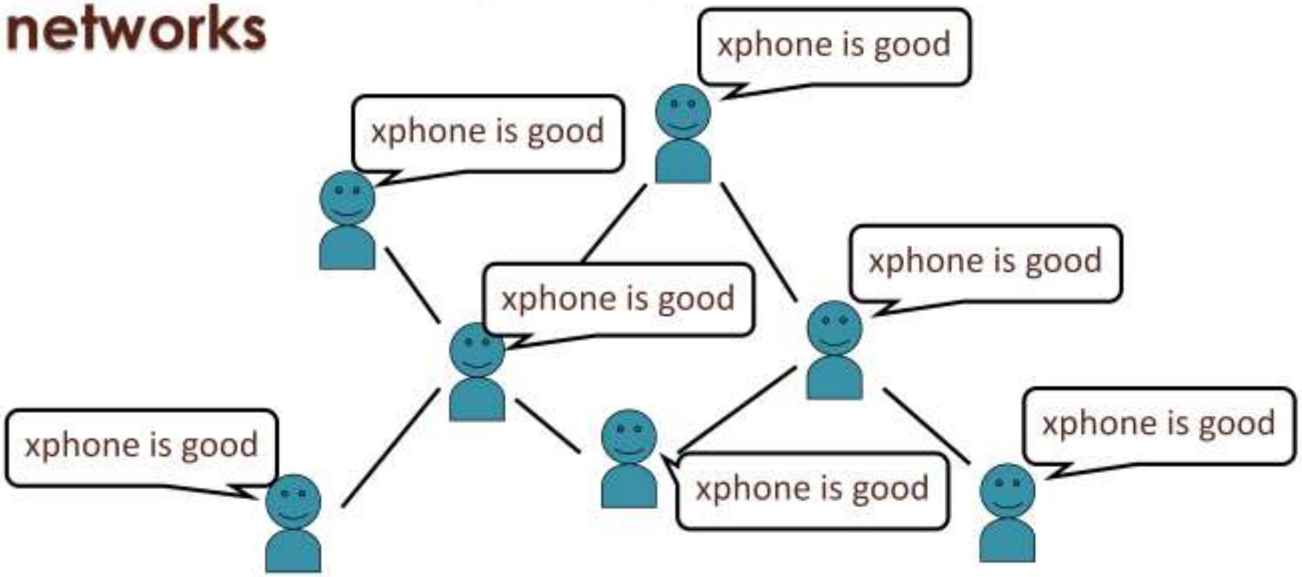
Part I → Part II → Part III → Part IV → Part V

Influence Maximization

Word-of-mouth (WoM) effect in social networks



- Word-of-mouth (viral) marketing is believed to be a promising marketing strategy.
- Increasing popularity of online social networks may enable large scale viral marketing

Outline

- Diffusion/propagation models and the Influence Maximization (IM) problem
- The theory: greedy methods to optimize submodular functions
- Scalable influence maximization

Diffusion/Propagation Models and the Influence Maximization (IM) Problem

The first definition of IM problem: A Markov random fields formulation

- Each node i has random variable X_i, indicating bought the product or not, $\boldsymbol{X} = \{X_1, \ldots, X_n\}$
- Markov random field formation: X_i depends on its neighbors' actions $\boldsymbol{N}_i \subseteq \boldsymbol{X}$
- Marketing action $\boldsymbol{M} = \{M_1, \ldots, M_n\}$
- **Problem**: find a choice of $\boldsymbol{M}$ that maximizes the revenue obtained from the result of $\boldsymbol{X}$

[Domingos & Richardson KDD 2001, 2002]

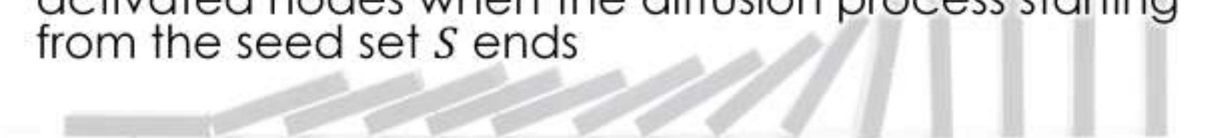
Matthew Richardson and Pedro Domingos:
Mining the network value of customers.
KDD 2001
http://doi.acm.org/10.1145/502512.502525

Matthew Richardson and Pedro Domingos:
Mining knowledge-sharing sites for viral marketing.
KDD 2002
http://doi.acm.org/10.1145/775047.775057

simplified model here, does not include product attributes $\boldsymbol{Y} = \{Y_1, \ldots, Y_k\}$.
Computation of revenue includes marketing cost, discount, and uses naïve Bayes model, etc. to simplify the computation.

Discrete diffusion models

- Static social network: $G = (V, E)$
  - V: set of nodes, representing individuals
  - E: set of edges, representing directed influence relationships
- Influence diffusion process
  - **Seed set S**: initial set of nodes selected to start the diffusion
  - **Node activations**: Nodes are activated starting from the seed nodes, in discrete steps and following certain stochastic models
  - **Influence spread $\sigma(S)$**: expected number of activated nodes when the diffusion process starting from the seed set S ends



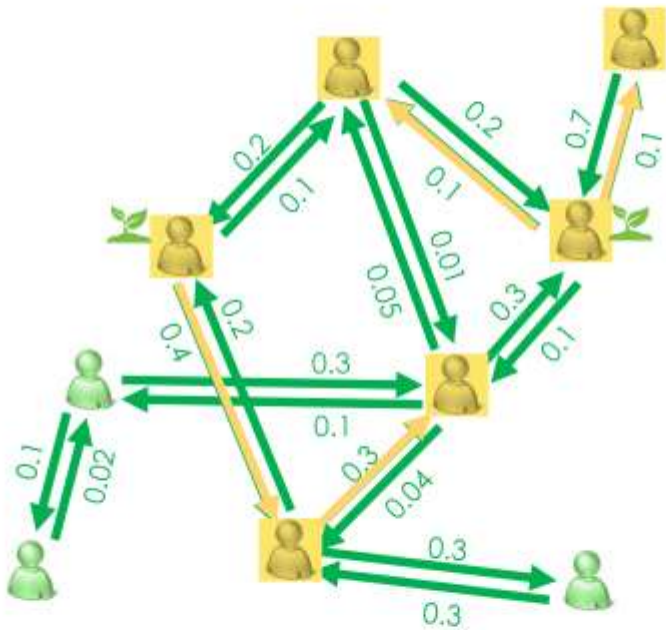
David Kempe, Jon Kleinberg, and Eva Tardos:
Maximizing the spread of influence through a social network.
KDD 2003
http://doi.acm.org/10.1145/956750.956769

