Influence Maximization in Social Networks under Linear Threshold with Negative Opinion

Zilong Guo
515030910556

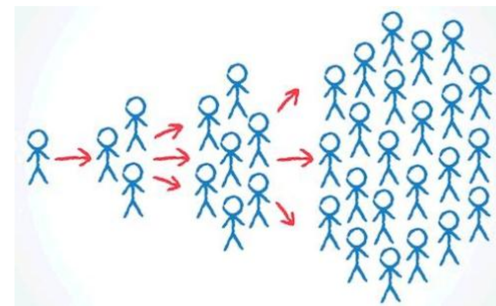
A small note:

Only the key idea is shown in this ppt. For detailed proof, algorithms and reference, please refer to the report

Advertising

People are widely and closely connected by social networks

Viral marketing: uses existing social networks to promote a product



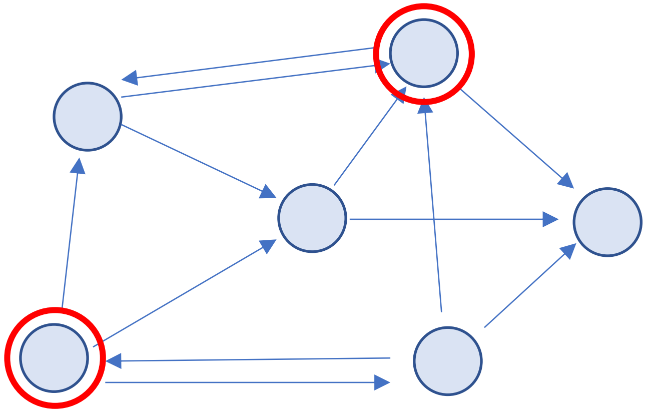
Advertise the right thing to the right person

Maximum profit

Which one(s) should we advertise to?

Here comes the Influence maximization problem:

Finding a small set of seed nodes S in a social network that maximizes the spread of influence under certain influence cascade models



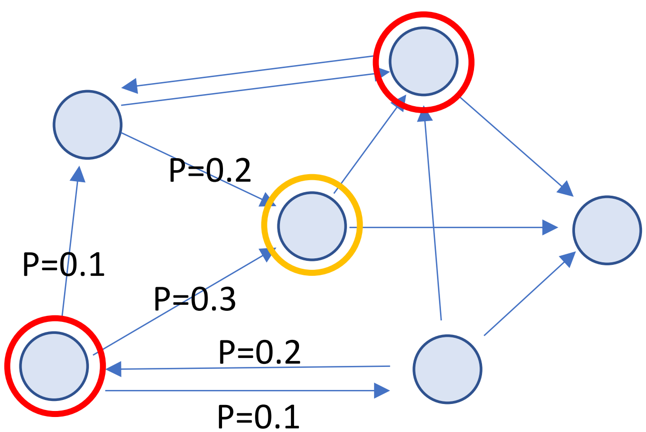
Two basic model

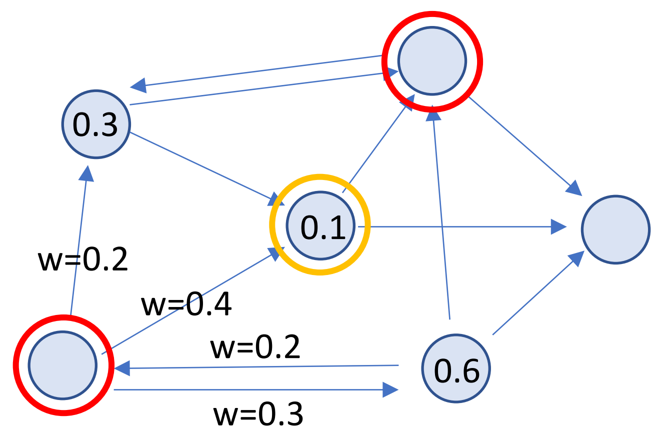
Probabilistic view

independent cascading (IC) model

Quantitative view

Linear threshold (LT) model





NP-hard

A lot less studied!

