ANALYZING AND LOCATING INFLUENCE IN NETWORKS

M. Vazirgiannis
École Polytechnique
January 2017

Outline

- Information Diffusion
- Modeling Information Diffusion
- Identifying Influential Spreaders
  - Identification of Single Spreaders
  - Identification of Multiple Spreaders

Outline

- Information Diffusion
- Modeling Information Diffusion
- Identifying Influential Spreaders
  - Identification of Single Spreaders
  - Identification of Multiple Spreaders

Social Networks

Source: Wikipedia & outsider news

Information Diffusion

A vast domain that attracted research interest from many fields as physics, biology, etc.

Researchers developed various techniques and models to **capture** information diffusion in online social networks, **analyze** it, **extract knowledge** and **predict** it.

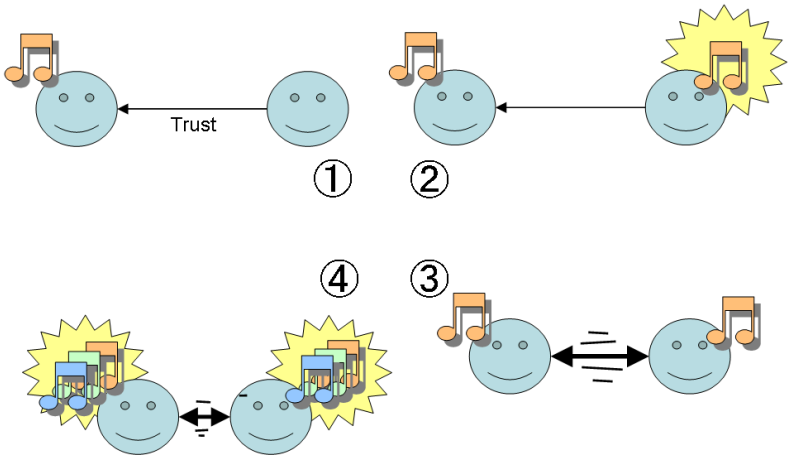
As computer scientists we focus on the case of information diffusion in online social networks that raises questions…

Information Diffusion

- Which pieces of information diffuse the most?
- How, why and through which paths information is diffusing and will be diffused in the future?
- Which members of the network play important roles in the spreading process?

Information Diffusion

Social Influence



[Matsuo et al. WWW'09]

Information Diffusion

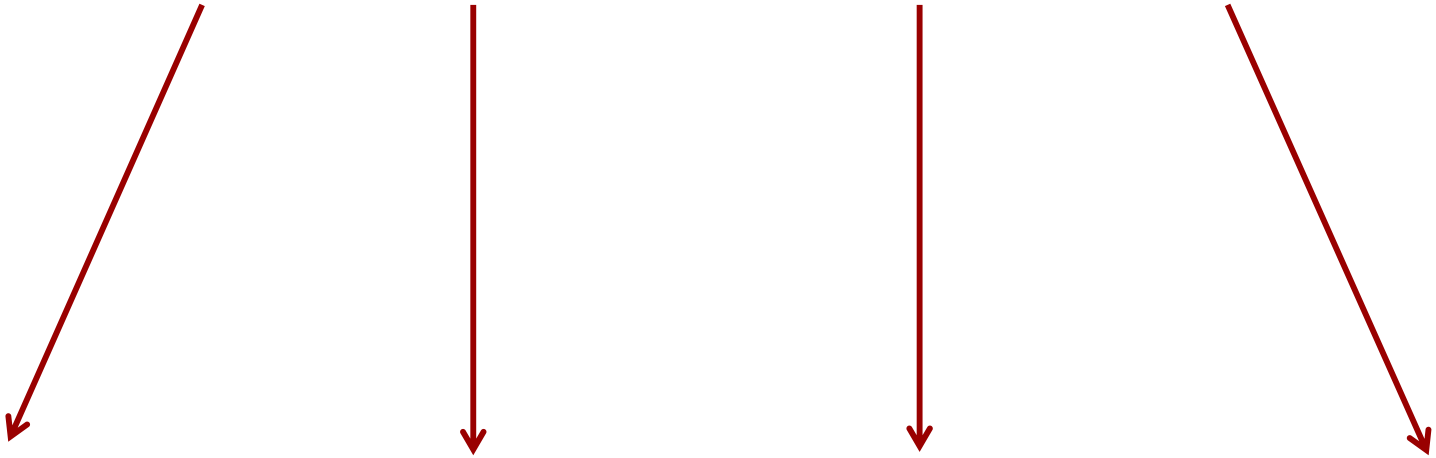
Involving a plethora of applications..

- Spread of technological information
- Word-of-mouth effects in marketing
- Spread of news and opinions
- Collection problem solving
- Virus propagation
- Expert finding
- “Friends” recommendation

Outline

- Information Diffusion
- Modeling Information Diffusion
- Identifying Influential Spreaders
  - Identification of Single Spreaders
  - Identification of Multiple Spreaders

Modeling Information Diffusion



→Threshold Models

Independent Cascade (IC) and Linear Threshold (LT)

→Epidemic Models

SIR/SIS models

→Rumor Dynamics

Maki and Thomson (MT) model

→Heat Diffusion Models

Hao et al. [2008]

Threshold Models

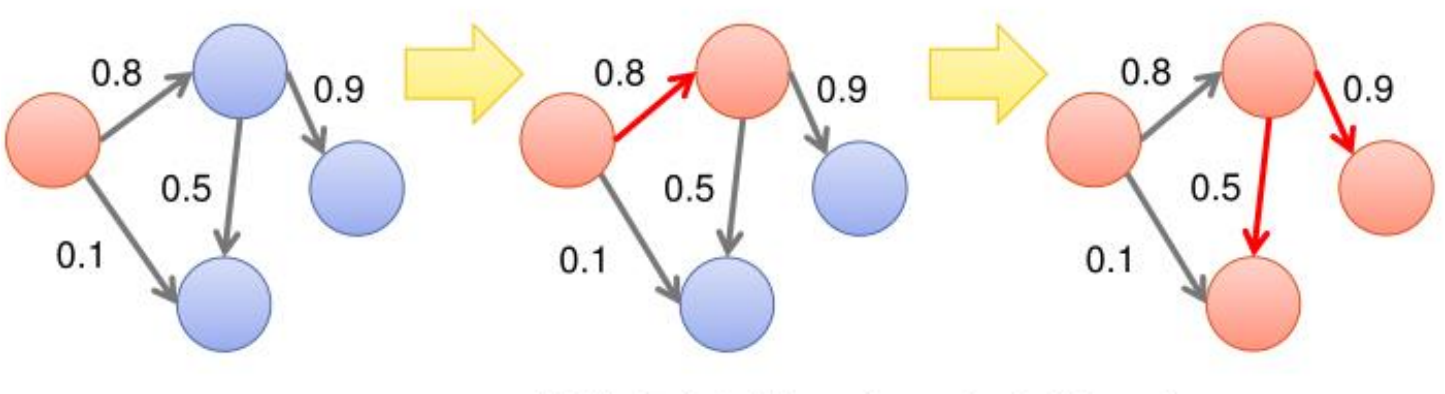
- Based on a directed graph where each node can be activated or not
  - Once activated, they do not deactivate (monotonicity assumption)
- The diffusion proceeds iteratively in a synchronous way along a discrete time axis starting from a set of initially activated nodes

David Kempe, Jon Kleinberg, and Éva Tardos. "Maximizing the spread of influence through a social network." ACM SIGKDD,2003.

Threshold Models

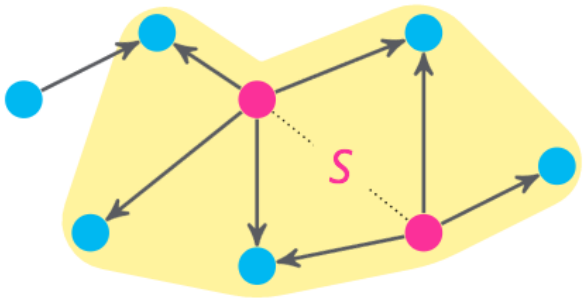
→Independent **C**ascade model

- Every edge $e=(u,v)$ has a propagation probability $p(u,v)$
- Initially some nodes are activated
- At each step t, all the nodes u activated at $t-1$ activate their neighbor v with a probability $p(u,v)$
- In live case: u is infected => v is infected.
- Once the nodes are activated, they stay activated

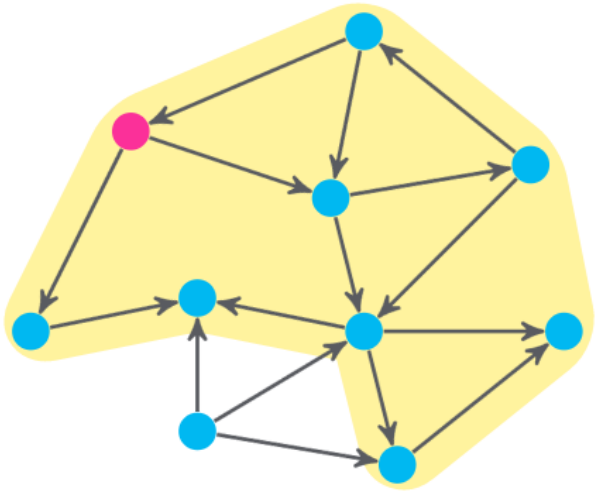


David Kempe, Jon Kleinberg, and Éva Tardos. "Maximizing the spread of influence through a social network." ACM SIGKDD,2003.

Definition of Influence
Given a set of nodes: expected number of reachable nodes from S



1. Influence Computation
Given a seed node set S:
What is the influence of S in G?



2. Influence Maximization
Given a number N: Compute sequence S of seed nodes of length N such that
influence for every prefix of S close to maximum for its size.

Threshold Models

→Linear Threshold model

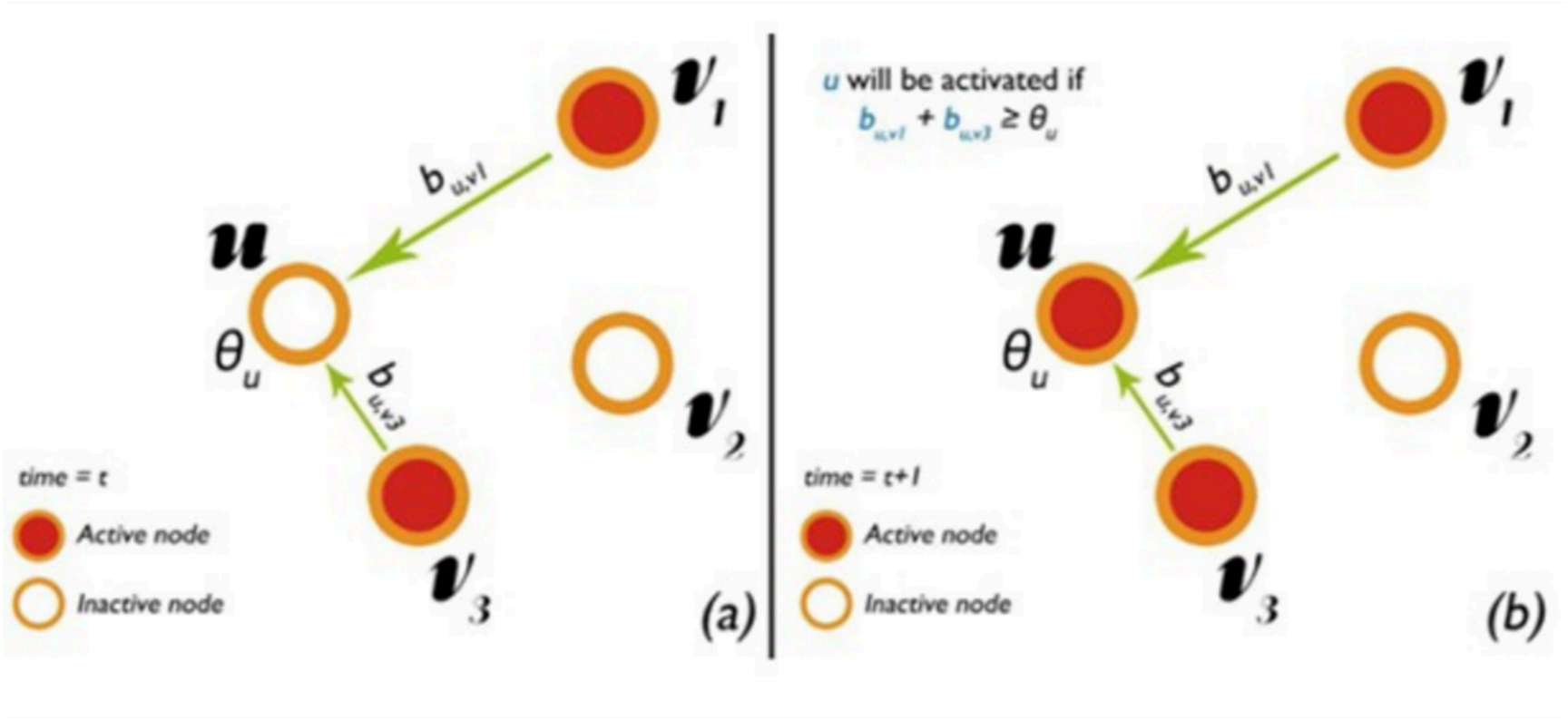
- Every edge $e=(u,v)$ has a weight $w(u,v)$
  - when $u, v \notin E$, $w(u,v) = 0$
  - $\Sigma_u w(u,v) \leq 1$

- Each node v selects a threshold $\theta_v \in [0,1]$ uniformly at random
- Initially some seed nodes are activated
- At each step, v checks if the weighted sum of its active neighbors is greater that the threshold θ_v and if so v is then activated
- Once the node is activated, it stays activated

David Kempe, Jon Kleinberg, and Éva Tardos. "Maximizing the spread of influence through a social network." ACM SIGKDD,2003.

