中国科学技术大学
University of Science and Technology of China

Minimum Cost Seed Selection for Multiple Influences Diffusion in Communities

Guoju Gao[1], **Mingjun Xiao**[*1], **Jie Wu**[2], **He Huang**[3], **Guoliang Chen**[1]

[1]School of Computer Science and Technology
/ Suzhou Institute for Advanced Study,
University of Science and Technology of China, China
[2]Temple University, USA
[3]Soochow University, Suzhou, China
*Correspondence to: xiaomj@ustc.edu.cn

Outline

- Motivation
- Problem
- Solution
- Simulation
- Conclusion

Motivation

The traditional influence maximization model including Independent Cascade (IC) and Linear Threshold (LT):

- one single influence;
- probability sum in LT;
- without community;
- without users' preferences.

Motivation

A special case

A company intends to select some users to promote its multiple products (called influences) in online social network consisting of many communities, in which each user has different preferences for each influence.

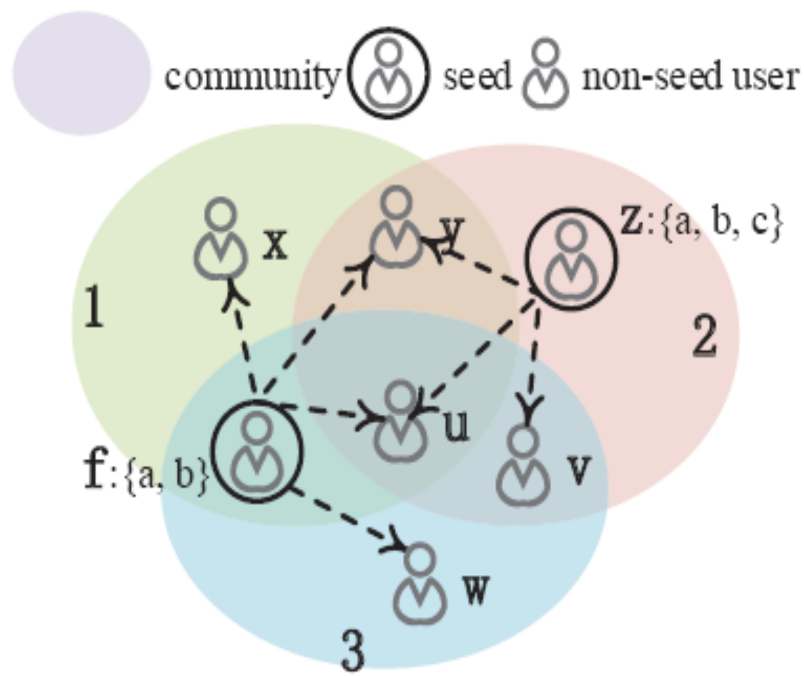
Objective

How to select some seeds with minimum cost so that the average influenced probability of all users in each community is not less than a threshold.

Motivation

A new influence maximization model, that is, multiple influences diffusion in communities:

- multiple influences;
- multiple communities;
- users' different preferences for different influences;
- mutual interferences of multiple influences



Problem

Online social network model: a five tuple $\langle N, E, W, M, C \rangle$.

$N \rightarrow$ social users; $E \rightarrow$ directed edges (the relationships among users);
$W \rightarrow$ edge weights; $M \rightarrow$ total communities;
$C \rightarrow$ recruitment costs for all social users.

User interest model: keywords + interest profile.

$A \rightarrow$ all influences; $K \rightarrow$ all keywords; $K_a \rightarrow$ influence a's keywords;
$D_i \rightarrow$ interest profile of user i; $w_i^a \rightarrow$ interest probability.

Multiple influences diffusion model: mutual interferences + influence probability + joint influence probability + average influenced probability.

Problem

We aim to select a seed set with minimum cost, so as to ensure that the average influenced probability of all users in each community is not less than a threshold.

$$\begin{aligned} &Minimize \quad && C(\mathcal{S}) = \textstyle\sum_{i\in\mathcal{S}} c_i \\ &Subject\ to && \mathcal{S} \subseteq \mathcal{N} \\ & && AIP_j^a(\mathcal{S}) \geq \eta_a \\ & && j\in\mathcal{M}, a\in\mathcal{A} \end{aligned}$$

*an NP-hard problem

Problem-extended

We focus on the minimum cost seed selection, where the number of acceptable influences for a user is limited and heterogeneous, and the cost is proportional to the number of allocated influences.

