A new structural and semantic approach for identifying influential nodes in social networks

Nesrine Hafiene[1]
[1] *Université de Sousse,*
Laboratoire MARS LR17ES05, ISITCom,*
4011 Hammam Sousse, Tunisie
hafiene.nesrine@gmail.com

Wafa Karoui[1,2]
[2] *Université de Tunis El Manar,*
Institut Supérieur d'Informatique,
2080 Ariana, Tunisie
karoui.wafa@gmail.com

***Abstract*—Influence Maximization is one of the major tasks in the field of viral marketing and community detection. Several approaches have studied the identification of influential nodes in communities that consider only network structure. Therefore, the semantic richness of the information associated with users is not supported. Recently, new approaches have proposed a joint modeling of these two aspects. In this paper, we present a new approach called SND (Semantic and structural influential Nodes Detection) that aims to solve the problem of identifying influential nodes. SND exploits on the one hand the relations between the vertices of the network and on the other hand the attributes characterizing them. This approach combines both the structural and semantic aspect of the network. We test and compare our algorithm with prior algorithms on real world data sets. Experimental results demonstrate the efficiency and effectiveness of SND in maximizing the influence propagation in social networks.**

***Keywords*-Social networks, influential nodes, influence propagation, semantic, community detection.**

I. Introduction

Today's networks are modeled as a graph where a node represents a social entity (human, animal, machine, cell, etc.) and an arc represents social interaction (friendship, common interest, intimate relationship, professional relationship, etc.). The goal of social networks on the internet is to create a circle of friends who have common interests, to keep in touch with these friends, to maintain the link with distant people or to find business partners, a job or others.

Users of social networks are influenced by the information flow that is constantly shared by their friends or families. In a social network the influence is the ability of a person to lead another person to promote an idea, an opinion, to believe in information or to use services or products. The influence propagation represents a mechanism that defines how information is transmitted from one individual to another throughout the network. This mechanism has continued to receive increasing interest from the scientific community in a wide range of contexts. As an example, viral marketing aims to create a «word of mouth» effect where social interactions help spreading a message from one person to another.

In this paper, we focus on the spread of information in a social network and more specifically on the way users disseminate this information. This research area is very close to the diffusion of innovations in marketing. We will discuss the different approaches that have studied the identification of influential nodes and then present our approach. The remainder of this paper is organized as follows. The related works are reviewed in Section 2. In Section 3, the proposed method for maximizing the influence spread are described. The details of the experiments on real dataset are presented in Section 4. Finally, the paper is concluded in section 5.

II. Literature Review

The only challenge we face while diffusing an information is that we need to pick out a set of customers that maximizes the information flow within a network. The main question is: how to maximize the influence within a social network? The problem mentioned above is known as the influence maximization problem, which was first introduced by Domingos and Richardson [1, 2] as a discrete optimization problem. Kempe, Kleinberg and Tardos [3] are the first who formulate the problem as an optimization problem. This problem consists in finding the K best initial diffusers, K being a fixed number. The goal is to know which one of these diffusers will give a maximum diffusion.

The study of diffusion in network is mainly achieved by the development of models that represent well the dissemination process. These models of propagation are used in many applications, such as the identification of influential nodes and the choice of initial content diffusers to get the highest value of spread. Two fundamental propagation models discussed in [3] are the Linear Threshold (LT) Model and the Independent Cascade (IC). First, the LT model introduced by Granovetter, in 1978, in which a user is activated if a predefined number of his neighbors (the threshold parameter) are already active. The second type of model is the IC model introduced by Goldenberg *et al.*, in 2001, where a user who becomes active has more probability to successfully activate his neighbors. In both models, at any time step, a user is either active or inactive.

* Modeling of Automated Reasoning Systems

2161-5330/17 $31.00 © 2017 IEEE
DOI 10.1109/AICCSA.2017.196

There are different research problems in the field of social networks other than the problem of maximizing influence, such as: community detection and extraction of influential nodes. Our work is also based on community detection. Communities can also be called clusters, partitions, cohesive subgroups or modules which share common properties. Many real networks such as social networks, biological networks exhibit community structure. Finding communities is crucial because these communities in a network can help to classify the nodes according to their structural position. Moreover center nodes of the communities can help in obtaining knowledge about the information flow or critical objects. This property can be used in various applications such as to study the spread of influence in social networks. The considered approach for detecting influential nodes is based on community detection. The study of detecting communities in complex networks has many practical applications such as identification of influential nodes of sub communities within large communities can help in identifying influential spreaders in social networks. Since the influence of one node depends on its effects on his neighbours in the same community and nodes from other communities of the whole network.

