Maximizing influence in Social Networks using Game Theoretic Framework

Prasanna Patil, *CSA, Indian Institute of Science*
Deepak Poonia, *CSA, Indian Institute of Science*

Abstract—The problem of information diffusion in social networks is one of the most studied problem in recent times. The problem has received significant attention because of its wide application such as viral marketing and assessing behavior of the nodes in the network. The aim here is to identify set of nodes in the social network which are most influential. This is called target set selection problem where aim is to find a target set which would cause the most expected information spread in the network. Many algorithms and heuristics have been developed to address the problem from efficiency and accuracy perspective. This report is the initial study of game theory based approaches to solve influence maximization problem in social networks.

Index Terms—Social Networks, Influence Maximization, Shapley Values, Coalition Games, Network Centrality

1 Introduction

Social network is structured representation of underlying network between real world identities such as people or organizations where nodes in the network represent these identities and edges represent relationship between them. Complex interaction between various real world identities can be easily represented using a graphical model. These models allow us to study complex interaction that take place in these identities. Social networks are being used in many real world applications such as collaborative filtering, viral marketing, citation networks, web search and identifying most used hubs in a internet network.

Social networks play a key role in information diffusion among the nodes in the network. The structure of the social network represents the extent to which a node influences its neighbors in the social network. In social networks it is interesting to identify set of nodes which play important role in affecting the behavior of most nodes in social network. Such nodes are called influential nodes. Consider the following example where this kind of study is important.

A company would like to market its new product. However it has a limited budget. Hence the problem is to figure out which nodes in the network would cause most information spread. We want to discover a target set of nodes, which are also known as seed nodes such that by marketing the product to these nodes would cause the maximum information spread. That is maximum nodes in the network would come to know about a particular product.

Another motivating example is to analyze co-authorship network. There exists a natural social interaction among researchers from various communities. The co-authorship network contains an edge between two authors if they have worked together previously. This co-authorship network could be useful to find out most influential researchers.

1.1 Diffusion of Information

The concept of diffusion of information refers to propagation of ideas, concepts and opinions in the social network. There are two models studied widely for the diffusion of information in the social network. They are linear threshold model [1] and independent cascade model [2]. The network is represented using a graph $G = (V, E)$ where V is set of nodes and E is set of edges in the network.

Linear Threshold Model: Linear Threshold (LT) model was proposed in [1] and generalized in [3]. A node is active if it has adopted the information otherwise it is inactive. The information is propagated in discrete time interval through the network.

Initially, seed nodes are the only active nodes. As time passes information propagates through the network based on threshold value of each node. The propagation happens as follows: Node i is influenced by its neighbor node j according to weight W_{ij}. Here, it is assumed that all the weights have been normalized that is $\sum_{j \in N_i} W_{ij} \leq 1$, where N_i is set of neighbors of node i.

Each node has threshold θ_i associated with it, which is randomly selected from interval $[0, 1]$. The threshold represents the fraction of neighboring nodes that must be activated for i^{th} node to be active. A node gets activated if $\sum_{j \in A(i)} W_{ij} \geq \theta_i$, where $A(i)$ is number of neighbors of i which are active. This process continues until no more nodes can be activated and network reaches a stable point.

Independent Cascade Model: Independent Cascade (IC) model has been studied in the context of marketing. In the model, the directed edge between node $(u, v) \in E$ has a probability P_{uv} associated with it which reflects the extent to which node u can influence node v. Here, also, diffusion proceeds in discrete time stamps. Initially set of seed nodes are the only active nodes.

The diffusion proceeds as follows. At each time interval, we have a set of nodes which are recently activated in the previous time interval. These recently activated nodes can activate their neighbor nodes with probability determined by weights on directed edges between nodes. The activation of a neighbor node is a random process, that is a node u succeeds in activating a neighbor node v with probability

P_{uv} or fails with probability $1-P_{uv}$. The activation of nodes is independent of each other and past attempts of activation.

Now, the nodes that are activated in the interval become recently activated node for next time interval. Note that at each interval only recently activated node gets a chance to activate its neighbors. Hence, each node gets exactly one chance to activate its neighbors. This is different from Linear Threshold Model when a node can get activated at any time instant when $\sum_{j \in A(v)} W_{jv} \geq \theta_v$ is satisfied.

Nodes activated at any time instant remain activated throughout the rest of the diffusion process. The process stops when no new nodes can be activated.

1.2 Target Set Selection Problem

The target set selection problem is to select a set of initial seed nodes which would cause maximum information diffusion throughout the network. The diffusion is determined by either Linear Threshold Model or Independent Cascade Model as described above. There are two primary ways in which target set selection problem can be formulated.

1. Top-K Nodes: One way to address target set selection is to pick top K nodes from network which have most information spread in the network when selected as seed nodes. We define an objective function $\sigma(\cdot)$. If S is set of initially active nodes then $\sigma(S)$ is the number of nodes active at the end of diffusion process. Hence, objective in top-k node problem is to find a set S' such that,

$$\sigma(S') \geq \sigma(S)\,,\ \forall S \subset N, |S| = k \ and\ |S'| = k$$

2. λ-coverage problem: Here, we are given $\lambda \in [0, 100]$. The problem is to find set of seed nodes S, having minimum size, such that $\sigma(S)$ contains at least λ percentage of nodes of entire network.

