Individual Influence Maximization via Link Recommendation

Guowei Ma[1], Qi Liu[1], Enhong Chen[1], and Biao Xiang[2]

1 University of Science and Technology of China
gwma@mail.ustc.edu.cn, {qiliuql,cheneh}@ustc.edu.cn
2 Search Technology Center Asia (STCA), Microsoft
bixian@microsoft.com

Outline

- Motivation
  - Related Work
- Problem Formulation
- Solutions
- Experiments
- Conclusion

Motivation

- Social Networks
  - New resources and platforms
  - Important and fundamental role for the spread of information

- Exploiting social influence
  - Viral Marketing
  - ……

Motivation

- Individual Influence
  - One person or company want to improve her influence
    - Make new friends
    - Cooperate with other companies
  - One country want to improve her politic influence
    - Establish new diplomatic relations

- Research Problem:

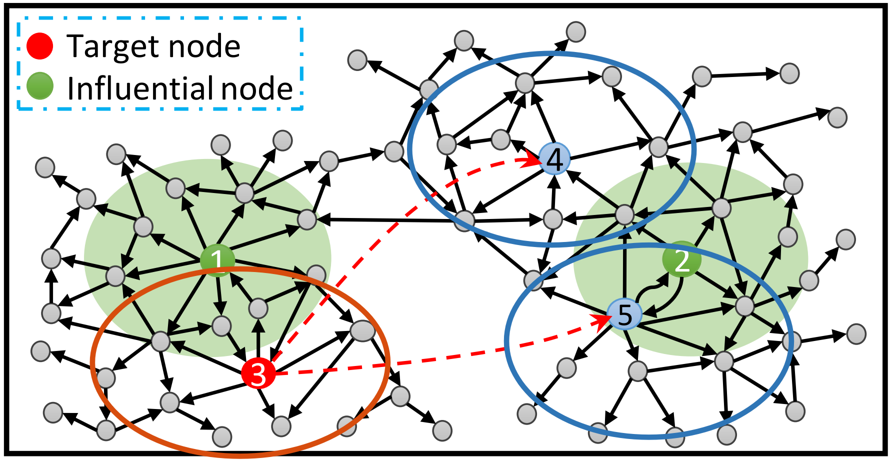
  Individual Influence Maximization via Link Recommendation

Motivation

- Research Problem:

  Individual Influence Maximization via Link Recommendation

- Challenge
  - Influence overlap
  - Computational efficiency



Problem Formulation

- Problem Formulation

$$\arg\max_{\mathbf{S}} \ \{ f_{t\to V}^{\mathbf{S}} - f_{t\to V} \} \text{, subject to } c(\mathbf{S}) \le B$$

Problem Formulation

- Problem Formulation

$$\arg\max_{\mathbf{S}} \ \{ f_{t\to V}^{\mathbf{S}} - f_{t\to V} \} \text{ , subject to } c(\mathbf{S}) \le B$$

$$\arg\max_{\mathbf{S}} \mathcal{F}(\mathbf{S}) = \{ f_{t\to V}^{\mathbf{S}} - f_{t\to V} \}, \quad \text{s.t., } |\mathbf{S}| \le \mathrm{K}.$$

Problem Formulation

- Problem Formulation

$$\arg\max_{\mathbf{S}} \ \{ f_{t\to V}^{\mathbf{S}} - f_{t\to V} \} \text{ , subject to } c(\mathbf{S}) \leq B$$

$$\arg\max_{\mathbf{S}} \mathcal{F}(\mathbf{S}) = \{ f_{t\to V}^{\mathbf{S}} - f_{t\to V} \}, \quad \text{s.t., } |\mathbf{S}| \leq \mathrm{K}.$$

- Objective Function Definition

$$\begin{aligned}
\mathcal{F}(\mathbf{S}) &= f_{t\to V}^{\mathbf{S}} - f_{t\to V} = \sum_{c\in\mathbf{S}} \mathcal{F}(\{c\}) \\
&= \sum_{c\in\mathbf{S}} \lambda_c (1 - f_{t\to c}) * \sum_{i=1}^{n} (f_{c\to i} * [1 - f_{t\to i}]).
\end{aligned}$$