Threshold Models

→Linear Threshold model



David Kempe, Jon Kleinberg, and Éva Tardos. "Maximizing the spread of influence through a social network." ACM SIGKDD,2003.

Epidemic Models

→SIR

S: Number of susceptibles, **I:** Number of Infected, **R:** Number of Recovered
β: infection **γ:** recovery rate

Daryl J. Daley, Joe Gani, and Joseph Mark Gani. Epidemic modelling: an introduction. Vol. 15. Cambridge University Press, 2001.

Epidemic Models

→SIR

$$\frac{dS}{dt} = -\beta SI$$

$$\frac{dI}{dt} = \beta SI - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

Let N be the population
where each individual has an equal probability of contacting the disease with rate of β.

→ βN contacts transmit the disease in the population per unit time
→ S/N is the fraction of contacts by one infected individual with a susceptible
→ I number of infectives

Therefore the consumption rate of the susceptibles is (βN) * (S/N) * (I) =βSI and the rate of the infected and recovered nodes are formulated similarly.

Daryl J. Daley, Joe Gani, and Joseph Mark Gani. Epidemic modelling: an introduction. Vol. 15. Cambridge University Press, 2001.

Epidemic Models

→SIS

$$\frac{dS}{dt} = -\beta SI + \gamma I$$

$$\frac{dI}{dt} = \beta SI - \gamma I$$

Daryl J. Daley, Joe Gani, and Joseph Mark Gani. Epidemic modelling: an introduction. Vol. 15. Cambridge University Press, 2001.

Rumor Dynamics

- Similar to epidemic spreading, rumor diffusion is simulated considering that nodes are *spreaders*, *ignorants* and *stiflers*.

- *Stiflers* know the rumor but are not interested in spreading the information anymore.

- Main difference with epidemic models:
  - A spreader turns into a *stifler* by a process that involves contacts.

Daniel P. Maki and Maynard Thompson. Mathematical models and applications: with emphasis on the social life, and management sciences. No. 511.8 M3. 1973.

Rumor Dynamics

- The fraction of ignorants $(\psi(t))$, spreaders $(\phi(t))$ and stiflers $(s(t))$ at time t are defined such that

$$(\psi(t))+ (\phi(t)) +(s(t)) =1.$$

- Process starts with one spreader and $N\text{-}1$ ignorants (where N the total number of nodes in the network).
- Spreaders spread the rumor to their ignorant neighbors with a rate λ.
- If a spreader contacts a spreader or a stifler, they become a stifler at a rate δ.

Daniel P. Maki and Maynard Thompson. Mathematical models and applications: with emphasis on the social life, and management sciences. No. 511.8 M3. 1973.

Heat Diffusion Model

- Innovators and early adopters act as *heat sources* and have a high amount of heat.
- Assume an undirected social network graph $G=(V,E)$
- $f_i(t)$: heat at node v_i at time t, from an initial distribution of heat given by $f_0(t)$ at time zero.
- At time t node v_i receives an amount $M(i,j,t,\Delta t)$ heat from its neighbor v_j which is proportional to the time period Δt and the heat difference $f_j(t)-f_i(t)$.

$$M(i,j,t,\Delta t) = \alpha*(f_j(t)-f_i(t))*\Delta t$$

α: thermal conductivity or heat diffusion coefficient.

HaoMa et al. "Mining social networks using heat diffusion processes for marketing candidates selection." CIKM, 2008.

Heat Diffusion Model

- As a result the heat difference at node i between time t+Δt and time t will be equal to the sum of the heat that it receives from all its neighbors.

$$\frac{f_i(t+\Delta T)-f_i(t)}{\Delta t}=\alpha\sum_{j:(v_j,v_i)\in E} f_j(t)-f_i(t)$$

or in matrix form: $$\frac{\boldsymbol{f}(t+\Delta t)-\boldsymbol{f}(t)}{\Delta t}=\alpha\boldsymbol{H}\boldsymbol{f}(t)$$

where $$H_{ij}=\begin{cases}1, & (v_i,v_j)\in E \text{ or } (v_i,v_j)\in E,\\ -d(v_i), & i=j,\\ 0, & otherwise.\end{cases}$$

HaoMa et al. "Mining social networks using heat diffusion processes for marketing candidates selection." CIKM, 2008.

Heat Diffusion Model

- In the limit $\Delta t \to 0$:

$$\frac{d}{dt} f(t) = aH f(t)$$

Solving the differential equation we get:

$$f(t) = e^{atH} f(0)$$

- matrix e^{atH} is called the diffusion kernel as the heat diffusion process continues infinitely many times from the initial heat diffusion.

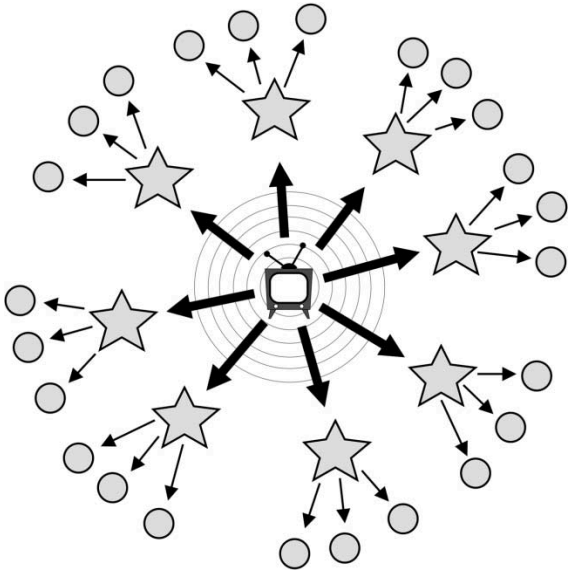
HaoMa et al. "Mining social networks using heat diffusion processes for marketing candidates selection." CIKM, 2008.

Outline

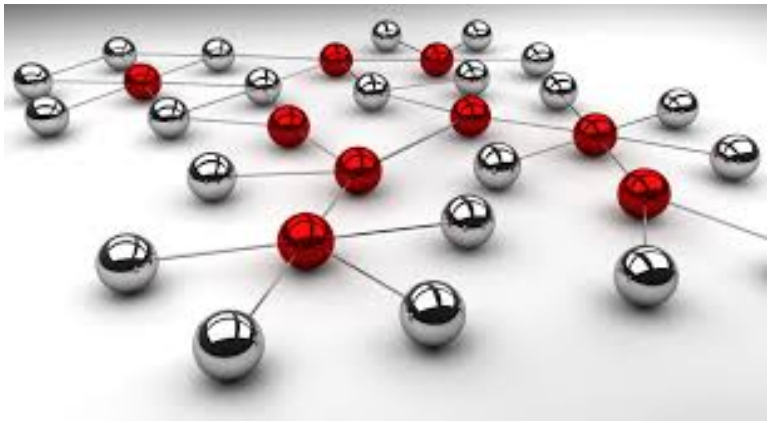
- Information Diffusion
- Modeling Information Diffusion
- Identifying Influential Spreaders
  - Identification of Single Spreaders
  - Identification of Multiple Spreaders

Identifying Influential Spreaders

→ Locate “opinion/tribe leaders” or “influentials” that act as intermediates between mass media and the majority of the society



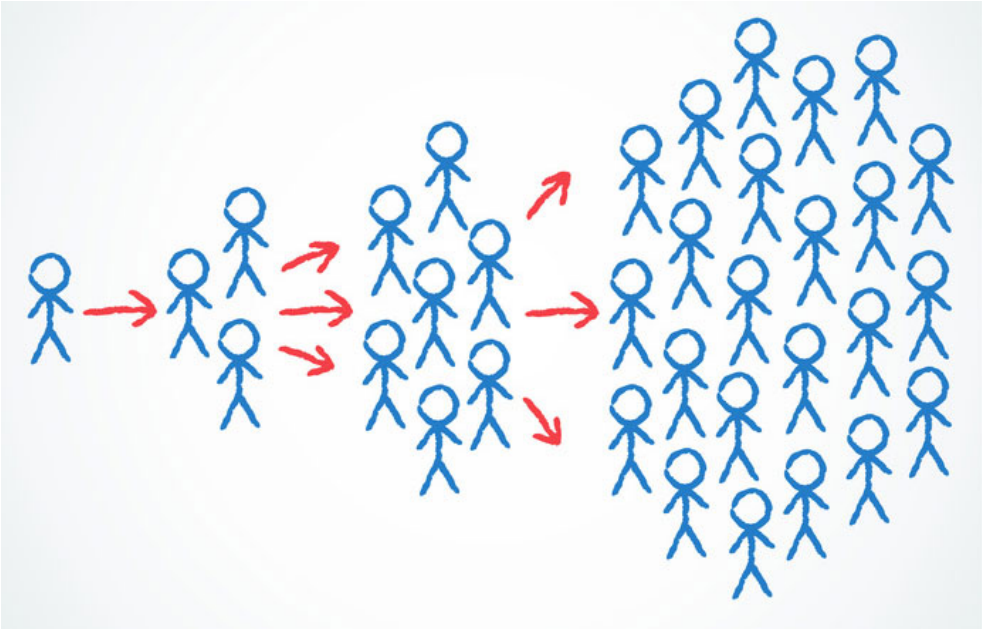
[Watts and Dodds ‘07]



Identification of Single Spreaders

→ Identification of individual influential nodes

→ Several studies have been working towards locating those entities and evaluating their hypotheses by simulating the spreading process using epidemic models SIR/SIS on real or artificial networks.

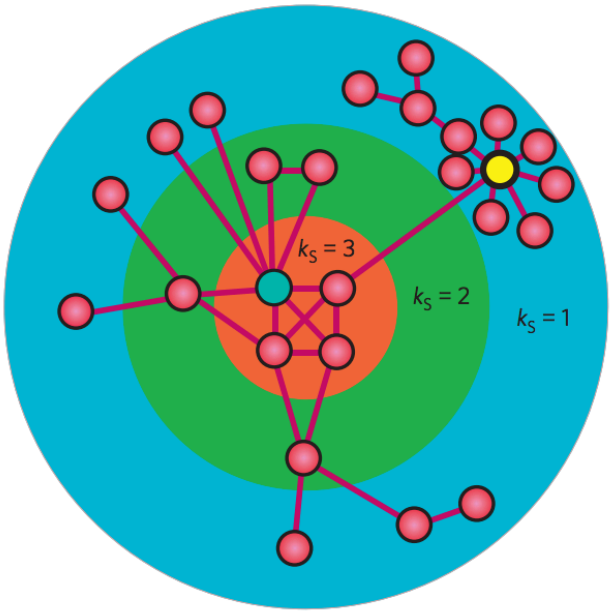


Identification of Single Spreaders

- straightforward approach: consider node centrality criteria.

→But!! There exist cases where a node can have high degree when its neighbors are not well connected..

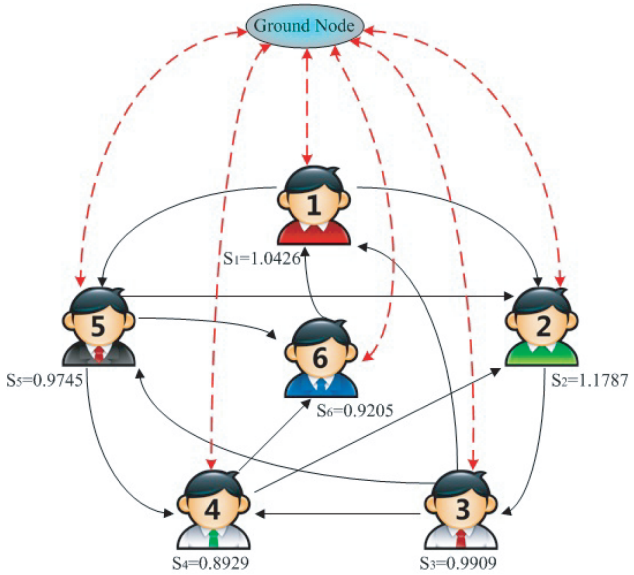
This can occur when a high degree node is located to the periphery of the network.



