社群網路分析

Influence Propagation & Maximization

Cheng-Te Li (李政德)

chengte@mail.ncku.edu.tw

Dept. of Statistics, NCKU

Course Info

- Reminder: **HW2** will be due on 5/20 (Sunday) 23:59
- **HW3** is announced today
  - Topic: Influence Maximization
  - Teamwork, Competition
  - Try to come up with your own methods!
  - All homeworks are out! ☺
- **Final Project**
  - Next week we will provide you a list of topics and datasets
  - You can start thinking your topic (very welcome!)
- Today
  - Lecture 7: Influence Propagation & Maximization
    - For HW3
  - Issues in Paper Presentation

Leaderboard

排名	組名	Influence Spread	最後上傳時間	總上傳次數
1	良心建議	205	2017-05-28 15:43:30	32
2	拍森GOGOGO	201	2017-05-28 21:05:12	83
3	今晚喝飲料	199	2017-05-27 21:10:38	36
4	小象隊	197	2017-05-27 20:25:01	30
5	*New Greedy IC Method*	195	null	null
6	balilarder	193	2017-05-28 00:36:16	21
7	Netizen.10	192	2017-05-28 03:11:53	6
8	*Degree Discount Method*	178	null	null
9	*Single Discount Method*	177	null	null
10	*Degree Heuristic Method*	170	null	null
11	元芳隊	157	2017-05-28 21:53:32	4
12	*Random Heuristic Method*	42	null	null
13	baku shou	30	2017-05-11 02:46:52	6

SNA 2017 Final Projects

組別	學生姓名	期末專題題目	所採用分析的資料
1	林彥儒	Inferring Networks of Diffusion for Disease	台南市政府開放資料、登革熱傳染擴散資料
2	丁羅邦芸	Information Diffusion-based Team Formation on Social Networks	DBLP 資訊科學學術發表資料與論文引用資料
3	陳宜均	FT2Vec: Forecasting Foot Traffic of Places in the Near Future	Gowalla 社群網站打卡資料、美國紐約市地理資料
4	賴愛華 夏婕禎	Multi-Curator When-to-Post on Social Networks	Twitter、Facebook 與 Google+之社群網站貼文分享資料
5	朱瑋瑩 高郁雯	Analyzing the Food Networks	Allrecipes 食譜資料庫
6	姜司原 方茜	Event participation forecasting in Event-based Social Networks	Meetup 社群事件活動資料
7	蔡秉翰 鍾定諺	Community Detection in Social Networks Based on Influence Propagation	Amazon 商品購買與評分資料
8	翁嘉良 陳建億	Rating Prediction of Movies	IMDB 電影資料庫
9	陳欣瑜	PSEISMIC: Personalized Self-Exciting Point Process Model for Predicting Tweet Popularity	Twitter 貼文與資訊擴散資料

Paper Presentation

- 5/16
  - Community Detection 王偉銘,簡湘霖, 賴暐婷, 吳展任, 陳品穎
  - Node Importance & Ranking 蔡永鴻(?), 詹京哲, 林沛權
- 5/30
  - Information Diffusion 趙晗揚, 李美諭
  - Social Link Analysis 郭子瑛, 黃文姍, 蔡永鴻(?)
  - Location-based Social Networks 陳昱忻, 林虹妤
- 6/13
  - Social Recommendation 丁國騰, 柯韋帆
  - Opinion/Sentiment Analysis 林巧玲, 王儷媛, 王薇筑
  - Health on Social Networks 洗鈺淇, 黃新幃
- 6/20
  - Privacy in Social Networks 曾子芸, 楊智婷, 吳澧宸

References

- D. Kempe, J. Kleinberg, and E. Tardos. "Maximizing the Spread of Influence through a Social Network." KDD 2003.
  - http://www.cs.cornell.edu/home/kleinber/kdd03-inf.pdf

KDD 2003 Best Paper Award
KDD 2014 Test-of-Time Award

5360 cites

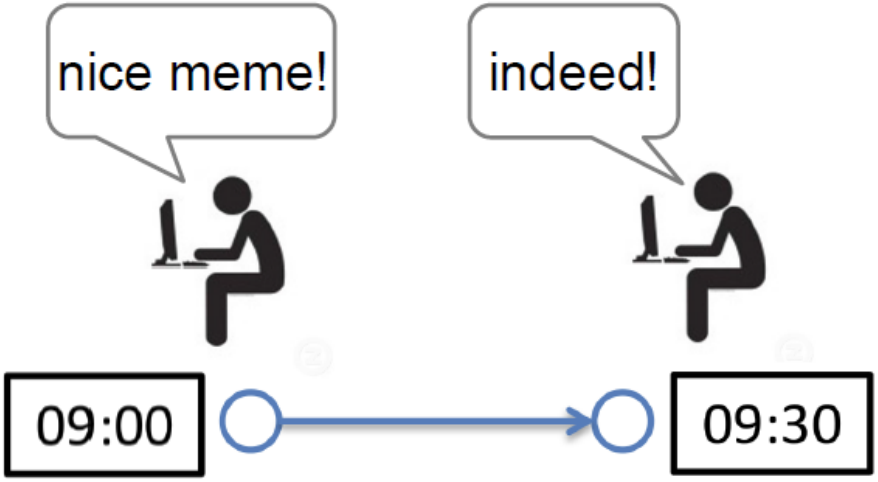
- H. Zhang et al. "Recent Advances in Information Diffusion and Influence Maximization of Complex Social Networks."
  - http://www.cise.ufl.edu/~mythai/files/diffusion.pdf
- W. Chen, et al. "Information and Influence Propagation in Social Networks." Morgan & Claypool Publishers, 2013.
  - https://wiki.eecs.yorku.ca/course_archive/2014-15/F/4412/_media/social_networks.pdf

Information Diffusion in Social Media

Grumpy Cat 不爽貓

- 25K+ votes in Reddit (< 1 day)
- 1M+ views in Imgur
- 300+ variants in Reddit
- 100+ Quickmeme macros

(< 2 days)



People are **connected** and perform **actions**

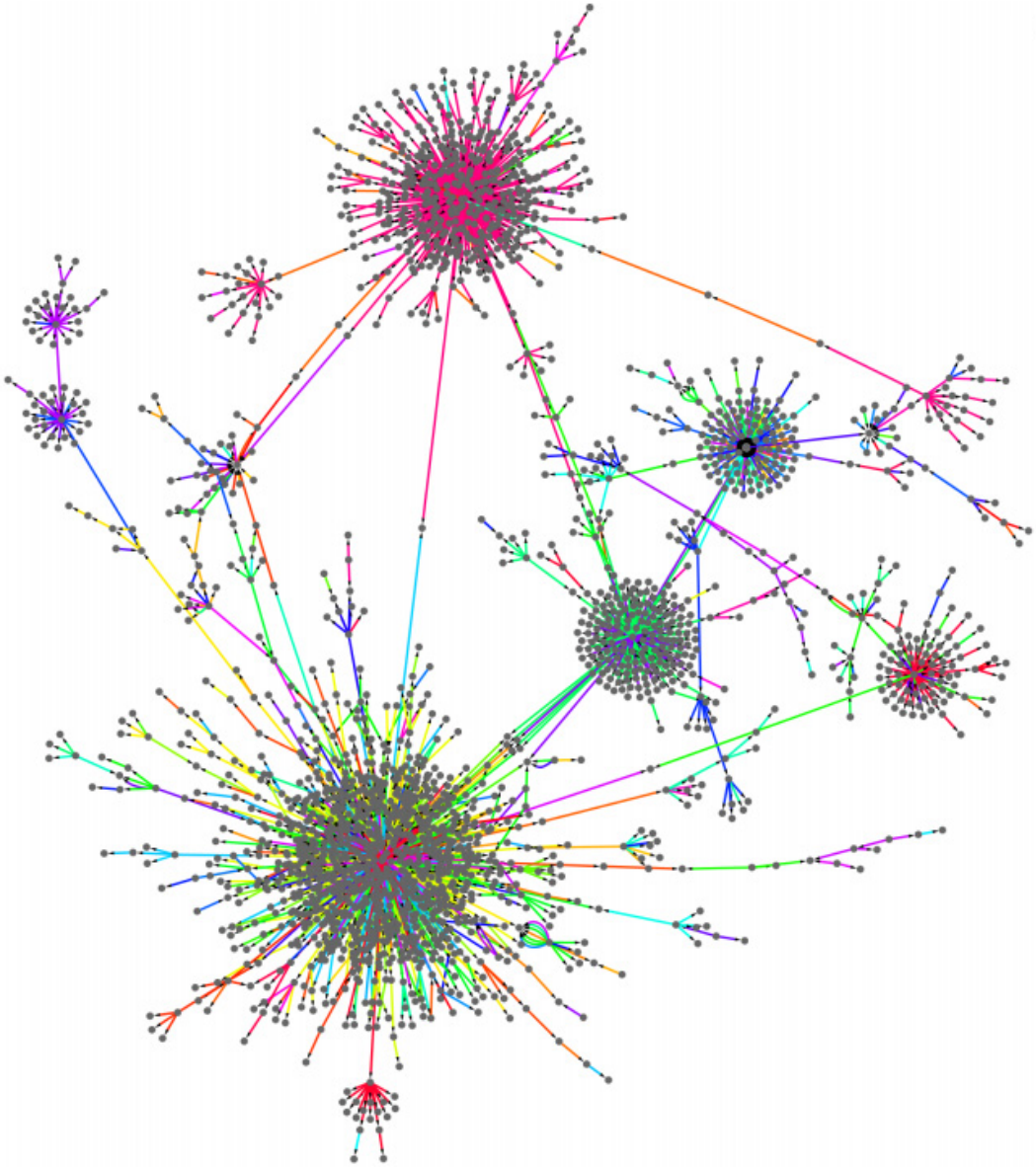
friends, fans, followers, etc.

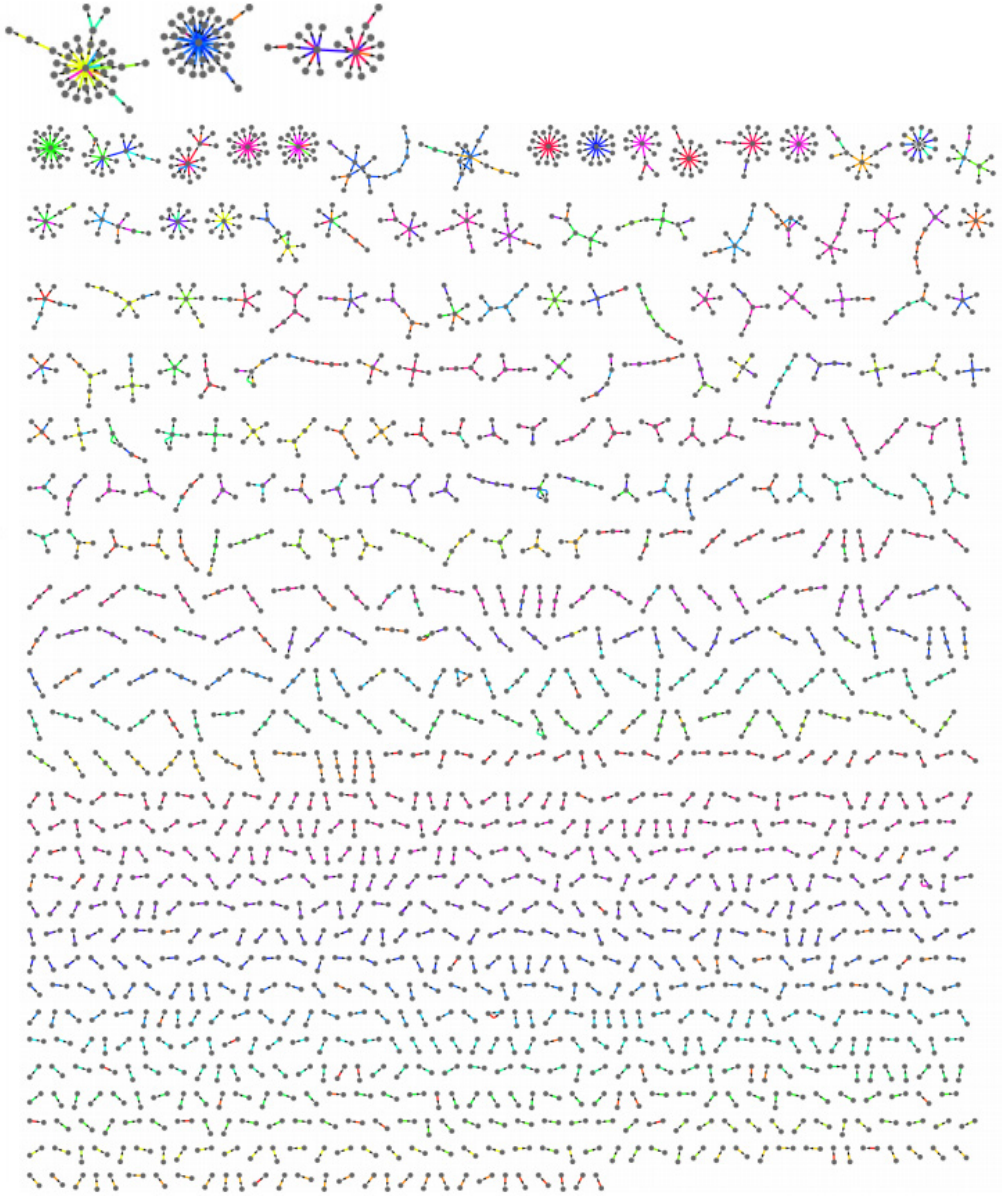
comment, link, rate, like, retweet, post a message, photo, or video, etc.

https://zh-tw.facebook.com/TheOfficialGrumpyCat

875萬粉絲...

The Shape of Information Diffusion in Twitter



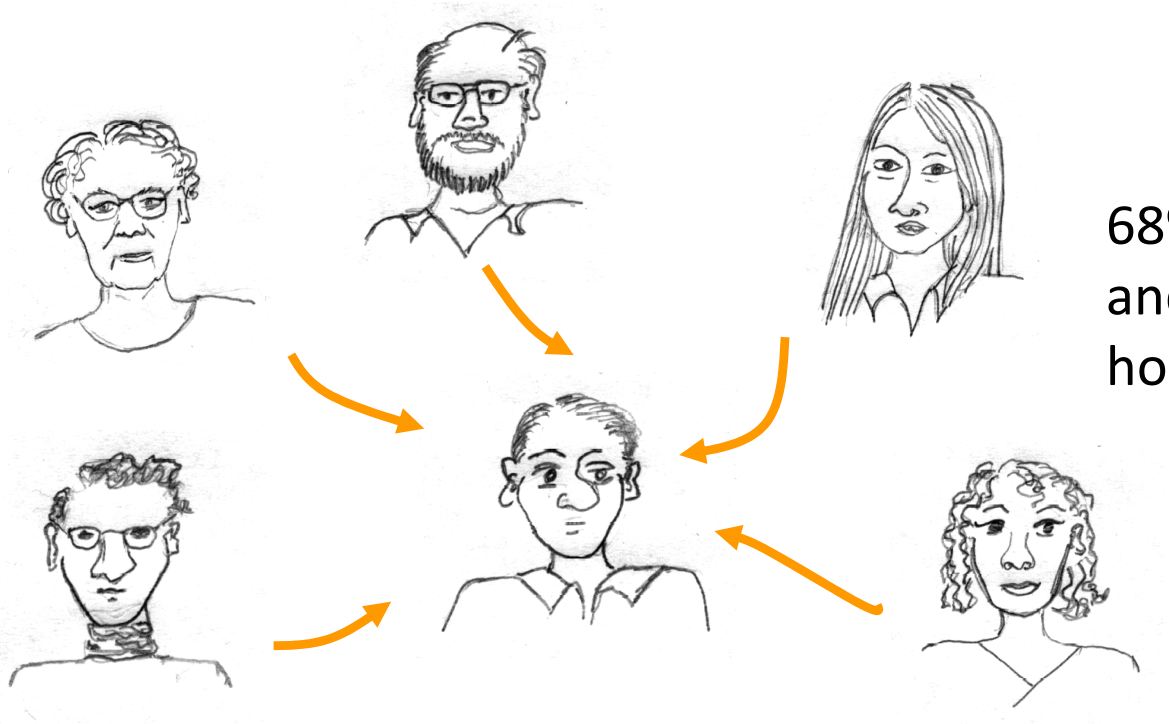


Social Network and Spread of Influence

- Social network acts as a medium for the spread of INFLUENCE among its members
  - E.g., opinion, ideas, information, innovation
- Direct marketing takes the "word-of-mouth" effects to significantly increase profits

Viral Marketing

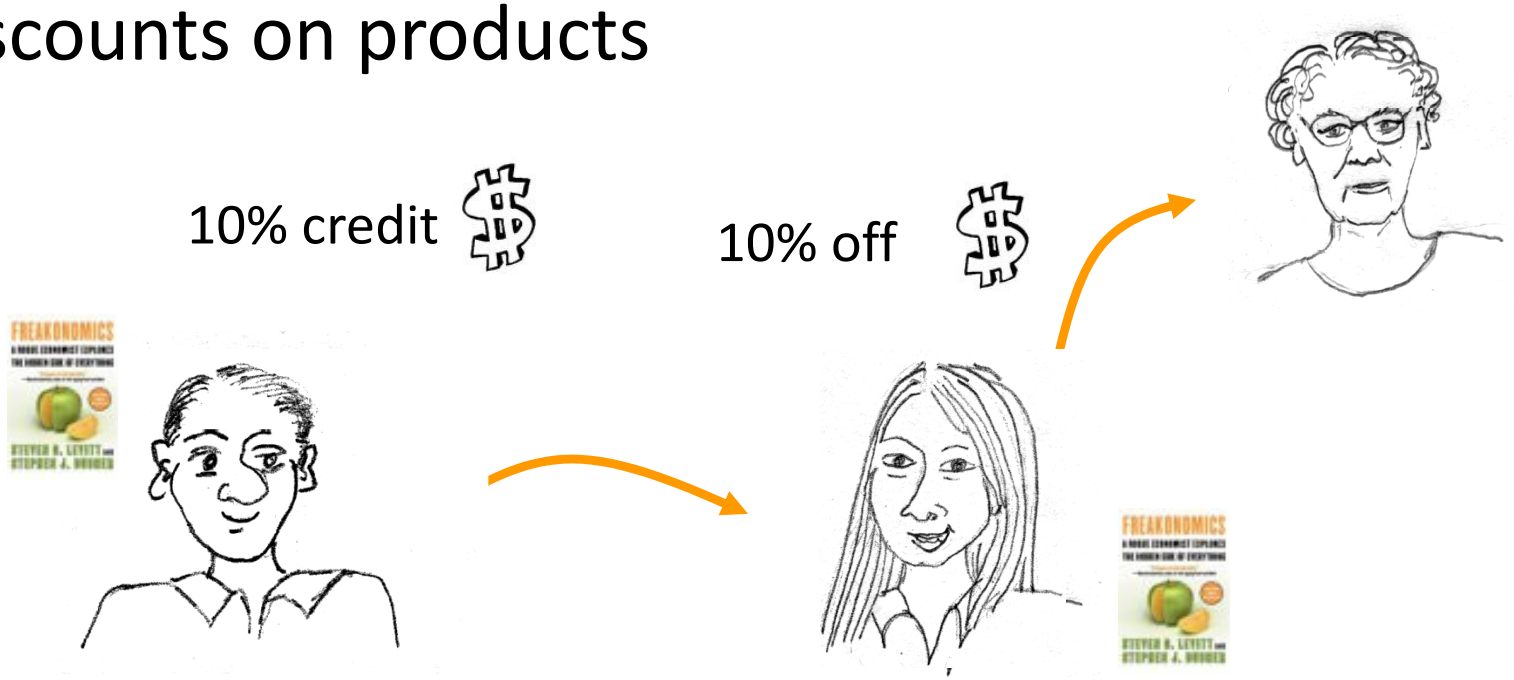
- Utilize pre-existing social networks to produce increases in brand awareness or to achieve other marketing objectives (e.g., product sales) through self-replicating viral processes



68% of consumers consult friends and family before purchasing home electronics

Viral Marketing

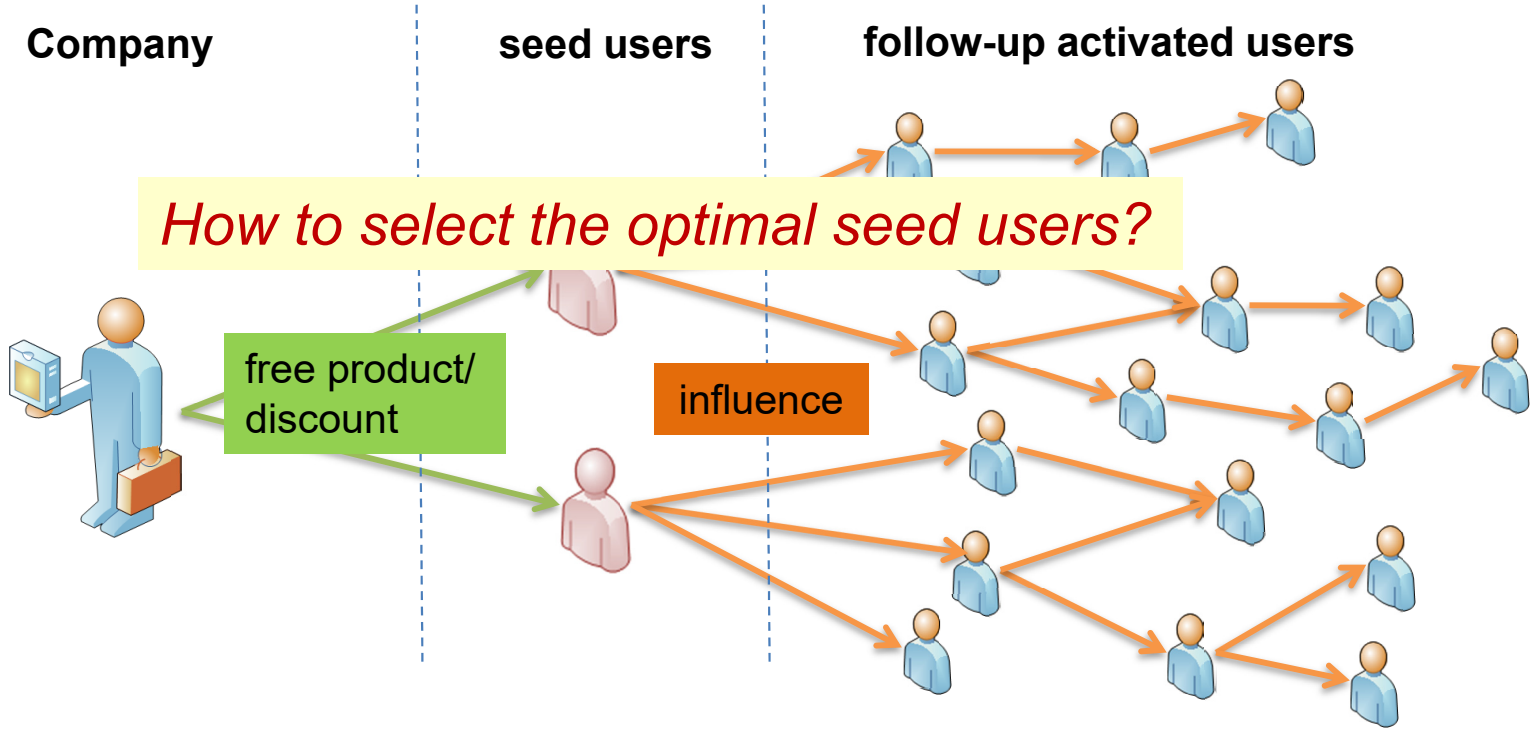
- Senders and followers of recommendations receive discounts on products