Problem Formulation

- Objective Function Definition

$$
\begin{aligned}
\mathcal{F}(\mathbf{S}) &= f^{\mathbf{S}}_{t\to V} - f_{t\to V} = \sum_{c\in\mathbf{S}} \mathcal{F}(\{c\}) \\
&= \sum_{c\in\mathbf{S}} \lambda_c(1 - f_{t\to c}) * \sum_{i=1}^{n} (f_{c\to i} * [1 - f_{t\to i}]).
\end{aligned}
$$

- Properties of the Objective Function

1. $\mathcal{F}(\emptyset) = 0$,
2. $\mathcal{F}(\mathbf{S})$ is nonnegative and monotonically increasing.
3. $\mathcal{F}(\mathbf{S})$ is *submodular*.

Solutions

- Greedy Strategy
  - *greedy* algorithm
    - Solution>= (1-1/e)*OPT
    - O_T (n * K)

Algorithm 1: Greedy Algorithm for IM

Input: $G(V, E)$, K
Output: $\mathbf{S}$ with K nodes

initialize $\mathbf{S} = \emptyset$;
while $|\mathbf{S}| < K$ **do**
  select $s = \arg\max_{x \in V \setminus \mathbf{S}} \ f(\mathbf{S} \cup x) - f(\mathbf{S})$;
  $\mathbf{S} = \mathbf{S} \cup s$;
Return $\mathbf{S}$;

- Greedy Strategy
  - *greedy* algorithm
    - Solution>= (1-1/e)*OPT
    - $O_T(n * K)$

Algorithm 1: Greedy Algorithm for IM

Input: $G(V, E)$, K
Output: $\mathbf{S}$ with K nodes
initialize $\mathbf{S} = \emptyset$;
while $|\mathbf{S}| < K$ **do**
  select $s = \arg\max_{x \in V \setminus \mathbf{S}} f(\mathbf{S} \cup x) - f(\mathbf{S})$;
  $\mathbf{S} = \mathbf{S} \cup s$;
Return $\mathbf{S}$;

  - *lazy* algorithm
    - Solution>= (1-1/e)*OPT
    - $O_T(n + \theta * K)$

Input: $G(V, E, T)$, a given target node t, K
Output: $\mathbf{S}$ with K nodes
initialize $\mathbf{S} = \emptyset$;
for *each node* $i \in V$ **do**
  calculate $\mathcal{F}(\{i\}) = f_{t \to V}^{\{i\}} - f_{t \to V}$;
  $flag_i = |\mathbf{S}|$; // here, $|\mathbf{S}| = 0$
  // $flag_i$ indicates that $\mathcal{F}(\{i\})$ is
  // calculated in the $|\mathbf{S}|$ iteration
while $|\mathbf{S}| < K$ **do**
  s = Find the greatest $\mathcal{F}(\{s\})$ in $\mathcal{F}$;
  if $flag_s == |\mathbf{S}|$ **then**
    $\mathbf{S} = \mathbf{S} \cup s$;
    $\mathcal{F}(s) = 0$;
  else
    recalculate $\mathcal{F}(s) = f_{t \to V}^{\mathbf{S} \cup s} - f_{t \to V}^{\mathbf{S}}$;
    $flag_s = |\mathbf{S}|$;
Return $\mathbf{S}$;

Solutions

General Influence Propagation Models

- Greedy Strategy
  - greedy alg.
    - Solution >= (1-1/e)*OPT
    - O_T (n*K)
  - lazy alg.
    - Solution >= (1-1/e)*OPT
    - O_T (n + θ*K)

Linear Model

- Upper Bound algorithm (uBound)
  - Solution >= (1-1/e)*OPT
  - O_T (1 + η*K)

