International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 9 Issue 1 ǁ 2021 ǁ PP. 58-66

Detecting Communities in Social Network Using Spectral Clustering

Hien-Trinh Nguyen[1] Thanh-Tung Cap[2] Vinh-Quang Vu[3]

[1]*Department of Computer Science, TNU-University of Information and Communication Technology, Viet Nam*
[2]*Department of Mathematics, TNU-University of Education, Viet Nam*
[3]*Department of Computer Science, TNU-University of Information and Communication Technology, Viet Nam*

Abstract

Social network has become diversified and conglomerate. This is an interaction network in society or relations between organizations or persons. Social network analysis (SNA) is network censoring to understand the sequence and behavior of participants. Exploring social network community has been an important research orientation of SNA. Network community structure provides detailed, comprehensive information about organization, behavior and function of network system. Social network is often expressed in the form of data graph structure. Hence, the mining of structure of social network is mainly associated with graph clustering problem. Many algorithms have been proposed to solve the problem. Here we introduce research results of spectral clustering approach to reduce the number of dimensions of data set along with optimization technique in order to reduce calculation complicity of algorithm. Experimental results on real data set have displayed the efficiency of proposed algorithm.

Keywords: *Social network, Community structure, Graph data mining, Graph clustering, Community determining, Spectrum.*

Date of Submission: 02-01-2021 Date of acceptance: 15-01-2021

I. INTRODUCTION

Clustering is the most important application for graph data mining problem, especially social network community finding. The first analysis of community structure was made by Weiss and Jacobsen [1] in the research on dividing civil servant working in a government organization into groups. Up to now many algorithms have been studied and developed. For example, Flake [2], Radicchi [3, 4]… proposed a method to find structure based on dividing G graph cluster into smaller sub-graph with unique characteristics. This problem belongs to NP- Hard class, Girvan and Newman [5] proposed hierarchical clustering to find community, by which one must calculate edge betweenness and then cut edge of the highest betweenness. The complexity of the algorithm is equivalent to O (k^2n) with k edges to be trimmed. In order to improve the speed of Girvan- Newman algorithm, some approaches were proposed by Tyler [6], Gregory [7], Brandes. However, the complexity remains high, about O(mn^2) with n as the number of vertex and m as the number of edges. The orientation proposed by the authors is aimed at reducing the number of dimensions of vector space (source data) by spectral clustering, then reducing calculation volume when the community structure is detected, so the calculation complexity is reduced correspondingly.

The structure of the article consists of 4 parts. Part 1 (Foreword): introduction of research contents. Part 2: summarizing basic knowledge regarding algorithms. Part 3: presentation of community detecting by spectral clustering, proposal of algorithm, experimental running on data set. Part 4: Conclusion.

Appraisal: As evidenced by experimental results on conglomerat data set shown in Table 3 and Fig 5-6, the community finding algorithm yields good results with higher speed (by an average of 30%) compared with algorithm of Ulrike Von Luxburg [13]. The quality of detected communities are also better (by an average of 23%). This proves the efficiency of the algorithm.

II. SUMMARIZING BASIC KNOWLEDGES

2.1 Social network graphs: Notation G = (V, E), where V is the set of vertices representing the members of the social network; E is the set of edges representing the social relationships among the members. A community C is a subset of vertices of V such that for each vertex $v \in C$ there are many edges connecting v_i with vertices *u* in C and few edges connecting v*i* with other vertices *w* in V \ C [9] , [10].

2.2.Similar graph: Notation G (V, E, W), where V = $\{X_1, X_2, ..., X_n\}$ is the set of vertices, E is the set of edges $\{(X_i, X_j)\}$ satisfying the measure W (X_i, X_j)> 0 where W is the measure of the similarity [10], [11]. The graph

G is divided so that the edges in the group have the greatest similarity and the edges connecting the groups have the smallest similarity measures. To represent a G graph, the following methods can be used:

- + Adjacent matrix $W = (w_{i,j})_{n\times n}$, where $\mathrm{w}_{i,j} = \begin{cases} 1, (i,j) \in E, \\ 0, (i,j) \notin E. \end{cases}$
- + The connection matrix $A = (a_{i,j})_{n\times n}$, determined by the measure W, is defined by the problem.
- + Degree matrix $D = (d_{i,j})_{n\times n}$,

$d_{i,k} = \begin{cases} d(v_i), i = k; \\ 0, i \neq k. \end{cases}$, where $d(v_i)$ is the degree of the vertex; G is an unweighted undirected graph.

