Influence Maximization on Big Social Graphs

Challenges and Techniques

Ju Fan (范举)

fanj@ruc.edu.cn

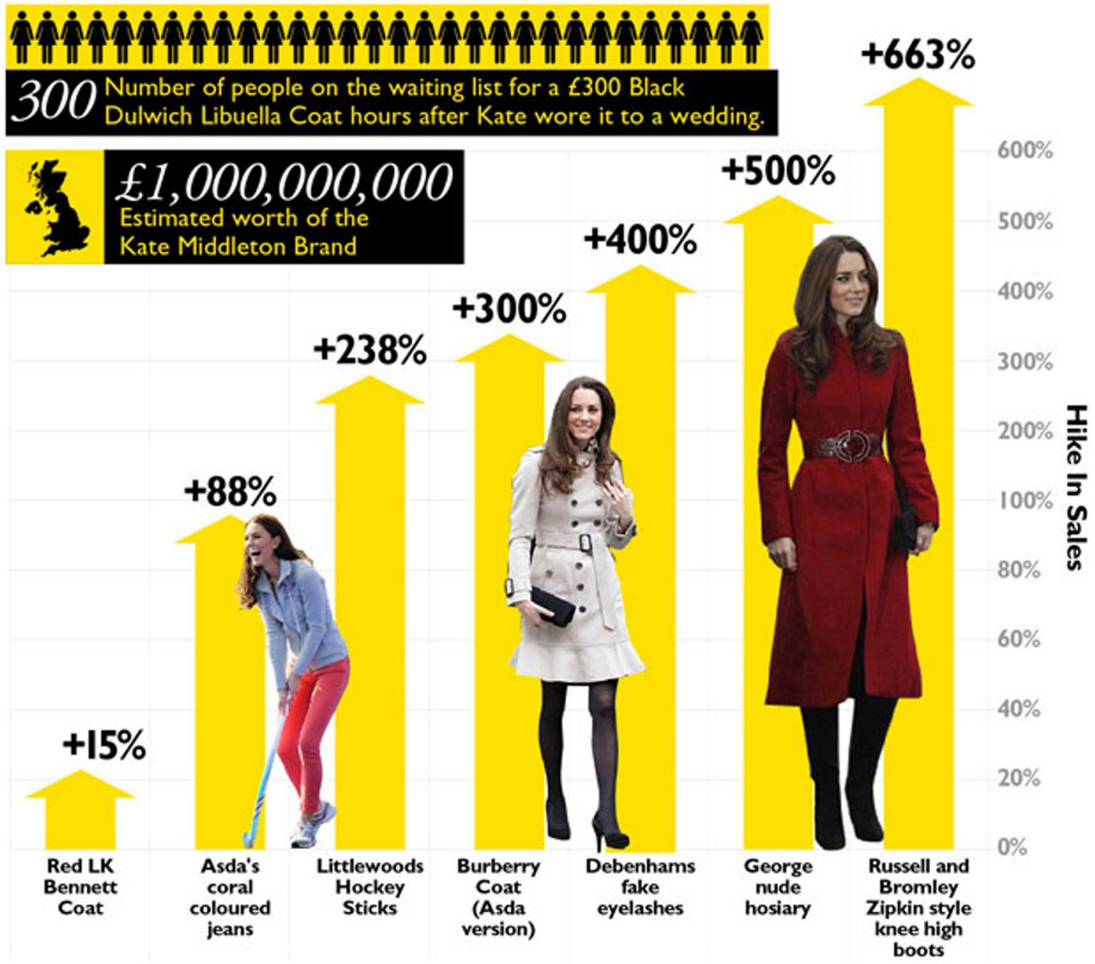
http://iir.ruc.edu.cn/~fanj/

中國人民大學
RENMIN UNIVERSITY OF CHINA

Talk Objectives

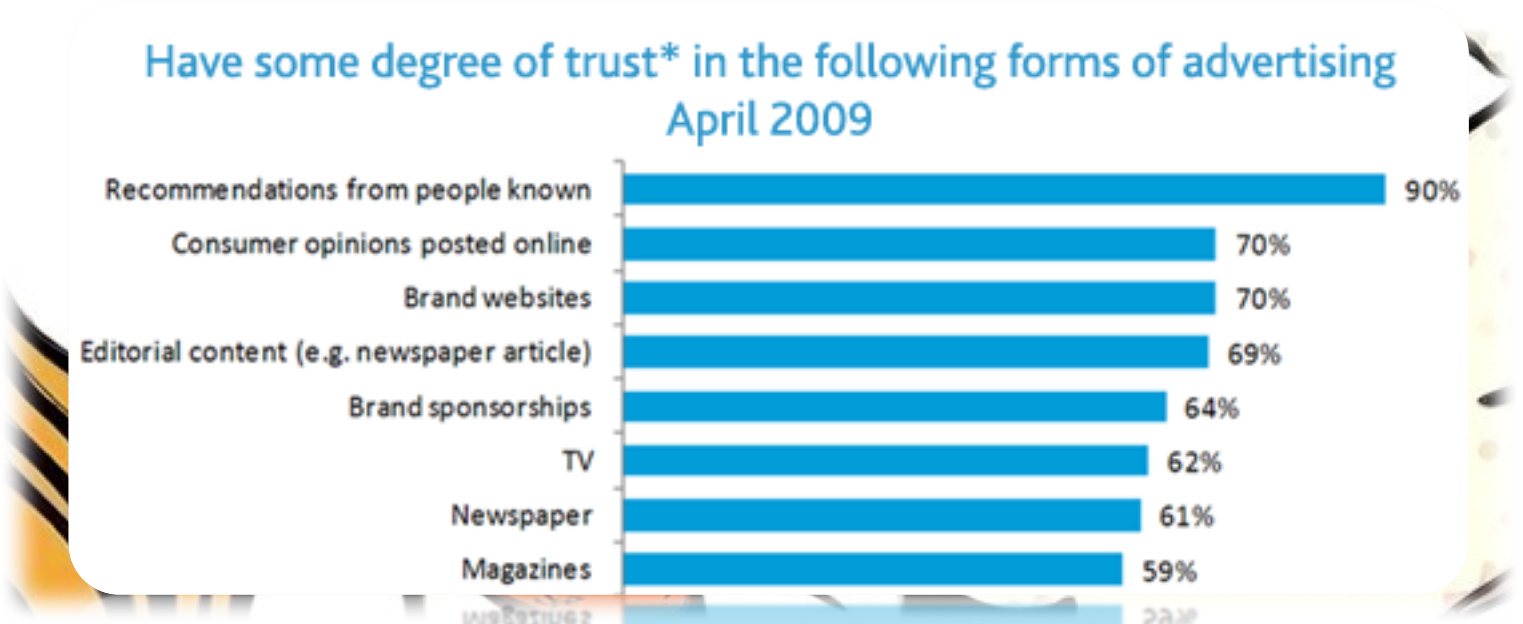
- What is Influence Maximization (IM) for social networks?
- How to formally model IM?
- How to solve IM on BIG social graphs?

