Working Paper on CMSC 190

Patricio T. Casurao
Cary M. Quibada

As of September 4, 2018

1 Review on Graph Theory [1]

1.1 Preliminaries

- **set** - a group of objects
- **element** - an object inside a set
- **empty set** - null set;contains no element; denoted by $\varnothing$ or $\{\}$
- If n element a is in set S, then $a \in S$.
- If a set t contains some or all the elements of a set S, then $t \subseteq S$.
- If set S contains all the elements of T and vice-versa, then $t = S$.
- Let A and B be sets,
  - $A \cup B$ or the *union of A and B* is the set containing all the element/s of both set A and set B.
  - $A \cap B$ or the *intersection of A and B* is the set containing the element/s that are present in both set A and set B.
  - set A and set B are disjoint sets if they have no same elements. Disjoint sets can be written as $A \cap B = \varnothing$.
  - $A \setminus B$ or the *set difference of A and B* is the set containing all the elements of A that are not in B.

- $A\Delta B$ or the *symmetric difference of A and B* is equal to $(A \setminus B) \cup (B \setminus A)$.

- **finite set** - a countable set; set that contains countable number elements
- **cardinality of set S or $|S|$** - the number of elements of set S
- **power set of S or $P(S)$** - the set that all the possible subsets of S including the empty set
- Let $k \in \mathbb{N}$ and $a_1, a_2, ..., a_k$ are any k number of objects; *ordered k-tuple or k-tuple* can be written as $(a_1, a_2, ..., a_k)$.
- Some sets of numbers:
  - set of natural numbers or $\mathbb{N} = \{1, 2, 3, ...\}$
  - integer set or $\mathbb{Z} = \{..., -2, -1, 0, 1, 2, ...\}$
  - set of rational numbers or $\mathbb{Q} = \{a/b : a, b \in \mathbb{Z}, b \neq 0\}$
  - set of real numbers or $\mathbb{R}$

1.2 Graphs

- **graph or general graph** - ordered triple $G = (V, E, \phi)$ where, $V \neq \emptyset$, $V \cap E = \emptyset$ and $\phi : E \to P(V)$ such that the cardinality of ϕ is either 1 or 2 for every $e \in E$.
- Let G be a graph with $G = (V, E, \phi)$,
  - The vertex set or V is the set of vertices of G
  - The edge set or E is the set of edges
  - $\phi(e)$ contains the endvertex/endvertices of each e which are the elements of the vertex set.

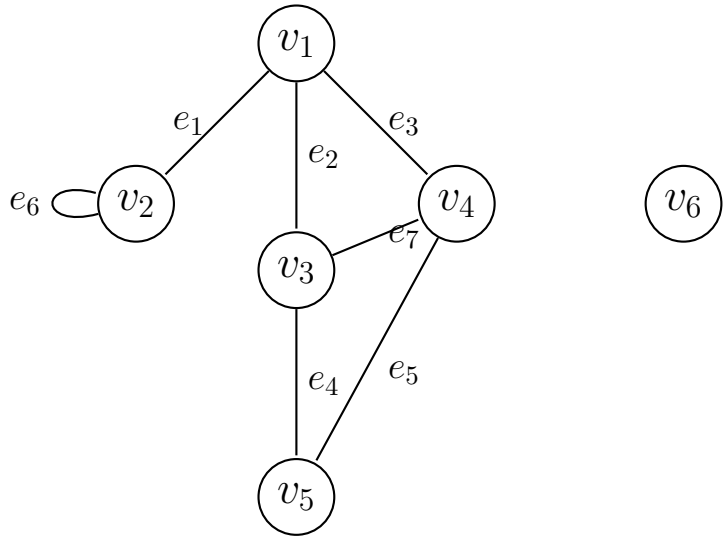

Figure 1: Graph G with 6 vertices and seven edges.

In figure 1,

$$\begin{aligned}
&G = (V, E, \phi)\\
&V = \{v_1, v_2, v_3, v_4, v_5, v_6\}\\
&E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}\\
&\phi(e_1) = \{v_1, v_2\}\\
&\phi(e_2) = \{v_1, v_3\}\\
&\phi(e_3) = \{v_1, v_4\}\\
&\phi(e_4) = \{v_3, v_5\}\\
&\phi(e_5) = \{v_4, v_5\}\\
&\phi(e_6) = \{v_2, v_2\}\\
&\phi(e_7) = \{v_3, v_4\}
\end{aligned} \tag{1}$$

Let the graph $G = (V, E, \phi)$,

- The vertices u and v are in V and are *adjacent/neighbor* of each other , if there is some edge $e \in E$ that has $\phi(e) = \{u, v\}$ or $\{v, u\}$. In figure 1, v_1 and v_2 are neighbors.
- The edges e_1 and e_2 are in E and are *adjacent* if they contain atleast one same endvertex on their respective ϕ, such that $\phi\{e_1\} \cap \phi\{e_2\} \neq 0$. In figure 1, e_1 and e_2 are adjacent
- Vertex v and edge e are in V and E respectively and are *incident*,

if v is in $\phi(e)$. In figure 1, v_1 and e_3 are incident.

