An Algorithm for Advertisement Propagation on Social Network

Jyoti Dua
Computer Science and Engineering
United Institute of Technology, Allahabad
India
jyotidua_ecimt@yahoo.com

Prashant Shukla
Computer Science and Engineering
United Institute of Technology, Allahabad
India
prashantshukla.uit@gmail.com

ABSTRACT
Centrality plays a key indicator role to identify influential node/edge on the social network. A network is considered as a graph through which information flows. This paper presents an algorithm for m-path edge centrality forming a path through random walk to identify the central edge through which information flows effectively and influences maximum number of users closest to each other on the network. As the network grows the distance between the nodes also increases so random walk can become tedious. To address this situation we have performed clustering and applied m-path edge centrality through random walk on the cluster having highest density thus reducing the time complexity.

General Terms
Social Network

Keywords
Centrality, clustering coefficient, connected component, degree distribution.

1. INTRODUCTION
Social networks are gaining popularity in attracting people of various fields which can lead generation and availability of information and valuable data on the social network. It plays a vital role in influencing people by propagating information throughout the network. The biggest challenge is to propagate information in such a way that it affect huge amount of users on social network. Various authors studied and analyzed networks on the basis of topological structure by means of degree distribution, community detection [7] [8], global and local clustering coefficient [9], influential propagation. Some focused on evolutionary aspects of social network and analyzed its working on the basis of structural and temporal dynamics [13]. Some of them considered the characteristics of the network such as link prediction. According to Freeman et al. [15] information does not flow from geodestic path that is information does not know the ideal path to flow from one user to another. To address this problem Freeman et al. proposed a betweenness measure based on the idea of maximum flow also known as flow betweenness. The worst time of the algorithm is computed as $O\ (m^3)$. The flow betweenness using augmenting path takes worst time of $O\ (m^2n)$. Still the information needed ideal path to realize maximum flow for this reason concept of shortest path was taken into consideration. Then again random walk using matrix was proposed by [12] which takes the worst time of $O\ ((m + n)^2)$. Recently, based on these concepts various developments and survey been done to diffuse information [16][19] on social network as well as to propagate information on the network.

2. TERMINOLOGY
Different important features were analyzed by authors in their research while studying the social networks. They are:

2.1 Degree Distribution
The degree of a node is the number of links associated to other nodes and the probability distribution of these degrees of the whole network is called degree distribution. The degree distribution $P(k)$ of a network is defined as a fraction of nodes in the network of degree ‘k’. Degree distribution is important feature for studying both theoretical network and real network such as social network. This feature can also be applied to a random graph in which each node ‘n’ is connected (not connected) with an independent probability $p$ (or 1-$p$) has a binomial distribution of degree $k$. Some social networks are found to have degree distribution that approximately follow power law $P(k) \sim k^{-\gamma}$ where $\gamma$ is constant [18]. Such networks are called scale-free networks [6] and so attracted the attention for their structural and temporal variations. The real networks usually have different degree distribution. In this type of networks most of the nodes have smaller degree and few nodes have lager degree. The nodes with larger degree are called hub.

2.2 Connected Component
Connected Component for both undirected graph and directed graph are identified in different ways. An undirected graph $G = (V, E)$ is considered to be connected if a path exists between all pairs of vertices thus making each of the vertices in a pair in a pair reachable from the other. The connected components includes the equivalence classes of an equivalence relation that is defined on the vertices of the graph/network. The two well-known algorithm used to find connected components are DFS and BFS [17]. Reachability is considered as a equivalence relation that is it is reflexive, symmetric and transitive. The connectedness
of a simple directed graph becomes more complex because direction must be considered. For instance, if vertex ‘a’ is reachable from vertex ‘b’, it may be possible that vertex ‘b’ is not reachable from vertex ‘a’. There are three distinct forms of connectedness in simple directed graph:

- Weakly connected graph, where the direction of the graph is ignored and the connectedness is defined as if the graph was undirected.
- Unilaterally connected graph is defined as a graph for which at least one vertex of any pair is reachable from other.
- Strongly connected graph is one in which for all pair of vertices, both vertices are reachable from the other.

### 2.3 Clustering Coefficient

Due to the rippling influence of network dynamic quantitative analysis is needed. Clustering plays an important role to group people having similar characteristics. It is an unsupervised classification. There are various quantifiable similarity measures to analyze the network. The clustering of a network identify sub-graph that some nodes in the network tends to cluster together. The clustering of a network can be performed in two ways:

- **Local clustering**: clusters can be computed one at a time based on only partial views of the graph topology.

By definition, the local clustering coefficient [1] for node vi, denoted by ci, is defined as the ratio of the number of (open triplet) or three (closed triplet) ties [2].

Formally,

\[ c_i = \frac{2l_i}{d_i(d_i-1)} \]

The network average clustering coefficient [1], denoted by c_, is defined by

\[ c_\bar{} = \frac{1}{n} \sum_{i=1}^{n} c_i \]

Fig. 4 shows average clustering coefficient.

- **Global clustering**: It is dealing with up to few millions of vertices on sparse graph. The global clustering coefficient [1], denoted by cg, is defined as the ratio of the total number of triangles to the total number of connected triplets.

Formally,

\[ c_g = \frac{2\sum_{i=1}^{n} l_i}{\sum_{i=1}^{n} d_i(d_i-1)} \]

The global clustering coefficient is concerned with then density of triplets of nodes in a network. A triplet can be defined as three nodes that are connected by either two (open triplet) or three (closed triplet) ties [2].

