Maximizing the Spread of Influence through a Social Network

David Kempe, Jon Kleinberg and Eva Tardos

Group 9

Lauren Thomas, Ryan Lieblein, Joshua Hammock and Mary Hanvey

Introduction

In a social network, information diffusion occurs.

Information diffusion is the “word of mouth effect”.

For example:

Your friend visits a great restaurant and posts about it on facebook. Other friends go to the restaurant and post.

Introduction

This is most useful in marketing; creates “viral marketing”.

Viral marketing:

- want to pick influential members
- if you pick the right members who recommend the restaurant, more customers should come
- how do you pick the most influential members?
- what are some things to take into consideration?

Problem

We want to choose the most influential individuals.

We only know the social network structure.

It will be an NP-hard problem.

Basic Assumptions

Each node will be either active or inactive.

- Active- has adopted
- inactive- has not adopted

Tendency to become active increases monotonically, as more neighbors become active.

Assume nodes can only go from inactive -> active and not inactive->active->inactive.

Linear Threshold Diffusion Model

Granovetter and Schelling were among the first to propose.

Based on node-specific thresholds:

Each node, v, is influenced by neighbor, w, based on a weight, $b_{v,w}$ such that:

$$\sum_{w \text{ neighbor of } v} b_{v,w} \leq 1 .$$

Linear Threshold Diffusion Model

Process:

- each node will chose a threshold, θ_v , at random from interval [0,1]
- θ_v represents the fraction of v's neighbors that must become active in order to activate v

given initial set of active nodes (A_0), the following steps occur:

Linear Threshold Diffusion Method

step t:

- all nodes that were active in step t-1, remain active.
- activate node v for which the total weight of its active neighbors is
- θ_v:

$$\sum_{w \text{ active neighbor of } v} b_{v,w} \geq \theta_v.$$

basically any node whose neighbors weights equal its active threshold

Independent Cascade Diffusion Model

Again we start with A_0 active nodes.

Unfolds in discrete steps:

- when node v first becomes active in step t, it is given a single chance to activate its inactive neighbors.
- It succeeds with probability, $p_{v, w}$, a parameter of the system.
- if v succeeds, in the following steps, w will be activated
- if v fails, it cannot active w again in future steps.

Process stops when no more activations are possible

Approximation Guarantees

Take an arbitrary function f(·) that maps subsets of a finite ground set, U, to non-negative, real numbers.

f(·) is submodular if the marginal gain from adding an element to a set S is at least as high the marginal gain from adding the element to a superset of S.

$$f(S \cup \{v\}) - f(S) \quad \geq \quad f(T \cup \{v\}) - f(T)$$

Approximation Guarantees

We use this submodular function f(·) because submodular functions have unique, helpful properties.

Specifically, suppose the function f is submodular, takes only non-negative values and is monotone (adding an element to a set cannot cause f to decrease, or $f(S \cup \{v\}) \geq f(S)$, for all elements v and sets S).

To find a k-element set S for which f(S) can be maximized, the following algorithm approximates the solution within (1-1/e).

Approximation Guarantees

The algorithm is as follows:

- start with an empty set
- repeatedly add an element that gives that maximum marginal gain

Approximation Guarantees

All of this can be summarized as:

For a non-negative, monotone submodular function f, let S be a set of size k obtained by selecting elements one at a time, each time choosing an element that provides the largest marginal increase in the function value. Let $S*$ be a set that maximizes the value of f over all k-element sets. Then $f(S) \geq (1-1/e) \cdot f(S*)$; in other words, S provides a $(1-1/e)$- approximation.

Thus to come up with an approximation guarantee for the problem, we must show the function $\sigma(\cdot)$, to find the influence, is submodular.