Kate Middleton Effect

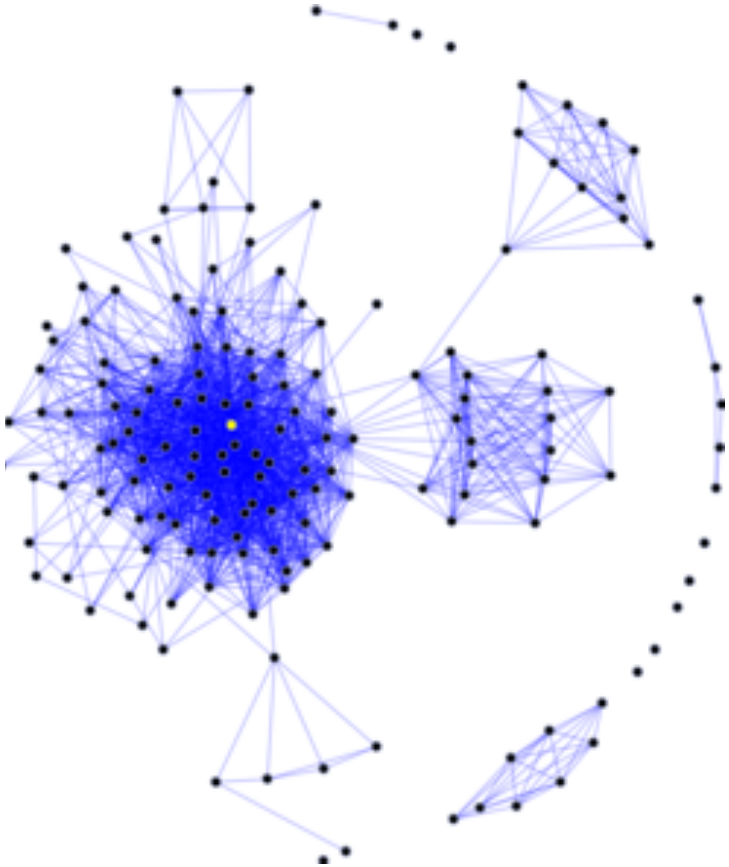


Social Influence

- Social influence occurs when a person's emotions, opinions, or behaviors are affected by others



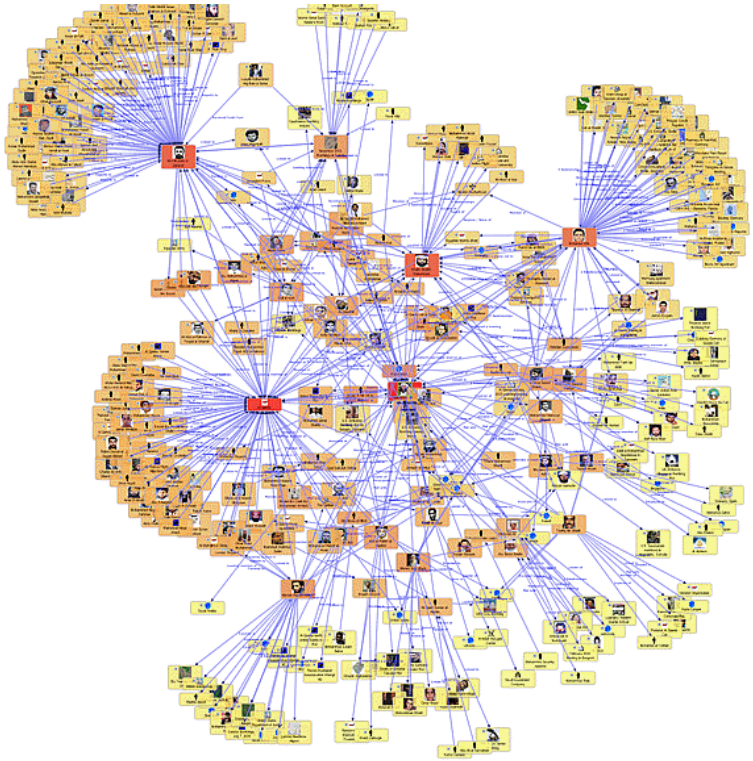
Booming of Online Social Networks



- How to benefit from online social networks?