Major stochastic diffusion models

- **Independent cascade** (IC) model
- **Linear threshold** (LT) model
- General threshold model
- Others
  - Voter model
  - Heat diffusion model

Independent cascade model

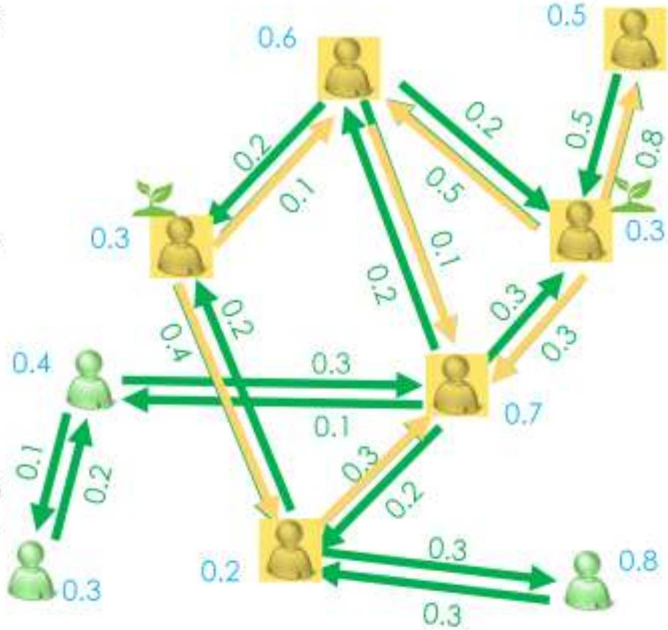
- Each edge (u, v) has a propagation probability $p(u, v)$
- Initially some seed nodes are activated
- At each step t, each node u activated at step $t-1$ activates its neighbor v with probability $p(u, v)$
- once activated, stay activated



[Kempe, Kleinberg and Tardos, KDD 2003]

Linear threshold model

- Each edge (u, v) has weight $w(u, v)$:
  - when $(u, v) \notin E, w(u, v) = 0$
  - $\sum_u w(u, v) \leq 1$
- Each node v selects a threshold $\theta_v \in [0,1]$ uniformly at random
- Initially some seed nodes are activated
- At each step, node v checks if the weighted sum of its active neighbors is greater than its threshold θ_v, if so v is activated
- once activated, stay activated



[Kempe, Kleinberg and Tardos, KDD 2003]

Influence maximization

- Given a social network, a diffusion model with given parameters, and a number k, find a seed set S of at most k nodes such that the influence spread of S is maximized.

- May be further generalized:
  - Instead of k, given a budget constraint and each node has a cost of being selected as a seed
  - Instead of maximizing influence spread, maximizing a (submodular) function of the set of activated nodes

Key takeaways for diffusion models and IM definition

- stochastic diffusion models
  - IC model reflects simple contagion
  - LT model reflects complex contagion (activation needs social affirmation from multiple sources [Centola and Macy, AJS 2007])
- maximization objective focuses on expected influence spread
  - others to be considered later

Damon Centola and Michael Macy: Complex Contagions and the Weakness of Long Ties.
American Journal of Sociology 2007
http://www.jstor.org/stable/10.1086/521848

Time for **Theory**: Optimizing Submodular Functions

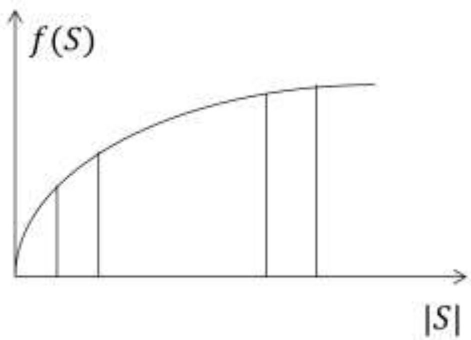
Credit: Rico3244 @ DeviantArt

Hardness of influence maximization

- Influence maximization under both IC and LT models are NP hard
  - IC model: reduced from k-max cover problem
  - LT model: reduced from vertex cover problem
- Need approximation algorithms

Optimizing submodular functions

- **Sumodularity** of set functions $f: 2^{V} \rightarrow R$
  - for all $S \subseteq T \subseteq V$, all $v \in V \setminus T$, $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$
  - diminishing marginal return
  - an equivalent form: for all $S, T \subseteq V$ $f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$



- **Monotonicity** of set functions f: for all $S \subseteq T \subseteq V$,

$$f(S) \leq f(T)$$

from $f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$ to prove $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$,

just consider two sets $S \cup \{v\}$ and T, and apply the first formula

For the opposite direction, Let $T \backslash S = \{u_1, u_2, \dots, u_k\}$, Let $T_j = \{u_1, u_2, \dots, u_j\}$, let $A_j = (S \cap T) \cup T_j$, and let $B_j = S \cup T_j$ we have

$f(A_j \cup \{u_{j+1}\}) - f(A_j) \geq f(B_j \cup \{u_{j+1}\}) - f(B_j), j = 0,1,2, \dots, j-1.$