One area of research that has attracted much interest in social network analysis is community detection. This problem consists in finding an optimal partitioning into k partitions in a graph modeling the social network. This problem is NP-hard [4]. Since the introduction of the first community detection algorithm by Newman *et al.* [5], a very important number of algorithms have been proposed. Among the main methods of community detection proposed in the literature, we can cite those who optimize a function of quality to evaluate it in a given partition. The most used quality function is the modularity introduced by Newman [6]. It measures for each possible partition P a value $Q(P)$ which provides a score on the quality of the generated partition. A host of algorithms for community detection have been proposed. Ben Amor *et al.* [7] propose a new approach named SemMEP for the community detection in a social network. SemMEP takes into account the semantic aspects in a social network more than its structural aspects by grouping together the actors who are close semantically. Zardi and Ben Romdhane [8] propose MWEP a new approach for the community detection in a social network modeled by a weighted graph. MWEP has given good results of the quality of partitioning and the speed of execution. In MWEP, they have defined the «WEP» quality function and the MWEP algorithm to optimize it.

In this work, we focus on the problem of identifying influential spreaders in networks. In social networks, identifying people who can distribute information to a greater number of people (influential nodes) makes it possible to efficiently propagate high quality of information or new ideas. In this context, we distinguish three distinct approaches, namely greedy algorithms, heuristic algorithms and three phases algorithms.

A. Greedy approaches

Many algorithms based on a greedy algorithm have been proposed. The basic idea of the Greedy approaches is to calculate the influence set of each individual, and take turns to choose the node maximizing the marginal influence value until K nodes are selected. Leskovec *et al.* [9] propose an optimized greedy algorithm, called CELF (Cost-Effective Lazy Forward). CELF exploits submodularity to find near-optimal node selections. Chen *et al.* [10] propose two faster greedy algorithms called respectively NewGreedy and MixGreedy. The main idea behind NewGreedy is to remove the edges that will not contribute to propagation from the original graph to get a smaller graph and do the influence diffusion on the smaller graph. The first round of MixGreedy uses NewGreedy algorithm, and the rest rounds employ CELF algorithm. Based on the experiments, they showed that MixedGreedy is much faster than both NewGreedy and CELF. Based on the upper bound, Zhou *et al.* [11] propose an improved CELF algorithm called Upper Bound based Lazy Forward algorithm (UBLF in short) to discover influential nodes in social networks.

B. Heuristic approaches

Beyond that, many other algorithms based on heuristics have been proposed to study the problem of identifying influential nodes. Chen *et al.* [10] also presents a heuristic algorithm called DegreeDiscount that runs faster than MixedGreedy. DegreeDiscount assumes that the influence spread increases with the degree of nodes. Unlike Greedy algorithm, DegreeDiscount algorithm has no provable performance guarantee. The running time of DegreeDiscount is O($k\ log\ n + m$). The Shapley value-based influential nodes (SPIN) algorithm is presented by Narayanam and Narahari [12] to maximize the spread of influence. SPIN selects k nodes with top Shapley values as the target set where information diffusion in the social network is considered as a cooperative game. Wang *et al.* [13] propose a price-performance-ratio inspired from heuristic scheme, PPRank, which investigates how to economically select seeds within a given budget and meanwhile try to maximize the diffusion process. Bae *et al.* [14] propose a novel measure, coreness centrality, to estimate the spreading influence of a node in a network using the k-shell indices of its neighbors. According to the authors, the coreness centrality, which is estimated with the $k-shell$ indices of the neighbors that are adjacent to a spreader, is a simple but notably powerful indicator to assess the capability of information dissemination through the network.

C. Three-phases approaches

Other methods propose to follow a three-phase approach: (a) Detecting the communities of the social network, (b)

Identifying the candidate nodes based on the community structure and (c) Selecting the target set from the candidate nodes. Yan *et al.* [15] propose a two-phase method that combines community detection and naïve greedy algorithm and propose an algorithm named LICD to detect local influenced community. Lv *et al.* [16] considered each community of the social network as a player of a cooperative game and defined a measure based on the number of nodes and weight density of each community as the Shapley value of that community. The number of the nodes that is selected as target nodes from each community is proportional to its Shapley value. Then, in each community, the nodes are selected from two different groups of nodes: (a) bridge nodes; and (b) influential nodes obtained through applying the MixedGreedy [10]. Rahimkhani *et al.* [17] proposed a method where the candidate nodes are selected applying the degree centrality measure in the communities where the number of selected nodes from each community is proportional to its size. Song *et al.* [18] propose an algorithm called Community-based Greedy algorithm for mining $top-k$ influential nodes. It encompasses two components: dividing the large-scale mobile social network into several communities by taking into account information diffusion and selecting communities to find influential nodes by a dynamic programming. Then, to further improve the performance, they propose an efficient algorithm called Parallelized Community-based algorithm to parallelize the influence propagation based on communities and consider the influence propagation crossing communities. Pozveh *et al.* [19] propose two methods, C-SPIN (community-aware SPIN) and C-SGA (community-aware SGA). C-SPIN and C-SGA consist of three main phases: community detection, candidate seeds selection and target set identification. Jaouadi *et al.* [20] propose an algorithm called DIN which combines the structure and the semantic aspects of the network to detect influential nodes.