In a global clustering, each vertex of the input graph is assigned a cluster in the output of the method, whereas in a local clustering, the cluster assignments are only done for a certain subset of vertices, commonly only one vertex.

The average clustering coefficient of the network is the average of local clustering coefficients of all nodes in the network. The nodes in the networks can be clustered based on similarity measures such as: Euclidean distance, Jaccard Coefficient, Cosine Similarity etc. [10] [11].

### 2.4 Centrality

The identification of the influential node in the network is computed by centrality measure. The centrality computation on local topological properties based on degree of a node do not give faithful result. Due to this reason researches were made to compute centrality on whole social network called global measures. These global measures are betweenness centrality and closeness centrality.

Let \( \sigma_{st} = \sigma_{ts} \) denote the number of shortest paths from \( s \in V \) to \( t \in V \), where \( \sigma_{st} = 1 \) by convention. Let \( \sigma_{st}(v) \) denote the number of shortest paths from \( s \) to \( t \) that some \( v \in V \) lies on [3].

- **Closeness Centrality**: \( C_C(v) = \frac{1}{\sum_{s \neq v \neq t} \sigma_{st}} \)

- **Graph Centrality**: \( C_G(v) = \frac{1}{\max_{s \neq v \neq t} \sigma_{st}(v)} \)

- **Stress Centrality**: \( C_\sigma(v) = \frac{\sum_{s \neq v \neq t} \sigma_{st}(v)}{\sigma_{st}} \)

As the network size grows the problem of computing exact value of betweenness centrality arise [4]. It is not always possible for the information to propagate via shortest path. Therefore a random walk concept came into existence where the information travel through arbitrary path from source to target. The author of [5] introduced a node centrality measure called m-path centrality where ‘m’ is the length of path. Different techniques were adopted to compute node centrality in a social network such as Random Walk, Self-Avoiding Random Walk of length ‘m’ whose execution time is computed as \( O(m^n \alpha \log n) \) where \( n \) is number of nodes and \( \alpha = [-1/2, 1/2] \).

**Definition 1**: (m-path node centrality) For each node \( v \) of a graph \( G = (V, E) \) the m-path node centrality \( C^m(v) \) of \( v \) is defined as the sum, over all possible source nodes \( s \), of the frequency with which a message originated from \( s \) goes through \( v \), assuming that the message traversals are only along random simple paths of at most m-edges. It can be formalized as:

Recently, the work also done extending the concept of node centrality known as edge centrality.

**Definition 2**: (m-path edge centrality) For each edge \( e \) of a graph \( G = (V, E) \), the m-path edge centrality \( C^m(e) \) of \( e \) is defined as the sum, over all possible source nodes \( s \), of the frequency with which a message originated from \( s \)
traverses e, assuming that the message traversals are only along random simple paths of exactly m edges.

3. PROPOSED WORK
In this paper we are going to present m-path edge centrality forming a m-path through random walk. As the network grows it becomes difficult to find shortest path. It is also known that as the distance increases the influence of the node also decreases in the large network. In other words the closest nodes influence more than the nodes residing farther. Therefore to propagate information through the network effectively we perform density based clustering [14] and the cluster with highest density is selected to apply m-edge centrality performing random walk where ‘m’ is chosen randomly.

3.1 Algorithm
Perform density based cluster on a graph/network. Select the cluster with highest density.

Assumption: G’ is a cluster containing V’ vertex set and E’ edge set that is G’= (V’, E’), D(v) is the total number of edges incident on vertex v, Tr(e) is an array maintaining information of the edges in the form 1 and 0. If 1 then edge has been traversed else 0.

EdgeCentrality(G’=(V’,E’))
1) For v ∈ V’ do
   v ← 1/|V’|
End for
2) For e ∈ E do
   e ← 1/|E’|
End for
3) N←(|V’|/2)+1
4) m ← floor(N(N-1)/2)
5) For count ← 1 to N
   v ← Select uniformly random vertex for walk
   Advertisement_Propagation(v, m)
End for

Advertisement_Propagation(v, m)
1) While ( (m>0) and (D(v) > \sum_{e∈E(v)} Tr(e)) )
   e ← Select uniformly random edge
   w (e_i) ← w (e_i) + 0.1
   v_i ← v_i+1
   m ← m - 1
End while
2) Return w(e_i)
4. CONCLUSION

Different algorithms have been proposed to propagate message from a node that influence maximum number of people on a network. The most efficient algorithm proposed by [3] for unweighted graph has a time complexity of $O(ne)$ and $O(ne + n^2 \log n)$ for weighted graph where 'n' is the number of nodes and 'e' is the number of edges. The concept produced by [5] for m-path centrality has a time complexity of $O(k^3 n^{2-\epsilon} \log n)$. Our algorithm will take time less than $O(k^3 n^{2-\epsilon} \log n)$. As the proposed algorithm will take run time complexity:

$$T(n) = O(n \log n) + O(m^3 n^{2-\epsilon} \log n)$$

So the algorithm is efficient to propagate the advertisement and influence to the maximum number of people in the network.

5. FUTURE WORK

The most common approach is to use clustering to deal with the portion of graph. But the more efficient way is to partition the graph using multilevel partition algorithm [20] or to use MapReduce [21] algorithm and apply the edge centrality algorithm on each partition in a distributed manner. At last obtain the result by merging the partitions into a single whole network to determine the central edge.

6. REFERENCES


