AUT Journal of Mathematics and Computing

AUT J. Math. Comput., 3(2) (2022) 153-164
DOI: 10.22060/AJMC.2022.20727.1074

Original Article

Using principal eigenvectors of Laplacian-Plus matrix to identify spreaders of social networks under linear threshold diffusion model

Neda Binesh[a], Mehdi Ghatee*[a]

[a] *Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran*

ABSTRACT: Influence maximization (IM) is a challenging problem in social networks to identify initial spreaders with the best influence on other nodes. It is a need to solve this problem with the minimum diffusion time and the most coverage on the communities. However, the spreaders are rarely dependent on diffusion models. A recent research [N. Binesh, M. Ghatee, Distance-Aware Optimization Model for Influential Nodes Identification in Social Networks with Independent Cascade Diffusion, Information Sciences, 581 (2021) 88-105] proposed *DASF* algorithm for spreaders selection by the Independent Cascade (IC) diffusion model. Here, we present a new optimization model to find spreaders under Linear Threshold (LT) diffusion model. LT is one of the most important models to imitate the behavior of influence propagation in social networks. Our model is a quadratic programming problem based on *Laplacian-Plus* matrix. We derive its solution by some principal eigenvectors of *Laplacian-Plus* matrix. We organize the solution process as *DALT* algorithm. Without community detection, it can identify the spreaders with maximum inter-communities distance, minimum intra-communities distance, and the most significant degrees. By considering various well-known social networks, we show that *DALT* provides brilliant results and overcomes other local and global spreader finders, especially in social networks with community structures.

Review History:

Received:30 October 2021
Accepted:10 February 2022
Available Online:01 September 2022

Keywords:

Social networks
Influence maximization
Linear threshold model
Laplacian-Plus matrix
Eigenvectors

AMS Subject Classification (2010):

91D30; 90B10; 90-08

(Dedicated to Professor S. Mehdi Tashakkori Hashemi)

1. Introduction

Social networks have increasingly grown in different applications. Identifying influential nodes (spreaders) in these networks is a challenging problem. This problem depends on network structure, centrality, and communities [5]. When the number of spreaders is given, diffusion models [11] can be used to select the spreaders. The Independent Cascade (IC) model and the Linear Threshold (LT) model are two basic models for stochastic information diffusion [14]. In the IC model, for each inactive node, each edge has a threshold and, the node will be activated separately from its active neighbors. However, in the LT model, each node has a threshold that compares with the summation of the active neighbor's weight. When the sum of the edge weights is more than this threshold, the node will be activated. These models are progressive diffusion models, which means the activated nodes cannot be deactivated in later steps. To study a survey on typical non-progressive diffusion models, one can refer to [17]. The spreaders under these non-progressive models are different. Thus, in the current paper, we focus on the spreader selection problem under the LT model. The following concepts are important:

*Corresponding author.
E-mail addresses: n.binesh@aut.ac.ir, ghatee@aut.ac.ir

- In the LT model, the spreaders are located closely with more common neighbors.
- Most of the spreader selectors are either local or global. The first methods consider local features such as degrees and neighbors. However, nodes' positions, distances, and distributions do not affect the selection process. The complexity of these methods is low as they do not consider the entire network simultaneously. However, there is no guarantee to find separated spreaders in all network components [1]. On the other hand, global methods take node positions and distances. They scatter spreaders to cover all network components. These methods also use some local parameters such as node degrees to improve the quality of spreaders [19]. Thus, global methods are more interesting for social network applications [12]. We propose a method to consider both local and global properties efficiently.

Our LT-based model includes the following contributions:

- Creating a new regularized distance to consider both local and global features based on LT model,
- Modeling the spreaders selection problem to affect most nodes in a short time and cover most of the communities,
- Using a regularized distance matrix to find appropriate spreaders diffused in the entire network with no need to detect communities directly.

By solving this model, we propose a spectral greedy algorithm, namely *DALT*. Although the role of eigenpairs for community detection and graph partitioning has been proved [6, 7, 24], based on the best of our knowledge, there is not any research on LT-based spreader selectors. To cover this gap, *DALT* applies the following novelties:

- To reduce the computation time, it first selects some anchor nodes as candidates for spreaders.
- For considering the distance between spreaders, it uses the social distance defined in [2] taking the neighborhood structures.
- For capturing the effects of the LT diffusion model in the selection of the spreaders, it regularizes the distances with appropriate mutual neighbors.
- It finds the appropriate spreaders under the LT diffusion model from several principal eigenvectors of *Laplacian-Plus* of the distance matrix.

We evaluate *DALT* on five well-known social networks with some measures.

2. Preliminaries and related works

Let $G = (V, E)$ represents a social network with notations indicated in Table 1. An active node $v \in V$ accepts the new information. The diffusion of information proceeds in discrete time steps ($t = 0, 1, 2, ..., T$) where T is the last step that all possible nodes are activated. A_0 denotes the set of initial spreaders, and A_T denotes the set of final active nodes under a diffusion model. A stochastic diffusion model specifies a randomized process of generating active sets A_t for all $t \geq 1$ by starting from A_0. The Influence Maximization (IM) problem finds A_0 such that A_T covers most of the nodes under a diffusion model. The *influence spread* $\sigma_G(A_*)$ is the expected number of nodes influenced byA_*. The following optimization problem defines A_0 with k spreaders:

$$A_0 = \underset{A_* \subset V \wedge |A_*| = k}{\arg\max} \quad \sigma_G(A_*) \tag{1}$$

2.1. Linear Threshold (LT) diffusion model

Linear Threshold (LT) is a popular model for diffusion in social networks [14]. Let i be active. i has an influential value on its neighbour, denoted as $\gamma(i, j)$. $\gamma(i, j)$ relates on real data analysis. Here, they are equal for all links. For node j with $D_{out}(j)$ neighbors, we set $\gamma(i, j) = \frac{1}{D_{out}(j)}$ for each $i \in N(j)$. Moreover, each node j has a random uniform value $\theta_j \in [0, 1]$ as its threshold, which changes randomly in different iterations. In the diffusion step t, node j will be activated where the summation of influential values from its active neighbors is greater than its threshold. It is equivalent to the following rule:

$$\text{if} \sum_{i \in \{N(j) \cap A_{t-1}\}} \gamma(i, j) \geq \theta_j \text{ then add j to } A_t \tag{2}$$

The diffusion process continuous until none exists to be activated. In LT mode, the probability of activation of j advances when its neighbors are activated in previous steps.