Identification of Single Spreaders

Leader Rank algorithm [Lu et al. PlosOne '11]

- Random walk based algorithm that identifies influential users in networks.
- assign to each node, except for the ground node, one unit of resource evenly distributed to the node's neighbors through the directed links.
- The process continues until steady state is attained.
- equivalent to random walk on the directed network, probability flow corresponds to the vote from fan to leader.
- Convergence state at time t_c →



$$s_i(t+1) = \sum_{j=1}^{N+1} \frac{a_{ji}}{k_j^{out}} s_j(t).$$

$$S_i = s_i(t_c) + \frac{s_g(t_c)}{N}$$

- Similar to the PageRank[1] parameter free algorithm

[1] : [Page et al. Technical Report Stanford InfoLab '99] , [Brin et al. Comput Networks and ISDN Systems '98]

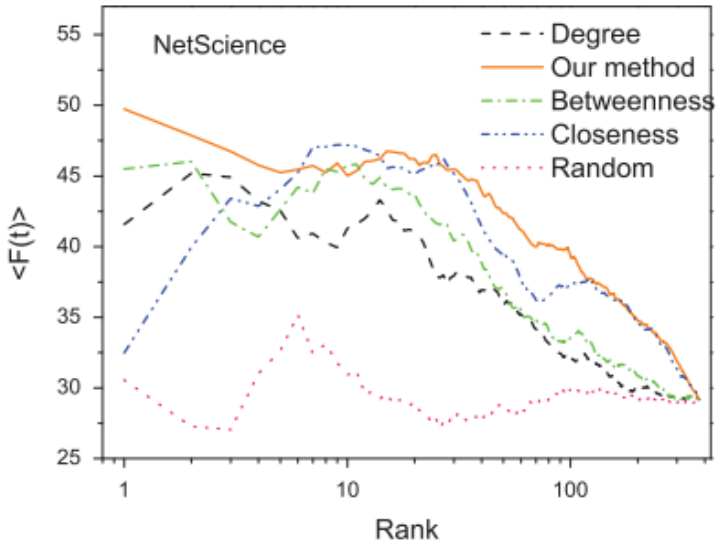
Identification of Single Spreaders

→ Chen et al. proposed a semi-local centrality measure C_L that serves as a tradeoff between degree other computationally complex measures (e.g., betweeness and closeness centrality):

$$Q(u) = \Sigma_{w \in \Gamma(u)} N(w),$$

$$C_L(v) = \Sigma_{u \in \Gamma(v)} Q(u)$$

$\Gamma(u)$: set of hop-1 neighbours of u
$N(w)$ is the number hope-1 and hope-2 neighbors of node w.



→ Their method seems to work well compared with the use of other famous node centralities.

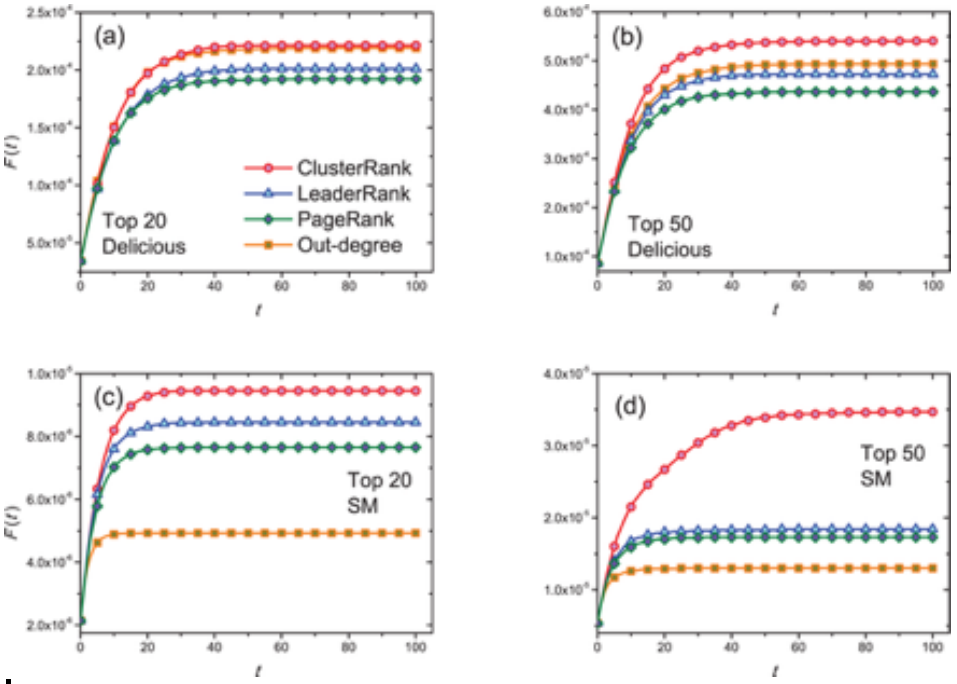
[Chen et al. Physica A '11]

Identification of Single Spreaders

ClusterRank [Chen et al. PlosOne '13]

Local ranking algorithm based on:
- number of neighbor or neighbor influences
- clustering coefficient.

$$C_i = \frac{|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}$$

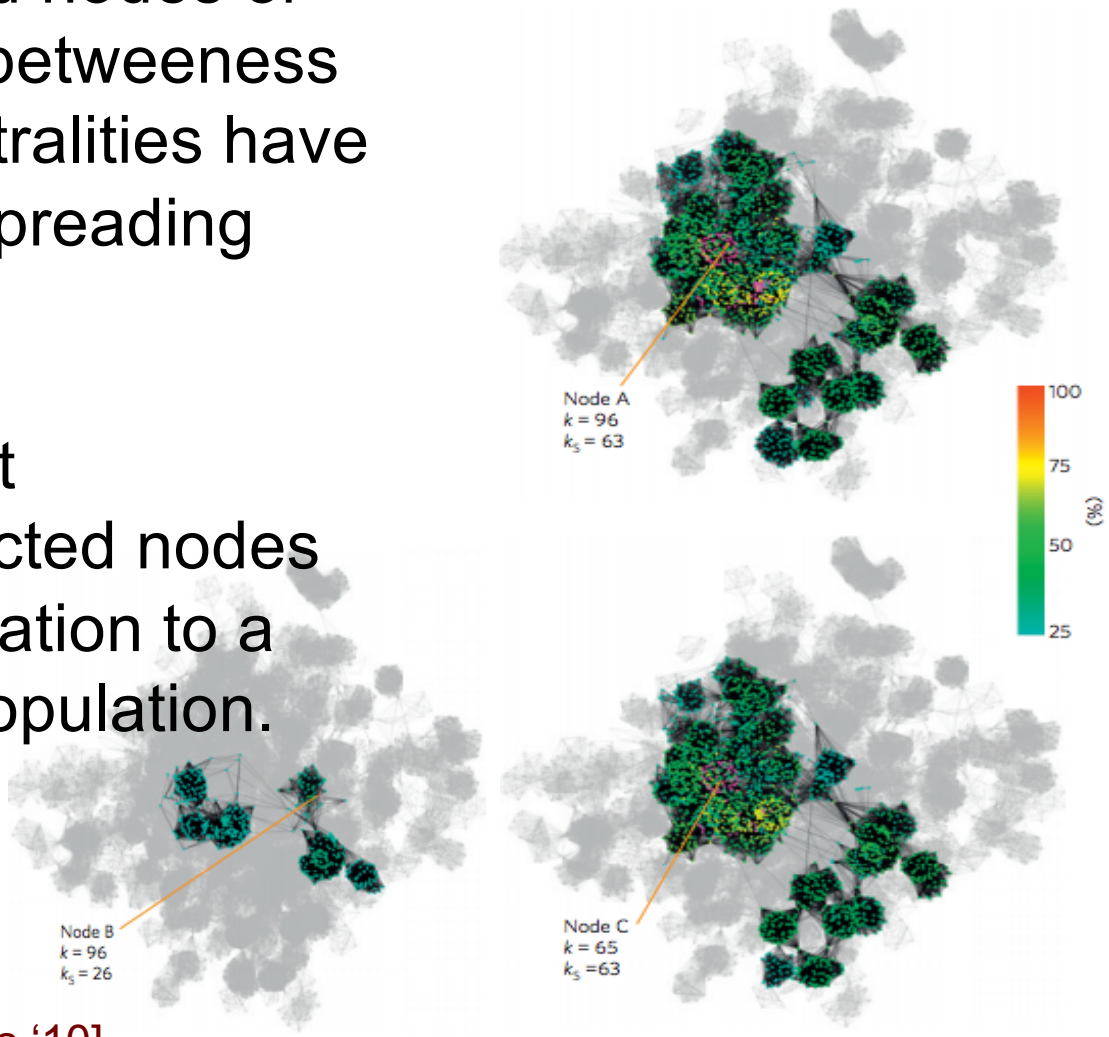


Outperforms PageRank and LeaderRank and as it uses only local information it is more efficient than global methods.

Identification of Single Spreaders

- → highly connected nodes or those having high betweeness and closeness centralities have little effect on the spreading process.

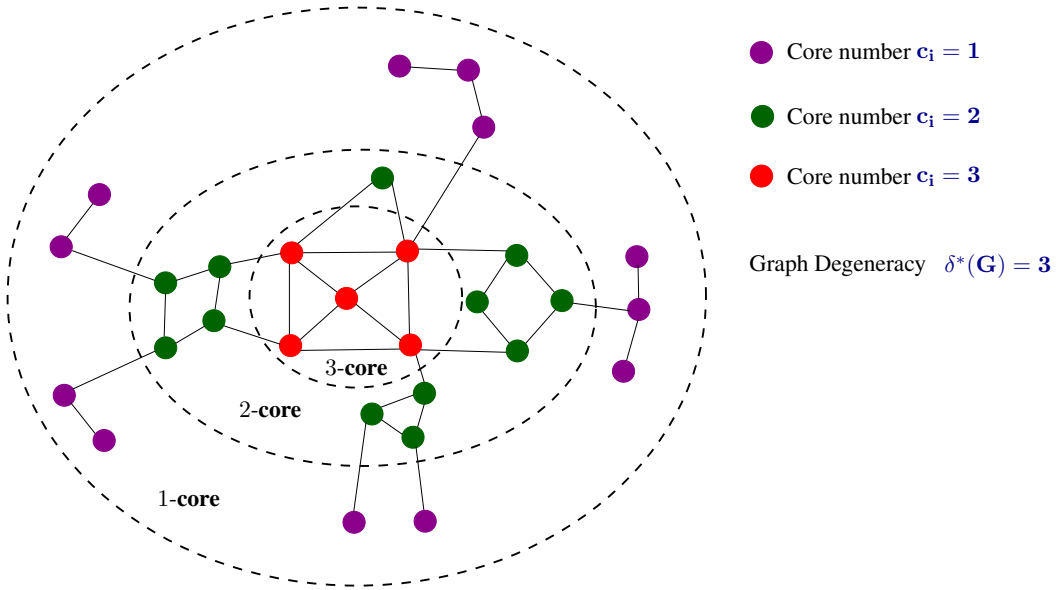
- Less connected but strategically connected nodes disseminate information to a larger part of the population.



[Kitsak et al. Nature Physics '10]

Identification of Single Spreaders

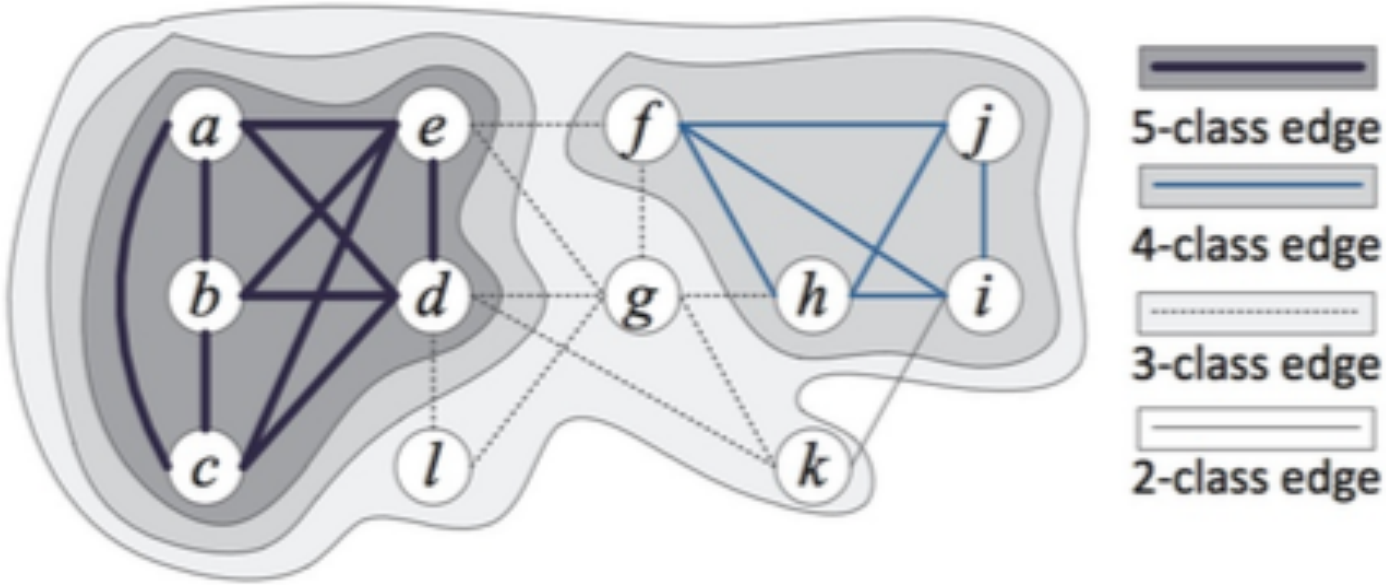
- the k-core decomposition algorithm applied
- The k-core algorithm removes nodes that do not satisfy a particular degree-based threshold
- Approximates the densest subgraph