2 Summary of Relevant Papers

Much attention has been given to the problem of top-k nodes selection and λ-coverage by algorithmic community. Following is the summary of few approaches based on game theoretic framework.

2.1 Shapley Value based Approach for influence maximization in Social Networks

This approach was first proposed by Suri and Narahari [4]. There have been many approaches for target set selection in social network such as ı) The greedy algorithm [5], known as KKT ii) the CELF algorithm [6], known as LKG iii) CWY algorithm proposed in [7]. It is known that LKG algorithm outperforms KKT by 700 times. Also, it has been proven that the problem of target set selection in IC model is NP-Hard by [5], [8] and [9].

The fundamental idea is to utilize a game theoretic framework called cooperative game theory to identify most influential nodes in social network. First a cooperative game is defined over the nodes of the social network where each node is treated as an agent in the game.

Specifically, a cooperative game is defined between a set of players $N = 1, 2, ..., n$ where n is the number of players participating in the game. A *characteristic function* $V : 2^N \rightarrow R$ assigns a real value to every coalition of players. The value $V(s)$ represents total transferable utility that can be achieved by players in S without any help from players in $N \setminus S$.

The Shapley value is one of the most celebrated solution concept in game theory which provides a method to divide the total gain that is achieved by the coalition of players within the coalition. It is a fair method of dividing the total gain as some players are more important to coalition than others. Effectively Shapley value describes the average marginal contribution of a node to any coalition of nodes in the network. The Shapley value of a player can be calculated as follows:

$$\Phi_i(N, v) = \frac{1}{n!} \sum_{\pi \in \Omega} \{V(C_i(\pi) \cup \{i\}) - V(C_i(\pi))\}$$

Here, Ω is set of all permutations of players. π is one permutation from Ω and $C_i(\pi)$ is set of all players that occur before player i in the permutation.

The authors of [4] propose a cooperative game over social network where nodes are players of the game and characteristic function defined as $V(C) = \sigma(C)$ where $C \subseteq N$. The idea can be summarized in following steps:

1) Compute the Shapley values of all nodes in network and then rank nodes according to their Shapley values.
2) Then select top K nodes from this ranking list and select them as seed nodes.

The idea of information diffusion is captured as coalition formation in this game. Hence, Shapley values calculated for the game captures the marginal contribution of each node in process of information diffusion. This algorithm is known as SPIN algorithm.

The top-k nodes are then selected as follows: (i) first node from ranked list is added to the list of top-k nodes. (ii) the subsequent nodes are added to the list of top-k nodes if they are not adjacent to nodes the top-k list. (iii) the process ends when size of list is equal to k. The top-k nodes are selected in this way because a node will activate its neighbor node if it is seed node. Hence, there is not much to gain by selecting neighbor nodes.

The authors of [4] apply their SPIN algorithm on Linear Threshold Model to identify top-k influential nodes. The authors also suggest a simple modification to the algorithm using which λ-coverage problem can also be solved. To solve λ-coverage problem, the nodes are selected one by one from the ranked list of nodes based on their Shapley values until λ-coverage is achieved.

Through extensive experimental analysis authors have shown that the SPIN algorithm performs quite well on non submodular $\sigma(\cdot)$ function. The algorithm outperforms KKT algorithm in terms of quality of solution when $\sigma(\cdot)$ is known to be non submodular function. A function is submodular if it satisfies following property:

$$\sigma(S \cup \{i\}) - \sigma(S) \geq \sigma(T \cup \{i\}) - \sigma(T)\,, S \subset T \subset N\,, i \in N \setminus T$$

The authors have proposed two threshold models, namely, i) Multiplication threshold model and ii) Minimum threshold model, which are non submodular threshold models.

The approximation guaranteed by KKT algorithm works well under the assumption of monotonically increasing and submodularity of $\sigma(\cdot)$ function. However, SPIN algorithm is found to work quite well without this assumption, too. However, the target set produced by SPIN is occasionally little inferior to the ones produced by KKT, LKG or CWY algorithm under submodularity property of $\sigma(\cdot)$. Also, SPIN algorithm is faster compared to LKG algorithm, hence much faster compared to KKT algorithm.

2.2 Efficient Computation of Game Theoretic Network Centrality Measures

A centrality of a node in the network measures the importance of the node in some sense. There have been many approaches to measure the centrality of a node in the network. However, all of these measures treat nodes as individuals and do not take the activity of a group of nodes into consideration. To address this issue, measures for group centrality have been devised. Group centrality measures how much importance a particular group has in the network. However, there is no clear way to determine importance of individual nodes from group centrality. Hence, cooperative game theoretic framework has been derived to address this issue.

One of the first approach to use cooperative game theory for measuring importance of a node was proposed by [4] to find out target set of a social network to maximize the influence in the network. The key idea here is to define a cooperative game over the network and model the characteristic function such that it addresses the underlying problem and calculate Shapley values of each node to measure centrality.