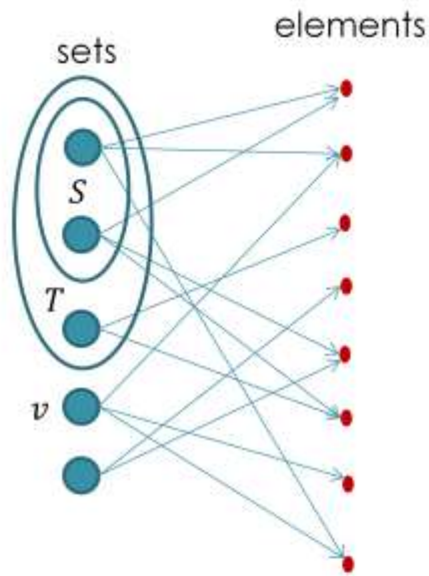
Summing up all these equations, we have

$f(T) - f(S \cap T) \geq f(S \cup T) - f(S).$

Example of a submodular function and its maximization problem

- set coverage
  - each entry u is a subset of some base elements
  - coverage $f(S) = |\bigcup_{u \in S} u|$
  - $f(S \cup \{v\}) - f(S)$: additional coverage of v on top of S

- k-max cover problem
  - find k subsets that maximizes their total coverage
  - NP-hard
  - special case of IM problem in IC model



Greedy algorithm for submodular function maximization

initialize $S = \emptyset$;
for $i = 1$ to k do
    select $u = \mathrm{argmax}_{w \in V \setminus S}[f(S \cup \{w\}) - f(S))]$
    $S = S \cup \{u\}$
end for
output S

Also referred as greedy hill-climbing algorithm, greedy approximation algorithm.

Property of the greedy algorithm

- Theorem: If the set function f is monotone and submodular with $f(\emptyset) = 0$, then the greedy algorithm achieves $(1 - 1/e)$ approximation ratio, that is, the solution S found by the greedy algorithm satisfies:
  - $f(S) \geq \left(1 - \frac{1}{e}\right) \max_{S' \subseteq V, |S'| = k} f(S')$

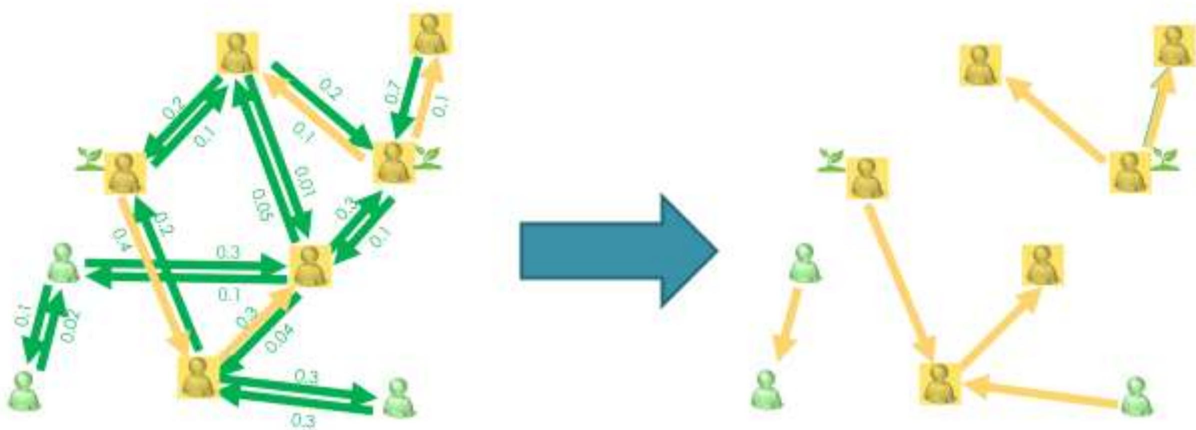
17

[Nemhauser, Wolsey and Fisher, Mathematical Programming, 1978]

G. L. Nemhauser, L. A. Wolsey and M. L. Fisher:
An analysis of approximations for maximizing submodular set functions. Mathematical Programming 1978
http://dx.doi.org/10.1007/BF01588971

Submodularity of influence diffusion models

- Based on equivalent live-edge graphs



diffusion dynamic

random live-edge graph: edges are randomly removed

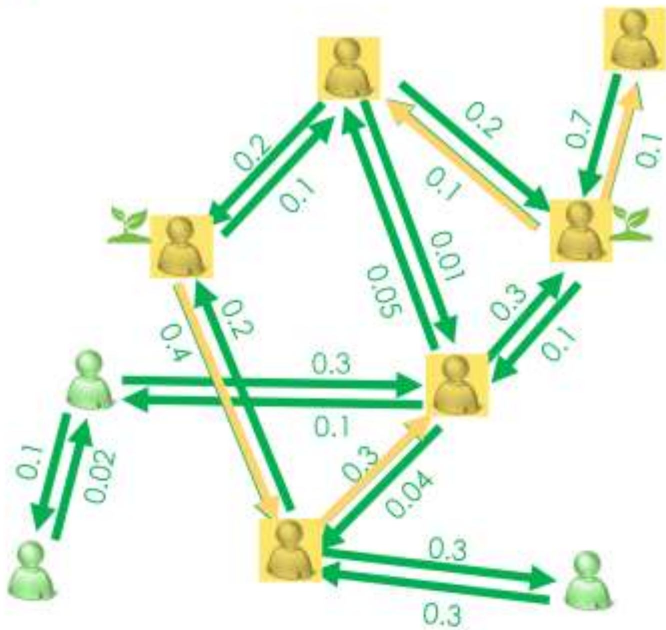
Pr(set A is activated given seed set S) = Pr(set A is reachable from S in random live-ledge graph)

Based on equivalent live-edge graphs

- Find a random edge selection process, such that each edge is either live or blocked, and all live edges form a random live-edge graph.
- Show that a diffusion model is equivalent to the reachability in random live-edge graphs: Given any seed set S, the distribution of active node sets after the diffusion process with seed set S ends is the same as the distribution of node set reachable from S in a random live-edge graph.
- Applicable to both IC and LT model.