$d_{i,j} = \begin{cases} \sum_{k=1}^{n} a_{\mathrm{ij}}, i \neq j; \\ 0, i = j. \end{cases}$, where a_{ij} is the connection value between the vertices, G is an weighted directed graph.

In research and experiments, the determination of the equivalence between 2 objects *Xi, Xj* is evaluated according to the Gaussian distribution:

$$\mathrm{W}(i,j) = \exp(-\frac{\|X_i - X_j\|^2}{2\sigma^2}) \qquad (1)$$

Where σ is the standard deviation. In the experiment, we choose value σ to adjust the cluster size. The higher the value of W (i, j), the stronger the association between Xi and Xj. In addition, the distance between 2 objects is also determined by the Euclidean distance:

$$d(i,j) = \|X_i - X_j\| \qquad (2)$$

Obviously the bond is higher if and only if the distance is smaller. Then, from the distance, it is possible to determine the linkage through many different methods. One of the methods we selected is the spectral method [12], [13], in order to reduce the number of dimensions of the data being examined and thus the procedure for determining the community on the graph is much more effective in terms of time and computational complexity.

2.3.Some concepts of spectrum:

Spectrum is a set of characteristic values of a matrix L: Spec(*L*) = $\begin{pmatrix} \lambda_1 \ldots \lambda_t \\ m_1 \ldots m_t \end{pmatrix}$

Where $\lambda_1 \ldots \lambda_t$ are eigenvalues, $m_1 \ldots m_t$ are the adjustment coefficients, L is a Laplace matrix. Laplace matrix has some basic properties [10], [14], [15]:

- + The sum of items on rows or columns is equal to zero.
- + L is a symmetric, inverse square matrix.
- + L is a positive half determination.
- + L is an operator $L: V \to R$, with V is the set of vertices in the graph G, R is the set of real numbers.
- + The eigenvalues of L are othonormal basic.
- + L depends on the order of vertices while spectrum is invariant for the graph.

The problem is that from the adjacent matrix A representing the graph, we need to build up an equivalent matrix L also capable of characterizing the graph that we are considering. There are many ways to construct an L matrix from A, for example: $L = I - D^{-\frac{1}{2}} A \mathrm{D}^{-\frac{1}{2}}$, or *L*=*D*-*A*.

From the equation $Lu = \lambda u$, we will determine the eigenvalues and the eigenvectors of L. The values of eigenvectors are the spectral values for the graph vertices and are used to compute clustering graph and are used to compute clustering the graph. In addition, the eigenvectors will be standardized for easy computation [16], [17]. Thus, the number of dimensions of the spectrum is less than the number of dimensions of the original set of vertices in the graph. From the adjacent matrix A of dimensions $n \times n$, we have transformed to proccesing problem with n elements of the eigenvector. The number of clusters detected in spectral clustering method corresponds to the *k* value that we select for the *K-mean* clustering technique.

2.4.Criteria for evaluating community discovery: Newman and Girvan [18], X. Liu et al [19] proposed the quantity Modularity to evaluate the quality of the detected community:

$$Q = \frac{1}{2m}\sum_{i,k=1}(a_{\mathrm{ij}} - p_{\mathrm{ij}})\delta(C_i, C_j) \tag{3}$$

Where *A* is the adjacency matrix, p_{ik} is the number of edges expected in C, $\delta(C_i, C_j) = 1$ if *i, j* belong to the same community and $\delta(C_i, C_j) = 0$ if vice versa. Based on the probability of connectivity between vertex i and vertex j we have:

$$Q = \frac{1}{2m}\sum_{\mathrm{ij}}(a_{\mathrm{ij}} - \frac{k_i k_j}{2m})\delta(C_i, C_j) \tag{4}$$

Where k_i, k_j is the degree of vertex *i* and vertex *j*. Denote n_c is the number of communities, l_c is the number of edges connecting the vertices of community C, *dc* is the total number of degrees of vertices of community C, we have:

$$Q = \sum_{c=1}^{n_c}(\frac{l_c}{m} - (\frac{d_c}{2m})^2) \tag{5}$$

The maximum value of Q is determined:

$$Q_{\max} = -\frac{1}{m}\min_C\{[(\mathrm{m}\text{-}\sum_{c=1}^{n_c} l_c) - (m - \sum_{c=1}^{n_c} E_x(l_c))]\} = -\frac{1}{m}\min_C\{|Cut_C|\text{-}\mathrm{E}_x\mathrm{Cut}_C\} \tag{6}$$