Online Social Influence Analysis

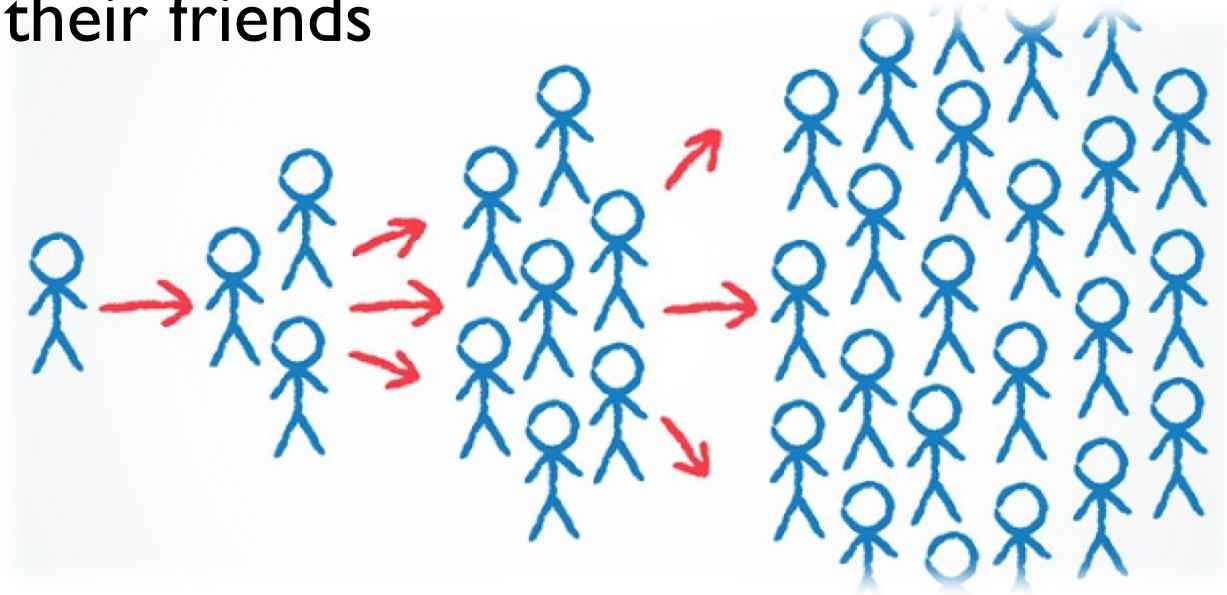
- Big Datasets
  - Real time, Dynamic



- New Services
  - Social Influence → $

Applications - Viral Marketing

- Identify influential customers, aka., seeds
- Convince them to adopt the product, e.g., by offering discount/free samples
- These customers endorse the product among their friends



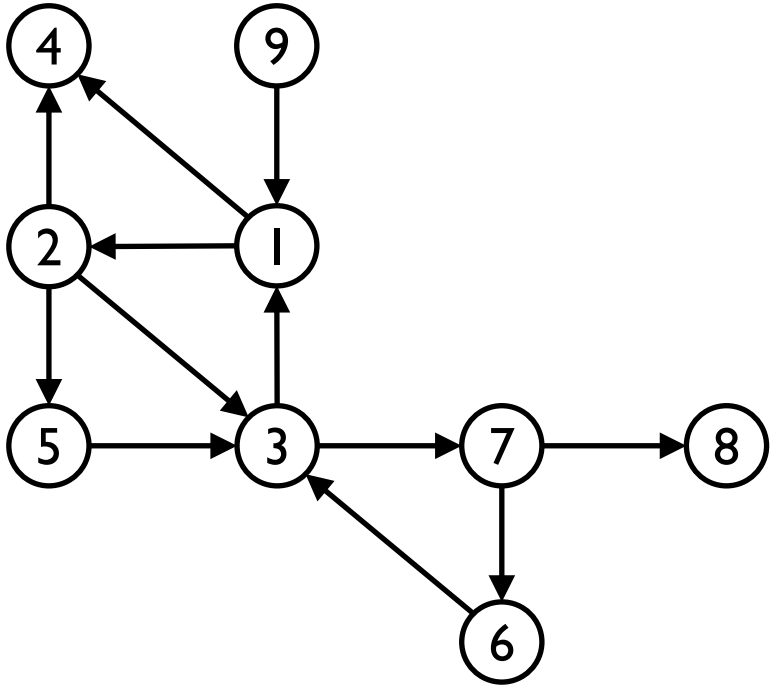
Other Applications

- Rumor Monitoring and Control
- Effective Competition
  - Minimize the influence of the competitor or negative influence
- Political Campaigns
- We Media, aka. Self-Media
  - Internet celebrity economy (网红经济)

Influence Maximization (IM)

- Data: a directed graph $G = (V, E)$
  - V denotes the set of vertices (users)
  - E denotes the set of edges, e.g., the followee/follower relationship on Twitter

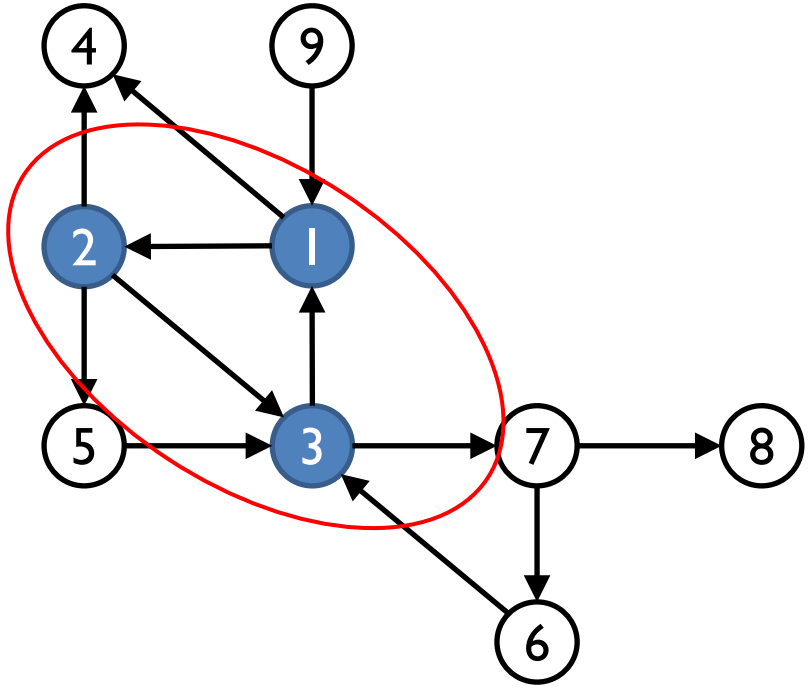
Input



Influence Maximization (IM)