- Recommendations are made to any number of people at the time of purchase
- Only the recipient who buys first gets a discount

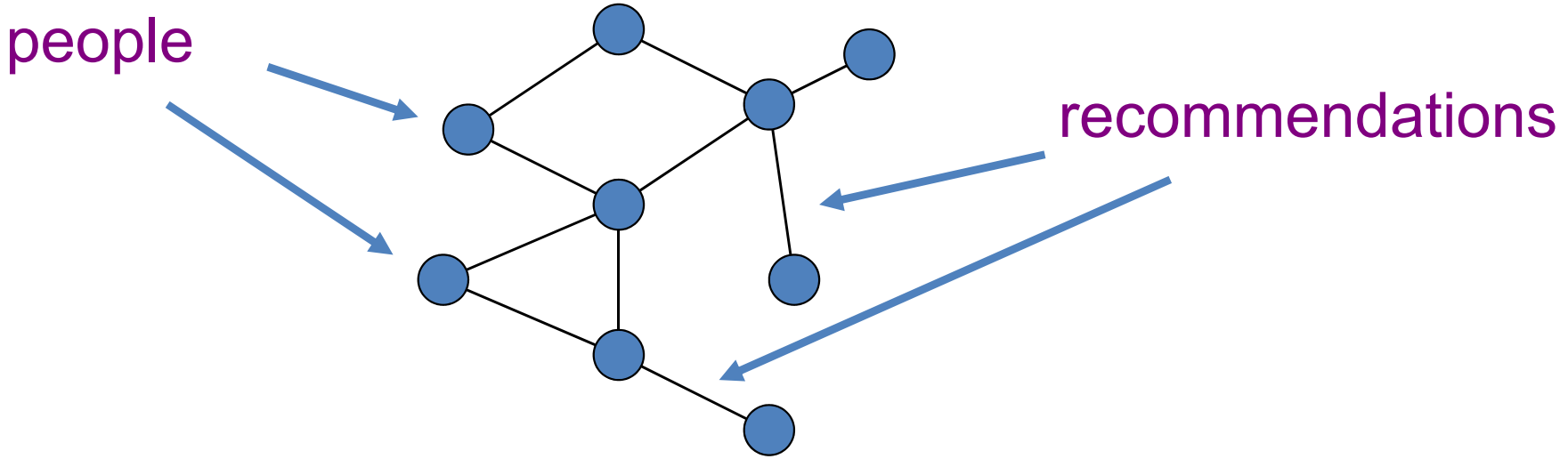
Viral Marketing

- To promote a product by seeding a few users; users adopting the product will recommend it
- Advantages: efficient; cost-effective

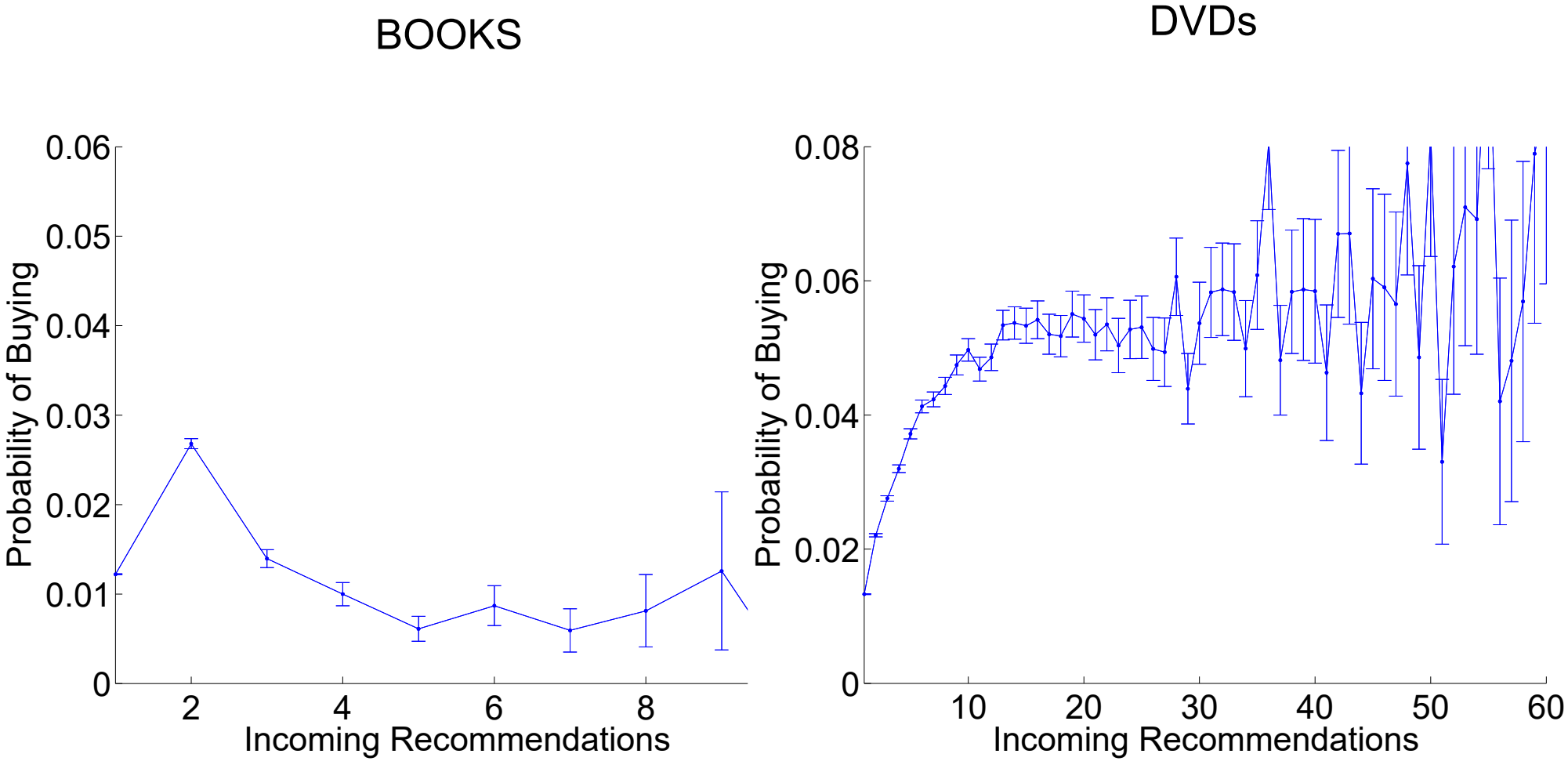


Statistics by Product Groups

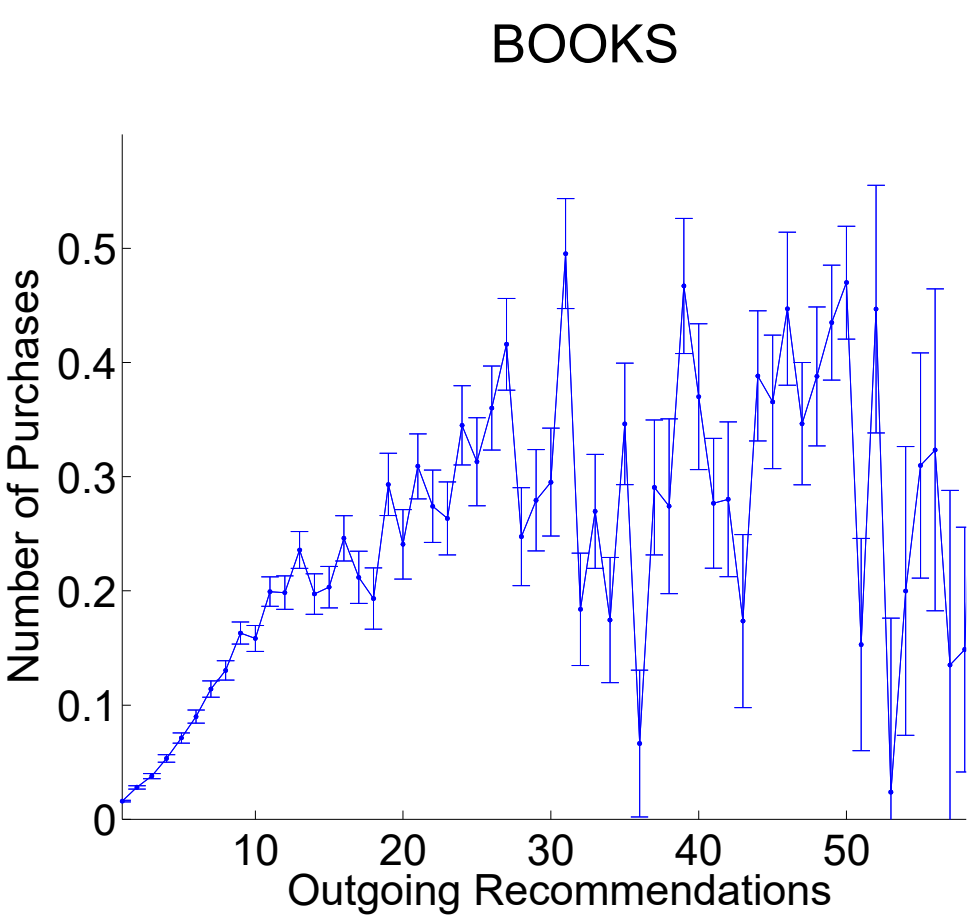
	products	customers	recommenda-tions	edges	buy + get discount	buy + no discount
Book	103,161	2,863,977	5,741,611	2,097,809	65,344	17,769
DVD	19,829	805,285	8,180,393	962,341	17,232	58,189
Music	393,598	794,148	1,443,847	585,738	7,837	2,739
Video	26,131	239,583	280,270	160,683	909	467
Full	542,719	3,943,084	15,646,121	3,153,676	91,322	79,164

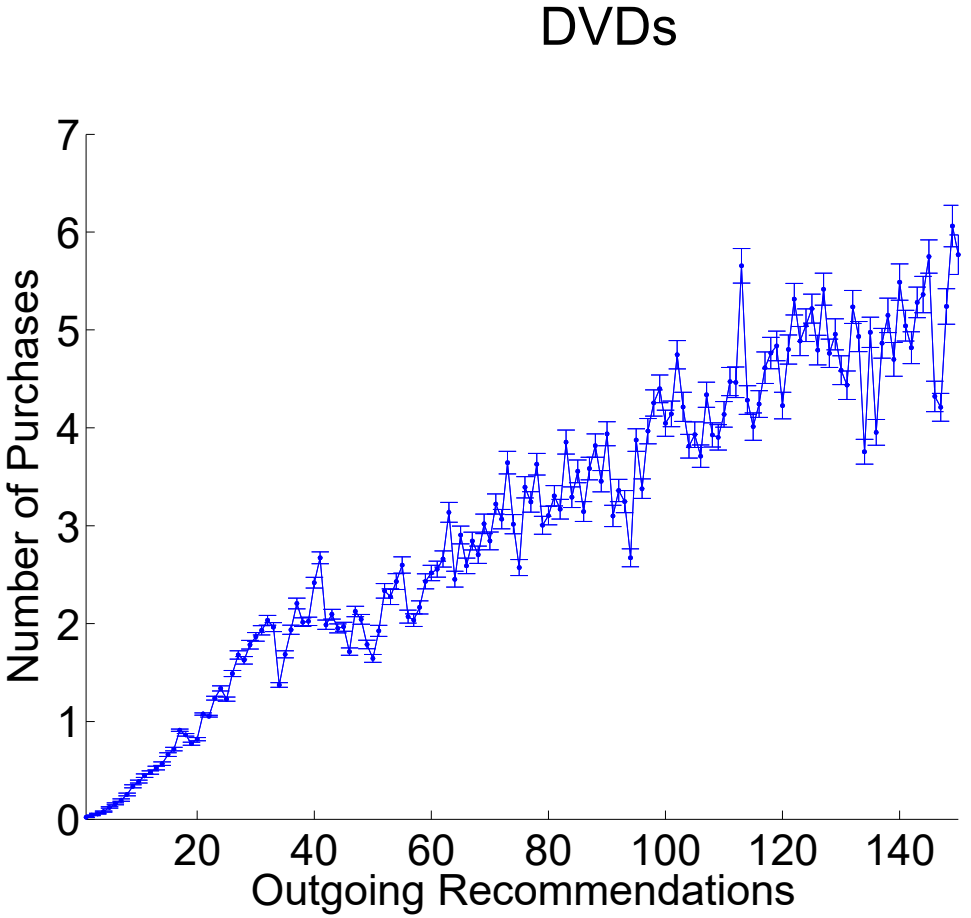


Does receiving more recommendations increase the likelihood of buying?

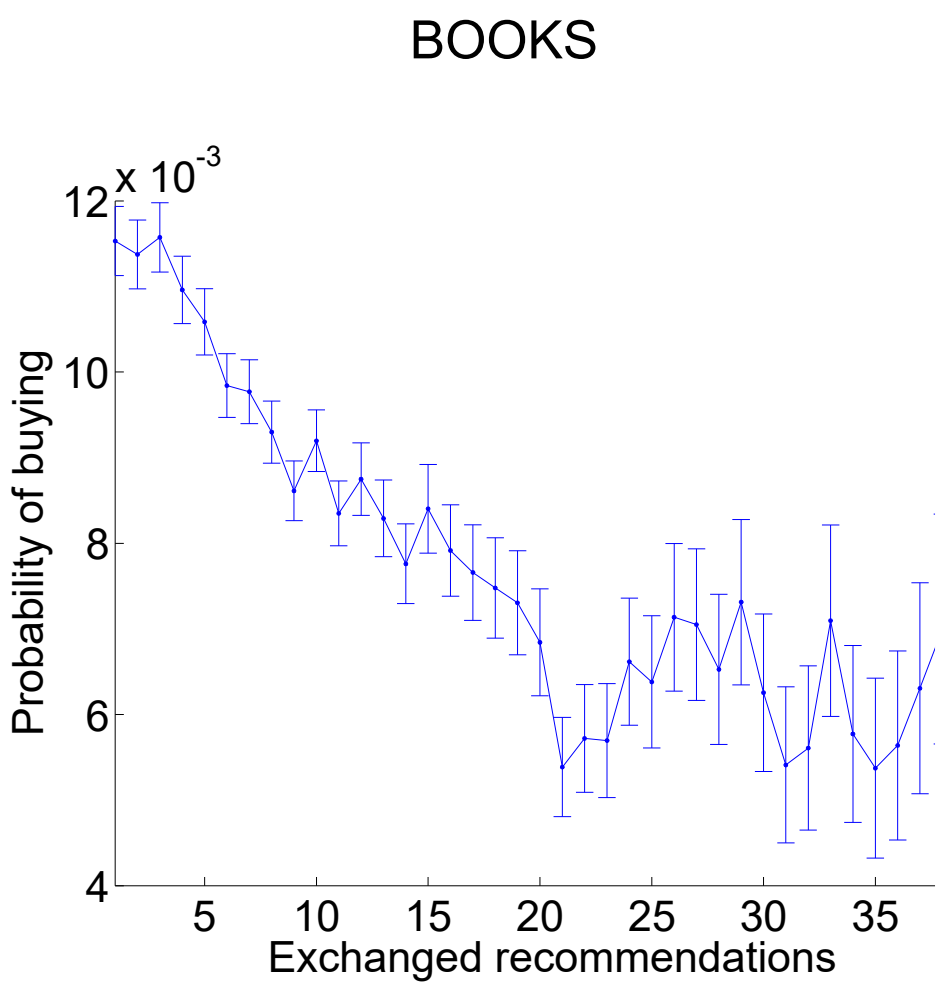


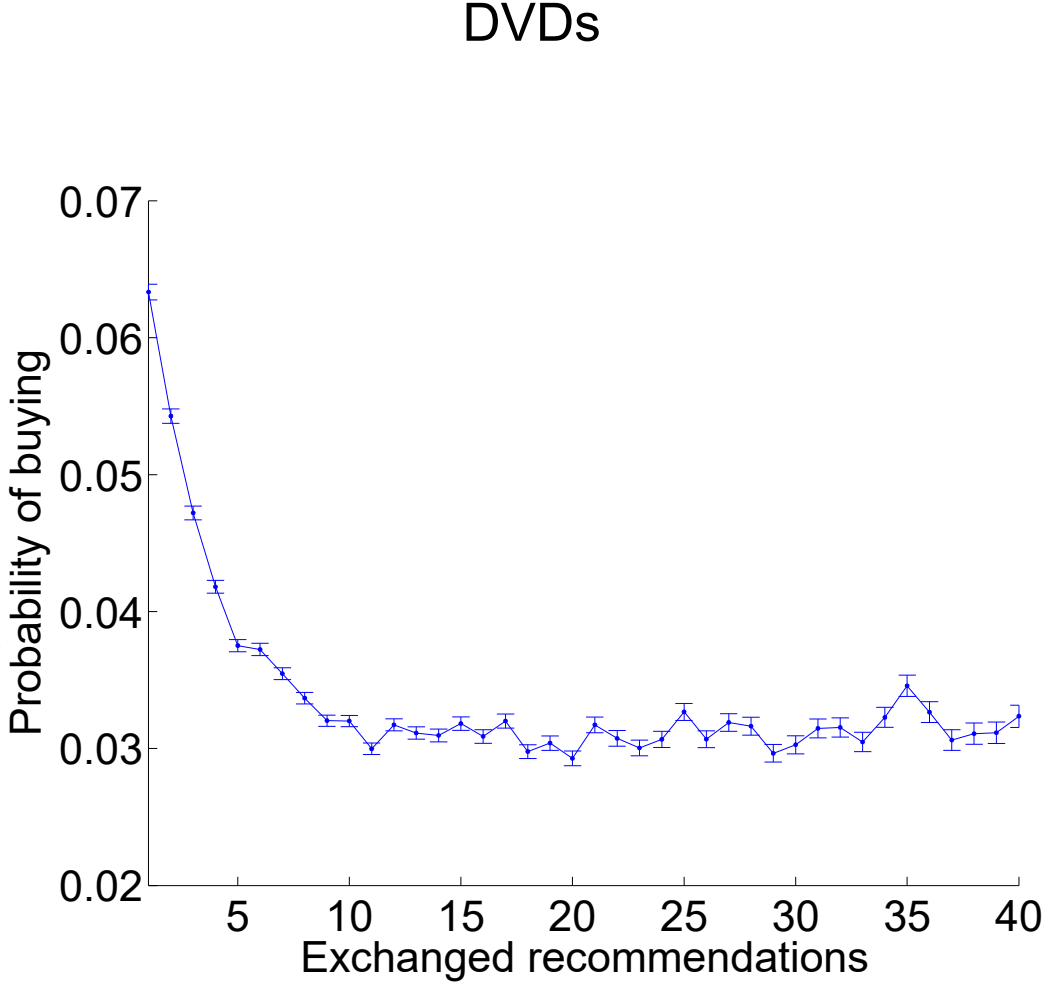
Does sending more recommendations influence more purchases?



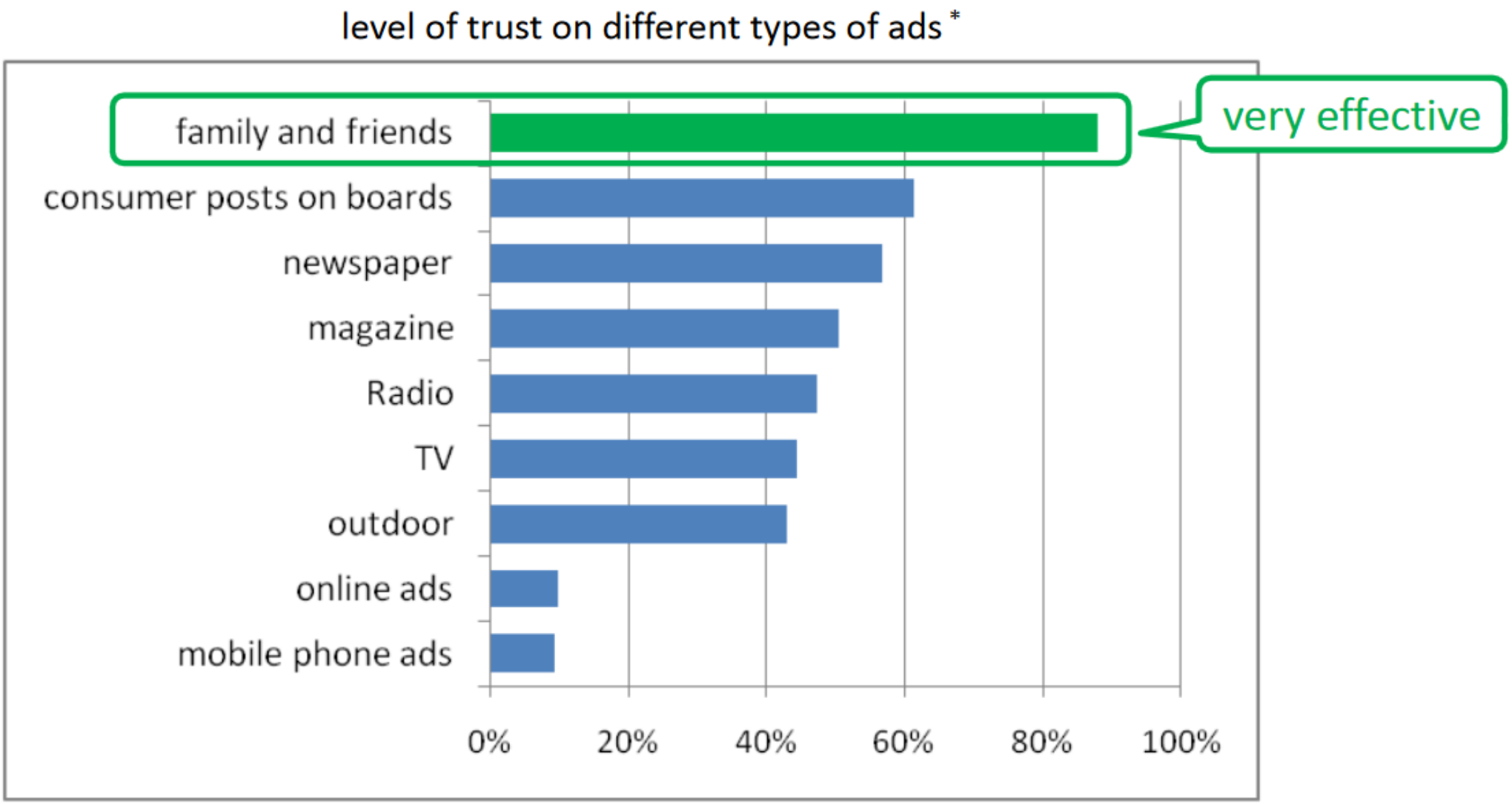


Multiple recommendations between two individuals weaken the impact of the bond on purchases





Effectiveness of Viral Marketing



*source from Forrester Research and Intelliseek

Processes and Dynamics

- Influence (Diffusion, Cascade)
  - Each node get to make decisions based on which and how many of its neighbors adopted a new idea or innovation
  - Rational decision making process
  - Known mechanics
- Infection (Contagion, Propagation)
  - Randomly occur as a result of social contact
  - No decision making involved
  - Unknown mechanics

Mathematical Models

- Models of Influence
  - Independent Cascade (IC) Model
  - Linear Threshold (LT) Model
  - Research Question:
    - **Influence Maximization:** who are the most influential nodes?
- Models of Infection
  - SIS: Susceptible-Infective-Susceptible (e.g. flu)
  - SIR: Susceptible-Infective-Recovered (e.g. chickenpox)
  - Research Question:
    - **Epidemic Thresholding:** will the virus take over the network?