Negative Opinion

Negative opinion is **pervasive** and **influential**

However, negative opinion is seldom studied in influence maximization
(Only two paper as far as I know)

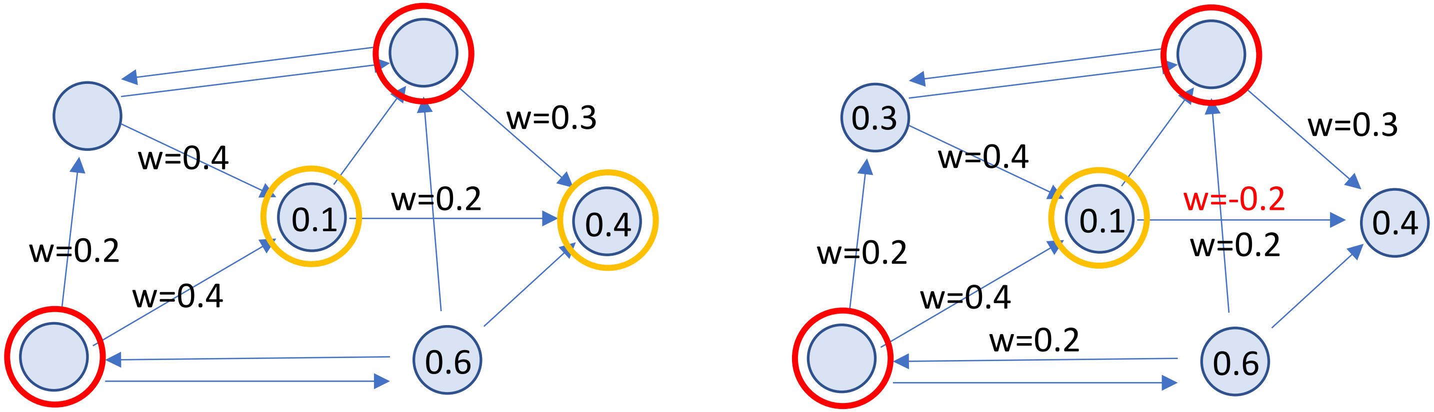
LT model with negative opinion (LT-N)

Formal Definition:

Social networks are modeled as a directed graph $\mathbf{G=(V, E)}$.

- Each edge in $\mathbf{G}$ has a weight $\mathbf{w}$.
- For any in-activated vertex $\mathbf{u, v}$ that $(\mathbf{u, v}) \in \mathrm{E}$, $0 < w(u, v) \leq 1$.
- For any in-activated node v, $\sum_{\{u:(u, v) \in E\}} w(u, v) \leqslant 1$.
- Each node $\mathbf{v}$ has a threshold $\boldsymbol{\lambda}$ which $\mathbf{0 < \lambda \leq 1}$.
- A satisfaction probability $\mathbf{q}$ is introduced as the probability that one node turns positive after activation. If a node turns negative, its influence turn negative.

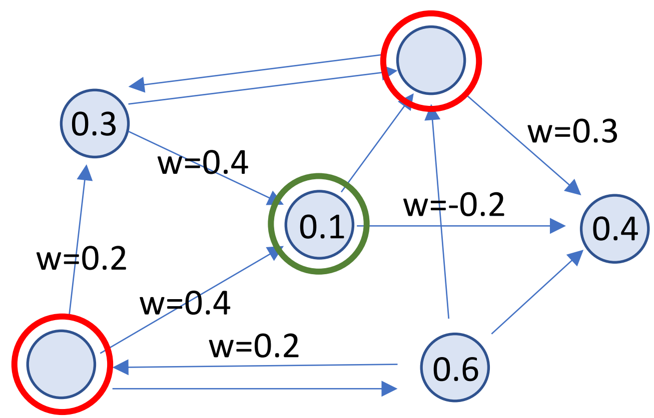
$\mathbf{Inf(S)}$ is defined as the number of positive activated nodes



LT model with negative opinion (LT-N)

This model captures real world phenomena:

- Product defect or some other unhappy experience is the source of negative opinion
- Consumers take both positive and negative opinion into consideration



Two important properties

Monotonicity

For node sets $S \subseteq T$ and, $Inf(S) \leq Inf(T)$

Submodularity

For node sets $S \subseteq T \subseteq V$ and a node $v \in V \backslash B$, $Inf(S \cup \{v\})-Inf(S) \geq Inf(T \cup \{v\})-Inf(T)$