- **Loop** is an edge e with the same endvertices given that $|\phi(e)| = 2$. In the figure, e_2 is a loop because $\phi(e) = \{v_2, v_2\}$
- E' is a edge set with *multiple/parallel edges*, if $|E'| \leq 2$ and for any edges e and f in E', $\phi(e) = \phi(f)$
- The vertex v is *isolated* if v is not in $\phi(e)$ for all e in E

- A **simple graph** is a graph without multiple/parallel edges or loops.
- A graph $G = (V, E)$ is a simple graph if,
  1. $V \neq \emptyset$
  2. $E = \emptyset$
  3. $E = \{\{v_1, v_2\} : v_1, v_2 \in V, v_1 \neq v_2\}$

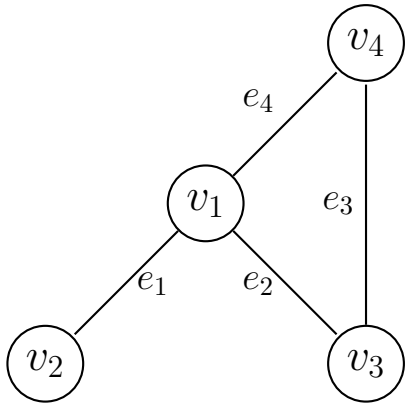

Figure 2: An example of simple graph

Some examples of Common graphs:

- A graph $G = (V, E)$ is a *null graph* on n vertices if,
  1. $V = \{v_1, v_2, ..., v_n\}$
  2. $E = \emptyset$

- A graph $G = (V, E)$ is a *path graph* on $n \geq 2$ vertices if,
  1. $V = \{v_1, v_2, ..., v_n\}$

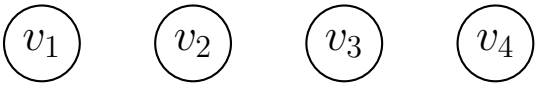

Figure 3: An example of a null graph on 4 vertices

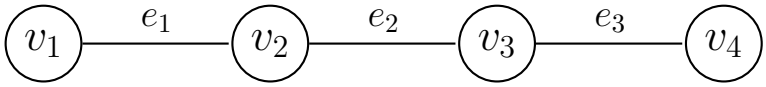

Figure 4: An example of a path graph one 4 vertices

2. $E = \{\{u_1, u_2\}, \{u_2, u_3\}, ..., \{u_{n-1}, u_n\}\}$

- A graph $G = (V, E)$ is a *cycle graph or cycle* on $n \geq 3$ vertices if,
  1. $V = \{v_1, v_2, ..., v_n\}$
  2. $E = \{\{v_1, v_2\}, \{v_2, v_3\}, ..., \{v_{n-1}, v_n\}, \{v_n, v_1\}\}$

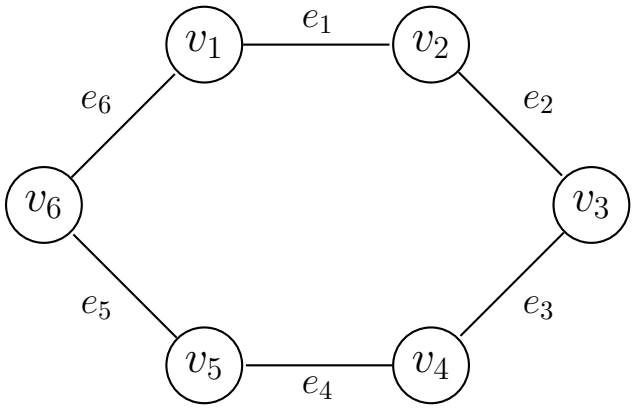

Figure 5: An example of a cycle graph on 6 vertices

- A graph $G = (V, E)$ is a *complete graph* on n vertices if,
  1. $V = \{v_1, v_2, ..., v_n\}$
  2. $E = \{\{v_i, v_j\} : 1 \leq i \leq j \leq n\}$