III. Preliminaries and Problem Statement

A. Problem formulation

In this work, we propose a new approach to identify influential nodes. As discussed above, the actors in the social networks are connected to each other for various interest. So, either a product or an idea spreads in the network, may be interested to an only particular set of individuals. Thus, we propose a new approach for the influence maximization problem which mines the influential nodes in each community rather than the whole network.

We denote the graph as $G = (V, E)$, where $V = \{v_1, ..., v_i, ..., v_n\}$ represents nodes and E represents edges. For each user, we will associate a set of interests that characterize him. These interests will be presented as an attributes vector. Each user is represented by an attributes vector $X_i = (x_{i1},...,x_{ij})$, where x_{ij} is the value taken by the attribute j of the vertex v_i. In the proposed approach this value is binary, either 1 (if the user likes a center of interest) otherwise 0. Let n and m denote the number of nodes of G, $n = |V|$, and the number of edges of G, $m = |E|$, respectively, our goal is to propose a structural and semantic approach for the detection of influential nodes in a social network. Our approach exploits, on the one hand, the relations between the vertices of the network and, on the other hand, the attributes that characterize them. Table I lists the notations to be used extensively in the sequel of this paper.

Table I
Notation Explanation

Notations	Descriptions
$G = (V, E)$	Social network G with a set of nodes V and a set of edges E
m	Number of edges in G
n	Number of nodes in G
K	Number of influential nodes to be mined
C_i	The i^{th} community
M	Number of communities
$\mathbb{NL}$	List of leaders nodes
$\mathbb{NA}$	List of active nodes
θ_c	The threshold value
N_{inf}	The set of influential nodes
A_v	The set of active nodes

B. Basic definitions

Definition 1 (Community)**.** A group of nodes such that the density of edges between nodes of the group is higher than the average edge density in the graph.

Definition 2 (Semantic network)**.** A network that represents semantic relations between concepts. It is a directed or undirected graph consisting of nodes, which represent concepts, and edges, which represent semantic relations between concepts.

Definition 3 (Degree centrality)**.** A simple centrality measure that counts how many neighbors a node has.

Definition 4 (Closeness centrality)**.** The average length of the shortest path between the node and all other nodes in the graph. Thus the more central a node is, the closer it is to all other nodes.

Definition 5 (Influence degree)**.** The number of active nodes in each community divided by the total number of nodes in the graph.

IV. SND (Semantic and Structural Influential Nodes Detection)

SND looks to detect the most influential nodes in an assigned graph. It is interested in the structural and semantic aspects of the network. For this reason, the main idea is to propose a three-phases approach. Indeed, the first phase of our approach explains the structural aspect of the network, the second and third ones focus on the semantic aspect. Figure 1 describes the different phases of the proposed approach.

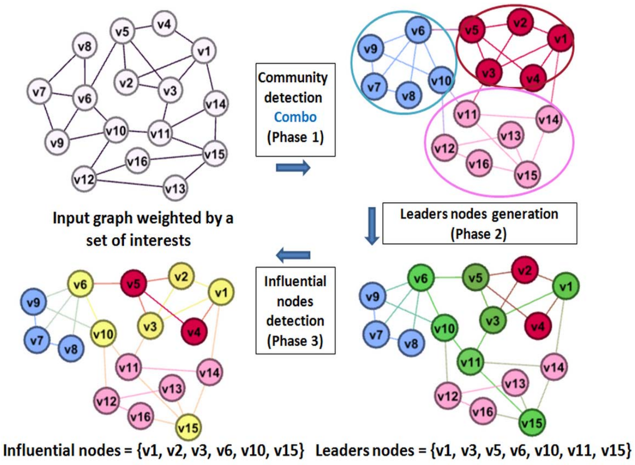
Community detection Combo (Phase 1)
Input graph weighted by a set of interests
Leaders nodes generation (Phase 2)
Influential nodes detection (Phase 3)
Influential nodes = {v1, v2, v3, v6, v10, v15}
Leaders nodes = {v1, v3, v5, v6, v10, v11, v15}
v1
v2
v3
v4
v5
v6
v7
v8
v9
v10
v11
v12
v13
v14
v15
v16

Figure 1. General principle of the proposed approach

In the remainder of this paper, we focus on the attributed networks, where the vertices of the network are described by vectors of attributes. Each user is characterized by a set of interests that are represented as an attributes vector. For each link between two users, we measure the semantic similarity of their centers of interest, if they share it with its neighbors. Thus, they can play an important role in the spread of influence.