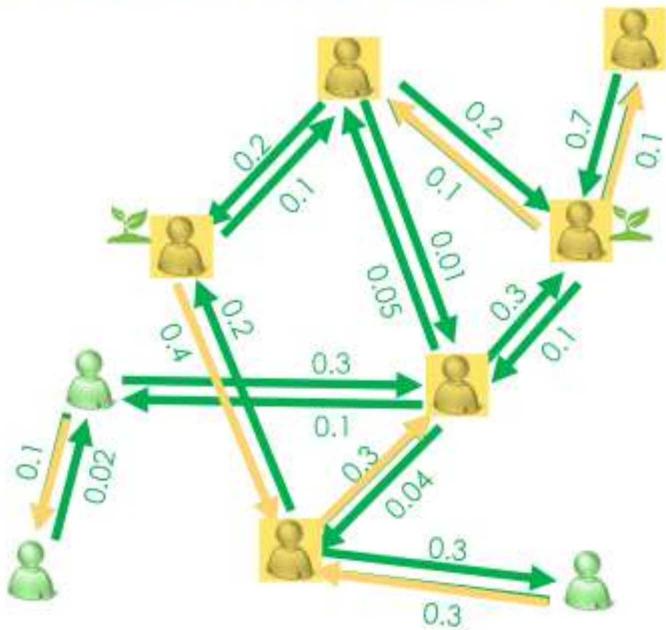
(Recall) active node set via IC diffusion process

- yellow node set is the active node set after the diffusion process in the independent cascade model



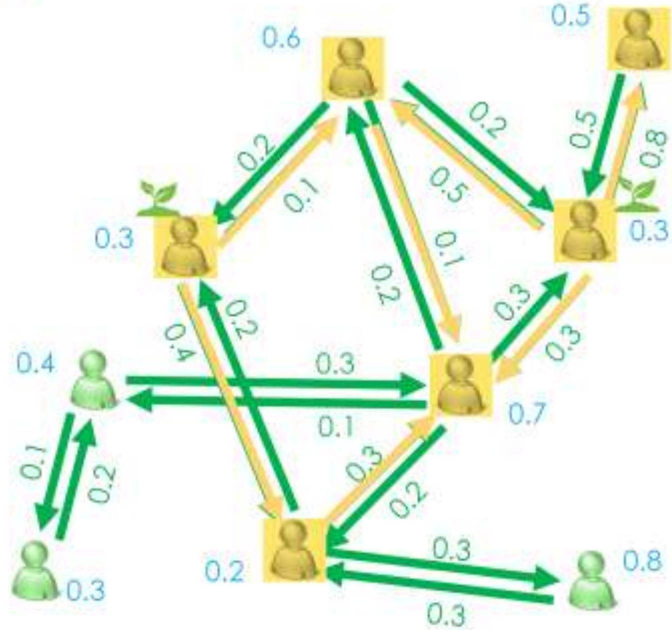
Random live-edge graph for the IC model and its reachable node set

- random live-edge graph in the IC model
  - each edge is independently selected as live with its propagation probability
- yellow node set is the active node set reachable from the seed set in a random live-edge graph
- Equivalence is straightforward



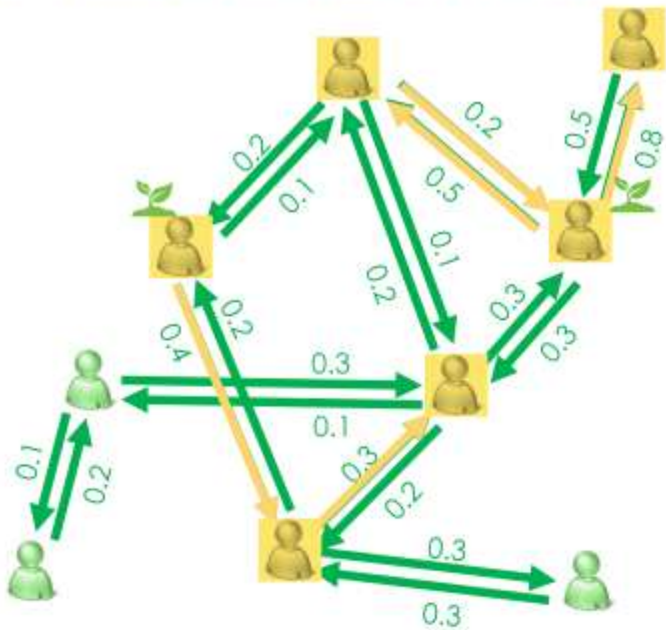
(Recall) active node set via LT diffusion process

- yellow node set is the active node set after the diffusion process in the linear threshold model



Random live-edge graph for the LT model and its reachable node set

- random live-edge graph in the LT model
  - each node select at most one incoming edge, with probability proportional to its weight
- yellow node set is the active node set reachable from the seed set in a random live-edge graph
- equivalence is based on uniform threshold selection from [0,1], and linear weight addition



Submodularity of influence diffusion models (cont'd)

- Influence spread of seed set S, $\sigma(S)$:

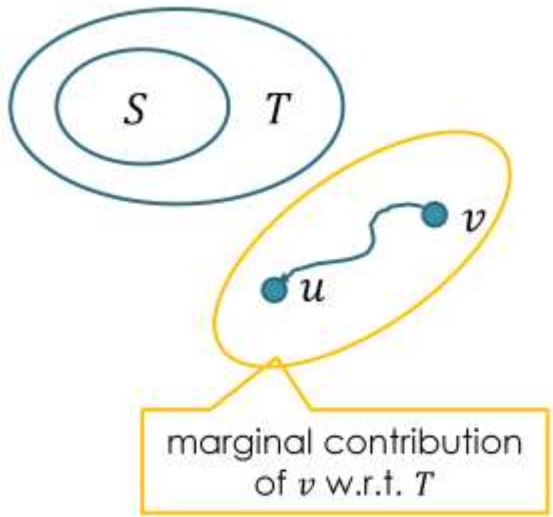
$$\sigma(S) = \sum_{G_L} \Pr(G_L)\,|R(S, G_L)|,$$

  - G_L: a random live-edge graph
  - $\Pr(G_L)$: probability of $G_L$being generated
  - $R(S, G_L)$: set of nodes reachable from S in G_L