- A graph is a *complete bipartite (m,n)-graph* on $m + n$ vertices, if

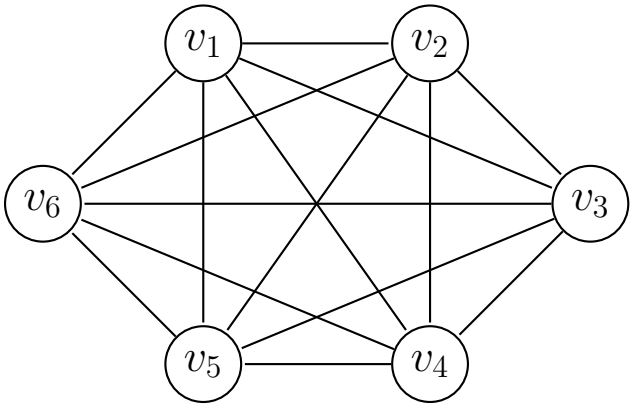

Figure 6: An example of a complete graph on 6 vertices

1. $V = \{s_1, s_2, ..., s_m\} \cup \{v_1, v_2, ..., v_n\}$
2. $E = \{\{s_i, v_j\} : 1 \leq i \leq m, 1 \leq j \leq n\}$

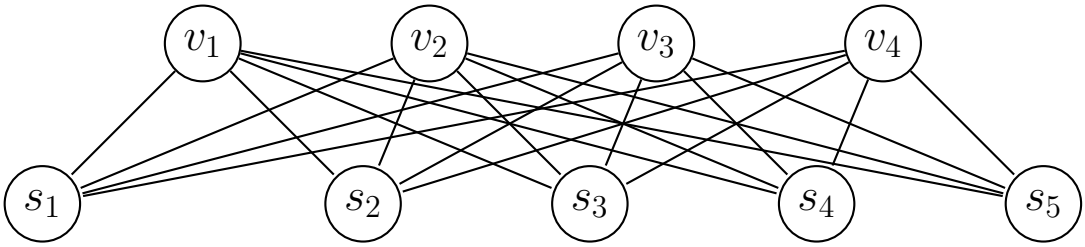

Figure 7: An example of a complete bipartite $(4, 5)$-graph on $4 + 5$ vertices

1.3 Degrees and Regular Graph

Let $G = (V, E, \phi)$ be a graph with vertex $v \in V$,

- $d_G(v)$ or $d(v)$ (when there is no other given graphs) is the *degree of the vertex* v given by,

$$d(v) = |\{e \in E : v \in \phi(e), |\phi(e)| = 1, 2\}|$$

It is the cardinality of edges that are connected to the vertex v.
Some Remarks:

Type of Graph	$d(v)$
Simple	number of neighbors of v
Loopless	number of edges that have v as endvertex
General	same as loopless but loops are counted twice

- The set $N_G(v)$ or $N(v)$ is called the *open neighborhood of* v or the *neighborhood of* v.It is the set of all the vertex neighbors of the vertex v in graph G. It is given by,

$$N(v) = \{w \in V : \phi(e) = \{v, w\} or \{w, v\}, e \in E\}$$

- Additionally, the *closed neighborhood of* v denoted by $N[v]$ is the set of all the elements in the open neighbor with the vertex v itself.

$$N[v] = N(v) \cup v$$

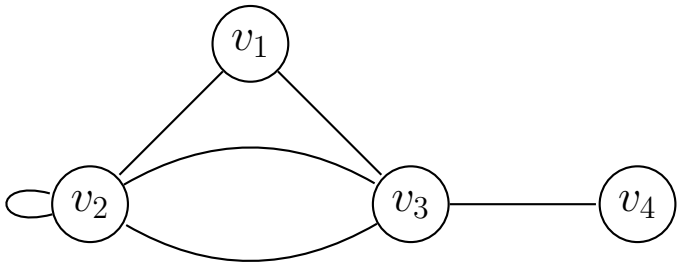

Figure 8: An example of a graph with various degrees.

