Extracting edge centrality from social networks using heat diffusion algorithm

Pegah Barekati*, Mehrdad Jalali, Majid Vafaei Jahan and Vahideh Amel Mahboob
Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

Abstract. Social networks are generally sets of individuals or organizations that are connected with one or more links. Usually, social networks are presented by undirected graphs, where the set of vertices \( V \) and the set of edges \( E \) state the individuals and relation between them respectively. One of the most applicable problems in these networks is the centrality values allocation problem to the vertices and edges. Recently, a new evaluation criterion for the edge centrality so called centrality index of \( k \) paths has been proposed which is based on intranet issuing the messages along with random paths composed of \( k \) edges. From the other side, it has been vivid the importance of computing the edges centrality through these years. In this study, by referring to message propagation along random paths, a new diffusion model was reached by applying heat diffusion algorithm. This model was based such that the vertex on the way of heat diffusion of most of vertices could be considered as an important node and it could obtain centrality edges by scoring the edges on the heat diffusion path. The proposed technique was compared with centrality drawing method by means of random paths of length \( k \) and the results of analyzing the algorithm performance on online large social networks’ data set show a remarkable efficacy of the proposed method to the mentioned method. Utilizing known large social networks in evaluation proves the efficiency of the proposed method for analyzing the large scale network.

Keywords: Social networks, edge centrality, betweenness centrality, heat diffusion

1. Introduction

Social network is to study social identities (individuals in an organization named user) and their interactions and relations. Interactions and relations could be represented by a network or graph, where each vertex shows a user and each link showing a relation. By having a network, it is possible to study its structure’s properties and the role, the position and the prestige of each social user. Moreover, it may construct various sub-graphs like networks composed of the groups of users.

Social network analysis is useful for web because a web is a society and a virtual social network in which every page is considered as a social network and each hyperlink is considered as a relation [11]. Many of outputs of social networks could be extracted and developed to be used in web. The ideas of analyzing the social network are constructive and effective, in fact, for web search engines success. In this regard, not only in scientific aspect, but also for commercial and strategic aims, identifying the main factors of a network is very important. So many researchers have introduced different problems and subjects about centrality efficiency, such as finding out where a disease spreads in medical data, managerial decisions for selling the products in commercial networks, finding political leaders in political networks, clustering and classification in social networks and etc. to recognize this, it is required to define an important criterion to assess the vertices and/or edges. The simplest methods of calculating the centrality just check the local topological characteristics of a vertex/an edge in a social network’s graph. As an instant, the plainest criterion of evaluating the vertex centrality is exhibited by a vertex degree which is the number of social relations of a user. The majority of local criteria of the centrality calculation, whose value is computationally possible even on large networks, do not give such trustworthy results [2].
For these reasons, in order to compute important values, many scholars have proposed to study the social network topology globally. Lately, a new family called global benchmarks has been generated for the centrality criteria. Examples of these are closeness centrality [3] and betweenness centrality (for vertices [4] and edges [5,6]).

Betweenness centrality is one of the most well-known evaluation benchmarks and computing it is a main component of many algorithms and applications. One Betweenness centrality is based on this concept that the information flow in the shortest paths in social networks: consequently, a vertex or an edge has betweenness centrality if lots of very short paths pass through it. Some of the researchers are worried about computing betweenness centrality. Firstly, computing precise value of betweenness centrality of each vertex or edge of a graph or specified graph is computationally necessary (if the analyzed network size is increasing). Therefore, the need to techniques for reaching rapidly to betweenness centrality has been raised, and newly, it becomes one of the specific research subjects for analyzing social networks.

Secondly, the hypothesis of information transfer in social networks in the shortest paths might be incorrect [2]. To say this, many informative diffusion models have been presented in which encoded information is generated as some messages in a source vertex and they are managed toward an objective vertex in a network that they may flow along mandatory paths. In such model, Newman [8], Nuh and Reiger [9] proposed performing random paths on social networks to compute centrality values.

A prominent approach following this research is the proposed method in [10]. In this work, a new criterion so called \( k \)-path centrality for the vertex centrality. More accurately, using particular random paths of length \( k \) were suggested to calculate centrality values [11]. In these pieces of research, an approximate algorithm of the order \( O(k^3n^{2-2\alpha} \log n) \) was introduced, where \( n \) is the number of vertices and \( \alpha \in [-\frac{1}{2}, \frac{1}{2}] \).

In another work, the edge centrality of [10] was extended by introducing a new criterion. This criterion is called \( k \)-path edge centrality. In this method, edge centrality computation is treated as a problem of information diffusion. More thoroughly, firstly it was assumed that some messages have been created and diffused in a social network, and an edge has been considered as the centrality which was often used for transferring the information. Due to this idea, message diffusion via random paths was simulated on social network’s graphs. Additionally, it was presumed that random paths are simple and are of a limited length defined by the user up to a fixed value \( k \) [12].

The first hypothesis is because of the fact that a random path should not traverse an edge more than one time; the second one is for that as mentioned in [13], the more distance between two vertices is, the less influence they have on each other [12].

According to the mentioned ideas, in this research it was presumed if heat diffusion phenomenon is used instead of information transferring, it is possible to assume that those edges and vertices that are mostly along the diffusion path absorb more heat. If each vertex is the representative of heat diffusion one time and heat process continues during the interval of length \( K \), the previous hypothesis can be generalized by it. In the current hypothesis, it was supposed that the more distance is between two vertices, the less influence they have on each other. The main goals of this paper are followings:

1. A new method was presented both for computing edge centrality and vertex centrality. with a profound look, it is observed that so many methods have been proposed to compute the vertex centrality, but a few studies have been conducted to compute edge centrality comparatively from which the edge betweenness centrality introduced in Girvan-Newman algorithm [6] is mentioned here. In addition, Newman [8], Noh and Reiger [9], Brandes [14] successfully utilized random paths to compute vertex betweenness in networks. In this paper, development of these ideas to the vertices and edges centrality is obtained.