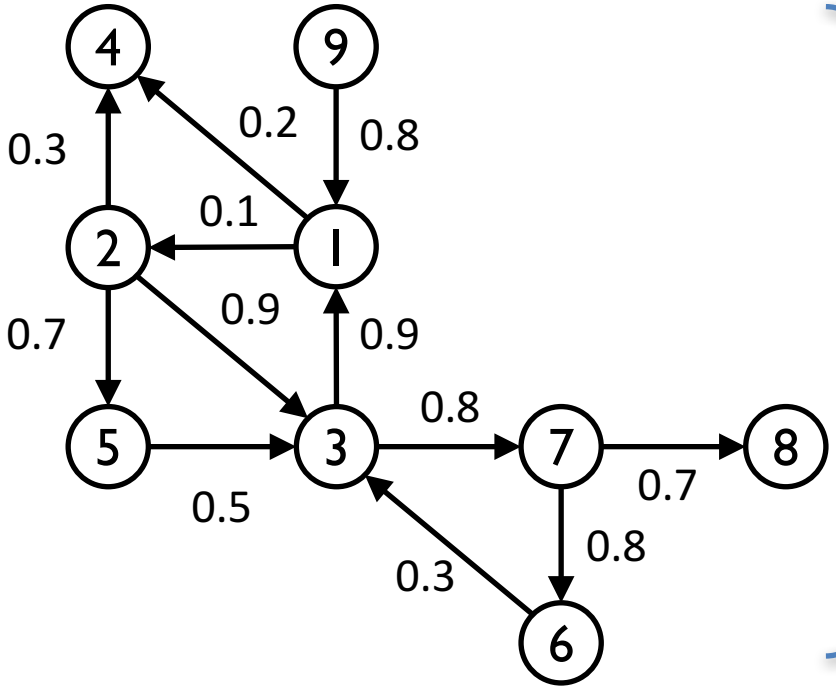
- Problem: selecting k users such that by activating them, the expected spread of influence is maximized.

Output



Motivating Example

- A company is to carry out a promotion campaign for their product, by sending free samples to initial users.