Given a graph $G = (V, E)$ where V is the set of nodes and E is the set of edges. The proposed approach begins by detecting communities. In the first step, we proceed to generate the set of nodes playing the role of leaders. We assume that the leading nodes will play the role of the «active» nodes. Thus, they will be the initiating nodes of the diffusion of a new idea. Once, we have generated the leading nodes we apply our influence propagation model. This model is based on the semantic similarity between two nodes in order to identify the active ones in all the network. Finally, once we have selected the set of active nodes, we can then identify all the influential elements.

A. Phase 1: Community detection

To start, the proposed approach detects communities. In this step, we propose to use Combo [21]. It is an optimization algorithm for the community detection, which is able to handle different objective functions. The vast majority of search strategies take one of the following steps to evolve starting partitions: merging two communities, splitting a community into two, or moving nodes between two distinct communities. Combo involves all three possibilities. The performance of Combo is analyzed by modularity and length of the code. This phase permits to form groups so that, the nodes within the same community are connected in a dense manner. Thus, the links between communities are weak.

B. Phase 2: Leaders nodes

In the first part, the phase enables to generate the set of leaders nodes. On the fact that these nodes correspond to the initially active ones. So that, we can select all the influential elements. A node is a leader if its degree centrality is greater or equal then its neighbors. The degree centrality permits to measure the sum of the connection of the node with its neighbors. The equation used to calculate the degree centrality of a node v of a given graph $G = (V, E)$ is defined as follows:

$$dc(v) = \frac{deg(v)}{|V| - 1} \tag{1}$$

Where $deg(v)$ is the number of arcs incidents to v and $|V|$ is the number of nodes of G. In the second part of this phase, we will apply our diffusion model to determine the inactive nodes that can be activated. The two popularly used models are the Independent Cascade (IC) and Linear Threshold (LT) models. They ignore any information related to the user of the network such as its activity, profile or interests. In our diffusion model, we try to introduce the semantic in a graph G. In G, we have associated the link between two actors with the weight which is defined by the semantic similarity of their information which is given in the following equation:

$$Sim(u, v) = \frac{Commun(u, v)}{long(u) + long(v)} \tag{2}$$

Where $Common(u, v)$ represents the common attributes between two nodes u and v. $long(u)$ is the length of the attributes vector of a node u and $long(v)$ the length of the attributes vector of a node v. The length of a vector allows to count the number of 1 in each attributes vector of the binary matrix.

Using the PSI model [22], we can adopt the following model. Each node v in a partition $\mathcal{P} = \{C_1, ..., C_r\}$ is associated with a threshold value $\theta_c(v)$. This value is generated in our algorithm according to the degree of each node between 0.3 and 0.9. Considering the generated threshold values and a set of initially active nodes. A node v becomes active if the sum of the similarity or the total weight of its active neighbors exceeds the threshold value $\theta_c(v)$ associated with a node v. We formally define our diffusion model as follows:

$$\sum_{u \in \mathbb{A}_v} w(v, u) > \theta_c(v) \tag{3}$$

Where $\theta_c(v)$ is the threshold value associated with a node u, $w(v, u)$ is an activation probability that reflects the sum of the similarity between two nodes u and v active and $\mathbb{A}_v$ is the set of active neighbors of v. This process is repeated until none of the inactive nodes can be activated. For example, if $\theta_c(v) = 0.3$, the v node has two neighbors s and t, $w(v,s) = 0.25$ and $w(v,t) = 0.5$. In this case v can activate only t, because $w(v,t)$ is greater than $\theta_c(v)$. Applying our diffusion model, we can obtain a set of active nodes $\mathbb{NA}$ which allows us to select the set of influential nodes in the third phase.

```
Algorithm 1: Generate Leaders
Data: A network $G = (V, E)$, a community $C$;
Result: A $\mathbb{NL}$ set of leaders nodes;
$\mathbb{NL} \leftarrow \emptyset$;
Begin
    Apply the equation 1 to measure the degree
      centrality of each node of $G$;
    foreach node $v \in V$ of a community $C$ do
        if isLeader(v) then
            $\mathbb{NL} \leftarrow \mathbb{NL} \cup \{v\}$;
        end
    end
End
```

The Algorithm 1 describes the phase of leaders nodes generation. To distinguish the leaders nodes from the other ones, we used the $isLeader()$ function. This function allows us to compare the degree centrality of the nodes.

C. Phase 3: Detection of influential nodes

The previous phase allows us to determine the set of active nodes. In this last phase, we aim to determine the set of influential nodes. Our objective is to select from the active nodes those that have a maximum influence. Inspired by CGA [18], we start with detecting the influential nodes from each community.