In figure 8,

$$\begin{array}{cc} d(v_1) = 2 & N(v_1) = \{v_2, v_3\} \\ d(v_2) = 5 & N(v_2) = \{v_1, v_3\} \\ d(v_3) = 4 & N(v_3) = \{v_1, v_2, v_4\} \\ d(v_4) = 1 & N(v_4) = \{v_3\} \\ N[v_1] = N[v_2] = \{v_1, v_2, v_3\} & N[v_3] = V \\ N[v_4] = \{v_3, v_4\} & \end{array} \tag{2}$$

- Hand Shake Theorem
  The *hand shake theorem* was inspired from the hand shaking scenario on a party. Each person in the party needs to shake hands with the other persons in the room. The amount of shake hands made is twice the number of persons in the party. In parallel to graphs, people in the party represents the vertex set of the graph and the shake hands made are the edge set of the graph. The graph $G = (V, E)$ has a total degree of,
  $$\sum_{u \in V} d(v) = 2|E|$$

- The graph $F = (V', E', \phi')$ is a subset of graph $G = (V, E, \phi)$ if,

  1. $V' \subseteq V$,
  2. $E' \subseteq E$,
  3. $\phi'(e) = \phi(e) \forall e \in E'$

  The subgraph relationship is denoted by $\subseteq$. So that, we can say $F \subseteq G$

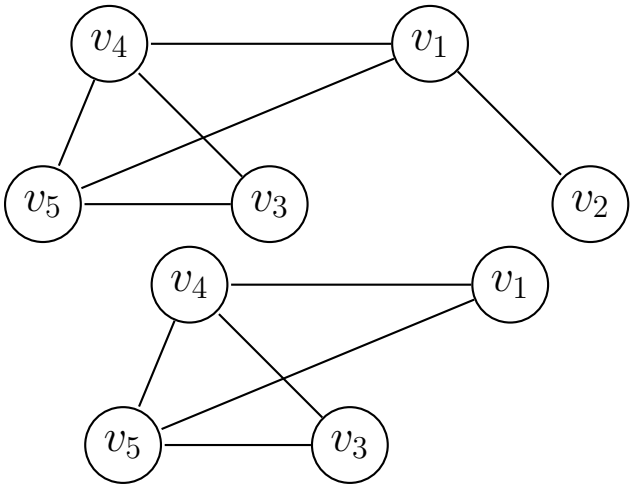

Figure 9: The graph F (bottom) is a subgraph of graph G (top).

1.4 Directed Graphs

- A *directed graph or digraph* is an ordered triple given by $G = (V, E, \eta)$ where,
  1. $V \neq \emptyset$
  2. $V \cap E = \emptyset$
  3. $\eta : E \mapsto V \times V$
  - V is the vertex set
  - E is the set of directed edges
  - If $u, v \in V$,$e \in E$ and $\eta(e) = (u, v)$, then u is the tail of e and v is the head of e.

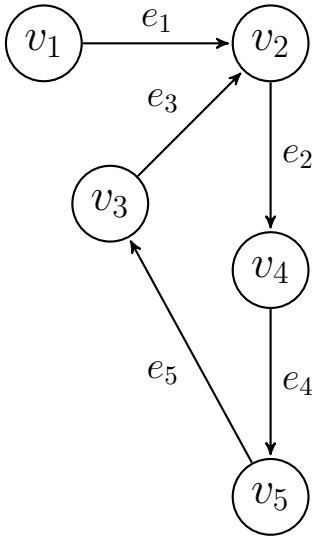

Figure 10: An example of a directed graph.

In figure 10,

$$\begin{aligned} V &= \{v_1, v_2, v_3, v_4, v_5\}, \\ E &= \{e_1, e_2, e_3, e_4, e_5\}, \\ \eta(e_1) &= (v_1, v_2), \\ \eta(e_2) &= (v_2, v_4), \\ \eta(e_3) &= (v_3, v_2), \\ \eta(e_4) &= (v_4, v_5), \\ \eta(e_5) &= (v_5, v_3), \end{aligned} \tag{3}$$

2 Introduction to the Problem

"Viral Marketing" is a term first introduced in 1997 by Steve Jurvetson and Time Draper[2]. It is an effective way of advertising a product to a large group of people by having little groups eventually spread awareness to your product. Companies make a commercial with this in mind. Many over-the-top and absurd commercial having been making rounds across the internet [3]. First, being seen by a few people who were entertained by the commercial enough for them to hit the share button creating a network to other people who will eventually see the commercial shared by their friend and promptly clicking the share button themselves. Thus, repeating the cycle and expanding the network.