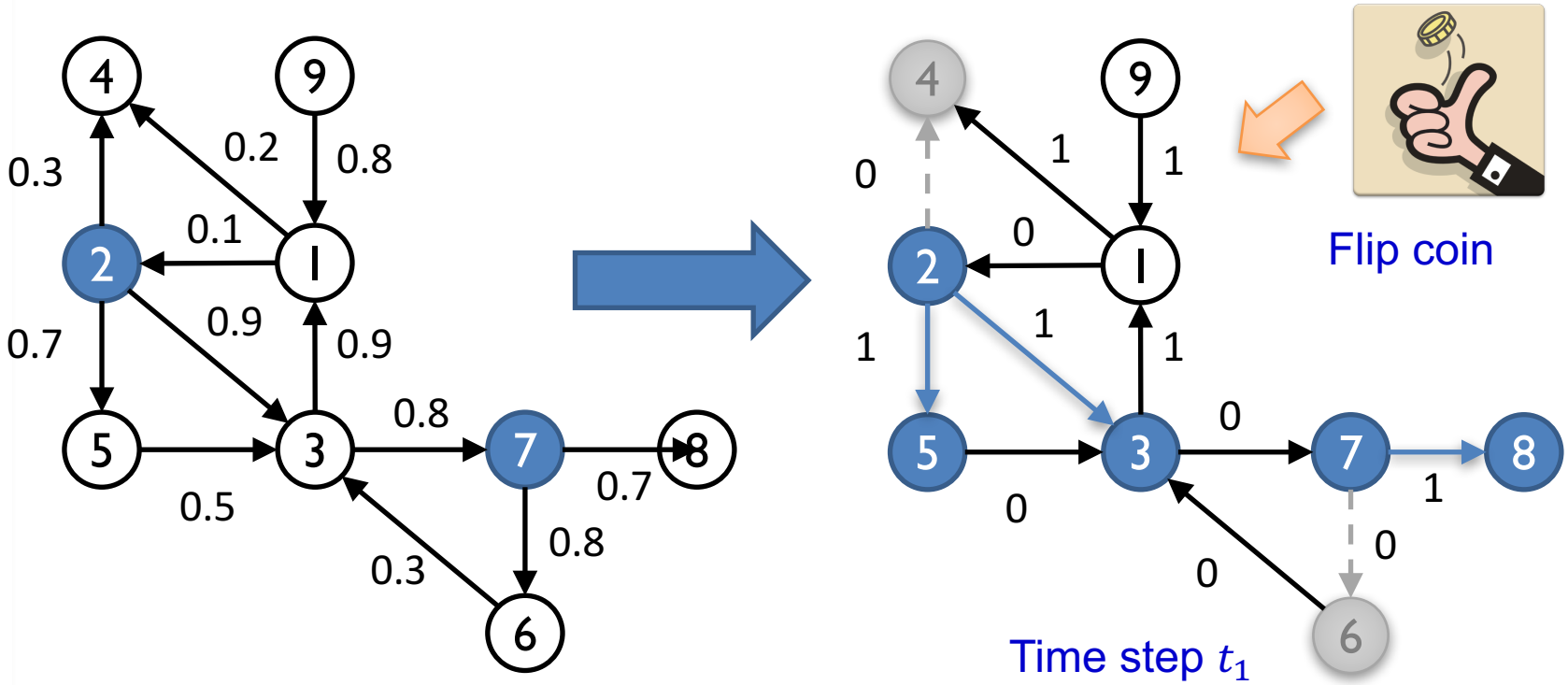


Independent Cascade (**IC**) Model

Influence Probabilities

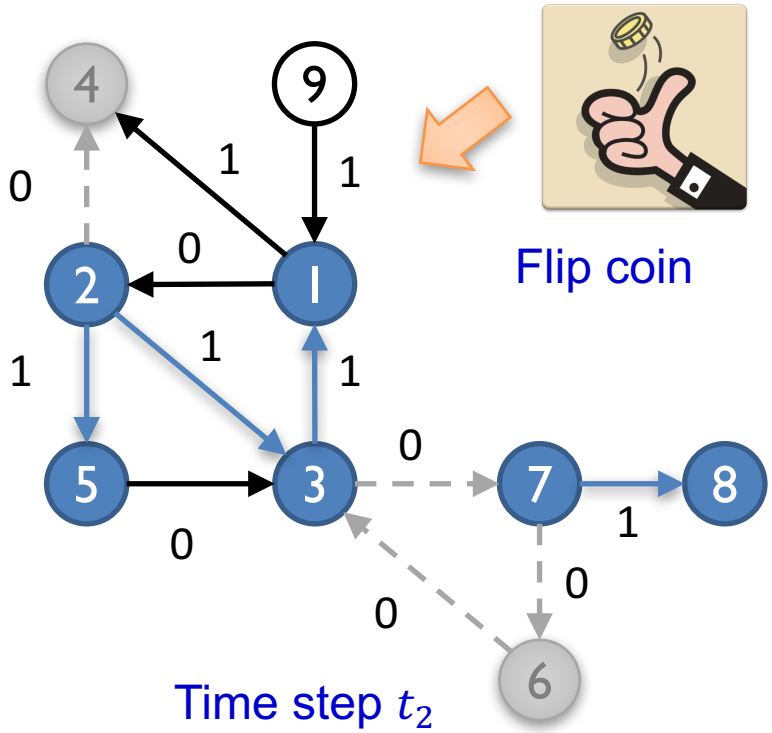
Motivating Example

- Suppose to choose u_2 and u_7 as seeds
- How do they activate their neighbors?



Motivating Example

- Continue to activate neighbors' neighbors



Flip coin

Until no more node is activated

- ✓ u_1
- ✓ u_2
- ✓ u_3
- ✓ u_5
- ✓ u_6
- ✓ u_7

Influence Spread of u_2 and u_7

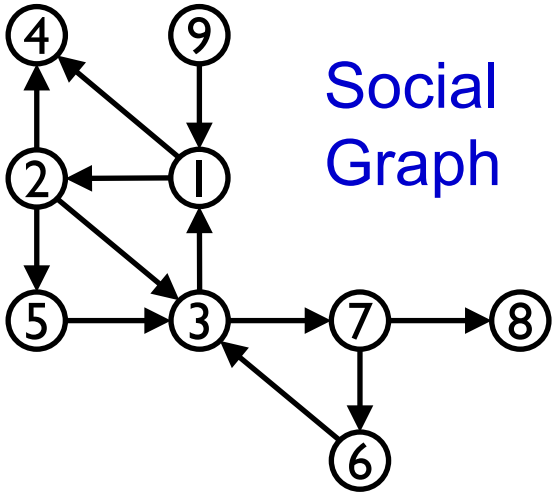
Formal Definition

- Social graph $G = (V, E)$
- Seed set $S \subset V$
- Expected influence spread $\sigma(S)$
- Problem
  - Select a seed set S^* that maximizes the expected influence spread, i.e.,

$$S^* = \arg_S \max \sigma(S)$$

Challenges of IM

- Model of Information Diffusion
  - Can IC model capture how the information is diffused in the entire social network?



Social Graph

Action Log

User	Action	Time
u_2	activated	t_1
u_3	activated	t_2
u_5	activated	t_3
…	…	…

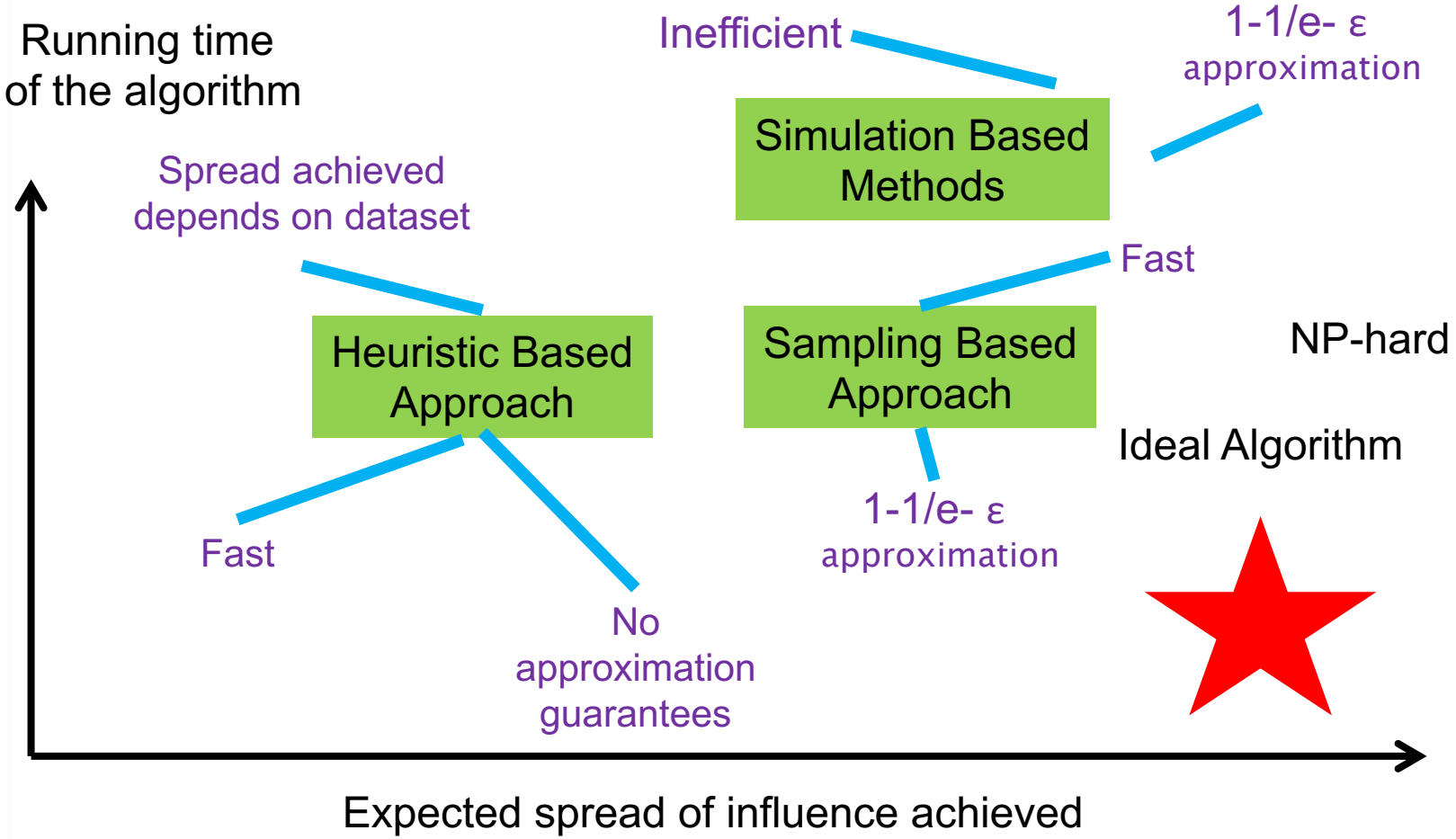
  - Challenge: How to learn a diffusion model from social graph and action log?