*They only hold true when q>0.5

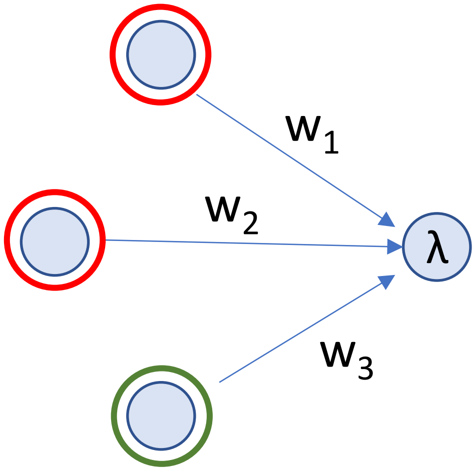
Why they are important?

Lemma 1:

For any monotone and submodular set function f with $f(\emptyset)=0$, the greedy algorithm guarantee an $1\text{-}1/e$ approximation.

Key idea of proof

Key idea: treat this quantitative problem in a probabilistic view



For a node v with a threshold λ, what is the probability that being activated in this condition?

$P(v) = \max(0, w_1 + w_2 - w_3)$ for λ is uniformly random in $(0,1]$

Then, many probabilistic analysis can be put into use

Overview

a Greedy algorithm

b Local directed acyclic graph with negative opinion (LDAG-N)

c Evolutionary algorithm (EA)

Greedy algorithm

In each round, select the nodes that could cause maximum incremental influence

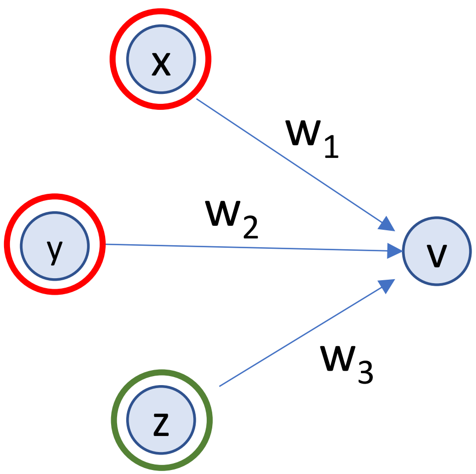
Algorithm 1 Greedy Algorithm

1: $S = \emptyset$
2: **for** i=1 to k **do**
3: $\quad u = argmax_u Inf(S \cup \{u\})$
4: $\quad S = S \cup \{u\}$
5: **end for**
6: **return** S

Inf(S) in computed by Mont-Carlo simulation, which is very slow.
Have a approximation guarantee of 1-1/e

LDAG-N

Key idea: influence can be calculated in linear time in a DAG. So, we can construct local DAGs to approximate the influence



Remember:

$P(v) = \max(0, w_1 + w_2 - w_3)$ for λ is uniformly random in $(0,1]$

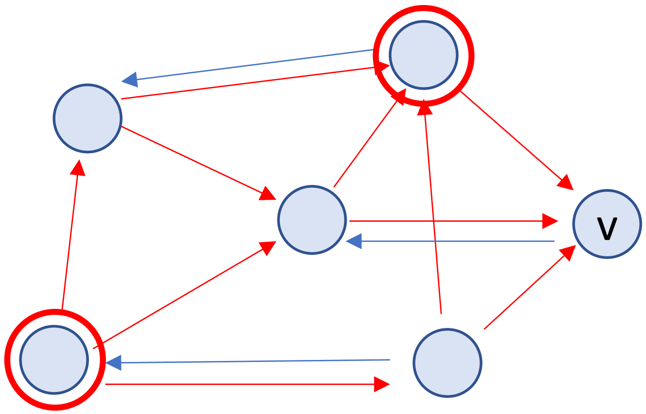
$$P(v) = q\sum_{\{u:u\in\{x,y,z\}\}} P(u)\cdot w(u) - (1-q)\sum_{\{u:u\in\{x,y,z\}\}} P(u)\cdot w(u)$$
$$= (2q-1)\sum_{\{u:u\in\{x,y,z\}\}} P(u)\cdot w(u)$$

We can get all the P in a topological order and in linear time!

From a expectation view:

$$\mathrm{Inf}(S) = q \sum_{\{v:\, v \in V\}} P(v)$$

So, we can construct a DAG for each node and compute each P(v) in linear time. Then, we can get the Inf(S) without any simulation!