However, one issue with calculating Shapley values is that its order of calculation is exponential in terms of time. The authors [10] propose 5 special cases of cooperative games which can be applied to networks to measure the centrality of a node by calculating exact Shapley values in polynomial time. One of the cooperative games proposed by authors is similar to the one used by [4] for social network influence maximization.

Here, we show the equation proposed by [10] to calculate exact Shapley values of nodes in a social network for cooperative game developed by [4]. N is number of nodes and v is characteristic function and $deg_G(v_j)$ is degree of a node v_j in graph. The Shapley value of of a node can be calculated as follows:

$$\Phi_i(N, v) = \sum_{v_j \in \{v_i\} \cup N_G(v_i)} \frac{1}{1 + deg_G(v_j)}$$

The formula can be interpreted using probability that a node v_i will influence its neighbor node v_j. This probability can be measured by calculating probability that a randomly permutation of nodes N such that node v_i appears just before node v_j and all of the other neighbors of node v_j appear afterv_j. This probability is,

$$P(\epsilon) = \frac{1}{1 + deg_G(v_j)}$$

, where ϵ = Event where v_i appears before v_j and all of the other neighbors of v_j appear after v_j. Then, the marginal contribution of node v_i is simply the sum of above probabilities for each node $v_j \in N(v_i)$.

Note that this formula captures notion of absolute influence in which v_i influences v_j with probability 1. This is not the case when various information diffusion models are considered. Hence, this formula, even though exact for absolute influence, is not applicable when stochastic information diffusion models are considered.

2.3 Information Diffusion in Social Network in Two Phases

The authors in [11] study information diffusion in social network in multiple phases. They specifically focus on two phase diffusion where given a certain budget (in terms of total number of seed nodes) the selection of seed nodes is split into two phases. Initially only seed nodes of first phase are activated. After some fixed time d, nodes of second phase are activated. The problem is to select the set of seed nodes k_1 and K_2 such that at the end of information diffusion process total number of activated nodes is maximized.

Diffusion of information is a random process, hence most algorithms try to select seed nodes which maximize the information spread in expectations, however for some instances the set of seed nodes might turn out to have worse performance. In such a case, a multi phase diffusion is natural solution. It allows to intervene and modulate decisions during diffusion process in order to avoid such instances.

However, there is a trade off associated with multiple phase diffusion because of delay associated with activating second set of seed nodes. This delay between different phases is defined as Scheduling time by the authors. This may be undesirable when the value of information decreases with time or when the company is in a highly competitive market and people get influenced by products which are first introduced to them.

The authors study this problem under Independent Cascade Model and propose an objective function which would maximize two phase information diffusion in social network. The authors also propose several modifications to existing algorithms to maximize this objective function instead of single phase diffusion objective function.

The authors propose two formulation to solve the multiple phase information diffusion problem: i) *Myopic* variation tries to find out seed nodes of first phase K_1 without looking at the value of seed nodes K_2. Once obtained optimal value of K_1, algorithm tries to find out K_2 which optimizes influence at the end of second phase. ii)*Farsighted* approach calculates the value of K_1 by also considering the optimal value of K_2 and final spread of information in the network. Hence, this variation takes more time to run than the Myopic version.

The authors of the paper carry out experimental study and show that gain in expected spread is 7% over single phase information diffusion, on an average,for various algorithms. Moreover, the authors note that both Farsighted and Myopic variation have almost the same gain in expected diffusion. The authors also explore the idea of optimal scheduling and budget splitting to maximize influence in social network.

The authors conclude by suggesting to use multiple phase diffusion under moderate to non-existent temporal conditions. However, in case of strict temporal conditions they suggest to use single phase diffusion.

2.4 Competitive Threshold Models for information diffusion

Competitive information diffusion models allow us to model scenario in which more than one companies are competing with each other to market their products among users. Essentially, both of them want to select seed nodes such that they end up capturing a larger market segment. Note that, here, the problem of finding the most influential seed nodes in the network depends upon the set of seed nodes selected by the other company.

The separated threshold model:- This model is essentially similar to LT model. A separate threshold is maintained for each company for each node. A node that is influence by a particular product stays influenced with that product throughout the diffusion process.

Wave proportional competitive LT model:- This model is restricted to two companies competing in the market. The probability that a node becomes influenced by company A is proportional to the ratio of the sum of weights of active neighbor influenced by company A to the sum of weights of total active neighbors.

For more information regarding above models, we redirect interested readers to [12].

2.5 Bounds for TR Model

Recently, Khim et el [13] proved lower bound and upper bound on total number of nodes activated at the end of information diffusion under TR model for a given set of seed nodes A. They also proposed a greedy algorithm for finding most influential nodes based on finding the set of nodes that maximize the lower bound.