For each community we calculate its degree of influence. The degree of influence (defined by the equation 4) is equal to the number of active nodes in each community (V_{active}) divided by the total number of nodes in the graph $G(V)$.

$$R(C_r) = V_{active}/V \tag{4}$$

To start, the proposed approach detects influential nodes from the community that admits the value of the maximum influence degree. After, our goal is to determine for each active node its closeness centrality (defined by the equation 5). It is the sum of the length of the shortest paths between the node and all other nodes in the graph. A node is considered important if it can quickly reach the other nodes. The closeness centrality is defined by the following equation:

$$CCenter(v_i) = \frac{1}{\sum_{v_j \epsilon V} |ShortPath(v_i, v_j)|} \tag{5}$$

Where $ShortPath(v_i, v_j)$ represents one of the shortest paths between two nodes where v_i belongs to the set of leaders nodes and v_j belongs to the set of active nodes. With $|ShortPath(v_i, v_j)|$ is the length of the shortest path that leads v_i to v_j.

Once we have calculated the closeness centrality of each leader node, we classify these nodes following an ascending order. The node admits the greatest closeness centrality is the influential node. For example, in a context where nodes are streets and the tops are intersections, the hubs with the greatest closeness centrality are the best candidates for hosting services. The third phase of the proposed method is detailed in Algorithm 2. It consists in detecting the set of influential nodes. In fact, our goal is to select among the active nodes those that will have a maximum influence. The algorithm of SND is outlined in Algorithm 3. It first detects communities applying the Combo algorithm [21] (line 3). Then the algorithm proceeds to generate the set of nodes playing the role of the leaders. We have assumed that the leaders nodes will play the role of the «active» nodes and subsequently they will be the initiating nodes of the diffusion of a new idea (Algorithm 1). Once we have generated the set of the leaders that are the initiators, we apply our diffusion model. This model is based on the semantic similarity between two nodes in order to identify the active nodes in all the network (from line 6 to line 15 of Algorithm 3). Finally, once we have selected the set of active nodes, we can identify all the influential nodes (Algorithm 2). The time complexity of SND is: O$(n^2 log(M) + M(m+n) + Ml^3T)$, where T is the shortest path time between a leader node and an active node and l is the number of the leaders nodes in each community M.

```
Algorithm 2: Detecting Influential Nodes
Data: An graph $G = (V, E)$, $\mathbb{NA}$;
Result: A set $N_{inf}$ of influential nodes;
$N_{inf} \leftarrow \emptyset$, $\mathbb{NL} \leftarrow \emptyset$, $\mathbb{NA} \leftarrow \emptyset$;
Begin
    $N_{inf} \leftarrow \emptyset$;
    $C \leftarrow$ detect community with Combo
    $M = |C|$
    while $M \neq \emptyset$ do
        Apply the equation 4 to measure the degree of
          influence of each community. We start by
          detecting the influential nodes within the
          community $C_{max}$ which admits the value of
          the maximum degree of influence;
        $A_{C_{max}} \leftarrow$ { active nodes in $C_{max}$ };
        while $(C_{max} \neq \emptyset) et (A_{C_{max}} \neq \emptyset)$ do
            Calculate the closeness centrality of each
              node in $A_{C_{max}}$ using the equation 5;
            Classify these nodes in ascending order of
              closeness centrality;
            $v_{max} \leftarrow$ the node having the largest
              closeness centrality;
            $N_{inf} \leftarrow N_{inf} \cup \{v_{max}\}$;
        end
    end
    return $N_{inf}$;
End
```

Algorithm 3: SND

```
Data: An initial graph $G = (V, E)$ weighted by the
      centers of interest of the users, a set of leaders
      nodes $\mathbb{NL}$, a set of active nodes $\mathbb{NA}$;
Result: A set $N_{inf}$ of influential nodes;
$N_{inf} \leftarrow \emptyset$;
$\mathbb{NL} \leftarrow \emptyset$;
$\mathbb{NA} \leftarrow \emptyset$;
Begin
    Apply the Combo algorithm to detect communities;
    foreach community $C$ do
        Generate Leaders$(C, G)$;
        foreach Node $u$ in $C$ do
            Assign a weight $\theta_c$ for each node;
        end
        foreach Edge $(u, v)$ in $C$ do
            Calculate the similarity sim$(u, v)$ using the
              equation 2;
            $w(u, v) \leftarrow sim(u, v)$;
            if $\left(\sum_{u \in \mathbb{A}_v} w(v, u) > \theta_c(v)\right)$ then
                $\mathbb{NA} \leftarrow \mathbb{NA} \cup \{u\}$;
            end
        end
    end
    Detecting influential nodes $(\mathbb{NA}, G)$;
    return $N_{inf}$;
End
```

V. Experiments

We evaluate the effectiveness and efficiency of the proposed SND algorithm on real networks where we associated a set of centers of interest with each node. The algorithm proposed in this paper is compared with other ones that are known in literature. We will describe these algorithms in the sequel of this section. All of the implementations are performed in C++ and run on a Intel Core i3-4005U processor with a 2 GHz CPU and 4 GB memory.