Table 1: Frequent notations used across the paper

Notation	Meaning
k	number of spreaders
G/V/E/ V'	a social network/ set of nodes/ set of links/ set of anchor nodes
n/m/ l / c	number of nodes/ number of links/ number of anchor nodes / number of clusters
Adj/W	adjacency matrix/ weight matrix
$D_{out} = [D_{out}(i)]$	vector of nodes' degrees in undirected graphs and out-degrees in directed graphs
t/T	time step in diffusion process/number of diffusion time steps
$\Delta(G)$ / $\langle D_{out} \rangle$	maximum degree of network's nodes/ average degree of the nodes
A_0/ A_t/ A_T	initial spreaders set/active nodes set in time step t/ final active nodes set
$\gamma(i, j)$	probability of information transfer from node i to node j in the LT model
$\theta(j)$	the random uniform number assigned to each node j in each LT iteration
$D_{bad}(i)$	number of neighbors of node i whose degrees are greater than or equal to $D_{out}(i)$
D_{nei}	neighborhood-degree index
$\hat{D}_{nei}$	a diagonal matrix that $\hat{D}_{nei}(i, i) = D_{nei}(i)$
d	number of random walk steps
Dis$\in \mathbb{R}^{l \times l}$	social distance matrix
$Dis_R \in \mathbb{R}^{l \times l}$	a regularized distance based on 4
$MN \in \mathbb{R}^{n \times n}$	a matrix containing the normalized number of mutual neighbors between all pairs of nodes in whole the network
$\hat{Dis}_R$	a diagonal matrix whose diagonal shows the summation of each row of matrix Dis_R
$\mathbf{s} \in \{0, 1\}^{l \times 1}$	a binary vector where $s(i) = 1$ when $i \in A_0$ and $s(i) = 0$ otherwise
$\mathbf{h}$	normalized vector of $\mathbf{s}$ (divided on its norm)
H$\in \mathbb{R}^{l \times c}$	a matrix whose columns are the principal eigenvectors corresponding to c largest eigenvalues of *Laplacian-Plus* matrix L^+.
$h_{max}(i)$	the maximum absolute value of the i^{th} row of matrix H and the i^{th} entry of $\mathbf{h}_{max}$
$\mathbf{h}_{max} \in \mathbb{R}^{l \times 1}$	a vector containing $h_{max}(i)$ values
$A'_0 \in \{1, 2, ..., l\}$	set of indices of the greatest values of $\mathbf{h}_{max}$
Diam	graph diameter that is the greatest shortest distance among all pairs of nodes
R	number of iterations in Monte Carlo simulation
IS	influence spread (average number of final active nodes in R iterations)
DR	diffusion rate (average number of final active nodes in R iterations in a unit time)

2.2. Related works

Finding the best spreaders to activate the most nodes is an NP-hard problem [4]. Usually, the researchers limit the number of spreaders with k and use some heuristics based on degrees, neighborhood structures, centrality, etc. Degree is used in many algorithms such as degree centrality (DC) index [13] and k-shell decomposition method [10] . Local Index Rank algorithm (LIR) [18] also combines the degree with a new local rank. There are various kinds of centrality measures such as betweenness centrality (BC) index [8], closeness centrality (CC) index [28] and generalized closeness centrality index (GCC) [19]. In some cases, user-specific features such as topical focus rate, activeness, authenticity, and speed of getting reaction are considered to select the spreaders [9]. In addition, probability-based methods and some greedy algorithms are proposed to find the influential spreaders for large-scale social networks, see, e.g. [15]. We also proposed a mathematical model to select the spreaders in the IC model in a recent work [2]. Based on the best of our knowledge, there is not any research on spreader selectors that uses local and global information jointly. In what follows, we extend the mentioned approaches to determine A_0 for the LT model.

3. Proposed Method

DALT is a special algorithm for finding spreaders in social networks with community structures. The details are described in the following subsections.

3.1. Anchor points selection

Similar to *DASF* method given in [2], the Neighborhood-degree (D_{nei}) index is used for anchor selection. It is defined as the following for any node i:

$$D_{nei}(i) = \frac{D_{out}(i)}{D_{bad}(i) + 1}, \tag{3}$$

where, $D_{bad}(i)$ shows the number of neighbors of i whose degrees are more than or equal to $D_{out}(i)$. A larger $D_{nei}(i)$ shows a better spreader. i is better than its neighbors when $D_{bad}(i) = 0$ and $D_{nei}(i) = D_{out}(i)$, so the nodes with the highest degrees will be chosen as anchor nodes. In the case of equal degrees, a node with the minimum $D_{bad}(i)$ is better. Further, many spreader selectors use node degrees as a key parameter. By combining both parameters, $D_{nei}(i)$ is an effective index to select brilliant spreaders. The anchor nodes V' can be selected as l nodes with the greatest $D_{nei}(i)$. l depends on the memory limitation. We changed l, but it does not affect the results significantly.

3.2. Distance between Anchor nodes

To scatter the spreaders in various communities, we need to maximize the distance between the spreaders. We use the social distance (Dis) between all pairs of anchor nodes defined in [2]. In the computation part of Dis, we use random walk similarities with $d = 4$ random walk step.

3.3. Spreaders selection under LT model

Since the activation of neighbors of each node in the LT model, advances its activation probability in the next step, we try to select spreaders with more common neighbors. In other words, nodes with the highest D_{nei} index are good candidates as the spreaders. Besides, they should be diffused in various communities of the social network. Our model follows these considerations to find spreaders under the LT model:

1. Maximize global coverage: The spreaders should be scattered in different communities with the longest global distances.

2. Maximize local overlap: The spreaders inside of a community become close together with minimum local distance. Thus, they support more common neighbors.

3. Maximize global coverage and minimize local overlap simultaneously: The spreaders with great D_{nei} probably meet these conditions.