For any given vertex subset $A \subseteq V$ in the graph with weighted adjacency matrix B of graph G with $B = (b_{ij})$. Let $\bar{A} = V \setminus A$. Define a vector $b_{\bar{A}} \in R^{|\bar{A}|}$ such that for any index $i \in \bar{A}$, $b_{\bar{A}}(i) = \sum_{j \in A} b_{ji}$. A walk in a graph is a sequence of nodes $V_1, V_2, ..., V_r$ such that $(V_i, V_i + 1) \in E$, for $1 \leq i \leq r - 1$. The weight of the walk is defined as $\omega(w) := \Pi_{e \in w} b_e$. For a set of walks $W = w_1, w_2, ..., w_n$, we denote weight of set of walks to be $\omega(W) = \sum_i^r \omega(w_i)$.

Upper Bound:- In general, any subset A of nodes V of graph G satisfies the following inequality

$$I(A) \leq |A| + b_{\bar{A}}^T \left(\sum_{i=1}^{n-|A|} B_{\bar{A},\bar{A}}^{i-1} \right) 1_{\bar{A}}$$

This provides an upper bound on the total number of nodes that can be activated at the end of diffusion process in TR model.

Lower Bound:- The influence of seed set A satisfies the inequality

$$I(A) \geq \sum_{i \in V} \sup_{p \in P_{A \to i}} \omega(p)$$

where $P_{A \to i}$ is the set of all paths from A to i such that only the starting vertex lies in A.

3 Proposed Experiment

3.1 Triggering Model

Triggering Model was initially proposed by Kempe et el. [5] as a generalization of both Independent Cascade and Linear Threshold model. More specifically, Kempe proves that IC model and LT model are special cases of Triggering Model. The Triggering Model as described by Kempe in [5] diffuses information in a social network as follows:

- Each node V has a probability mass function P_v associated with it which maps each subset of neighbors of node V, denoted as $N(v)$ to a real number between $[0, 1]$ such that:

$$\sum_{A \in 2^{N(v)}} P_v(A) = 1$$

- At the beginning of diffusion process, a subset of nodes A_v for each node $V \in G$ is sampled according to P_v.
- Diffusion happens at each discrete time interval. At each interval, it is checked whether one of the neighbors of a node is active and is part of the subset of nodes A_v. If it is then the corresponding node is also activated.

We experiment with the Triggering Model (TR) of information diffusion on co-authorship network. A co-authorship network is a networks in which nodes represent the authors of research papers and edges between them represent whether the authors have collaborated. Our contributions are as follows:

1) We perform experiments and evaluate performance of various algorithms for social network influence maximization problem in context of Triggering model of information diffusion.
2) We propose two general purpose approaches for generating probability mass function associated with each node in the graph.

3.1.1 Generating PMF for each node

There are three general methods for generating PMF associated with each node as described by TR model of information diffusion.

Sampling uniformly from unit simplex:- A unit simplex in $k + 1$ dimension Δ^{k+1} is defined as a point

$$\{\mathbf{x} \in \mathbf{R^{k+1}} : \sum_{i \in [k+1]} x_i = 1, x_i \geq 1\}$$

.

To generate a probability distribution for node V, we sample a point at uniformly random from unit simplex $\Delta^{|2^{N(v)}|}$. We then assign each x_i to each subset of $N(v)$ such that $P_v(A) > P_v(B)$ if $B \subseteq A \subseteq N(v)$.

Intersection of neighbors of neighbors to generate PMF :- In this method, the probability of a subset A of the neighbors of a node V is proportional to common neighbors

of nodes in A, that is, $P(A) \propto |\cap_{u \in A} n(u)|$. Then we normalize P(A) such that

$$\sum_{A \in 2^{N(v)}} P_v(A) = 1$$

Union of neighbors of neighbors to generate PMF :- In this method, the probability of a subset A of the neighbors of a node V is proportional to union of all neighbors of nodes in A, that is, $P(A) \propto |\cup_{u \in A} n(u)|$. Then we normalize P(A) such that

$$\sum_{A \in 2^{N(v)}} P_v(A) = 1$$

We experiment with the intersection method for the purpose of Co-Authorship network. This can be justified because a group of authors are most likely to collaborate among their common neighbors as it represents common research interests.
We propose that the union method of generating PMF for nodes can be used for Social networks because nodes in social network can propagate information to any of their neighbors and union method represents how likely it is that a group of nodes will diffuse information to the node for which PMF is being generated.

4 ALGORITHMS

We performed experiments on TR model of information diffusion using the following algorithms for finding the most influential nodes in social network.

4.1 Greedy (KKT)

Greedy algorithm (also referred as KKT) was proposed by Kempe et el [5]. This is an approximation algorithm that provides approximation bound of $(1 - \frac{1}{e})$ when information diffusion function $\sigma(\cdot)$ is submodular in nature. However, the problem with this algorithm is that the $\sigma(\cdot)$ function is #P-hard function. Hence, its exact value can't be calculated. Generally, the value of this function is estimated using monte carlo simulations over many runs.

The greedy algorithm selects influential nodes in iterative manner. In each iteration, it selects a node that maximizes the influence spread at the end of diffusion process given already selected seed nodes. For this, it goes through all nodes and finds out a node that gives best increase in influence spread.

