2018

Disrupting diffusion: Critical nodes in network

Preeti Bhardwaj
Iowa State University

Disrupting diffusion: Critical nodes in network

by

Preeti Bhardwaj

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Samik Basu, Major Professor
Pavan Aduri
Andrew Miner

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this thesis. The Graduate College will ensure this thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2018

DEDICATION

I would like to dedicate this thesis to my sister Kirti Bhardwaj and to my parents Jaikumar Bhardwaj and Nirmala Devi without whose support I would not have been able to complete this work. I would also like to thank my friends and family for their encouragement and support during the writing of this work.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Samik Basu and Dr. Pavan Aduri for their guidance, patience and support throughout this research and the writing of this thesis. Their insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my committee member Dr. Andrew Miner. This work is partially supported by NSF Grant CCF 1555780

ABSTRACT

With the advent and proliferation of connected entities such as social, marketing, scientific and computer networks, it has become immensely important to understand and analyze the impact of one entity's influence on another in the network. In this context, our objective is to identify a set of entities, which when made ineffective (quarantined or protected) will maximally disrupt the spread of influence in the network. We formulate and study the problem of identifying nodes whose absence can maximally disrupt propagation of information in the independent cascade model of diffusion. We present the notion of impact and characterize critical nodes based on this notion. Informally, impact of a set of nodes quantifies the necessity of the nodes in the diffusion process. We prove that the impact is monotonic. Interestingly, unlike similar formulation of critical edges in the context of Linear Threshold diffusion model, impact is neither submodular nor supermodular. Hence, we develop heuristics that rely on greedy strategy and modular or submodular approximations of impact function. We empirically evaluate our heuristics by comparing the level of disruption achieved by identifying and removing critical nodes as opposed to that achieved by removing the most influential nodes.

CHAPTER 1. INTRODUCTION

Diffusion is the phenomenon of spread of information in connected network of entities. Information can be influence, opinion, disease while entities can be people/person, groups and communities. There can be a good spread like promotion of a new product, spread of medical and technological innovations and there can be a bad spread like spread of a disease or spread of a fake news which can have negative effect. In market and social sciences understanding information diffusion helps in designing viral marketing strategies, adoption of new idea or product by large number of people. Understanding information diffusion such as spread of a disease in epidemiological network, spread of computer virus in computer network plays a very important role in mitigating its effect.

1.1 Background

Two of the widely studied problems in this context involve (a) *influence maximization problem*—finding the set S of entities, called *seed set*, such that when the information originates from S, its diffusion in the network is maximal (Kempe et al. (2003); Chen et al. (2009)). (b) *source identification problem*—once the diffusion has occurred, identify a set of entities that can be classified as source/seed of the diffusion(Lappas et al. (2010); Shah et al. (2011); Jiang et al. (2018)). Addressing influence maximization problem results in finding a seed set, called *max seed*, of entities that can cause maximal information spread. Whereas source identification leads to identifying a possible seed that caused the observed influence propagation. Given a seed set S, if $\sigma(S)$ denote the expected number of nodes that are influenced, when the origin of information propagation is S, then max-seed identification is same as computing $\text{argmax}_S \sigma(S)$. Given a network and a set of influenced nodes *Inf*, source identification amounts to computing an S whose influence $\sigma(S)$ maximally aligns with *Inf*.

1.2 Motivating Problem

In this work, we study a problem that is orthogonal to both of the above problems: *identify a set of size k of entities, which when removed from the network, maximally disrupts the diffusion of influence that may have started at any seed set.* More formally, the goal is to identify a set of nodes C such that, after removal of C from the network, $\sigma(S)$ is maximally reduced for every seed set S. We refer such entities C as *critical nodes*, and we call problem of computing such nodes as the *identifying critical nodes* (ICN) problem. The importance of addressing this problem cannot be understated. In social networks, influence of un-founded opinions or propagation of fake news can be avoided by identifying and informing/isolating the critical nodes. In computer network security, protecting critical nodes from known worms (via patching, security updates) can help in protecting the critical network-infrastructure from repeated disruption due to worm-attacks. In the context of disease propagation, helping critical communities that were once impacted by epidemics can make a difference in overall health of the population.

Note that, the critical nodes are not necessarily the max-seed; rather the critical nodes can be viewed as the ones whose presence is "critical" in ensuring that the max-seed indeed has maximal influence on the network. In other words, criticality of a nodes can be described equivalently as how their presence is important for maximizing the result of diffusion or (conversely) how their absence is important for minimizing the result of diffusion.

1.3 Illustrative Example

To illustrate the unique nature of critical nodes, consider the example network in the Figure 1.1(a) and the objective is to identify one critical node. Directed edges in the network indicate that the influence diffuses from the source to the destination, and the edge annotations capture the probability of the diffusion. Such a diffusion model is referred to as the independent cascade (IC) model, which directly captures the notion that new information/behaviors are contagious (Kempe et al. (2003); Kleinberg (2007)). Following the IC model, each node gets one chance to influence its neighbors. For simplicity, assume that the all probabilities are set to 1. Now, the most influential node is v_0 as it can influence the entire network. However, removing v_0 shown in 1.1(b) does not disrupt the influence diffusion if some other seed is chosen. For instance, any one of v_2, v_3, or v_4 can still act as a source of influence that spreads to the majority of the network. The critical node, in this network, is v_4; removal of v_4 in figure 1.1(c) will maximally disrupt information diffusion from any other node. For instance, in its absence, the expected diffusion from v_0 is 4, and the expected diffusion from each of the other nodes is 1. Intuitively, v_4 is most critical implies that the removal of any other node cannot reduce that sum of expected diffusion from all nodes any further.

1.4 Contributions

Consider the ICN problem when k equals 1, i.e., identify a single critical node. A naive approach to critical node identification works as follows: For each node v, remove it from the network and compute how much $\sigma(S)$ is reduced due to removal of v. This approach has at least two bottlenecks. It is immediate that such strategy in not viable even for reasonably small networks as one has to cycle through all possible seed sets. Secondly, this approach may not find such v. Consider the following scenario: Let v_1 and v_2 be two nodes and S_1 and S_2 be two seed sets such that removal of v_1 will maximally reduces $\sigma(S_1)$, whereas removal of v_2 maximally reduces $\sigma(S_2)$. There is no single vertex whose removal will maximally reduce both $\sigma(S_1)$ and $\sigma(S_2)$.

One of our contributions is to characterize criticality by introducing the notion of *impact* of a set of nodes. Intuitively, impact of a set of nodes S quantifies the reduction in the expected diffusion from all nodes when the set S is removed from the network. That is, rather than reviewing the reduction in the expected diffusion from each seed set, we consider the reduction in the expected diffusion for all nodes. Consequently, higher impact of set of nodes implies higher criticality of the set. We formalize the ICN problem as finding a set of nodes with maximal impact.

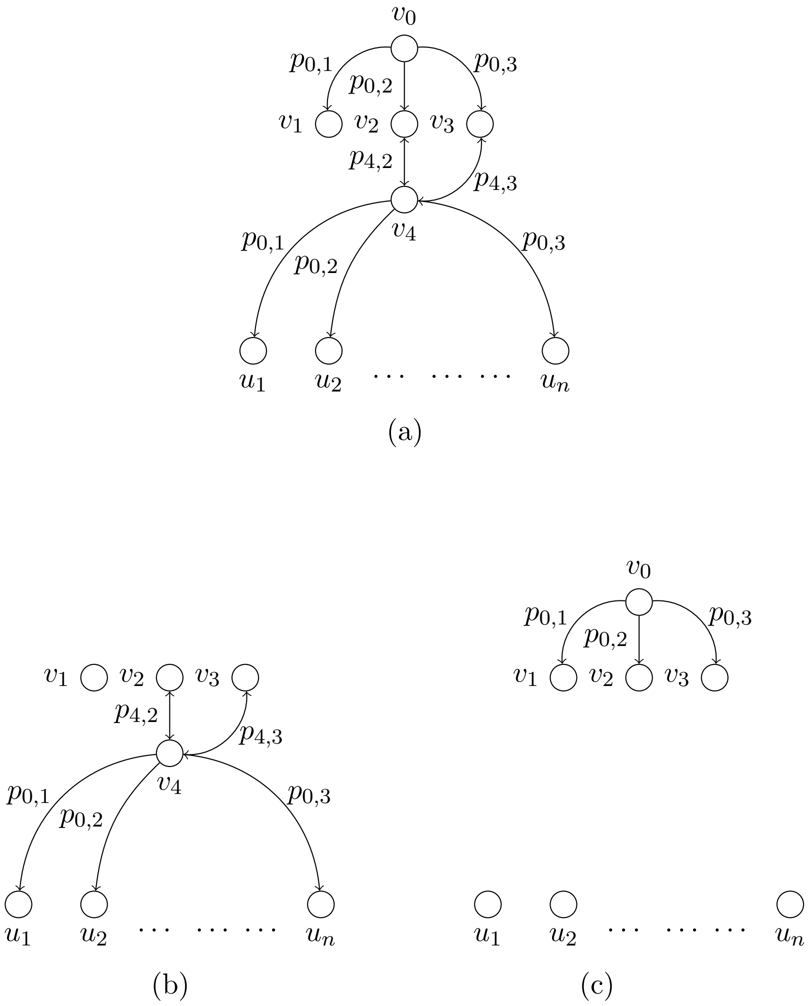
We prove that impact is monotonic and is neither submodular nor supermodular. As a result, greedy algorithm applied to optimization of impact does not provide usual $(1 - 1/e)$ approximation guarantees as it does when applied to address different variations of influence maximization problems and source detection problems. Given the hardness of the problem, greedy algorithm is still a viable strategy, where the impactful set is computed assuming submodularity of the impact function.

However, the greedy algorithm is expensive and inefficient on even moderate size graphs. In the context of influence maximization, the work of (Borgs et al. (2014); Tang et al. (2014, 2015)), give an efficient, randomized, approximate algorithm to estimate the expected influence of any seed set. Using the ideas from their work, we obtain a more efficient algorithm to compute high impact nodes. We refer to this algorithm as CRIT-SET.

We empirically validate that high impact nodes are indeed critical nodes. We conduct extensive experiments to show that using our heuristic CRIT-SET, removal of high impact nodes indeed disrupts the diffusion in the network. We compare our strategy against the baseline strategy, TOP-INFL, where the nodes in the max seed set are removed from the network. We show that removal of high impact set of size k (as per CRIT-SET) causes more reduction (up to $20 - 30\%$) in the influence than the removal of best possible seed set of size k (as per TOP-INFL). Consider another heuristic that identifies top k-impactful nodes as critical nodes (TOP-CRIT)—the strategy results in an optimal solution if the impact function is modular. Our experiments indicate that this heuristic is much faster and still produces a solution whose quality (in terms of disruption of influence) remains between that of CRIT-SET and TOP-INFL. Collectively, the experiments validate our claim that the characterization of criticality in terms of impact is viable and effective.