Online social networks are a good medium to advertise on. Viral Marketing on online social networks is a very cost-effective technique that most companies today utilize. For only spending a few dollars, the payoff can be amazing.[2] In 1996, this strategy was used by the once popular e-mail service Hotmail: Every mail sent via Hotmail contained advertisement for the service. A total of $50,000 was spent on this campaign. Hotmail expanded into 12 million users in just 18 months.[2]

Recently, the video game company Epic Games, used this tactic to advertise their new battle royale game "Fortnite" and it gave astounding results. The number one video streaming site, Youtube, is a very large network and resource to put to use. They sponsored Youtubers with hundred to millions of subscribers to promote their game.[4] Youtuber Evan Fong "VanossGaming" was one of the content creators Epic Games sponsored for advertising the beta release of Fortnite [5]. He currently has 23 million subscribers(August 2018). The video, titled "Fortnite Funny Moments-Launcing the Rocket!(Gameplay)", has garnered 10 million views in a span of a year.

With this in mind, finding the set of people to target would be ideal. Target Set Selection is basically finding the smallest set of people in a social network to influence all of the people in said network.[6, 7, 8] This network can be represented with a graph G with each person as a node/vertex V. Connecting each node/vertex V is an edge E. Connected vertices are considered neighbors n. The degree $d(v)$ of each node is how many neighbors a node has. The threshold $k(v)$ of each node is a factor to gauge the suceptability of a node from being influenced. The neighbor $N(v)$ is the set of neighbors of the node.

2.1 Two Basic Diffusion Models

Two basic diffusion models have been introduced in 2003 by Kempe, Kleinberd and Tardos.[6] These models are improved upon by works proceeding it. These two models are considered in order to further understand the basics and foundations of the models the algorithm is based/taken from.

In the following models, as information spreads through the network nodes will be referred to as either *active* or *inactive*. The chance of the node to be active increases as more of its neighbor become active. Inactive nodes can become active nodes but the other way around is not to be considered.

2.1.1 Linear Threshold Model

In this model, a node v is influence by each neighbor n according to a $weight\, b_{v,w}$ such that $\sum_{w\, neighbor\, of\, v} b_{v,w} \leq 1$.[6] Each node v is assigned a threshold k uniformly at random from interval [0,1], $[1,\frac{d(v)}{2}]$ or [1,10].[7] The threshold range varies when applied to different variations of social network problem. In this model, the threshold lies in the interval [0,1].

This represents the fraction of neighbors of a node to be active in order for that node to be active itself. A weight is assigned to each of their neighbor. If the cumulative weight of that node's neighbors is the same or more as their threshold, the node itself will become active. Thresholds are selected at random because of the lack of knowledge of the tendencies of the node.[6]

2.1.2 Independent Cascade Model

This model uses probablity instead of thresholds. Nodes have a certain probability pertaining to how a single interaction from an active node to an inactive node can go. The active can either activate its neighbor or not. Once the inactive node is influenced, it can do the same to its neighbors with their own probabilities and the cycle continues for each time step until no more nodes can be activated.[6]

2.2 Other threshold models

We study other threshold models since the model in this study uses thresholds as a way to gauge how easy or how hard it is to influence/activate a certain node. Learning the tendencies and limitations of other thresholds models

will help in understanding the tendencies of the threshold function used in the study. This study has a threshold function that uses real numbers in the set [1,2,...,10].

2.2.1 General Threshold Model

To be able to express the notion that a node v's tendency to be active can be based on an arbitrary monotone function of the set of neighbors of v is a monotone threshold function f_v that maps subsets of v's neighbor set to real numbers in [0,1], with the condition that $f_v(\emptyset) = 0$. The diffusion process follows the general structure of the previously discussed Linear Threshold Model. However, v becomes active in time step if $f_v(S) \geq t_v$, where S is the set of neighbors of v are active in the previous time step.[6]

2.2.2 Majority Thresholds

One important and well-studied threshold model is the majority threshold. This is where a vertex becomes active if at least half of its neighbors are active $[1, \frac{d(v)}{2}]$.[7] This can be applied to networks that have voting systems and use the majority rule.

2.2.3 Small Thresholds

This case is when all the thresholds are small constants. However, this threshold becomes more complex as the threshold increases from $k = 1$. Past researches have studied that, for any $k \leq 2$, the Target Set Selection Problem is NP-hard.A statement constructed by Chen et.al(2009) states that the Target Set Selection problem is NP-hard when the thresholds are at most 2. Earlier before that, Dreyer(2000) proved this instance for only $k \geq 3$.[9] [10]

- P or Polynomial time is a set of languages that are relatively easy to get the solution and to verify it (i.e Multiplication, sorting, finding primes, etc).
- NP or Non-deterministic polynomial time is the set of languages that can verified in polynomial time, or equivalently, that can be solved in polynomial time by a "guessing computer", whose guesses are guaranteed to produce an output of "yes" if at all possible. Basically NP Problems are problems that when given a solution, it is easy to verify the solution, but hard to get the solution itself.