To meet these needs, the distance between a pair of spreaders is regularized based on the number of their common neighbors. To this end, the distances with a normalized number of Mutual Neighbors (MN) can be used:

$$Dis_R(i,j) = \frac{Dis(i,j)}{MN(v_i,v_j)+1}, \tag{4}$$

where $MN(i,j)$ is the normalized number of mutual neighbors of i and j in whole network.

$$MN = Adj * Adj, \tag{5}$$

$$MN = \frac{MN}{max(MN)}. \tag{6}$$

Here '' $*$'' means the multiplication operator for two matrices. Now, we construct a new graph with V' nodes and assign $Dis_R(i,j)$ as the weights between anchor nodes i and j. If nodes i and j are close and have more common neighbors, the corresponding weight $Dis_R(i,j)$ is small. Also, for the nodes within different communities, this weight is high. If Dis_R is the regularized distance presented in 4, we can define $Set_{Dis}(A,B)$ for two node sets A and B as the following:

$$Set_{Dis}(A,B) = \sum_{a\in A}\sum_{b\in B} Dis_R(a,b). \tag{7}$$

Also $Set_{D_{nei}}(A)$ for the set A of nodes is stated as the following:

$$Set_{D_{nei}}(A) = \sum_{a\in A} D_{nei}(a) \tag{8}$$

For a network with c clusters, denotes c_i as the set of spreaders in community i. We have $A_0 = \bigcup_{i=1}^{c} c_i$. We state the following weighted objective function, where the terms of objective function refer to the maximizing the regularized distance between the spreaders within different clusters, minimizing the regularized distance between the spreaders in the same cluster, and selecting the spreaders with more common neighbors.

$$\max \sum_{i=1}^{c} (\omega_1 * Set_{Dis}(c_i,V) - \omega_2 * Set_{Dis}(c_i,c_i) + \omega_3 * Set_{D_{nei}}(c_i)) \tag{9}$$

For $i = 1,...,c$, we define the binary vector $\mathbf{s}_i \in R^{l\times 1}$, where $s_i(j) = 1$ if j is a spreader in cluster c_i. It is simple to show that the summation of distances between c_i and other nodes is $\mathbf{s}_i^T\ \hat{Dis}_R\ \mathbf{s}_i$ where $\hat{Dis}_R$ is a diagonal matrix whose diagonal shows the summation of each row of matrix Dis_R. Also the distance between the nodes of c_i is equal to $\mathbf{s}_i^T\ Dis_R\ \mathbf{s}_i$. In addition, $Set_{D_{nei}}(c_i)$ is equal to $\mathbf{s}_i^T\ \hat{D}_{nei}\ \mathbf{s}_i$, where $\hat{D}_{nei}$ is a diagonal matrix whose diagonal is D_{nei}, i.e. $\hat{D}_{nei}(i,i) = D_{nei}(i)$. So the objective function (9) can be rewritten as the following:

$$\max \sum_{i=1}^{c} (\omega_1 \ \mathbf{s}_i^T \ \hat{Dis}_R \ \mathbf{s}_i \ - \ \omega_2 \ \mathbf{s}_i^T \ Dis_R \ \mathbf{s}_i \ + \ \omega_3 \ \mathbf{s}_i^T \ \hat{D}_{nei} \ \mathbf{s}_i). \tag{10}$$

We replace $L^+ = \ \omega_1 \ \hat{Dis}_R - \ \omega_2 \ Dis_R + \ \omega_3 \ \hat{D}_{nei}$, which is referred as the *Laplacian-Plus* matrix. Eq. 10 can be simplified as:

$$\max \sum_{i=1}^{c} \mathbf{s}_i^T \ L^+ \ \mathbf{s}_i \simeq \max \sum_{i=1}^{c} \frac{\mathbf{s}_i^T \ L^+ \ \mathbf{s}_i}{\mathbf{s}_i^T \mathbf{s}_i}. \tag{11}$$

By setting $\mathbf{h}_i = \frac{\mathbf{s}_i}{\sqrt{\mathbf{s}_i^T \mathbf{s}_i}}$, we have

$$\max \sum_{i=1}^{c} \mathbf{h}_i^T \ L^+ \ \mathbf{h}_i \tag{12}$$

where $\mathbf{h}_i$ is the i^{th} column of matrix $H \in \Re^{n \times c}$ and

$$H(j,i) = \begin{cases} \frac{1}{\sqrt{|c_i|}}, & if \ j \ \in c_i, \\ 0, & otherwise. \end{cases} \tag{13}$$

$|c_i|$ is the number of spreaders in cluster i. The diagonal entities of $F1 = H^T L^+ \ H$, are equivalent to $\mathbf{h}_i^T \ L^+ \ \mathbf{h}_i$. Then, $F1(i,j)$ shows the summation of distances between spreaders in cluster i and cluster j with a negative sign, and $F1(i,i)$ accumulates the summation of distances between spreaders in cluster i and the other nodes minus the distance between spreaders in cluster i. Thus, maximizing the diagonal entities of $F1$ is equivalent to the maximization of the previous objective function (12). Thus, we can solve the following:

$$\max \ Tr \ (H^T \ L^+ \ H), \tag{14}$$

where H satisfies (13). Thus, this matrix has orthonormal columns, i.e. $H^T H = I_c$, but not vice versa (here I_c is $c - dimensional$ identity matrix). Now, by reducing the model (14), we have the following relaxed optimization problem:

$$\begin{aligned} \max \ Tr \ (H^T \ L^+ \ H), \\ s.t. \quad H^T \ H = I_c. \end{aligned} \tag{15}$$

Theorem 1. *The optimal solution of model (15) is a matrix H whose columns are the eigenvectors corresponding to c largest eigenvalues of Laplacian-Plus matrix L^+.*

Proof. Similar to Rayleigh-Ritz theorem [23], one can prove the result. For details, see, e.g. Section 5.2.2. of [20]. □

Now, according to the definition of matrix H in (13), we can select the maximum of absolute values of rows of H. The maximum elements are candidate for spreaders, i.e., $h_{max}(i) = \max_{j=1,\dots,c} \ |H(i,j)|$ and $\mathbf{h}_{max} = (h_{max}(i))$. Algorithm 1 shows the details of *DALT* for finding influential spreaders under the LT model.