Challenges of IM

- Computational Complexity
  - Even under the simple IC model, the influence maximization problem is NP-Hard.
- Properties of $\sigma(S)$ under IC model
  - Monotonicity: For all $S \subseteq T \subseteq V, \sigma(S) \leq \sigma(T)$
  - Submodularity: For all $S \subseteq T \subseteq V$ and $u \in V \setminus T$, $\sigma(S \cup \{u\}) - \sigma(S) \geq \sigma(T \cup \{u\}) - \sigma(T)$
- A greedy algorithm can achieve $1 - \frac{1}{e}$ appr. ratio
- However, computing the expected influence spread $\sigma(S)$ of a node set S is #P-Hard.

Challenges of IM

- Overview of IM algorithms



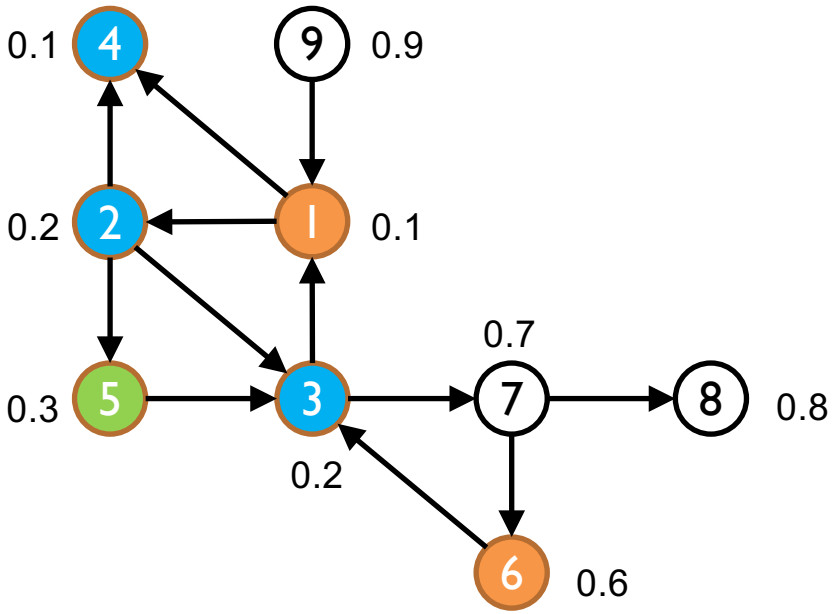
Outline

- **Diffusion Models**
- Algorithms
  - Simulation-Based
  - Heuristic-Based
  - Sampling-Based
- Topic-Aware IM
- Summary and Future Directions

Linear Threshold (LT) Model

- Each node v is activated by its each in-neighbor u according to weight $w_{u,v}$ s.t. $\sum_{u\ s.t.\ e_{u,v}\in E} w_{u,v} \leq 1$
- Initially, there is an initial set of active nodes A_0
- Each node v chooses a threshold $\theta_v \in [0,1]$ uniformly at random;
- At each discrete step $t = 1,2, \ldots \ldots$
  - Initially, $A_t = A_{t-1}$
  - For each $v \in V \setminus A_{t-1}$: If $\sum_{u\in A_{t-1}\ s.t.\ e_{u,v}\in E} w_{u,v} \geq \theta_v$, then add v into A_t;
  - If $A_t = A_{t-1}$, the diffusion process terminates

Example: LT Model

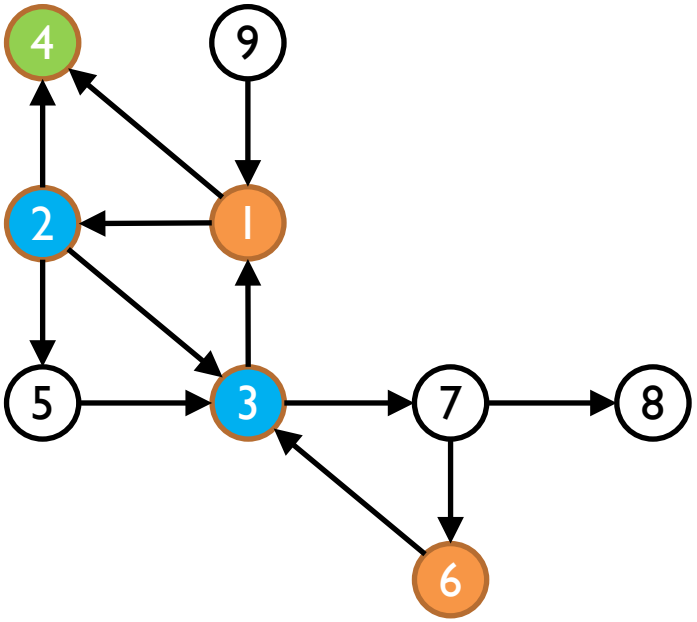


- Suppose Weights of all edges is 0.3; Thresholds are labelled;
- Initial Active Set $A_0 = \{1,6\}$;
- Step 1: $A_1 = \{1,2,3,4,6\}$ activates;
- Step 2: $A_2 = \{1,2,3,4,5,6\}$ activates;
- Step 3: No further nodes can be activated, $A_3 = A_2$, end;
- The influence spread of A_0 is $\{1,2,3,4,5,6\}$ for this diffusion.