- To prove that $\sigma(S)$ is submodular, only need to show that $|R(\cdot, G_L)|$ is submodular for any G_L
  - sumodularity is maintained through linear combinations with non-negative coefficients

Submodularity of influence diffusion models (cont'd)

- Submodularity of $|R(\cdot, G_L)|$
  - for any $S \subseteq T \subseteq V$, $v \in V \setminus T$,
  - if u is reachable from v but not from T, then
  - u is reachable from v but not from S
  - Hence, $|R(\cdot, G_L)|$ is submodular
- Therefore, influence spread $\sigma(S)$ is submodular in both IC and LT models



General threshold model

- Each node v has a threshold function
  $$f_v: 2^V \to [0,1]$$
- Each node v selects a threshold $\theta_v \in [0,1]$ uniformly at random
- If the set of active nodes at the end of step $t-1$ is S, and $f_v(S) \geq \theta_v$, v is activated at step t
- reward function $r(A(S))$: if $A(S)$ is the final set of active nodes given seed set S, $r(A(S))$ is the reward from this set
- generalized influence spread:
  $$\sigma(S) = E[r(A(S))]$$

[Kempe, Kleinberg and Tardos, KDD 2003]

IC and LT as special cases of general threshold model

- LT model
  - $f_v(S) = \sum_{u \in S} w(u, v)$
  - $r(S) = |S|$

- IC model
  - $f_v(S) = 1 - \prod_{u \in S}(1 - p(u, v))$
  - $r(S) = |S|$

Submodularity in the general threshold model

- Theorem [Mossel & Roch STOC 2007]:
  - In the general threshold model,
  - if for every $v \in V$, $f_v(\cdot)$ is monotone and submodular with $f_v(\emptyset) = 0$,
  - and the reward function $r(\cdot)$ is monotone and submodular,
  - then the general influence spread function $\sigma(\cdot)$ is monotone and submodular.
- Local submodularity implies global submodularity

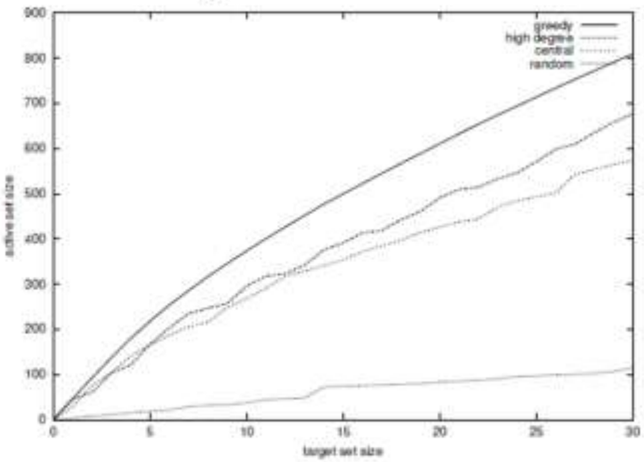
Elchanan Mossel, Sébastien Roch: On the submodularity of influence in social networks. STOC'2007. 128-134

The greedy approximation algorithm for the IM problem

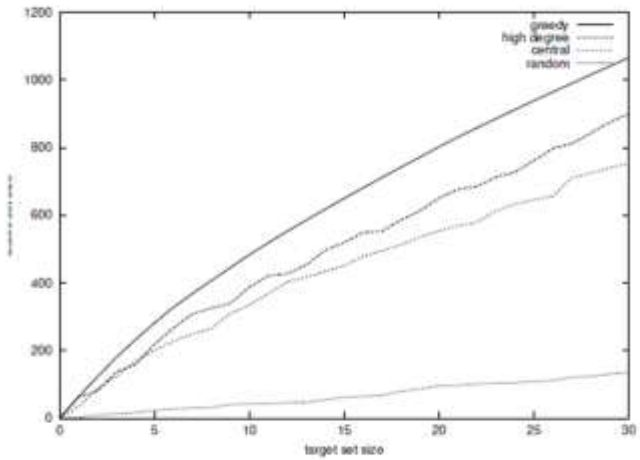
- $\left(1-\frac{1}{e}-\varepsilon\right)$-approximation greedy algorithm
  - Use general greedy algorithm framework
  - However, need evaluation of influence spread $\sigma(S)$, which is shown to be #P-hard
  - Use Monte Carlo simulations to estimate $\sigma(S)$ to arbitrary accuracy with high probability
  - For any $\varepsilon > 0$, there exists $\gamma > 0$, s.t. $\left(1-\frac{1}{e}-\varepsilon\right)$-approximation can be achieved with $1-\gamma$ approximate values of $\sigma(S)$

Performance of greedy algorithm

weighted cascade model



linear threshold model



- coauthorship graph from arxiv.org, high energy physics section, n=10748, m≈53000
- allow parallel edges, $c_{u,v}$ = number of edges between u and v
- weighted cascade: $p(u,v) = 1 - \left(1 - \frac{1}{d_v}\right)^{c_{u,v}}$
- linear threshold: $w(u,v) = c_{u,v}/d_v$

[Kempe, Kleinberg and Tardos, KDD 2013]

Key takeaways for theory support

- submodularity of diffusion models
  - diminishing return
  - shared by many models, but some model extensions may not be submodular (see Part IV)
- submodular function maximization
  - greedy hill-climbing algorithm
  - approximation ratio $(1 - 1/e)$

Scalable Influence Maximization

Theory versus Practice

...the trouble about arguments is, they ain't nothing but theories, after all, and theories don't prove nothing, they only give you a place to rest on, a spell, when you are tuckered out butting around and around trying to find out something there ain't no way to find out... There's another trouble about theories: there's always a hole in them somewheres, sure, if you look close enough.
- *"Tom Sawyer Abroad"*, *Mark Twain*

Inefficiency of greedy algorithm

- It take days to find 50 seeds for a 30K node graph
- Too many evaluations of influence spread
  - Given a graph of n nodes, each round needs evaluation of n influence spreads, totally $O(nk)$ evaluations
- Each evaluation of influence spread is hard
  - Exact computation is #P-hard, for both IC and LT models [Chen et al. KDD/ICDM 2010]
  - Monte-Carlo simulation is very slow