$$\begin{array}{ll} Minimize & C(\mathcal{S}_\dagger) = \sum_{(i:\mathcal{A}_i)\in\mathcal{S}_\dagger} \beta_i |\mathcal{A}_i| \\ Subject\ to & \mathcal{S}_\dagger \subseteq \mathbf{\Omega} \\ & |\mathcal{A}_i| \leq \delta_i, (i, \mathcal{A}_i) \in \mathcal{S}_\dagger \\ & AIP_j^a(\mathcal{S}_\dagger) \geq \eta_a, j \in \mathcal{M}, a \in \mathcal{A} \end{array}$$

*also an NP-hard problem

Solution

- 1) define a utility function, i.e., the utility of the actual average influenced probability of all users in each community via the seed set;
- 2) turn the minimum cost seed selection problem into a minimum submodular cover with submodular cost problem;
- 3) design a greedy selection strategy, i.e., greedily select the user who has the maximum marginal contribution per cost.

Solution

The greedy algorithm called *G-MCSS* is shown as follows.

Algorithm 1 *The G-MCSS Algorithm*

Require: $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}, \mathcal{W}, \mathcal{M}, \mathcal{C} \rangle, \mathcal{N}_j, \mathcal{K}, v_k, \mathcal{D}_i, \mathcal{A}, \eta_a, a \in \mathcal{A}$

Ensure: Seed set $\mathcal{S}$ and total cost $C(\mathcal{S})$

Initialize $\mathcal{S} = \phi$;

while $U(\mathcal{S}) < M \sum_{a=1}^{A} \eta_a$ **do**

  Select a user $i \in \mathcal{N} \backslash \mathcal{S}$ to maximize $\frac{U_i(\mathcal{S})}{c_i}$;

  $\mathcal{S} = \mathcal{S} \cup \{i\}$;

$C(\mathcal{S}) = \sum_{i \in \mathcal{S}} c_i$;

return $\mathcal{S}$, $C(\mathcal{S})$

*Although G-MCSS looks similar to traditional set cover approximation algorithms, it is intrinsically different from them. The computation complexity of G-MCSS is $O(N^3)$.

Solution

We give the performance analysis on G-MCSS.

Theorem 3: The proposed G-MCSS algorithm achieves the $(1+\ln\frac{\varphi M\sum_{a=1}^{A}\eta_a}{opt})$ approximation of the optimal cost, where opt is the cost of the optimal solution to the MCSS problem.

*φ=max{φ1, φ2}

$$\begin{cases} \varphi_1 = \frac{\sum_{i=1}^{N} c_i}{M\sum_{a=1}^{A}\eta_a}, \\ \varphi_2 = \max\{\frac{c_i \mid 1 \leq i \leq N}{\eta_a - AIP_j^a(\mathcal{S}) \mid AIP_j^a(\mathcal{S}) < \eta_a, \mathcal{S} \subset \mathcal{N}, j \in \mathcal{M}, a \in \mathcal{A}}\}. \end{cases}$$

Solution-extended

- 1) define a new marginal utility function for a same user with an influence set;

- 2) design a greedy selection strategy, i.e., greedily select a user and an influence for this user with maximum margin utility function per cost.

*The computation complexity of E-MCSS is still O(N^3).

Algorithm 2 The E-MCSS Algorithm

Require: $\mathcal{G} = \langle \mathcal{N}, \mathcal{E}, \mathcal{W}, \mathcal{M}, \mathcal{C} \rangle, \mathcal{N}_j, \mathcal{K}, v_k, D_i^k, \mathcal{A}, \eta_a$ for $a \in \mathcal{A}$, δ_i, β_i for $i \in \mathcal{N}$

Ensure: Seed set $\mathcal{S}_\dagger$ and cost $C(\mathcal{S}_\dagger)$

Initialize $\mathcal{S}_\dagger = \phi$;
while $U(\mathcal{S}_\dagger) < M \sum_{a=1}^{A} \eta_a$ **do**
  for $i \in \mathcal{N}$ **do**
    if $(i, \mathcal{A}_i) \in \mathcal{S}_\dagger$ and $|\mathcal{A}_i| = \delta_i$ **then**
      Skip i and remove $(i, \mathcal{A}_i)$ from $\mathbf{\Omega}$;
    else
      Select an influence $a \in \mathcal{A} \setminus \mathcal{A}_i$ to maximize $\frac{\Delta_{\{a\}} U_i(\mathcal{S}_\dagger)}{\beta_i}$, denoted as $\Delta[i][a]$;
  Select $(i^*, \{a^*\})$ with maximum value $\Delta[i][a]$;
  $\mathcal{S}_\dagger = \mathcal{S}_\dagger \cup \{(i^*, \{a^*\})\}$;
  Clear $\Delta[i][a]$;
$C(\mathcal{S}_\dagger) = \sum_{(i, \mathcal{A}_i) \in \mathcal{S}_\dagger} \beta_i |\mathcal{A}_i|$;
return $\mathcal{S}_\dagger, C(\mathcal{S}_\dagger)$

Simulation

Algorithms in Comparison

- **MaxDegree --** select seeds with the highest out-degree per cost in each iteration.
- **MaxBreadth --** select the seeds which can cover maximum communities in each iteration.