2. It was referred to the message transfer algorithm along random paths in social networks.

3. Heat diffusion phenomenon was used instead of message transfer which it multiplied the answers diversity and increased the precision. (it was the first attempt in computing edge centrality by means of heat diffusion algorithm)

4. This method was compared with centrality extraction method via random paths of length \( K \).

5. Real online social networks were used for analyzing this method. The size of the applied social networks proves the efficacy of the proposed method to large bases.

Social networks of real life are often composed of more than millions vertices and edges [15] and hence, they might not scale well. In contrast, the proposed
algorithm performs well on huge social networks of real life even in the presence of limited computational sources. In this paper, the results of conducted experiments are presented which show that the proposed method is capable of generating reproducible results even if it is dependent on selecting random vertices. In this study, basic concepts of analysis criteria of social networks and some definitions of heat diffusion algorithm are given in the first section. In this section, a literature review of conducted works on betweenness extraction is presented. In the next section, the proposed method and applied algorithms are described. In the third section, the conducted experiment and how the proposed method has been appraised are discussed. Finally, an example of applications of betweenness and important vertices extraction are introduced.

2. Definitions and literature review

2.1. Definitions

In this subsection, two types of social networks analysis, that is centrality and prestige are introduced which closely relate to hyperlink and web search [11]. Centrality and prestige are both criteria of prominence degree of a user in social network. Also, other used concepts, like diffusion model om graph (heat diffusion) are discussed.

2.1.1. Centrality

Important or prominent users are those who are vastly relate with others. In the concept of an organization, an individual who has more extended relations with others is more remarkable than the one with more limited relations. These links are called ties. A central user is the one who involves in most of ties. Figure 1 depicts a simple example by means of an undirected graph. Each vertex is a user in a social network and each link states that the users at two ends of the link correlate. It is directly seen that the user \( i \) is the most central and the most important user since it can be in touch with the most other users.

There are different types of links between users. Therefore, various centralities are definable on directed and undirected graphs. Three famous types will be discussed in following.

2.1.1.1. Degree centrality

Central users are the most active users who have the most links with others. Let the total number of users be \( n \). This amount is between zero and one as the maximum of \( d(i) \) is \( n - 1 \).

\[
C_D(i) = \frac{d_0(i)}{n - 1}
\]

(1)

2.1.1.2. Closeness centrality

\[
C_C(i) = \frac{(n - 1)}{\sum_{j=1}^{n-1} d(i, j)}
\]

(2)

which is an amount between zero and one because the minimum amount of denominator is \( n - 1 \) which is a summation of the shortest distances between user \( i \) and all other graphs. It should be considered that this equation is just meaningful for a connected graph. Directed graph: this equation is applicable for directed graph. The direction of links or edges should be accounted for calculating the distance.

2.1.1.3. Betweenness centrality

If two non-adjacent users \( i \) and \( k \) want to have relation with each other and user \( i \) is on the path between users \( j \) and \( k \), user \( i \) controls their relations. Betweenness calculates user \( i \)’s control on other pairs of users. Thereafter, if user \( i \) is on most of the paths of these relation, it is an important user.

Undirected graph: \( p_{jk} \) is the number of the shortest paths between users \( j \) and \( k \), user \( i \)’s betweenness is the number of the shortest paths passing \( i \), which is normalized by the total number of the shortest paths of all pairs not including the vertex \( i \):

\[
C_B(i) = \sum_{j<k} \frac{p_{jk}(i)}{p_{jk}}
\]

(3)

It is likely that there are multiple short paths between users \( j \) and \( k \); some of them pass through \( i \) and some do not. Here it is assumed that all the paths are ap-
In a network of Fig. 2, user 1 is the most central user. This vertex is located on 15 short paths linking 6 other users. $C_B(1)$ has the maximum value 15.

If it is desired to have this value between zero and one, it is possible to normalize it by $\frac{(n-1)(n-2)}{2}$ which is the maximum amount of $C_B(i)$. Standardized betweenness of user $i$ is defined as:

$$C'_B(i) = \frac{2 \sum_{j<k} p_{jk}(i)}{(n-1)(n-2)} \quad (4)$$

Unlike the closeness criterion, betweenness may be computable even if the graph is not connected.

Directed graph: the above equation can be used in directed graph too, but it should be multiplied by 2 because there are $(n-1)(n-2)$ different pairs of users which means that a path from $j$ to $k$ is different from a path from $k$ to $j$. Additionally, $p_{jk}$ should take into account some paths of two directions.

### 2.1.2. Prestige

Prestige is an evaluation criterion of a user importance that is due to later definitions is more enhanced than centrality. Input and output vertices are distinguished in this criterion. A prestigious user is the one which is extensively in receiver ties. To put it in another word, just conducted or mentioned vertices or ties (links) to the user are studied. Therefore, the prestige cannot be computed unless the relation or the graph is undirected. The main difference between the centrality and prestige concepts is that the former is concentrated on output link and the latter on input link.

Three criteria were defined to measure the prestige. The third one (degree prestige) is the basis of so many analyzing algorithms of website link, like PageRank and HITS.

#### 2.1.2.1. Degree prestige

Due to the prestige definition, it is clear that a user is prestigious if it receives a great amount of input links. Hence, the simplest degree criterion of user $i$ is its degree.

$$P_D(i) = \frac{d_I(i)}{n-1} \quad (5)$$

where, $d_I(i)$ is $i$’s degree (the number of input links to $i$) and $n$ is the total number of users in the network. The same as centrality degree, the degree is normalized by dividing $n-1$ getting an amount between zero and one. The maximum amount is one when each user is in touch with user $i$ or selects it.

#### 2.1.2.2. Proximity prestige

Degree prestige of user $i$ considers just the users adjacent to user $i$. proximity prestige take into accounts both groups of users directly and indirectly being in contact with $i$. That is, every user $j$ is considered which is connected to $i$, in other word, there is a directed path from $j$ to $i$.