Input: $G(V, E, T)$, a given target node t, K
Output: **S** with K nodes
initialize $\mathbf{S} = \emptyset$;
Compute the upper bound vector $\mathbb{U} = diag(\lambda_1, \lambda_2, ..., \lambda_n) \cdot diag(\alpha_1, \alpha_2, ..., \alpha_n) \cdot (I - DT)^{-1} \cdot \mathbf{e}$ in $O(|E|)$ time;
for *each node* $i \in V$ **do**
  $\mathcal{F}(i) = \mathbb{U}_i$;
  $flag_i = 0$; // here, $|\mathbf{S}| = 0$
while $|\mathbf{S}| < K$ **do**
  s = Find the greatest $\mathcal{F}(s)$ in $\mathcal{F}$;
  if $flag_s == |\mathbf{S}|$ **then**
    $\mathbf{S} = \mathbf{S} \cup s$;
    $\mathcal{F}(s) = 0$;
  else
    recalculate $\mathcal{F}(s) = f_{t \to V}^{\mathbf{S} \cup s} - f_{t \to V}^{\mathbf{S}}$;
    $flag_s = |\mathbf{S}|$;
Return S;

Experiments

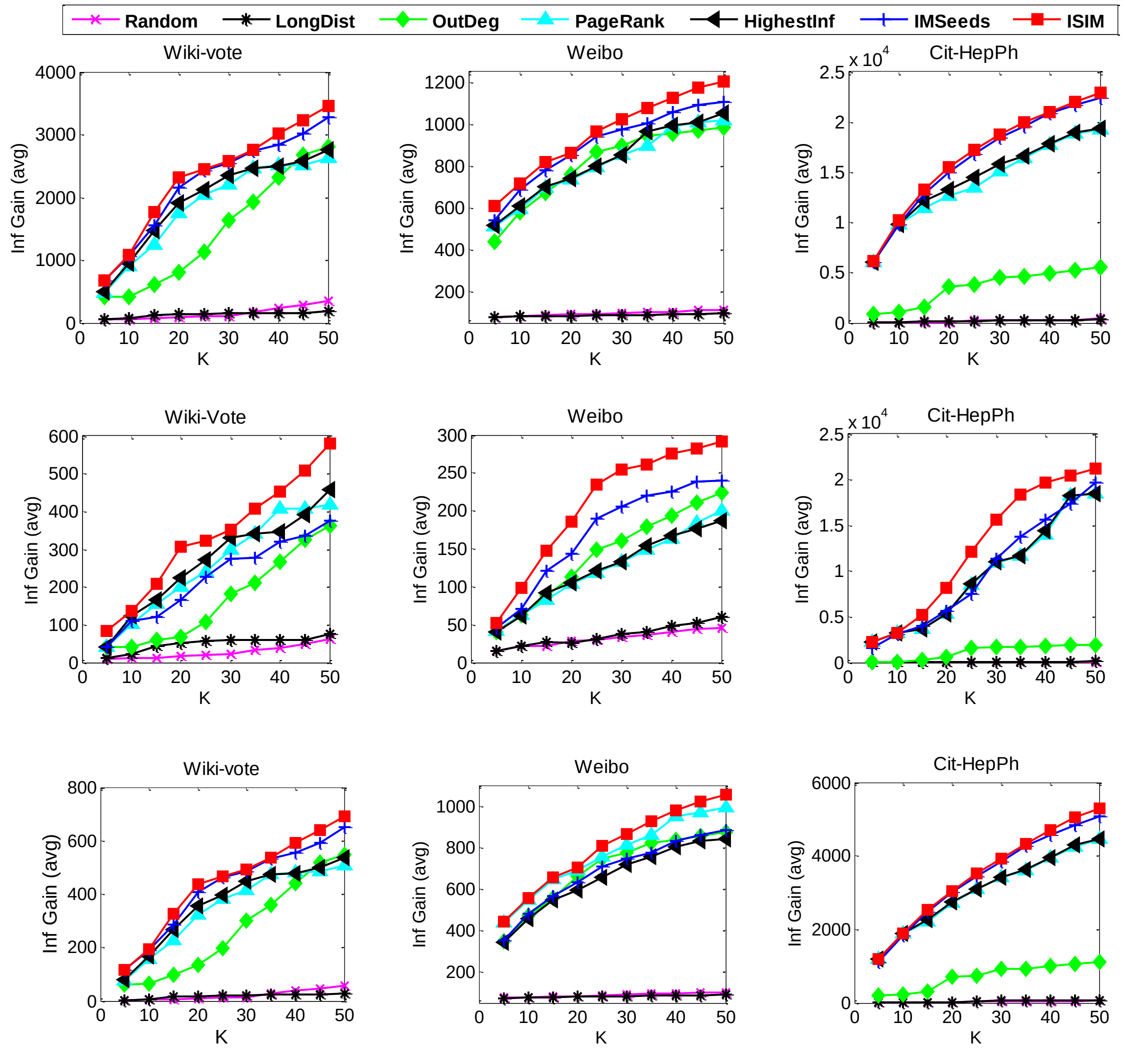
- Experiments
  - Real Influence Gain Comparison
  - Time Complexity Analysis
  - Case Study