A. Algorithm and Parameters

In the experiments, we compare our algorithm SND with existing representative algorithms for influence maximization: 1) **Coreness Centrality** [14] where the authors propose a novel influence measure, the coreness centrality, to quantify the spreading capability of a node using the coreness of its neighbors. This approach is based on the idea that a powerful spreader has more connections to the nodes that reside in the core of network and 2) **C-SPIN and C-SGA** [19] where two methods considering the community structure of the social networks and influence-based closeness centrality measure of the nodes are presented to maximize the spread of influence.

We use two metrics to evaluate the performance of SND, namely the influence propagation and the runtime. We also evaluate the effect of parameters on performance, including the number of nodes that can be influenced by the set of K influential nodes and the threshold for each node, θ, which is generated in our algorithm randomly between 0.3 and 0.9. For parameter K, SND generates a significant number of influential nodes independently of the K increase. For the other models, we make the execution several times and choose $top-k$ influential nodes that maximize the influence propagation.

B. Experimental Results

We tested the performance of the different algorithms on real networks of the UCI database [23]. The real networks which are applied in this study evaluate the performance of the different algorithms such as Zachary's Karate Club, Political Books, Adjacency of nouns, Celegans and Netscience. These networks are widely used by several algorithms to test their performance since their community structure is known in advance. In order to test the algorithm, our model randomly associates a set of centers of interest to nodes.

*1) **Varying the influence propagation with the size of the network:*** by varying the size of the network and like we observe in Figure 2, we find that in the first three networks SND maintains influence propagation values close to CC. These two approaches are based on the centrality of the neighbors of a node to determine the active nodes. By varying network size and in a larger network like Netscience, the influence propagation of SND is always better than the two approaches C-SPIN and CC.

With a small number of initially active nodes, our approach covers a large number of activated nodes. According to the results, when we increase the network size, SND is still able to have a better value of influence propagation.

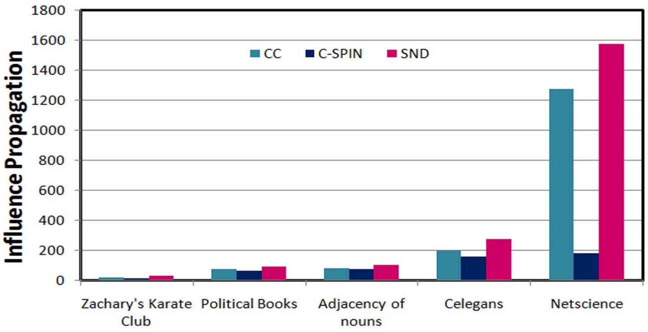

Figure 2. Varying the influence propagation with the size of the network

*2) **Varying the influence propagation with** K:* in this experiment, we will vary the number of initially active nodes K from 5 to 30 for the three approaches on the Netscience network which contains 1589 according to the influence propagation. By varying the number of initially active nodes K and according to the Figure 3, we observe that the influence propagation of CC is close to SND. CC determines its ranking based on the centrality of nodes neighbors. From the results of C-SPIN, we can observe that with the increase of K its influence propagation is still very far from those of SND and CC. We can conclude that SND generates a significant number of influential nodes independently of the K increase.

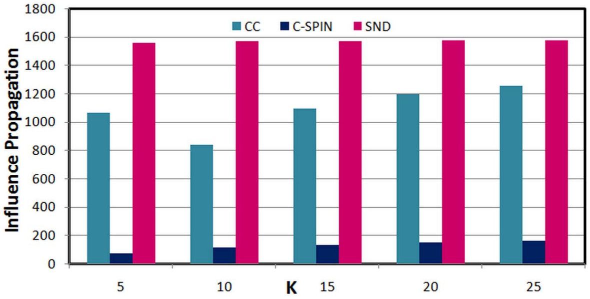

Figure 3. Varying the influence propagation with K

*3) **Varying the time execution with the size of the network:*** for the time execution, while varying the size of the network and according to Figure 4, we note that SND initially admits time execution close to C-SPIN. For a small network respectively 34, 105, 112, 297, SND maintains an time execution close to C-SPIN. In a larger network C-SPIN has a better time execution. This is due to the fact that the partitioning phase of the proposed approach takes time for SND to achieve a higher time execution than C-SPIN on the Netscience network while the C-SGA approach has the highest time execution.