$I_i$ is assumed as the set of users that can reach to user $I$, which is called influential domain of user $i$. proximity is defined as the closeness of or distance between other users and user $i$.

$$d(j,i)$$

is the shortest distance between user $j$ and $i$; each link has a unit distance. To calculate the proximity prestige, distance average is used which is as following:

$$\frac{\sum_{j \in I_i} d(j,i)}{|I_i|} \quad (6)$$

where, $|I_i|$ is the cardinality of $I_i$. If the number of users that can reach $i$ are compared with the average distance between them and $i$, proximity prestige is obtained, which has an amount in [0,1]:

$$P_p(i) = \frac{|I_i|}{n-1} \sum_{j \in I_i} d(j,i) / |I_i| \quad (7)$$

where, $\frac{n-1}{|I_i|}$ is the ratio of users who can reach $i$. In a neighborhood, each user is able to find $i$, so $\frac{n-1}{|I_i|} = 1$. The denominator is one if each user is adjacent to $I$, so, $P_p(i) = 1$. In another neighborhood, no user can reach $i$. Thereafter, $|I_i| = 0$ and $P_p(i) = 0$. 

![Fig. 2. An example of a network showing betweenness centrality.](image-url)
2.1.2.3. Rank prestige

Two aforementioned prestigious are based on input degrees and distances. Anyway, a considerable factor that has not been considered is the importance of special users; which votes or selects. In real world, the individual chosen by an important one is more prestigious than the person selected by an unimportant one. To make it clear, a CEO firm’s poll for someone is much more important than a worker’s poll for that person. If the impact domain of a user is prestigious, its prestige is also so high. As a result, a user’s prestige is influenced by involved users’ ranks or positions. Based on this, the rank prestige \( P_R (i) \) is defined as a linear combination of links referring to \( i \):

\[
P_R (i) = A_{ii} P_R (1) + A_{i1} P_R (2) + \ldots + A_{in} P_R (n) \tag{8}
\]

Equation (8) says that rank prestige of a user is a function of users’ prestige’s values:

If \( j \) mentions \( i \), \( A_{ji} = 1 \), else, \( A_{ji} = 0 \). Equation (8) influences users’ prestige by involved users’ ranks or positions.

A column vector \( P \) is a vector composed of all rank prestige’s values:

\[
P = (P_R (1), P_R (2), ..., P_R (n))^T
\]

Here, \( T \) denotes the transpose of a vector or a matrix. Matrix \( A \) (where \( A_{ij} = 1 \) if \( i \) refers to \( j \), otherwise, \( A_{ij} = 0 \)) represents matrix of adjacency or graph. Hence,

\[
P = A^T P \tag{9}
\]

This equation is exactly a specific applied one to find eigenvector of matrix \( A^T \) and \( P \) is the eigen vector of matrix \( A^T \). This equation and the idea behind it is so beneficial for web search. In fact, the most famous ranking algorithms for web search are PageRank and HITS which are directly connected to this equation.

2.2. Heat diffusion

Heat diffusion is a physical phenomenon. Generally, heat moves from a position of higher temperature to a lower temperature position. Recently, heat diffusion based methods have been applied in such vast domains as dimension classification and dimension reduction \[16\]-[19]. Heat approximated the nucleus in a closed form for a multiple family which had more improvement than Gaussian method and linear nuclei \[20\]. Kondor proposed a separate heat diffusion nucleus for classification and showed that a simple nucleus diffusion on a polyhedral had an acceptable efficiency on this kind of data \[17\].

On the other hand, M. Belkin et al. used heat nucleus to weight a neighborhood graph and applied it in a dimension reduction algorithm \[16\]. Yang et al. by using heat diffusion suggested a ranking algorithm called Rank Diffusion \[21\]. The simulation states that this method is highly effective in recognizing spam.

2.2.1. Heat diffusion on the graph

Consider the undirected graph \( G = (V, E) \) where typically \( V = \{v_1, v_2, \ldots, v_n\} \) is the set of graph vertices and also \( E = \{(v_i, v_j) \mid \text{there is an edge between } v_i \text{ to } v_j \} \) is the set of edges. We suppose edge \((v_i, v_j)\) as a pipe connecting \( v_i \) and \( v_j \) vertexes. The value \( f_i (t) \) shows the amounts of heat at vertex \( v_i \) at time \( t \), beginning from an initial distribution of heat given by \( f_i (0) \) at time \( 0 \). \( f_i (t) \) represents a vector that its components are \( f_i (t) \). The mathematic model of this algorithm is based as: suppose at the time \( t \), the vertex \( i \) receives the amounts \( M (i, j, t, \Delta t) \) heat from its neighbor vertex \( j \) during a time period \( \Delta t \). The amounts of exchanged heat should be related to \( \Delta t \) and also related to the difference of \( |f_i (t) - f_j (t)| \). Moreover, the heat transfers through a pipe connecting these vertexes. So we can say:

\[
M (i, j, t, \Delta t) = \alpha (f_j (t) - f_i (t)) \Delta t \tag{10a}
\]

\( \alpha \) is the thermal conductivity of the heat diffusion coefficient. As a result, the difference of amount of heat on the vertex \( i \) at the times \( t \) and \( t + \Delta t \), is equal to some of the heat that this vertex receives from its neighbor vertexes. Mathematically speaking, we have:

\[
\frac{f_i (t + \Delta t) - f_i (t)}{\Delta t} = \alpha \sum_{j,(v_i, v_j) \in E} (f_j (t) - f_i (t)) \tag{10b}
\]

\( E \) is the graph edges set.