Table 1. Experimental Datasets

Name	Wiki-Vote	Weibo	cit-HepPh	Twitter
Nodes	7,115	7,378	34,546	11,316,811
Edges	103,689	39,759	421,578	85,331,845

Real Influence Gain Comparison

- Our method:
  - ISIM
- Bechmarks:
  - Random
  - OutDeg
  - LongDist
  - PageRank
  - HighestInf
  - IMSeeds



Time Complexity Analysis

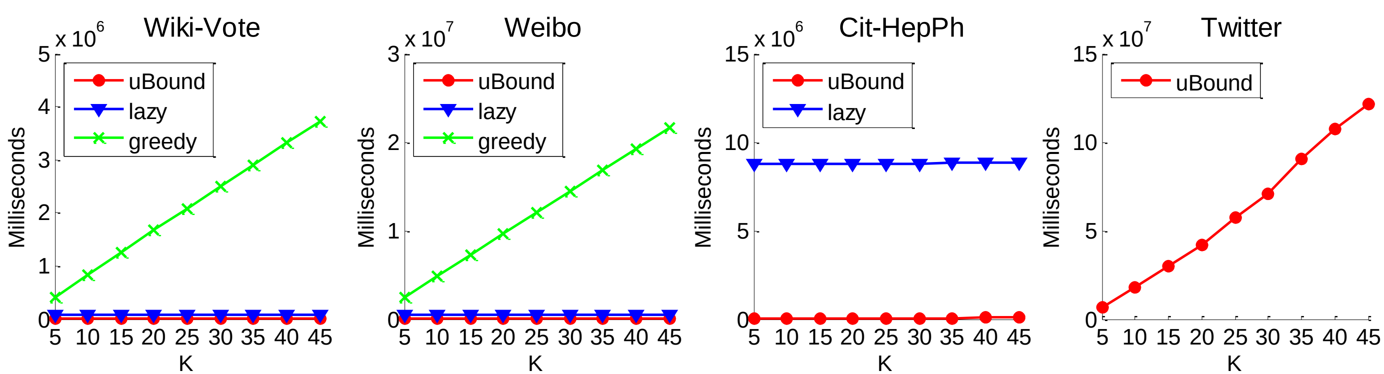
- On Linear Model
  - ➢ Greedy
  - ➢ Lazy
  - ➢ uBound