Approximation Guarantees- Independent Cascade

The approximation guarantee for the independent cascade model and the linear threshold model is (1-1/e)

Experiments

Co-authorship Networks

- Network of papers with multiple authors
- Data compiled from complete list of papers in high energy physics theory sections of the e-print arXiv
- Node assigned to each researcher who has at least 1 co-authored paper
- Edge for each pair of authors who wrote papers with 2+ authors
    - single author papers are eliminated in this case
    - multiple parallel edges are kept to indicate stronger social ties
- Results: 10,748 nodes; edges between about 53k pairs of nodes

Experiments

Types of Algorithms For Each Influence Model

1. Greedy
    - Making locally optimal choice at each stage
    - Hopeful result - finding a global optimum

2. High-Degree Heuristic

    - Choose nodes, v, in order of decreasing d_v
    - Considering high-degree nodes as most influential

Experiments

Types of Algorithms For Each Influence Model

3. Distance Centrality
   - Node with shortest path distance to another node is more influential
   - Distance ‘n’ is distance between any pair of unconnected nodes which accrues an infinite distance

4. Random (baseline algorithm)
   - Random process is simulated 10,000 times for each set, re-choosing thresholds or edge outcomes randomly from [0,1] every time
   - Least accurate approximation

Three Influence Models

1. Linear Threshold model

    - Parallel edges (authors have multiple papers together) are kept
        - Treat the multiplicity of edges as weights
    - Nodes u,v have $c_{u,v}$ parallel edges between them, and degrees d_u, d_v, then edge (u,v) has weight $c_{u,v} / d_v$ and edge (v,u) has weight $c_{v,u} / d_u$

Three Influence Models

2. Independent Cascade Model

   - uniform probability ‘p’ assigned to each edge
   - p = 1% and 10% in separate trials
   - For each edge, already activated u has a chance of p(1-10%) of activating v
   - Therefore, total probability = $1-(1-p)^{c_{u,v}}$
   - Property
     - high-degree nodes not only have a chance to influence many other nodes, but also to be influenced by them

Three Influence Models

3. Weighted Cascade Model

    - Each node from node u to v is assigned probability $1/d_v$ of activating v
    - Similar to Linear Threshold model in that the expected number of neighbors who would succeed in activating a node v is 1 in both models

Results from Experiment (Linear Threshold model)