- NP-Complete problems are problems that are in NP and are NP-hard(i.e $n^2 by\ n^2$ sudoku, Boolean Satisfiability Problem).
- A problem is NP-hard if the problem is atleast as hard or even harder than the problems in NP. This means that if you find an algorithm that solves an NP-hard problem, you can use that very algorithm to solve any other problem in NP in polynomial time(i.e Cryptography, Routing, Scheduling).

The Target Set Selection problem is NP-hard when the threshold k for every vertex v is at most 2

For the instance of $k = 1$ it can be solved trivially. Take any graph G and set of all node's thresholds $k = 1$. Activate any of the nodes in that graph and you will create a chain reaction of activating **every** node in that graph. So, with $k = 1$ we can take any one node from that graph that is our target set.[7]

2.2.4 Unanimous Thresholds

This settings is considered the most influence-resistant of the previously mentioned.[7] All the nodes have unanimous thresholds i.e, the threshold of each node is the same as it's degree. This threshold can be applied to an ideal virus-resistant network, where a vertex is infected only if all of its neighbors are infected with the virus. With this in mind, this threshold can be used in constructing robust virus resistant network structures.[7]

2.3 Target Set Selection Algorithms compared in this study

2.3.1 TIP-DECOMP

- d_i^{in}= degree of vertex v_i
- At lines 1-3, a for loop is used for computing the $k_i's$ for each vertex v_i. k_i is defined as $k_i = [t(v_i) \cdot d_i^{in}]$
- At lines 4-6, a for loop is used to compute for the distribution or $dist_i$. This will later be used as distinguishing what the current v_i is to be used in the inner while loop.

Algorithm 1 TIP-DECOMP

Require: Threshold function, t and social network $G = (V, E)$
Ensure: V'

for all *vertex* v_i **do**
    compute k_i
end for
for all *vertex* v_i **do**
    $dist_i = d_i^{in} - k_i$
end for
FLAG=TRUE
while FLAG=TRUE **do**
    for $v_i \in V$ **do**
        if v_i *has* $min(dist_i)$ **then**
            v_i = current v
        end if
    end for
    if $dist_i = \infty$ **then**
        FLAG=FALSE
    else
        Remove v_i from G
        for all $v_j \in n_i^{out}$ **do**
            if $dist_j > 0$ **then**
                $dist_j = dist_j - 1$
            else
                $dist_j = \infty$
            end if
        end for
    end if
end while
return All nodes left in G

- At line 7 a FLAG is instantiated as a boolean variable which will be used in the while loop for identifying the target set selection. We can see it being used in line 8.
- The for loop in line 9-13 is for selecting the vertex v_i where the result of the degree and the threshold is minimal.
- Lines 14-16 are for escaping the main while loop. If this condition is met, it means that the procedure is done.
- The else statement in 16-25 is for removing the vertex where the degree and the threshold is almost the same(line 17).
- The inner for loop in lines 18-24 is used for updating the distributions in the neighborhood of v_j.
- The process returns the reduced vertex set with the vertices removed if necessary. This is the target set.

TIP-DECOMP or Tipping Decomposition is model based on the idea of node "tipping" when a node adopts a property or behavior if a number of his neighbors currently exhibit the same. It is a type of Target Set Selection Algorithm.

The algorithm above inputs a threshold function t and the social network G and outputs the network with vertices removed based off the condition in the algorithm.[11]

2.4 VirAds

- ρ = The influence factor is a decimal between 0-1 that is multiplied to the degree of current vertex v to get $n_v^{(a)}$
- r_v = the round in which the vertex v is activated
- $n_v^{(e)}$ = the number of new active edges after adding v into the seeding
- $n_v^{(a)}$ = the number of extra active neighbors v needs in order for it to activate.
- r_v^i = the number of activated neighbors of v up to round i where i=1..d

Algorithm 2 VirAds Algorithm

Require: Graph $G = (V, E), 0 < \rho < 1, d \in N^+$
Ensure: A small $d - seeding$