Scalable Influence Maximization

- Reduction on the number of influence spread evaluations.
- Batch computation of influence spread
- Scalable heuristics for influence spread computation

Lazy forward optimization

- Exploiting submodularity, significantly reduce # influence spread evaluations
  - S_{t-1} – seed set selected after round $t-1$
  - v_t – selected as seed in round t: $S_t = S_{t-1} \cup \{v_t\}$
  - u is not a seed yet, u's marginal gain $MG(u|S_t) = \sigma(S_t \cup \{u\}) - \sigma(S_t)$
  - by submodularlity, $MG(u|S_t) \leq MG(u|S_{t-1})$
  - This implies, if $MG(u|S_{t-1}) \leq MG(v|S_t)$ for some node v, then no need to evaluate $MG(u|S_t)$ in round $t+1$.
  - Can be implemented efficiently using max-heap
    - take the top of the heap, if it has MG for the current round, then it is the new seed;
    - else compute its MG, and re-heapify
- Often, top element after round $k-1$ remains top in round k.
- Up to 700 X speedup

35

[Leskovec et al., KDD 2007]

Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance NS (2007) Cost-effective outbreak detection in networks. In: KDD '07
Chen W, Wang Y, Yang S (2009)

CELF++

- With each heap node u, further maintain
  - $u.mg1 = MG(u|S)$, S = current seed set.
  - $u.prev\_best$ = node with max. MG in the current iteration, among nodes seen before u;
  - $u.mg2 = MG(u|S \cup \{u.prev\_best\})$;
  - $u.flag$ = iteration where $u.mg1$ was last updated;
- $u.prev\_best$ chosen in current iteration ➔ no need to compute $MG(u|S \cup \{u.prev\_best\})$ in next iteration.
- $MG(u|S \cup \{u.prev\_best\})$ can be computed efficiently along with $MG(u|S)$ in one run of MC.

36

[Goyal, Lu and L., WWW 2011]

Amit Goyal, Wei Lu, and Laks V.S. Lakshmanan. CELF++: Optimizing the Greedy Algorithm for Influence Maximization in Social Networks. In WWW 2011 (poster).

CELF++ (contd.)

- Pick the top node u in the heap
- $u.flag = |S|$ ➔ $u.\mathrm{mg1}$ is correct MG, thus u is the best pick of current itn; add u to S; remove from Q;
- else, if $u.prev\_best = last\_seed$ & $u.flag = |S| - 1$ ➔ $u.mg1 \leftarrow u.mg2$ (no need to compute $MG(u|S + last\_seed)$)

Dataset	Running time (min)			Avg. # node lookups		
	CELF	CELF++	Gain	CELF	CELF++	Gain
Hept WC	245	**178**	27%	18.7	**13.6**	27.2%
Hept IC	5269	**2082**	60.5%	190.5	**113.0**	40.7%
Phy WC	1242	**1028**	17.2%	18.6	**17.9**	3.8%

Table 1: Comparison between CELF and CELF++. Number of seeds = 100 in all test cases.

Batch computation of influence spread

- Based on equivalence to reachable set in live-edge graphs
  - Generate a random live-edge graph
  - batch computation of size of reachable set of all nodes [Cohen, JCSS 1997]
  - repeat above to increase accuracy
- In conflict with lazy-forward optimization
  - MixedGreedy: In the first round uses reachability-based batch computation; in remaining rounds, use lazy forward optimization

38

[Chen, Wang and Yang, KDD 2009]

Chen W., Wang Y., and Yang S., Efficient influence maximization in social networks. In: KDD '09. [pdf]

E. Cohen. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. Syst. Sci.*, 55(3):441–453, 1997.

Scalable heuristics for influence spread computation

- MIA: for IC model [Chen, Wang and Wang, KDD 2010]
- LDAG: for LT model [Chen, Yuan and Zhang, ICDM 2010]
- Features:
  - Focus on a local region
  - Find a graph spanner allow efficient computation
  - Batch update of marginal influence spread

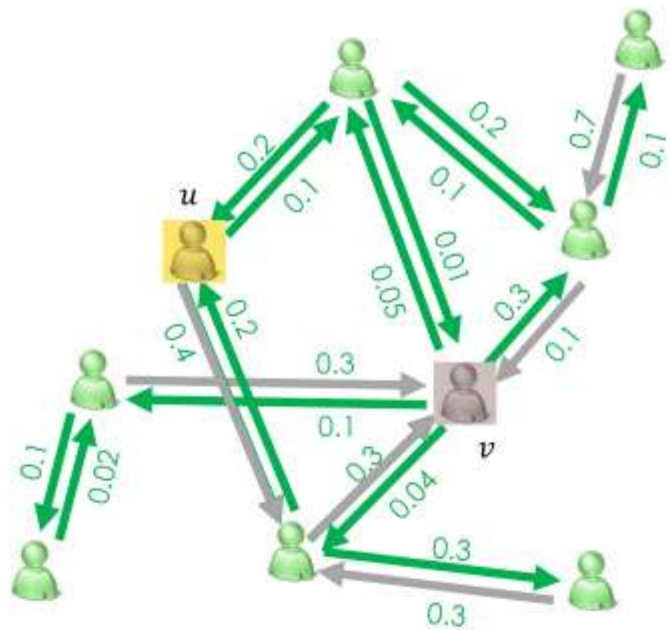
39

Chen W., Wang C,, and Wang Y. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: KDD '10. [pdf]
Journal version:
Chi Wang, Wei Chen, and Yajun Wang. Scalable influence maximization for independent cascade model in large-scale social networks. Data Mining and Knowledge Discovery Journal, to appear, 2012. [pdf]

Chen W., Yuan Y., Zhang L. Scalable influence maximization in social networks under the linear threshold model. In: ICDM '10. [pdf][full technical report: MSR-TR-2010-133]