1.5 Organization

The rest of the paper is organized as follows. In Chapter 2, we discuss prior work related influence maximization problem, source identification problem and the study done in the area of disrupting the influence and how these problems are different from our problem. In Chapter 3, we formally define the ICN problem and complexity of the problem. We introduce the notation of strength $ST(G)$ and impact IM_G function for a graph G. Also we discuss the modular characteristics of the impact function IM_G. In Chapter 4, we present the greedy computation for critical nodes as well as we provide the efficient implementation of greedy algorithm using reverse reachable sets and its efficiency. In Chapter 5, we present our experimental setup, results performed on various real world social networks and show that maximum influence after removal of node following Crit-Set nodes is less than removal of node following Top-Infl nodes. In Chapter 6, we summarize our contributions, and discuss the possible extensions to this work.



(a)

(b) (c)

Figure 1.1 Illustrative Example

CHAPTER 2. REVIEW OF LITERATURE

To study diffusion in epidemiology, computer security, marketing, and social networks we need a mathematical framework which can best represent these networks as operational models. The model in which a social network is represented by a graph $G = (V, E)$ where V are the nodes i.e., entities (people, groups, communities) of the network and E are the edges which represent relationship between those entities. The spread of information is represented by state of node being active (active stating believer of the information) or inactive in case of social networks. It is also represented by infected or susceptible for the disease spread in epidemiology. The strength of influence between the neighbors decide whether or not influence spread from infected nodes to its susceptible neighbors. This type of process is best represented by Independent Cascade (IC) model and the Linear Threshold (LT) model. These two models are the most basic and well-studied diffusion models. In this paper, our focus will be on the Independent Cascade (IC) model.

2.1 Diffusion models

2.1.1 Independent Cascade model

Independent Cascade (IC) is the model in which at every (discrete) time step i, each node u, which is newly activated at time step i - 1, will activate each of its (inactive) neighbor v with probability $p_{u,v}$. This captures diffusion at the i^{th} step. The diffusion process continues till no new node is activated. Every edge has a probability associated with it, which shows the infection it can spread on its neighbor. The probability $p_{u,v}$ is the probability of u infecting v where u is the source and v is the target. Based on the probability of the edge, each infected node can infect its neighbor in the next time step. Each node once infected remains infected for the rest of diffusion process but has only one chance to infect its neighbors. When there is no new node to infect the diffusion process stops.

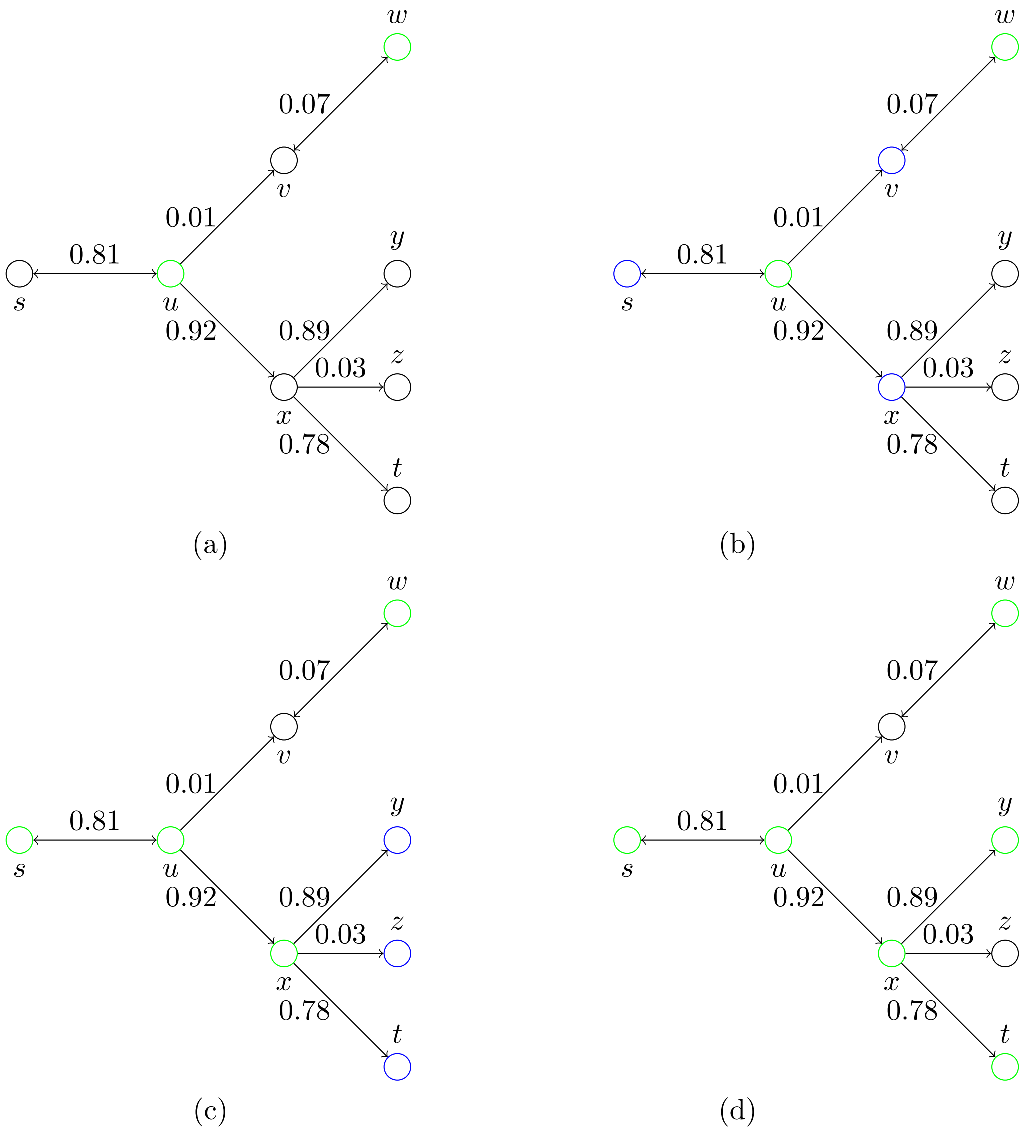
Here is the example of illustration of Independent cascade model. In the example 2.1 (a)green color nodes are the activated nodes and blue nodes are those susceptible nodes that has chance to get activated at next time stamp. Let us suppose node u and w are active nodes at time t shown in figure 2.1 (b). So we push nodes u and w in the running queue. We process every node from the running queue. First we pop u from queue, it has neighbors s,v and x. At time $t+1$ node u will try to activate node s,v and x. All the activating events are independent of each other and depends on the coin toss. In this case it may successfully activates s and x because of high propagation probability of ($p_{u,s}$ =0.81) and ($p_{u,x}$ =0.92), the chances of getting random number less than this probability is very high. It might not succeed in activating v as its propagation probability is very less 0.01 2.1 (c). Now we push the nodes s and x in the running queue as they are the activated nodes. We process w which has only one inactive neighbor and it may fails to activate v as $p_{w,v}$ =0.07. At this time, susceptible nodes are y,z and t. Nodes y and t gets activated at time $t+2$ because of high propagation probability of ($p_{x,y}$ =0.89) and ($p_{x,t}$ =0.78) but node z may not get activate as its $p_{x,y}$ =0.03 . Node s has no inactive neighbor. Now at time t+2, there is no new nodes to get activated. So the diffusion process stops. To understand and get knowledge of Linear threshold model, it is explained in the next section. But our focus is this research is on Independent cascade model.

2.1.2 Linear Threshold model

In Linear Threshold(LT) model, a node v is influenced by each neighbor w according to a weight $b_{v,w}$ such that $\sum_{w\ neighbors\ of\ v} b_{v,w} \leq 1$. In this model every node u has a random threshold associated with it θ_u from [0,1]. In the case of Linear threshold model, an inactive node gets infected by all of its active infected neighbor if its infected neighbors surpass that threshold.

A node v is infected by its infected neighbors if $\sum_{u \to v, u\ active} p_{u,v} \geq \theta_v$.

For example, figure 2.2 (a) initially node u is active and it infects its neighbor x and u as in our example if the weight of edge (u, x) is greater than threshold of x. Assuming that weight of edge (u, x) is greater than threshold of x so, x will successfully activates, similarly in case of node w as



(a)

(b)

(c)

(d)

Figure 2.1 Illustration of IC model

shown in figure 2.2 (b) but may not able to infect v as $\theta_v \geq p_{u,v}$. At next time step, both x and w infects y as both are infected. They will succeed if threshold of y i.e. θ_y is less than weight of edges (x, y) + (w, y) as shown in figure 2.2 (c). When no new active node exists the process will stop.

2.2 Influence Maximization Problem

Influence maximization problem was introduced in the context of social network by Domingos et al. (2001). Kempe et al. (2003) discussed different diffusion models (in particular independent cascade model, linear threshold model) and proved that the problem of influence maximization is NP-hard. Furthermore, the authors presented the first greedy algorithm for maximization with $(1-1/e)$ approximation guarantee. The guarantee relies on three properties of the influence resulting from diffusion: non-negative, monotonic (i.e. function is either entirely non-increasing, or entirely non-decreasing) and submodular (is a set function whose value, informally, has the property that the difference in the incremental value of the function that a single element makes when added to an input set decreases as the size of the input set increases). Formally, if $\sigma(S)$ is the expected number of nodes influence in the network when diffusion starts at set S then (a) $\sigma(S) \geq 0, \forall S$ (b) $\forall S_1 \subseteq S_2 \Rightarrow \sigma(S_1) \leq \sigma(S_2)$ and (c) $\forall S_1 \subseteq S_2, \forall v \notin S_2 \Rightarrow f(S_1 + v) - f(S_1) \geq f(S_2 + v) - f(S_2)$.

Greedy algorithm due to (Kempe et al. (2003)) starts with an empty seed set S and then it looks for the vertex which has maximum marginal influence spread i.e. $argmax_{v \in V} \sigma_G(S \cup v) - \sigma_G(S)$ and add that vertex to the seed set S. The influence spread is calculated by doing random choices and diffusion process sufficiently many times. It is done by monte carlo simulations for R rounds. We repeat this process until size of seed set is equal the value of k (pre-defined seed set size). In each iteration, this algorithm calculated the influence spread for every node in the graph to calculate the maximum influence spread which increases its computation time. The efficiency of greedy algorithm is a big limitation because we have to calculate the spread on various seed sets.