DAG construction for v

*The actual process is complicated. And a trick is used for updating inf(S)

LDAG-N

Algorithm 2 LDAG-N without trick

$S = \emptyset$
for v in V **do**
    D(v) = FIND_DAG(G, v, θ)
    Inf(v) = $\sum_{u:v \in DAG(u)}$ DAG_INF(D(u), $\{v\}$)
end for
for i=1 to k **do**
    $u = argmax_u Inf(v)$
    $S = S \cup u$
    for $v : u \in D(v)$ **do**
        Inf(v) = $\sum_{u:v \in DAG(u)}$DAG_INF(D(u), $S + \{v\}$)
    end for
end for
return S

Evolutionary algorithm

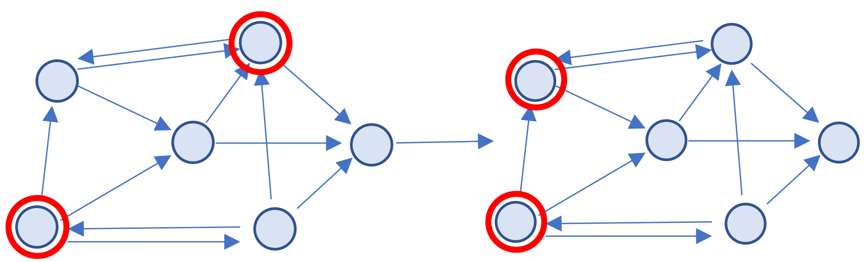
Evolutionary algorithm is a powerful tool for discrete optimization in large search space which is inspired by Evolution Theory. However, it is often ignored by computer scientists.

$$
\begin{tikzcd}
 & \text{Reproduce (Crossover and mutation)} \arrow[dr] & \\
\text{Selection} \arrow[ur] & & \text{Evaluate} \arrow[ll]
\end{tikzcd}
$$

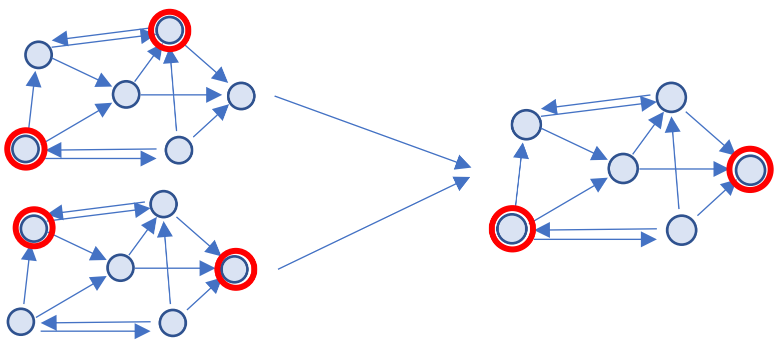
The flow of EA

EA is the only algorithm that considers the interaction within S

Evolutionary algorithm



A example of mutation

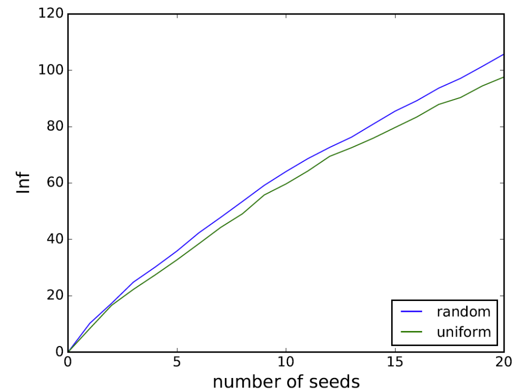


A example of crossover

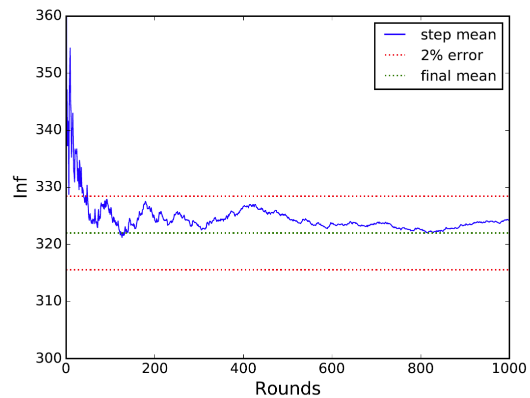
EA is the only algorithm that considers the interaction within S