Where $E_x(l_c) = \frac{d_c^2}{4m}$ is the expected number of links, $|Cut_C| = m - \sum_{c=1}^{n_c} l_c$ is the number of intercommunities of C and $E_x Cut_C$ is the expected number of edges of C's communities. A community *C* with Q_{max} reaches positive value and the greater the community, the more clearly defined the community, ie the separation of the community is good.

III. COMMUNITY DETECTING BY SPECTRAL CLUSTERING

3.1.Spectral clustering problem and method

3.1.1 Problem: Consider the graph of network G = (V, E) with $V = \{v_1, ..., v_N\}$ is the set of vertices**,** a subset $Z \subset V$ with measure : $\mathrm{W}(Z_i, Z_j) = \sum_{i \in Z_i, j \in Z_j} A(i, j)$; where A is the connection matrix or the adjacency matrix. The value of subset Z is determined as: $Vol(Z) = \sum_{i \in Z} D_i$; where $D_i = \sum_{j=1}^{N} A(i, j); i = 1..N$ is the value of the ith vertex of the graph. Determine the set of non-empty sets $Z_1, ...Z_k$ such that $Z_i \cap Z_j = \phi$ and $Z_1 \cup ... \cup Z_k = V$; at the same time satisfy the good zoning criteria (in each group, the number of edges is the largest, but the number of edges between the two groups is the smallest).

3.1.2. Spectral clustering method: When the network is represented by graphs, community detection has a special relationship to graph clustering. Divide the graph G into 2 groups A, B such that the weights of the edges connecting vertices from A to the vertices of B are minimum [20], [21] and the edges in a group have high weight. Using the Min-cut method with slices, wij is the edge weight (i, j), the problem is done with choosing the slice to reach min, and reach max. Cut is done as standard:

$$J_{NCut}(A, B) = Cut(A, B)(\frac{1}{vol(A)} + \frac{1}{vol(B)}) \tag{7}$$

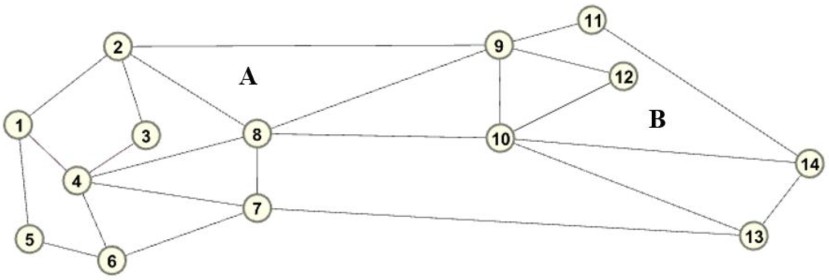


Figure 1. *The graph is divided into 2 clusters A, B*

Where :

$$Vol(A) = \sum_{i \in A} \sum_{j=1}^{n} \mathrm{W_{ij}} = \sum_{i \in A} d_i \tag{8}$$

$$Vol(B) = \sum_{i \in B} \sum_{j=1}^{n} \mathrm{W_{ij}} = \sum_{i \in B} d_i \tag{9}$$

This calculation has $O(|V||E|)$ complexity and will not perform cluster division if an isolated vertex is encountered. To overcome this, we will find other more efficient methods, one of which is to use spectral method [18], by using eigenvector $X = (v_1, v_2, ..., v_k)$ with L = D-A. Or use eigenvector $Y = (u_1, u_2, ..., u_k)$ with $L = D^{-\frac{1}{2}} A\mathrm{D}^{-\frac{1}{2}}$. The method of Ulrike Von Luxburg [13] has good clustering results with faster time than traditional clustering methods.