Theorem 2. *Assume $DE(l)$ is the complexity of finding c largest eigenvalue of an $l \times l$-matrix. Algorithm 1 approximates the influential spreaders under the LT model that its complexity is $O(n^2 ld + DE(l))$.*

Proof. Since a relaxed version of model 14 is solved implicitly, the obtained solution is an approximation, not an exact one. To analyze the worst case of Algorithm 1, in Phase 1.1, we process all nodes with $O(n^2)$ operations. Phases 1.2 and 1.3 have linear complexities. Phase 2.1 needs $O(n^2 ld)$ iterations. Phase 2.2 executes on anchors and so its complexity is $O(l(l(\log l) + l + l^2 + 2l^2))$. Phase 3.1 needs l^2 iterations. Phase 3.2 also needs $DE(l)$ iterations to find c greatest eigenvalues and the corresponding eigenvectors. Phase 3.3 runs $O(kl)$ and Phase 3.4 and Phase 3.5 have linear complexities $O(l)$ and $O(k)$. This ends the proof. □

```
Algorithm 1: DALT: Distance-Aware Spreaders Selection under Linear Threshold (LT)
Input: $Adj$ , $n$ , $k$, $l$ , $d$ ; Output : Influential spreaders (set $A_0$)
  Phase 1. Anchor node selection:
  Input of Phase 1: $Adj$ and $l$ , Output of Phase 1: $V'$ and $D_{nei}$.
      1.1. Compute $D_{bad}(i)$ and $D_{out}(i)$ for each node $i$.
      1.2. Compute $D_{nei}$ index based on (3).
      1.3. Select $l$ anchor nodes with the largest $D_{nei}(i)$ and combine them in $V' = \{v_i : i \in \{1, 2, ..., l\}\}$.
Phase 2. Regularized distance computation:
  Input of Phase 2: $Adj$, $V'$, Output of Phase 2: $Dis_R$.
      2.1. Compute distance between all anchor pairs as matrix $Dis$ by the social distance of [2]
      2.2. For all $j \in \{1, ..., l\}$, compute the regularized distance $Dis_R(i, j)$ based on (4).
  Phase 3. Spreaders selection by the optimization model:
  Input of Phase 3: $V'$, $Dis_R$, $D_{nei}$ and $k$ , Output of Phase 3: $A_0$.
      3.1. Set $\omega_1 = \omega_2 = 1$, $\omega_3 = l$, and define Laplacian-Plus matrix $L^+ = \omega_1 \hat{Dis}_R - \omega_2 Dis_R + \omega_3 \hat{D}_{nei}$.
      3.2. Find $H$ including the eigenvectors of $c$ greatest eigenvalues of $\hat{L^+}$.
      3.3. For each node $i \in \{1, ..., l\}$, $h_{max}(i) = \max_{j=1,...,c} |H(i, j)|$ and $\mathbf{h}_{max} = (h_{max}(i))$.
      3.3. Define set $A'_0 \subset \{1, 2, ...., l\}$ including the indices of the $k$ greatest values of $\mathbf{h}_{max}$.
      3.4. Define the binary $l-$vector $\mathbf{s} = [s(i)]$ that $s(i) = 1$ for $i \in A'_0$.
      3.5. Define spreader set $A_0 = \{v_i : i \in A'_0\}$.
```

3.4. DALT results interpretation

In *DALT* model, each column k of H is an eigenvector of *Laplacian-Plus* matrix L^+. To present a simple interpretation, temporarily, consider Laplacian matrix L instead of L^+. Since $\hat{Dis}_R(i, j) = \sum_{j=1}^{n} Dis_R(i, j)$, we get:

$$(L\ h_k)(i) = \sum_{j=1}^{n} (\hat{Dis}_R(i, j) - Dis_R(i, j))h_k(j) = \sum_{j=1}^{n}\ Dis_R(i, j)(h_k(i) - h_k(j))$$

Based on the definition of eigenvector, we have $L\mathbf{h}_k = \lambda \mathbf{h}_k$ and so one can see that h is a linear combination as the following:

$$h_k(i) = \frac{(L\ h_k)(i)}{\lambda} \simeq \sum_{j=1}^{n} dis_R(i, j) * (h_k(i) - h_k(j)) \tag{16}$$

which can be rewritten as the following:

$$h_k(i) \simeq \sum_{j=1}^{n} Dis_R(i, j) * h_k(i) - \sum_{j=1}^{n} Dis_R(i, j) * h_k(j) \propto - \sum_{k=1}^{n} Dis_R(i, j) * h_k(j).$$

The value of $h_k(i)$ is the importance of node i as a spreader in community k. It depends on the importance of the other nodes with the highest values in $h_k(j)$ and with the least distance $Dis_R(i, j)$. This means that the spreaders are close to node i.

3.5. DALT vs. DASF

To compare *DALT* with our previous model *DASF* [2], the following criteria can be considered:

- Influencing a great number of nodes,
- Covering most communities.

In both methods, the same social distance *Dis* between the anchor nodes [2] is used. In *DALT*, we regularized this distance with MN values to define Dis_R to reflect the effects of the social distance and the mutual neighbor information simultaneously. Thus, the *Laplacian-Plus* matrix L^+ contains both effects, too. By optimizing the model 14, we select the spreaders with maximum inter-clusters distance, minimum intra-clusters distance, and the most significant degrees.

However, *DAFS* uses D_{nei} values to construct Dis'_R. [2] showed that by getting the principal eigenvector for the maximum eigenvalue of the regularized distance matrix Dis'_R, the spreaders under the IC diffusion model could be obtained. The same principal eigenvectors of the *Laplacian-Plus* matrix L^+ can also be used to select spreaders under the LT diffusion model. Thus the greatest entries of the principal eigenvectors of the different matrices Dis'_R andL^+ indicate the spreaders for IC and LT models, respectively. Table 2 summarizes the differences between *DALT* and *DASF* clearly.

Table 2: Comparison between properties of *DALT* and *DASF*([2])

Algorithm	Objective function	Mechanism to select spreaders and considerations
DASF	$\max_{\mathbf{h}\in\Re^n}\ \mathbf{h}^T Dis'_R\ \mathbf{h}$, s.t. $\mathbf{h}^T\mathbf{h} = 1$	• Using $Dis'_R = (\hat{D}_{nei})^{1/2} * Dis * (\hat{D}_{nei})^{1/2}$ causes to consider the distance and the neighborhood degree. • Finding the principal eigenvector of the regularized distance matrix Dis'_R and selecting spreaders by greedy approach. • $h(i) \propto \sum_{j=1}^{n} Dis'_R(i,j) * h(j)$, i.e., importance of node i depends on importance of the farthest nodes from i. • Influencing a large number of nodes by selecting the nodes with greatest D_{nei}. • Covering most communities by selecting the nodes with farthest distances.
DALT	$\max\ Tr\ (H^T\ L^+\ H)$, s.t. $H^T H = I_c$	• Using $L^+ = \omega_1\ \hat{Dis}_R - \omega_2\ Dis_R + \omega_3\ \hat{D}_{nei}$ leads to consider the maximum inter-clusters distance, minimum intra-clusters distance, and the greatest degrees. • Computing the matrix H containing c first principal eigenvectors of the *Laplacian-Plus* matrix L^+, and selecting spreaders by greedy approach on absolute values of each row of matrix H. • $h_k(i) \propto \sum_{k=1}^{n} -(Dis_R(i,j) * h_k(j))$, i.e. importance of node i depends on the importance of the closest nodes to i. • Influencing a large number of nodes by selecting the spreaders under three objectives: 1. maximum mutual neighbors 2. maximum inter-clusters distance, 3. minimum intra-clusters distance. • Covering most communities by selecting the farthest nodes in various communities.