Evaluation Metric

- Total Recruitment Cost

Simulation

Traces Used

TABLE II
STATISTICS OF INVESTIGATED DATASETS.

Dataset	*WikiVote*	*BlogCatalog*	*Douban*
Nodes	7,115	10,312	154,907
Edges	103,689	333,983	654,188
Avg. OD	14.6	32.4	4.2

- J. Leskovec and A. Krevl. SNAP Datasets: Wiki-vote social network. http://snap.stanford.edu/data, 2014.
- R. Zafarani and H. Liu. Social computing data repository at ASU. http://socialcomputing.asu.edu, 2009.

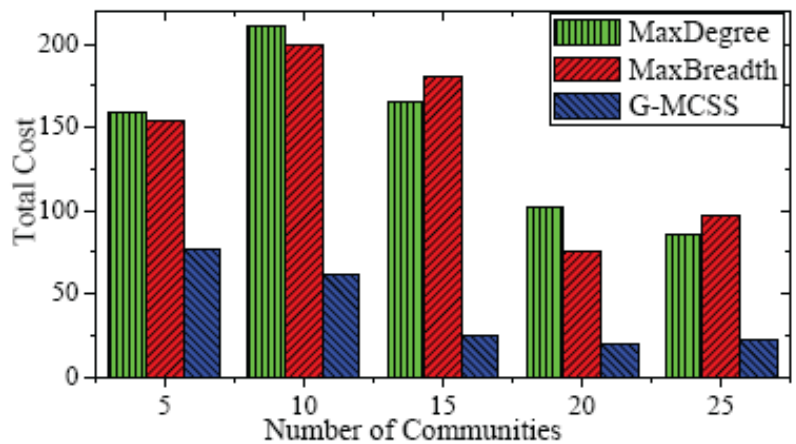
Simulation

Settings

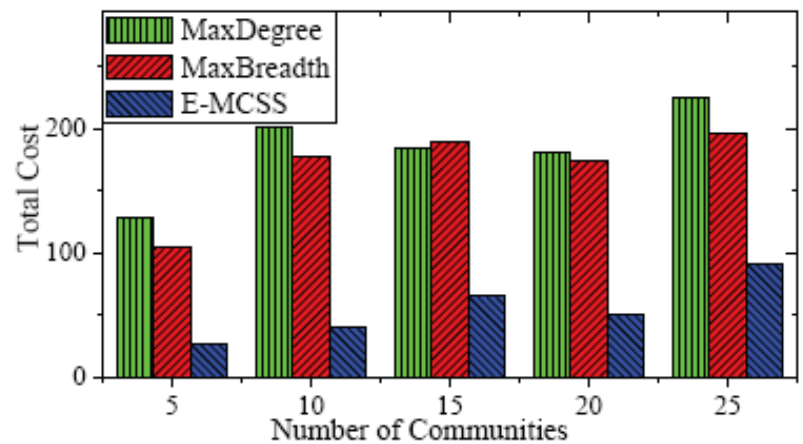
Parameter name	Default value	Range
Number of communities: M	20	5-100
Number of users in each community	50	10-100
Number of influences: A	15	5-25
Parameter: η	0.4	0.2-0.6

Simulation

Simulation Results on the real dataset *WikiVote* with different numbers of communities