3.2. Our proposed algorithm

Our proposed method is based on the principle of using the eigenvectors of the Laplace matrix, transforming the set of original data objects into a set of points in the space whose coordinates are elements of vector. The points are then clustered using standard k-mean techniques. Unlike other methods, we choose the Gaussian distribution function to determine the value of the bonding matrix *A*, at the same time the k value is quantified by experience, then computes and selects k eigenvetor from the matrix *L* and perform *k-mean* clustering method on spectral set of original data set. The process of calculating eigenvalues and eigenvectors is also improved to reduce computation time (Algorithm 3.2.1). Existing methods use the estimation function corresponding to each data set to determine the connection matrix A. Through experimental process with many data sets (https://snap.stanford.edu), we found that the selection of Gaussian distribution function is very efficient, because in practice, most of the properties of the objects with their relationship obey the normal Gaussian distribution. The spectrum method can quickly calculate and show the most common and the best ability to enroll in the community, which is measured by the modularity value that determines the quality of the community.

Description of the algorithms:

```
Algorithm 3.2.1 (Determining the eigenvector of the matrix V)
Input: Matrix V has dimension: $n$ x $n$
Output: Eigenvetor of V
Begin
Step 1:
    Init $x = u^{(0)}$; $\lambda = \lambda^{(0)}$ assign $z^{(0)} = (x * x)^{-\frac{1}{2}} x$; such that $\|z^{(0)}\| = 1$. Let $t$ = 1.
Step 2:
    +  Defines the matrix $M = V - \lambda I$
    +  Solve the equation $My^{(t)} = z^{(t-1)}$; calculate $z^{(t)} = (y^{(t)^*} y^{(t)})^{-\frac{1}{2}} y^{(t)}$
Step 3:
    +  Calculate $\rho_t = \frac{z_{k_h}^{(t)}}{y_{k_h}^{(t)}}$; If V is the Hermitian matrix then calculate $\mu_t = (z^{(t-1)^*} y^{(t)})^{-1}$
    +  Calculate $\lambda^{(t)} = \lambda^{(t-1)} + \rho_{t^*}$; or $\lambda^{(t)} = \lambda^{(t-1)} + \mu_{t^*}$
Step 4:
    +  $t := t + 1$; Return to Step 2. Stop repeating if : $\|z^{(t)} - z^{(t-1)}\| \le \varepsilon$
End.
```

```
Algorithm 3.2.2 (Algorithm SC-NT)
Input: Given the data set $P \in R^{N\mathrm{x}F}$, $N$: number of data points, $F$ :number of dimensions, $k$: number of
        communities.
Output: Communities $Z_1, Z_2, ..., Z_k$; where $Z_i = \{\mathrm{i} | y_i \in C_i, \mathrm{i=1..k}\}$
Begin
1. For $P_i \in P$ ( $i \in 1..N$ ) if ( $conect_{P_j \in P}(P_i, P_j)$ then $A(i, j) = \exp(-\frac{\|p_i - p_j\|}{2\sigma^2})$ ; $i, j \in 1..N$
2. For $P_i \in P$ deg($v_i$)=deg($P_i$); $P_i \in P$
3. For $i, j \in 1..N$  $D_{ij} = diag(v_1, v_2, ..., v_N)$
4. Comput L=D-A // Laplace matrix
5. Comput k eigen-vector $u_1, u_2, ..., u_k$ of L so that: $Lu = \lambda u$
6. Select $u \in \{u_1, u_2, ..., u_k\}$ // select $u$ from set $\{u_1, u_2, ..., u_k\}$
7. Select $y \in \text{stand}\{u\}$ // Select $y_i \in R^k$ from the set of vectors u, normalize u get y
8. $C_i = k\_means(y)$, i=1..k // Clustering points $y_i$ into k cluster $C_1, C_2, ..., C_k$ according to k-Means Method
9. Return $Z_i = \{P_i \mid y_i \in C_j\}$
10. Comput Q
End.
```

Appraisal:

i. When using the Gaussian distribution function to determine the connection matrix *A*, the quality of the communities is better. The connection between the two objects *xi, xj* is strong if the objects are very similar. The similarity between two objects *xi, xj* is determined by the Gaussian distribution**.**

ii. Because the number of eigenvectors in *y* is much smaller than the number of dimensions *F*, it is obvious that the implementation of the *k_mean* algorithm on *y* will reduce the computational complexity throughout the space.

iii. Because the number of communities to be determined is *k* and from the k-Means algorithm (step 8), it is easy to see that the complexity of the algorithm is assessed to be equivalent to $O(k * N^2)$.

