling and received the Meritorious Winner Prize. Her research interest includes recommendation systems.

BIWEI CAO is currently pursuing the bachelor's degree (Hons.) in software engineering with The Australian National University (ANU), Her research interest includes data analysis.

PAN WANG is currently pursuing the master's degree with the Southeast University-Monash University Suzhou Joint Graduate School. His research interests include data analysis and social computing.

SHANCANG LI is currently with the Department of Computer Science and Creative Technologies, University of the West of England, Bristol, U.K. His current research interests include digital forensics for emerging technologies, cyber security, the Internet of Things (IoT) security, data privacy-preserving, the IoT, Blockchain technology, and the lightweight cryptography in resource constrained devices.

BO LIU is currently an Associate Professor with the School of Computer Science & Engineering, Southeast University, China. Her current research interests include online social networks and big data, which include spammer detection in social networks, the evolution of social community, and social influence, especially in multi-agent technology and in using agent technology to solve social network problems.

MUDDESAR IQBAL is currently a Visiting Senior Lecturer with the School of Computer Science and Electronic Engineering, University of Essex, U.K. His research interests include 5G networking technologies, Blockchain, artificial intelligence, social media profiling, mobile edge computing and fog computing, and the Internet of Things.