水痘

Viral Marketing Problem

- Given
  - A limit budget B for initial advertising
    - E.g., give away free samples of product
  - Estimate for influence between individuals
- Goal
  - Trigger a large cascade of influence
    - E.g., further adoptions of a product
- Question we ask
  - Which set of individuals should B target at?

Common Settings of Influence Models

- A social network is represented a directed graph, with each customer being one node
- Each node is started as active or inactive
- A node, once activated, will activate his neighboring nodes
  - A node can switch to active from inactive
  - But it cannot switch to inactive from active

What We Need

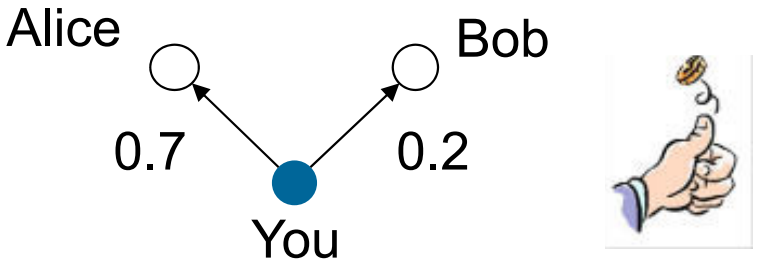
- Have the models of influence in social networks
- Obtain data about particular network
  - To estimate inter-personal influence
- Devise an algorithm to maximize spread of influence

Outline: Influence Maximization

- Models of Influence Propagation
  - Independent Cascade (IC)
  - Linear Threshold (LT)
- Influence Maximization Problem
  - Submodular
- Solutions
  - Greedy
  - CELF Greedy
  - New Greedy
  - Centrality Heuristic
  - Static Greedy
  - Pruned BFS
  - Degree Discount Heuristic

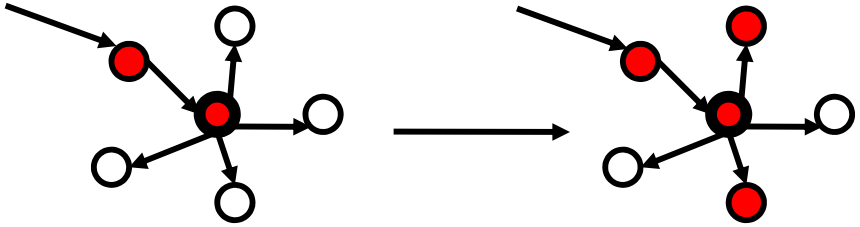
Independent Cascade Model

- When node *v* becomes active, it has a single chance to activate each currently inactive neighbor *w*
- The activation attempt succeeds with probability p_{vw} (a parameter of the system)



- Whether or not *v* succeeds, it cannot make any further attempts to activate *w* in subsequent rounds
- Run until no more activations are possible

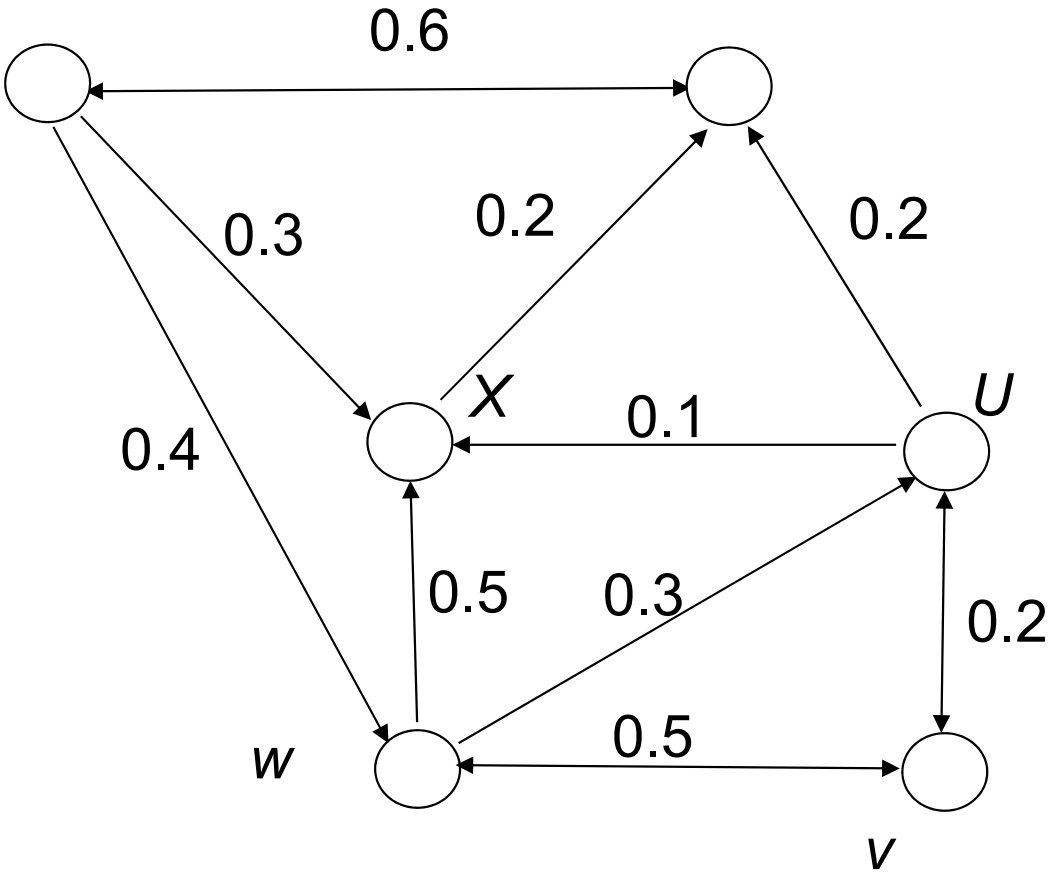
P=50%

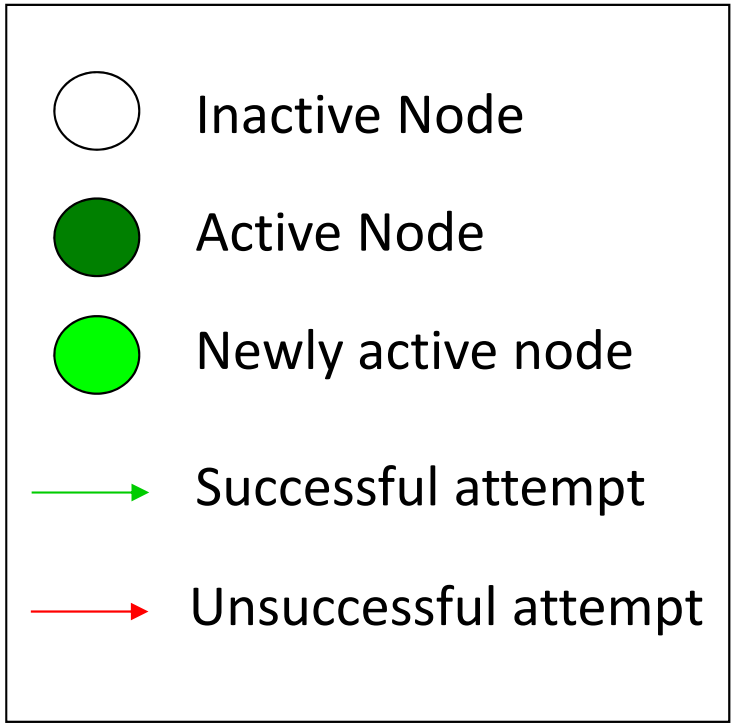


Progressive

History-Insensitive

An Illustration





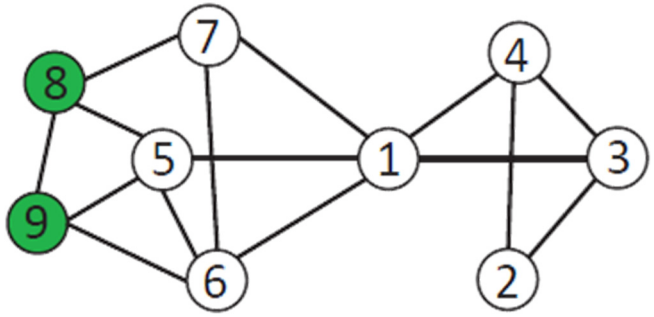
Stop!

Independent Cascade Model Procedure

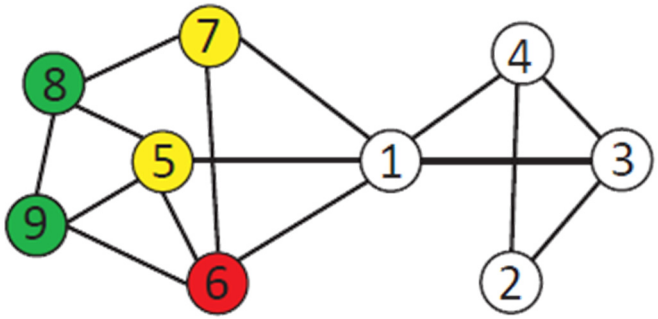
- A node *v, once activated* at step *t , has one chance to activate each of its neighbors randomly*
  - For a neighboring node w, *the activation* succeeds with probability p_{wv} (e.g. $p = 0.05$)
- If the activation succeeds, then *w will become active at step* $t + 1$
  - In subsequent rounds, v *will NOT attempt to activate* w *anymore* (no matter whether it succeeds or not)
- *The diffusion process,* starts with an initial activated set of nodes, then continues until no further activation is possible

IC Model: Another Example

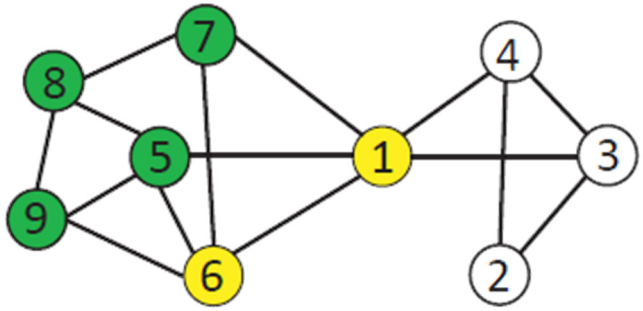
- $p_{w,v}$=0.5 for all edges in the network, i.e., a node, once activated, will activate his inactive neighbors with a 50% chance.
- NOTE: IC model activates one node with certain success rate. Thus, we might get "a different result" for another run.



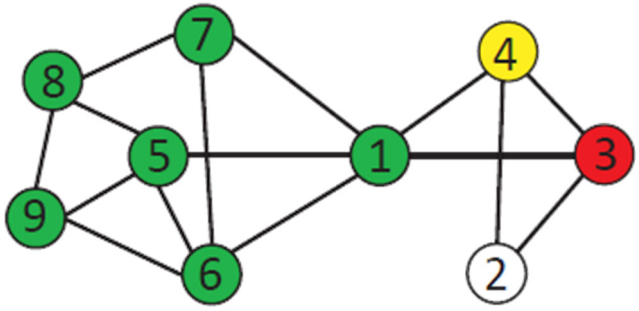
Step 0



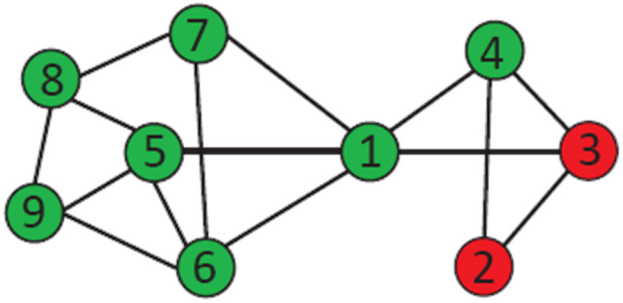
Step 1



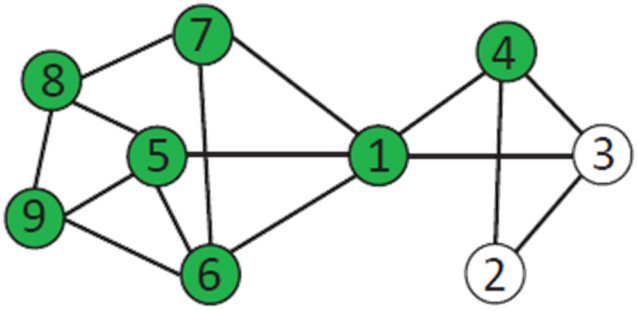
Step 2



Step 3



Step 4



Final Stage

IC Model: Algorithm

Algorithm 7.1 Independent Cascade Model (ICM)

Require: Diffusion graph $G(V, E)$, set of initial activated nodes A_0, activation probabilities $p_{v,w}$

```
1:  return Final set of activated nodes A_∞
2:  i = 0;
3:  while A_i ≠ {} do
4:
5:      i = i + 1;
6:      A_i = {};
7:      for all v ∈ A_{i-1} do
8:          for all w neighbor of v, w ∉ ∪_{j=0}^{i} A_j do
9:              rand = generate a random number in [0,1];
10:             if rand < p_{v,w} then
11:                 activate w;
12:                 A_i = A_i ∪ {w};
13:             end if
14:         end for
15:     end for
16: end while
17: A_∞ = ∪_{j=0}^{i} A_j;
18: Return A_∞;
```

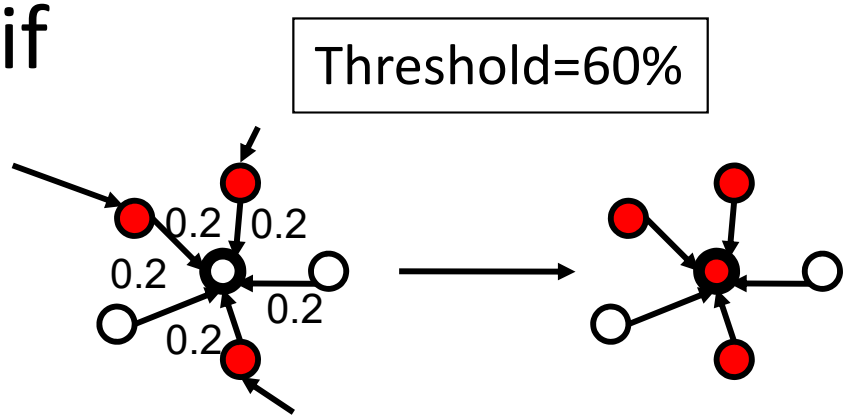
Random activation

Determine the success or not

Linear Threshold Model

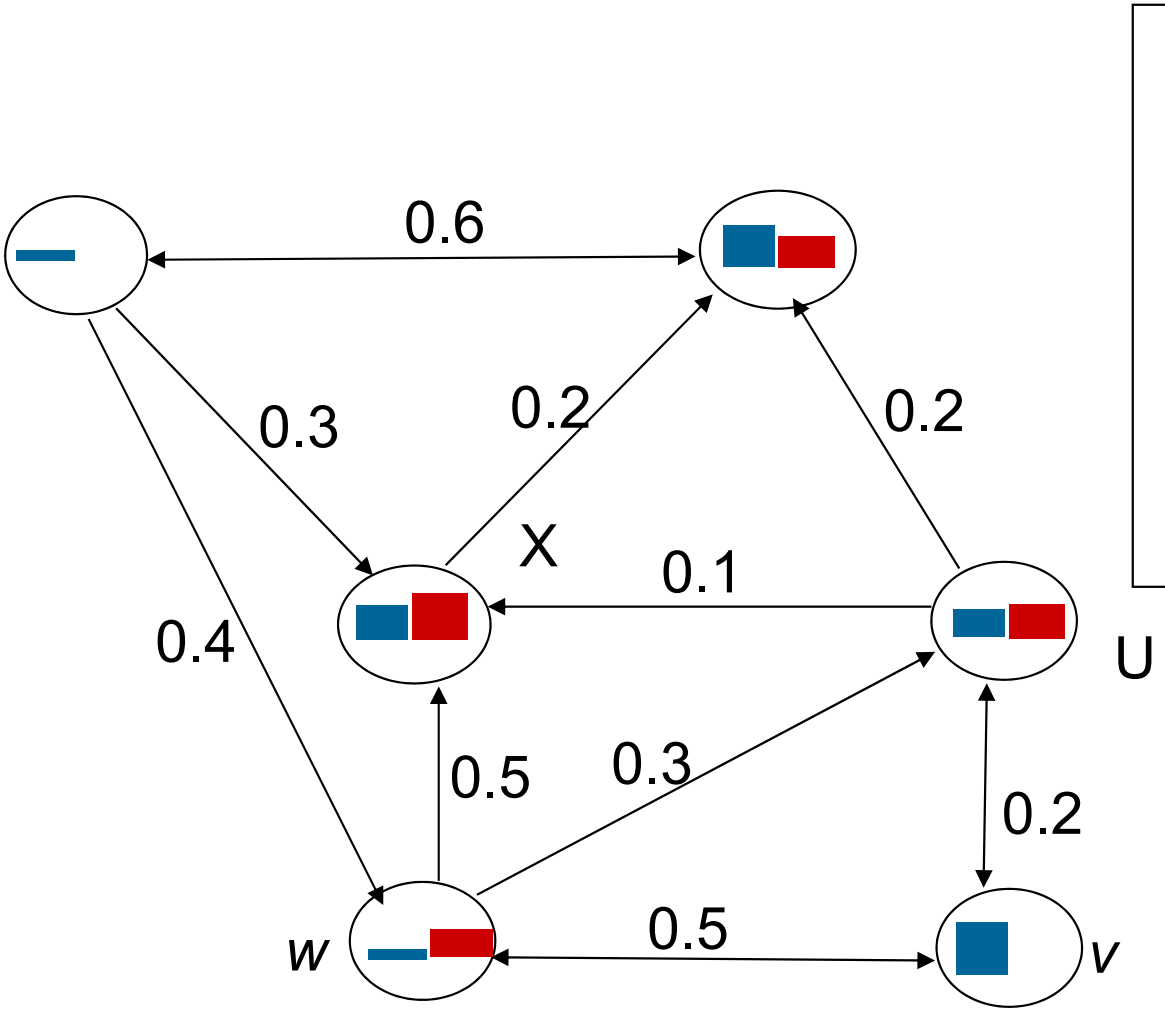
- A node v is influenced by each neighbor w according to a weight b_{vw}
- Each node v has a threshold $\boldsymbol{\theta_v}$ which is chosen uniformly at random from [0, 1]
- A node v becomes active if

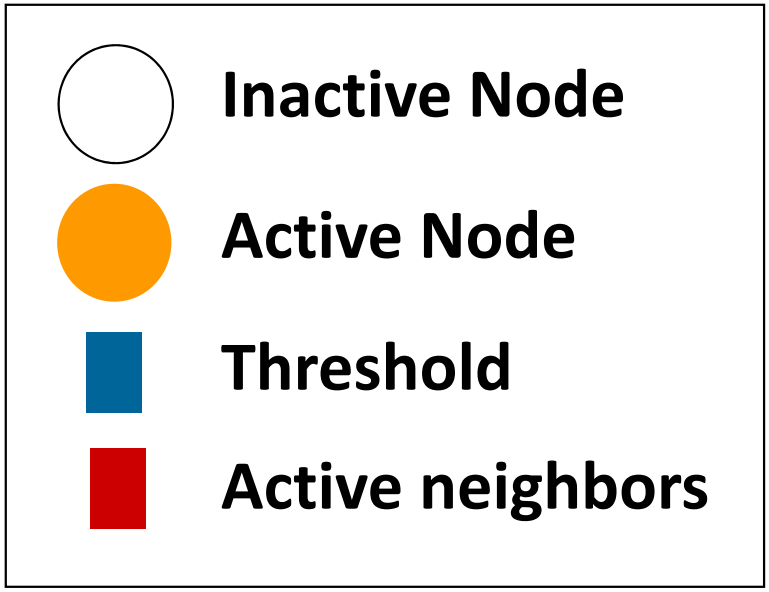
$$\sum_{w \text{ neighbor of } v \text{ (w is active)}} b_{v,w} \geq \theta_v$$



- Runs until no more activations are possible

An Illustration





Stop!

Linear Threshold Model

An individual would take an action if the number of his friends who have taken the action exceeds (reaches) a certain threshold

- Each node v *chooses a threshold* θ_v *randomly from a uniform* distribution in an interval between 0 and 1
- In each discrete step, all nodes that were active in the previous step remain active
- The nodes satisfying the following condition will be activated

$$\sum_{w \in N_v, w \text{ is active}} b_{w,v} \geq \theta_v$$

LT Model: Another Example

the network is directed

weights $b_{w,v}$ and $b_{v,w}$ are different

Assumption

$b_{w,v} = 1/d_v$

$b_{v,w} = 1/d_w$

for each node v, $\theta_v = 0.5$.

start from two activated nodes 8 and 9 ($S_0 = \{v_8, v_9\}$)

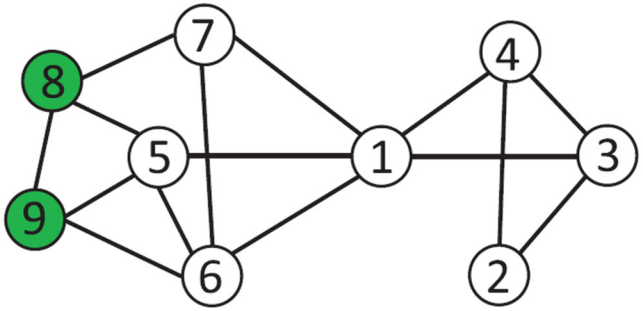
In the first step,

with two of its neighbors being active, v_5 receives weights

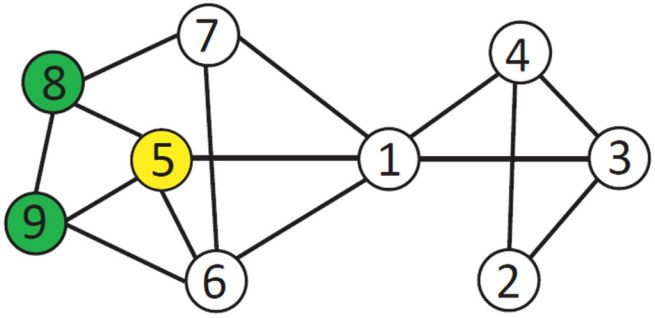
$b_{8,5} + b_{9,5} = \frac{1}{3} + \frac{1}{3} = \frac{2}{3} \geq 0.5$.

Thus, v_5 is activated.