Maximum Influence Arborescence (MIA) Heuristic

- Local influence regions of node v
  - For every nodes u, find the maximum influence path (MIP) from u to v, ignore it if $\Pr(path) < \lambda$ (λ is a small threshold value)
  - all MIPs to v form its maximum influence in-arborescence (MIIA)
  - MIIA can be efficiently computed
  - influence to v is computed over MIIA



MIA Heuristic III: Computing Influence through the MIA structure

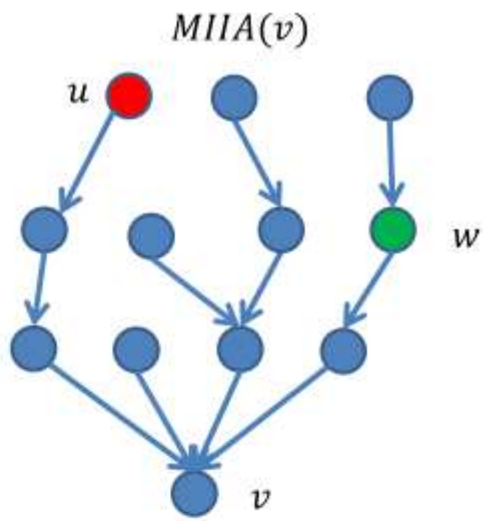
- Recursive computation of activation probability $ap(u)$ of a node u in its in-arborescence, given a seed set S ($N^{in}(u)$ is the in-neighbors of u in its in-arborescence)

Algorithm 2 Computing activation probability of u, $ap(u, S)$

if $u \in S$ **then**
    $ap(u) = 1$
else if $N^{in}(u) = \emptyset$ **then**
    $ap(u) = 0$
else
    $ap(u) = 1 - \Pi_{w \in N^{in}(u)}(1 - ap(w) \cdot p(w, u))$
end if

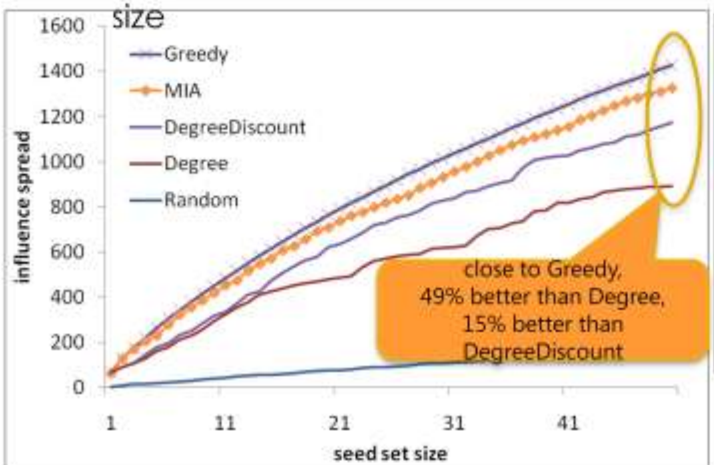
MIA Heuristic IV: Efficient updates on incremental activation probabilities

- u is the new seed in $MIIA(v)$
- Naive update: for each candidate w, redo the computation in the previous page to compute w's marginal influence to v
  - $O(|MIIA(v)|^2)$
- Fast update: based on linear relationship of activation probabilities between any node w and root v, update marginal influence of all w's to v in two passes
  - $O(|MIIA(v)|)$

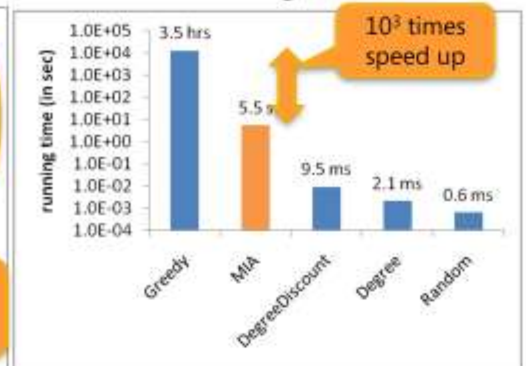


Experiment Results on MIA heuristic

Influence spread vs. seed set size



running time



Experiment setup:

- 35k nodes from coauthorship graph in physics archive
- influence probability to a node v = 1 / (# of neighbors of v)
- running time is for selecting 50 seeds

Time for some simplicity

The SimPath Algorithm

Vertex Cover Optimization

Improves the efficiency in the first iteration

Look ahead optimization

Improves the efficiency in the subsequent iterations

In lazy forward manner, in each iteration, add to the seed set, the node providing the maximum marginal gain in spread.

[Goyal, Lu, & L. ICDM 2011]

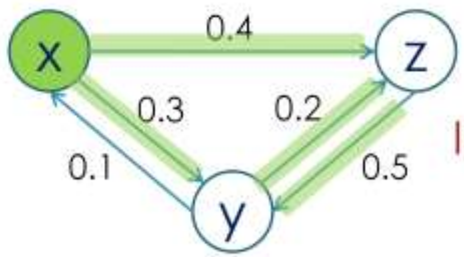
Simpath-Spread

Compute marginal gain by enumerating simple paths

Amit Goyal, Wei Lu, and Lakshmanan. Simpath: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model. In ICDM 2011.

Estimating Spread in SimPath (2)

- Thus, the spread of a node can be computed by enumerating simple paths starting from the node.