4. Experiments and Results

The methodology of comparisons of [2] is followed in this section to give a fair evaluation of *DALT*. To this aim, *DALT* is compared with some well-known spreader finders including *Degree* [13], *LIR* [18], Local Centrality (*LC*) [3], *GCC* [19] and *DEIM* [16] on a computer with Intel(R) CoreTM i7-9700 CPU 3 GHz and 64 GB memory. For the spreaders of *CR*, we used the results of [27]. Also, the spreaders of *DIMM* and *ADIM* algorithms [25] are obtained by using their source codes [1]. Noting that, in *DALT*, the number of main communities should be stated. We set $c = k$ in the experiments to give an opportunity to select a spreader in every community. However, in some communities, the algorithm can select zero or multiple spreaders. Also, similar to [2], *Influence Spread* (*IS*) and *Diffusion Rate* (*DR*) measures are evaluated for all methods over $R = 1000$ experiments. *IS* indicates the average of the numbers of the final active nodes after a diffusion process as the following:

- $IS = (\sum_{i=1}^{R} |A_T(i)|)/R$,

where $A_T(i)$ shows the last set of activated nodes in the i^{th} experiment.

Also, *DR* measures the average of the last activated nodes in a constant time step as the following:

- $DR = (\sum_{i=1}^{R} |A_T(i)|)/(\sum_{i=1}^{R} T(i))$,

where $T(i)$ indicates the total diffusion time steps in the i^{th} experiment.

All of the methods are compared on several social networks [21] including Facebook[2], CAGRQC[3], Cahepth[4], Youtube[5] and EnronEmail [6]. Details are given in Table (3) [2]. In this table, $0 - LI$ denotes a parameter of *LIR* algorithm [18] and means $LI = 0$.

[1] https://colab.research.google.com/

[2] http://snap.stanford.edu/data/egonets-Facebook.html

[3] http://snap.stanford.edu/data/ca-GrQc.html

[4] http://snap.stanford.edu/data/ca-HepTh.html

[5] http://snap.stanford.edu/data/com-Youtube.html

[6] https://snap.stanford.edu/data/email-Enron.html

Table 3: The properties of datasets of the experiments [2].

Name	Type	n	m	$\sharp$0-LI	Diam.	$\Delta(G)$	$< deg >$
Facebook	undirected	4039	88234	5	8	1045	44
CAGRQC	undirected	5242	14496	837	17	81	6
Cahepth	undirected	12008	118521	664	13	491	20
Youtube	undirected	19017	119470	19	10	1129	13
EnronEmail	undirected	36692	183831	2467	11	1383	10

Table 4: The effects of the number of anchors ($|V'|$) on the spreaders of Facebook network.

$\|V'\|$	$0.01 * n$	$0.1 * n$	$0.3 * n$	$0.5 * n$	$0.7 * n$	$0.9 * n$
Computation time	0.5033	0.7071	1.6181	3.7286	7.7956	13.6990
IS	1415.70	1426.13	1459.02	1431.25	1419.94	1449.44
DR	44.11	45.33	47.79	45.21	45.70	45.13
spreader-1	107	107	107	107	107	107
spreader-2	1912	1912	1912	1912	1912	1912
spreader-3	3437	3437	3437	3437	3437	3437
spreader-4	1684	1684	1684	1684	1684	1684
spreader-5	1888	1888	1888	1888	1888	1888
spreader-6	2543	483	483	483	483	483
spreader-7	483	2543	2543	2543	2543	2543
spreader-8	2347	2347	2347	2347	2347	4039
spreader-9	4039	4039	4039	4039	4039	2347
spreader-10	686	686	686	686	686	686

4.1. Sensitivity analysis on the number of anchors

Table 4 presents the *IS* and *DR* results for various numbers of anchors ($l = |V'|$) on Facebook data set. For each l, we generate the corresponding *Dis* matrix. We assumed $k = 10$. It is worthwhile that most of the spreaders for different l values, appear in the same order. Regardless of their ranks, the sets of spreaders for all of these scenarios are similar. Thus, we can set $l = 0.1 * n$ in the next experiments.

4.2. Sensitivity analysis on Laplacian-Plus matrix

Let $\omega_1 = \omega_2 = 1$. For different ω_3 in Laplacian-Plus matrix definition, Table 5 gives *IS* and *DR* results on Facebook dataset. Since ω_3 changes Laplacian-Plus matrix L^+, it is important to analyze the spreader changes. Again, we considered $k = 10$. As seen, the sets of spreaders are similar for the last three columns for $\omega_3 = l, n, n^2$. In these columns, most of the spreaders appear in the same order. Thus, the importance weights of spreaders are almost stable. So, we can set $\omega_3 = l$ in the next experiments.

4.3. Comparisons between different methods

Table 6 compares the spreaders of *DALT* with other algorithms when $k = 10$ for different datasets. The fourth column gives the Matching Percentage (*MP*) of spreaders between *DALT* and other algorithms. The results of

Table 5: Results for different values of ω_3 to define Laplacian-Plus matrix for Facebook network.

ω_3	0	1	l	n	n^2
IS	126.65	1137.89	1440.26	1435.47	1459.36
DR	10.42	37.94	45.65	44.52	44.94
spreader-1	4023	107	107	107	483
spreader-2	4030	1912	1912	1912	1912
spreader-3	833	3437	3437	3437	686
spreader-4	824	1684	1684	1684	2543
spreader-5	705	686	1888	1888	107
spreader-6	694	4030	483	483	4039
spreader-7	3743	4023	2543	2543	3437
spreader-8	781	705	2347	2347	1684
spreader-9	3725	824	4039	4039	1888
spreader-10	830	833	686	686	2347

Degree, *LIR*, *GCC* and *DASF* are mostly matched with *DALT*, for different datasets. Thus, *DALT* retains the locality property of *Degree* and *LIR*, and the globality property of *GCC*. Since *DASF* algorithm is based on IC diffusion model, it is not fair to compare it with *DALT* that is based on LT diffusion model. However, this table shows that *DALT* overcomes on *DASF* for all datasets.