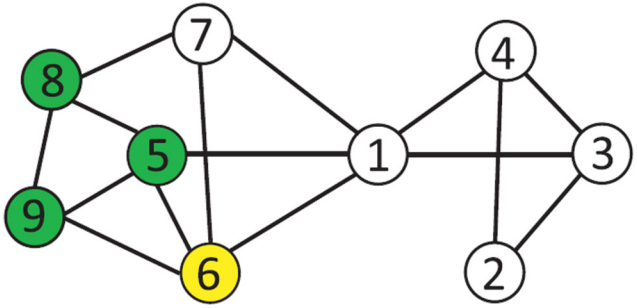
Diffusion process terminates after nodes 1, 5, 6, 7, 8, 9 become active



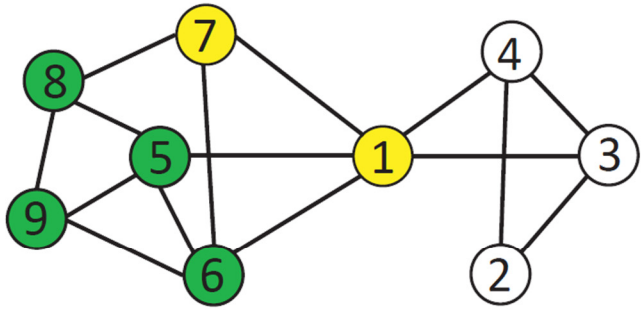
Step 0



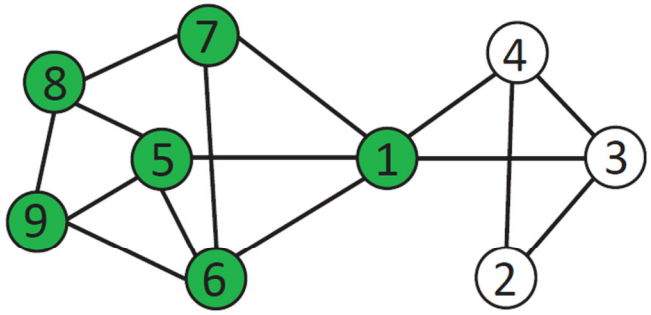
Step 1



Step 2



Step 3



Final Stage

LT Model: Algorithm

Algorithm 8.1 Linear Threshold Model (LTM)

Require: Graph $G(V, E)$, set of initial activated nodes A_0

1: **return** Final set of activated nodes A_∞
2: i=0;
3: Uniformly assign random thresholds θ_v from the interval [0, 1];
4: **while** $i = 0$ or $(A_{i-1} \neq A_i, i \geq 1)$ **do**
5: $A_{i+1} = A_i$
6: inactive $= V - A_i$;
7: **for all** $v \in$ inactive **do**
8: **if** $\sum_{j \text{ connected to } v, j \in A_i} w_{j,v} \geq \theta_v$. **then**
9: activate v;
10: $A_{i+1} = A_{i+1} \cup \{v\}$;
11: **end if**
12: **end for**
13: $i = i + 1$;
14: **end while**
15: $A_\infty = A_i$;
16: Return A_∞;

Thresholding

Using active neighbors' weights to determine the success or not

IC vs. LT

- IC is sender-centric
  - Once a node is activated, it tries to activate all its neighbors
  - Does not depends on the neighbor nodes
    - An active node activates each of its neighbors independently
  - It varies (undeterministically) depending on the cascading process → need to perform up to a large number of times and compute the average number of active nodes
- LT is receiver-centric
  - By looking at all the neighboring nodes of one node, it determines whether to activate the node based on its threshold
  - Depend on the whole neighborhood of one node
  - The diffusion process is deterministic if the weights are given
- Both models involve randomization:
  - IC succeeds activates a neighboring node with probability $p_{w,v}$
  - LT randomly selects a threshold θ_v for each node v

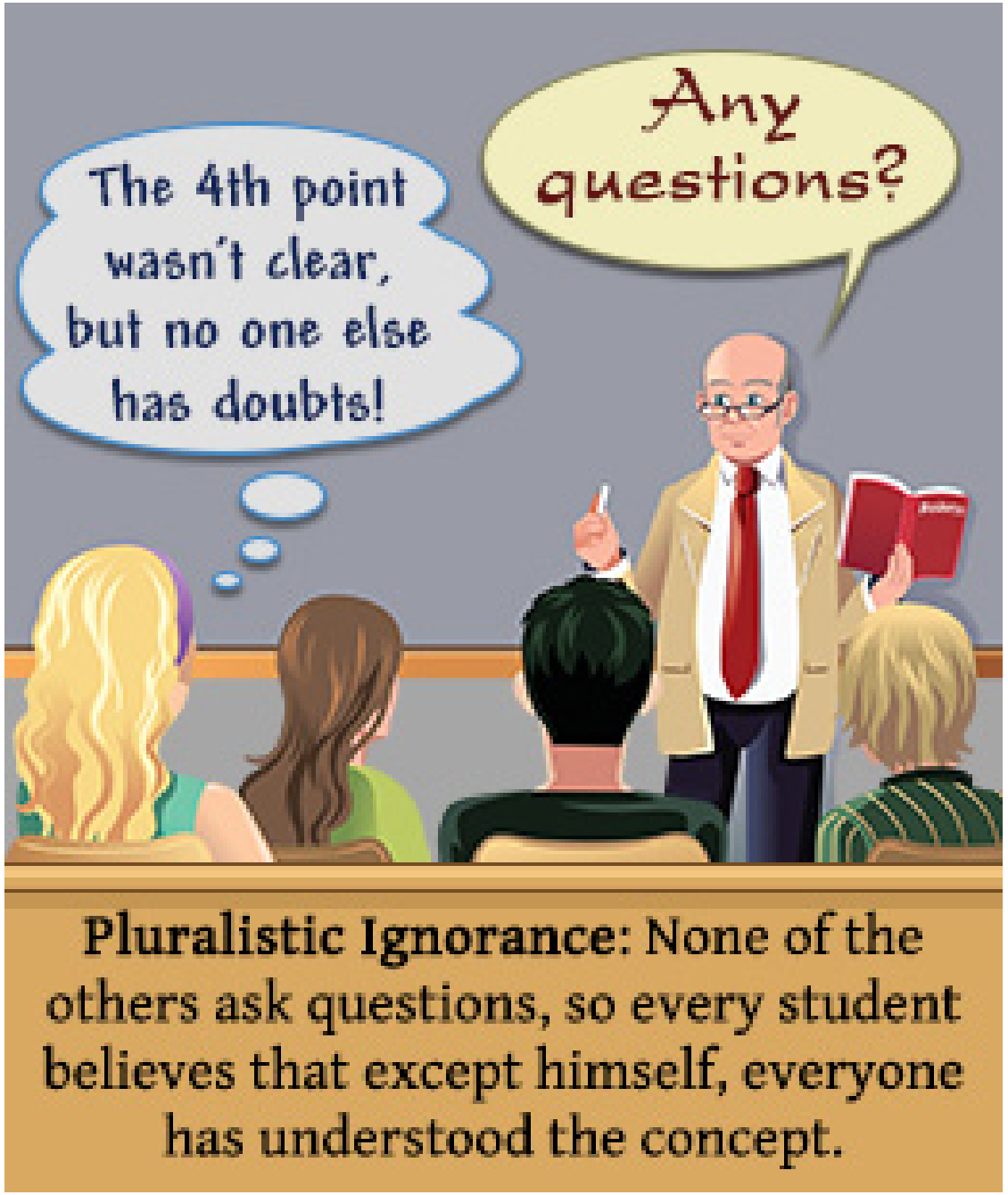
How should we organize revolt?

- You live an in an oppressive society
- You know of a demonstration against the government planned tomorrow
- If a lot of people show up, the government will fall
- If only a few people show up, the demonstrators will be arrested and it would have been better had everyone stayed at home

“Pluralistic Ignorance” (多數無知)

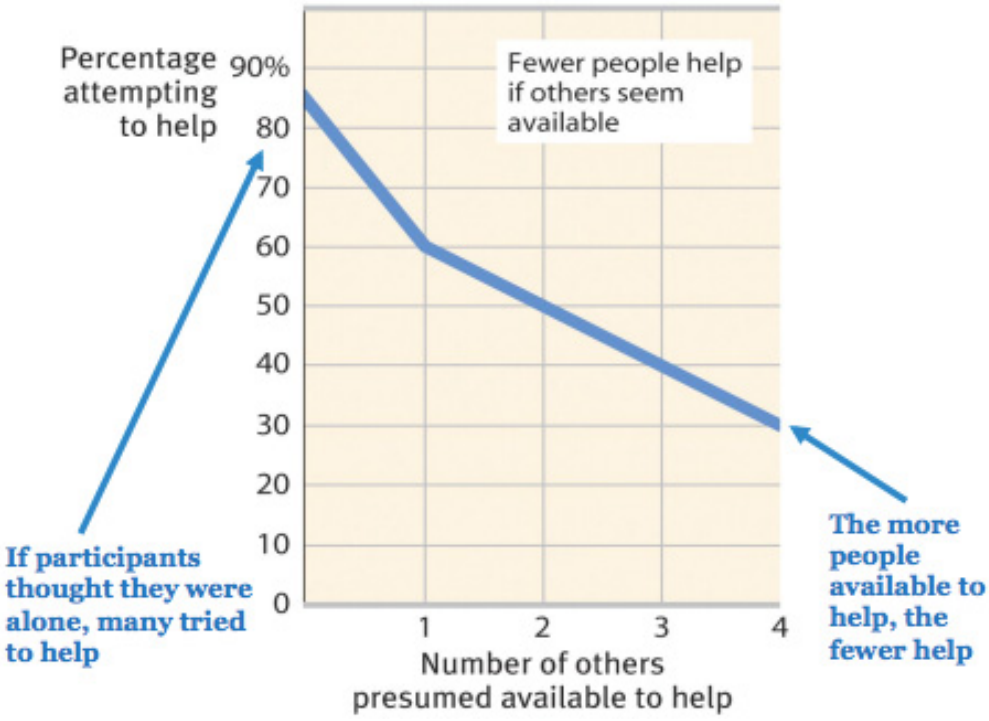
- You should do something
  if you believe you are in the majority!
- Dictator tip: Erroneous estimates about the prevalence of certain opinions in the population

“Pluralistic Ignorance”



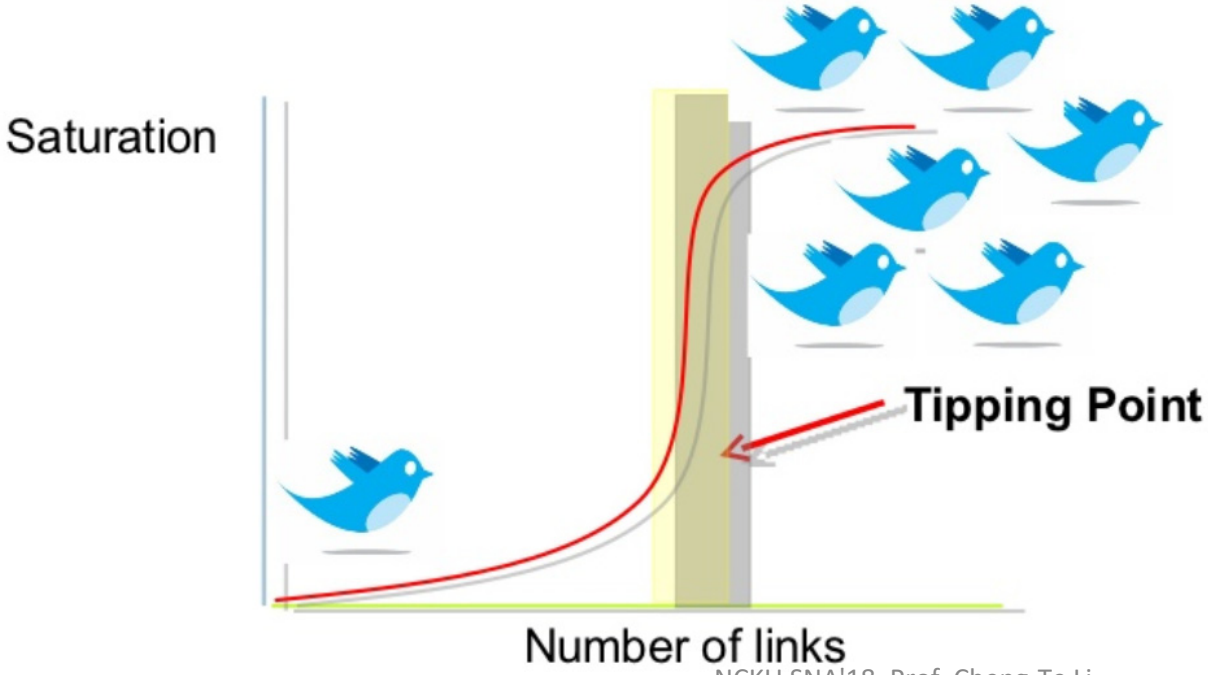
“Pluralistic Ignorance”

- No one actually believes, but everyone believes that everyone else believes



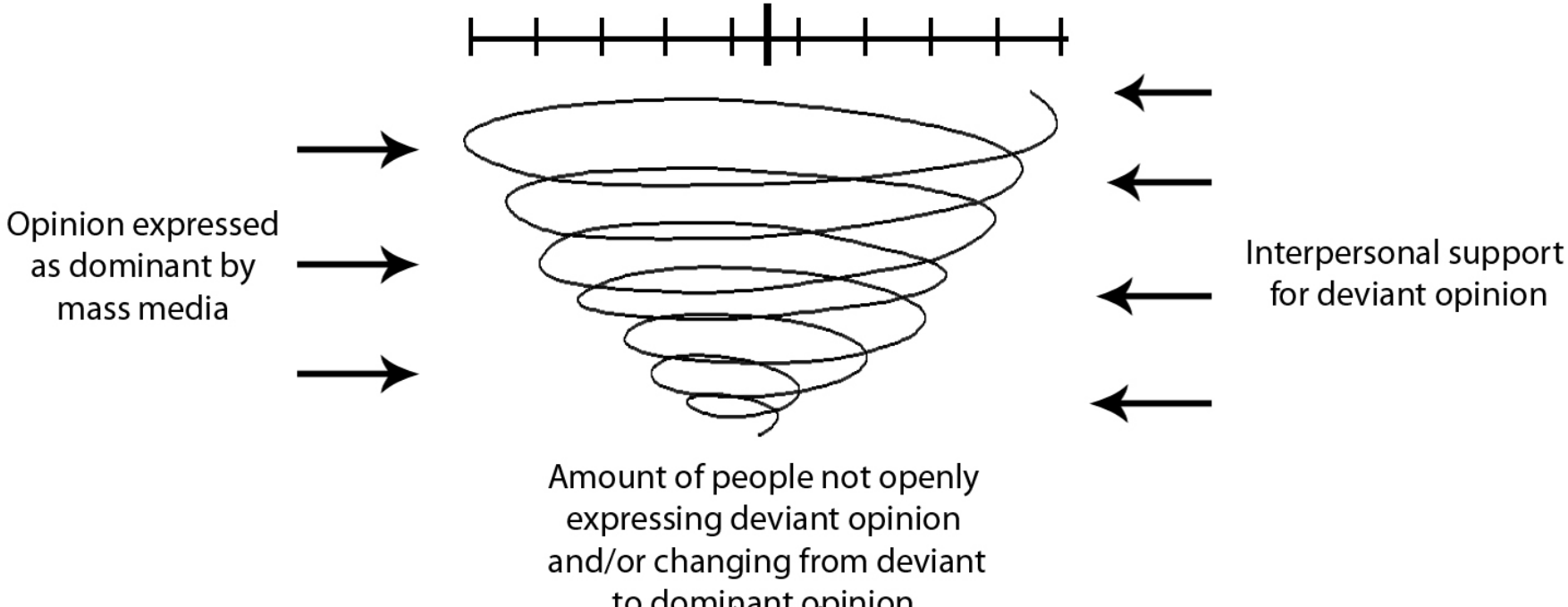
Organizing the Revolt: The Model

- Personal threshold k: *"I will show up if am sure at least k people in total (including myself) will show up."*
- Each node only knows the threshold and attitude based on all their direct friends
- Unsolved/open problem: can we predict if a revolt can happened based on the network structure?

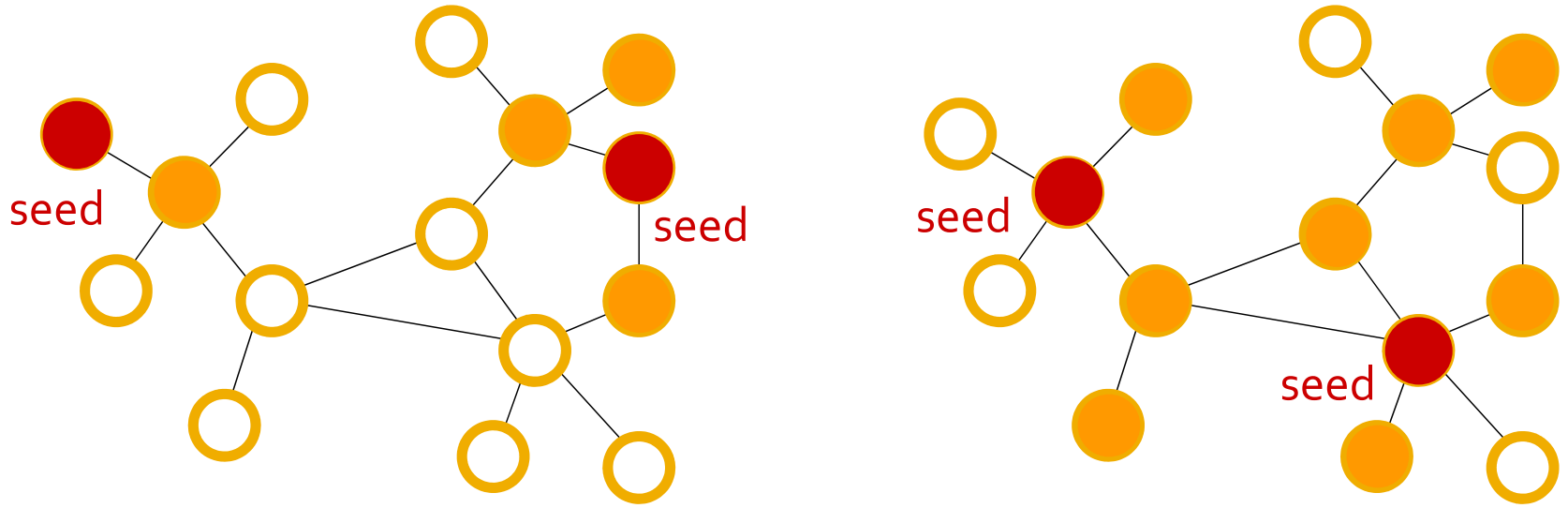


“Spiral of Silence”

- 「沉默螺旋理論」假定，為免在重要公共議題產生孤立情況，許多人會受到主流意見或衰微意見影響
- 「如果人們覺得自己的觀點是公眾中的少數派，他們將不願意傳播自己的看法；而如果他們覺得自己的看法與多數人一致，他們會勇敢的說出來。而且媒體通常會關注多數派的觀點，輕視少數派的觀點。於是少數派的聲音越來越小，多數派的聲音越來越大，形成一種螺旋式上升的模式。」(Wikipedia)



Viral Marketing: Social **Influence Maximization**



- Given a limit budget k for initial seeding, how to identify a small set of influential customers (as seeds) such that by convincing them to adopt the product and finally trigger a larger cascade of influence

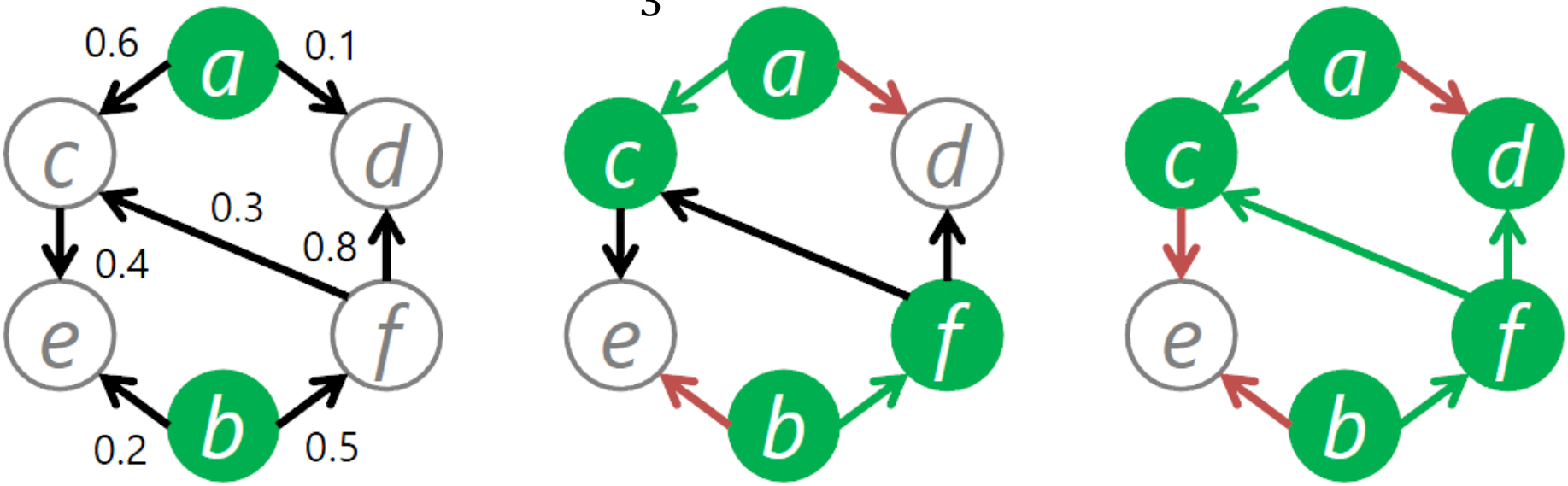
Influence Maximization Problem

- **Influence Spread** of node set S denoted by: $f(S)$, $\sigma(S)$, or $R(S)$
  - Expected number of active/influenced nodes at the end, given the seed set S **is the initial active set**

Active u to Inactive v
- Success →
- Failure →

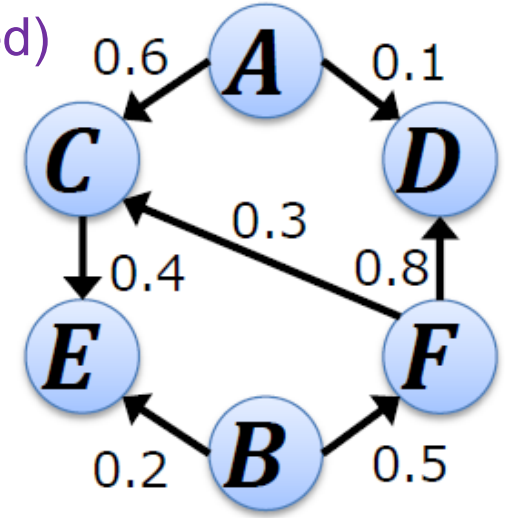
$$\sigma(S) := \mathbf{E}[\#\ \text{active vertices given}\ S]$$

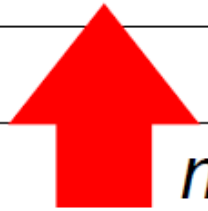
Assume $S = \{a, b\}$ $\sigma(S) = \frac{2+4+5}{3} = 3.67$



Influence Maximization Problem

- Input
  - Directed graph $G = (V, E)$ (can be bi-directed)
  - Edge probability p_e $(e \in E)$ Propagation probability
  - Size of seed set k
- Problem : Find a seed set S
  - maximize $\sigma(S)$ $(|S| \leq k)$
    - $\sigma(\cdot)$: the spread of influence





- Motivation *mathematically formalizing*
  - Viral (word-of-mouth) Marketing
    [Domingos, Richardson. KDD'01], [Richardson, Domingos. KDD'02]

Q. How to find a small group of influential individuals?

An Approximation Solution

- Influence Maximization problem is NP-hard
  - Could not be solved within polynomial time
- Greedy method