The number of influence spread estimations

Datasets	Alg. \ Top K	5	10	15	20	25	30	35	40	45
Wiki-Vote	*greedy*	35208	70390	105547	140679	175786	210868	245925	280957	315964
	lazy	7124	7139	7171	7195	7220	7250	7306	7331	7374
	uBound	**19**	**53**	**84**	**119**	**157**	**204**	**265**	**297**	**352**
Weibo	*greedy*	36836	73646	110431	147191	183926	220636	257321	293981	330616
	lazy	7414	7446	7505	7572	7638	7738	7838	7917	8047
	uBound	**50**	**98**	**180**	**278**	**356**	**494**	**622**	**726**	**868**
Cit-HepPh	*lazy*	34554	34566	34586	34608	34642	34660	34691	34710	34744
	uBound	**17**	**37**	**64**	**89**	**137**	**161**	**194**	**221**	**264**
Twitter	*uBound*	**16**	**43**	**71**	**99**	**136**	**166**	**212**	**253**	**286**

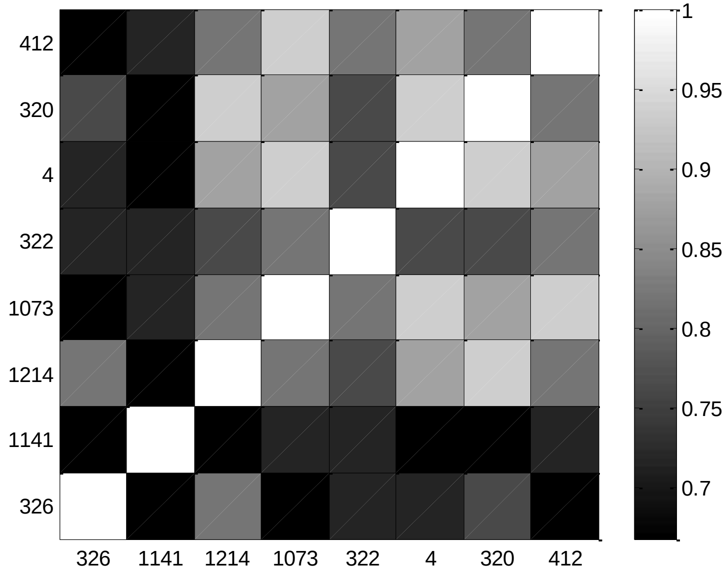
- Evaluation
  - ➢ # influence spread estimations
  - ➢ Running time

Real Runtime Comparison

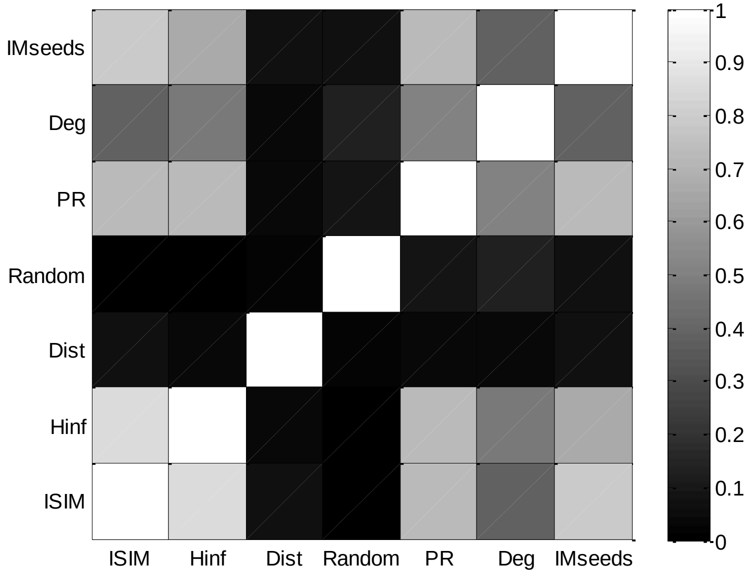


Case Study

- It's necessary to design individualized link recommendation algorithms.



(a) Different Target Nodes



(b) Different Algorithms

Conclusion

- Study the individual influence maximization problem

- Formulate the problem

  and design the objective function

- Algorithms
  - greedy, lazy and uBound

- Validate the performance of the proposed algorithms

Thank you!