4.2 CELF++

CELF++ [14] is an improvement of CELF [15] algorithm. CELF (Cost Effective Lazy Propagation) is based on the idea that marginal contribution of a node in current iteration of Greedy algorithm can't be better than its marginal contribution in previous iteration. Hence, it systematically calculates marginal contribution only when it is required and avoids unnecessary computation of $\sigma(\cdot)$ function.

CELF maintains a list $< u, \Delta_u(S) >$ sorted in decreasing order of $\Delta_u(s)$. The quantity $\Delta_u(S)$ represents, marginal contribution of node u w.r.t current set of seed nodes S. $\Delta_u(S)$ is evaluated only for the top node at a time and the list is resorted if necessary. If the top node remains on top then there is no need to calculate $\Delta_u(S)$ for any of the other nodes, as marginal contribution of a node w.r.t a set of seed nodes S can only decrease as set of seed nodes S increases. Hence, top node is picked as next seed node. If the node no longer remains on top then $\Delta_u(S)$ is recalculated for top node in re-sorted list.

CELF++ optimizes CELF algorithm by avoiding computation of $\Delta_u(S)$ for the top node. Essentially, along with $\Delta_u(S)$, CELF++ keeps track of cur_best node which denotes the node which has largest $\Delta_u(S)$ over all nodes examined in current iteration. CELF++ also calculates $\Delta_u(S \cup \{cur\_best\})$ for each node, in addition to $\Delta_u(S)$. For each node u. it then keeps track of $prev\_best$ which denotes the cur_best node at the time of evaluating $\Delta_u(S \cup \{cur\_best\})$. At each iteration, top node with highest $\Delta_u(S)$ is checked and added to the seed nodes S. Then, for any node u where $prev\_best$ was the last picked seed nodes, the algorithms simply updates $\Delta_u(S') = \Delta_u(S \cup \{cur\_best\})$. Hence, it avoids computation of $\sigma(\cdot)$ for nodes most of the time by pre-computing it beforehand.

4.3 SPIN

SPIN algorithm is Shapley Value based Influence Maximization algorithm. SPIN algorithm defines a coalition game over social network where nodes of the graph G are agents of the game. A coalition game is defined by $\Gamma(N, v)$ where N = agents and v = characteristic function of the game. Characteristic function is defined as $v : 2^{|N|} \rightarrow R$. It maps each coalition of agents in N to a real number in R which denotes the marginal contribution of that coalition in the game.

SPIN algorithm finds out influential nodes for given network by defining $v = \sigma(\cdot)$. Hence, value of a particular coalition of nodes N is equal to the influence spread achieved by that coalition.

Shapley value is very important concept in such games because of it satisfies certain properties. Most importantly, Shapley value allows us to estimate marginal contribution of any agent of N in the game. It allows fair division of total value of the game among individual players. The concept of Shapley values was developed by Shapley [16], it takes into account relative importance of each player into account. History of SPIN algorithm is already covered section 2.1.

4.4 Heuristic based approaches

Heuristic based algorithms are independent of underlying information diffusion model. These algorithms find most influential nodes by applying heuristic searching methods which makes sense empirically.

4.4.1 Iv-Greedy

IV Greedy is a recently proposed algorithm by Wang et el. [17]. This algorithm is independent of underlying information diffusion model considered. This algorithm builds an influence vector for each node in the social network. The influence vector of a node captures, how much influence is exerted by a node on its neighboring nodes in non stochastic version, that is directly considering the probability of a node influencing its neighboring nodes.

The algorithm first computes the influence vectors of all nodes and selects nodes iteratively such that at each iteration, the node whose influence vector corresponds to highest influence exerted on all nodes with respect to already selected set of seed nodes. Although this algorithm was developed to solve influence maximization problem under multi-path asynchronous threshold model (MAT) proposed by authors, we observe good quality of influential nodes obtained by applying this algorithm on Triggering Model, as well.

4.4.2 Top K

This algorithm first finds the expected influence spread of each node individually and selects top K nodes as the seed node for most influential nodes in the network. This is similar to greedy approach but each node is analyzed individually. Hence, it is similar to performing a single iteration of greedy algorithm for finding top 1 node and then selecting top K nodes with highest influence spread.

4.4.3 Degree discount

This algorithm iteratively selects highest degree node and reduces degrees of all neighboring nodes by 1, essentially removing the selected node from the graph.

5 Experiments

We experiment with above algorithms to solve influence maximization under Triggering Model of information diffusion.

5.1 Experimental Datasets

We consider two graph datasets to carry out experiments. We generate one synthetic graph and take one real world dataset of collaboration network.

1. Synthetic datasets There are several popular methods for generating random graphs. We use Sparse Random Graphs also known as Erdos Renyi Random Graphs. This random graph is defined by two parameters: N = Number of nodes, p = probability of an edge between any two nodes. We generate a random graph with $N = 500$ and $p = 0.02$. Graphs generated with this algorithm have are balanced with similar vertex degrees and have shorter path lengths between nodes. They have low clustering coefficient and can be efficiently generated.