Independent Cascade (IC) Model

- Each edge $e_{u,v} \in E$ associated with probability $p_{u,v} \in (0,1)$;
- Initially, there is an initial set of active nodes A_0
- At each discrete step $t = 1,2, \ldots \ldots$:
  - A_{t-1}: the set of nodes activated at step $t-1$;
  - A_t: the set of nodes activated at step t, initialized to $\emptyset$;
  - for each node $u \in A_{t-1}$:
    - u has a single chance to activate all its inactive out-neighbor v;
    - It succeeds with a probability $p_{u,v}$, then v is added to A_t;
    - Otherwise, u will not have further chance to activate v;
  - If $A_t = \emptyset$, the diffusion process terminates
- The set of all active nodes after the diffusion process are the influence spread of A_0.

Example: IC Model

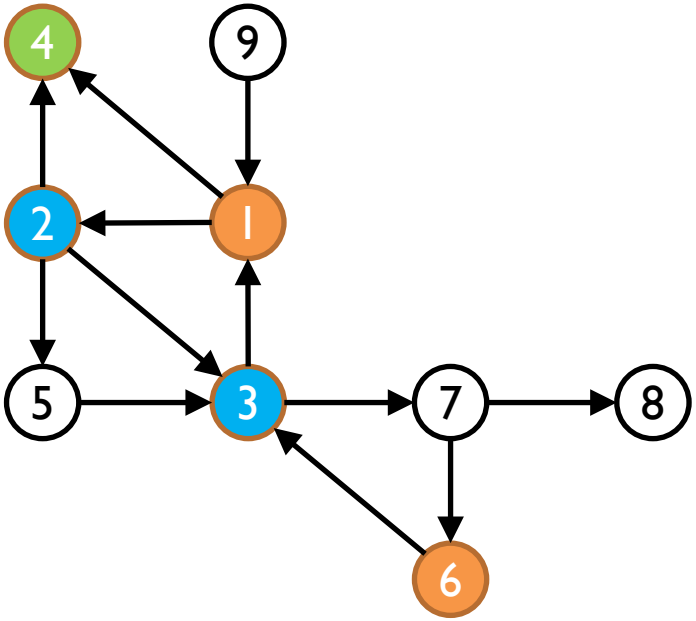


- Assume the probabilities of all edges is 0.5;
- Initial Active Set $A_0 = \{1,6\}$;
- Step 1: Node 1 tries 2,4; Node 6 tries 3; $A_1 = \{2,3\}$ activates;
- Step 2: Node 2 tries 4,5; Node 3 tries 7; $A_2 = \{4\}$ activates;
- Step 3: No further nodes can be activated, $A_3 = \emptyset$, end;
- The influence spread of A_0 is $\{1,2,3,4,6\}$ for this diffusion.

Continuous Time IC (CT-IC) Model

- The diffusion process is similar to IC
- However, the diffusion time depends on an underlying distribution.
  - $f(t_i|t_j;\alpha_{j,i})$ is the conditional likelihood of transmission between a node j and node i;
  - $\alpha_{j,i} \geq 0$ is the pairwise transmission rate; when it decreases, the expected transmission time becomes longer;
  - t_j is the activation time of node j;
  - t_i is the expected activation time of node i if the transmission form j to i exists.
- Here, we use the well-adopted exponential function as example: $f(t_i|t_j;\alpha_{j,i}) = \begin{cases} \alpha_{j,i} \cdot e^{-\alpha_{j,i}(t_i - t_j)}, & (t_j < t_i) \\ 0, otherwise \end{cases}$

Example: CT-IC Model

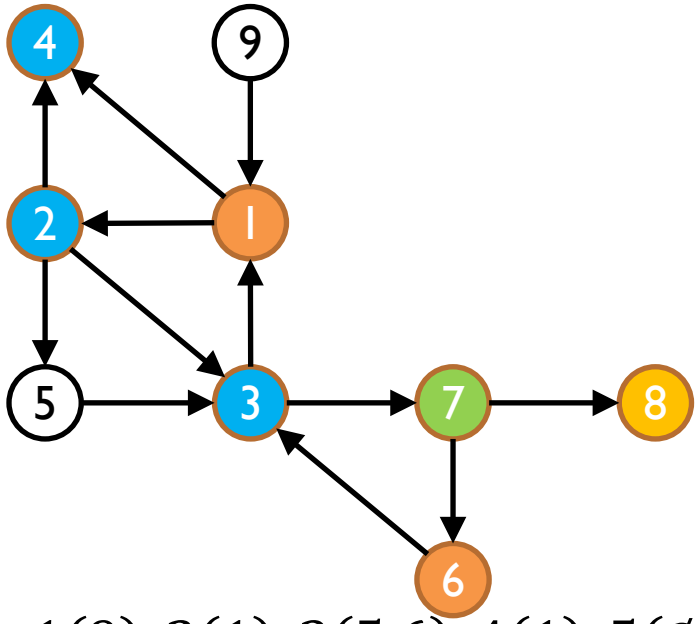


- We assume all transmission rates is 1, which means that the transmission time $(t_i - t_j)$ conforms to $EXP(\lambda = 1)$.
- Initial Active Set $A_0 = \{1,6\}$, we have $t_1 = 0, t_6 = 0$;
- At step (1), 2 is activated by 1, the transmission time $t_2 = 0.86$; 3 is activated by 6, $t_3 = 1.55$;
- At step (2), 4 is activated by 2, the transmission time $t_4 - t_2 = 0.92$, thus, $t_4 = 1.78$.

Triggering Model

- Each node v independently chooses a random "triggering set" T_v according to some distribution over subsets of its neighbors.
- Initially, there is an initial set of active nodes A_0
- For each step t, an inactive node v becomes active if it has a neighbor in its chosen triggering set T_v that is active at $t-1$.
- IC and LT model are special cases of Triggering model with different distributions of selecting triggering sets.