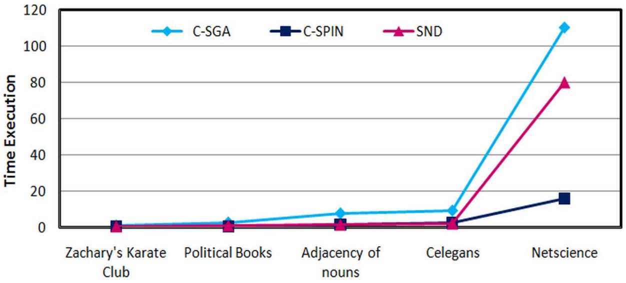

Figure 4. Varying the time execution with the size of the network

*4) **Varying θ_c with the time execution and the influence propagation:*** this experiment enables to evaluate the variation of threshold θ_c on the Celegans network, with K = 5.

Table II
VARYING θ_c WITH THE TIME EXECUTION AND THE INFLUENCE PROPAGATION

θ_c	0.3	0.4	0.5	0.6	0.7	0.8
Propagation	278	236	125	22	15	4
Time execution	1.7	1.56	1.4	0.8	0.4	0.1

According to the Figure 5, the influence propagation decreases by the increase of the value of θ_c. Due to the increase of θ_c, the number of nodes that will be activated by our diffusion model will decrease because the probability of activation becomes less than θ_c. Thus, the number of diffusers, the propagation and the runtime of the algorithm decrease.

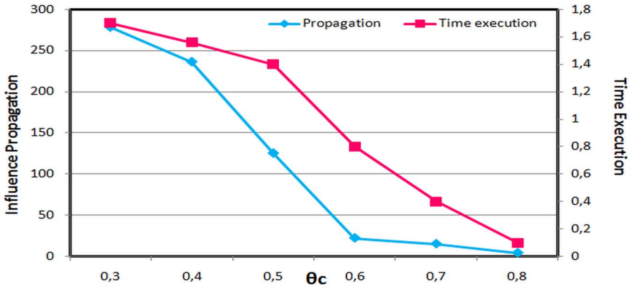

Figure 5. Varying θ_c with the time execution and the influence propagation

*5) **Varying the time execution with K:*** in this experiment we will vary the number of initially active nodes K from 5 to 30. We evaluate the three approaches in the Netscience network respecting time execution.

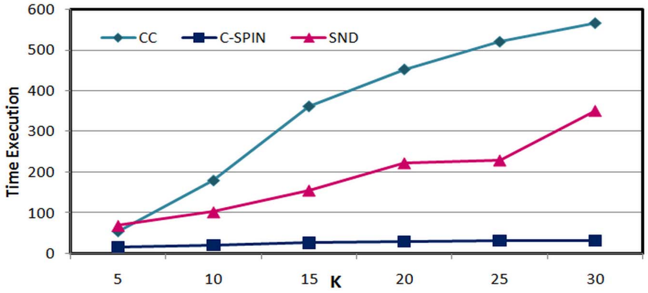

Figure 6. Varying the time execution with K on the Netscience network

According to the Figure 6, when we vary K the C-SPIN approach admits the best values of time execution. For C-SPIN, when we vary K the influence propagation is very low because this approach admits the best time execution. SND maintains better values than CC which takes more time execution on the Netscience network.

*6) **Varying the time execution with the size of the network:*** we see in Figure 7 that by varying the size of the network, SND achieves better values of the influence propagation better than those of two other algorithms C-SPIN And CC. According to the results of this evaluation, while increasing the size of the network SND still able to have a better spread of influence. Thus, it covers a large number of influential nodes in any type of network.

VI. Conclusion

In this paper, we have studied the problem of identifying influential nodes in information networks and we proposed an approach called SND (Semantic and structural influential Nodes Detection), which combines the structural and semantic aspects of the network. Indeed, in order to validate the

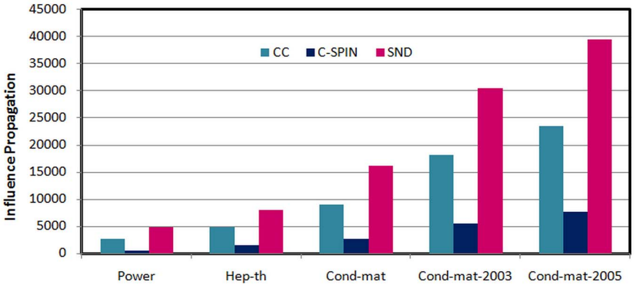

Figure 7. Varying the time execution with the size of the network

utility of the approach, we have compared it with three well-known approaches then, we used real networks where we associated a set of centers of interest with each node. From the simulations carried out on the selected networks, we found that SND shows a great performance in the detection of a set of influential nodes that maximize the influence propagation. Empirical studies show that our algorithm has great improvement on the influence propagation compared with CC, C-SPIN and C-SGA algorithms. At the moment, our goal is to test SND on large scale social networks such as «Twitter». Unfortunately, these data are not available to the general public because of the privacy of twitters. There are several interesting future directions. Notably, it is relevant to ameliorate the time execution of the used community detection algorithm. It is also interesting on the level of experiments on real networks that are not randomly weighted as in the present case but which will be previously weighted by centers of interest.