For each of k iterations:
Add a node u to set S that
maximizes $\sigma(S \cup \{u\}) - \sigma(S)$

- Theorem

The Greedy method is a $(1 - 1/e)$ approximation for both IC and IT models

General Greedy Algorithm

Initialize $S = \emptyset$ and $R = 20000$

for $i = 1$ to k do:

  for each node $u \in V \backslash S$ do:

    $s_u = 0$

    for $round = 1$ to R do:

      $s_u \mathrel{+}= \sigma(S \cup \{u\})$ // run IC or LT model

    $s_u = s_u / R$

  $S = S \cup \{argmax_{u \in V \backslash S}\{s_u\}\}$

Output S

Greedy Algorithm: In Selecting the i^{th} Seed

Original network G

Compute influence spread $\sigma(S)$

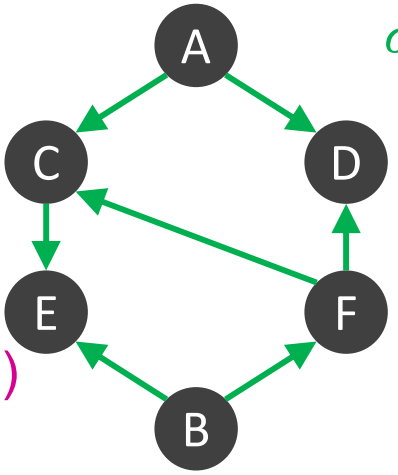
Assume $k = 2$ (aim to select 2 seeds)
We have already selected
one seed node $S = \{A\}$
Now we are finding the 2nd seed

For each node $v \in V \backslash S$:
Run the Independent Cascade (IC) model simulation up to R times

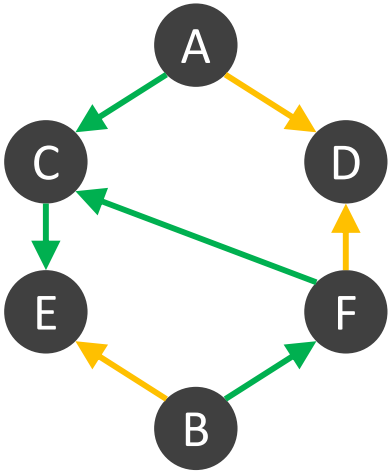
$\sigma_{G_i}(S)$: the influence spread of S in G_i

$$s_u = \frac{1}{R}\sum_{j=1}^{R} \sigma_{G_i}(S \cup \{u\})$$

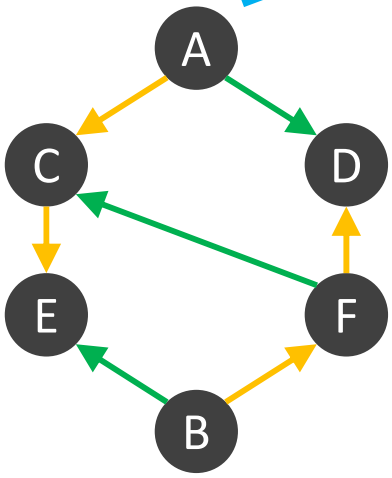
$$S = S \cup \{argmax_{u \in V \backslash S}\{s_u\}\}$$



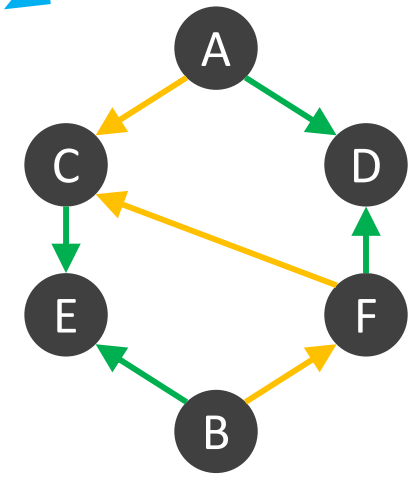
1st 2nd ... Rth



IC Result G_1



IC Result G_2



IC Result G_R

Greedy Algorithm: In Selecting the i^{th} Seed

$$\sigma(S) \approx \frac{1}{R}\sum_{i=1}^{R}\sigma_{G_i}(S)$$

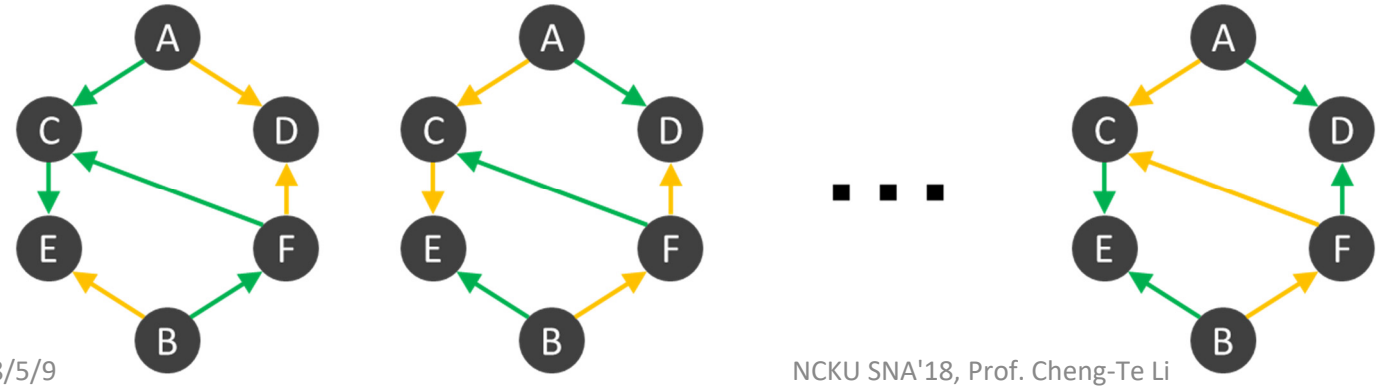
$\sigma_{G_i}(S)$: the influence spread of S in G_i based on IC model

$R = 20000$

	v	$\sigma_{G_1}(\{v\})$	...	$\sigma_{G_R}(\{v\})$	$\sigma(\{v\})$
R	A	3	...	2	2.4
R	B	4	...	2	2.8
R	C	2	...	2	1.6
R	D	1	...	1	1
R	E	1	...	1	1
R	F	3	...	2	2.2

10^6

For each node v, we need to run R=20000 IC simulation



In selecting each seed, IC will be run NR times in total ($N = |V|$). To select k seeds, IC will be run kNR times

Properties of $f(S)$

- Non-negative
- Monotone: $f(S \cup \{v\}) \geq f(S)$
- Submodular ← this is the key!
  - Let N be a finite set
  - A set function $f: 2^N \rightarrow \mathbb{R}$ is submodular iff

$$\forall S \subseteq T \subseteq N, \forall v \in N \backslash T$$

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

That is, diminishing returns

- The marginal gain from adding an element to a set *S* is at least as high as the marginal gain from adding the same element to a superset of *S*

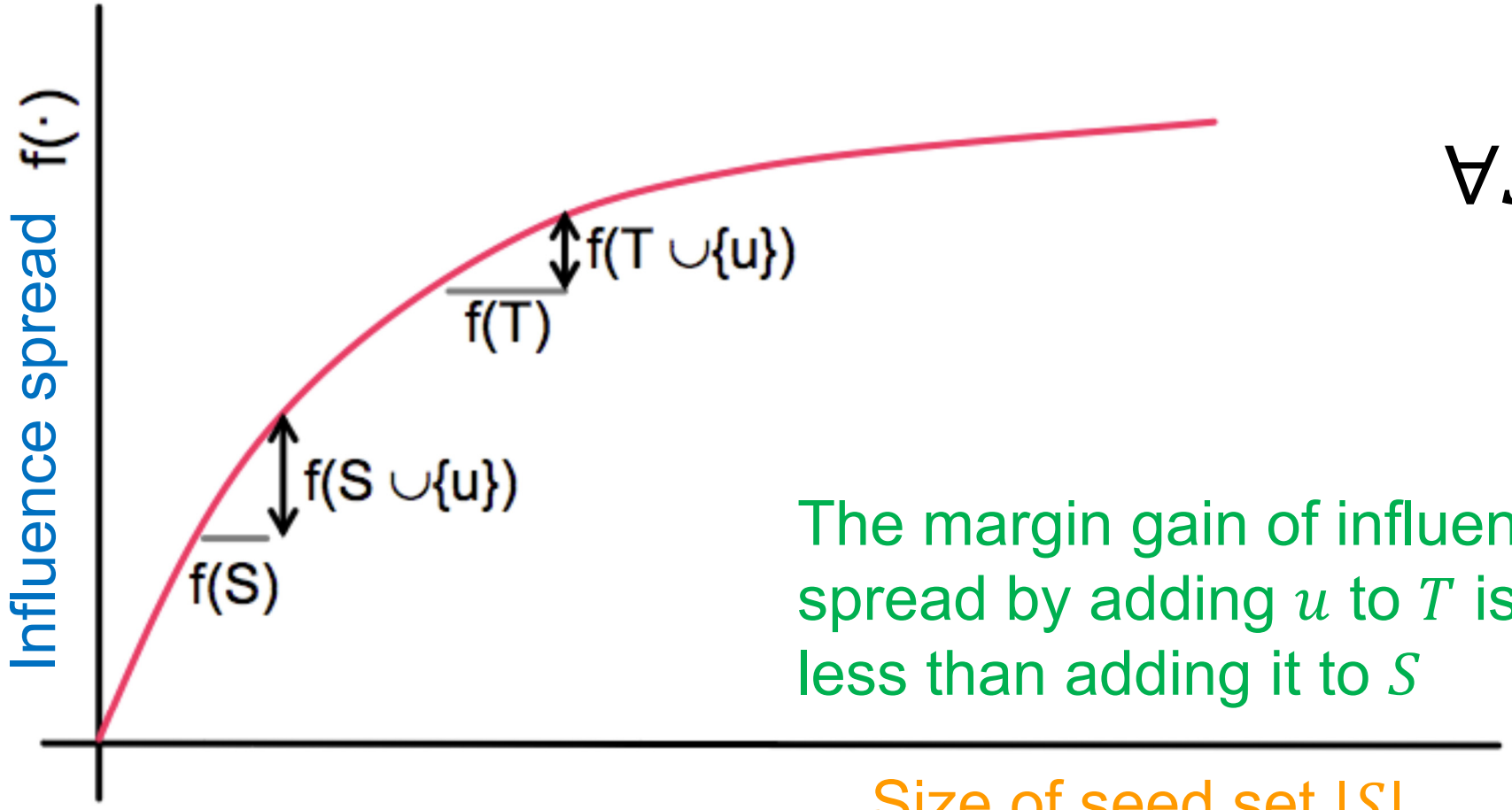
At the beginning of the propagation, information propagates faster than the following round

Diminishing Returns

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

Gain of adding a node to a small set

Gain of adding a node to a large set



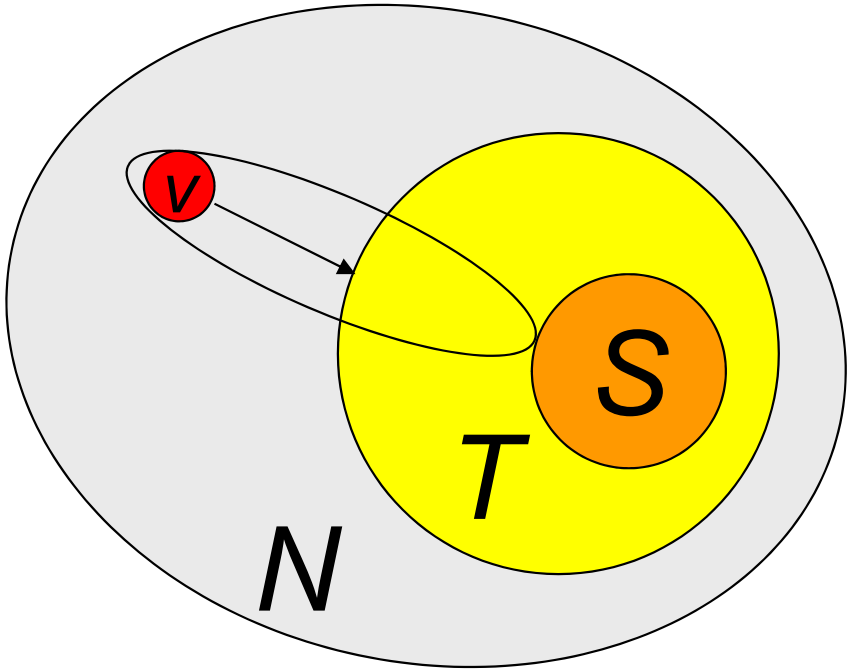
$$\forall S \subseteq T$$

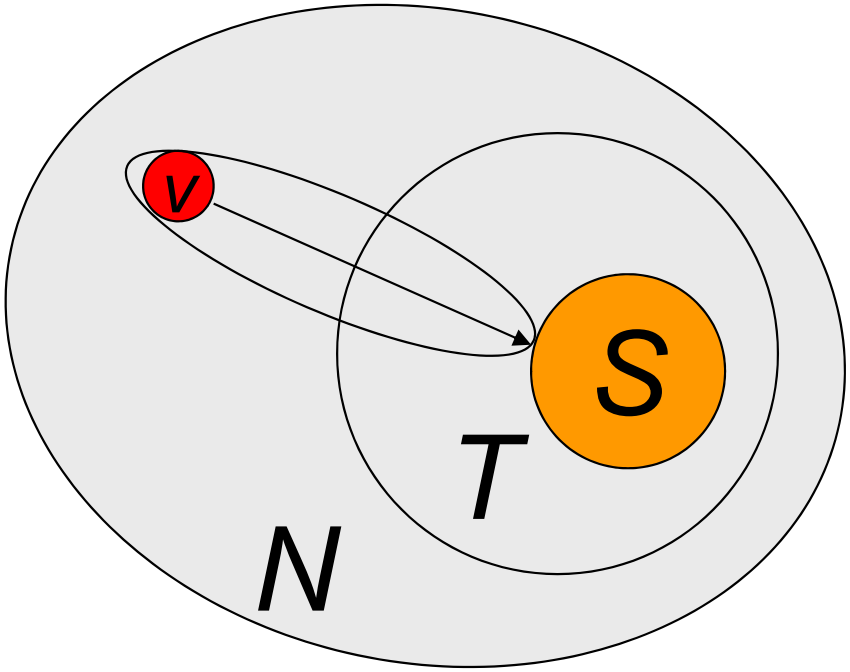
The margin gain of influence spread by adding u to T is less than adding it to S

An Illustration for Submodular

$$\forall S \subseteq T \subseteq N, \forall v \in N \backslash T$$

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



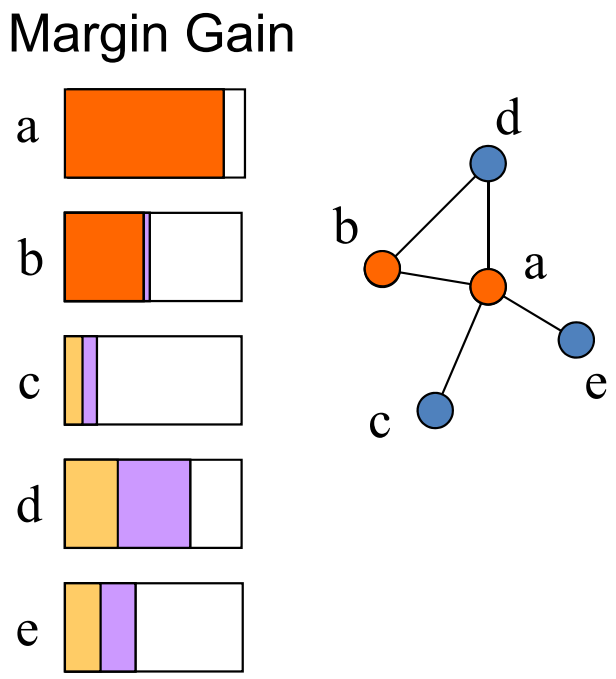


Good News

- Greedy Algorithm

  Add node v to S that maximizes $f(S \cup \{v\}) - f(S)$ for the k iterations

- How good or bad it is?
  - Theorem: the greedy algorithm is a $1 - 1/e$ approximation *
  - The resulting set S activates **at least** $(\mathbf{1 - 1/e})\sim\mathbf{63\%}$ of the number of nodes that any size-k S could act $f(S) \geq (1-1/e) \cdot f(S^*)$



S^* is the set that maximizes the value of f over all k-element sets

* G. Nemhauser, L. Wolsey, M. Fisher.
An analysis of the approximations for maximizing submodular set functions, 1978.

Proof: Greedy has 1-1/e approximation

- Notation:
  - S^*:best solution
  - S^g:the seed from Greedy algorithm
  - S_i^*:the best i solution $\{s_1^*, \ldots\ldots, s_i^*\}$
  - S_i^g:the i seed from Greedy $\{s_1, s_2 \ldots s_i\}$

Proof: Greedy has 63% approximation

- $f(S^*) \leq f(S_i^g \cup S^*)$ ➔ by monotonic

$$= f(S_i^g \cup S_{k-1}^* \cup \{s_k^*\}) \downarrow \text{by submodularity}$$

$$\leq f(S_i^g \cup \{s_k^*\}) - f(S_i^g) + f(S_i^g \cup S_{k-1}^*)$$

greedy

$$\leq f(S_{i+1}^g) - f(S_i^g) + f(S_i^g \cup S_{k-1}^*)$$

repeating k times $\leq k\left(f(S_{i+1}^g) - f(S_i^g)\right) + f(S_i^g)$

Proof: Greedy has 63% approximation

- Rearrange the inequality
- We have $f\left(S_{i+1}^{g}\right) \geq \left(1-\frac{1}{k}\right) f\left(S_{i}^{g}\right)+\frac{f(S^{*})}{k}$
- Multiplying by $\left(1-\frac{1}{k}\right)^{k-i-1}$ and add up all inequalities for i=0~k-1

Proof: Greedy has 63% approximation

0

- $\left(1-\frac{1}{k}\right)^{k-1} * f\left(S_1^g\right) \geq \left(1-\frac{1}{k}\right)^{k} f\left(S_0^g\right) + \frac{f(S^*)}{k} * \left(1-\frac{1}{k}\right)^{k-1}$
- $\left(1-\frac{1}{k}\right)^{k-2} f\left(S_2^g\right) \geq \left(1-\frac{1}{k}\right)^{k-1} f\left(S_1^g\right) + \frac{f(S^*)}{k} * \left(1-\frac{1}{k}\right)^{k-2}$
- .....
- $\left(1-\frac{1}{k}\right)^{0} * f\left(S_k^g\right) \geq \left(1-\frac{1}{k}\right)^{1} f\left(S_{k-1}^g\right) + \frac{f(S^*)}{k} * \left(1-\frac{1}{k}\right)^{0}$
- ---------- (+)

1

Proof: Greedy has 1-1/e approximation

- $f(S^g) = f(S_k^g)$

$$\geq \sum_{i=0}^{k-1} \left(1 - \frac{1}{k}\right)^{k-i-1} * \frac{f(S^*)}{k}$$

$$= \left(1 - \left(1 - \frac{1}{k}\right)^k\right) f(S^*)$$

$$\geq \left(1 - \frac{1}{e}\right) f(S^*)$$

Approximately 63%

Bad News

- It is NP-hard to determine the optimum for influence maximization for both independent cascade model and linear threshold model

Initialize $S = \emptyset$ and $R = 20000$

for $i = 1$ to k do:

  for each node $u \in V \backslash S$ do:

    $s_u = 0$

    for $round = 1$ to R do:

      $s_u \mathrel{+}= f(S \cup \{u\})$ // run IC or LT model

    $s_u = s_u / R$

  $S = S \cup \{argmax_{u \in V \backslash S}\{s_u\}\}$

Output S

Outer loops: time complexity of O(nk), where n=|V| is #nodes and can be very large

Inner loop: to have accurate influence spread, R is required to be very large, say 20,000

Total Complexity = O(knRm)

m=|E|

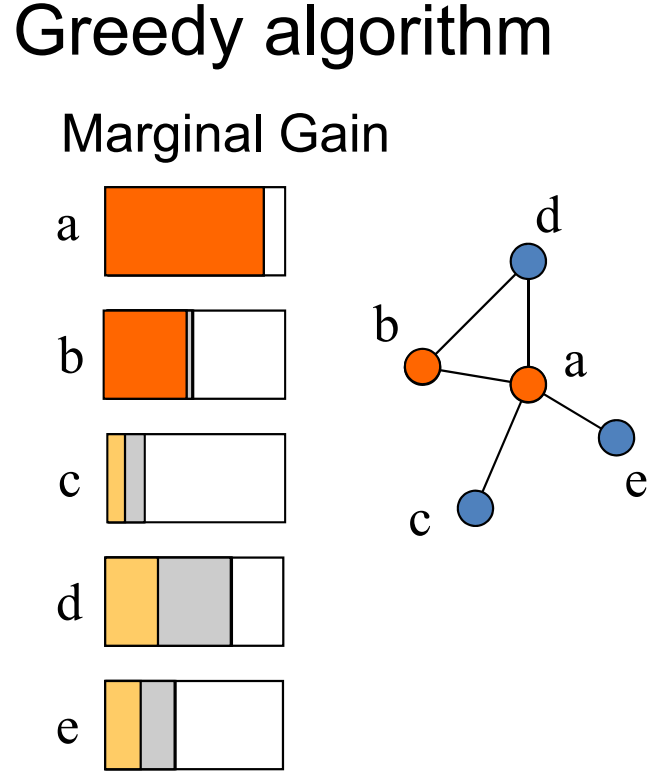
Cost-Effect Lazy Forward (CELF) Greedy

- Recall: Submodular

$$\forall S \subseteq T \subseteq N, \forall v \in N \backslash T$$

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

- When adding a node v to seed set S, the marginal gain of adding v is larger if S is smaller
- → A large number of nodes need not to be re-evaluated

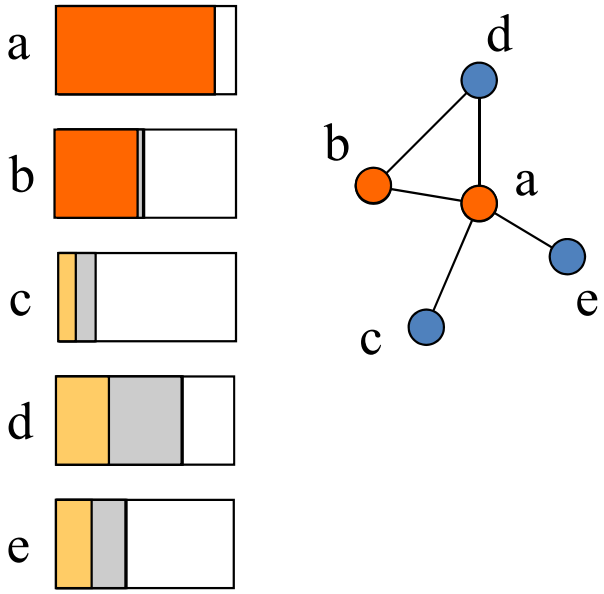


原本影響力(influence spread)小的node，加入更大的Seed Set後，所帶來的影響力(gain)只會更小!