Several subsequent work focused on efficient implementation of the greedy strategy (Leskovec et al. (2007); Goyal et al. (2011); Chen et al. (2010); Ohsaka et al. (2014); Chen et al. (2009, 2010); Goyal et al. (2011); Jung et al. (2012); Cheng et al. (2013); Galhotra et al. (2016)), some of which do not admit to the same approximation guarantee. Recently, (Borgs et al. (2014)) introduced an efficient technique based on random reachable sets to realize the greedy strategy with approximation guarantees. The technique was further refined and improved by (Tang et al. (2014, 2015)), making influence maximization problem solvable for very large networks.

We will use this technique for efficient implementation of greedy strategy for finding critical nodes. In the following we will present the details of random reachability.

2.2.1 Random Reachability

This algorithm looks at many graphs and the reachable nodes in each graph gives the proportionality to the node influence. In this algorithm for a network $G = (V, E)$ first step is to generate the reverse graph i.e. G^r. Here, $C_G(S)$ are the set of nodes reachable from S in G and $C_G^r(S)$ are the set of nodes reachable from S in G^r.

$$
\begin{aligned}
&\sigma_G(S) \\
&= \sum_{u \in V} Pr(\exists v \in S \ such\ that\ u \in C_G(v)) \\
&= \sum_{u \in V} Pr(\exists v \in S \ such\ that\ v \in C_G^r(u)) \\
&= |V| \times Pr(\exists v \in S \ such\ that\ v \in C_G^r(u))
\end{aligned}
$$

The observation here is that the influence of a set of nodes S is precisely $|V|$ times the probability that a node u, chosen uniformly at random, influences a node from S in the transpose graph G^r.

Given a network G, let G^r is the same network with the edges reversed. A set $RR = \{G_1^r, G_2^r, \ldots, G_N^r\}$ of graphs is constructed as follows. For each G_i^r, randomly pick a node v in G^r and conduct a random walk in G^r (using the edge probabilities) starting from v. Borgs et al. proved that if a vertex v belongs to M number of elements in RR, then expected influence of v can be estimated as $\hat{\sigma}(v) = (M/N) \times |V|$ where $|V|$ is the total number of nodes in the graph and N is the number of reverse reachable graphs. It follows from Chernoff bounds that $\hat{\sigma}$ approximates σ with relative error ϵ when $N = O(|V|/\epsilon^2)$. By the example shown in figure 2.3(a) shows the reverse graph G^r.

In figure 2.3(b) G_1^r random walk starts from node u_3 and it is reachable to node u_2 so here RR_1 $=(u_2,u_3)$. In figure 2.3(c) G_2^r randomly selected node is u_1 and it reaches node u_2 and u_4 here RR_2 $=(u_1,u_2,u_4)$ and in the third graph figure 2.3(c) G_3^r process starts from u_4 which is reachable to node u_2 and $RR_3 =(u_2,u_4)$. Here M value of u_1=1 as it is present only in RR_2, u_2=3 it is present in (RR_1, RR_2, RR_3). Similarly we will calculate the M value of u_3=1, and u_4=2.

We know $\hat{\sigma}(v) = (M/N) \times |V|$. In our example as we created 3 reverse reachable graphs so N is 3.

$\hat{\sigma}(u_1) = (1/3) \times 4$,

$\hat{\sigma}(u_2) = (3/3) \times 4$,

$\hat{\sigma}(u_3) = (1/3) \times 4$,

$\hat{\sigma}(u_4) = (2/3) \times 4$. It gives maximally influential node as u_2. Here we are using $\hat{\sigma}$ instead of σ as $\hat{\sigma}$ approximates the estimated value by ϵ.

2.3 Source Identification

Another important line of work focus on identifying the source(s) of a given diffusion. Source identification is studied in many observations. In the complete observation there are different algorithms that has been proposed. The problem associated is finding single rumor source, local rumor source and multiple rumor source. In the case of snapshot observations where only some of the nodes has been observed different algorithm related to Jordan Center, Dynamic message passing and effective distance based algorithm has been proposed. (Jiang et al. (2018); Zang et al. (2014)) relies on reverse diffusion in the influenced network and classifies the nodes with high centrality as the likely source of diffusion. They studied different observations of network. (Jiang et al. (2018)) studied the time varying social networks and converted time-varying network to series of static networks by introducing time-integrating window. The categories of their observation were Wavefront (only contagious nodes which are going to get infected at time t+1), Snapshot(all nodes in latest time window susceptible, infected or recovered) and Sensor(all the infected nodes with the time when they got infected). Both used reverse propagation algorithms to get the source of the network. However, (Zang et al. (2014)) used snapshot observation and converted multi source locating problem to different single source locating problem. Both used maximum likelihood to determine the real source from the suspects.

Similarly, distance-based measures are used to estimate the likelihood of nodes being sources. Chen et al. (2016) provided different heuristic for clustering and localization technique to get the multiple sources in the graphs and Jorden center method in the case of trees. They expand their research from tree networks to general networks and also finding the size of seed. Shah et al. (2011)introduced the rumor centrality for the maximum likelihood detection and provided a linear time message-passing algorithm to evaluate rumor centrality. They constructed rumor source estimators for general trees, general graphs, and regular trees.

Note that, identifying source nodes, while being an important problem for understanding the reason for diffusion, does not ensure that the removal of source nodes will reduce any subsequent influence (other than the one originating from the source).

2.4 Disrupting Influence

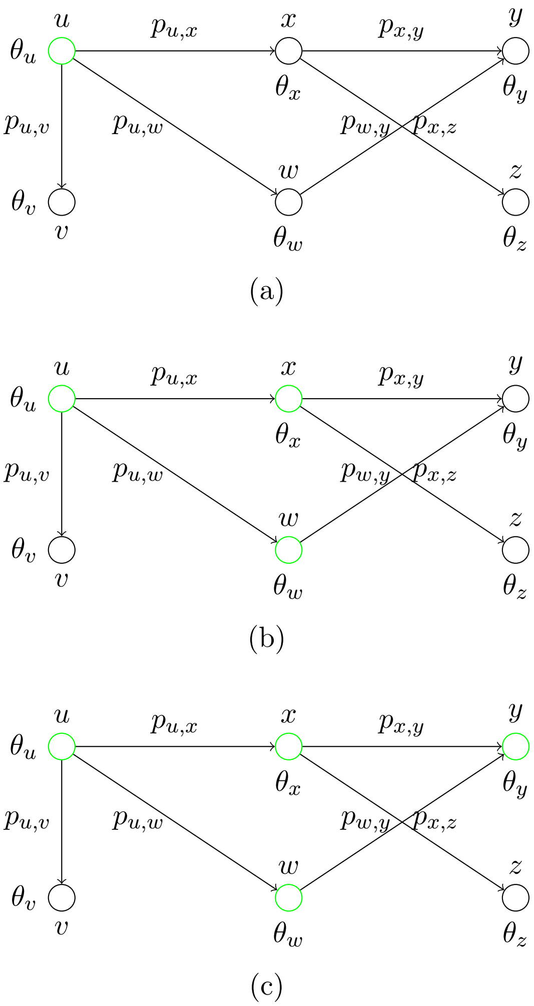
The closest to our work is the work presented by Boutros Khalil et al. (2014). The authors focus on removing edges for disrupting diffusion in linear threshold model. They prove that the function $f(E) = \sum_{v \in V} \sigma_{G/E}(v)$, where E is a set of edges and G/E corresponds to the network G with edges in E removed, is a supermodular function. The objective of disruption is achieved by minimizing f, which involves maximizing the negation of f—negation of f being a submodular function. In short, the critical edge identification problem in linear threshold diffusion model reduces to maximizing a submodular function. In contrast, we will show that the optimization function is neither submodular nor supermodular. So our node removal method is different from their edge removal because both the formalized functions has different properties.

Lappas et al. (2010) introduced the notion of effectors, which are most likely to have caused the influence. They explained their k-effectors problem as finding the subset of active nodes that best explain the observed activation state. Formally, find a set X of active nodes (effectors), of cardinality at most k such that $C(X) = \sum_{v \in V} |a(v) - \alpha(v, X)|$ where $\alpha(v, X)$ is the computation probability that the node v is active at the end of the process. They provided the algorithm of how to extract the most probable active tree that spans all the active nodes in the network. After getting the tree they provided the dynamic programming algorithm to get the effectors. In the event, the entire network is influenced, the problem becomes equivalent to identifying the set of nodes that maximally influences the network. Their work is approximately equal to finding the source of the node as they are looking for the observed activation state and the nodes that caused that observed state. In their case if influence start from the node other than the node which causes the activation state then in that case influence is not maximally disrupted.

The work done by (Aspnes et al. (2005)) focus on identifying the nodes (under some cost constraint) in the network, which when vaccinated (corresponding to removal in our case), will contain the diffusion. Such nodes can be viewed as critical nodes in our setting. The authors present a game-theoretic formulation of the problem, develop a reduction to a graph partitioning problem and provide a poly-time greedy approximation algorithm. However, the authors assumed a simplistic diffusion model, where each active node deterministically activates its susceptible neighbors. This assumption along with the nature of the greedy strategy for partitioning does not make the process a feasible technique in the context of large social networks, where diffusion is probabilistic.

2.5 Our Solution

Our contributions rely on the prior work on influence maximization 2.2 in two dimensions. First, just as maximization of diffusion can be realized using a greedy algorithm, we deploy greedy algorithm (though the same approximation guarantees cannot be achieve as the impact function is neither submodular nor supermodular). Second, we have implemented the greedy algorithm for finding the critical nodes using random reachable sets, a strategy developed by Borgs et al. (2014) for finding most influential nodes. When entire network is influenced our problem is equivalent to source identification problem. Identifying source node is very important part of finding the critical nodes.