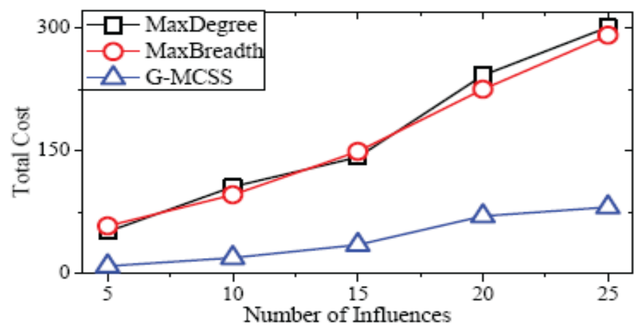
(a) G-MCSS vs. M



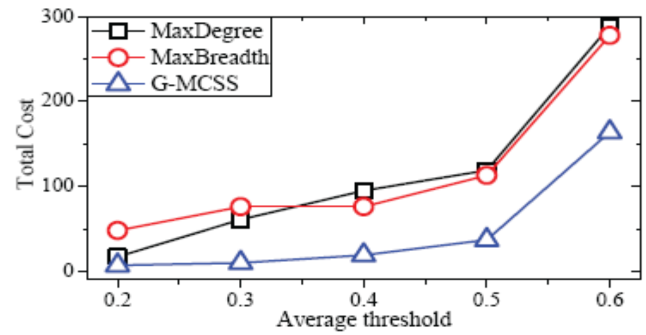
(b) E-MCSS vs. M

Simulation

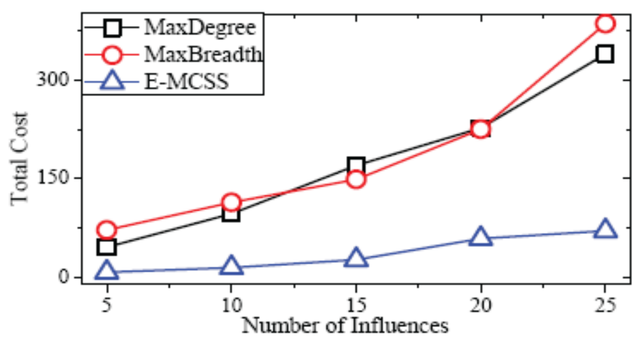
Simulation Results *on WikiVote:* A and η



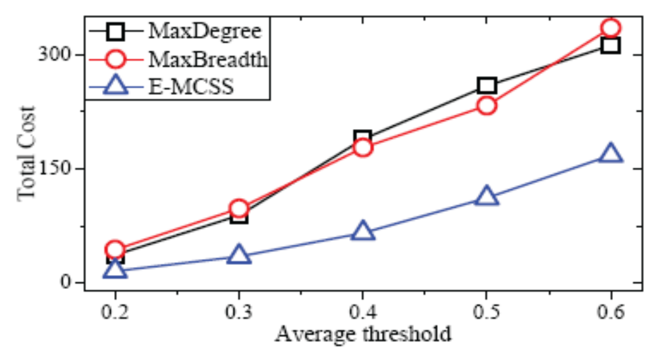
(a) G-MCSS: cost vs. A



(b) G-MCSS: cost vs. η



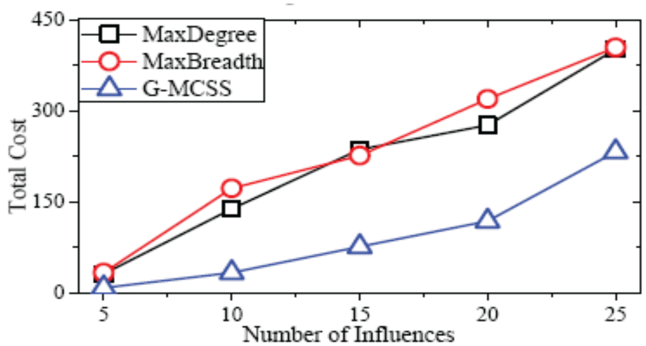
(c) E-MCSS: cost vs. A



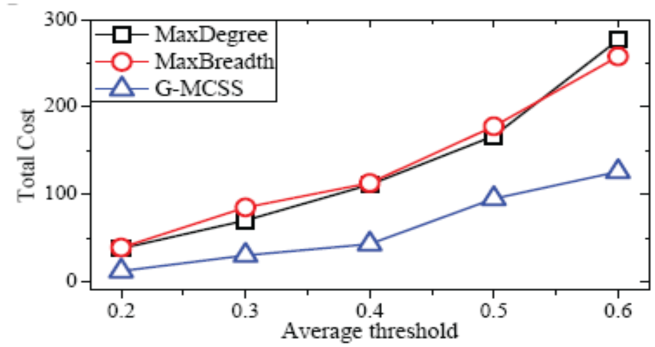
(d) E-MCSS: cost vs. η

Simulation

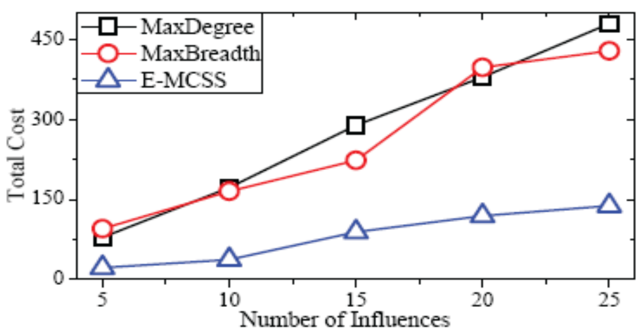
Simulation Results *on Blogcatalog:* A and η