[Batagelj et al. CoRR '03]

K-Truss Decomposition

T_K, $K \geq 2$: the *K*-truss subgraph of *G* , the largest subgraph where all edges belong to $K - 2$ **triangles.**

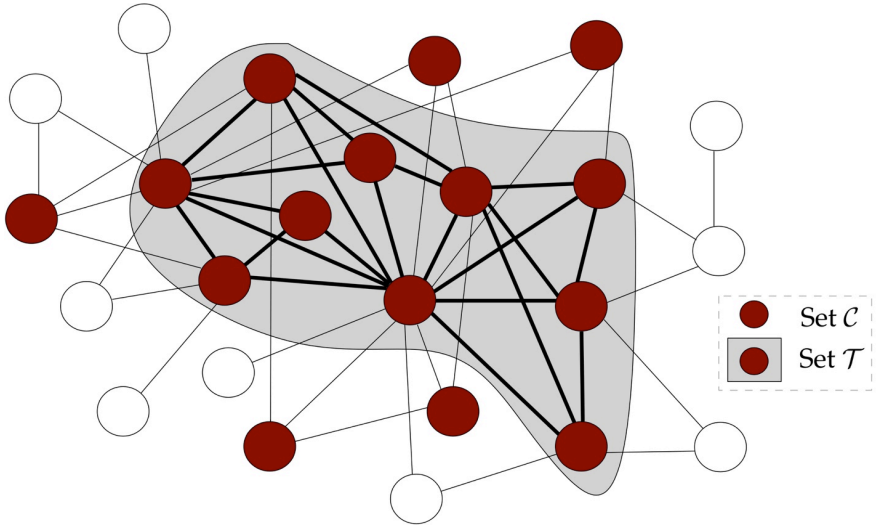


Identification of Single Spreaders

It was recently proved that the nodes belonging to the best K-truss subgraph, as identified by the K-truss decomposition of the network, perform even better leading to faster and wider epidemic spreading.

The K-truss decomposition extends the notion of k-core using triangles, i.e., cycle subgraphs of length 3.

The maximal **k-core** and **K-truss** subgraphs (i.e., maximum values for k;K) overlap, with the latter being a subgraph of the former; in fact, K-truss represents the core of a k-core that filters out less important information.



[Rossi, Malliaros & Vazirgiannis WWW '15]

[Malliaros, Rossi & Vazirgiannis Scientific Reports '16]

Experiments - Datasets

Network Name	Nodes	Edges	k_{max}	K_{max}	$\|\mathcal{C}\|$	$\|\mathcal{T}\|$
EMAIL-ENRON	33,696	180,811	43	22	275	45
EPINIONS	75,877	405,739	67	33	486	61
WIKI-VOTE	7,066	100,736	53	23	336	50
EMAIL-EUALL	224,832	340,795	37	20	292	62
SLASHDOT	82,168	582,533	55	36	134	96
WIKI-TALK	2,388,953	4,656,682	131	53	700	237

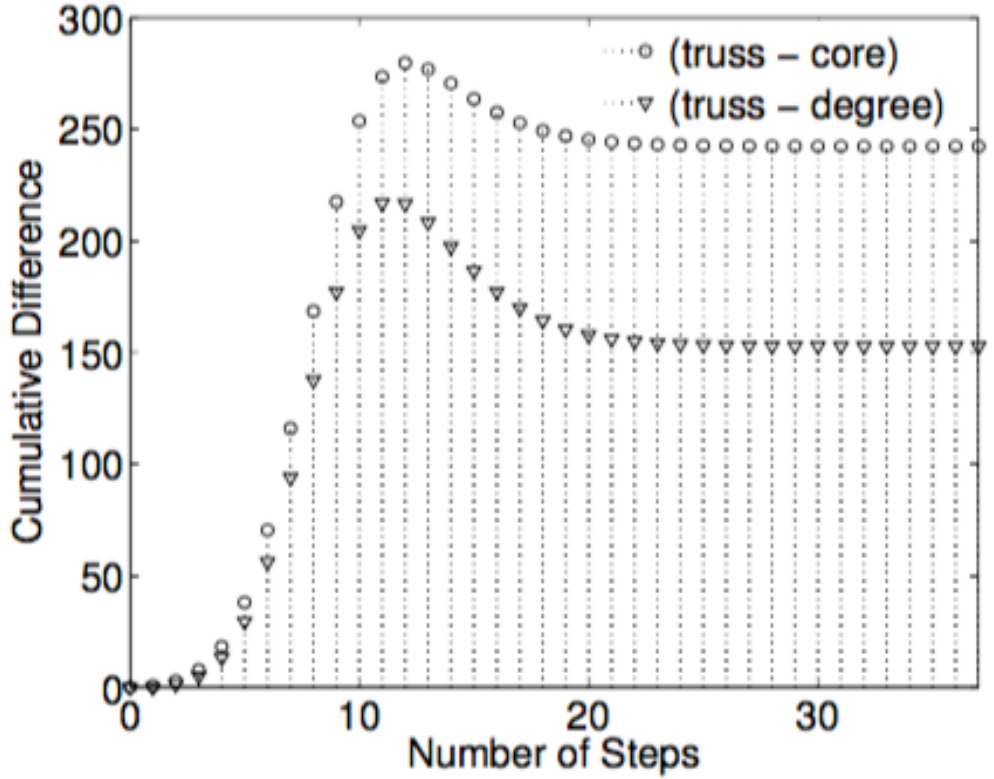
Jure, L. and Andrej, K. Stanford Network Analysis Project. http://snap.stanford.edu

Stepwise evolution of spreading performance/node

		Time Step							
	Method	2	4	6	8	10	*Final step*	σ	*Max step*
EMAIL-ENRON	**truss**	8.44	46.66	204.08	418.77	355.84	2,596.52	136.7	33
	core	4.78	31.97	152.55	367.28	364.13	2,465.60	199.6	37
	top degree	6.89	34.13	155.48	360.89	357.08	2,471.67	354.8	36
EPINIONS	**truss**	4.17	19.70	75.04	204.14	329.08	2,567.69	227.8	37
	core	3.45	14.72	55.27	158.56	280.03	2,325.37	327.2	43
	top degree	4.22	16.03	58.84	166.23	289.49	2,414.99	331.7	47
WIKI-VOTE	**truss**	2.92	6.92	15.27	28.73	42.46	560.66	114.9	52
	core	1.92	4.78	10.65	20.66	32.40	466.01	104.5	57
	top degree	2.43	5.46	12.05	23.05	35.55	502.88	104.5	62
EMAIL-EUALL	**truss**	11.62	62.25	240.97	584.87	725.42	5,018.52	487.94	36
	core	9.85	40.82	158.72	433.81	644.76	4,579.84	498.71	38
	top degree	17.96	39.93	144.69	503.18	548.25	4,137.56	1,174.84	39
SLASHDOT	**truss**	5.36	66.21	461.35	1,390.52	1,359.99	8,207.46	368.37	32
	core	6.48	61.13	410.19	1,272.29	1,344.33	8,002.76	518.43	32
	top degree	13.95	83.29	483.95	1,426.81	1,403.80	8,489.45	59.01	32
WIKI-TALK	**truss**	64.21	3,259.05	34,543.23	9,853.84	1,186.41	93,491.81	476.22	21
	core	41.77	2,027.69	31,223.21	13,055.45	1,664.52	93,496.50	767.35	23
	top degree	88.84	2,475.01	29,694.45	13,720.15	1,937.89	93,411.18	1,166.77	24

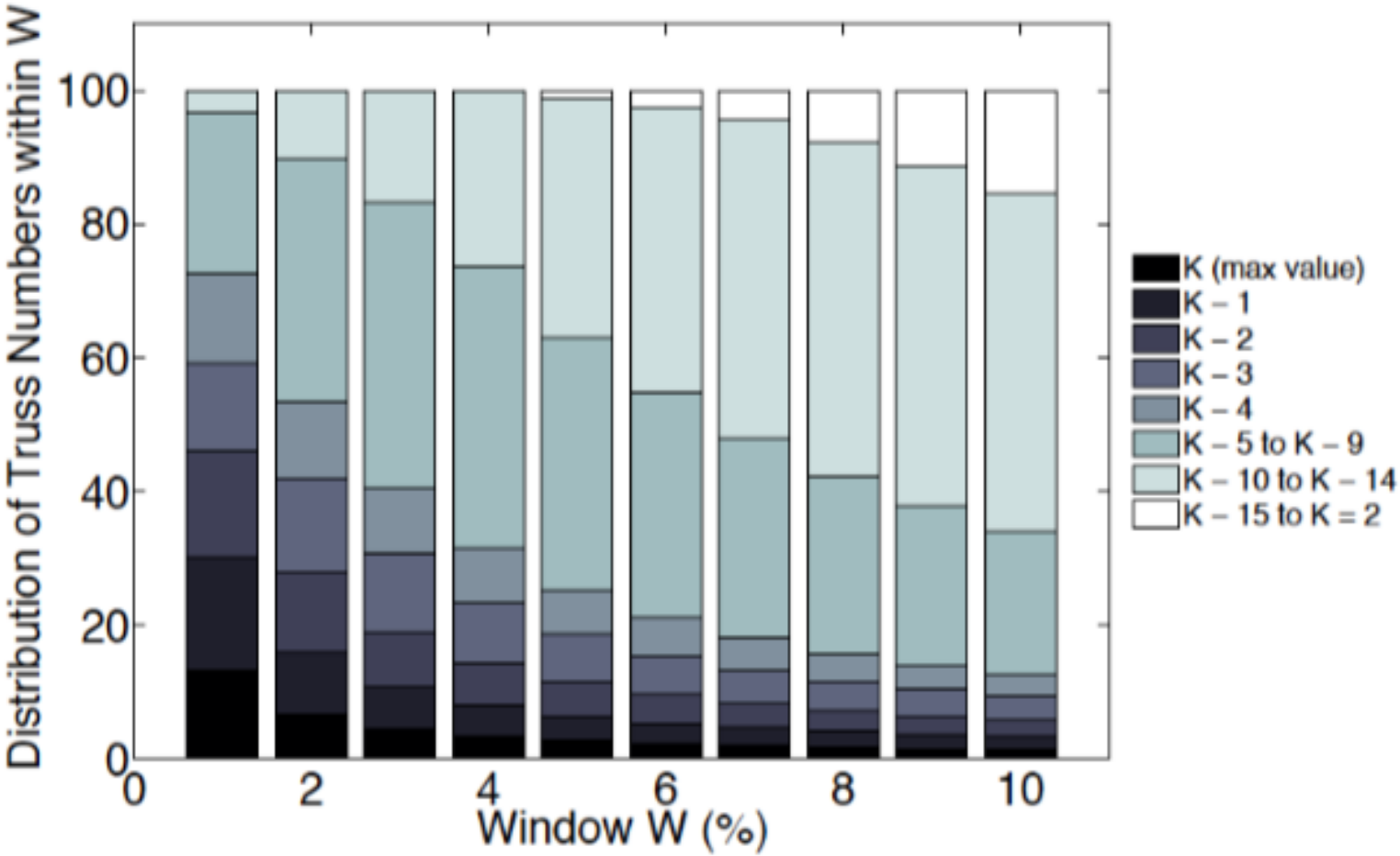
Evaluation of the spreading performance per step of the process.

Stepwise evolution of spreading performance/node



(a) EPINIONS: $\beta = 0.007$

EPINIONS	**truss**	4.17	19.70	75.04	204.14	329.08	2,567.69	227.8	37
	core	3.45	14.72	55.27	158.56	280.03	2,325.37	327.2	43
	top degree	4.22	16.03	58.84	166.23	289.49	2,414.99	331.7	47



Distribution of node's truss number with respect to the ranking of the nodes under their spreading effectiveness. We report results for the EMAIL-ENRON dataset.