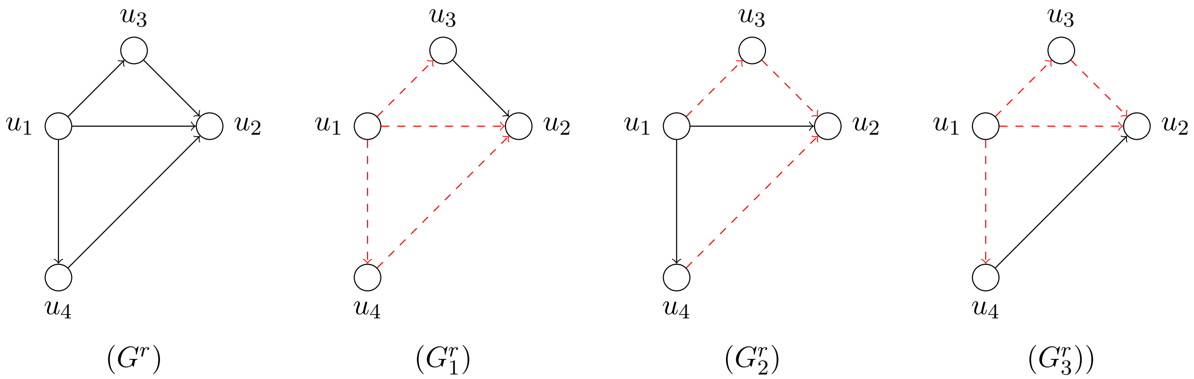


(a)

(b)

(c)

Figure 2.2 Linear Threshold model



(G^r) (G^r_1) (G^r_2) $(G^r_3))$

Figure 2.3 Random Reachability Example

CHAPTER 3. FORMALIZING CRITICALITY

We present some of the basic definitions in the context of information diffusion in network. A network $G = (V, E)$, where V is a finite set of nodes and $E : V \times V \to [0, 1]$ is a directed edge relation between nodes annotated with a probability measure. The direction in the edge $u \xrightarrow{p_{u,v}} v$ indicates the direction of diffusion from u to v and the annotation $p_{u,v}$ indicates the probability (*propagation probability*) of that diffusion. An undirected edge can be viewed as bi-directional with the same propagation probability in both directions. Each node in the network can be in two states: inactive (idle or susceptible) and active (influenced or infected); a node can evolve from being inactive to active and an active node remains active. Such a network forms the basis of several diffusion models. In this work, we concentrate on Independent Cascade (IC) model.

Given a seed $S \subseteq V$ in a network G, $\sigma_G(S)$ denotes the *expected* number of nodes influenced at the end of diffusion (we omit the subscript G, when the network information is immediate in the context). For example, in figure 1.1 $\sigma(v_0)$ is $5 + n$. The problem of influence maximization involves identifying a seed S of a pre-specified size k such that $\sigma_G(S)$ is maximized. The seminal work by (Kempe et al. (2003)) proved that the maximization problem is NP-Hard, and presented a greedy algorithm with $(1 - 1/e)$ approximation guarantee. As noted in Section 1, our objective is to find the critical nodes and such critical nodes may not be the most influential nodes. We formalize the objective, our proposed characterization of the objective followed by the necessary definitions.

3.1 Critical Nodes as Impactful Nodes

Identifying Critical Nodes Problem (ICN) is the problem of finding critical nodes of size k from the graph G such that after removal of these nodes, the influence from any possible seed set in the resulting graph G' is minimized.

Problem 1 (Identifying Critical Nodes Problem (ICN))**.** *Given a network $G = (V, E)$ and k, the ICN(k) involves computing a set of k nodes such that removal of these k nodes from G results in a network $G' = (V', E')$ where $\forall S \subseteq V' : \sigma_G(S) - \sigma_{G'}(S)$ is maximized.* □

The brute force method is a problem solving technique which enumerates over all possible candidates and checks whether the candidate satisfies the problem statement. In case of ICN problem this method is not a feasible option even for small networks. As discussed in the introduction, this notion of criticality is too restrictive and such set of critical nodes may not exist. To address this, we introduce the concept of *impact* of node(s) and claim that impact can be used effectively to compute the criticality of node(s). We first present the notion of *strength of diffusion.*

3.1.1 Strength of Diffusion

Definition 1 (Strength of Diffusion)**.** *Given a network $G = (V, E)$, the strength of diffusion in G, denoted by $\mathcal{ST}(G)$, is $\sum_{v \in V} \sigma_G(v)$.* □

Intuitively, the strength of diffusion indicates sum of the expected number of nodes each node may influence. Thus if the strength of diffusion in a network is high, then it indicates that the network has "many nodes" that can influence a lot of nodes of the network. This can be interpreted as: the network has many good seed sets that can collectively influence a large population of the network. Conversely, if the strength of influence is small, it is an indication that there are no (or very few) seed sets having high influence.

For example, for the graph in Figure 3.1, v_0 is reachable to nodes (v_0, v_1, v_2, v_3, v_4 and $u_1, u_2, ...u_n$) the strength of v_0 is $5+n$. v_2 is reachable to (v_2, v_3, v_4 and $u_1, u_2, ...u_n$) the strength of v_2 is $3+n$. Similarly v_3 and v_4 are also reachable to (v_2, v_3, v_4 and $u_1, u_2, ...u_n$). The strength of v_1 is 1 as it is reachable to just itself. Here $u_1, u_2, ...u_n$ are all reachable to just themselves so their strength is $1 \times n$. So, the total strength of diffusion is

$$(5 + n) + 1 + 3 \times (3 + n) + n = 15 + 5n$$

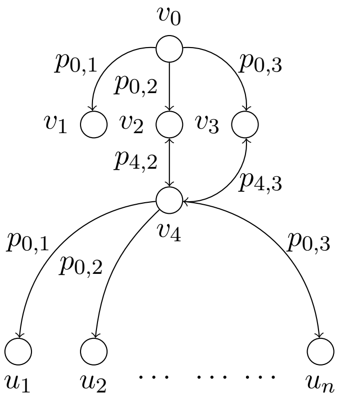


Figure 3.1 Illustrative Example

Thus if removal of a set of nodes from a network causes the strength of diffusion to go down, then it indicates the influence of all (or many) seed sets is also reduced. Thus a set of nodes whose removal will cause maximal reduction in the strength of diffusion can be considered as critical nodes. Based on this, we introduce the impact as follows.

3.1.2 Impact of Node(s)

Definition 2 (Impact of Node(s))**.** *Given a network $G = (V, E)$, the impact of $S \subseteq V$, denoted by $\mathcal{IM}_G(S)$, is $\mathcal{ST}(G) - \mathcal{ST}(G/S)$.* □

The impact, therefore, corresponds to the decrease in the strength of diffusion in the network. Going back to the example in Figure 3.1, after removing the node v_0, $\sigma_G(v_0)$=0, $\sigma_G(v_1)$=1, $\sigma_G(v_2)$=$\sigma_G(v_3)$=$\sigma_G(v_4)$=3+n and $\sigma_G(u_1)$ =$\sigma_G(u_2)$....=$\sigma_G(u_n)$=1

$\mathcal{IM}_G(v_0)$= $\mathcal{ST}(G) - \mathcal{ST}(G/v_0) = 15 + 5n - [1 + 3 \times (3 + n) + n] = 5 + n,$

while in the case when v_4 is removed from the graph $\sigma_G(v_0)$=4, $\sigma_G(v_1)$=1, $\sigma_G(v_2)$=1, $\sigma_G(v_3)$=1, $\sigma_G(v_4)$=0 and $\sigma_G(u_1)$ =$\sigma_G(u_2)$....=$\sigma_G(u_n)$=1

$\mathcal{IM}(\{v4\}) = 15 + 5n - [4 + 3 + n] = 8 + 4n$

We re-formalize the objective in Problem-statement 1 as follows:

Problem 2 (ICN as Identifying Impactful Nodes)**.** *Given a network $G = (V, E)$ and k, the ICN(k) problem involves identifying a set $S \subseteq V$ of size k such that $\mathcal{IM}_G(S)$ is maximized.* □

The reformulation of the ICN problem stems from the following. For any seed, its influence does not increase if some nodes from the network is removed. Larger impact indicates that each node can influence (and can be influenced by) lesser number of nodes. As a result, if S_1 and S_2 are two different sets of nodes such that $\mathcal{IM}_G(S_1) < \mathcal{IM}_G(S_2)$, then the influence of any seed is likely to be less (or equal) when S_2 is removed from G when compared to the case when S_1 is removed.

3.2 Properties of Impact

From Definition 2, one can infer that the $\mathcal{IM}_G(S)$ depends on the expected influence of each vertex v in G, where the diffusion from v occurs via at least one element in S. We will first prove that when all of the edge probabilities are 1, then $\mathcal{IM}_G$, is monotone but is neither submodular nor supermodular. The general case when edge probabilities are not all equal to 1 follows by arguing that as in the work of (Kempe et al. (2003)).

3.2.1 Monotone

Theorem 1. *$\mathcal{IM}_G$ is monotonically increasing.*

Proof. Let S be a set of nodes. Recall that

$$\mathcal{IM}_G(S) = \mathcal{ST}(G) - \mathcal{ST}(G/S) = \sum_{v \in V} \sigma_G(v) - \sum_{v \in V} \sigma_{G/S}(v)$$

When the probabilities are 1, $\sigma_G(v)$ is precisely the number of nodes reachable from v in G. If a node u is reachable from v only via a node from S, then u is not reachable from v in the graph G/S. Thus, $\forall v \in V : \sigma_G(v) - \sigma_{G/S}(v)$ is the number of nodes reachable from v *only* through some nodes in S. Thus

$$\forall S_1, S_2 \subseteq V : S_1 \subseteq S_2 \;\Rightarrow$$

$$\forall v \in V : (\sigma_G(v) - \sigma_{G/S_1}(v)) \leq (\sigma_G(v) - \sigma_{G/S_2}(v))$$

Therefore,

$$\forall S_1, S_2 \subseteq V : S_1 \subseteq S_2 \;\Rightarrow$$
$$\sum_{v \in V}(\sigma_G(v) - \sigma_{G/S_1}(v)) \leq \sum_{v \in V}(\sigma_G(v) - \sigma_{G/S_2}(v))$$

□

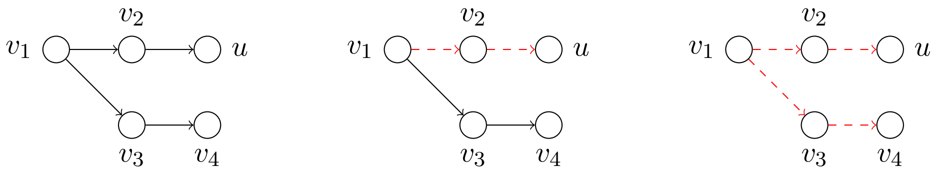


Figure 3.2 Example of monotonically increasing

Here in figure 3.2 $S_1 = (v_2)$ and $S_2 = (v_2, v_3), S_1 \subseteq S_2$

reachable nodes from $v_1 = (v_1, v_2, u, v_3, v_4)$ and set of nodes reachable from v_1 when graph is $(G/S_1) = (v_1, v_3, v_4)$. $\sigma_G(v_1) - \sigma_{G/S_1}(v_1)$ is the number of nodes reachable from v_1 *only* through some nodes in S_1 i.e. $u = 2$. Similarly, set of nodes reachable from v_1 when $G/S_2 = (v_1)$ an empty set which gives set of nodes for $\sigma_G(v_1) - \sigma_{G/S_2}(v_1)$ as (u, v_2, v_3, v_4) =4

$\sigma_G(v_1) - \sigma_{G/S_1}(v_1)) \leq (\sigma_G(v_1) - \sigma_{G/S_2}(v_1)$

In the above example assume that the edge probabilities are all 1. Suppose that is not the case. Consider the sample space in which each sample point is a sub graph of G that is formed as follows: For each edge e, keep in the graph with probability p_e. Suppose that $G_1, G_2, \cdots G_\ell$ are all the sample points in the sample space. Now $\sigma(S)$ is precisely

$$\sum \mathit{Reach}(S, G_i) \times \Pr[G_i]$$

where $\mathit{Reach}(S, G_i)$ denotes the number of nodes reachable from S in the graph G_i, and $\Pr[G_i]$ is the probability that the graph G_i is obtained by the above probabilistic process. Proof of Theorem 1 is showing that $\mathit{Reach}(S, G_i)$ is monotone for every graph G_i and this implies that $\mathcal{IM}_G(S)$ is monotonically increasing.