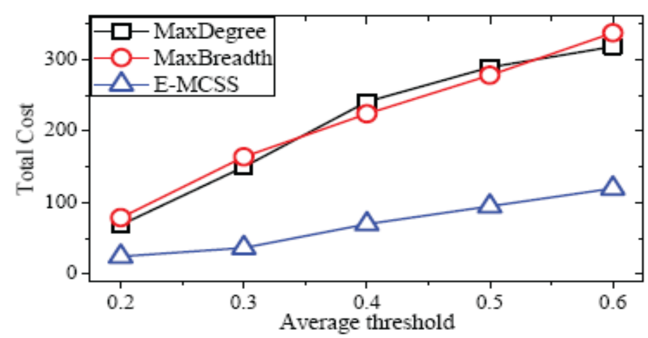
(a) G-MCSS: cost vs. A



(b) G-MCSS: cost vs. η



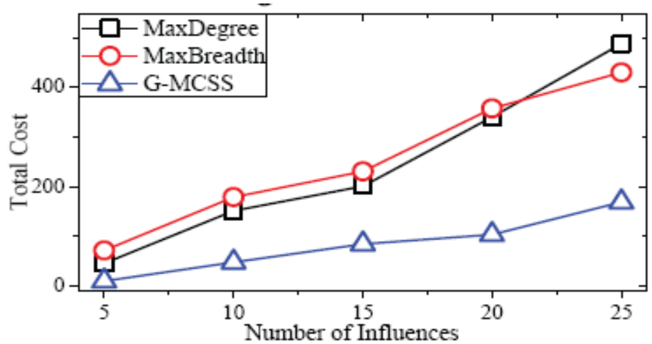
(c) E-MCSS: cost vs. A



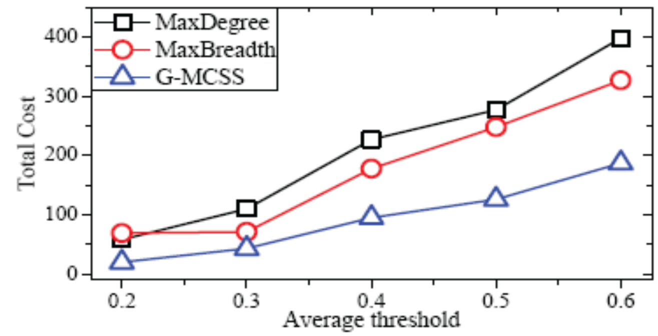
(d) E-MCSS: cost vs. η

Simulation

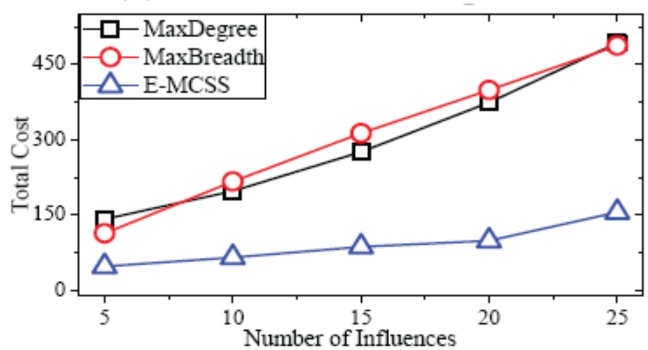
Simulation Results *on Douban:* A and η



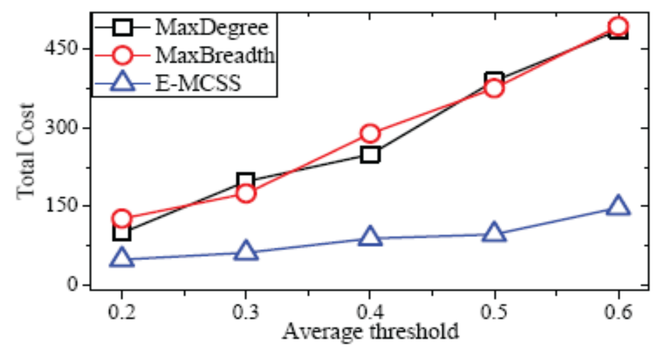
(a) G-MCSS: cost vs. A



(b) G-MCSS: cost vs. η



(c) E-MCSS: cost vs. A



(d) E-MCSS: cost vs. η

Conclusion

- In all simulations, our G-MCSS and E-MCSS algorithms always outperform the two compared algorithms based on the real datasets WikiVote, Blogcatalog, and Douban.

- G-MCSS and E-MCSS have about 64% and 70% smaller total costs than the compared algorithms, respectively.

Thank You!

Q&A