Outline

- Information Diffusion
- Modeling Information Diffusion
- Identifying Influential Spreaders
  - Identification of Single Spreaders
  - Identification of Multiple Spreaders

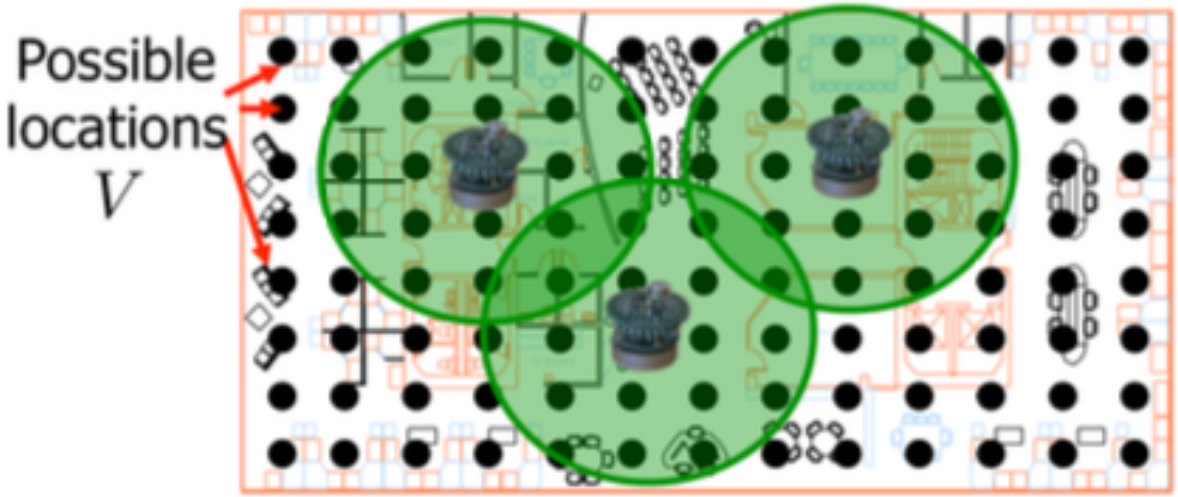
Identification of Multiple Spreaders

- Also called Influence Maximization (IM)

- Solutions to the problem:
  - Using the LT and IC diffusion models – Part I
  - Using the Heat Diffusion Process model – Part II

Influence Maximization – a motivating example

Problem : place **sensors** to monitor temperature



Set function : $f : 2^V \to \mathbb{R}^+$

Idea : $\forall A \subseteq V, f(A) :=$ "Area covered by sensors placed at A"

Influence Maximization – a motivating example

Budget constraint : we can place at most k sensors

Question : how to **maximize the covered area** ?

$$A^* \in \underset{A \subseteq V,\ |A| \leq k}{\arg\max}\ f(A)$$

Property : **NP-hard** in general...

Influence Maximization – ex. 2: viral marketing

Problem : **influence maximization** in social networks

We give **free items** to k customers (**seed nodes**)

We want to **maximize the spread of influence** through the social network

$\longrightarrow$ *Several propagation models : Independent Cascade Model, Linear Threshold Model...* [see Kempe et al. 2003]

Influence Maximization – ex. 2: viral marketing

Directed graph $\mathcal{G} = (V, E)$

Set function : $f : 2^V \to \mathbb{R}^+$

$\forall A \subseteq V, f(A) :=$ "expected size of the set of **influenced nodes**, when $A =$ seed nodes"

Problem : $\boxed{A^* \in \underset{A \subseteq V,\ |A| \leq k}{\arg\max}\ f(A)}$

Property : as before, **NP-hard** in general

Similarities between Ex. 1 and Ex. 2

Some similarities between Example 1 and Example 2 ?

As explained before :

- constrained maximization problem : $A^* \in \underset{A \subseteq V,\ |A| \leq k}{\arg\max}\ f(A)$
- NP-hard in general

But also :

- Set functions f are **monotone**...
- ...and are **submodular**

[Kempe et al., SIGKDD 2003 ; Krause, ICML 2013]

Monotonicity and Submodularity

Definition (Monotonicity)

The set function f is **monotone** if adding an element to a set cannot cause f to decrease :

$$\forall v \in V,\ \forall A \subseteq V,\ f(A \cup \{v\}) \geq f(A)$$

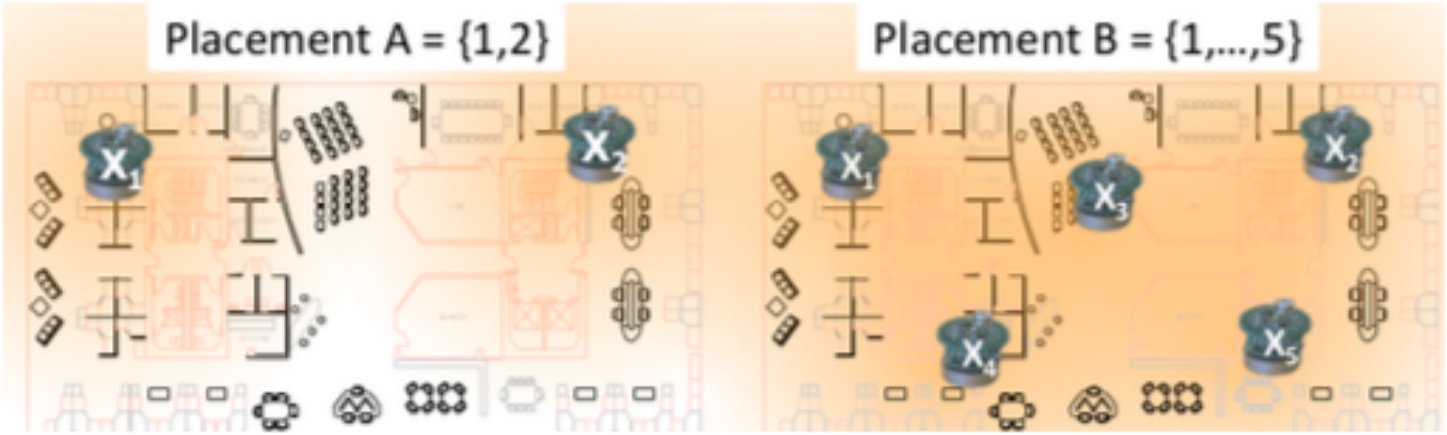
Definition (Submodularity)

The set function f is **submodular** if, for any $A \subseteq B \subseteq V$ and for any $v \in V \setminus B$, we have :

$$f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$$

Submodularity in the sensor placement problem

Illustration of submodularity, from [Krause, ICML 2013]



$\longrightarrow$ Adding sensors can only help : f is **monotone**

$\longrightarrow$ The marginal coverage gain is larger when less sensors are in place ($A \subset B$) : f is **submodular**

Submodularity

Theorem (Nemhauser et al. 1978)

Let $f : 2^V \to \mathbb{R}^+$ be a non-negative, monotone and submodular set function.

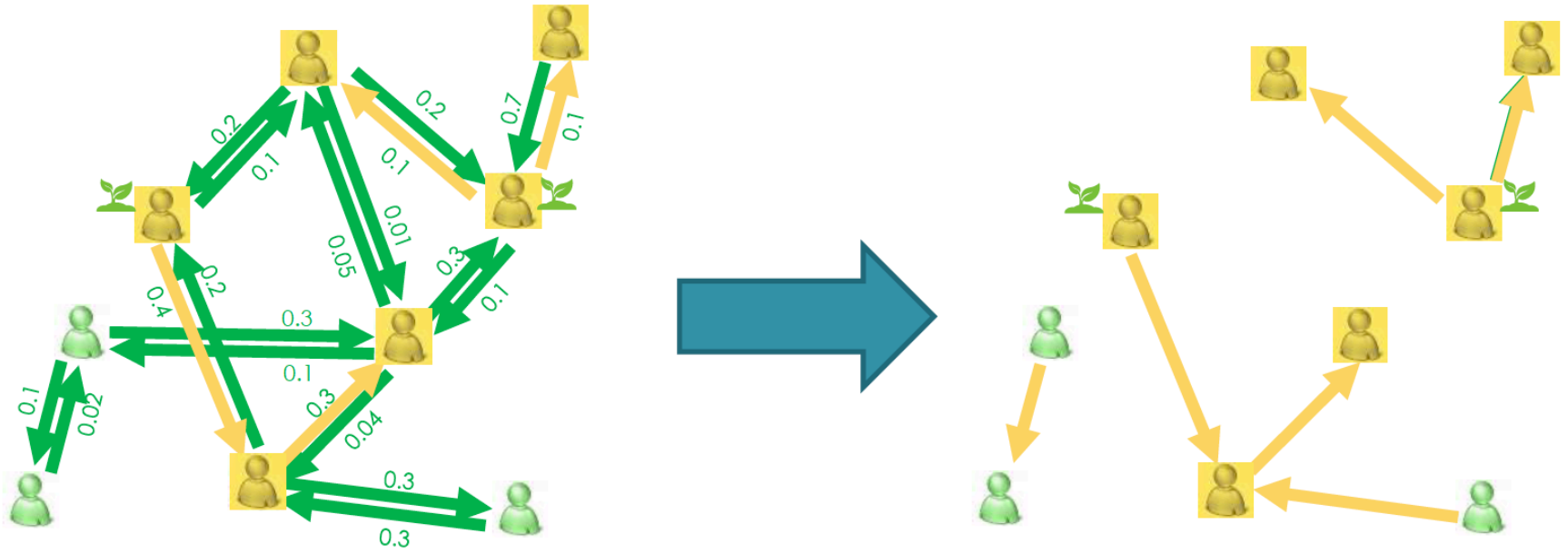
Let A^{Greedy} (with $|A^{Greedy}| \leq k$) be the set obtained via a ***greedy algorithm****, and let the set A^* be an optimal solution of the problem : $A^* \in \underset{A \subseteq V,\ |A| \leq k}{\arg\max}\ f(A)$.*

We have :

$$f(A^{Greedy}) \geq \underbrace{(1 - \frac{1}{e})}_{\simeq 63\%} f(A^*)$$

Influence Maximization

Optimizing submodular functions



diffusion dynamic

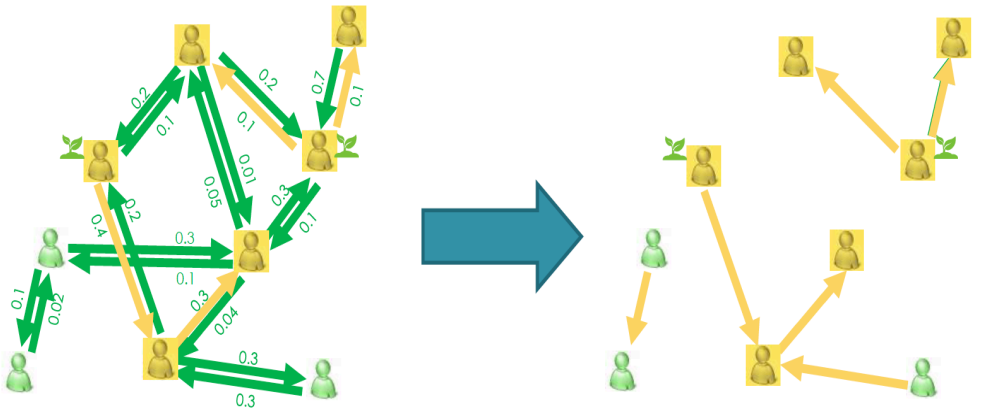
random live-edge graph

P (set I is activated given seed set S) = P (set I is reachable from S in random live-edge graph)

Influence Maximization

Optimizing submodular functions

- yellow nodes belong to the active set of nodes after the diffusion process
- For both the **I**ndependent **C**ascade and the **L**inear **T**hreshold model.
- Random live-edge graph -IC:
  - Each edge is independently selected as live with *its propagation probability*
- Random live-edge - LT:
  - Each node selects at most one incoming edge with probability proportional to its weight



Influence Maximization – Part I

Greedy Algorithm

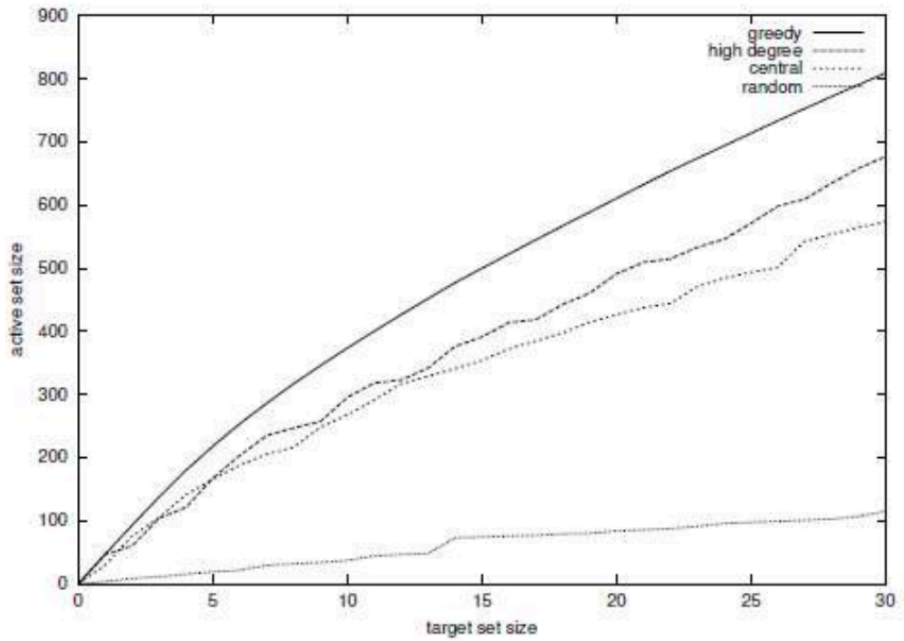
Data: Graph $\mathcal{G} = (V, E)$, budget k, function f
Result: Greedy set A^{Greedy}
begin
  $A = \emptyset$
  while $|A| < k$ **do**
    $A = A \cup \arg\max_{v \in V \setminus A} \Delta(v|A) := f(A \cup \{v\}) - f(A)$
    (Exact computations or Monte Carlo simulations)
  end
  Return A
end

$\longrightarrow$ Possible improvements : *lazy* greedy algorithm, CELF, LDAG...
[Leskovec et al., KDD 2007; Kim et al., ICDE 2013...]