3.2.2 Not Supermodular

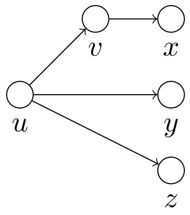
We next establish $\mathcal{IM}_G$ is neither submodular nor supermodular. Submodularity (supermodularity) of a function is defined in terms of the marginal gain for the function. In our context, let S be a set and $v \notin S$ be a node, then the marginal gain in terms of $\mathcal{IM}_G$ is defined as follows:

$$\texttt{imgain}_G(S, v) = \mathcal{IM}_G(S \cup \{v\}) - \mathcal{IM}_G(S)$$

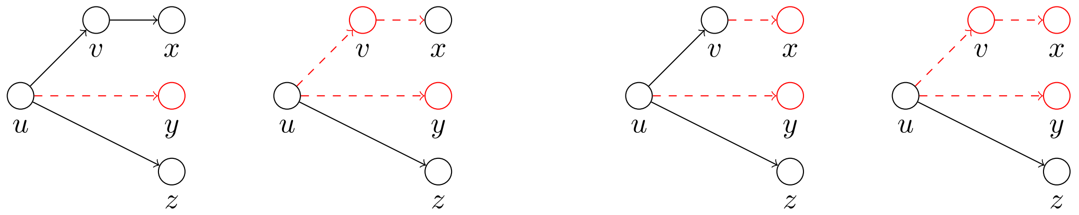
Submodularity of $\mathcal{IM}_G$ requires for all S_1, S_2 and $v \notin S_2$, $S_1 \subseteq S_2$ implies $\texttt{imgain}_G(S_1, v) \geq \texttt{imgain}_G(S_2, v)$. Conversely, for supermodularity, it is required to satisfy $\texttt{imgain}_G(S_1, v) \leq \texttt{imgain}_G(S_2, v)$.

Theorem 2. *$\mathcal{IM}_G$ is not supermodular.*

Proof. Consider the network G where the probability associated with each edge is 1.



(G)



(G/S_1) (G/$S_1 \cup \{v\}$) (G/S_2) (G/$S_2 \cup \{v\}$)

Figure 3.3 Supermodular counter example

For proving our claim, we need to show that there exists S_1, S_2 and v such that $S_1 \subseteq S_2$, $v \notin S_2$ and $\texttt{imgain}_G(S_1, v) > \texttt{imgain}_G(S_2, v)$.

Note that $\mathcal{ST}(G) = 5 + 2 + 3 = 10$. Let S_1 be $\{y\}$, S_2 be $\{x, y\}$. Then,

$$\mathcal{ST}(G/S_1) = 4 + 2 + 2 = 8 \text{ and}$$
$$\mathcal{IM}_G(S_1) = \mathcal{ST}(G) - \mathcal{ST}(G/S_1) = 2$$

$$\mathcal{ST}(G/S_1 \cup \{v\}) = 4 \text{ and } \mathcal{IM}_G(S_1 \cup \{v\}) = 6$$

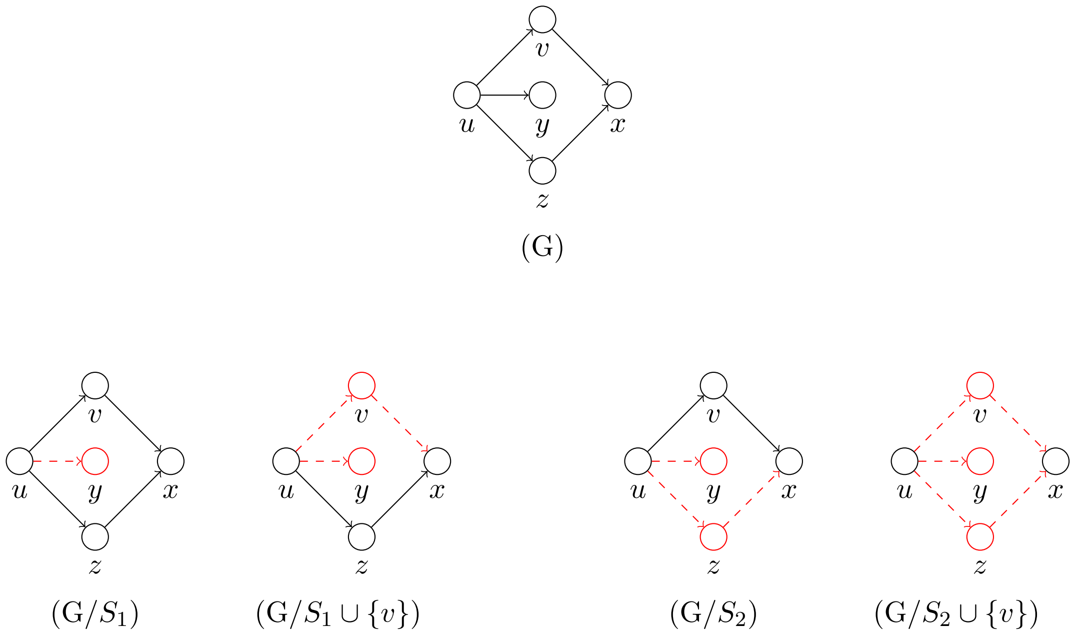
$$\texttt{imgain}_G(S_1, v) = 6 - 2 = 4$$

Proceeding further, $\mathcal{IM}_G(S_2) = 5$ and $\mathcal{IM}_G(S_2 \cup \{v\}) = 7$, and therefore, $\texttt{imgain}_G(S_2, v) = 7 - 5 = 2 < \texttt{imgain}_G(S_1, v)$. □

3.2.3 Not Submodular

Theorem 3. *$\mathcal{IM}_G$ is not submodular.*

Proof. Consider the network G where the probability associated with each edge 1.



(G)

(G/S_1)

(G/$S_1 \cup \{v\}$)

(G/S_2)

(G/$S_2 \cup \{v\}$)

Figure 3.4 Submodular counter example

For proving our claim, we need to show that there exists S_1, S_2 and v such that $S_1 \subseteq S_2$, $v \notin S_2$ and $\texttt{imgain}_G(S_1, v) < \texttt{imgain}_G(S_2, v)$.

Note that, $\mathcal{ST}(G) = 11$. Let S_1 be $\{y\}$ and S_2 be $\{y, z\}$. Then,

$$\texttt{imgain}_G(S_1, v) = \mathcal{IM}_G(S_1 \cup \{v\}) - \mathcal{IM}_G(S_1) = 5 - 2 = 3$$

$$\texttt{imgain}_G(S_2, v) = \mathcal{IM}_G(S_2 \cup \{v\}) - \mathcal{IM}_G(S_2) = 9 - 5 = 4$$

Therefore, $\texttt{imgain}_G(S_1, v) < \texttt{imgain}_G(S_2, v)$.

□

Interestingly, if the network G has the property that there is at most one path between any two nodes, then $\mathcal{IM}_G$ is submodular.

3.2.4 Submodular if there is at most one path

Theorem 4. *$\mathcal{IM}_G$ is submodular if there is at most one path between, any two nodes in G.*

Proof. Again, we assume that all the edge probabilities are 1. The general case follows as per the arguments presented after proof of Theorem 1. We need to prove that for any S_1, S_2 and v,

$$S_1 \subseteq S_2 \ \wedge \ v \notin S_2 \Rightarrow \texttt{imgain}_G(S_1, v) \geq \texttt{imgain}_G(S_2, v)$$

Recall

$$\texttt{imgain}_G(S_1, v) = \mathcal{IM}_G(S_1 \cup \{v\}) - \mathcal{IM}_G(S_1)$$

$$= \mathcal{ST}(G/S_1) - \mathcal{ST}(G/(S_1 \cup \{v\}))$$

That is, $\texttt{imgain}_G(S_1, v)$ is the number of nodes that are reachable from v and are not reachable from S_1. If any of the elements in S_1 can reach v, then $\texttt{imgain}_G(S_1, v) = 0$, as there is at most one path between any two nodes in the network.

Next, for any S_2 such that $S_1 \subseteq S_2$, there are three possibilities in which elements in $S_2 - S_1$ can be selected. (a) there are some elements in $S_2 - S_1$, that can reach v, in which case, $\texttt{imgain}_G(S_2, v) = 0$; (b) None of the elements in $S_2 - S_1$ are reachable from v, in which case, $\texttt{imgain}_G(S_2, v) = \texttt{imgain}_G(S_1, v)$; (c) Some of the elements in $S_2 - S_1$ that are reachable from v, in which case $\texttt{imgain}_G(S_2, v) < \texttt{imgain}_G(S_1, v)$. □

CHAPTER 4. PROPOSED METHOD

Our objective as per Problem statement 2 is to compute a set S of size k such that

$$S^* = \texttt{argmax}_{|S|=k}\ \mathcal{IM}_G(S)$$

It is known that for a large class of monotonic submodular functions, the maximization problem with cardinality constraint is NP-hard. Furthermore, (Yannakakis (1978)) showed that a class of node deletion problems that retains hereditary graph properties is NP-Hard. Our problem falls in such a class. In the context of influence maximization problem, where the influence is monotonic and submodular, (Kempe et al. (2003)) proposed a greedy algorithm with $(1-1/e)$ approximation guarantee. Note that in our problem, we have established that $\mathcal{IM}_G$ is neither submodular nor supermodular. Greedy strategy provides the usual approximation guarantees if the network satisfies the property: any two nodes have at most one path between them ensuring submodulariy of the impact function (see Theorem 4). The greedy strategy is still a viable heuristic even for general network.