Table 6: The comparison between the selected spreaders by different algorithms for $k = 10$

Dataset	**Alg**	**Selected spreaders**	MP	IS	DR
Facebook	Degree	107, 1663, 1684, 1800, 1888, 1912, 2347, 2543, 3437, 4039	0.8	1352.10	43.48
	LIR	1, 2, 3, 4, 5, 107, 686, 1912, 3437, 3980	0.4	884.35	32.37
	LC	107, 1199, 1352, 1431, 1663, 1800, 1888, 1912, 2347, 2543	0.5	775.10	26.06
	GCC	107, 483, 1663, 1800, 1888, 1912, 2347, 2543, 3437, 4039	0.8	1133.80	37.42
	CR	3437, 107, 1684, 1912, 4039, 3980, 414, 348, 686, 698	0.6	1420.82	47.35
	DIMM	107, 1912, 1684, 3437, 348, 414, 686, 3980, 698, 2047	0.5	1318.26	42.82
	ADIM	107, 1912, 1684, 3437, 348, 414, 686, 3980, 1086, 2047	0.5	1343.40	42.76
	DEIM	3437, 107, 3460, 1684, 4039, 1574, 3604, 41, 3980, 2949	0.4	1057.64	38.67
	DASF	107, 686, 1684, 1888, 1912, 2347, 2543, 3437, 3980, 4039	0.9	1381.58	45.29
	DALT	3437, 1912, 1684, 107, 4039, 686, 2543, 1888, 483, 2347	1	**1433.58**	**48.14**
CAGRQC	Degree	5591353, 1975, 2466, 3495, 3893, 4234, 4283, 4325, 4554	0.2	210.31	17.12
	LIR	555, 897, 1563, 2727, 2765, 2775, 2956, 4234, 4543, 4630	0.8	230.64	17.96
	LC	559, 1353, 1975, 2466, 2902, 3495, 4234, 4283, 4325, 4554	0.2	213.43	16.66
	GCC	1353, 1975, 2466, 3495, 3893, 4185, 4234, 4283, 4325, 4554	0.2	217.49	17.39
	CR	108, 2138, 11, 1731, 1137, 53, 1118, 315, 20, 123	0	70.69	8.33
	DIMM	1486, 432, 286, 2003, 1635, 570, 2765, 255, 617, 494	0	86.38	9.22
	ADIM	1486, 432, 2003, 1635, 255, 494, 286, 544, 617, 1352	0	83.48	9.06
	DEIM	212, 2619, 634, 697, 1977, 2604, 410, 241, 2727, 4524	0.1	164.52	16.40
	DASF	112, 897, 1352, 1832, 2765, 2775, 2956, 4234, 4283, 4543	0.6	224.35	17.32
	DALT	4234, 2956, 2727, 4283, 4543, 2775, 897, 1281, 555, 4630	1	**269.48**	**20.33**
Cahepph	Degree	1157, 2413, 4128, 4289, 4517, 4989, 5484, 8846, 9647, 10135	0.9	1317.44	41.38
	LIR	908, 2930, 3652, 4296, 4580, 5997, 6254, 9138, 9647, 11666	0.2	338.90	19.74
	LC	1157, 2413, 4289, 4517, 4989, 5484, 6992, 9647, 10135, 11877	0.8	1342.09	41.50
	GCC	1326, 1354, 2689, 4768, 7964, 9647, 9979, 10412, 11774, 11904	0.1	263.05	17.03
	CR	1056, 564, 788, 976, 464, 4549, 1284, 871, 2157, 879	0	119.38	10.55
	DIMM	1425, 1274, 1157, 166, 1009, 2695, 989, 757, 354, 1071	0.1	790.75	31.19
	ADIM	1425, 166, 1274, 1157, 1009, 2695, 989, 757, 3965, 1096	0.1	863.73	32.99
	DEIM	5666, 1157, 4650, 5245, 154, 7455, 1096, 662, 11792, 1712	0.1	824.71	30.02
	DASF	1157, 2413, 3652, 4296, 4517, 5997, 6848, 7031, 9647, 11666	0.5	813.48	30.08
	DALT	4517, 5484, 2413, 9647, 1157, 3652, 10135, 4289, 4989, 8846	1	**1423.36**	**42.74**
Youtube	Degree	50, 180, 245, 438, 492, 1713, 2221, 2499, 2535, 4418	0.6	4675.78	147.2
	LIR	106, 245, 492, 4808, 6343, 6862, 10293, 11201, 11939, 18855	0.5	2639.51	98.14
	LC	50, 83, 92, 245, 438, 1100, 1533, 1713, 2011, 2216	0.3	3464.82	110.03
	GCC	50, 58, 245, 438, 492, 1713, 2221, 2499, 2535, 4418	0.5	4411.8	138.26
	CR	Not Reported	-	-	-
	DIMM	492, 245, 4418, 180, 50, 2499, 2535, 1713, 3206, 6343	0.6	4555.69	147.61
	ADIM	245, 492, 180, 50, 2499, 4418, 2221, 438, 2535, 3206	0.6	4653.46	146.19
	DEIM	180, 492, 3206, 2499, 1071, 1204, 58, 63, 6343, 3754	0.3	3488.66	136.12
	DASF	50, 106, 180, 245, 492, 4418, 6343, 6862, 10293, 11201	0.80	249.86	21.05
	DALT	106, 50, 180, 6343, 438, 4418, 492, 245, 4152, 6862	1	**4830.7**	**152.58**
EnronEmail	Degree	136, 140, 195, 273, 370, 458, 566, 1028, 1139, 5038	0.8	**8209.39**	**275.11**
	LIR	273, 458, 5022, 5036, 5038, 5069, 9137, 9504, 20764, 26576	0.4	3918.05	182.74
	LC	76, 136, 175, 195, 273, 292, 370, 416, 734, 1028	0.4	6091.95	199.87
	GCC	136, 140, 273, 370, 458, 566, 1028, 1139, 1768, 5038	0.7	7493.94	257.88
	CR	144, 80, 92, 191, 197, 148, 245, 244, 85, 93	0	968.37	58.44
	DIMM	5038, 458, 273, 140, 1028, 588, 566, 95, 370, 893	0.6	7611.94	258.12
	ADIM	5038, 458, 273, 140, 588, 566, 1028, 893, 370, 5030	0.6	7706.19	256.19
	DEIM	Out of memory	-	-	-
	DASF	140, 195, 273, 370, 458, 1028, 5022, 5036, 5038, 5069	0.7	6821.68	236.08
	DALT	458, 273, 5038, 195, 1139, 140, 1028, 370, 286, 543	1	8040.30	269.74

We conclude that *DALT* achieves the best *IS* and *DR* for all datasets except of Enron Email network. In this exception, the best *IS* and *DR* are obtained from *Degree*. However, in this dataset, *DALT* is the second rank. Thus, our proposed algorithm has surpassed most spreader finders in reasonable time steps.