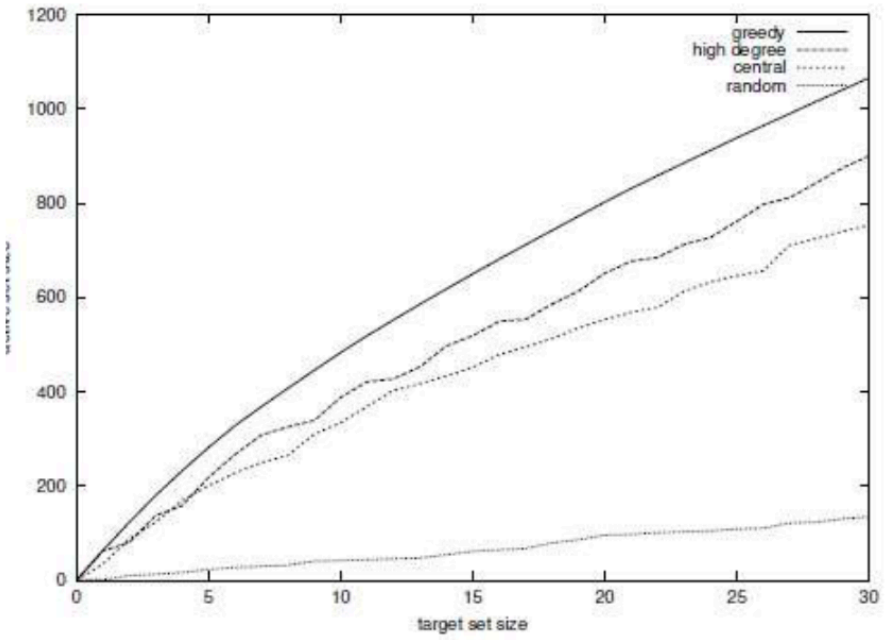
Influence Maximization – Part I

Performance of Greedy Algorithm

weighted cascade model



linear threshold model



[Kempe, Kleinberg and Tardos, KDD 2013]

Influence Maximization – Part I

Scalable Influence Maximization

The greedy algorithm is computationally expensive. Given a graph of n nodes and k seed nodes needed, each round needs evaluation of n influence spreads, meaning O(nk) evaluations….

Actions needed:

- Reduced number of influence spread evaluations
- Batch computation of influence spread
- Scalable heuristics for influence spread computation

Extensions of the Greedy algorithm - DaScIM

- MATM(Matrix Method) under LT and IC, B. Shi, Dr. Nikos Tziortziotis, M. Rossi, F. Malliaros

- Adaptive Influence Maximization on Networks - An application of the Adaptive Submodularity property G. Salha, Dr. Nikos Tziortziotis

- MATM - MATM(Matrix Method) under LT and IC

Issues with the greedy algorithm: time-consuming.
More than 1 week for a graph with one million edges.

Reasons:

- Monte-Carlo simulation for influence estimation
- Repetitive computation in influence computation for a set of nodes

We propose MATM(Matrix Method) under LT and IC model,

- 1000 faster than greedy algorithm.
- Replacing Monte-Carlo simulation with simple path enumeration.

MATM – for LT

- For each each graph node x
  - generate (with DFS) a tree $T(x)$ m simple paths
  - For the $i\text{-}th$ path from node x: $pr(i) = [pr_{x1},...,pr_{ij}]$

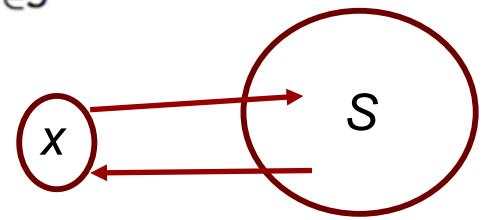
- single nore its influence is: $\sigma(x) = \sum_{y \in T(x)} IF(x,y)$

where $IF(x,y) = \sum_{i=1}^{m} pr_i(ind(y))$

- Node set influence:

$$\sigma(S+x) = \sigma(x) + \sigma(S) - \sum_{y \in S} SF(x,y) - \sum_{y \in S} SF(y,x)$$

where $SF(x,y) = \sum_{i=1}^{m} \sum_{j=ind(x)}^{l(p_i)} pr_i(j)$



MATM – for IC

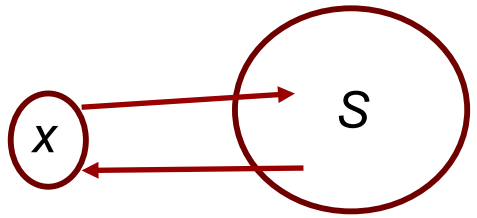
- For each each graph node x
  - generate (with DFS) a tree $T(x)$ m simple paths
  - For the $i\text{-}th$ path from node x: $pr(i) = [pr_{x1},...,pr_{ij}]$

- single nore its influence is: $\sigma(x) = \sum_{y \in T(x)} IF(x,y)$

where $IF(x,y) = 1 - \prod_{i=1}^{m} (1 - pr_i(y)\mathbb{1}_{y \in Pr_i})$

- Node set influence heuristics:

  - S1. $\forall p \in P(S)$, keep only the subpath $p[:x]$ if $x \in p$, $\forall p \in P(v)$, keep only the subpath $p[:y]$ before y if $y \in p$ $(y \in S)$
  - S2. Sum the probability along the revised paths



MATM – Experiments

1. Datasets:

Dataset	NETHEPT	Epinions	DBLP	Email-EuAll
#Nodes	15K	75K	654K	224K
#Edges	62K	405K	2M	340K

2. Test environment: Python, on a Linux machine with a 3.00GHz Intel Xeon CPU and 64 GB memory
3. Algorithm compared: Greedy algorithm, High-Degree, LDAG[2]
4. Performance analysis: quality of seeds, algorithm efficiency(running time, memory consumption)

MATM – Experiments

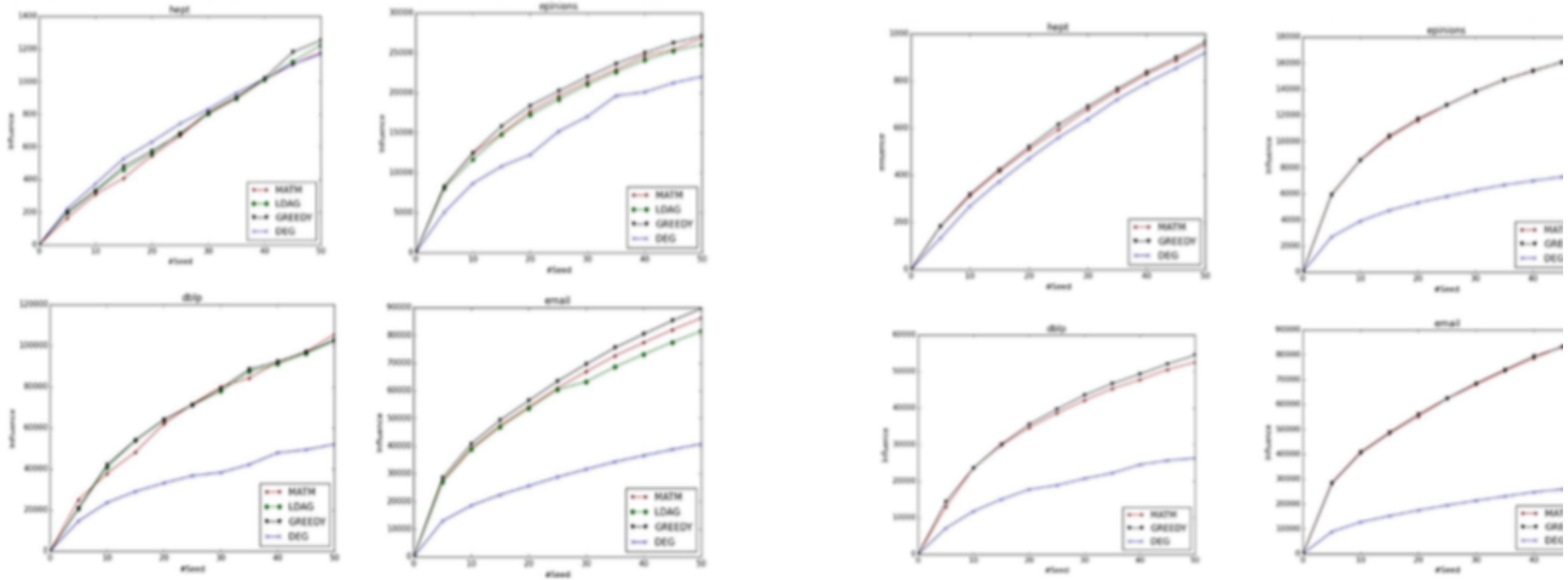


Figure: Influence Spread on various datasets(Left: LT model, Right IC model)

Quality of seeds selected by MATM is very close to greedy algorithm(ground-truth) under both IC and LT model

MATM – Execution time

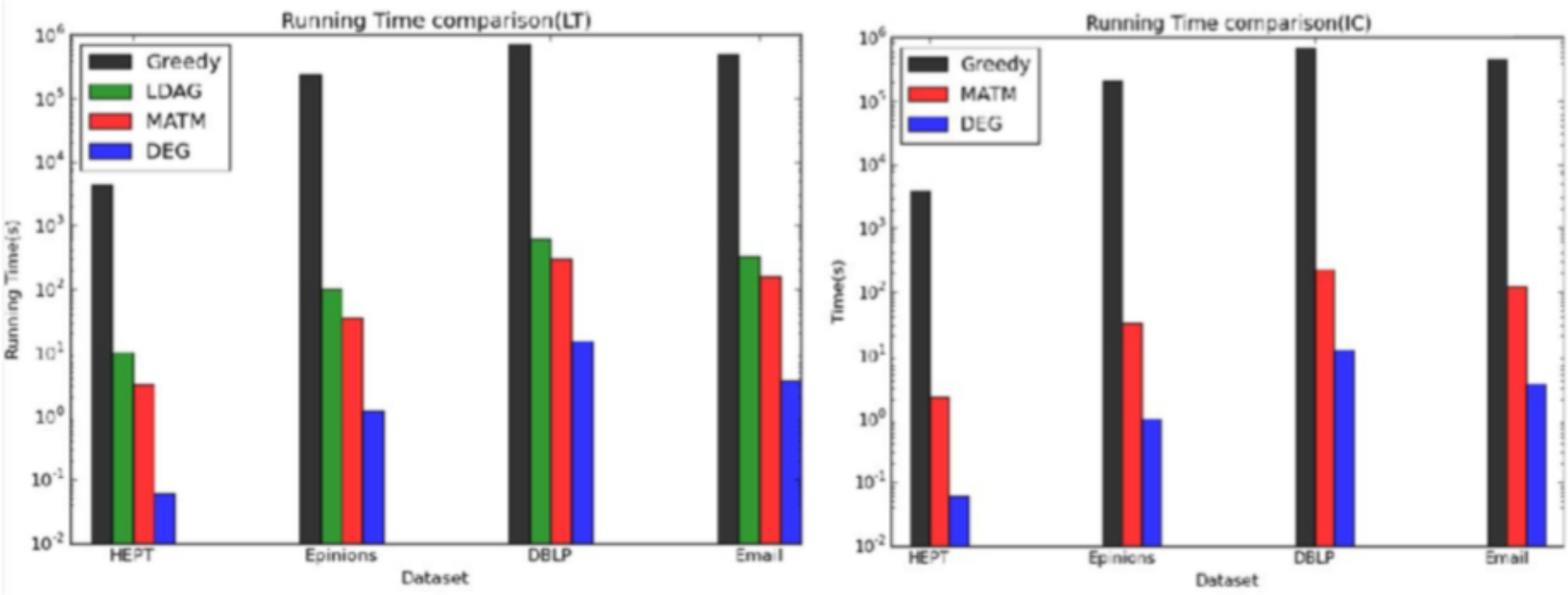


Figure: Time Comparison(Left: LT model, Right: IC model)

MATM is much faster than both greedy algorithm and LDAG. Though High-Deg is fast, it cannot output a high-quality seed set.