$n_v^{(e)} = d(v), n_v^a = \rho \cdot d(v), r_v = d + 1, v \in V$
$r_v^i = 0, i = 0..d, P = \emptyset$
while there exist inactive vertices **do**
  repeat
    $u = argmax_{v \notin P}\{n_v^{(e)} + n_v^{(a)}\}$
    Recompute $n_v^{(e)}$ as the number of new active edges after adding u.
  until $u = argmax_{v \notin P}\{n_v^{(e)} + n_v^a\}$
  $P = P \cup \{u\}$
  Initialize a queue: $Q = \{(u, r_v)\}$
  for all $x \in N(u)$ **do**
    $n_x^{(a)} = max\{n_x^a - 1, 0\}$
  end for
  while $Q \neq \emptyset$ **do**
    $(t, r_t) = Q.pop()$
    for all $w \in N(t)$ **do**
      for all $i = r_t$ *to* $\min\{r_t - 1, r_w - 2\}$ **do**
        $r_w^i = r_w^i + 1$
        if $(r_w^{(i)} \geq \rho \cdot d_w) \wedge (r_w \geq d) \wedge (i + 1 < d)$ **then**
          for all $x \in N(w)$ **do**
            $n_x^{(a)} = \max\{n_x^{(a)} - 1, 0\}$
          end for
          $r_w = i + 1$
          if $w \notin Q$ **then**
            $Q.push(w, r_w)$
          end if
        end if
      end for
    end for
  end while
end while

- Lines 1-2 are for initialization. Line 1 is initialization of variables with values while line 2 is initialization of zero variables.
- Line 3 shows that all inactive vertices will be activated or considered in the algorithm
- Line 4-6 finds the value where the maximum occurs in the sum of the degree of v and it's effectiveness.
- u is added to P in line 7
- A queue is initialized by adding the vertex and its corresponding round, which at the start is zero (u, r_v)
- Lines 9-11 updates the neighbors of u, reducing their degree by 1. If less than zero is reached, it goes back to zero.
- At the start of the while loop in line 13 (t, r_t) is set as the head of the queue. This is obtained by the pop function.
- For all neighbors w of the current node t, another for loop is used for checking if the neighbors of t have reached the threshold defined by $\rho \cdot d_w$ AND $round\ r_w \geq the\ max\ round\ d$ AND the index i of the for loop doesn't exceed the max rounds d.
- This then leads to the thresholds of the neighbors of the that neighbor w being decremented.
- In line 21 the round of the neighbor w is incremented and if that current $w \notin Q$ then it is added to the queue and the while loop starts again.

VirAds selects in each step the vertex u with the highest effectiveness which is defined as $n_u^{(e)} + n_a^{(e)}$. Which is basically the vertex v which can activate the most number of *edges*. It also considers the vertex with the most active neighbors. After that, the algorithm updates the measures for all the remaining vertices. VirAds will make the selection based on the information within d-hop neighbor around the considered vertices rather than only one-hop neighbor as in the degree-based heuristic. When a vertex u is selected, it causes a chain-reaction and activate a sequence of vertices or lower the rounds in which vertices are activated.[12]

The algorithm utilizes a queue. A queue is a data structure which is FIFO. Imagine, a line at any commercial establishment where the first who came is the first who's served.

3 Main algortihm developed in the study

3.1 Preliminaries

Let $G = (V, E)$ be a graph modeling a network. Γ is the neighborhood of current vertex v and $d_g(v)$ or $|\Gamma_g(v)|$ as the degree of current vertex v. Let $t : V \to N_0 = 0, 1, ...$ be the threshold function assigned to the vertices of G. For each node $v \in V$, the threshold $t(v)$ indicates how hard or how easy it is to influence vertex v. Low thresholds means easier to influence and high thresholds are hard to influence.

The activation process in G starting at S is a sequence of vertex subsets $Active_G[S, 0] \subseteq Active_G[S, 1] \subseteq ... \subseteq Active_G[S, \ell] \subseteq ... \subseteq V$ of vertex subsets, with $Active_G[S, 0] = S$ and

$Active_G[S, \ell] = Active_G[S, \ell-1] \cup u : |\Gamma_G(u) \cap Active_G[S, l-1]| \geq t(u), for\ \ell \geq 1$

A target set for G is set $S \subseteq V$ such that $Active_G[S, \lambda] = V$ for some $\lambda \geq 0$

This means at each round ℓ the set of active nodes is augmented by the set of nodes u that have anumber of already activated neighbors greater or equal to u's threshold $t(u)$. The vertex v is said to be activated at round $\ell \geq 0$ if $v \in Active_G[S, \ell]\ Active_G[S, \ell - 1]$.