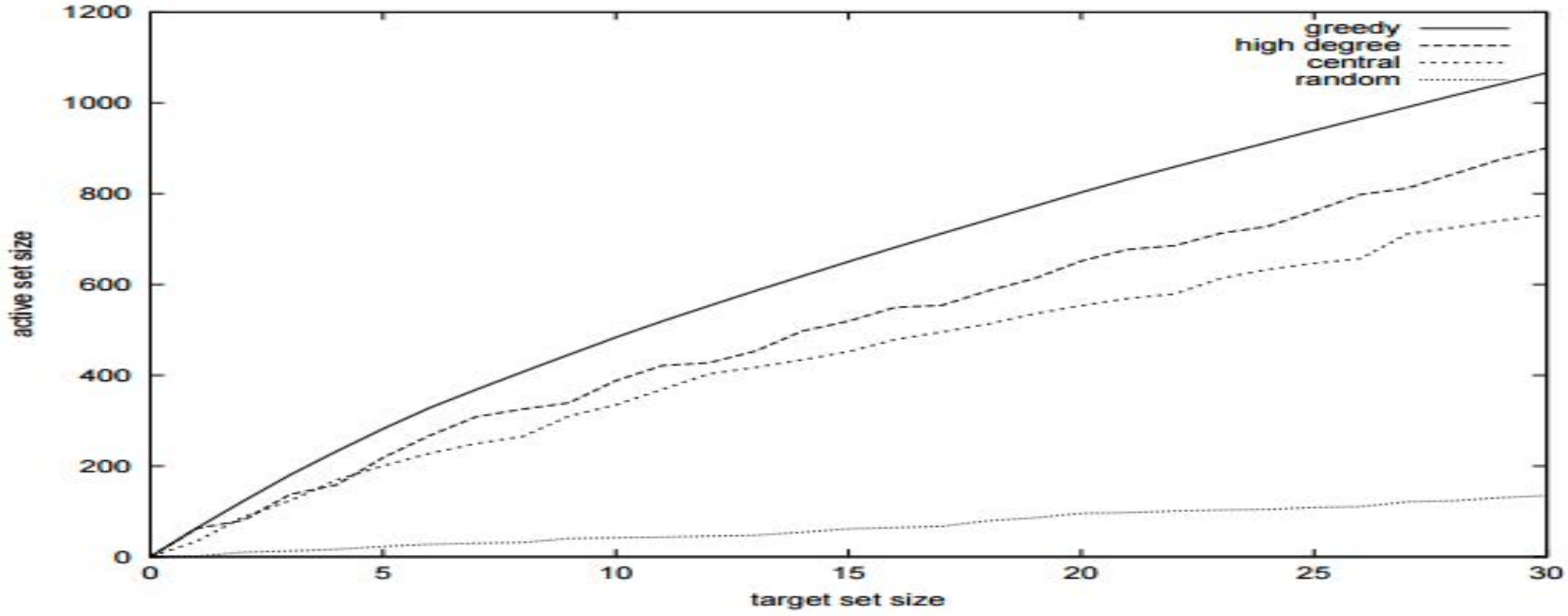

Figure 1: Results for the linear threshold model

Results from Experiment (Linear Threshold model)

- Results from Previous Slide
  - Greedy Algorithm
    - outperforms the high-degree heuristic by about 18%
    - outperforms distance centrality by over 40%
    - Better marketing results can be obtained by considering dynamics of info in a network rather than relying only on structural properties of the graph
  - Failure of degree/distance algorithms
    - Ignored fact that many central nodes may be clustered, so targeting all of them is unnecessary
    - Uneven curves on graph suggest the network influence of many nodes is not accurately reflected by their degree or distance

Results from Experiment (Weighted Cascade model)

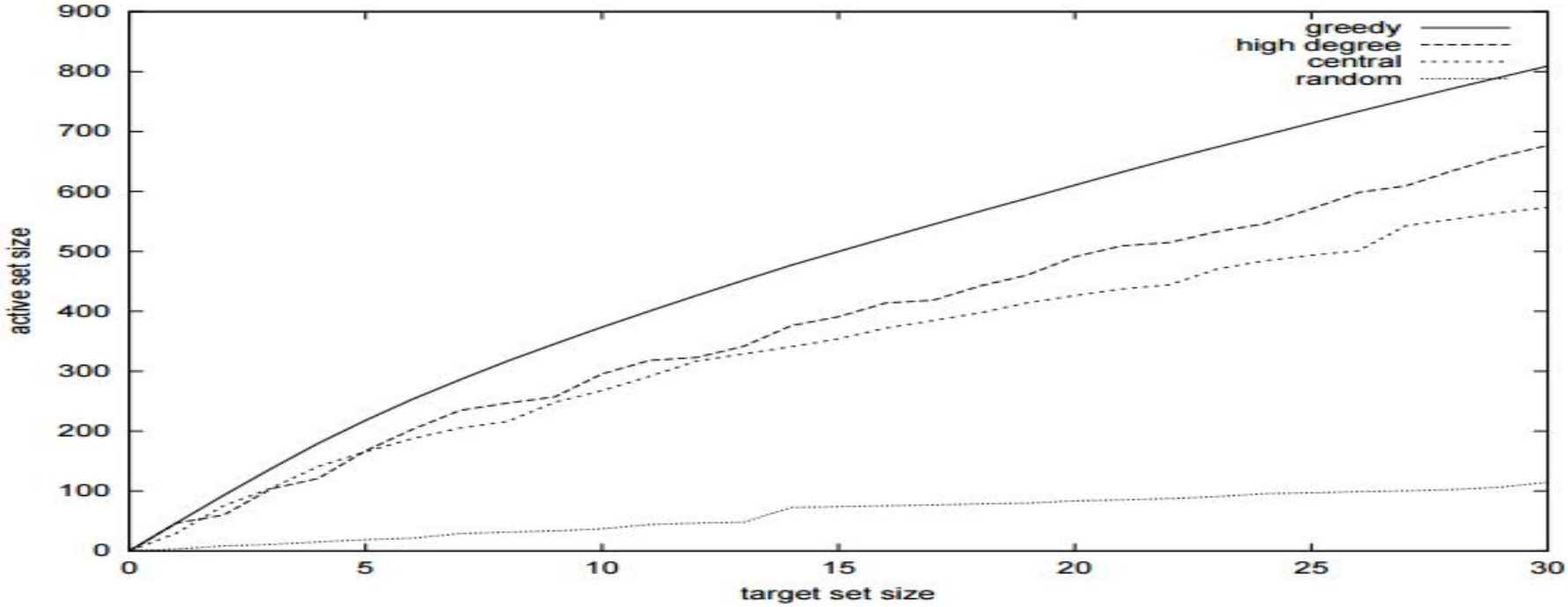

Figure 2: Results for the weighted cascade model

Results from Experiment (Weighted Cascade model)