References

[1] Pedro Domingos and Matt Richardson. Mining the network value of customers. *ACM/SIGKDD International Conference on Knowledge Discovery and Data Mining (ICKDDM 2001) 57–66*.

[2] Matt Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. *ICKDDM (2002) 61–70*.

[3] David Kempe, Jon Kleinberg and Éva Tardos. Maximizing the spread of influence through a social network. *ICKDDM (2003) 137-146*.

[4] Chris Ding, Xiaofeng He, Hongyuan Zha, Ming Gu and Horst Simon. A min-max cut algorithm for graph partitioning and data clustering. *IEEE International Conference on Data Mining (ICDM 2001) 107-114*.

[5] Michelle Girvan and Mark Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the USA, 98-2 (2001) 404-409*.

[6] Mark Newman. The structure of scientific collaboration networks. *National Academy Journal of Sciences 98-2 (2001) 404-409*.

[7] Sami Ben Amor, Lotfi Ben Romdhane and Mounira Harzallah. SemMEP: nouvelle approche sémantique pour la détection des communautés dans un réseau social. *Journées francophones d'Ingénierie des Connaissances (IC 2016)*.

[8] Hédia Zardi and Lotfi Ben Romdhane. WMEP: efficiently mining community structures in weighted large scale social graphs. *International Conference on Reasoning and Optimization in Information Systems (ROIS 2013)*.

[9] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen and Natalie Glance. Cost-effective outbreak detection in networks. *ICKDDM (2007) 420-429*.

[10] Wei Chen, Yajun Wang and Siyu Yang. Efficient influence maximization in social networks. *ICKDDM (2009) 199–208*.

[11] Chuan Zhou, Peng Zhang, Jing Guo, Xingquan Zhu and Li Guo. UBLF: an upper bound based approach to discover influential nodes in social networks. *IEEE International Conference on Data Mining (ICDM 2013) 907-916*.

[12] Ramasuri Narayanam and Yadati Narahari. A shapley value-based approach to discover influential nodes in social networks. *IEEE Transactions on Automation Science and Engineering (T-ASE 2011) 130-147*.

[13] Yufeng Wang, Athanasios V. Vasilakos, Qun Jin and Jianhua Ma. PPRank: economically selecting initial users for influence maximization in social networks. *IEEE Systems Journal PP-99 (2015) 1-12*.

[14] Joonhyun Bae and Sangwook Kim. Identifying and ranking influential spreaders in complex networks by neighborhood coreness. *Physica A: Statistical Mechanics and its Applications Journal 359 (2014) 549–559*.

[15] Qiuling Yan, Shaosong Guo and Dongqing Yang. Influence maximizing and local influenced community detection based on multiple spread model. *International conference on Advanced Data Mining and Applications (ADMA 2011) 82-95*.

[16] Jiaguo Lv, Jingfeng Guo and Huixiao Ren. A new community-based algorithm for influence maximization in social network. *Computational Information Systems Journal 9-1 (2013) 5659–5666*.

[17] Khadije Rahimkhani, Abolfazl Aleahmad, Maseud Rahgozar and Ali Moeini. A fast algorithm for finding most influential people based on the linear threshold model. *Expert Systems with Applications Journal 42-3 (2015) 1353-1361*.

[18] Guojie Song, Xiabing Zhou, Yu Wang and Kunqing Xie. Influence maximization on large-scale mobile social network: a divide-and-conquer method. *IEEE Transactions on Parallel and Distributed Systems (TPDS 2015) 1379-1392*.

[19] Maryam Hosseini-Pozveh, Kamran Zamanifar and Ahmad Reza Naghsh-Nilchi. A community-based approach to identify the most influential nodes in social networks. *Information Science Journal 43-2 (2016) 2229-2252*.

[20] Myriam Jaouadi and Lotfi Ben Romdhane. Din: an efficient algorithm for detecting influential nodes in social graphs using network structure and attributes. *IEEE/ACS International Conference of Computer Systems and Applications (AICCSA 2016)*.

[21] Stanislav Sobolevsky, Carlo Ratti and Riccardo Campari. General optimization technique for high-quality community detection in complex networks. *Physical Review E-90 (2014) 1-19*.

[22] Myungcheol Doo and Ling Liu. *IEEE Transactions on Services Computing (TSC 2014) 387–400*.

[23] Christopher DuBois. UCI network data repository. http://networkdata.ics.uci.edu, *2008*.