2. Real World Datasets There are many real world social networks, citation networks and co-authorship networks available on Stanford snap repository [18]. We consider co-authorship network of High Energy Theoretical Physics collaboration network. However, network is very huge with 9877 nodes and 25998 edges. The highest degree of a node is 65 which is very large for generating node probability mass functions required for TR model. The size of this distribution is 2^{65} and time required for generating PMF is very long.

Hence, we consider a subset of this graph. Specifically, we only consider co-authorship network for papers published in a single year span. This yields graph which is comparatively smaller. This graph contains 1312 nodes, 1334 edges and highest degree of a node is 7. This graph may not be of any interest for real world application but we found it to be suitable for carrying out experiments with TR model. Only limitation while considering large graphs is generating node distributions and time required to find out influential nodes.

As mentioned previously, we generate node distributions for node using intersection method described in section 3.

5.2 Experiment Setting

We consider Triggering Model of information diffusion for our experiments. We use intersection method as described in section 3 for generating node distribution. All code is implemented in Python. Code is executed on a standalone machine with Intel Xeon E5 processor, 32 GB of RAM.

The number of Monte Carlo simulations is fixed to 30 for HeP dataset whereas it is set to 50 for Sparse Random Graph. Number of permutations for calculating Shapley values (T) is fixed to 10 for HeP dataset whereas it is fixed to 30 for Sparse Random Graph. Moreover, for calculating Shapley Values on HeP dataset we run 32 threads simultaneously all with 10 permutations and 30 Monte Carlo simulation and then average the Shapley values over outputs obtained by all threads to improve precision of Shapley values. The run time measurements are shown for a single thread of execution.

Implementation of above algorithms along with required information diffusion model is available at https://github.com/prasanna08/SNInfluenceMaximization. The code has been written in Python programming language.

5.3 Experiment Results

5.3.1 Quality of generated influential nodes

We experimented with HeP 2003 dataset for measuring quality of influential nodes generated by various algorithms described in section 4. We try to solve the Top K influential nodes problem and compare algorithms based on quality of top K nodes generated. The quality of seed nodes is measured in terms of expected influence achieved by them. The expected influence is measured using Monte Carlo simulation of $\sigma(\cdot)$ function.

Figure 1 shows expected influence achieved by various algorithms. X - axis represents value of K given as input to the algorithm. Y - axis represents the expected amount of influence generated by seed nodes.

As can be seen from the figure above, the highest quality of seed nodes is generated by CELF++ algorithm which is not surprising, since it is highly optimized version of Greedy algorithm given by Kempe et el. [5]. Here, we do not perform experiments on Greedy (KKT) algorithm as it takes significantly long time to generate the output.

The SPIN algorithm is performing well compared to other algorithms such as CELF++, given that it doesn't take as much time to run. The other heuristic based algorithm IV Greedy is performing on par with CELF++.

It can also be noted that Degree discount heuristic based algorithm, which is simplest algorithm of all, is also performing very good. This can be justified because it is a sparsely connected graph. As previously mentioned, HeP 2003 graph consists of only 1312 nodes and 1344 edges

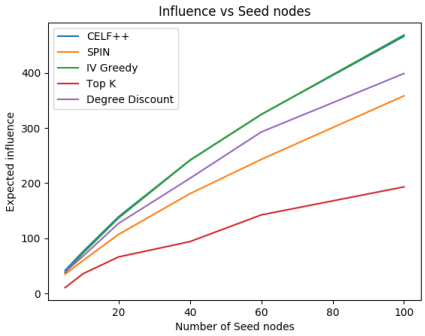

Fig. 1. Expected influence achieved by the seed set generated using various algorithms.

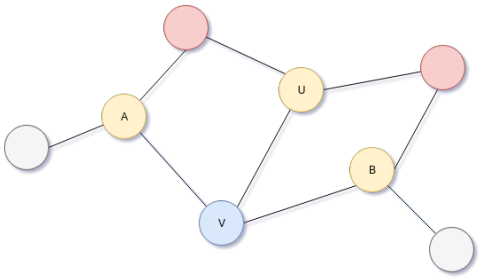

Fig. 2. A pathological case when generating node distribution using intersection method

TABLE 1
Expected influence spread when picking top K nodes in SPIN Directly vs Non Adjacently

K	Non Adjacent Selection	Direct Selection
5	34.965	34.89
10	59.63	60.1
20	107.145	106.115
40	180.93	180.68
60	243.11	242.565
100	358.28	357.165

which is very sparse. Moreover, IV Greedy and Degree discount are robust to how node distribution for each node is generated, i.e. they don't depend on it.

In such sparse graphs, a pathological case as shown in Figure 2 could occur when we generate node distributions using intersection method. Consider that we want to calculate distribution for node V. From neighboring nodes, it can be seen that the value $P(A) > 0$, for any subset $A \subseteq N(v)$ only when $U \in A$. Hence, whenever U is active, V is also activated and added to it as its Shapley value as marginal contribution. Hence U has higher chances of activating V since any sampled subset of neighboring nodes will contain U. However, when we order nodes according to their Shapley values we consider that marginal contribution and not exactly the nodes which they are activating. Hence, it may happen that nodes with higher Shapley value may be activating same nodes and cumulative influence spread because of them is small. Currently top K nodes are selected by picking top K non adjacent nodes with highest Shapley value.