- Results from Previous Slide
  - Extremely similar to Linear Threshold model as stated previously
    - However scales are about 25% smaller but qualitatively are the same
  - Each node is influenced by the same number of other nodes
  - The degree is relatively concentrated around 1
  - Rely on low-degree nodes as multipliers

Results from Experiment (Independent Cascade 1%)

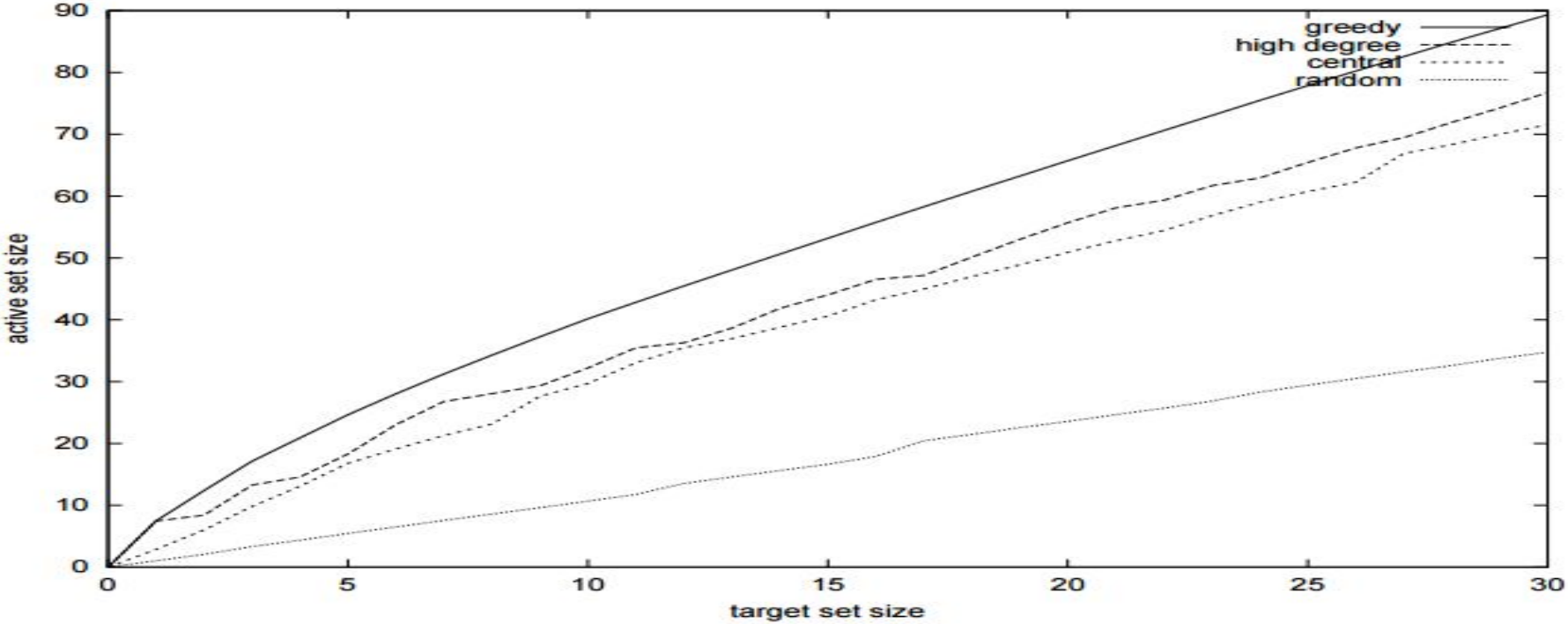

Figure 3: Independent cascade model with probability 1%

Results from Experiment (Independent Cascade 1%)