We rewrite the above equation in a matrix form to find out a solution close to it.

\[
F (t + \Delta t) - F (t) = \alpha (H - D) F (t) \tag{11}
\]
Where:

\[ H_{ij} = \begin{cases} 
1, & (v_i, v_j) \in E \text{ or } (v_j, v_i) \in E \\
0, & i = j \\
0, & \text{otherwise}
\end{cases} \] (12)

and

\[ D_{ij} = \begin{cases} 
(v_i), & i = j \\
0, & \text{otherwise}
\end{cases} \] (13)

In which \( d(v_i) \) is the degree of the vertex \( v_i \). From the definition, it is clear that the \( D \) is a Diagonal matrix. We normalize all the elements of the matrices \( H \) and \( D \) by the degrees of all vertices to have a more general representation. So the \( H, D \) matrices got improved as follows:

\[ H_{ij} = \begin{cases} 
\frac{1}{d(v_i)} (v_i, v_j) \in E \\
0, & i = j \\
0, & \text{otherwise}
\end{cases} \] (14)

and

\[ D_{ij} = \begin{cases} 
1, & i = j \\
0, & \text{otherwise}
\end{cases} \] (15)

Now, while \( \Delta t \) tends to zero, Eq. (11) can be rewrite as the following:

\[ \frac{d}{dt} F(t) = \alpha (H - D) F(t) \] (16)

And we got to this answer when solving this differential equation:

\[ F(1) = e^{\alpha (H - D)} F(0) \] (17)

In which \( d(v) \) is the degree of vertex \( v \). Using the following expansion we can calculate the amounts of \( e^{\alpha (H - D)} \) to the arbitrary precision.

\[ e^{\alpha (H - D)} = I + \alpha (H - D) + \frac{\alpha^2}{2!} (H - D)^2 + \frac{\alpha^3}{3!} (H - D)^3 + \frac{\alpha^4}{4!} (H - D)^4 + \ldots \] (18)

The \( e^{\alpha (H - D)} \) matrix is called transfer kernel because the heat diffusion process repeats endlessly from the initial step. This problem is very important to assign vertexes in a graph. There are complete explanations related to heat diffusion on an undirected graph and its related algorithm time complexity in \[22\].

An example of heat diffusion is obtained below. In order to interpret Eq. (7) and the heat diffusion process more intuitively, we construct a small undirected graph with only five nodes as showed in Fig. 1(a).

Initially, at time zero, suppose node 1 is given 3 units of heat, and node 2 is given 2 units of heat; then the vector \( f(0) \) equals \([3, 2, 0, 0, 0]^T\). The entries in matrix \( H-D \) are as this:

\[
\begin{bmatrix}
-4 & 1 & 1 & 1 & 1 \\
1 & -1 & 0 & 0 & 0 \\
1 & 0 & -1 & 0 & 0 \\
1 & 0 & 0 & -1 & 0 \\
1 & 0 & 0 & 0 & -1
\end{bmatrix}
\]

Without loss of generality, we set the thermal conductivity \( U \), \( = 1 \), and vary time \( t \) from 0 to 1 with a step of 0.05. The curve for the amount of heat at each node with time is shown in Fig. 5(b). We can see that, as time passes, the heat sources node 1 and node 2 will diffuse their heat to nodes 3, 4, and 5.

The heat of nodes 3, 4, and 5 will increase respectively, and the trends of their heat curves are the same since these three nodes are symmetric in this graph.

Another example is shown in Fig. 3(c). Initially, at time zero, suppose node 1 is given 4 units of heat; then the vector \( f(0) \) equals \([4, 0, 0, 0]^T\). The related heat curve is shown in Fig. 3(b). We can see that the node 2, the nearest node to the heat source, gains more heat than other nodes. This also indicates that if a node has more paths connected to the heat source, it will potentially obtain more heat. This is a perfect property for recommending relevant nodes on a graph.

2.3. Related works

2.3.1. Background of betweenness evaluation criteria and applications

In this section, a concept of centrality criterion is studied and some of recent methods of calculating it are given.

Centrality evaluation criterion in social networks: one of the most primary criteria of vertex centrality as it was defined is betweenness centrality \[4\].

Betweenness centrality: according to graph \( G = (V, E) \), betweenness for vertex \( v \in V \) is defined as:

\[ C_{Bv}(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \] (19)
where, $s$ and $t$ are vertices in $V$, $\sigma_{st}$ is the number of shortest paths connecting $s$ to $t$ and $\sigma_{sv}(v)$ is the number of the shortest paths connecting $s$ to $t$ passing vertex $v$, if there is no path connecting $s$ to $t$.

A concept of centrality for edges of a graph is also defined. By a glimpse of the past, the first method of computing the vertex centrality was proposed by Antonisse in 1971 [5,23] and was executed in GRADAP software package. In this method, the edge centrality is interpreted as a flow centrality criterion. Considering a graph $G = (V, E)$, $s \in V$, $t \in V$ is a fixed pair of vertices. One unit of flow is injected to the network by selecting $s$ as a source vertex and this flow is along the shortest paths. rush index, $\delta_{st}(e)$, related to the pair $(s, t)$ and the edge $e \in E$ is defined as:

$$\delta_{st}(e) = \frac{\sigma_{st}(e)}{\sigma_{st}}$$  \hspace{1cm} (20)

That similar as before, $\sigma_{st}$ is the number of the shortest paths connecting $s$ to $t$ and $\sigma_{st}(e)$ is the number of the shortest paths connecting $s$ to $t$ passing edge $e$. similarly, if there is no path connecting $s$ to $t$, $\sigma_{st}(e) = 0$.

Girvan and Newman proposed a definition of edge betweenness centrality [6] in 2002, which strongly resembles to that of Antonisse. According to what mentioned, edge betweenness centrality for edge $e \in E$ is stated as:

$$C_{B}(e) = \sum_{s \neq t \in V} \frac{\sigma_{st}(e)}{\sigma_{st}}$$ \hspace{1cm} (21)

This definition is different from the one of Antonisse because the vertex of source $s$ and the vertex of goal $t$ have to be different. Some other different definitions of betweenness centrality were proposed by Brands [24], such as bounded-distance, distance-scaled, edge and group betweenness and stress and load centrality.