3.3. Experimental results

To evaluate the accuracy of the algorithm, we run the algorithm with some specific data sets. We use small datasets to test the accuracy of the clustering algorithm. We use large real data sets [22] to compare the proposed algorithm with other algorithms. The experiment was done on Matlab version 2019.

3.3.1 Example 1:

Consider the network graph including 8 objects A (5,3,2,3); B (9,7,8,8); C (7,5,6,8); D (6,8,7,9); E (7,8,9,8); F (8,9,7,6); G (5,7,7,6); H(8,6,9,8) and the links as corresponding graph represent the network (Figure 2):

The processing of algorithm is as follows:

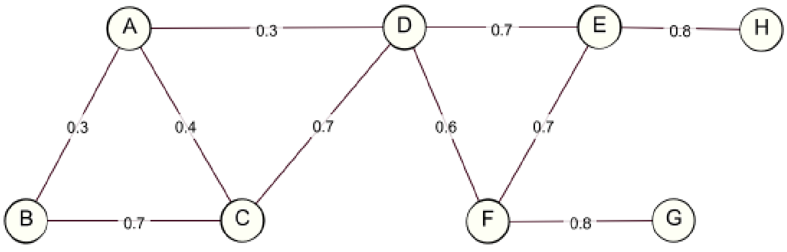


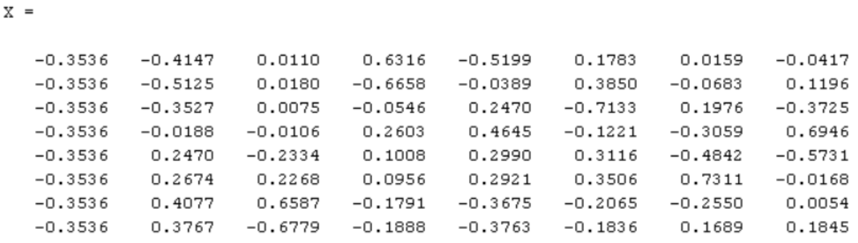
Figure 2. *Network with 8 vertices and weight links*

Step1, 2, 3: Values of connection matrix A (determined by the Gauss distribution) and order matrix D:

$$A = \begin{bmatrix} 0 & 0.3 & 0.4 & 0.3 & 0 & 0 & 0 & 0 \\ 0.3 & 0 & 0.7 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0.7 & 0 & 0.7 & 0 & 0 & 0 & 0 \\ 0.3 & 0 & 0.7 & 0 & 0.7 & 0.6 & 0 & 0 \\ 0 & 0 & 0 & 0.7 & 0 & 0.7 & 0 & 0.8 \\ 0 & 0 & 0 & 0.6 & 0.7 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 \end{bmatrix} \quad L = \begin{bmatrix} 1 & -0.3 & -0.4 & -0.3 & 0 & 0 & 0 & 0 \\ -0.3 & 1 & -0.7 & 0 & 0 & 0 & 0 & 0 \\ -0.4 & -0.7 & 1.8 & -0.7 & 0 & 0 & 0 & 0 \\ -0.3 & 0 & -0.7 & 2.3 & -0.7 & -0.6 & 0 & 0 \\ 0 & 0 & 0 & -0.7 & 2.2 & -0.7 & 0 & -0.8 \\ 0 & 0 & 0 & -0.6 & -0.7 & 2.1 & -0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.8 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & -0.8 & 0 & 0 & 0.8 \end{bmatrix}$$

Step 4: The matrix *L* is calculated: *L=D-A*

Step 5: Eigenvectors x is defined: $Lx = \lambda x$; We have eigenvalue λ

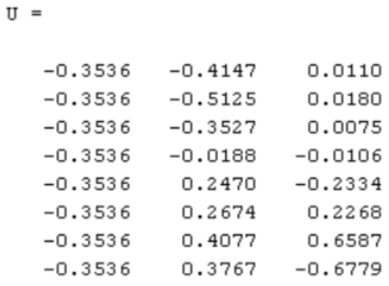


```
X =

   -0.3536   -0.4147    0.0110    0.6316   -0.5199    0.1783    0.0159   -0.0417
   -0.3536   -0.5125    0.0180   -0.6658   -0.0389    0.3850   -0.0683    0.1196
   -0.3536   -0.3527    0.0075   -0.0546    0.2470   -0.7133    0.1976   -0.3725
   -0.3536   -0.0188   -0.0106    0.2603    0.4645   -0.1221   -0.3059    0.6946
   -0.3536    0.2470   -0.2334    0.1008    0.2990    0.3116   -0.4842   -0.5731
   -0.3536    0.2674    0.2268    0.0956    0.2921    0.3506    0.7311   -0.0168
   -0.3536    0.4077    0.6587   -0.1791   -0.3675   -0.2065   -0.2550    0.0054
   -0.3536    0.3767   -0.6779   -0.1888   -0.3763   -0.1836    0.1689    0.1845
```