MATM – Memory requirements

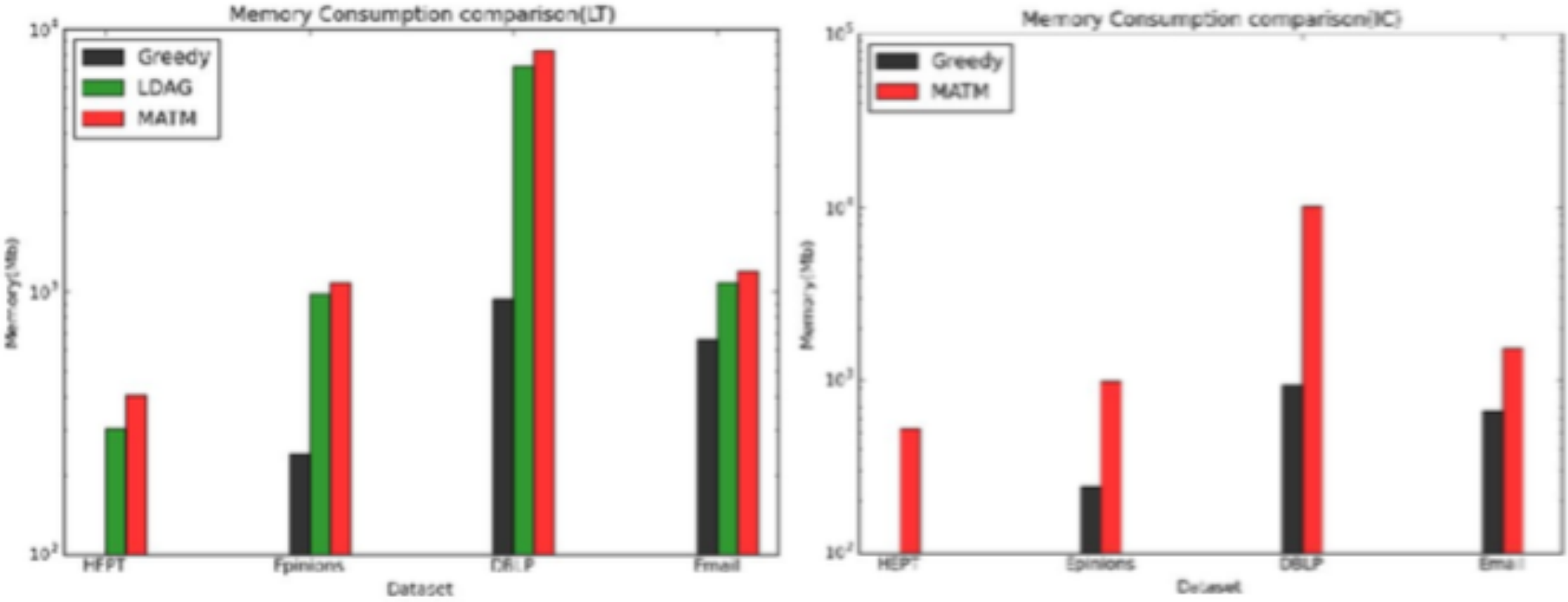


Figure: Memory Consumption Comparison(Left: LT model, Right: IC model)

MATM consumes large amount of memory compared with other algorithms

Adaptive Influence Maximization with Submodularity - Greedy Algorithm

Data: Graph $\mathcal{G} = (V, E)$, budget k, function f
Result: Greedy set A^{Greedy}
begin
  $A = \emptyset$
  while $|A| < k$ **do**
    $A = A \cup \arg\max_{v \in V \setminus A} \Delta(v|A) := f(A \cup \{v\}) - f(A)$
    (Exact computations or Monte Carlo simulations)
  end
  Return A
end

Adaptivity in the IF process

Key challenge : can we generalize submodularity (and monotonicity) for sequential decision making ?

Main contribution : [Golovin & Krause, JAIR 2011]

Adaptivity in the IF process

Problem statement of [Golovin & Krause, JAIR 2011] :

- IC Model, directed graph $\mathcal{G} = (V, E)$, **utility** $f(S, \phi)$
- $\phi : E \to O$ **realization** of the influence graph
  $\longrightarrow$ r.v. Φ, with known proba. distribution $p(\phi) := \mathbb{P}[\Phi = \phi]$
- **Partial realizations** $\psi \subseteq E \times O$
  $\longrightarrow$ **domain** of ψ: $dom(\psi)$?
  $\longrightarrow$ ψ **consistent** with ϕ : $\psi \sim \phi$?
  $\longrightarrow$ ψ **sub-realization** of ψ' ?
- We need to design a **policy** π : what does it mean ?

Optimization problem:

$$\pi^* \in \arg\max_{\pi} f_{avg}(\pi) := \mathbb{E}_{\Phi}[f(E(\pi, \Phi), \Phi)]$$

$$\text{s.t.} \quad |E(\pi, \phi)| \leq k, \forall \phi.$$

Adaptive monotonicity and submodularity

Definition (Conditional Expected Marginal Benefit)

Given a partial realization ψ and a node v, the **conditional expected marginal benefit** of v, conditioned on having observed ψ is :

$$\Delta(v|\psi) := \mathbb{E}\Big[f(dom(\psi) \cup \{v\}, \Phi) - f(dom(\psi), \Phi)|\Phi \sim \psi\Big]$$

Definition (Adaptive monotonicity and submodularity)

f is **adaptive monotone** if, for all ψ such that $\mathbb{P}(\Phi \sim \psi) > 0$, we have :

$$\Delta(v|\psi) \geq 0$$

f is **adaptive submodular** if, for all $\psi \subseteq \psi'$ (ψ sub-realization of ψ') and for all $v \in V \setminus dom(\psi')$ we have :

$$\Delta(v|\psi) \geq \Delta(v|\psi')$$

Full feedback vs Myopic feedback

[Golovin & Krause, JAIR 2011] studied IC model with:

- **Full feedback**: after choosing a seed node at time t, we observe the entire propagation in the graph at $t+1$
- and **Myopic feedback**: we only observe the status (active or not) of the neighbors of the seed node at $t+1$

For the Full feedback model :
(+) the set function f is adaptive monotone and submodular
(-) this model is, in general, not very realistic...

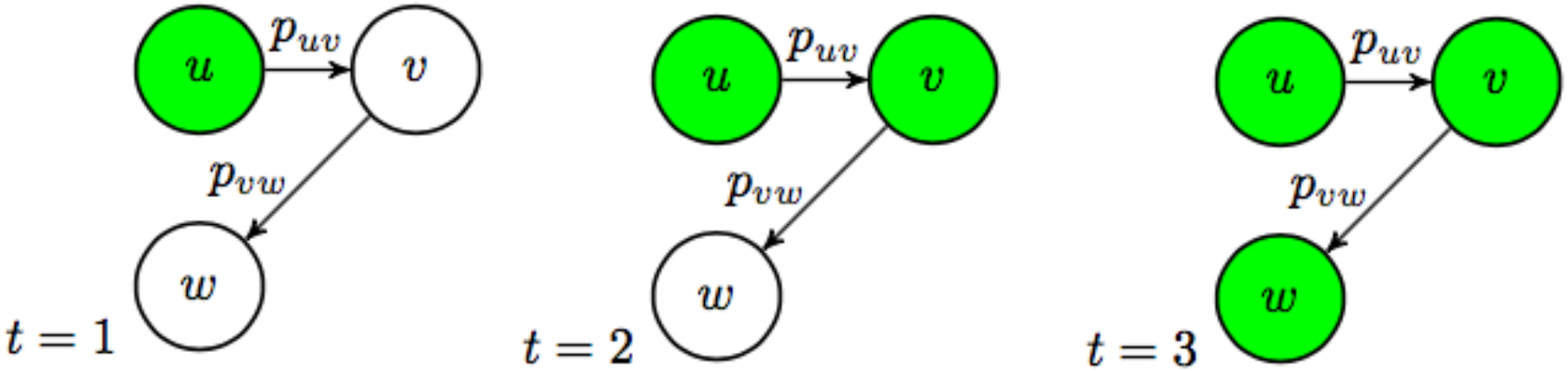
For the Myopic feedback model :
(+) this model is, in general, more realistic
(-) f is NOT adaptive submodular

Modifying the Utility function

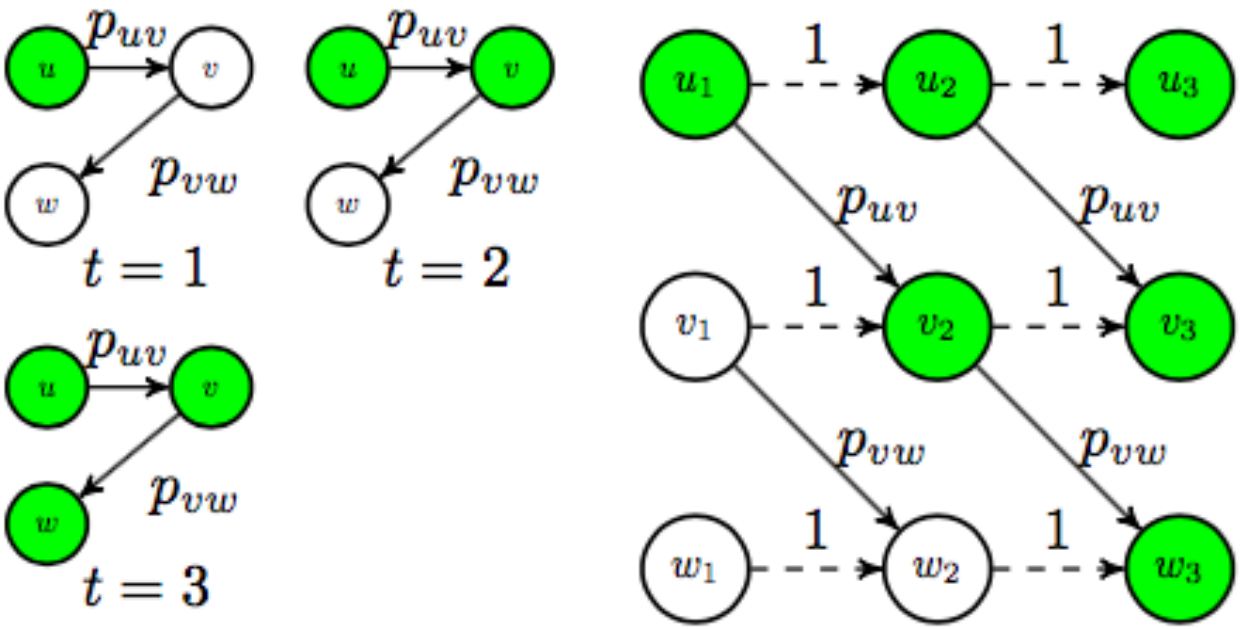
Finite **time horizon** T

Utility: cumulated number of active nodes over time

$$\tilde{f}(S,\phi) := \sum_{t=1}^{T} |\sigma_t(S,\phi)|$$



Layered graph representation



Lemma

For all set S of seed nodes (with corresponding time indexes) and for all realizations ϕ of the graph, $\tilde{f}_{\mathcal{G}}(S, \phi) = f_{\mathcal{G}^L}(S, \phi)$.

Analysis

Definition

Let Ψ be the set of all possible partial realizations. The **time function** $\mathcal{T} : \Psi \rightarrow \{1, ..., T\}$ returns, for a particular ψ, the largest time index from observed nodes and edges, and 1 if $\psi = \emptyset$.

Definition

The **marginal gain of choosing u as a seed node, having observed** ψ with $\mathcal{T}(\psi) = t$, and for the ground truth realization ϕ of the network, is:

$$\delta_\phi(u|\psi) := \tilde{f}_\mathcal{G}(dom(\psi) \cup \{u_t\}, \phi) - \tilde{f}_\mathcal{G}(dom(\psi), \phi).$$

Theoretical guarantees

Problem:

$$\pi^* \in \arg\max_{\pi} \tilde{f}_{avg}(\pi) := \mathbb{E}_{\Phi}[\tilde{f}_{\mathcal{G}}(E(\pi, \Phi), \Phi)] \text{ s.t. } |E(\pi, \phi)| \leq k, \forall \phi$$

Theorem

*The **adaptive greedy policy** π^G obtains at least $(1 - 1/e)$ of the value of the best policy for this A.I.M. problem, in the IC model with Myopic feedback:*

$$\tilde{f}_{avg}(\pi^G) \geq (1 - 1/e)\tilde{f}_{avg}(\pi^*).$$

Experiments

Three real-world networks from **Stanford's SNAP** website

Network	Nodes	Edges	Mean degree	Max. degree	A.P.L.	Diam.	Type
Twitter	228	9 938	43.6	125	2.1	6	Directed
ArXiv GR-QC	5 242	28 980	11.1	162	6.1	17	Undirected
Facebook	4 039	88 234	43.7	1 045	3.7	8	Undirected