4.1 Algorithm for Finding Critical Nodes

Algorithm 1 presents the basic steps necessary to solve $ICN(k)$.

```
input  : Network $G = (V, E)$ and $k$
output: $S \subseteq V$
GreedyImpact
$S = \emptyset$
while $|S| < k$ do
    $w = argmax_{v \in V}\ imgain_G(v, S)$
    $S = S \cup \{w\}$
end
return($S$)
```

Algorithm 1: Greedy Computation of Critical Nodes

The algorithm incrementally computes the set (of size k) of nodes with maximal impact; at each iteration, identifying the node that results in maximal marginal gain in impact with respect to the set computed in the previous iteration.

Note that, the maximal marginal gain computation at each step for each node (yet to be considered in S) is an expensive process. In 2.2.1, the authors presented random reachable set based efficient implementation for computing marginal gains in the context of influence maximization problem. We will employ the same implementation strategy for impact computation. As we already explained in section 2.2.1 about what is Borgs et al. (2014) algorithm of random reachability and expected influence of v can be estimated as $\hat{\sigma}(v) = (M/N) \times |V|$. The marginal gain in influence due to a vertex v with respect to some set S, therefore, can computed by considering the number of RR elements which contains v but none of the elements of S. Incrementally computing marginal gain can be easily realized as follows: at each iteration identify the vertex with maximal coverage of (existing) RR set and remove all the RR elements that vertex covers before proceeding to the next iteration.

In the following, we will present the strategy that we use to compute the marginal gain in impact due to a vertex with respect to a given set using random reachable set.

4.1.1 Impact Computation using Random Reachability

In our context, we need to compute the impact of a set S, which involves computing $\sigma_G(v) - \sigma_{G/S}(v)$ for all nodes v. Let $M_{\overline{S}}$ indicate the number of elements in RR set that contains v such that there is at least one path to v independent of any node in S. Conversely, M_S indicate the number of elements in RR set that contains v such that all paths to v involve some node in S.

Therefore,

$$\begin{aligned}
&\sigma_{G/S}(v)\\
&= \sum_{u \in V} Pr(\exists u : v \text{ influences } u \text{ without involving any } w \in S)\\
&= \sum_{u \in V} Pr(\exists u : u \text{ reaches } v \text{ in } G^r \text{ without involving any } w \in S)\\
&= |V| \times \; Pr(\exists u : u \text{ reaches } v \text{ in } G^r \text{ without involving any } w \in S)
\end{aligned}$$

We know that $\hat{\sigma}_G(v) = |V| \times M/N$ we can write $\hat{\sigma}_{G/S}(v)$ as : $\hat{\sigma}_{G/S}(v) = |V| \times M_{\overline{S}}/N$. Proceeding further,

$$\hat{\sigma}_G(v) - \hat{\sigma}_{G/S}(v) = |V|/N \times (M - M_{\overline{S}}) = |V| \times M_S/N$$

Here, $M - M_{\overline{S}}$ can be written as M_S because when we subtract M (the number of RR sets in whoch node v is present) - $M_{\overline{S}}$ (number of RR sets in which node v is present such that there is at least one path to v independent of any node in S) is equivalent to number RR sets that contains v such that all paths to v involve some node in S. Recall that, $\mathcal{IM}_G(S) = \sum_{v \in V} \sigma_G(v) - \sum_{v \in V} \sigma_{G/S}(v)$. Therefore, $\mathcal{IM}_G(S)$ can be estimated by counting the number of times each node in G is reachable in graphs in RR set where the reachability requires the existence of some node in S.

4.1.2 Incremental Computation of Marginal Gain in Impact

Recall that the marginal gain in impact due to a node v with respect to S is $\texttt{imgain}_G(v, S) = \mathcal{IM}_G(S \cup \{v\}) - \mathcal{IM}_G(S)$. Computing $\mathcal{IM}_G(S)$ involves computing $|V| \times M^u_S/N$ for all $u \in V$ (let M^u_S denote the number of graphs in RR set where reachability of u requires some element in S).

That is,

$$\texttt{imgain}_G(v, S) = |V|/N \sum_{u \in V} \left[M^u_{S \cup \{v\}} - M^u_S \right]$$

Proceeding further, $M^u_{S \cup \{v\}} - M^u_S$ is equal to the difference between number of graphs in RR where reachability of u involves v or some elements in S and number of graphs in RR where reachability of u involves some elements in S. Therefore, $M^u_{S \cup \{v\}} - M^u_S$ is the number of graphs in RR where reachability of u involves v and does not involve any element from S.

Incremental computation of $imgain_G(v, S)$ (and avoid computing $\mathcal{IM}_G(S \cup \{v\})$) is realized as follows. Once $\mathcal{IM}_G(S)$ is computed using RR set, we remove all elements of S from each $G_i^r \in RR$. After removal, $|V| \times M_v^u/N$ for all $u \in V$ is equal to $M_{S\cup\{v\}}^u - M_S^u$, which, in turn, results in incremental computation of $\texttt{imgain}_G(v, S)$.

The greedy algorithm using random reachable sets follows.

```
input  : Network $RR = \{G_1^r, G_2^r, \ldots, G_N^r\}$ and $k$
output: $S \subseteq V$
GreedyImpactRR
$S = \emptyset$
while $|S| < k$ do
    $w = argmax_{v\in V} \ \sum_{u\in V} M_v^u$
    $S = S \cup \{w\}$
    Remove $w$ from $RR$ graphs
end
return($S$)
```

Algorithm 2: Greedy using Random Reachability

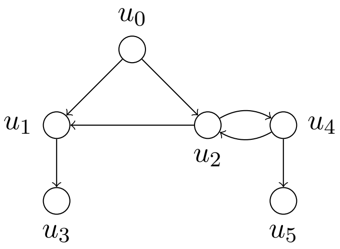
4.1.3 Efficient Implementation of Incremental Computation

Note that, the implementation of incremental computation has two efficiency bottlenecks. First, for the incremental computation one needs to perform reachability on each graphs in RR set in every iteration. Second, it is necessary to store the all graphs in RR set, which can lead to considerable space overhead. To counter these bottlenecks, we develop a data structure that succinctly captures the reachability information in each graphs of RR set and present effective algorithms to construct and maintain the structure. In our research we first used the structure in the form of matrix but we analyzed that it is creating a space overhead so we converted the structure in the form graph i.e. two vertices are connected by edge if source vertex influences target vertex and edge contains the RR set which represents the number of RR sets in which source node influences target node. When we find the maximum influential node we remove that node from this structure i.e. the source with all the edges are removed and we have track of RR set value with which the target node was associated.

For each node $v \in V$ and for each graph G_i^r in RR set, we maintain a set $\texttt{dependOn}(v, i) \subseteq V$. The set contains the nodes such that their reachability requires v in G_i^r. If U is the set of nodes in G_i^r, then $\texttt{dependOn}(v, i)$ can be computed by subtracting from U the nodes that are reachable in G_i^r after removing v. The impact of v proportional to $\sum_{i=1}^{N} \texttt{dependOn}(v, i)$ (equal to $\sum_{u \in V} M_v^u$).

Updating ***dependsOn*** *for Incremental Computation.* In order to facilitate incremental computation of marginal gain of impact, *imgain*, the $\texttt{dependOn}(w, i)$ must be updated for all $w \in V$ and $i \in [1, N]$ once a node $v \neq w$ with the highest impact is selected to be part of the solution. Incrementality requires the removal of v and recomputation of reachability in G_i^r. This repeated reachability can be avoided by the following update operation on $\texttt{dependOn}(w, i)$. If $u \in \texttt{dependOn}(v, i)$ then remove u from all $\texttt{dependOn}(w, i)$ $(w \neq v)$. This is because v in G_i^r impacts u (removing v will make u unreachable in G_i^r); reachability of u cannot be any more falsified (impacted) by further considering w.

This is illustrated in the following example G_i^r.



The corresponding $\texttt{dependOn}$ is represented using as matrix, where the first column represents the input and each cell (r, c) is set to 1, if the r-th element is present in the $\texttt{dependOn}$ of c-th element.

	u_0	u_1	u_2	u_3	u_4	u_5
u_0	1	1	1	1	1	1
u_1		1		1		
u_2			1		1	1
u_3				1		
u_4					1	1
u_5						1

If u_2 is selected as the one with the highest impact[1], then row corresponding to u_2, representing the set `dependOn`(u_2, i), will be rendered unreachable in G_i^r by the removal of u_2. As a result, subsequent computation of impact of nodes u_0, u_1 and u_3 should not consider the unreachable nodes $(u_2, u_4$ and $u_5)$, and hence, their entries (if present) are removed from the `dependsOn` of u_0, u_1 and u_3. This strategy avoids re-computation of impact using reachability.

[1] Note that the above simply illustrates one of the N random graphs in RR. Impact of a node based on the sum of its impact in all the N elements.

CHAPTER 5. RESULTS

1. The primary objective of our experiments is to evaluate the quality of the results obtained by removing critical nodes. We refer to the proposed method as Crit-Set.

2. To measure advantages of using our method, we developed two other methods, which are obvious and immediate choices for disrupting diffusion:

   (a) Top-Infl : one based on removing the top k most influential nodes. Here we used (Tang et al. (2014)) algorithm to get the most influential nodes and

   (b) Top-Crit: one based on removing the top k most critical nodes i.e. without doing further iterations we picked the top k nodes from first iteration of Crit-Set algorithm.

5.1 Experimental Evaluation

In the case of (Top-Infl) and (Top-Crit), due to the submodularity of diffusion function and impact function, respectively, the top k most influential nodes and the top k most critical nodes may not correspond to the most influential or most critical set of size k. As a result, both Top-Infl and Top-Crit are far more time-efficient when compared to Crit-Set, as the latter requires considerably expensive marginal gain computation.

Table 5.1 Dataset

Network-name	# Nodes	# Edges
condensed-Matter-Collab-Network	23,133	93,497
soc-Epinions	75,879	508,837
soc-sign-Epinion	131,828	841,372
com-DBLP	317,080	1,049,866

We will use Top-Infl as the baseline method and show that Top-Crit almost always outperforms both Top-Infl and Crit-Set always outperforms both Top-Infl and Top-Crit. This validates that:

1. most influential nodes are not always the ones that can disrupt diffusion, and

2. the characterization of criticality in terms of impact.