On the other hand, LIR is a local algorithm whose running time is less than $DALT$. However, by selecting $0-LI$ nodes, LIR disconnects the spreaders. Moreover, IS and DR of $DALT$ are better than LIR. Also, in LIR algorithm, for some networks such as Facebook, the numbers of $0-LI$ were less than k. In these cases, LIR algorithm could not find any solution. For $1-LI$, LIR could not recognize any difference between nodes, while most of these nodes are not appropriate to consider as spreaders. Thus, one can prefer to use $DALT$ instead of LIR in different networks.

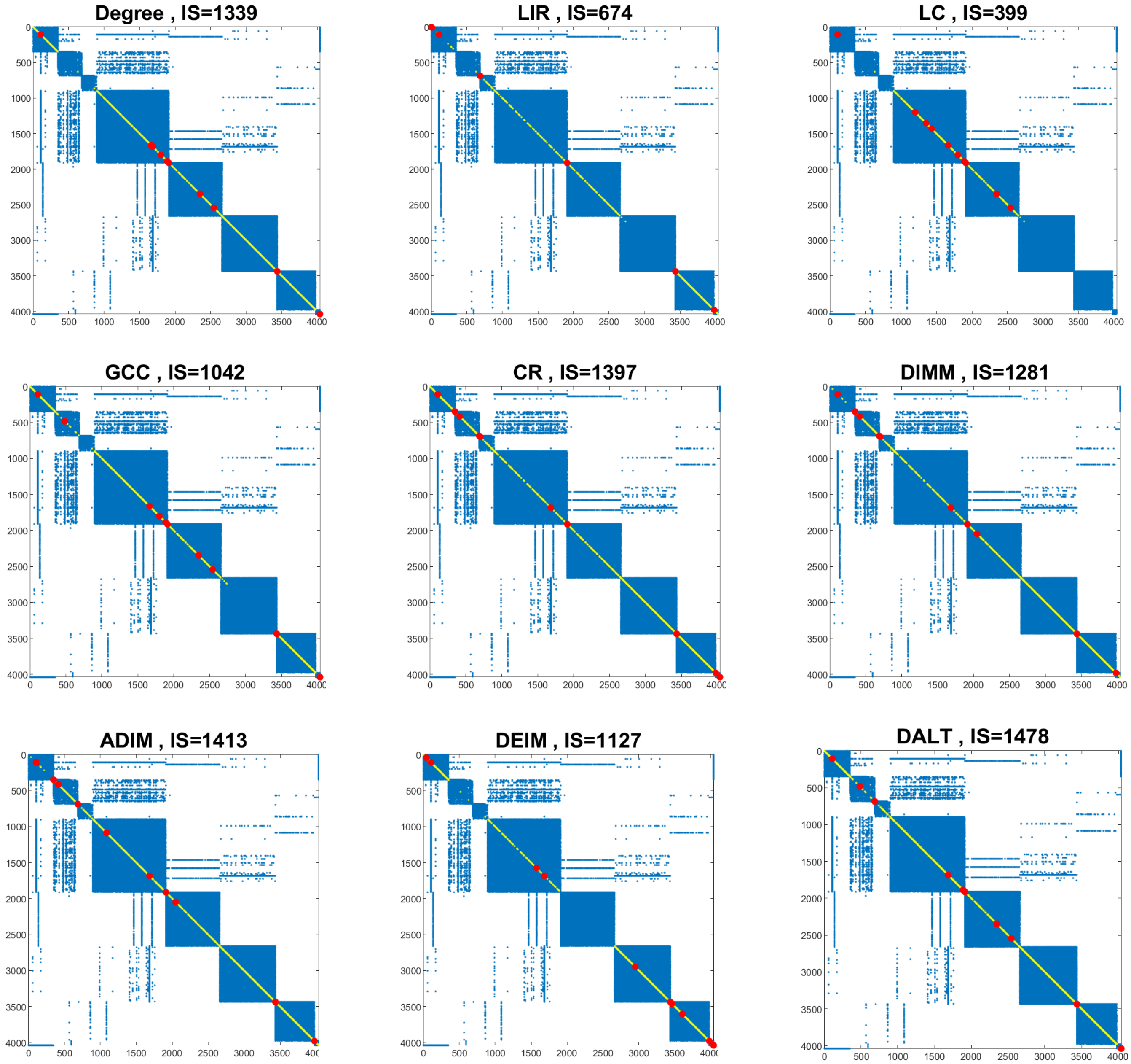

Figure 1: The adjacency matrix of Facebook network with Red points (initial spreaders of different methods), and yellow points (final activated nodes under LT model). Titles of sub-figures contain IS measures.

4.3.1. Visualization

The global methods can scatter the spreaders in whole social network. To investigate whether or not $DALT$ meets this property, Fig 1 visualizes the results of $DALT$ on Facebook network. In the adjacency matrix of this network, the adjacent nodes are in the same cluster, and so the shape of this adjacency matrix contains eight blue blocks on diagonal corresponding to its communities. Each sub-figure of Fig 1 shows ten spreaders of a method with red points, and the final activated nodes with yellow points under the LT diffusion model. The title of each sub-figure contains the corresponding *IS*. As seen, the IS of *DALT* is the best. Also, *DALT* scattered the spreaders in all of the clusters. Although *GCC* is a global algorithm for spreaders selection, it could not find appropriate spreaders to activate most of the nodes. Thus, *DALT* overcomes other methods under the LT diffusion model.