Cost-Effect Lazy Forward (CELF) Greedy

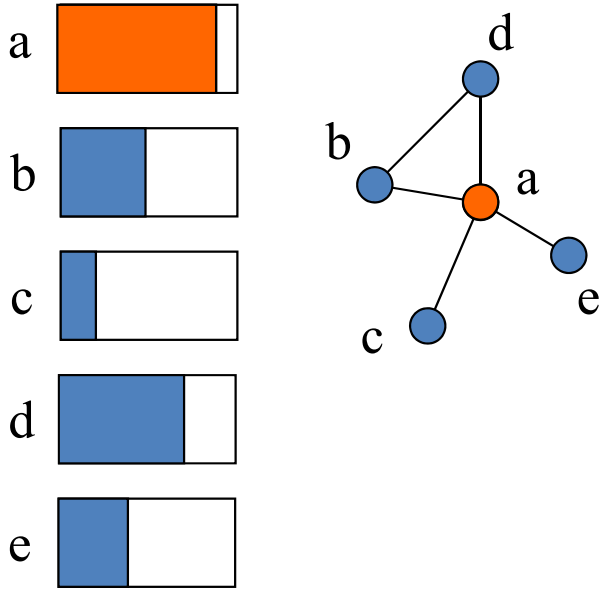
Greedy algorithm

Marginal Gain



CELF algorithm

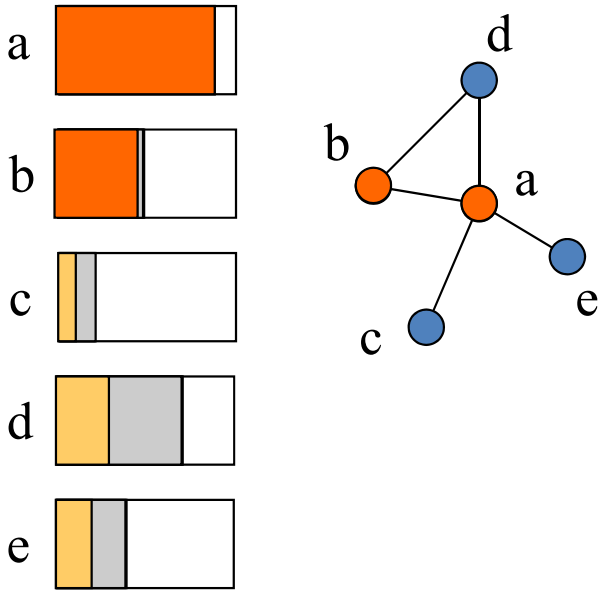
Marginal Gain



Cost-Effect Lazy Forward (CELF) Greedy

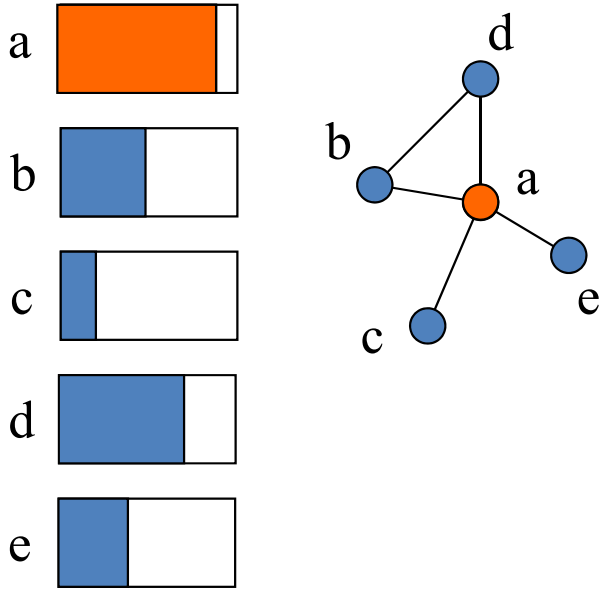
Greedy algorithm

Marginal Gain



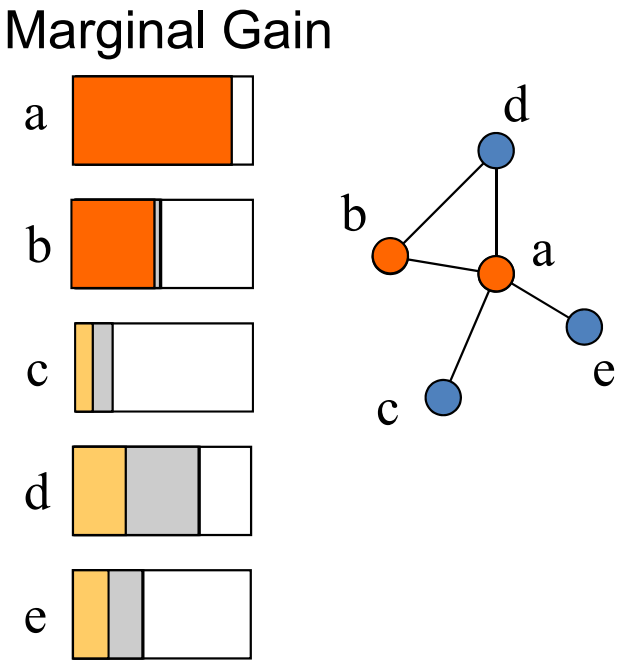
CELF algorithm

Marginal Gain

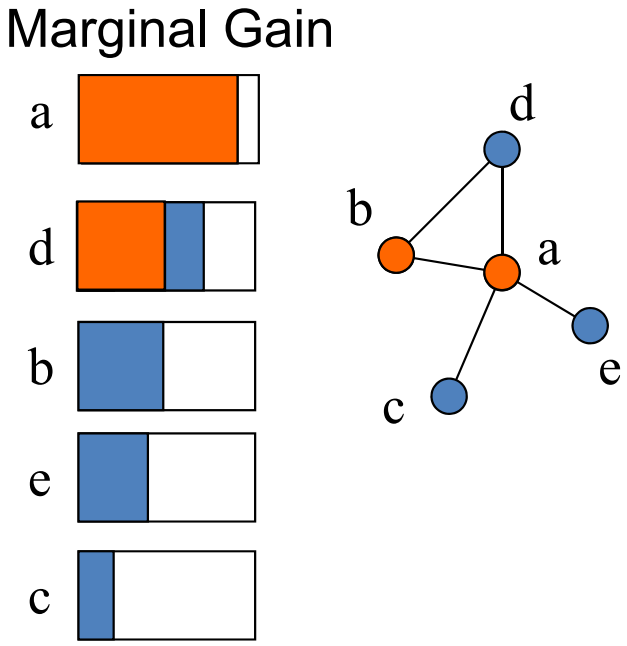


Cost-Effect Lazy Forward (CELF) Greedy

Greedy algorithm



CELF algorithm



Advantage: by setting up an upper bound, CELF reduces the Monte-Carlo calls and improves the greedy algorithm by up to 700 times

CELF Greedy Algorithm

Priority Queue

Initialize $S = \emptyset$, $Q = \emptyset$, and $R = 20000$

for each node $u \in V$ do:

$u.marGain = GetInf(\{u\}, R)$

$u.flag = false$ // flag = true if the node's marginal gain have been updated

add u into Q by $u.marGain$ in descending order

while $|S| < k$ do:

$u = Q[top]$

if $u.flag = true$ then:

$S = S \cup \{u\}$

$Q = Q \setminus \{u\}$

Else:

$u.marGain = f(S \cup \{u\}, R) - f(S, R)$

$u.flag = true$

Resort Q by $u.marGain$ in descending order

Output S

GetInf(S, R) // subprocedure

for $round = 1$ to R do:

s_u += $f(S)$ // run IC or LT model

$s_u = s_u / R$

Return s_u

If the re-sorted (updated margin gain) node u is still at the top of PriorityQueue, it is truly influential

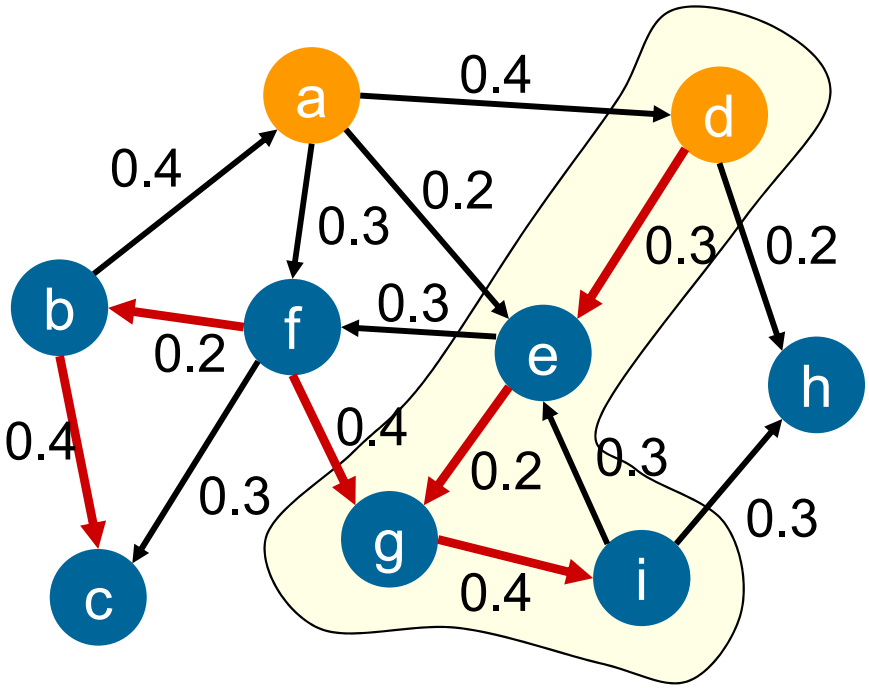
Alternative View: Pre-Flip (for IC model)

- We can **pre-flip** all coins
  and reveal results immediately
- Active nodes in the end are reachable via red paths (consists of live-edges) from initially targeted nodes
- Let $f(S) =$ size of the set reachable by live-edge paths

When selecting each new seed u^* ...

Original:
Run R IC simulations **for each $u \in V \backslash S$**
Cost = nR times of IC simulations
$f(S \cup \{u\})$ = average active nodes over R IC

Pre-flip:
Generate R random graphs **one time for all nodes in $V \backslash S$** (No IC simulation)
Cost = R random graph generation
$f(S \cup \{u\})$ = average reachable nodes



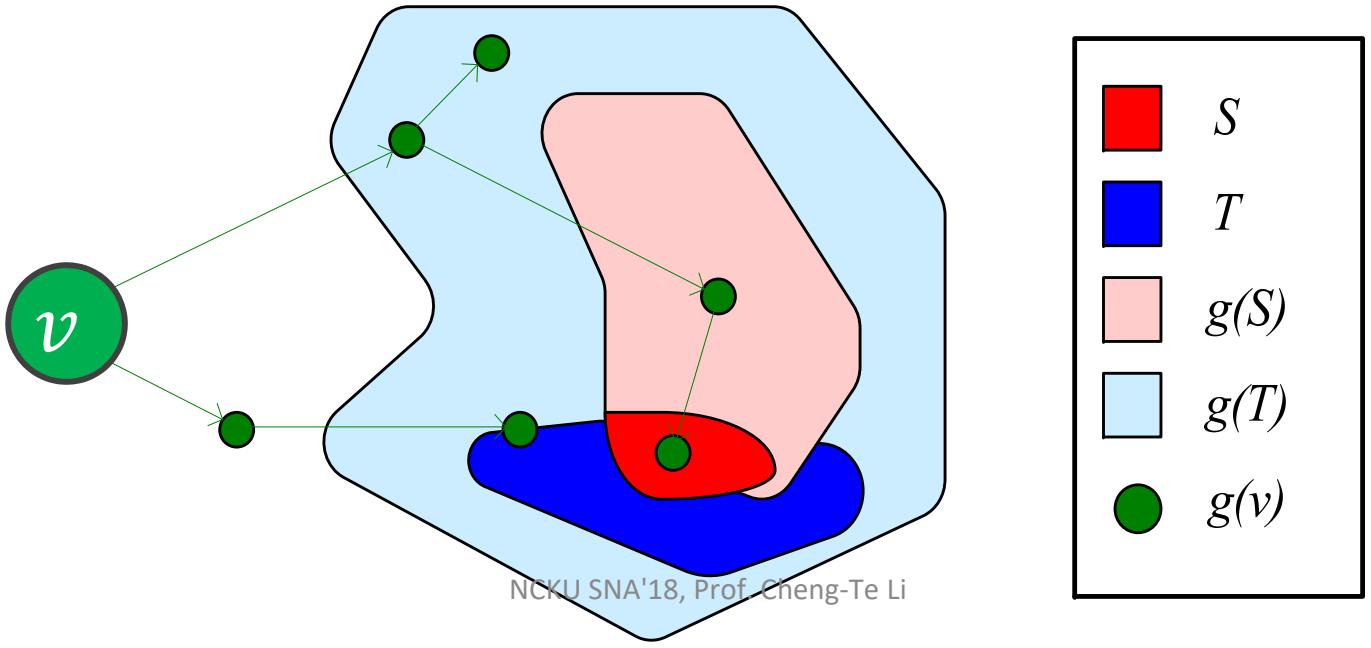
Submodularity for Pre-Flip Approach

- Fix "green graph" G:
  $g(S)$ is the number of nodes **reachable from *S*** in *G*
- Submodularity:

Given: $S \subseteq T$

$$g(T \cup \{v\}) - g(T) \leq g(S \cup \{v\}) - g(S)$$

$g(S \cup \{v\}) - g(S)$: marginal number of nodes reachable by $S \cup \{v\}$



In selecting a new seed...

Pre-Flip: Generate Random Graphs

Original: **for each node**
Run the IC simulation up to R times

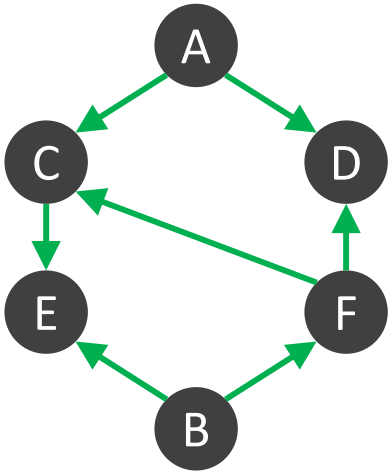
Original: **for all nodes**
Generate R random graphs

$$\sigma(\{B\}) = \frac{4 + 1 + \cdots + 2}{R}$$

Edge e lives with p_e

Live edge: success
Blocked edge: failure

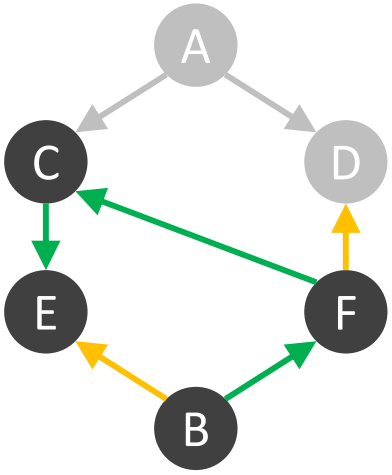
Original network G



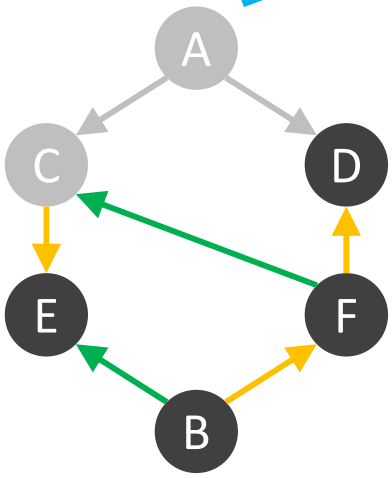
Assume $k = 2$ (aim to select 2 seeds),
we already have $S = \{A\}$
Now we are finding the 2nd seed

Compute influence spread $\sigma(S)$
≈
Count the number of nodes **reachable from S** by BFS in these random graphs

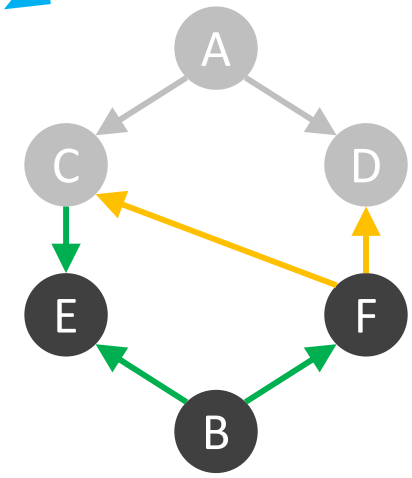
1st 2nd ... Rth



Random Graph G_1



Random Graph G_2



Random Graph G_R

Pre-Flip **New Greedy** for IC Model

Initialize $S = \emptyset$ and $R = 20000$

for $i = 1$ to k do:

$s_v = 0$ for all $v \in V \backslash S$

for $round = 1$ to R do:

compute G' by removing edges from G with $1 - p$

compute $Reach_{G'}(S)$

compute $Reach_{G'}(\{v\})$ for all $v \in V$

for each $v \in V \backslash S$ do:

if $v \notin \boldsymbol{Reach_{G'}(S)}$ then:

$s_v \mathrel{+}= |Reach_{G'}(\{v\})|$

$s_v = s_v / R$ for all $v \in V \backslash S$

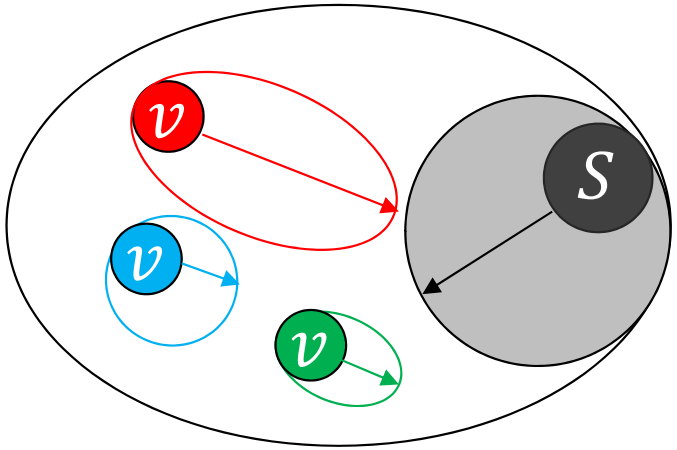
$S = S \cup \{argmax_{v \in V \backslash S}\{s_v\}\}$

Output S

15-34% faster than the original greedy!!!

Generate Pre-Flip Random Graph

Find the Reachable Set of v by performing BFS



Pre-Flip: To Approximate the Exact Influence Spread $\sigma(S)$

$$\sigma(S) \approx \frac{1}{R}\sum_{i=1}^{R}\sigma_{G_i}(S)$$

$\sigma_{G_i}(S)$ = # of nodes **reachable** from S on G_i

$R = 20000$

v	$\sigma_{G_1}(\{v\})$	...	$\sigma_{G_R}(\{v\})$	$\sigma(\{v\})$
A	3	...	2	2.4
B	4	...	2	2.8
C	2	...	2	1.6
D	1	...	1	1
E	1	...	1	1
F	3	...	2	2.2

10^6

CHALLENGE

Computing this table as **fast** as possible

Equivalence: Greedy and NewGreedy

- Multiple times of IC simulation for influence spread estimation
  - Approximate true values of influence spread by realizations

Method	Main Idea	Advantage	Disadvantage
Greedy	Faithfully reflect the process of influence propagation	Relatively low time complexity for each node	Estimate one seed set at a time. High complexity.
New Greedy	Removing each edge (u, v) from G with probability $1-p_{uv}$	Can estimate any seed set simultaneously	Relatively high time complexity in each round
CELF Greedy	Maintain only those with higher marginal gain	Can speed up Greedy by up to 700 times	Still very slow when the network is very large

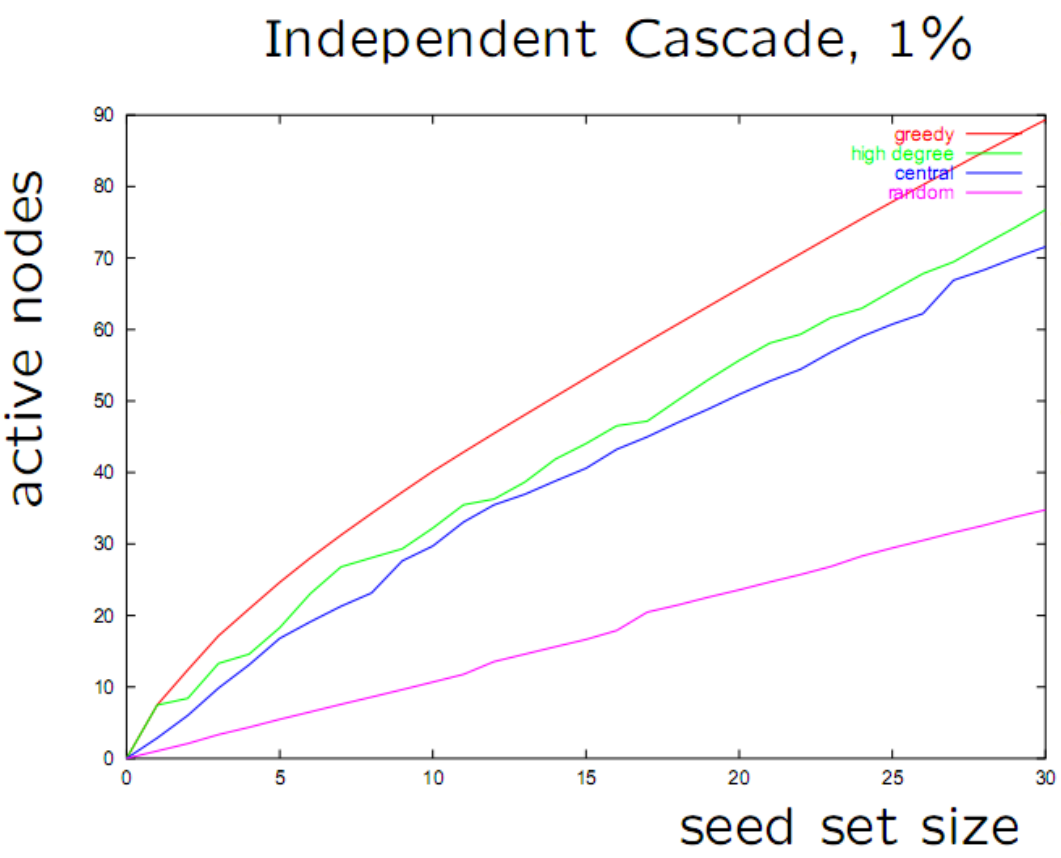
Centrality Heuristic Methods

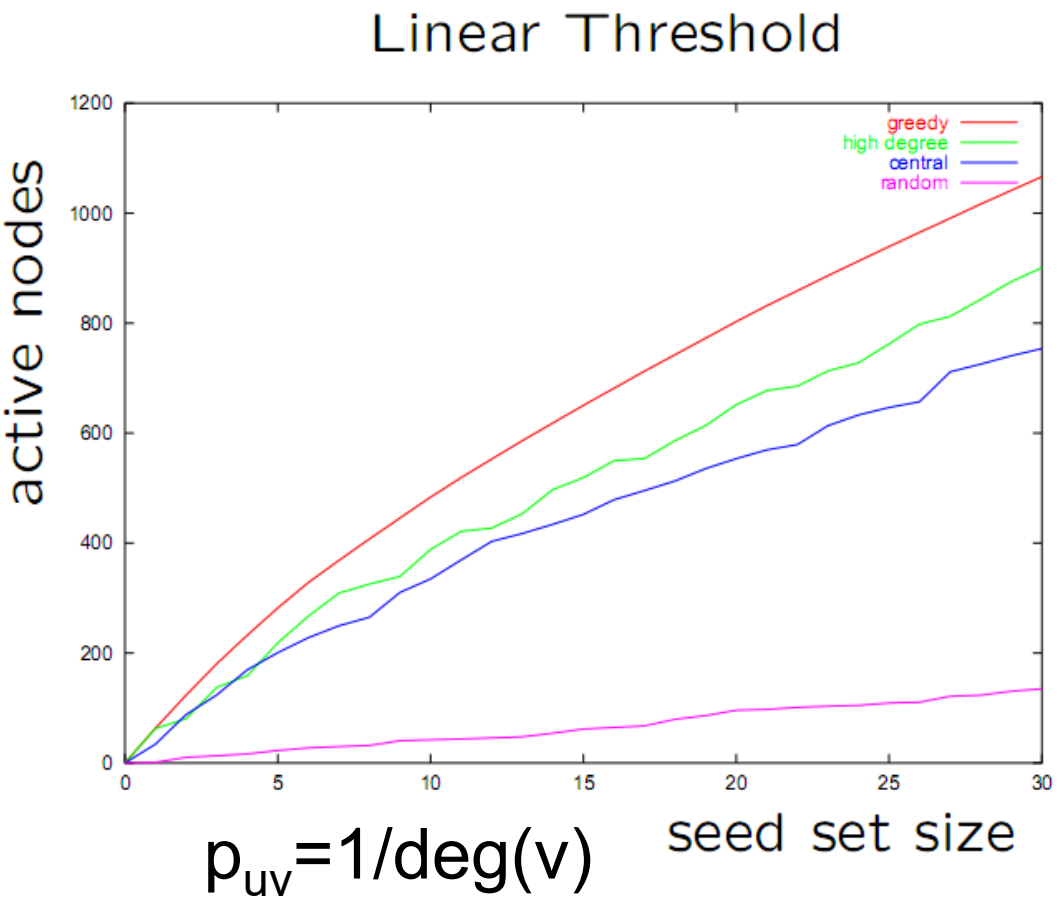
- Low-Distance heuristic (Closeness Centrality)
  - Consider the nodes with the shortest paths to other nodes as seed nodes
  - Intuition: Individuals are more likely to be influenced by those who are closely related to them
- High-Degree Heuristic (Degree)
  - Choose the seed nodes according to their degree.
  - Intuition: The nodes with more neighbors would arguably tend to impose more influence upon its direct neighbors