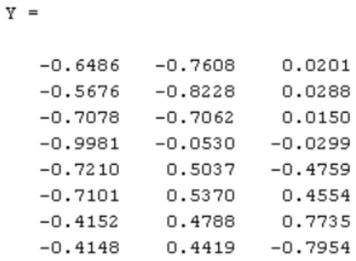
$$\lambda = \begin{bmatrix} 0 \\ 0.2754 \\ 0.5246 \\ 1.2272 \\ 1.4357 \\ 2.1580 \\ 3.0937 \\ 3.2854 \end{bmatrix}$$

Step 6, 7: Select $U \in R^3$; *k*= 3; Normalizing *U* we have *Y*:



```
U =

   -0.3536   -0.4147    0.0110
   -0.3536   -0.5125    0.0180
   -0.3536   -0.3527    0.0075
   -0.3536   -0.0188   -0.0106
   -0.3536    0.2470   -0.2334
   -0.3536    0.2674    0.2268
   -0.3536    0.4077    0.6587
   -0.3536    0.3767   -0.6779
```



```
Y =

   -0.6486   -0.7608    0.0201
   -0.5676   -0.8228    0.0288
   -0.7078   -0.7062    0.0150
   -0.9981   -0.0530   -0.0299
   -0.7210    0.5037   -0.4759
   -0.7101    0.5370    0.4554
   -0.4152    0.4788    0.7735
   -0.4148    0.4419   -0.7954
```

Step 8, 9: Clustering for points $(Y_i), i = 1..8$ into three clusters $1, 2, 3$ using the k-Means algorithm:

Table 1. *The clustering results in example 1*

Objects	Communities
A	1
B	1
C	1
D	1
E	2
F	3
G	3
H	2

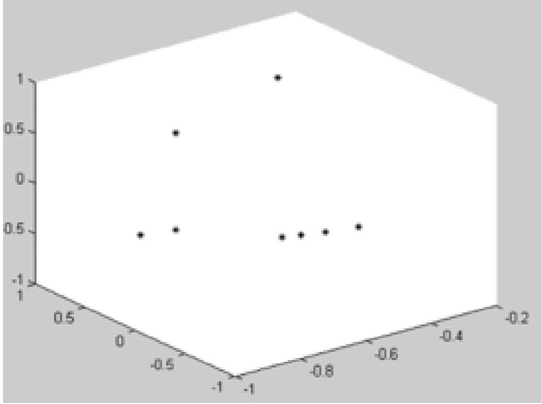


Figure 3. *Results received 3 communities*

Appraisal: The proposed algorithm has done good clustering, the obtained communities are reasonable.

3.3.2 Example 2. We use a large assumption data set, perform algorithm SC_NT and compare it with the UlrikeVon Luxburg algorithm [13]. We obtained the results in Table 2 (n - number of vertices, m-number of edges, k-number of communities, t-times).

Table 2. *Results comparing 2 algorithms (Running time: seconds)*

No	n	m	k	t_U_Von_LB	t_SC_NT
1	100	500	10	2,45	0,65
2	200	1000	25	7,9	1,85
3	400	2000	32	16,8	4,75
4	600	3000	43	21,3	11,8
5	800	4000	50	28,16	20,8
6	1000	5000	67	68,6	45,2