Although appropriateness of betweenness centrality is observable in expressing the importance of a vertex or an edge, it is not always a specific solution for a given problem. For instance, as Stephanson and Zalen mentioned [7], the first concept of betweenness centrality is due to the fact that information diffusion is not done only in the shortest paths. According to the impact of diffusion, it is clear that the more distance between to vertices is, the less they have influence on each other [14]. Moreover, in real applications, precise ranking computation is not usually needed in a network with respect to betweenness centrality of each vertex or edge. As a matter of fact, it leads to more useful recognition of arbitrary percentage of vertices/edges which are more connected to a specific given problem (that is information diffusion, identifying main factors).

2.3.2. Recent methods for calculating betweenness centrality

So far, multiple algorithms have been proposed to compute betweenness centrality. The most influential
one were suggested by Brands [25] which are performed on the order \( O(nm) \) for non-weighted graphs and on the order of \( O(nm + n^2 \log n) \) for the weighted graphs of \( n \) vertices and \( m \) edges.

Computational complexity of these methods makes them impossible to analyze large networks. For this aim, various approximative solutions were suggested. Among the others, Brands [26][27] and Pich [26] produced a random algorithm, called RA-Brands and similarly, Bader et al. proposed another approximative model, called AS-Bader by means of obtained techniques [28]. Newman introduced a random path based algorithm to compute betweenness centrality, which it has some similarities with the proposed method, that it uses a concept of message diffusion along the random paths. Using this idea, Alahakoon et al. [10] proposed \( k \)-path centrality evaluation criterion and generated an algorithm of the order \( O(k^3 n^2 \alpha \log n) \), called RA-Kpath, to compute it.

2.3.3. Application of centrality evaluation criteria in analyzing social networks

Applications of obtained centrality information from social networks were studied by Staab et al. [29].

Researchers defined different methods to study discovered data of marketing goals, confidence theory and analysis and etc. Multiple studies on marketing and business were applied for online social networks (NOSS), specially in order to find out effective channels to distribute information [30,31] and to study the effect development [32]. Current study can potentially give useful information for all these research perspectives, can recognize remarkable edges of edge centrality higher than \( k \) paths for which it insists on their importance in the social network. Those connected vertices to high central edges are important for topological condition. Additionally, they are able to effectively transfer information to their adjacency.

3. Proposed method

3.1. Researching objectives

Before describing the proposed algorithm, the main ideas behind it are determined. This paper was inspired from message diffusion in a network graph and finding centrality edges including random paths of length \( K \); consequently, this method will be discussed in details, first, the proposed method will be generalized.

3.2. An algorithm for computing edge centrality of length \( k \)

In this subsection, an algorithm called random walk \( k \)-path (ERW-Kpath) is discussed to effectively calculate edge centrality values. This method consists of two steps:

1. Allocating some weights to vertices or edges.
2. Simulation message diffusion via random simple paths.

In ERW-Kpath, the probability of choosing a vertex/an edge is homogenous; there is another type of this algorithm called weighted edge random walk \( k \)-path (WERW-Kpath) in which the probability of a vertex/an edge is not homogenous. After this, WERW-Kpath will be introduced as a generalization of ERW-Kpath algorithm.

3.2.1. Step 1: Weight allocation to vertex and edge

In the first step, some weights are designate to the vertices and edges of graph \( G = (V, E) \) showing a social network. The weights on the vertices are used to select source vertices from which the simulation starts. The weights on the edges state the initial values of edge centrality which will be updated during algorithm performance.

To calculate the vertices weights, a normalized degree \( \delta(v_n) \) of a vertex \( v_n \in V \) is defined as:

**Definition 1.** According to an undirected graph and a vertex the normalized degree is:

\[
\delta(v_n) = \frac{|I(v_n)|}{|V|} \quad (22)
\]

where, \( I(v_n) \) is the set of edges passing \( V_n \).

Normalized degree \( I^*(v_n) \) relates to \( V_n \)'s degree and total number of vertices. It implicitly shows that how many vertices help to all connections of a graph. Its value is in \([0,1]\) and the higher \( I^*(v_n) \) is, the better \( V_n \) is connected to the graph.

Following definitions are given for an edge weight.

**Definition 2.** According to an undirected graph \( G = (V, E) \) and an edge \( e_m \in E \), initial edge weight \( w_0(e_m) \) is:

\[
w_0(e_m) = \frac{1}{|E|} \quad (23)
\]

Indirectly, this equation explains that a set of \( E \) points is controlled first; these points are equally distributed among all possible edges; the value of received point by an edge shows its initial rank.

In Fig. 3(a) an example of Graph \( G \) along with distributed weights on vertices and edges.
3.2.2. Step 2: Simulating message diffusion via random simple path of length \( k \)

In the second step, random multiple paths on graph \( G \) are simulated.

For this goal, following algorithm executes some procedures \( p \) times. At each iteration, there are some steps:

1. A vertex \( v_n \in V \) is chosen with respect to one of two following strategies:
   - With a random homogenous probability:
     \[
     p(v_n) = \frac{1}{|V|}
     \]
   - With a probability proportional to its normalized degree \( I'(v_n) \) defined as this:
     \[
     p(v_n) = \frac{\delta(v_n)}{\sum_{v_k \in V} \delta(v_k)}
     \]

2. All edges in \( G \) are marked as unpassed.
3. Message diffusion method is performed and it creates a simple random path of a length not more than \( k \).

Here, message diffusion method is described. A loop is done if following conditions are satisfied:

1. Recent created path length is not more than \( k \), which is check by a counter of length \( n \).
2. To do this, a tag \( T(e_m) \) is attached to each edge \( e_m \in E \) such that:
   \[
   T(e_m) = \begin{cases} 
   1 & \text{if } (e_m) \text{ has already been traversed} \\
   0 & \text{otherwise} 
   \end{cases}
   \]

Following conditions should be satisfied:
\[
|I(v_n)| > \sum_{e_k \in I(v_n)} T(e_k)
\]

where, \( I(v_n) \) is a set of edges passing \( v_n \). If above conditions are satisfied, diffusion method selects an edge \( e_m \) by means of two strategies:

- With a random homogenous probability:
  \[
  P(e_m) = \frac{1}{|I(v_n)| - \sum_{e_k \in I(v_n)} T(e_k)}
  \]

All edges belonging to \( \{I(v_n) \mid T(e_m) = 0\} \) pass \( v_n \).