5. Conclusion

Finding an optimal subset of influential nodes as spreaders is an essential challenge in social networks. In the current paper, we proposed a global algorithm, namely *DALT* to find the spreaders under the LT diffusion model. This algorithm is stated based on a relaxed version of an optimization problem. In this model, the *Laplacian-Plus* matrix of a regularized distance matrix has a critical role. We derived the solution of this problem from some principal eigenvectors of the Laplacian-Plus matrices. The overall results showed the superiority of *DALT* in finding spreaders with high degrees and most diffusion in the whole of the social network communities. *DALT* increases the overlap between neighbors, approaches the spreaders inside the clusters, and scatter the spreaders in different communities. It also reduces the diffusion time steps and increases the *DR* measure.

Furthermore, other diffusion models can be used to select the spreaders [17]. They are classified into the progressive and the non-progressive models. Many progressive models were proposed as the extensions of IC and LT diffusion models. The modern algorithms similar to *DASF* and *DALT* followed progressive models. In future works, one can investigate the spreaders under the non-progressive models. Also, one can analyze the complexity of spreader selectors in the next works [22]. Presenting an approximation algorithm for the same purpose is also left to the future [26].

References

[1] M. Alshahrani, F. Zhu, L. Zheng, S. Mekouar, and S. Huang, *Selection of top-k influential users based on radius-neighborhood degree, multi-hops distance and selection threshold*, Journal of Big Data, 5 (2018), p. 28.

[2] N. Binesh and M. Ghatee, *Distance-aware optimization model for influential nodes identification in social networks with independent cascade diffusion*, Information Sciences, 581 (2021), pp. 88–105.

[3] D. Chen, L. Lü, M.-S. Shang, Y.-C. Zhang, and T. Zhou, *Identifying influential nodes in complex networks*, Physica A: Statistical Mechanics and its Applications, 391 (2012), pp. 1777–1787.

[4] W. Chen, L. V. Lakshmanan, and C. Castillo, *Information and influence propagation in social networks*, Synthesis Lectures on Data Management, 5 (2013), pp. 1–177.

[5] D. Fasino and F. Tudisco, *Spectral analysis of modularity matrices*, SIAM J. Matrix Anal. Appl, 35 (2014), pp. 997–1018.

[6] ——, *The expected adjacency and modularity matrices in the degree corrected stochastic block model*, Special Matrices, 6 (2018), pp. 110–121.

[7] ——, *A modularity based spectral method for simultaneous community and anti-community detection*, Linear Algebra and its Applications, 542 (2018), pp. 605–623.

[8] L. C. Freeman, *A set of measures of centrality based on betweenness*, Sociometry, (1977), pp. 35–41.

[9] Y.-H. Fu, C.-Y. Huang, and C.-T. Sun, *Using global diversity and local topology features to identify influential network spreaders*, Physica A: Statistical Mechanics and its Applications, 433 (2015), pp. 344–355.

[10] A. Garas, F. Schweitzer, and S. Havlin, *A k-shell decomposition method for weighted networks*, New Journal of Physics, 14 (2012), p. 083030.

[11] Y. Jiang and J. Jiang, *Diffusion in social networks: A multiagent perspective*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 45 (2015), pp. 198–213.

[12] K. Jung, W. Heo, and W. Chen, *Irie: Scalable and robust influence maximization in social networks*, in Data Mining (ICDM), 2012 IEEE 12th International Conference on, IEEE, 2012, pp. 918–923.

[13] D. Kempe, J. Kleinberg, and É. Tardos, *Maximizing the spread of influence through a social network*, in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2003, pp. 137–146.

[14] ——, *Maximizing the spread of influence through a social network*, Theory OF Computing, 11 (2015), pp. 105–147.

[15] M. Li, X. Wang, K. Gao, and S. Zhang, *A survey on information diffusion in online social networks: Models and methods*, Information, 8 (2017), p. 118.

[16] W. Li, K. Zhong, J. Wang, and D. Chen, *A dynamic algorithm based on cohesive entropy for influence maximization in social networks*, Expert Systems with Applications, 169 (2021), p. Article Number: 114207.

[17] Y. Li, J. Fan, Y. Wang, and K.-L. Tan, *Influence maximization on social graphs: A survey*, IEEE Transactions on Knowledge and Data Engineering, 30 (2018), pp. 1852–1872.

[18] D. Liu, Y. Jing, J. Zhao, W. Wang, and G. Song, *A fast and efficient algorithm for mining top-k nodes in complex networks*, Scientific Reports, 7 (2017), p. 43330.

[19] H.-L. Liu, C. Ma, B.-B. Xiang, M. Tang, and H.-F. Zhang, *Identifying multiple influential spreaders based on generalized closeness centrality*, Physica A: Statistical Mechanics and its Applications, 492 (2018), pp. 2237–2248.

[20] H. Lutkepohl, *Handbook of matrices.*, Computational statistics and Data analysis, 2 (1997), p. 243.

[21] J. McAuley and J. Leskovec, *Learning to discover social circles in ego networks*, NIPS, (2012).

[22] M. Niksirat and S. N. Hashemi, *The complexity of cost constrained vehicle scheduling problem*, AUT Journal of Mathematics and Computing, 2 (2021), pp. 109–115.

[23] L. N. Trefethen and D. Bau III, *Numerical linear algebra*, vol. 50, Siam, 1997.

[24] U. Von Luxburg, *A tutorial on spectral clustering*, Statistics and computing, 17 (2007), pp. 395–416.

[25] C. Wang, Q. Shi, W. Xian, Y. Feng, and C. Chen, *Efficient diversified influence maximization with adaptive policies*, Knowledge-Based Systems, 213 (2021), p. 106692.

[26] R. Yarinezhad and S. N. Hashemi, *Approximation algorithms for multi-multiway cut and multicut problems on directed graphs*, AUT Journal of Mathematics and Computing, 1 (2020), pp. 145–152.

[27] D. Zhang, Y. Wang, and Z. Zhang, *Identifying and quantifying potential super-spreaders in social networks*, Scientific Reports, 9 (2019), pp. 1–11.

[28] Z. Zhao, X. Wang, W. Zhang, and Z. Zhu, *A community-based approach to identifying influential spreaders*, Entropy, 17 (2015), pp. 2228–2252.

Please cite this article using:

Neda Binesh, Mehdi Ghatee, Using principal eigenvectors of Laplacian-Plus matrix to identify spreaders of social networks under linear threshold diffusion model, AUT J. Math. Comput., 3(2) (2022) 153-164
DOI: 10.22060/ajmc.2022.20727.1074