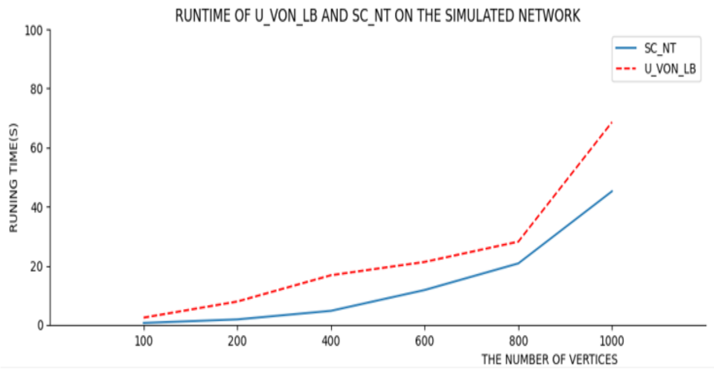


Figure 4. *Running time comparison*

3.3.3. Example 3. We used the international standard real data [22], tested the algorithm, calculated the quality of the community obtained and compared with the algorithm of UlrikeVon Luxburg [13]. The obtained results are shown in Table 3 (n - number of vertices, m - number of edges, k - number of communities, Q - community quality, t-time).

Table 3. *Results comparing 2 algorithms (Running time: seconds)*

No	Data set	n	m	k	U_Von_LB		SC_NT	
					Q	t	Q	t
1	Karate Club	34	78	2	0,31	1,35	0,45	0,53
2	Dolphin Group	62	159	3	0,35	1,55	0,49	0,86
3				2	0,53	2,05	0,55	0,97
4	Les Misérables Group	77	254	5	0,42	3,74	0,52	1,93
5	Book Amazon	105	441	3	0,47	3,93	0,58	1,26
6	Political blogosphere	1490	19090	3	0,49	77,35	0,67	55.23

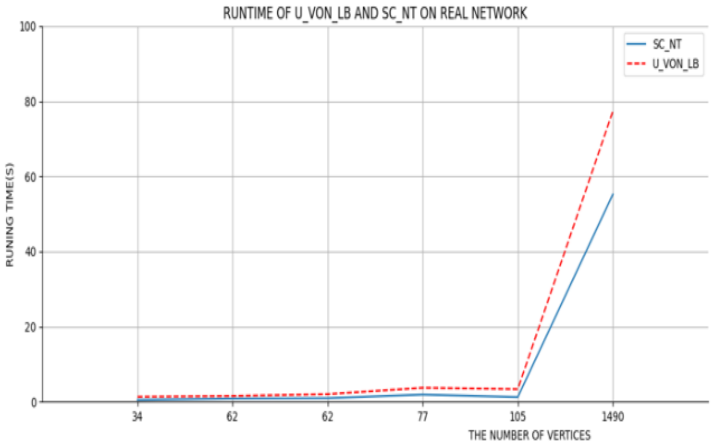


Figure 5. *Running time comparison*

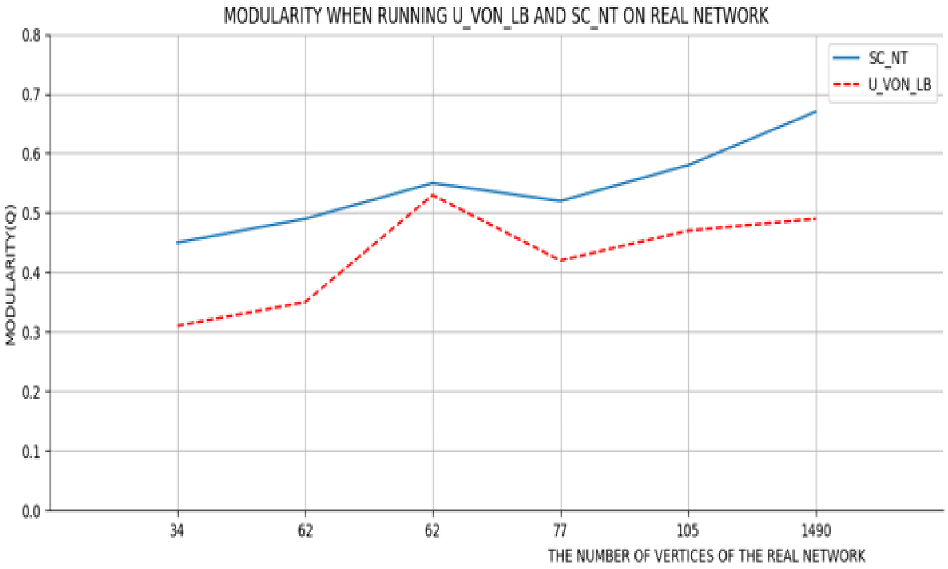


Figure 6. *Community quality comparison*

Appraisal: Through the experimental results on the large datasets given in Table 3 and Figures 5-6, we see that the proposed algorithm detected the communities as well. The runtime of the proposed algorithm is faster (on average about 30%) than that of UlrikeVon Luxburg [13]. The quality of the communities obtained is also higher (average 23%). This confirms the effectiveness of the proposed algorithm.