- With a probability proportional to edge weight \( w_1(e_m) \) is shown as:
  \[
  P(e_m) = \frac{w_1(e_m)}{\sum_{e_m \in I(v_n)} w_1(e_m)}
  \]

If we have in all steps:
\[
|I(v_n)| > |\{e_k \in I(v_n) \mid T(e_k) = 0\}|
\]

and
\[
w_1(e_m) = w_{l-1}(e_m) + \beta \cdot T(e_m)
\]

such that \( 1 \leq l \leq k_p \).

Selected edge \( e_m \) and vertex \( v_{n+1} \) connected to \( v_n \) by \( e_m \) are considered. Message diffusion method gives a score \( \beta \) to \( e_m \), \( T(e_m) = 1 \) and counter \( n \) increases 1 unit. Message diffusion continues from \( v_{n+1} \).

3.3. Proposed method

This problem is assumed based on heat diffusion in real world, where heat transfers from the location of higher temperature to the point of lower temperature. Hence, in this method, when vertices of a graph are heated, heat transfers to adjacent vertices via edges connected to it.

It could suppose that if each vertex of a network becomes the heat diffusion source one time, those vertices on the diffusion path of the majority of vertices receive more heat. Thereafter, in the proposed method vertices weight is their temperature during heat process; and edges weight also increases by each time of heat diffusion one unit.

Finally, vertices of higher temperature can be considered as centrality vertices. In next section, a thorough detail of the proposed algorithm is discussed.

The proposed system called Heat Diffusion Betweenness Centrality (HDBC) consists of three phase:

1. Creating a social network graph.
2. Heat diffusion process.
3. Determining centrality.
Table 1

<table>
<thead>
<tr>
<th>#</th>
<th>Network</th>
<th>Number of nodes</th>
<th>Number of edges</th>
<th>Directed</th>
<th>Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Wiki-Vote</td>
<td>7,115</td>
<td>103689</td>
<td>Yes</td>
<td>Elections</td>
<td>[33]</td>
</tr>
<tr>
<td>2</td>
<td>CA-HeapPh</td>
<td>12008</td>
<td>237010</td>
<td>No</td>
<td>Co-authors</td>
<td>[33]</td>
</tr>
<tr>
<td>3</td>
<td>CA-CondMat</td>
<td>23133</td>
<td>186932</td>
<td>No</td>
<td>Co-authors</td>
<td>[33]</td>
</tr>
<tr>
<td>4</td>
<td>Cit-HepTh</td>
<td>27770</td>
<td>352807</td>
<td>Yes</td>
<td>Citations</td>
<td>[33]</td>
</tr>
<tr>
<td>5</td>
<td>Facebook</td>
<td>63731</td>
<td>1545684</td>
<td>Yes</td>
<td>Online SN</td>
<td>[34]</td>
</tr>
<tr>
<td>6</td>
<td>Youtube</td>
<td>1138499</td>
<td>4945382</td>
<td>No</td>
<td>Online SN</td>
<td>[33]</td>
</tr>
</tbody>
</table>

3.3.1. Creating social network graph

For extracting centrality it is required that a graph on which operations should be done to be created. In this study, well-known social networks listed in Table 1 were used. First of all, vertices numbers connecting to each other were extracted from a text file and adjacency matrix was generated.

\[ g_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ is connected to vertex } j, \\ 0 & \text{otherwise} \end{cases} \]

This matrix is a sparse matrix for real social networks. Hence, for saving memory and time, sparse matrices structures were utilized.

3.3.2. Heat diffusion process

After creating a social network graph (matrix \( G \)), all vertices initial temperatures are set zero. A vector \( f_0 \) (defined in Section 2 in heat diffusion) composed of initial temperatures of vertices is defined and set zero. Then, one of the vertices \( (i) \) is randomly selected and a tag is given to it; that is, \( T(i) = 1 \) (here, the probability of selecting vertices is the same) which states that this vertex has diffused heat once. Next, a graph \( G' \) is defined equivalent to the first graph. Initial value of the selected vertex is given 1; i.e. \( f_0(i) = 1 \). After that, heat diffusion algorithm introduced in Section 2 is performed on equivalent graph \( G' \) and resulted temperatures are added to main graph \( G \) after some time of order \( k \).

This procedure continues until no representative vertex of heat diffusion remains.

Then the heat diffusion algorithm is performed on the graph \( G' \), and the generated amounts of heat are added to the corresponding vertices of \( G \).

This process is iterated for all vertices of \( G \).

3.3.3. Determining centrality

Vertices and edges having the highest heat value are selected. Then, to obtain edge centrality, the temperature of each edge is set to the sum of temperatures of two vertices composing it. Therefore, centrality edge is also equal to edges having higher temperature than the others. It means that centrality edge is equal to the edges of the highest temperatures. Details of this algorithm are given in Algorithms 1 and 2.

3.4. Algorithm

Algorithm 1: Determiner centrality algorithm

**Random path** (graph \( G = (V, E) \) and vector \( \tau \))

1. Cluster \( 1, \ldots, n \) to the vertices of \( G \). (\( n = |V| \))
2. For \( i = 1 \) to \( n \) do the following steps.
3. Create the new graph \( G_i \) as this. The vertices and edges of \( G_i \) are the same as \( G \). allocate 1 ammount of heat to the vertex \( i \), and zero to others.
4. The heat diffusion algorithm is performed for the graph \( G_i \).
5. The generated amounts of heat are added to the corresponding vertices of \( G \). 
6. The loop is finished.
7. Allocate to each edge the sum of heats of its two corresponding vertices.
8. Finally, the vertices and the edges with highest ammount of heat have more centrality.