3.2 The TSS Algorithm

Given the input graph G, at each round/iteration if the vertex has already been considered it will be removed from the graph. When the current vertex v is deleted in the graph its neighbors degree is updated and the current v is removed from the neighbor set of the neighbors of v.

If the vertex v is deemed applicable to be in the target set, the thresholds of their neighbors are decreased by one since the deleted node is able to activate them.

In this algorithm 3 cases are considered:

- Case 1: The threshold k of the current vertex v is 0.
  The vertex v is activated by the influence of its neighbors in $V - U$ only; it can then influence its neighbors in U.
  So, for each neighbor of vertex v denoted by u, reduce the threshold of u by 1.

- Case 2: The degree of v is less than the its threshold.
  The vertex v is added to S, since no sufficient neighbors remain in U to activate it; v can then influence its neighbors in U.
  Add the current v to S For each neighbor of vertex v denoted by u, reduce the threshold of u by 1.

- Case 3: The other conditions are not met.
  The vertex v will be influence by some of its neighbors in U.
  Set the vertex v as the vertex in which this equation $\frac{k(u)}{\delta(u)(\delta(u)+1)}$.
  Note: the maximum value that this equation bears when $k(u) = \delta(u) = 1$. When 1 is substituted to the equation $\frac{1}{2}$ is obtained. With the threshold and degree of a vertex = 1, the node is a leaf. This implies that for every change of v, the ones with low degrees and threshold are considered first.

3.2.1 Pseudocode of TSS Algorithm

Definition of Variables

- S: target Set
- $d(v)$: degree of vertex v
- $t(v)$: threshold of vertex v
- $Gamma(v)$: neighbor set of vertex v

Algorithm 3 TSS Algorithm

Require: $G = (V, E)$, Thresholds $t(v)$ for $v \in V$
Ensure: Target Set S

$S = \emptyset$
$U = V$
for all $v \in v$ **do**
    $\delta(v) = d(v)$
    $k(v) = t(v)$
    $N(v) = \Gamma(v)$
end for
while $U \neq \emptyset$ **do**
    if there exists $v \in U$ $s.t$ $k(v) = 0$ **then**
        for all $u \in Nv$ **do**
            $k(u) = max(k(u) - 1, 0)$
            $S = S \cup \{v\}$
        end for
    else
        if there exists $v \in U$ $s.t$ $\delta(v) \leq k(v)$ **then**
            for all $u \in N(v)$ **do**
                $k(u) = k(u) - 1$
            end for
        else
            $v = argmax_{u \in U}\{\frac{k(u)}{\delta(u)(\delta(u)+1)}\}$
        end if
    end if
    for all $u \in N(v)$ **do**
        $\delta(u) = \delta(u) - 1$
        $N(u) = N(u) - v$
    end for
    $U = U - v$
end while

References

[1] G. Agnarsson and R. Greenlaw, *Graph Theory: Modeling, Applications, and Algorithms*. Prentice-Hall, Inc., 2006.

[2] S. Jurvetson, "What exactly is viral marketing," *Red Herring*, vol. 78, pp. 110–112, 2000.

[3] R. J. Larson, "The rise of viral marketing through the new media of social media," 2009.

[4] M. Abuljadail, N. C. Bi, A. Fisher, C. Y. Joa, K. Kim, X. Wen, and F. R. Zhang, *The Audience and Business of YouTube and Online Videos*. Rowman & Littlefield, 2018.

[5] L. Ha, "Most popular youtube channels," *The Audience and Business of YouTube and Online Videos*, p. 135, 2018.

[6] 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

[7] N. Chen, "On the approximability of influence in social networks," *SIAM Journal on Discrete Mathematics*, vol. 23, no. 3, pp. 1400–1415, 2009.

[8] P. Shakarian and D. Paulo, "Large social networks can be targeted for viral marketing with small seed sets," in *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, ser. ASONAM '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/ASONAM.2012.11

[9] P. A. Dreyer, "Applications and variations of domination in graphs," Ph.D. dissertation, Citeseer, 2000.

[10] O. Chartrand, *Applied and Algorithmic Graph Theory*, 1993.

[11] P. Shakarian and D. Paulo, "Large social networks can be targeted for viral marketing with small seed sets," in *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*. IEEE Computer Society, 2012, pp. 1–8.

[12] T. N. Dinh, H. Zhang, D. T. Nguyen, and M. T. Thai, "Cost-effective viral marketing for time-critical campaigns in large-scale social networks," *IEEE/ACM Transactions on Networking (ToN)*, vol. 22, no. 6, pp. 2001–2011, 2014.