- Results from Previous Slide
  - On average, each targeted node only activates 3 additional nodes
    - Small probabilities are less effective in this case
  - Randomized selection accrued way better results in independent cascade than in any other model
  - Greedy algorithm is again the most effective followed by degree then distance

Results from Experiment (Independent Cascade 10%)

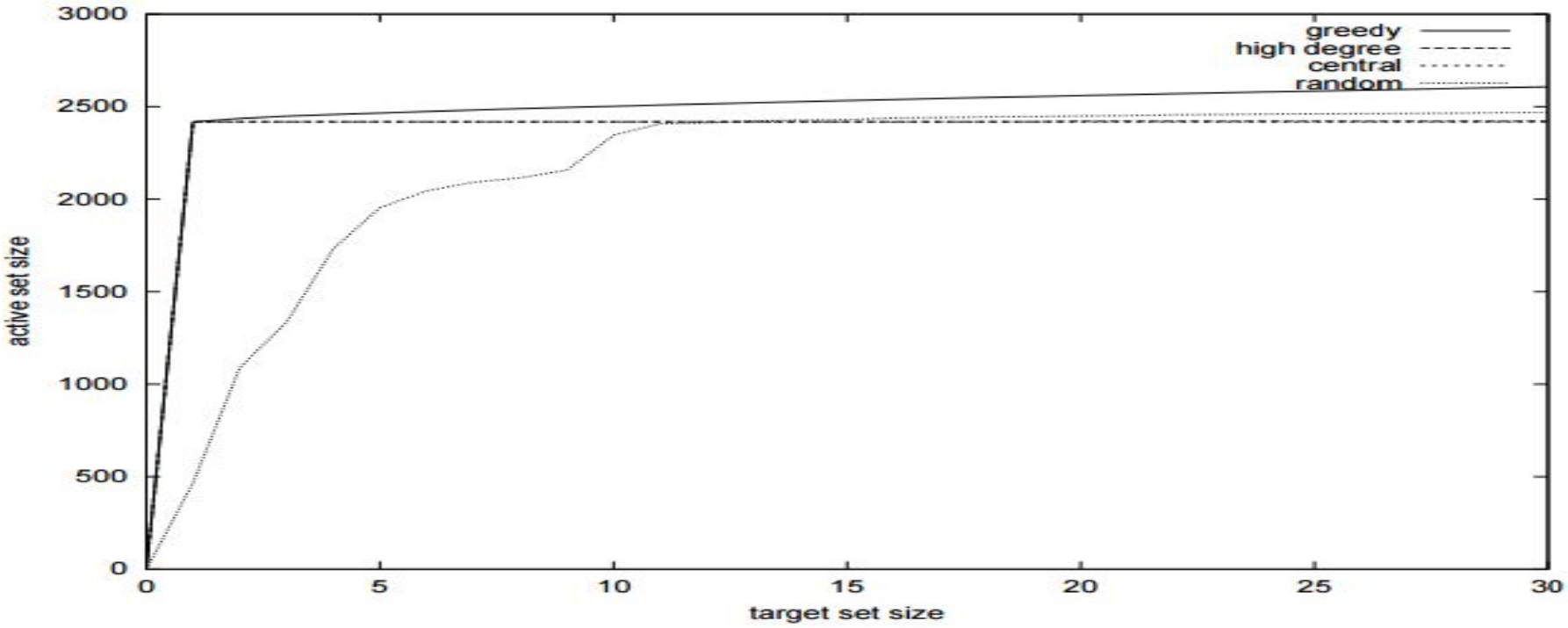

Figure 4: Independent cascade model with probability 10%

Results from Experiment (Independent Cascade 10%)

- Results from Previous Slide
    - Random nodes outperform degree and distance nodes when more than 12 nodes are targeted
        - This is because the first targeted node will activate about 25% of network
        - Additional nodes only reach a small additional fraction of the network
    - Distance and Degree nodes are very likely to be activated by the initially chosen node, which explains the shape of this graph
    - The greedy algorithm takes the same effect of the first activated node but also targets nodes with smaller marginal gain afterwards which is why it doesn't stop growing

General Threshold Model

- The model shows that a node's decision to become active can be based on an arbitrary monotone function of its neighbors.
  - The corresponding arbitrary monotone function (f_v) assigns the subsets of the neighbors to real numbers within [0,1].
  - The condition $f_v(0)=0$ must stand.