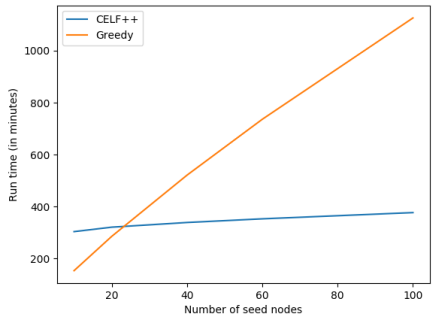

Fig. 3. Running time of CELF++ and Greedy algorithms

Moreover, we also tried to see if there is any difference in expected influence spread if we directly pick top K nodes according to their Shapley values rather than picking non adjacent nodes. We didn't find any difference between expected influence generated by the direct selection and the non-adjacent node selection methods. This can be seen from Table 1, where first column shows expected influence generated by picking non adjacent nodes, as proposed by Suri et el. [4]. Second column shows expected influence generated by picking top K nodes directly.

5.3.2 *Running time of various algorithms*

We used synthetic dataset, generated as described in section 5.1, for comparing time required to generated top K influential nodes using various algorithms. We restrict ourselves to compare time required by CELF++ and Greedy algorithm only. This can be justified because all other algorithms take same amount of time whether we want top 5 nodes or top 100 nodes. They are independent of value of K. We wanted to measure how time increases as we increase K.

Figure 3 shows run time comparison of CELF++ and Greedy algorithm. X - axis represents value of K and Y - axis represents the running time of algorithm in minutes.

As it can be seen from figure 3, running time of CELF++ is significantly lower than the running time of Greedy algorithm. However, for lower value of K, it seems that Greedy should be preferred over CELF++, in case of TR model. The extra time required by CELF++ could be arising from the fact that it keeps track of two different marginal contribution per each node, namely $\Delta_u(S)$ and $\Delta_u(S \cup \{cur\_best\})$. Hence, it needs to do twice as much as Monte Carlo simulation as compared to Greedy approach. However, it can be seen that such calculation pays off when value of K is increased beyond a threshold.

We don't make any exhaustive run time analysis for other algorithms. However, we observed that on an average it takes about an hour for SPIN algorithm to generate Shapley values when number of permutations considered is 40 and Monte Carlo simulation count is 30. Also, it takes about 42 minutes for Top K algorithm to generate top K nodes. Other algorithms such as IV Greedy and Degree discount run in order of seconds.

TABLE 2
Mean and STD of Shapley values

Model	Mean	STD
IC	19.22	1.592
LT	5.634	0.293
TR	54.493	7.053

TABLE 3
Top 10 Shapley value comparison

TR Model	IC Mode	LT Model
87.5	24.4	7.3
76.0	24.0	6.9
76.0	23.8	6.7
74.1	23.2	6.1
71.6	23.1	6.1
65.3	22.3	6.0
63.5	21.8	5.9
62.8	21.6	5.9
62.6	21.6	5.9
62.1	21.3	5.8

5.3.3 Comparison of Shapley values

We compare Shapley values generated by SPIN algorithm on TR model of information diffusion to those which are generated when running SPIN algorithm on IC or LT Model of information diffusion. For this purpose, we use HeP 2003 dataset.

We generate node distribution using intersection method as described previously. For generating edge distributions required for IC model, we use random sampling over $U[0, 1]$. That is each edge is assigned a value randomly from interval $[0, 1]$. Once generated, these values are fixed in the rest of the experiments.

Similarly, for LT Model, we generate weights associated with a pair of nodes W_{uv} by sampling a number randomly from interval $[0, 1]$ and normalizing it by in degree of that vertex. Note that these weights are not symmetric, i.e. $W_{uv} \neq W_{vu}$. Normalizing weights using in degree of node satisfies inequality $\sum_{u \in N(v)} W_{uv} \leq 1$, which is requirement for LT Model.

The Shapley values generated by SPIN algorithm for the three information diffusion models as described above, are shown in Table 2. The mean and standard deviation of Shapley values generated by SPIN algorithm is shown in Table 2. Table 3 enlists Shapley values of top 10 nodes.

It can be seen from the table that the Shapley values are approximately scaled version of each other. To confirm this, we calculate Spearman ranking correlation between these values for top 100 Shapley values and we find correlation of 0.99. Hence, it can be concluded with high confidence that Shapley values are just scaled in different models.

However, this is not the case for the nodes having these Shapley values. We observed that the top 10 nodes extracted by SPIN algorithm on these models are completely different without any overlap between them. This result shows that the notion of top influential nodes for SPIN algorithm is highly dependent on underlying diffusion process. The pathological case explained in previous section can also be used to explain this result. The TR model will place high emphasis on node U when it comes to activating node V, which need not be case in IC or LT model. Hence, these results show the consequences of using intersection method to generate node distribution.

The scale of generated Shapley values can be explained as follows: in case of IC model, each vertex activates its neighbor independently of other neighbors, hence marginal contribution of a node is expected to be high. However in case of LT model, a node is activated only when weights of already active neighbors crosses a threshold. Hence, a single node has lower chance of activating a neighbor node.