Empirical Results

- Use arXiv high-energy physics collaboration graph
- Compare greedy algorithm, degree centrality, closeness centrality, and random selection

10,748 nodes
53,000 edges



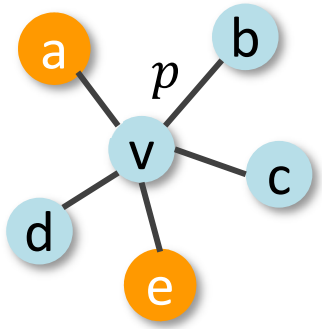


$p_{uv}=1/deg(v)$

Degree Discount Heuristic

- Basic Idea
  - Consider edge (u, v), with u in the seed set S and v being considered. Since u is in the seed set, by taking network effect into consideration, we should not count edge (u, v) towards v's degree
- Assumption
  - In independent cascade model, when propagation probability p is small, we may ignore indirect influence of v to multi-hop neighbors, and focus on the direct influence of v to its immediate neighbors

Degree Discount Heuristic



- d_v: degree of node v
- t_v: number of v's neighbors that in seed set S

$t_v = 2$

$d_v = 5$

Probability that v is influenced by its immediate neighbors: $1 - (1 - p)^{t_v}$

→ Selecting v does not contribute additional influence

(執行完IC model後, 已被影響者便不再具備影響力)

Probability that v is **NOT** influenced by its immediate neighbors: $(1 - p)^{t_v}$

→ Selecting v will in expectation influence $1 + (d_v - t_v) \times p$ nodes

Degree Discount Heuristic

- The expected number of additional nodes influenced by selecting v as a seed:

$$[1-(1-p)^{t_v}] \times 0 + [(1-p)^{t_v}] \times [1+(d_v - t_v) \times p] \quad \textbf{(A)}$$

- If no neighbors of v is selected as seed (i.e., $t_v = 0$), the answer above becomes

$$1 + d_v \times p \quad \textbf{(B)}$$

- Let dc_v be the degree discount caused by neighbors in the seed set, then

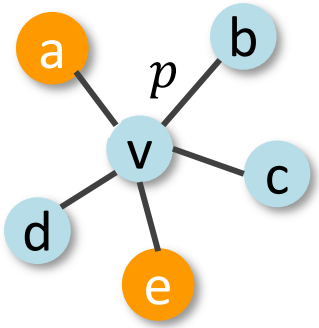
$$dc_v \times p = B - A$$

$$dc_v = 2t_v + (d_v - t_v)t_v p$$

Degree Discount Heuristic Algorithm

Algorithm 4 DegreeDiscountIC(G, k)

1: initialize $S = \emptyset$
2: **for** each vertex v **do**
3: compute its degree d_v
4: $dd_v = d_v$
5: initialize t_v to 0
6: **end for**
7: **for** $i = 1$ to k **do**
8: select $u = \arg\max_v \{dd_v \mid v \in V \setminus S\}$
9: $S = S \cup \{u\}$
10: **for** each neighbor v of u and $v \in V \setminus S$ **do**
11: $t_v = t_v + 1$ Update the number of neighbor seeds
12: $dd_v = d_v - 2t_v - (d_v - t_v)t_v p$ $dd_v = d_v - dc_v$
13: **end for** Apply degree discount heuristic
14: **end for**
15: output S



$t_v = 2$

$d_v = 5$

Performance Comparison

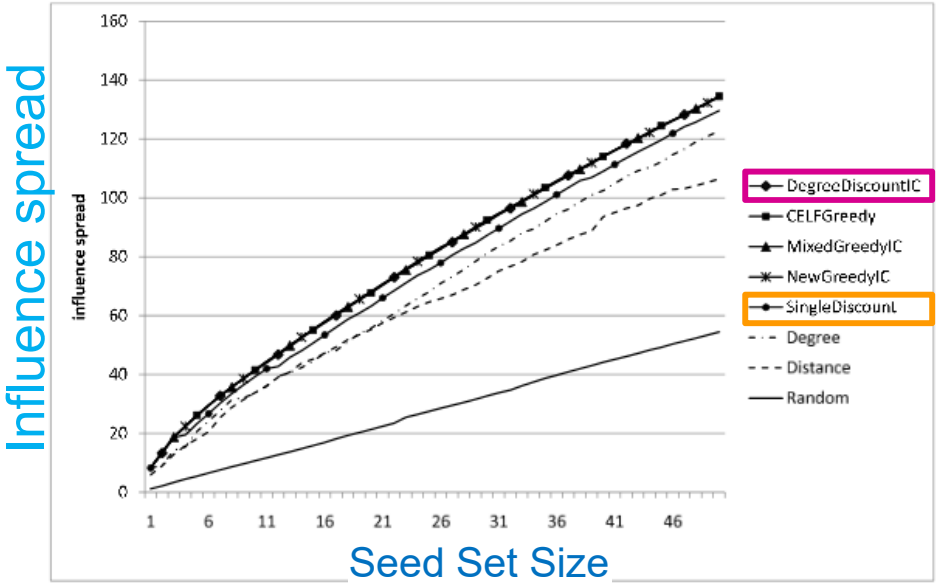


Figure 1: Influence spreads of different algorithms on the collaboration graph NetHEPT under the independent cascade model ($n = 15,233$, $m = 58,891$, and $p = 0.01$).

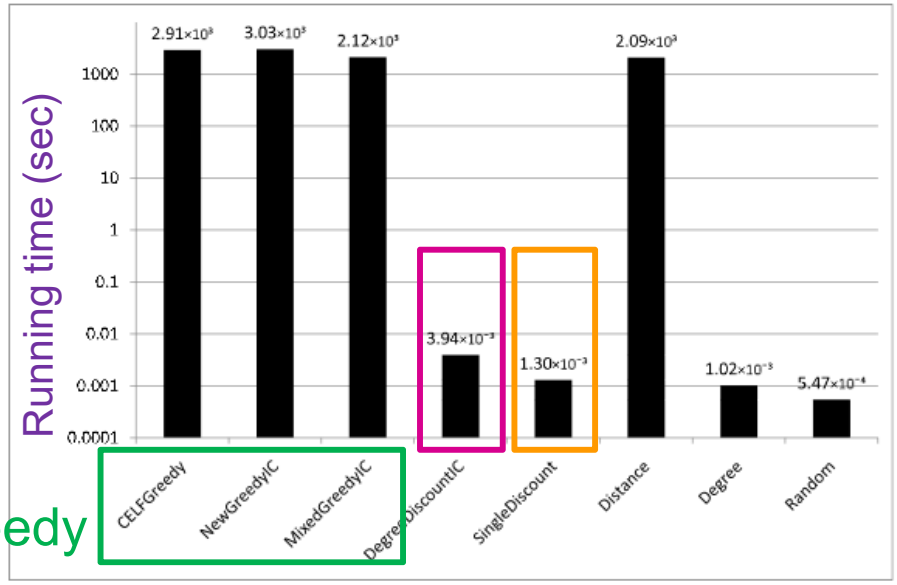


Figure 3: Running times of different algorithms on the collaboration graph NetHEPT under the independent cascade model ($n = 15,233$, $m = 58,891$, $p = 0.01$, and $k = 50$).

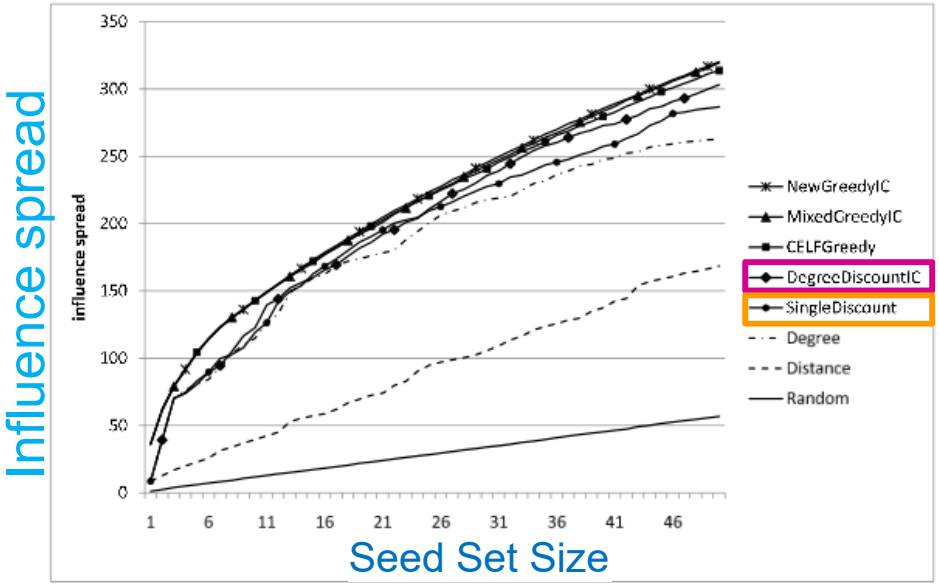


Figure 2: Influence spreads of different algorithms on the collaboration graph NetPHY under the independent cascade model ($n = 37,154$, $m = 231,584$, and $p = 0.01$).

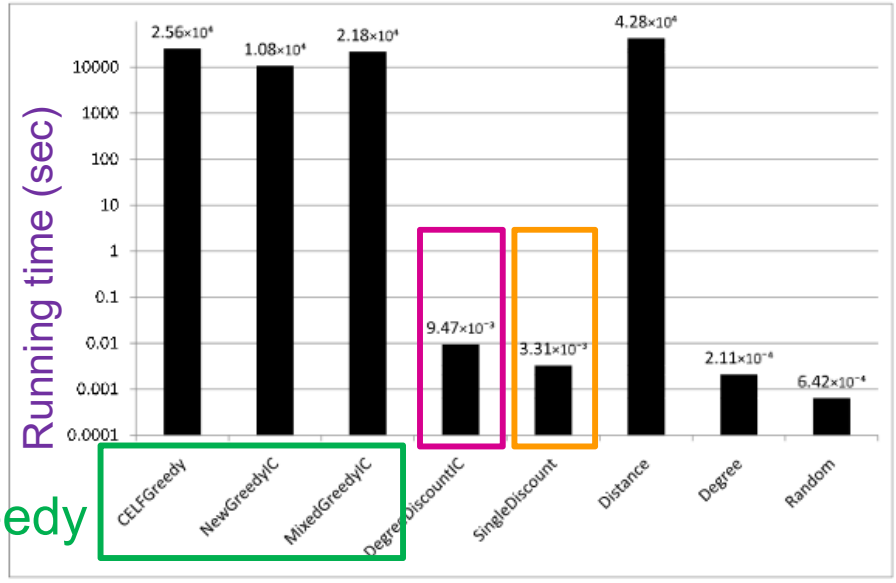
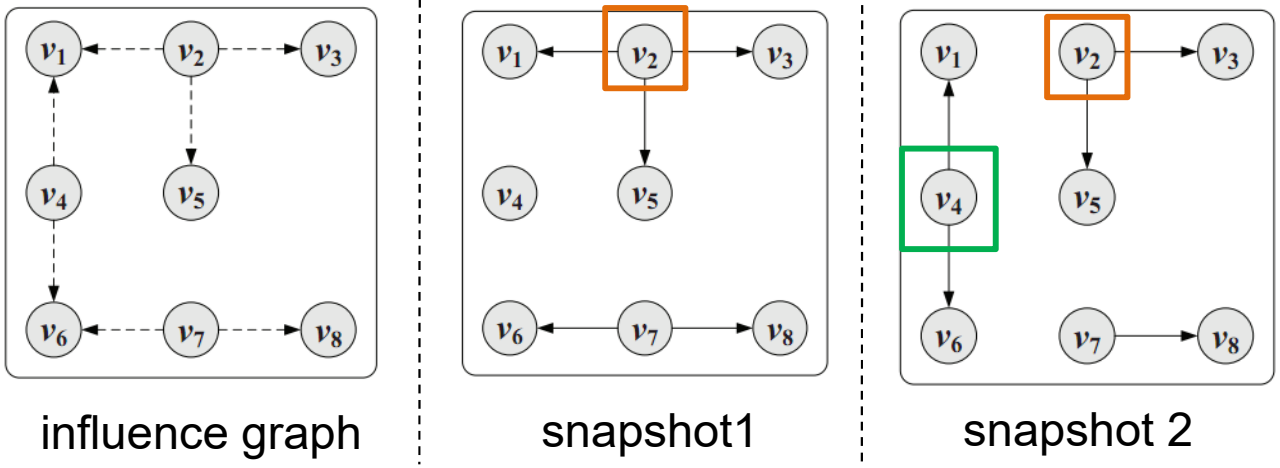


Figure 4: Running times of different algorithms on the collaboration graph NetPHY under the independent cascade model ($n = 37,154$, $m = 231,584$, $p = 0.01$, and $k = 50$).

Problems in Existing Greedy Algorithms

- In existing greedy algorithms
  - A risk of unguaranteed submodularity and monotonicity of influence spread function
  - Caused by using different results of Monte Carlo simulation across different influence spread estimation
  - A very large value of R is required, e.g. R=20000



$$I(S_0 \cup \{v_4\}) - I(S_0) = I(\{v_4\}) - I(\varnothing) = 1$$
$$I(S_1 \cup \{v_4\}) - I(S_1) = I(\{v_2, v_4\}) - I(\{v_2\}) = 3$$

Submodularity is broken!

StaticGreedy algorithm

- Basic idea: to always use the same snapshots for influence spread estimation
  - Influence spread function is submodular and monotone
  - A small value of R is required, e.g. R=100

Part 1: Generate **R** static snapshots

Part 2: Greedy selection

Algorithm 1 StaticGreedy(G,k,R)

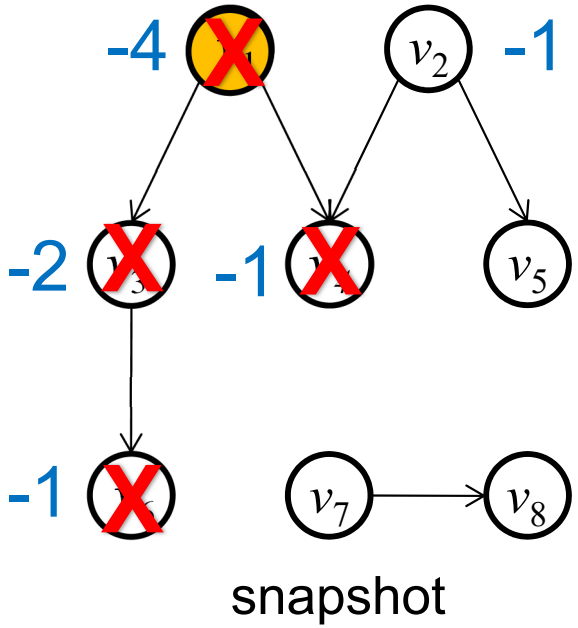
1: initialize $S = \varnothing$
2: **for** $i = 1$ to R **do**
3: generate snapshot G'_i by removing each edge $\langle u, v\rangle$ from G with probability $1 - p(u, v)$
4: **end for**
5: **for** $i = 1$ to k **do**
6: set $s_v = 0$ for all $v \in V \setminus S$ //s_v stores the influence spread after adding node v
7: **for** $j = 1$ to R **do**
8: **for all** $v \in V \setminus S$ **do**
9: s_v += $|R(G'_j, S \cup \{v\})|$ //$R(G'_j, S \cup \{v\})$ is the influence spread of $S \cup \{v\}$ in snapshot G'_j
10: **end for**
11: **end for**
12: $S = S \cup \{\arg\max_{v \in V \setminus S} \{s_v/R\}\}$
13: **end for**
14: output S

Speed-up StaticGreedy

- A dynamic update strategy: Calculate the marginal gain in an efficient incremental manner

at each step t, for each snapshot:

$$M(v) \leftarrow M(v) - |R(v) \cap R(v_t^*)|$$
$$R(v) \leftarrow R(v) - R(v) \cap R(v_t^*)$$



after select $v^* = v_1$

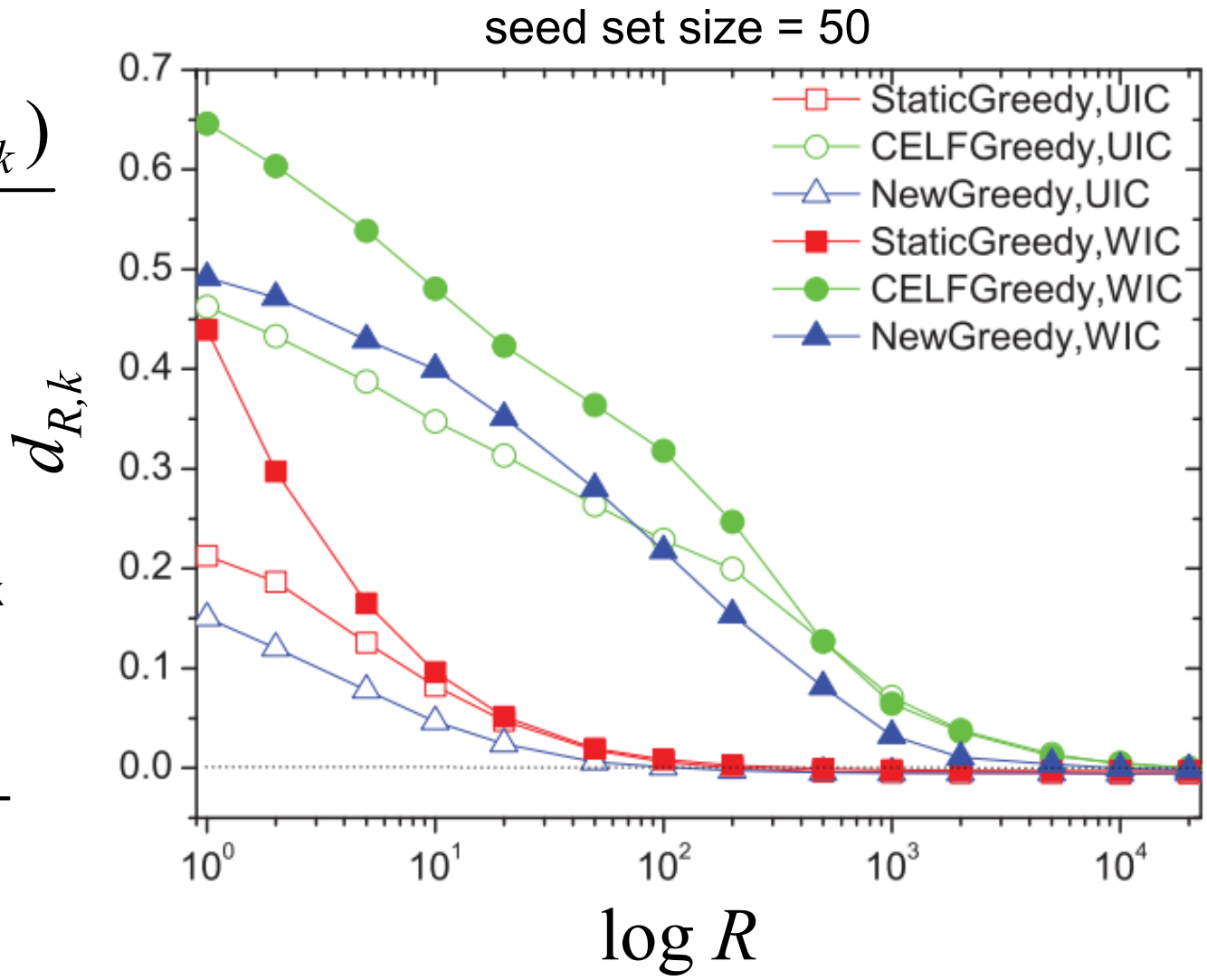
$M(v_1)=4$	$M(v_1)=0$
$M(v_2)=3$	$M(v_2)=2$
$M(v_3)=2$	$M(v_3)=0$
$M(v_4)=1$	$M(v_4)=0$
$M(v_5)=1$	$M(v_5)=1$
$M(v_6)=1$	$M(v_6)=0$
$M(v_7)=2$	$M(v_7)=2$
$M(v_8)=1$	$M(v_8)=1$

directly update

Comparison of Convergence Rate

- Provide approximation with a small value of R

$$d_{R,k} = \frac{I(S_k^*) - I(S_{R,k})}{I(S_k^*)}$$



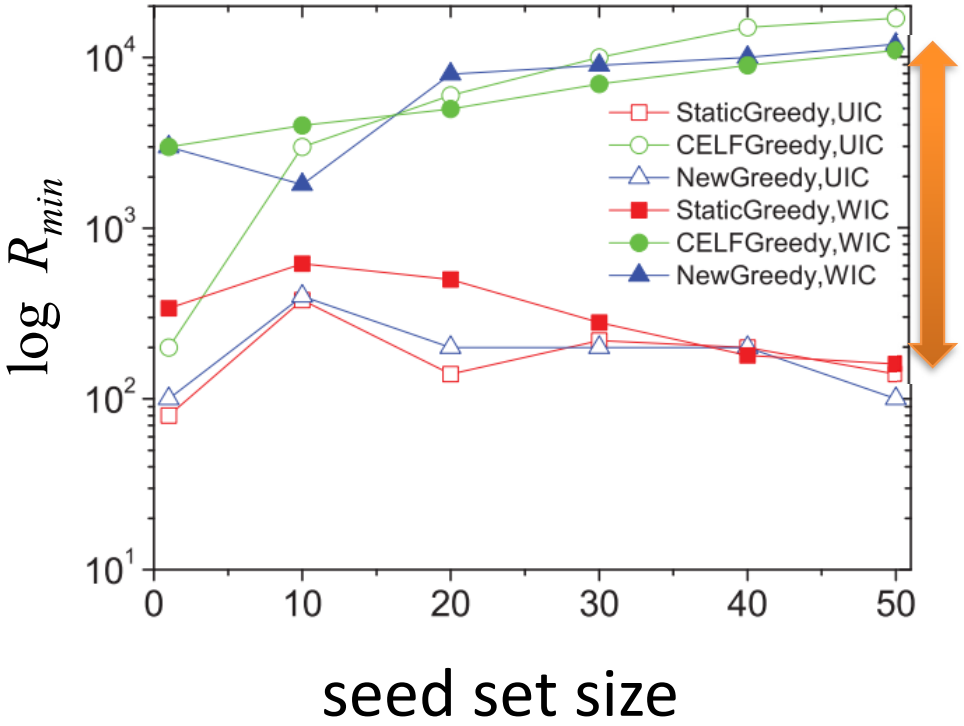
NetHEPT: a benchmark network
uniform independent cascade (UIC) model: p(u, v) = p = 0.01
weighted independent cascade (WIC) model: p(u, v) = 1/(# of in-neighbors of v)