Algorithm 2: Heat diffusion betweenness centrality algorithm HDBC

**Heat diffusion process** \( HDBC(G_i, v_i) \)

1. Consider the graph \( G_i \) as defined in algorithm 1.
2. Consider the initial heat vector as this: \( F(0) = (f_1(0), \ldots, f_n(0)) \)
3. The heat diffusion process is performed for the graph \( G_i \).
   \[ F(1) = e^{\alpha(H-D)}F(0) \]
4. The vector \( F(1) \) is the final result of the heat diffusion process.

4. Evaluation

4.1. Experiments

Several experiments were done on online social networks whose datasets are available. Taken datasets
were summarized in Table 1. Dataset 1 shows Wiki Pedia’s polling system for January 2008’s election. Datasets 2 and 3 show Arvix archives (an online archive for scientific preprints in the fields of mathematics, Physics and Computer Science, amongst others) of papers in the domain of high energy physics (phenomenology) and condensed materials physics. Dataset 4 represents a network of scientific citations among the papers of Arvix’s high energy physics. Dataset 5 describes a small sample of Facebook social network which shows its friendship graph. At last, Dataset 6 has a part of Youtube social network graph since 2007.

4.2. Evaluation

To evaluate this algorithm, it should be proved that output dataset are correlated at each round of performing algorithm. As it is seen, the output of algorithm is different because of its random structure. Therefore, following procedure was done to assess the algorithm:

4.2.1. Jaccard coefficient

The first evaluation criterion is a type of Jaccard coefficient defined as:

\[
J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}
\]

(29)

where, \(x\) and \(y\) stating a pair of compared distributions of edge centrality values of the proposed algorithm. In order to determine Jaccard coefficient, below problems are required to be taken into account. Two performances of the algorithm, \(x\) and \(y\), and an edge \(e\) as \(w_X(e)\) (e centrality index in \(x\)) are assumed. Implicitly, algorithm functionality is acceptable if \(w_X(e)\) is close to \(w_Y(e)\). Nevertheless, a direct comparison of these two values cannot show a result because for instance, edge \(e\) could have the maximum weight in both performances, but \(w_X(e)\) might be considerably different from \(w_Y(e)\). Hence, normalized values \(\frac{w_X(e)}{\max_{e \in X} w(e)}\) and \(\frac{w_Y(e)}{\max_{e \in Y} w(e)}\) have to be considered and be supposed that good results are obtained whenever these values are close to each other.

To make it more precisely \(\hat{I}(e)\) is defined:

\[
\hat{I}(e) = \frac{w_X(e)}{\max_{e \in X} w(e)} - \frac{w_Y(e)}{\max_{e \in Y} w(e)}
\]

It says that if \(\hat{I}(e)\) is less than a threshold \(\hat{I}_\mu\), the algorithm gives good results. In order to fix \(\hat{I}_\mu\), resulted amounts of \(\hat{I}(e)\) are considered for each \(e \in E\).

An upper bound \(\Lambda\) is introduced for \(\Lambda(e)\) by considering

\[
\text{Table 2: comparison of two methods } k\text{-path and HDBC with Euclidean distance}
\]

<table>
<thead>
<tr>
<th>Data set</th>
<th>(L_2(x, y)) k-path</th>
<th>(L_2(x, y)) HDBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiki-Vote</td>
<td>0.0161</td>
<td>0.002</td>
</tr>
<tr>
<td>CA-HepPh</td>
<td>0.0118</td>
<td>0.0012</td>
</tr>
<tr>
<td>CA-CondMat</td>
<td>0.0139</td>
<td>0.0095</td>
</tr>
<tr>
<td>Cit-HepTh</td>
<td>0.092</td>
<td>0.07</td>
</tr>
<tr>
<td>Facebook</td>
<td>0.0101</td>
<td>0.0012</td>
</tr>
<tr>
<td>Youtube</td>
<td>0.0131</td>
<td>0.0003</td>
</tr>
</tbody>
</table>

Fig. 5. comparison of two methods \(k\)-path and HDBC with Euclidean distance. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/KES-150306)

1. \(\langle e \rangle = \max_{e \in X} w(e)\) and \(w_Y(e) = \min_{e \in Y} w(e)\)

or

2. \(\langle e \rangle = \min_{e \in X} w(e)\) and \(w_Y(e) = \max_{e \in Y} w(e)\).

Without loss of generality, if case 1 happens, followings are also true for case 2. Consequently,

\[
\Lambda = 1 - \frac{\min_{e \in Y} w(e)}{\max_{e \in Y} w(e)}
\]

Due to what will be discussed, edge centralities are distributed with respect to exponential rule and \(\min_{e \in Y} w(e)\) is somehow smaller than \(\max_{e \in Y} w(e)\).

Therefore, ratio of \(\min_{e \in Y} w(e)\) to \(\max_{e \in Y} w(e)\) is zero and \(\Lambda\) is one. According to these, it is calculated how many times \(\hat{I}(e) \leq \tau \Lambda\) is satisfied, where, \(0 < \tau \leq 1\) is a tolerance threshold. Since \(\Lambda \approx 1\), these values specify that how many times \(\hat{I}(e) \leq \tau\). As a result, modified Jaccard coefficient could be defined as:

\[
J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}
\]

(30)

Following values of tolerance threshold were considered from experiments to recognize 1, 5, 10 percent of the maximum accepted modification of edge centrality value allocated to a specific edge during different performances with similar configurations.
Table 3

Comparison of two methods k-path and HDBC with Pearson correlation

<table>
<thead>
<tr>
<th>Data set</th>
<th>( \rho(x,y) )-k-path</th>
<th>( \rho(x,y) )-HDBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiki-vote</td>
<td>0.70</td>
<td>0.999</td>
</tr>
<tr>
<td>CA-hepPh</td>
<td>0.96</td>
<td>0.999</td>
</tr>
<tr>
<td>CA-CondMat</td>
<td>0.85</td>
<td>0.999</td>
</tr>
<tr>
<td>Cit-HepTh</td>
<td>0.84</td>
<td>0.93</td>
</tr>
<tr>
<td>Youtube</td>
<td>0.89</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Table 4