Furthermore, we observe that Top-Crit provides a reasonable balance between quality and cost (in terms of time) and can be an excellent choice for finding critical nodes when the network size is too large (for marginal gain computations as needed in Crit-Set).

5.2 Experimental Setup

5.2.1 Environment

All experiments are conducted on Linux server (Virtual machine) with Red Hat Enterprise Linux 7.x x64 operating system (4 cores) and 16GB main memory. All the algorithms were implemented in C++.

5.2.2 Dataset

We use three networks from http://snap.stanford.edu/data/.

1. Collaboration Network : condensed-Matter-Collab-Network network is from the e-print arXiv and covers scientific collaborations between authors papers submitted to Condense Matter category. The graph contains an undirected edge from i to j, if an author i co-authored a paper with author j. This generates a completely connected (sub)graph on k nodes, if the paper is co-authored by k authors. com-DBLP is the network of DBLP computer science bibliography which provides a comprehensive list of research papers in computer science. A node represents an author here and edge represents a co-author relationship. There is an edge between two authors if they publish at least one paper together. Publication venue, e.g, journal or conference, defines an

individual ground-truth community; authors who published to a certain journal or conference form a community.

2. Trust Networks (soc-Epinions, soc-sign-Epinion): This is a who-trust-whom online social network of a general consumer review site Epinions.com where edge represents trust relationship and a node represent a user.

In all the experiments, following the prior works[1], we chose $p_{uv} = 1/d_{in}(v)$, where $d_{in}(v)$ is the indegree of v. The size of RR is computed based on the chosen $\epsilon = 0.5$. We observe that the quality of the results does not improve much for smaller values of ϵ. Table 5.1 presents the basic information about the networks used in the experiments.

5.3 Criticality-indicator & Importance of Critical Nodes

This set of experiments is directed to validate two claims:

- Removing critical nodes indeed reduces the possible diffusion from any seed.
- Diffusion strength is a good indicator for criticality. That is, nodes that are critical are likely to reduce the strength of diffusion.

For each network, we identified (using Borgs et al. (2014)) the best influential seed set of different sizes. We then use a random diffusion from that seed set to generate the influence graph–the graph where all nodes are influenced. Assuming this influence graph to be the input (that is, the objective is to maximally disrupt diffusion in this influence graph), we conduct experiments to find the impact of removing k nodes in the influence graph.

Table 5.2 presents a subset of results obtained in this experiment. The columns are described as follows. The column infl-size is size of the influence graph generated by seed of size k (second

[1] Probability of diffusion based on indegree is a one of the many ways to quantify the strength of nodes in spreading information—typically, referred to as the *weighted independent cascade model.* Recently, Arora et al. (2017) raised some concerns on whether experiments using weighted cascade model provides validity to the efficiency of the proposed algorithms; however, their work, in particular the evaluation of algorithms, has been seriously refuted by Lu et al. (2017). Our objective is not focused on the debate of how probability of diffusion is measured or quantified and how the efficiency of influence maximization depends on the quantification; rather our focus is to validate our characterization of criticality in terms of impact and not the efficiency of general diffusion problem. In fact, any of the efficient and effective diffusion algorithms can be used in our implementation framework. We chose the basic random reachable set based method, which is at the core of some notable efficient algorithms Tang et al. (2014, 2015).

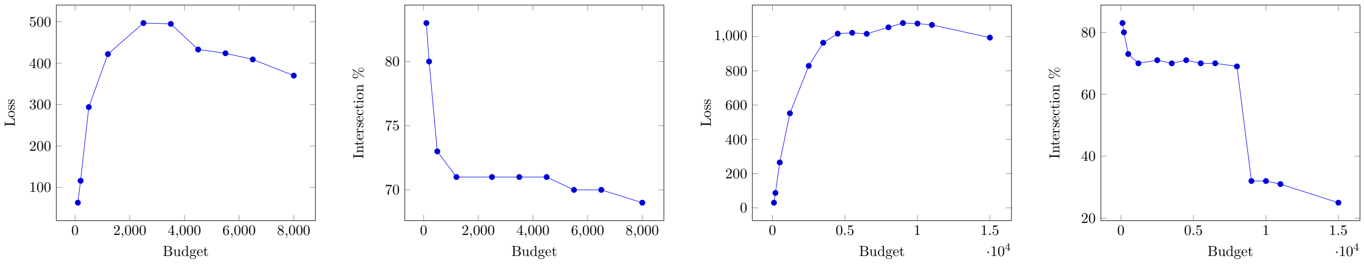
Table 5.2 Criticality-Indicator & Importance

Infl-Size	Seed-Size	Budget	Top-Infl			Top-Crit			Crit-Set		
			Strength	New Infl	Time	Strength	New Infl	Time	Strength	New Infl	Time
Network soc-Epinions											
5126	5	5	1290520	633	0.006	1156417	588 (38%)	0.15	1022415	547 (43%)	0.7
		10	1136926	510	0.007	1152737	405 (16%)	0.17	890993	328 (30%)	0.72
		15	1022866	421	0.004	1143976	327 (19%)	0.17	823687	267 (34%)	0.74
		20	956974	389	0.004	1043167	259 (33%)	0.16	767301	218 (43%)	0.75
7321	10	5	1800003	2097	0.01	1808783	2025 (5%)	0.21	1560588	1974 (8%)	0.97
		10	1742709	1628	0.01	1702486	1337 (16%)	0.23	1361319	1265 (22%)	1.08
		15	1602388	1006	0.01	1476288	833 (15%)	0.23	1263797	765 (23%)	1.05
		20	1491008	731	0.01	1453078	600 (18%)	0.23	1189530	496 (32%)	1.05
soc-sign-Epinion											
9315	10	5	2421060	3134	0.01	2309117	3015 (4%)	0.24	2133981	2857 (9%)	0.98
		10	2182834	1209	0.01	2141671	1033 (11%)	0.24	1882287	871 (25%)	1.03
		15	2044743	1121	0.01	2062708	925 (15%)	0.24	1714198	710 (35%)	1.06
		20	1866644	987	0.01	1993502	811 (17%)	0.24	1604701	629 (34%)	1.09
12967	15	5	3611180	4645	0.02	3378666	4464 (4%)	0.39	3088506	4343 (7%)	1.67
		10	3331051	2921	0.02	3210245	2446 (15%)	0.38	2835161	2290 (20%)	1.71
		15	3026631	2864	0.02	3014840	2604 (8%)	0.39	2654417	2412 (15%)	1.82
		20	2765539	1843	0.02	2972529	1493 (17%)	0.38	2507230	1355 (24%)	1.77
Network com-DBLP											
18298	10	5	6737334	4669	0.03	6638458	4587 (1%)	0.70	6345603	4017 (12%)	3.18
		10	6154251	3209	0.03	6220147	2608 (17%)	0.73	5905621	2807 **(12%)**	3.32
		15	5878247	2349	0.04	5998393	2128 (9%)	0.69	5615215	1898 (19%)	3.34
		20	5643405	2221	0.03	5834640	2066 (7%)	0.70	5325672	1898 (14%)	3.33
29085	20	5	12300875	9963	0.07	11867960	9380 (3%)	1.27	11292619	9698 (6%)	6.01
		10	11568140	9273	0.09	11202940	8276 (11%)	1.29	10586918	8384 (9%)	6.19
		15	11009373	7636	0.061	10681316	6647 (12%)	1.28	10113905	6396 (15%)	6.16
		20	10554641	6955	0.055	10414186	6316 (9%)	1.28	9739477	6116 (12%)	6.21

column). The budget indicates the number of nodes to be removed. The strength columns present the strength of diffusion after the nodes are removed. The method Top-Infl is used as a baseline method; the percentage decrease in the strength using other method Top-Crit and Crit-Set is presented in the respective strength-columns. The New-Infl column indicates the influence in the input graph (after nodes are removed). We use the same size for seed set and construct them by considering the objective of maximizing its influence on $x\%(x \in [20, 90])$ of the network for example x as 20 so in that case by taking 20% of the whole graph we selected the best seed of that 20% graph. We report (New-Infl) the average influence size in the network using these different seeds after the nodes are removed. It also includes the (average) percentage improvement over the baseline Top-Infl method. The timing results are given in seconds.

We have generated different size influence graphs in different types of networks (5k-30k nodes in the influence graph). First observe that, if there is a budget constraint on number of nodes that can be removed this is because we can't remove all the nodes from the graph, then identifying the critical nodes can indeed save majority of the network from un-wanted diffusion. For instance, for com-DBLP network a 20-node seed can influence 29k nodes; however, removing 20 critical nodes help to reduce the result of diffusion (by virtually any 20 nodes) to 6K nodes (a reduction of around 70%). Next observe that, in all experiments Top-Crit and Crit-Set have reduced the level of diffusion more than Top-Infl. This shows that influential nodes are not necessarily the ones that can maximize disruption in diffusion. Furthermore, reduction achieved by Crit-Set is considerable (compared to Top-Infl, in some case as high as 40%).

Our second goal is to validate that characterizing criticality using diffusion strength is appropriate. In other words, reducing strength is likely to reduce influence from any seed. The experiments show that in all cases, removal of critical nodes using Crit-Set reduces the diffusion strength considerably when compared to Top-Infl. That is, reduction in diffusion strength can be used quantify the criticality of nodes.



(a) (b) (c) (d)

Figure 5.1 (a) Influence size difference for seeds of size 300, (b) Intersection size of nodes being removed by Crit-Set and Top-Infl for budget range $[100 - 8,000]$; (c) Influence size difference for seeds of size $1,500$, (d) Intersection size of nodes being removed by Crit-Set and Top-Infl for budget range $[100 - 15,000]$.

5.3.1 Role of Budget on Node Removal

In the last subsection, we validated our claim that node removal based on criticality is an important consideration. Our next set of experiments analyze the relationship between budget (number of nodes to remove) and the node-removal strategy. In particular, we are interested in understanding the difference in quality of results obtained by Crit-Set and Top-Infl as the budget increases.

The setup is as follows. We consider the network condensed-matter-collab-network (see Table 5.1). We find k nodes to remove using Crit-Set and Top-Infl. We record the number of common nodes being removed (intersection size).

After removal of k nodes, we consider M size seed set to start and compute the level of diffusion. Different types of seeds are computed by considering it maximal influence on $x\%$ of the network ($x \in [20\%, 90\%]$). For each seed, the influence size is computed. The average difference between the influence sizes (after removal of nodes using Crit-Set and Top-Infl) is recorded.