Table : Statistics from the networks

Adaptive Greedy vs Alternative heuristics

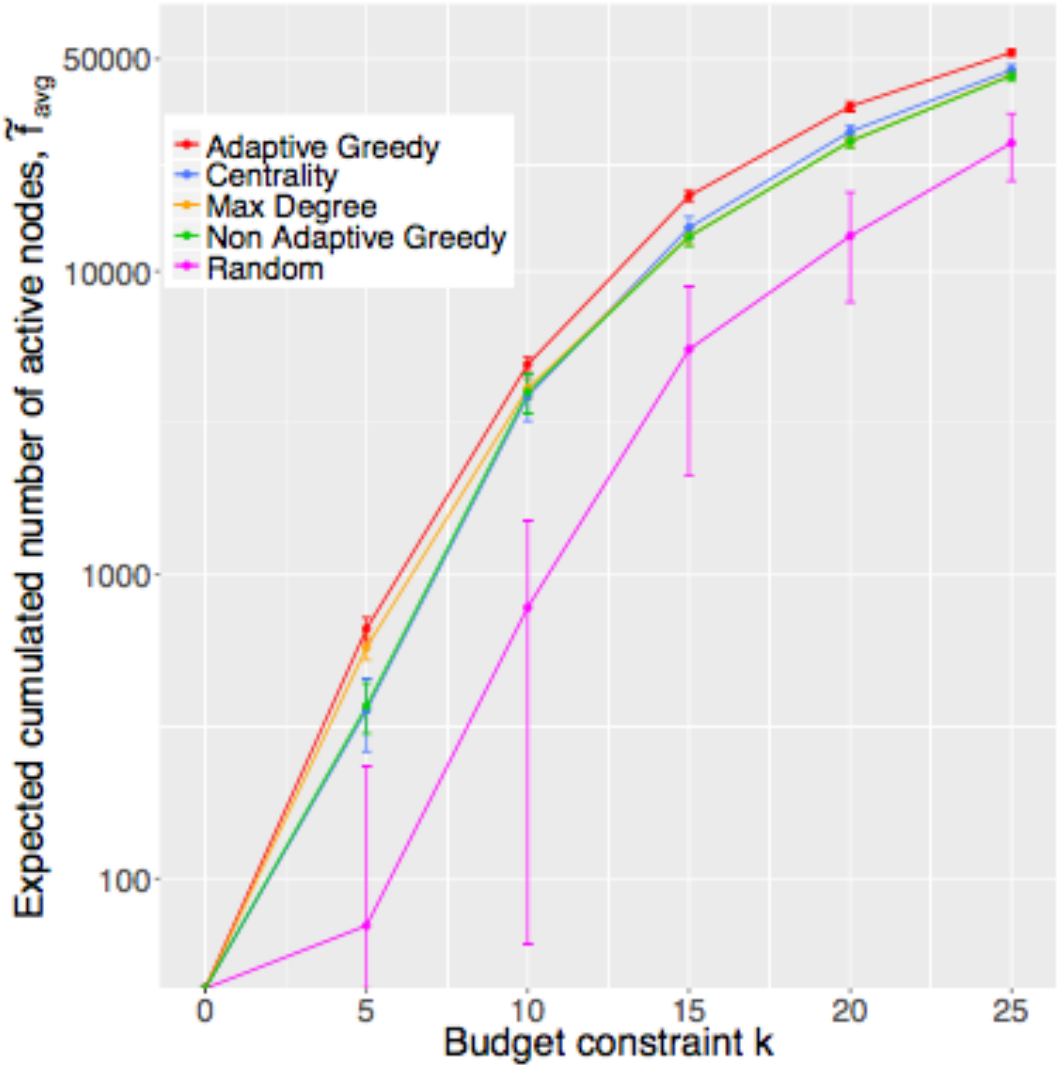


Figure : Arxiv, $p = 1/10$

Adaptive Greedy vs Alternative heuristics

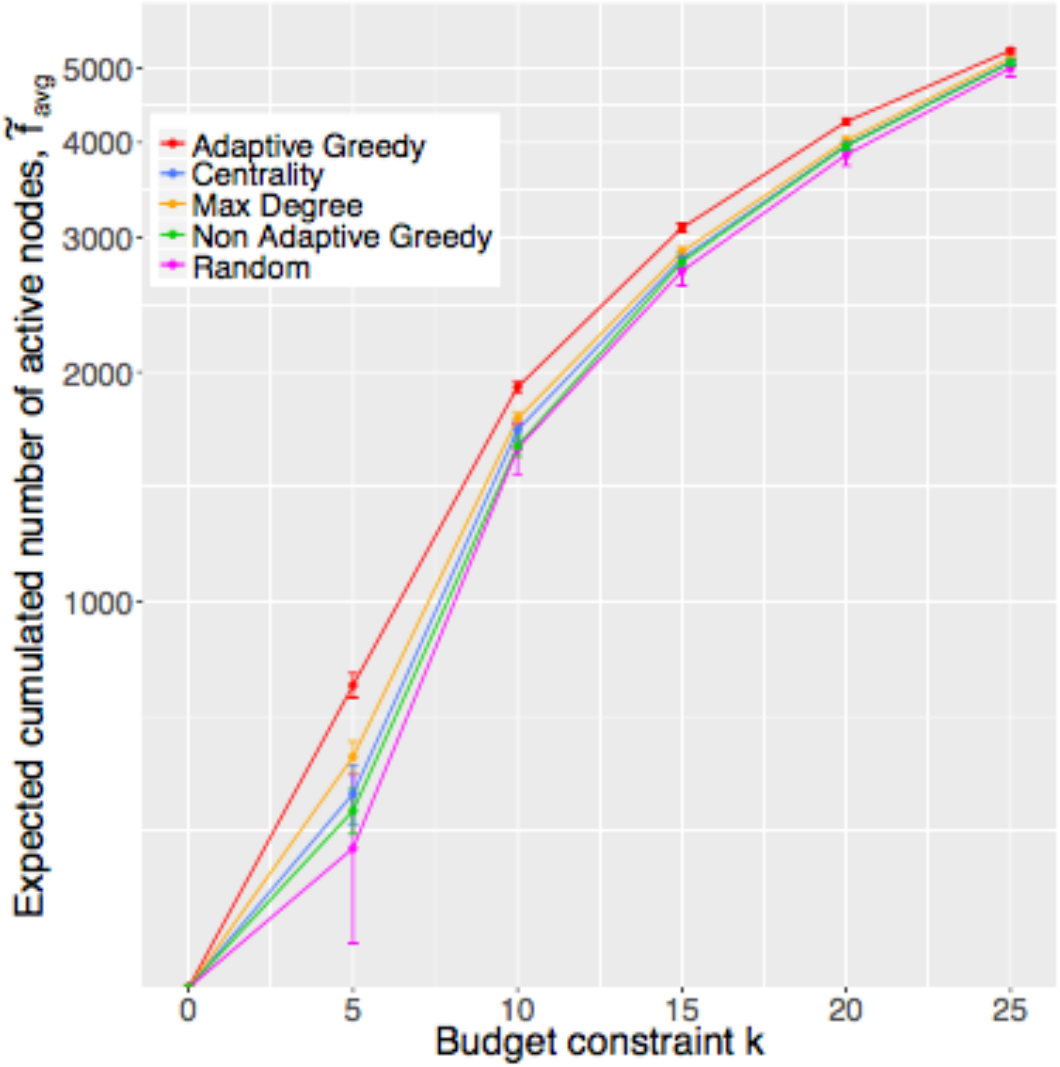


Figure : Twitter, $p = 1/10$

Adaptive Greedy vs Alternative heuristics

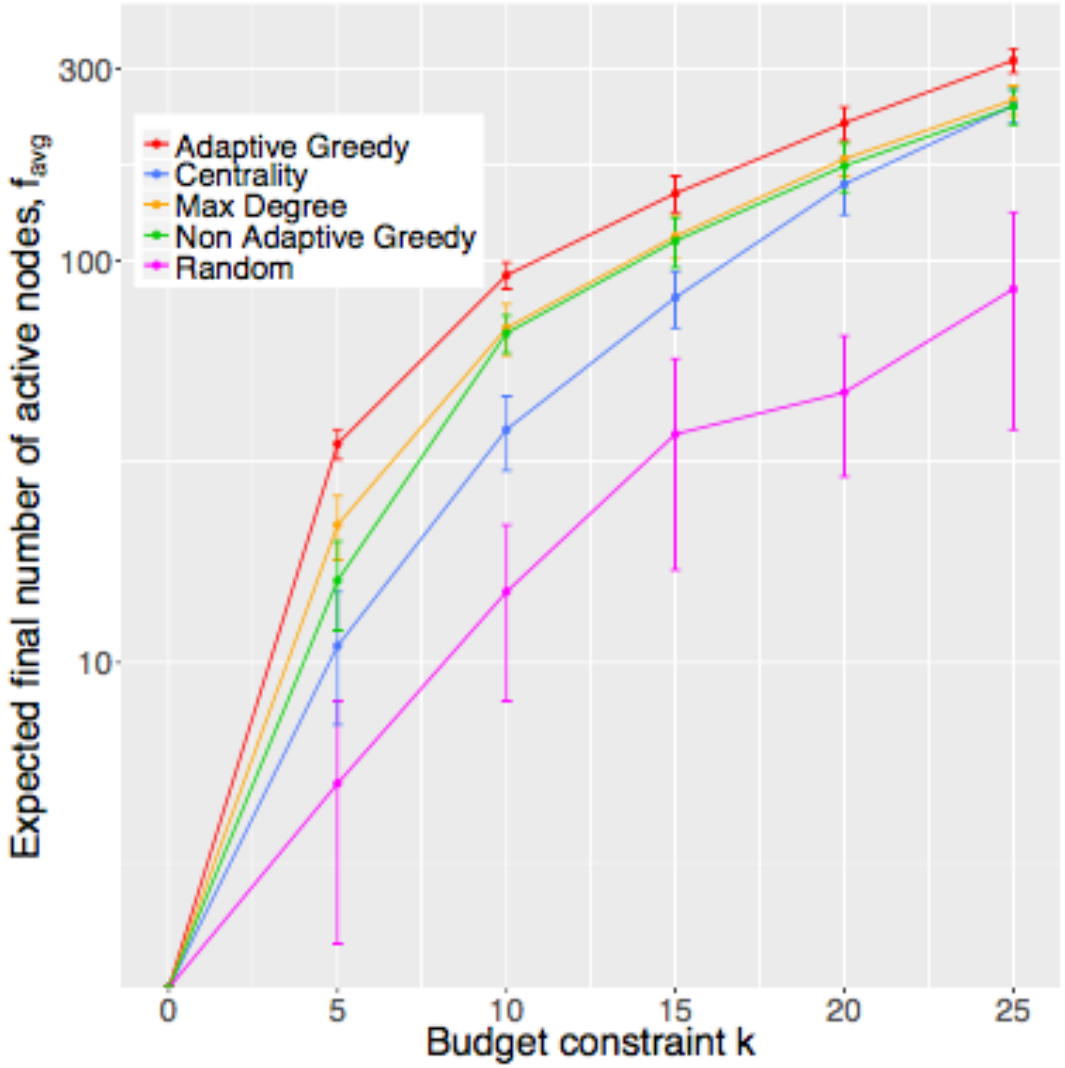


Figure : Final number of active nodes (Arxiv, $p = 1/100$)

References

1. Kempe, D., Kleinberg, J., & Tardos, É. (2003, August). Maximizing the spread of influence through a social network. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 137-146). ACM.

2. Newman, M., Barabasi, A. L., & Watts, D. J. (2006). The structure and dynamics of networks. Princeton University Press.

3. J. Wang and J. Cheng. Truss decomposition in massive networks. Proc. VLDB Endow., 5(9):812–823, 2012.

4. Vladimir Batagelj and Matjaz Zaversnik. An O(m) algorithm for cores decomposition of networks. CoRR, 2003.

5. Maksim Kitsak, Lazaros Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H. Eugene Stanley, and Herman Makse. Identication of influential spreaders in complex networks. Nature Physics, 6(11):888{893, Aug 2010.

6. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., & Glance, N. (2007, August). Cost-effective outbreak detection in networks. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 420-429). ACM.

References

7. Chen, W., Wang, C., & Wang, Y. (2010, July). Scalable influence maximization for prevalent viral marketing in large-scale social networks. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1029-1038). ACM.

8. Chen, W., Yuan, Y., & Zhang, L. (2010, December). Scalable influence maximization in social networks under the linear threshold model. In Data Mining (ICDM), 2010 IEEE 10th International Conference on (pp. 88-97). IEEE.

9. Goyal, A., Lu, W., & Lakshmanan, L. V. (2011, December). Simpath: An efficient algorithm for influence maximization under the linear threshold model. In Data Mining (ICDM), 2011 IEEE 11th International Conference on (pp. 211-220). IEEE.

10. Lü, L., Zhang, Y. C., Yeung, C. H., & Zhou, T. (2011). Leaders in social networks, the delicious case. PloS one, 6(6), e21202.

11. Chen, D. B., Gao, H., Lü, L., & Zhou, T. (2013). Identifying influential nodes in large-scale directed networks: the role of clustering.

References

12. Yu Wang et al. "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks." Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2010.

13. Chen, Yi-Cheng, Wen-Chih Peng, and Suh-Yin Lee. "Efficient algorithms for influence maximization in social networks." *Knowledge and information systems* 33.3 (2012): 577-601

14. Ma, Hao, et al. "Mining social networks using heat diffusion processes for marketing candidates selection." *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008.
15. Sketch-based Influence Maximization and Computation: Scaling up with Guarantees http://delivery.acm.org/10.1145/2670000/2662077/p629-cohen.pdf 996715&CFTOKEN=93026989&__acm__=1481908635_1ba4f4f9f496d3154af5026c9bcfd34e
16. Daniel P. Maki and Maynard Thompson. Mathematical models and applications: with emphasis on the social life, and management sciences. No. 511.8 M3. 1973