Comparison of two methods k-path and HDBC with Jacard coefficient

<table>
<thead>
<tr>
<th>Data set</th>
<th>( J(x,y) )-k-path</th>
<th>( J(x,y) )-HDBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiki-vote</td>
<td>( \tau ) 0.01</td>
<td>0.05</td>
</tr>
<tr>
<td>CA-hepPh</td>
<td>70.68</td>
<td>99.96</td>
</tr>
<tr>
<td>CA-CondMat</td>
<td>75.65</td>
<td>99.51</td>
</tr>
<tr>
<td>Cit-HepTh</td>
<td>35.63</td>
<td>95.80</td>
</tr>
<tr>
<td>Facebook</td>
<td>63.68</td>
<td>99.62</td>
</tr>
<tr>
<td>Youtube</td>
<td>27.92</td>
<td>82.29</td>
</tr>
</tbody>
</table>

4.2.2. Pearson correlation

The second evaluation criterion is Pearson correlation which is for assessing the relation of two distributions and defined as:

\[
\rho_{X,Y} = \frac{cov(X,Y)}{\sqrt{\text{var}(X) \cdot \text{var}(Y)}}
\]  

(31)

The result is normalized in \([-1, 1]\) with following interpretations:

- \( \rho_{X,Y} > 0 \): Distributions are directly related, specially:
- If \( \rho_{X,Y} > 0.7 \), they are strongly correlated.
- If \( 0.3 < \rho_{X,Y} < 0.7 \), they are averagely correlated.
- If \( 0 < \rho_{X,Y} < 0.3 \), they are weakly correlated.
- If \( \rho_{X,Y} = 0 \), they are linearly unrelated.
- If \( \rho_{X,Y} < 0 \), they are inversely correlated.

Vividly, the greater \( \rho_{X,Y} \) is, the better HDBC algorithm works. \( \rho_{X,Y} \) states that whether two distribution \( x \) and \( y \) are absolutely correlated or not. So, if \( Y = aX \) such that \( a \) is a real coefficient, HDBC algorithm may create two edge centrality distributions \( x \) and \( y \) in two different performances. In this case \( \rho_{X,Y} \) could take on one, but it is allowed to conclude that this algorithm performs completely. Indeed, coefficient \( a \) may be so small and two distributions can be very different if they preserve the same edge rankings.

4.2.3. Euclidean distance

Another evaluation criterion is considered to calculate the distance between two distributions \( x \) and \( y \). Euclidean distance \( L_2(X,Y) \) is defined as:

\[
L_2(X,Y) = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}
\]  

(32)

To obtain the average distance between two points in distributions \( x \) and \( y \) in a given dataset, just \( L_2(X,Y) \) should be divided by the number of edges in this...
dataset. Internal properties of analyzed dataset do not affect on results power. In fact, even by considering dataset showing different social networks (that is, group networks, critic networks and online societies), HDBC generates overlapping results through different runs. Taking on a low tolerance threshold, such as $\tau = 0.05$ or $\tau = 0.01$, they become so similar. The same occurs for Pearson correlation coefficient which identifies strong relationships between all various distributions.

4.3. Efficiency

Illustrated results after executing the algorithm express that although it is a random algorithm, centrality vertices are shown similarly on each run. Algorithm’s evaluation results in Tables 2–4 showing the improvement of its precision to $0.35$. As it is observed, Euclidean distance difference in the proposed algorithms is nearly zero; and Jackard and Pearson correlation coefficients are equal to or close to one expressing absolute dependence of two different distributions’ data.

5. Conclusion and future developments

5.1. Conclusion

In communication age, information is generated and transferred increasingly by technology and multiple informative tools. On the other hand, human beings’ traditional behaviors are changing simultaneously with instant growth of electronics evolutions. In developed societies, using modern technology in daily life to provide usual needs of life is inevitable.

One of the most important technologies, the basis of many communicational and informational creativities and inventions, is internet. Not only information is transferred rapidly and simply via internet, but also new domains have been born to evolve traditional behaviors. One of the new consequences which grew and developed due to internet penetration in today world was the birth of social networks that allow users to save and share their information and preferences. So a great amount of valuable information in internet has led to research blossom which help internet users to more optimally use this information. Many resulted outputs of social networks can be extracted and developed for web purposes. Ideas of analyzing social networks are primarily effective for web search engines success. In these studies named social network analysis, the importance and type of internal relations of the network (relationship between users and application) are recognized and validated. An example of these results attempts to find the most important ones in the network. These studies in politics, business, medicine and etc. are specifically applicable in finding political groups and leaders.

In this paper, it was tried to study one of the most important problems of social network analysis. This problem is to finding important vertices which are able to be leaders of social networks.

By applying heat diffusion algorithm and implementing it on social network graph such that consecutively, each vertex is heat diffusion representative in the graph and diffuse heat, centrality vertices are being found by assuming that those vertices of higher temperature are on the heat path of most vertices. In this study, a method was proposed to extract both edge centrality and vertex centrality. Comparing this method with previous one shows its precision improvement.

5.2. Future developments

The new criterion that suggested in this paper, can be considered as a new view to centrality, and other researchers can extend this ideas. In fact, we use a physical phenomenon for evaluate the centrality, and heat diffusion is a new idea. Similarly, other physical phenomenon as heat, energy, velocity, . . . can be used to get new views of centrality.

References


H. Ma, I. King and M.R. Lyu, Fellow, mining web graphs for recommendations, *IEEE Transactions on Knowledge and Data Engineering* 24 (2012), 1051–1064.


D. Kempe, J. Kleinberg and E. Tardos, Maximizing the spread of influence through a social network, *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM (2003), 137–146.