Experiment setting

- Main dataset: ca-GrQc (5424 nodes and 14496 undirected edges)
- Simulation rounds: 200 # have a strong influence on the performance of greedy algorithm, 10000+ was recommended
- Way of weight generation: uniform for every in-edge
- q: 0.9 # something more about negative opinion is in report
- Threshold: 1/640
- CPU: Intel core i7-4790k

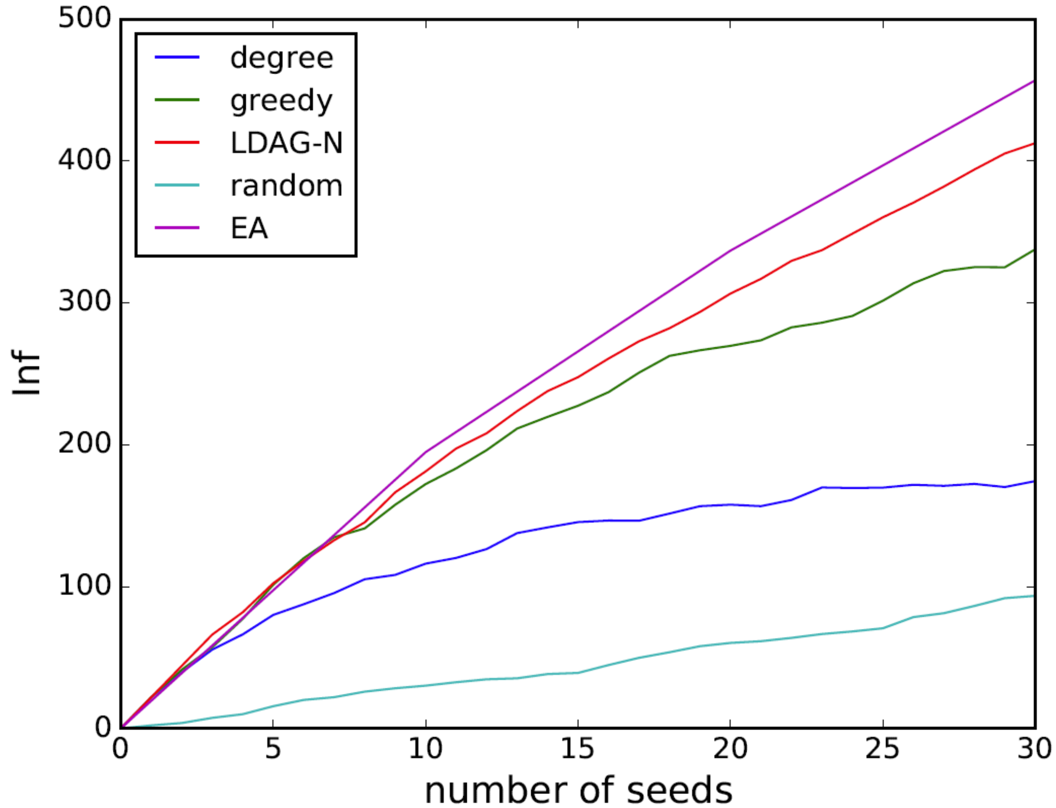


Test for random and uniform weight



Means of simulations with different times

Performance



Performance of the 5 algorithms *

EA always keeps the best

Greedy is harmed by insufficient simulations

LDAG-N works well

Degree and random perform poorly

*EA is only test for k=10,20,30

Time

Algorithm	Time/min	Relative Time
Greedy	2246.3	1
EA	148.6	0.066
LDAG-N	1.38	0.00062

Time for k=30

Greedy is unbearably slow. EA outperform greedy in both performance and running time. However, only LDAG-N is fast enough for much larger social networks.

Conclusion

In this project:

- A Linear threshold model with negative opinion for influence maximization is proposed, which has a strong connection with the real world advertising and capture the character of the mental activity of customers. This model keeps the important property of monotonicity and submodularity, which result in a 1-1/e approximation guarantee for greedy algorithm.

- Then, to tackle with the efficiency and quality problem, LDAG-N and EA are proposed or used in this problem. Both of them outperform greedy algorithm in speed, EA achieve the best performance and LDAG-N is scalable to larger dataset.

Thanks