General Threshold Model

- The process is structured the same way as the Linear Threshold Model.
  - Each node chooses θ_v from interval [0,1].
  - In this general model, v becomes active in step t if $f_v(s) \geq \theta_v$, where S is the active neighbors of v in the previous step.
- Linear Threshold Model is a special case of the general model.

General Cascade Model

- The general model states the probability that u activates its neighbor v, depending on v's neighbors that already tried.
- The incremental function is $p_v(u, S)$ in the interval [0,1].
  - S and {u} are subsets of the neighbor set.
- The Independent Cascade Model is a special case.

Converting from Threshold to Cascade

- To do so, it is necessary to know the probability that u can activate v based on which neighbors in the set S have already failed.
- The threshold must be in $(f_v(S), 1]$.
- Therefore,

$$p_v(u,S) = \frac{f_v(S \cup \{u\}) - f_v(S)}{1 - f_v(S)}$$

Converting from Cascade to Threshold

- Assuming nodes in S try to activate v in order, we can find the probability that v is not activated.
- From that, we can determine:

$$f_v(S) = 1 - \Pi^{K}_{t-1}(1-p_v(u_i, S_{i-1}))$$

Inapproximability Result

- “In general, it is NP-hard to approximate the influence maximization problem to within a factor of n^{1-E} for any $E > 0$.
  - u_i becomes active when a corresponding node s active.
  - For a large constant c, n^c nodes are added, each node connected to u_i and only becoming active when all of u_i is active.
  - If k sets cover all elements, activating the corresponding nodes will activate all nodes in u_i, meaning at least N+n+k nodes are active.
  - If no sets are covered, no nodes will be activated unless targeted, leading to less than n+k active nodes.
  - There is no algorithm to distinguish between the two.

Triggering Model

- Each node independently chooses a triggering set T_v over a distribution of its neighbors.
- v becomes active if a neighbor in T_v is active in the previous step.
- If u belongs to the triggering set of v, then the edge is considered live. The opposite is considered blocked.
  - v is activated if and only if there is a live-edge from the initial set to v.
- The influence function is submodular.

“Only-Listen-Once” Model

- Each node has a parameter p_v so the first neighbor to be activated causes v to be activated with probability p_v.
- This can also be expressed through the Triggering Set Model.
  - The triggering set T_v is the entire neighbor set of v or the empty set.
  - This proves to be submodular and produces the same approximation.

Decreasing Cascade Model

- This is a special case of the cascade model where the probability of an influencing node is restricted because it is non-increasing as a function of the set that has tried to influence v.
  - Specifically, this leads to $p_v(u, S) \geq p_v(u, T)$ when $S \subseteq T$

Monotone and Submodular Conjecture

- When threshold functions f_v are monotone and submodular at each node, the resulting influence function is monotone and submodular too.

Decreasing Cascade Model

- The previously discussed condition can be expressed as a threshold function.
  - $\frac{f(S \cup \{u\}) - f(S)}{1 - f_v(S)} \geq \frac{f(T \cup \{u\}) - f(T)}{1 - f_v(T)}$
  - This is a normalized submodularity property.

Non-Progressive Processes

- These are similar to progressive processes but nodes can switch in both directions.
- The process is very similar.
  - The difference is at each step, the node selects a new threshold value at random for the interval [0,1].
  - The node is active if $f_v(S) \geq \theta_v^{(t)}$.

Influence Maximization Problem

- For a particular node, at a particular time τ, we can target v for activation at time t.
- To do so, k interventions can be made.
- A is the set of k interventions. The influence of A is the sum over all nodes of the number of time steps v is active.

Influence Maximization Problem

- This can also be shown in a different graph.
- For G = (V, E) and time τ, a layered graph can be built for G^{τ} on $\tau^*|V|$ nodes.
  - This leads to a copy of each node at t.
  - Each node is connected with its neighbors and indexed from the last step.
- The non-progressive influence maximization problem on G over time τ is equivalent to the progressive version of the layered graph described above. v is active at t if v_t is active in the progressive process.
- This applies to approximation results for models for cascading failures in power grids by Asavathiratham et al.