IV.CONCLUSION

We introduced a summarized approach to reduce the number of dimensions of data (in the form of matrix, may be multi- dimensional) to the vector form (real number chain), along with optimization Min-cut function with the help of Laplace matrix, so it is very efficient in data processing and finding community structure on social network. The key element of the approach is Spectrum technique (spectral clustering). The method ensures the discovery of reasonable number of network communities. We have proved that the proposed algorithm can be efficient in finding community and it can be applied to explore complicated structure of social network (which really mus be described as multi- dimensional space). The running time is much less than that of Ulrike Von Luxburg algorithm. The quality of detected communities is also better. In the coming time, the author group will extend its research, complete algorithm (for directional graphs, real data, multidimensional ...) and improve graph clustering technique in order to quickly detect high quality community for analysis and exploiting information on social network.

REFERENCES

[1]. R. S. Weiss, and E. Jacobsen (1999) “A Method for the analysis of the strucre of complex organizations” *American Sociological review*, vol. 20 , pp. 661-668.

[2]. G. W. Flake, and W. Lawrence (2000) “Efficient identiication of web communities” In Proceedings of the sixth ACM SIGKDD.

[3]. F. Radicchi, and F. Castellano (2004) “Defining and identifying communities in network” Proceedings of the National Academy of Sciences of the United States of America.

[4]. F. Radicchi, and S. Fortunato (2008) “Benchmark graphs for testing community detection algorithms” *Physical review E.*, vol. 78, pp. 046110.

[5]. M. Girvan M, and M. E. Newman (2002) “Community structure in social and biological networks” *Physical review E.*, vol. 99, no. 12, pp. 7821-7826.

[6]. J. R. Tyler, and D. M. Wilkinson (2003) “Automated discovery of community structure within organization,” *Physical review E,* vol. 15, pp. 723-739.

[7]. S. Gregory (2007) *An algorithm to find overlapping community structure in network*. Springer Heidelberg,.

[8]. U. Brandes (2007) “A faster algorithm for betweenness centrality” *Journal of Mathematical sociology*, vol. 2, pp. 163-177.

[9]. F. Harary (1996) *Graph Theory*. Addison Wesley Reading MA.

[10]. S. Fortunato (2010) “Community Detection in Graphs” *Physics Reports*, vol. 486, pp. 75-174.

[11]. V. Zografos, and K. Nordberg, (2012) “Introduction in Spectral Clustering,” *Physics Reports,* vol. 17, pp. 321-330.

[12]. D. Hamad (2014) “Constrained Spectral embedding for k-way data clusting” LISIC ULCO

[13]. U. von Luxburg, (2007) “A Tutorial on Spectral Clustering” Max Planck Institute for Biological Cybernetics.

[14]. L. C. Freeman (2007) “A set of measures of centrality based on betweenness” *Sociometry*, vol. 40, pp. 35-41.

[15]. M. Clarles (2012) “Spectral Clustering,” *A quick Overview”* vol. 22, pp. 115-124.

[16]. H. Abdi (2007) “The eigenvector-Decomposition” The University of Texas at Dallas.

[17]. B. Ruhnau (2015) “Eigenvector-centrality – a node-centrality” Social Networks, vol. 22, pp. 357-365.

[18]. M. E. Newman, and M. Girvan (2004) "Finding and evaluating community structure in networks" Phys Rev E Stat Nonlin Soft Matter. Phys, vol. 21, pp. 235-251.

[19]. X. Liu, H. M. Cheng, and Z. Y. Zhang, (2019) "Evaluation of community detection methods" Physics Reports, vol. 10, pp. 251-265.

[20]. S. M. Wagner (2007) "A simple min cut algorithm," *J.ACM*, vol. 44, pp. 585-591.

[21]. Wagner (2013) "Between min cut and graph bisection" London Springger, vol. 711, pp. 744-750.

[22]. J. Leskovec, and Krev (2014) "A. SNAP Datasets tanford large network dataset collection". [Online]. Available: https://snap.stanford.edu. [Accessed Oct. 15, 2020].