Triggering Model allows a node to activate its neighbor node when the neighbor node is part of selected subset which is randomly sampled according to P distribution of node. This results in higher chances of a node u activating its neighbor v if it happens that all subsets in which u is present have very high probability of activating node u. Moreover, if u activates v and v activates other nodes in subsequent time steps then u has high marginal contribution. Hence, the scale of Shapley values is highest in case of TR model.

6 Conclusion and Future Work

We conclude the following statements from experiment results that we have obtained.

- Triggering Model is not suitable when number of nodes is large and high degree nodes are present in graph due to sheer amount of time involved in calculating node distributions.
- Either directly selecting Top K nodes or selecting non adjacent top K nodes in SPIN algorithm doesn't make much difference when graph is very sparse.
- Heuristic algorithms such as Degree discount and IV Greedy work very well for sparsely connected graphs.
- Shapley values obtained for IC, LT and TR model vary in their scales, also top nodes are non overlapping for all 3 models. However, this seems to be sensitive to particular choice of generating node distributions in TR model.

Further explorations can be done to get more insights into TR Model.

- Implement and test performance of algorithm that selects seed nodes based on maximizing lower bound (described in section 2.5).
- We couldn't perform experiment with union method of generating node distributions due to excessive amount of time required to obtain simulation results. One can explore how Shapley values are generated by such distributions.
- Address λ-coverage problem using different algorithms by measuring minimum number of nodes required to achieve λ percentage of influence in graph.
- Perform similar experiments on dense graphs and observe how it differs from Sparse graphs.

References

[1] M. Granovetter, "Threshold models of collective behavior," *American Journal of Sociology*, vol. 83, no. 6, pp. 1420–1443, 1978.

[2] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing Letters*, vol. 12, no. 3, pp. 211–223, Aug 2001. [Online]. Available: https://doi.org/10.1023/A:1011122126881

[3] D. J. Watts, "A simple model of global cascades on random networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 9, pp. 5766–5771, 2002. [Online]. Available: https://www.pnas.org/content/99/9/5766

[4] R. Narayanam and Y. Narahari, "A shapley value-based approach to discover influential nodes in social networks," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 1, pp. 130–147, Jan 2011.

[5] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03. New York, NY, USA: ACM, 2003, pp. 137–146. [Online]. Available: http://doi.acm.org/10.1145/956750.956769

[6] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Trans. Web*, vol. 1, no. 1, May 2007. [Online]. Available: http://doi.acm.org/10.1145/1232722.1232727

[7] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 199–208. [Online]. Available: http://doi.acm.org/10.1145/1557019.1557047

[8] D. Kempe, J. Kleinberg, and É. Tardos, "Influential nodes in a diffusion model for social networks," in *Automata, Languages and Programming*, L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1127–1138.

[9] J. Kleinberg, *Cascading Behavior in Networks: Algorithmic and Economic Issues*. Cambridge University Press, 2007, p. 613–632.

[10] T. P. Michalak, A. V. Karthik, P. L. Szczepanski, B. Ravindran, and N. R. Jennings, "Efficient computation of the shapley value for game-theoretic network centrality," *CoRR*, vol. abs/1402.0567, 2014. [Online]. Available: http://arxiv.org/abs/1402.0567

[11] S. Dhamal, P. K. J., and Y. Narahari, "Information diffusion in social networks in two phases," *CoRR*, vol. abs/1706.07739, 2017. [Online]. Available: http://arxiv.org/abs/1706.07739

[12] A. Borodin, Y. Filmus, and J. Oren, "Threshold models for competitive influence in social networks," in *Internet and Network Economics*, A. Saberi, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 539–550.

[13] J. Khim, V. Jog, and P.-L. Loh, "Computing and maximizing influence in linear threshold and triggering models," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. USA: Curran Associates Inc., 2016, pp. 4545–4553. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157382.3157605

[14] A. Goyal, W. Lu, and L. Lakshmanan, "Celf++: Optimizing the greedy algorithm for influence maximization in social networks," vol. 47-48, 01 2011, pp. 47–48.

[15] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '07. New York, NY, USA: ACM, 2007, pp. 420–429. [Online]. Available: http://doi.acm.org/10.1145/1281192.1281239

[16] L. S. Shapley, "A value for n-person games," in *Contributions to the Theory of Games II*, H. W. Kuhn and A. W. Tucker, Eds. Princeton: Princeton University Press, 1953, pp. 307–317.

[17] W. Wang and W. N. Street, "Modeling and maximizing influence diffusion in social networks for viral marketing," *Applied Network Science*, vol. 3, no. 1, p. 6, Apr 2018. [Online]. Available: https://doi.org/10.1007/s41109-018-0062-7

[18] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[19] S. Singh, K. Singh, A. Kumar, H. Shakya, and B. Biswas, *A Survey on Information Diffusion Models in Social Networks: Second International Conference, ICAICR 2018, Shimla, India, July 14–15, 2018, Revised Selected Papers, Part II*, 01 2019, pp. 426–439.