Influence of x on z

Influence of x on y

Influence of x on x itself

Total influence of node x is 1.96

Influence Spread of node $= \boxed{\Upsilon_{x,x}} + \boxed{\Upsilon_{x,y}} + \boxed{\Upsilon_{x,z}}$

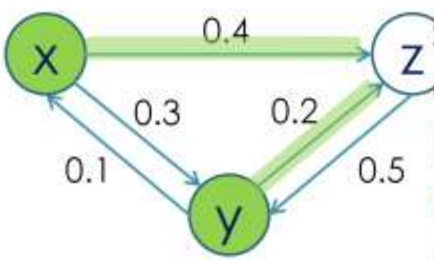
$= \boxed{1} + \boxed{(0.3 + 0.4 * 0.5)} + \boxed{(0.4 + 0.3 * 0.2)} = \boxed{1.96}$

Estimating Spread in SimPath (3)

Theorem 1. *In the LT model, the spread of a set S is the sum of the spread of each node $u \in S$ on subgraphs induced by $V - S + u$. That is,*

$$\sigma(S) = \sum_{u \in S} \sigma^{V-S+u}(u)$$

Total influence of the seed set {x, y} is 2.6



Influence of node y in a subgraph that does not contain x

Influence of node x in a subgraph that does not contain y

Let the seed set S = {x,y}, then influence spread of S is

$$\sigma(S) = \boxed{\sigma^{V-y}(x)} + \boxed{\sigma^{V-x}(y)} = \boxed{1+0.4} + \boxed{1+0.2} = \boxed{2.6}$$

Estimating Spread in SimPath (4)

~~Enumerating all simple paths is #P hard~~

Thus, influence can be estimated ~~by~~ enumerating all simple paths starting from the seed set.

On slightly different subgraphs

The majority of influence flows in a small neighborhood.

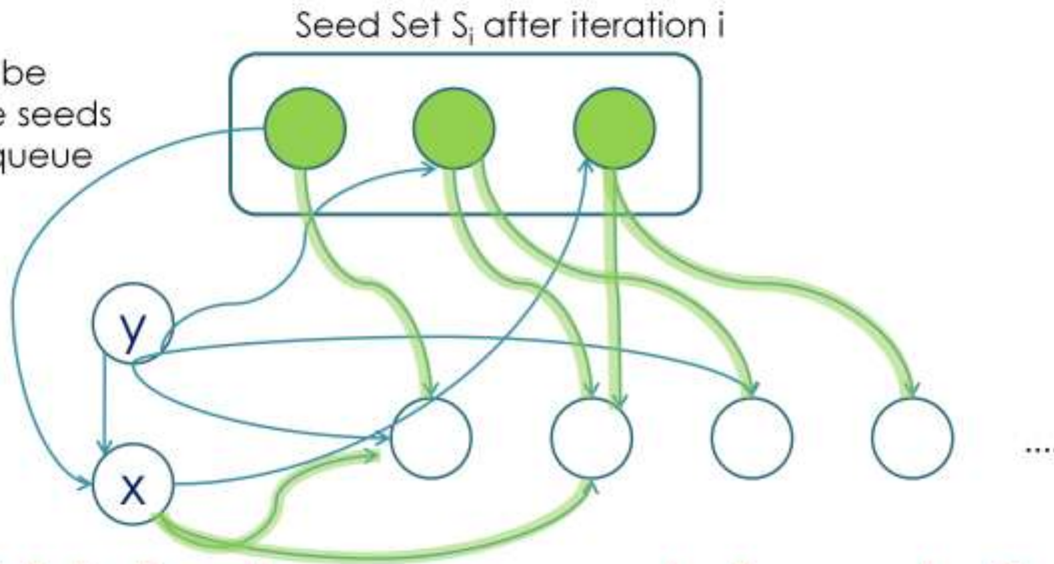
Look Ahead Optimization (1/2)

- As the seed set grows, the time spent in estimating spread increases.
  - More paths to enumerate.
- A lot of paths are repeated though.
- The optimization avoids this repetition intelligently.
- A look ahead parameter 'l'.

Look Ahead Optimization (2/2)

l = 2 here

Let y and x be prospective seeds from CELF queue



A lot of paths are enumerated repeatedly

1. $\sigma(S_i + x) =$ ad achieved by S+y
2. ad achieved by S+x

$\boxed{\sigma^{V-S_i}(x)} + \boxed{\sigma^{V-x}(S_i)}$

SimPath vs LDAG

TABLE II
SIMPATH'S IMPROVEMENT OVER LDAG

Dataset	Improvement in		
	Spread	Running Time	Memory
NetHEPT	8.7%	21.7%	62.9%
Last.fm	1.7%	42.9%	86.5%
Flixster	8.9%	33.6%	87.5%
DBLP	2.3%	67.2%	87.1%

Other means of speedup

- Community-based approach

  [Wang et al. KDD 2010]

- Network sparsification

  [Mathioudakis et al, KDD 2011]

- Simulated Annealing

  [Jiang et al. AAAI 2011]

Use this slide in lieu of the next three if running out of time.

Community-based influence maximization

- Use influence parameters to partition graph into communities
  - different from pure structure-based partition
- Influence maximization within each community
- Dynamic programming for selecting top seeds
- Orthogonal with scalable influence computation algorithms

[Wang et al. KDD 2010]

Yu Wang, Gao Cong, Guojie Song, Kunqing Xie: Community-based greedy algorithm for mining top-K influential nodes in mobile social networks. KDD 2010: 1039-1048

Sparsification of influence networks

- Remove less important edges for influence propagation
- Use action traces to guide graph trimming
- Orthogonal to scalable influence computation algorithms

[Mathioudakis et al, KDD 2011]

Michael Mathioudakis, Francesco Bonchi, Carlos Castillo, Aristides Gionis, Antti Ukkonen: Sparsification of influence networks. KDD 2011: 529-537

Simulated annealing

- Following simulated annealing framework
  - not going through all nodes to find a next seed as in greedy
  - find a random replacement (guided by some heuristics)
- Orthogonal to scalable influence computation algorithms

Qingye Jiang, Guojie Song, Gao Cong, Yu Wang, Wenjun Si, Kunqing Xie: Simulated Annealing Based Influence Maximization in Social Networks. AAAI 2011

Key takeaways for scalable influence maximization

- tackle from multiple angles
  - reduce #spread evaluations: lazy-forward
  - scale up spread computation:
    - use local influence region (LDAG, MIA, SimPath)
    - use efficient graph structure (trees, DAGs)
    - model specific optimizations (simple path enumerations)
  - graph sparsifications, community partition, etc.

- upshot: can scale up to graphs with millions of nodes and edges, on a ***single*** machine