Experiment is conducted by varying k starting from 100 for two different values of M equals to 300 and 1500. Figure 5.1(a, b) presents the difference and intersection size against the budget values for $M = 300$. Note that as the budget increases, the difference in the influence size increases rapidly and then plateaus, and finally decreases. On the other hand, as the budget increases, the intersection size of the nodes to be removed by two methods decreases and then flattens. The observations can be explained as follows. For smaller budget the intersection is high because highly critical nodes are also likely to be highly influential nodes. As a result, the difference in the influence size after removal of nodes using the two methods is not large. However, with the increase in the budget, the methods proceed to identify moderately critical nodes (Crit-Set) and moderately influential nodes (Top-Infl)–these sets are not likely to be same/similar. In other words, Crit-Set decides to remove nodes (critical nodes) that are markedly different from the nodes (influential nodes) being removed by Top-Infl. This, coupled with the fact that removal of critical nodes disrupts the diffusion more than the removal of influential ones (as observed in the last subsection), the difference between the influence sizes after removal of influential nodes and after removal of critical nodes increases as the budget increases. The pattern continues up to certain budget after which the nodes to be removed again exhibit the same level of criticality and influence, at which point, the difference between influence size flattens and starts decreasing. This is because all the critical and influential nodes, which have some significant impact on diffusion, have be already considered for removal–increasing budget does not expose any new impactful nodes.

The same pattern is observed when the seed set size is increased to $1,500$ (Figure 5.1(c, d)). The distinguishing aspect is that difference between influence size continues to grow with the budget till a much larger budget value. This is because, as a seed size increases, there are considerably larger number of vulnerable nodes (nodes that can be influenced through diffusion) and as a result, there is a larger number of highly critical nodes that can disrupt the diffusion.

5.3.2 Critical Nodes Removal and Maximal Influence

So far, we have validated that critical nodes play a vital role in disrupting diffusion for a specific influence network as well as for the entire network for different types of seeds. Our final set of experiments focus on validating that maximum influence achievable after removal of nodes following Crit-Set is considerably less than that achievable after removal of nodes following Top-Infl. We considered seed size of size 300 in the condensed-Matter-Collab-Network. We identify the seeds that can induce the maximal diffusion after removal of nodes, and report the number by which diffusion after the critical node removal is less than that after the influential node removal.

Observe that, Crit-Set always outperforms Top-Infl as the budget increases; the difference increases till the budget for removal is $2,000$. This is exactly the same pattern we observed in the last experimental setup; however, there is an important distinction between the two experiments. In the last experiment, the same seed set is used after the node-removal using Crit-Set and Top-Infl; in the current experiment, the best seeds (inducing maximal diffusion) is considered after removal of nodes. As a result, the seeds being considered after removal of nodes using Crit-Set is different from the one being considered after removal of nodes using Top-Infl. The observation validates the claim that the maximal diffusion achievable after critical node removal is less than that achievable after influential node removal; in other words, removing critical nodes disrupts the diffusion possible from the best seeds.

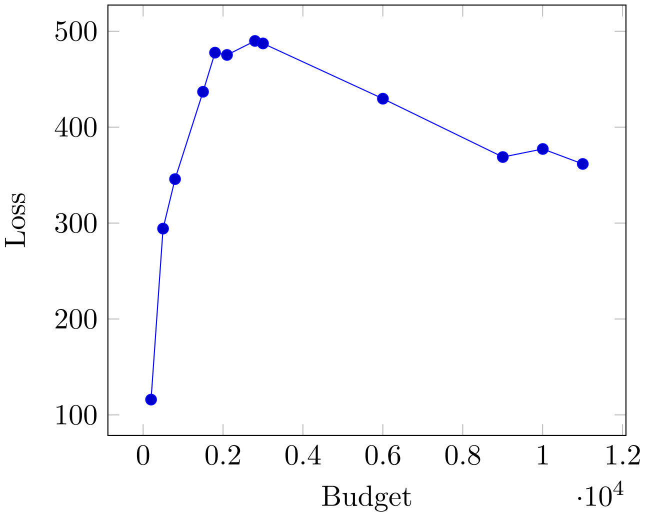


Figure 5.2 Disruption of Diffusion from the Best Seed

CHAPTER 6. CONCLUSION

6.1 Overview of contributions

We study the problem of disrupting influence in social network under independent cascade diffusion model. Our objective is to identify a set of nodes, the critical nodes, which when removed can maximally lower diffusion or maximally disrupt diffusion. We formalize the objective and introduced the characterization of criticality in terms of impact, which, in turn, describes the reduction in the diffusion strength of the network. We present a greedy heuristic for impact computation and further provided the efficient implementation of the algorithm. We design experiments and compared it with different strategies to validate the effectiveness of our characterization in realizing the objective.

6.2 Future Work

1. Different heuristics : As part of future work, we plan to consider different heuristics and implementation strategies to realize the computation of impact; how the quality of result change with the structure of the network.

2. Large Networks: The goal being application to very large networks efficiently without compromising the quality. How the result change when the network is dynamic i.e. continuously changing.

3. Constraints: Another avenue of research along this line of work, includes associating costs and hard constraints on the nodes (e.g., some nodes may not be removed, some nodes may incur prohibitive cost to remove) i.e. constraint that some nodes cannot be removed even if they are critical in that case, how would the solution will change in that case and addressing the problem of constrained cost-effective disruption.

4. Cost function: Associating costs for removing the nodes. So far we have studied the case where the cost for removal is same for all nodes. If different costs are used, then that needs to be factored in in the solution.

REFERENCES

Arora Akhil, Galhotra Sainyam, and Ranu Sayan (2017). Debunking the myths of influence maximization: An in-depth benchmarking study. In *ACM International Conference on Management of Data*, pages 651–666.

Aspnes James, Chang Kevin L., and Yampolskiy Aleksandr (2005). Inoculation strategies for victims of viruses and the sum-of-squares partition problem. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 43–52.

Borgs C., Brautbar M., Chayes J., and Lucier B. (2014). Maximizing Social Influence in Nearly Optimal Time. In *Proc. of the 25th SODA*, pages 946–957.

Chen W., Wang Y., and Yang S. (2009). Efficient influence maximization in social networks. In *Proc. of the 15th KDD*, pages 199–208.

Chen W., Yuan Y., and Zhang L. (2010). Scalable Influence Maximization in Social Networks under the Linear Threshold Model. In *Proc. of the 10th ICDM*, pages 88–97.

Chen W., Yuan Y., and Zhang L. (2010). Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proc. of the 16th ACM SIGKDD*, pages 1029–1038.

Chen Z., Zhu. K. and Ying L. (2016). Detecting Multiple Information Sources in Networks under the SIR Model. *IEEE Transactions on Network Science and Engineering*, 3(1):17–31.

Cheng Suqi, Shen Huawei, Huang Junming, Zhang Guoqing, and Cheng Xueqi (2013). Static-Greedy: Solving the Scalability-accuracy Dilemma in Influence Maximization. In *ACM International Conference on Information & Knowledge Management*, pages 509–518.

Domingos P., and Richardson M. (2001). Mining the network value of customers. In *Proc. of the 7th KDD*, pages 57–56.

Goyal A., Bonchi F., and Lakshmanan L. (2011). A Data-based Approach to Social Influence Maximization. In *Proc. VLDB Endow.*, 5(1):73–84.

Goyal A., Lu W., and Lakshmanan L. (2011). CELF++: Optimizing the Greedy Algorithm for Influence Maximization in Social Networks. In *Proc. of the 20th WWW*, pages 47–48.

Galhotra Sainyam, Arora Akhil, and Roy Shourya (2016). Holistic Influence Maximization: Combining Scalability and Efficiency with Opinion-Aware Models. In *International Conference on Management of Data*, pages 743–758.

He Xinran and Kempe David (2016). Robust Influence Maximization. In *Proceedings of KDD.*

Jiang J., Wen S., Yu S., Xiang Y., and Zhou W. (2018). Rumor Source Identification in Social Networks with Time-Varying Topology. In *IEEE Transactions on Dependable and Secure Computing*, 15(1):166–179.

Jung K., Heo W., and Chen W. (2012). IRIE: Scalable and Robust Influence Maximization in Social Networks. In *12th ICDM 2012.*, pages 918–923.

Kempe D., Kleinberg J., and Tardos E. (2003). Maximizing the Spread of Influence through a Social Network. In *Proc. of the 9th KDD.*, pages 137–146.

Khalil Elias Boutros, Dilkina Bistra, and Song Le (2014). Scalable Diffusion-aware Optimization of Network Topology. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1226–1235.

Kleinberg Jon (2007). *Cascading Behavior in Networks: Algorithmic and Economic Issues,* chapter 24. Cambridge University Press.

Lappas Theodoros, Terzi Evimaria, Gunopulos Dimitrios, and Mannila Heikki (2010). Finding Effectors in Social Networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1059–1068.

Leskovec J., Krause A., Guestrin C., Faloutsos C., VanBriesen J., and Glance N (2007). Cost-effective Outbreak Detection in Networks. In *Proc. of the 13th KDD*, pages 420–429.

Lu Wei, Xiao Xiaokui, Goyal Amit, Huang Keke, and Lakshmanan Laks V. S. (2017). Refutations on "Debunking the Myths of Influence Maximization: An In-Depth Benchmarking Study". *CoRR*, abs/1705.05144, http://arxiv.org/abs/1705.05144.

Ohsaka N., Akiba T., Yoshida Y., and Kawarabayashi K. I. (2014). Fast and accurate influence maximization on large networks with pruned Monte-Carlo simulations. In *Proceedings of the AAAI*, pages 138–144.

Shah D., and Zaman T. (2011). Rumors in a Network: Who's the Culprit? In *IEEE Trans. Inf. Theor.*, 57(8):5163–5181.

Tang Y., Shi Y., and Xiao X. (2014). Influence maximization: near-optimal time complexity meets practical efficiency. In *SIGMOD*, pages 75–86.

Tang Y., Shi Y., and Xiao X. (2015). Influence Maximization in Near-Linear Time: A Martingale Approach. In *SIGMOD*, pages 1539–1554.

Yannakakis Mihalis (1978). Node-and Edge-deletion NP-complete Problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC 78, pages 253–264.

Zang Wenyu, Zhang Peng, Zhou Chuan, and Guo Li (2014). Discovering Multiple Diffusion Source Nodes in Social Networks. In *Procedia Computer Science*, 29:443 –452.

Zang Wenyu, Zhang Peng, Zhou Chuan, and Guo Li (2014). A Note on Influence Maximization in Social Networks from Local to Global and Beyond. In *Procedia Computer Science*, STOC 78, pages 253–264.