Comparison of Scalability

Minimal R required

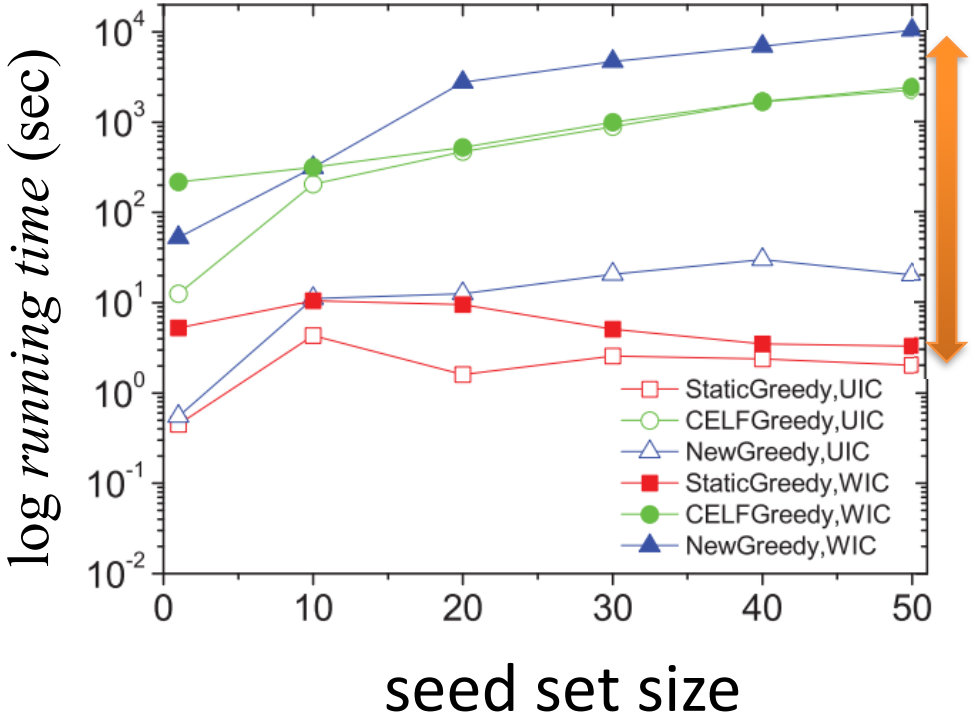
$$R_{\min} = \min\{R \mid d_{R,k} \le 0.005\}$$

≈10^2 times



Running time

≈10^3 times



R is significantly reduced ➡ Running time is significantly reduced

Problem in StaticGreedy

- Preprocessing: Generating random graphs

  → by Pre-Flip Techniques

- Greedy Seed Selection

$S \leftarrow \emptyset$

while $|S| < k$ do:

$\sigma(S \cup \{v\})$ ≈ $\sigma(S)$ ≈

$$t \leftarrow \underset{v \in V \setminus S}{\operatorname{argmax}} R(S \cup \{v\}) - R(S)$$

$$S \leftarrow S \cup \{t\}$$

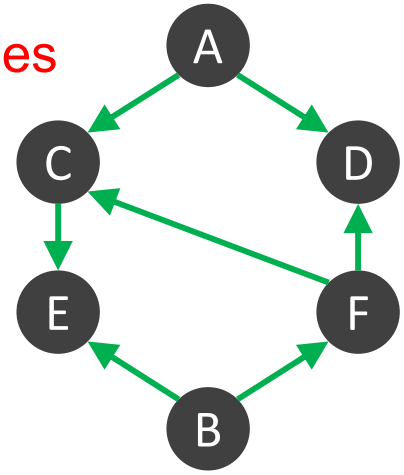
Find the reachable set via BFS may take too much time

Note:

1) R(X) is the approximated influence spread of node set X
2) R(X), computed by BFS, is reachable node set from node set X
3) R(X) is used to approximate $\sigma(X)$, which is the influence spread by original Greedy
4) We interchangeably use R(X) and σ(X) to denote influence spread of X

Recall: Pre-Flip in StaticGreedy

Original network G

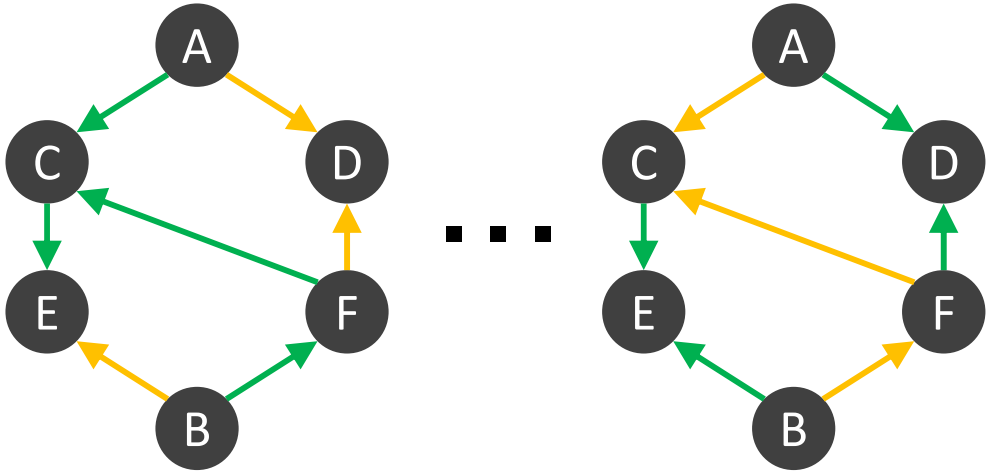


Run the IC simulation up to R times
= Generate R random graphs

Edge e lives with p_e

Live edge: success
Blocked edge: failure

1st ... Rth



Random Graph G_1 ... Random Graph G_R

Compute influence spread $\sigma(S)$
≈
Count the number of nodes **reachable from S** by BFS in these random graphs

$$\sigma(S) \approx \frac{1}{R}\sum_{i=1}^{R}\sigma_{G_i}(S)$$

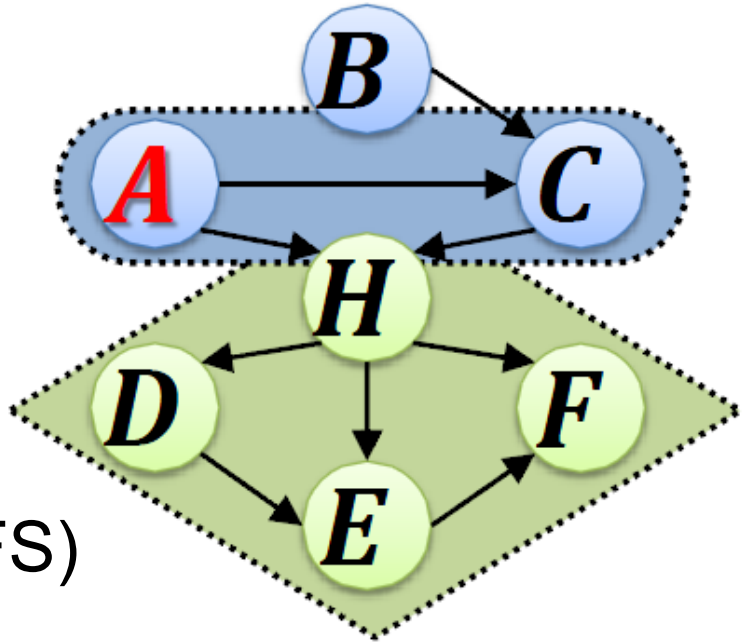
$R = 20000$

v	$\sigma_{G_1}(\{v\})$	...	$\sigma_{G_R}(\{v\})$	$\sigma(\{v\})$
A	3	...	2	2.4
B	4	...	2	2.8
C	2	...	2	1.6
D	1	...	1	1
E	1	...	1	1
F	3	...	2	2.2

10^6

Pruned BFS to Fast Obtain the Reachable Set

- Idea: Most BFSs are redundant
- Step 1: Smart Preprocessing
  - For each node H with **highest degree value**
  - Find H's **ancestors** and **descendants**
- Step 2: Pruning (BFS from v)
  If v is ancestors of H,
  we ignore descendants of H
  in the execution of BFS

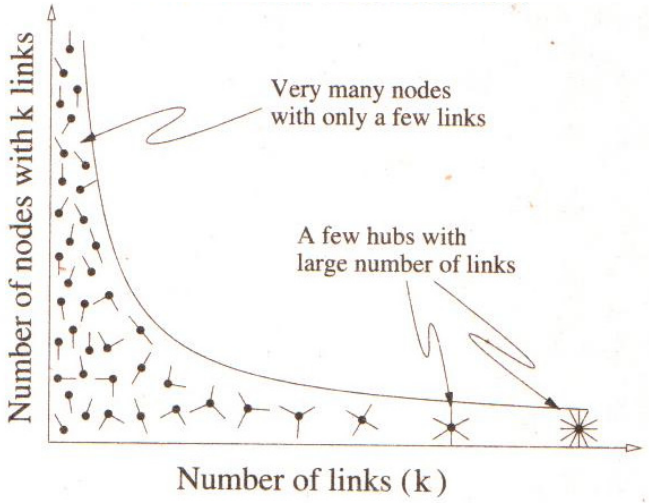


$\sigma_{G_i}(\{A\})$ = (# of nodes visited during BFS)
+ (# of descendants of H) = 2 + 4 = 6

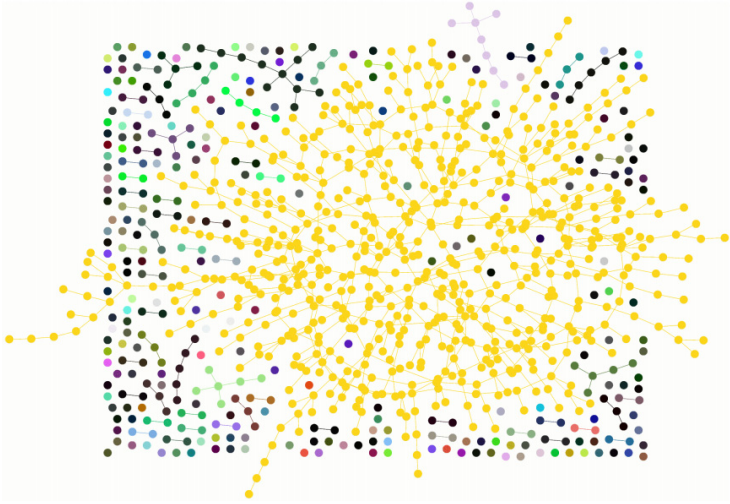
Is Pruned BFS Really Effective?

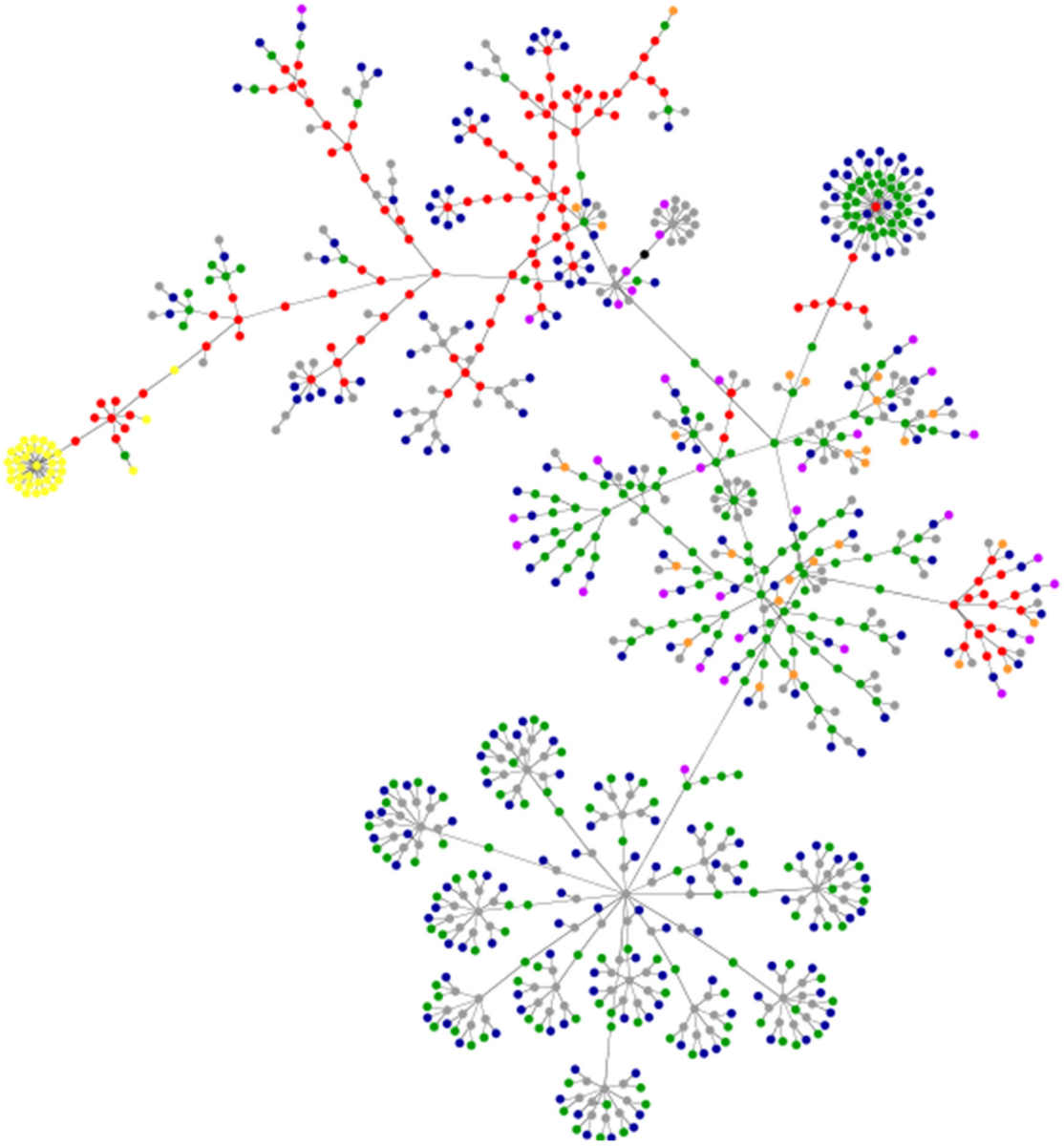
- Recall the essential properties of social networks

Power-Law Degree Distribution



Emergence of Giant Component

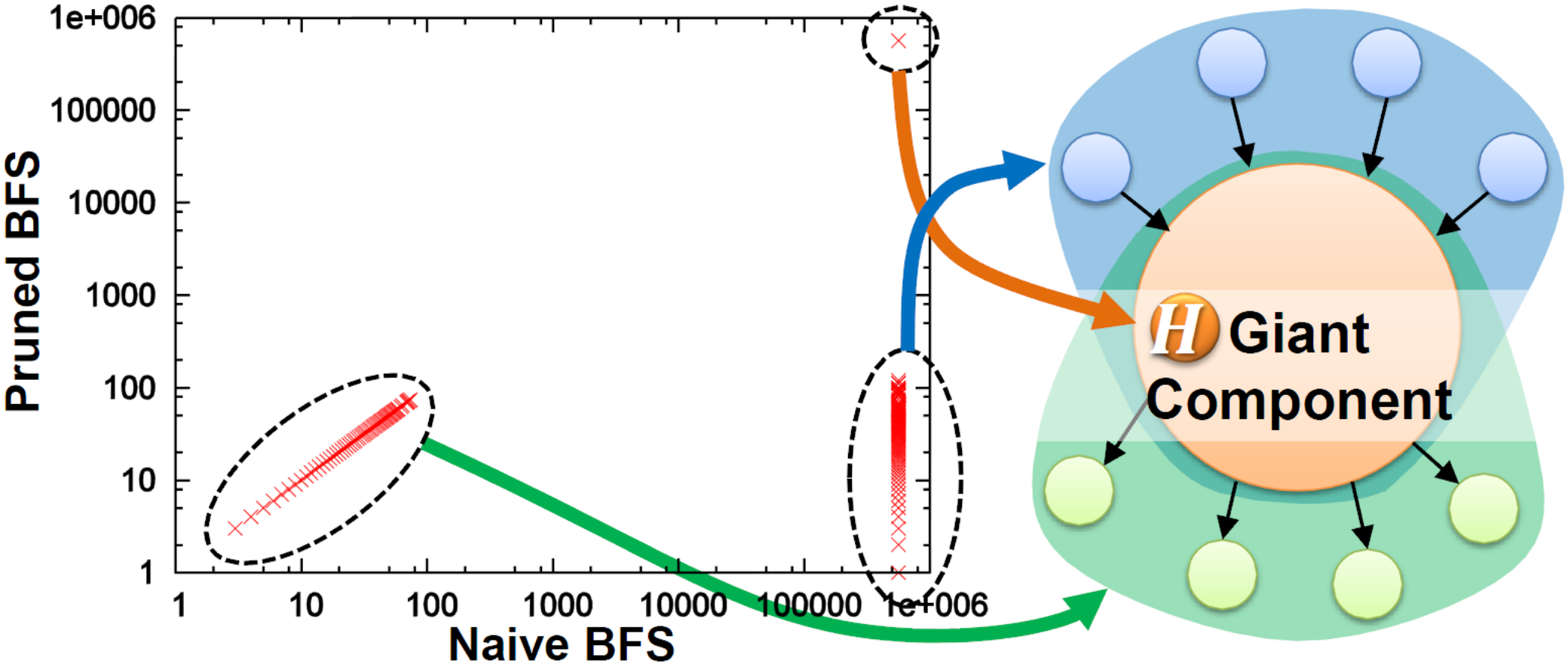




Effect of Pruned BFS on Social Networks

(LiveJournal social graph, $V = 4.8\text{M}$, $E = 69\text{M}$, $p_e = 0.1$ $(\forall e)$

- # of vertices visited during Naive & Pruned BFSs



- Average # of visited vertices (from each vertex):
  - **400,000** (Naive BFS) ⇨ **6** (Pruned BFS)

Summary of Influence Max Solutions

- Propose an influence maximization algorithm to *solve the scalability-accuracy dilemma*

	Algorithm		Accuracy	Scalability
Simulation	Greedy	[Kempe, KDD'03]	Guaranteed	low
	Greedy CELF	[Leskovec, KDD'07]	Guaranteed	low
	Greedy CELF++	[Goyal, WWW'11]	Guaranteed	low
	New Greedy	[Chen, KDD'09]	Guaranteed	low
	Static Greedy	[cheng, CIKM'13]	Guaranteed	low
Heuristics	Degree / Closeness	[Kempe, KDD'03]	Unguaranteed	high
	PageRank	[Page, 1999]	Unguaranteed	high
	Degree Discount	[Chen, KDD'09]	Unguaranteed	high
	PMIA	[Chen, KDD'10]	Unguaranteed	high
	IRIE	[Jung, ICDM'12]	Unguaranteed	high

Short Summary

	Low Quality	High Quality
Slow		Greedy CELF Greedy New Greedy Static Greedy Pruned BFS Simulation-based <100M edges
Fast	Degree Closeness Betweenness Degree Discount PMIA IRIE 100M~1B edges Heuristic-based	Challenges

Challenges Accepted

- C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. "Maximizing social influence in nearly optimal time." In SODA, 946-957, 2014.
- Y. Tang, Y. Shi, and X. Xiao. "Influence maximization in near-linear time: A martingale approach." In SIGMOD, 1539-1554, 2015
- Y. Tang, X. Xiao, and Y. Shi. "Influence maximization: Near-optimal time complexity meets practical efficiency." In SIGMOD, 75-86, 2014.

Extension

- **Initiator discovery: Given the state of the diffusion, find the nodes most likely to have initiated the diffusion**
  H. Mannila, E. Terzi. *Finding Links and Initiators: A Graph-Reconstruction Problem*. SDM 2009
- **Diffusion trees: Identify the most likely tree of diffusion tree given the set of influenced nodes**
  M. Gomez Rodriguez, J. Leskovec, A. Krause. *Inferring networks of diffusion and influence*. KDD 2010
- **Influence probabilities: Estimate the true influence probabilities**
  M. Gomez-Rodriguez, D. Balduzzi, B. Scholkopf. *Uncovering the temporal dynamics of diffusion networks*. ICML 2011
- **Topic-aware Influence: Naximize the spread of influence with respect to users relevant to the query topic**
  S. Chen, J. Fan, G. Li, J. Feng, K.-L. Tan, J. Tang. *Online Topic-Aware Influence Maximization*. VLDB 2015
- **Influence Minimization: Find a set of nodes that can minimize the number of influenced nodes**
  J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance. *Cost-Effective Outbreak Detection*. KDD 2007

Extension

- **Link Manipulation: Find and create Links to maximize the influence spread**
  H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Falutsos, C Faloutsos. *Gelling, and Melting, Large Graphs by Edge Manipulation*. CIKM 2012
- **Time-decay model**: **The probability of an infected node to infect its neighbors decays over time**
  B. Liu, G. Cong, D. Xu, Y. Zeng. *Time constrained influence maximization in social networks.* ICDM 2012
- **Dynamic Maximization: Maximize the influence spread in a time-evolving network** N. Ohsaka, T. Akiba, Y. Yoshida, and K.-i. Kawarabayashi. *Dynamic influence analysis in evolving networks*. VLDB 2016.
- **Competing Influence: Maximize the spread while competing with other products that are being diffused**
  A. Borodin, Y. Filmus, and J. Oren. *Threshold models for competitive influence in social networks*. AAAI 2010
- **Streaming Influence Maximization: Given the social streams of users, find the real-time influential seeds that maximize the influence spread**
  Y. Wang, Q. Fan, Y. Li, K.-L. Tan. *Real-Time Influence Maximization on Dynamic Social Streams*. VLDB 2017