Example: Triggering Model



- Triggering Sets: $1(9), 2(1), 3(5,6), 4(1), 5(\emptyset), 6(\emptyset), 7(3), 8(7), 9(\emptyset)$
- Initial Active Set $A_0 = \{1,6\}$;
- Step 1: node 2 triggered by $1, 4$ by $1, 3$ by 6, $A_1 = \{2,3,4\}$;
- Step 2: node 7 triggered by 3, $A_2 = \{7\}$;
- Step 3: node 8 triggered by 7, $A_3 = \{8\}$; no further triggers, end;
- The influence spread of A_0 is $\{1,2,3,4,6,7,8\}$ for this diffusion.

Outline

- Diffusion Models
- **Algorithms**
  - Simulation-Based
  - Heuristic-Based
  - Sampling-Based
- Topic-Aware IM
- Summary and Future Directions

IM Algorithm

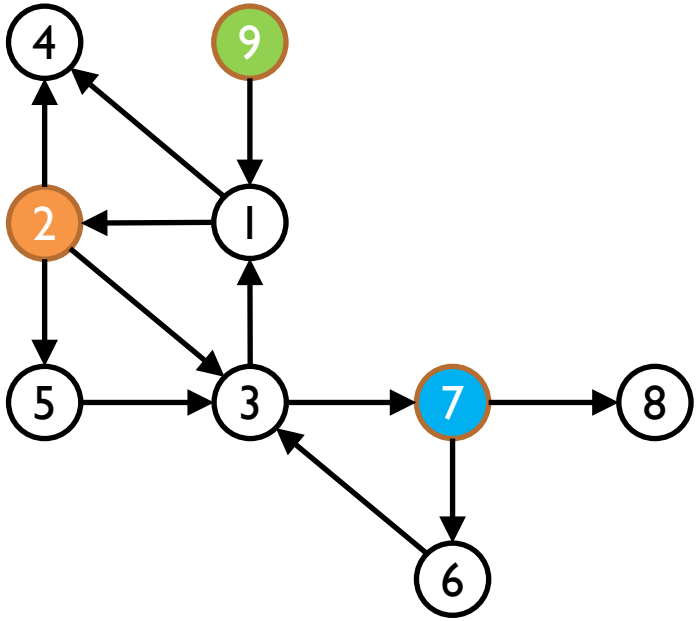
Simulation-Based Methods

The Naïve Greedy Algorithm

- Initialize $S = \emptyset$
- For $i = 1, \ldots\ldots, k$ do:
  - For each vertex $v \in V \setminus S$ do:
    - Simulate the diffusion with initial active set $A_i = S \cup \{v\}$ for R rounds, calculate the average influence spread $\bar{\sigma}_{S\cup\{v\}}$
  - $S = S \cup \{argmax_{v\in V\setminus S}\bar{\sigma}_{S\cup\{v\}}\}$
- Return S

- Theoretical Guarantee: $\sigma(S) \geq \left(1 - \frac{1}{e}\right)\sigma(OPT)$
- Time Complexity: $O(k \cdot R \cdot |V| \cdot |E|)$

Example: Greedy Algorithm



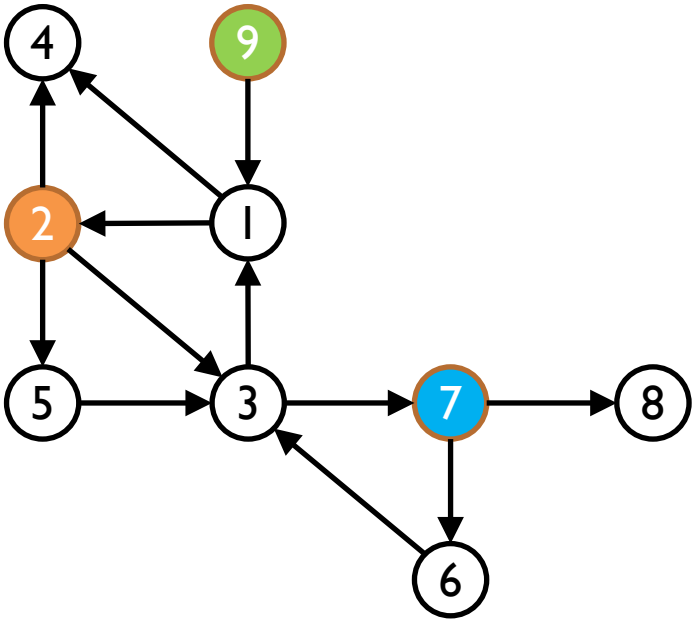
- Assume the probabilities of all edges is 0.5; $k = 3$;
- Initial Seed Set $S_0 = \emptyset$;
- Round 1: Run simulation for nodes $v = 1, \dots, 9$, $S_1 = \{2\}$;
- Round 2: Run simulation for nodes $v = 1,3, \dots, 9$ with $S_1 \cup \{v\}$, $S_2 = \{2,7\}$;
- Round 3: Run simulation for nodes $v = 1,3, \dots 6,8,9$ with $S_2 \cup \{v\}$, $S_3 = \{2,7,9\}$;
- The resultant seed set is $\{2,7,9\}$.

CELF

- Idea: Prune unnecessary simulations!

- Initialize $S = \emptyset$
- Run the first iteration $i = 1$ of Greedy Algorithm, add u_1 into S;
- Sort the remaining vertices by $\bar{\sigma}_v = \bar{\sigma}_{S \cup \{v\}} - \bar{\sigma}_S$;
- For $i = 2, \ldots \ldots, k$ do:
  - $max = 0$;
  - For each vertex $v \in V \setminus S$ in descending order of $\bar{\sigma}_v$ do:
    - If $\bar{\sigma}_v < max$, break;
    - Run simulation with $A_i = S \cup \{v\}$ for R rounds;
    - Calculate the average influence spread $\bar{\sigma}_{S \cup \{v\}}$;
    - $\bar{\sigma}_v = \bar{\sigma}_{S \cup \{v\}} - \bar{\sigma}_S$; If $\bar{\sigma}_{S \cup \{v\}} - \bar{\sigma}_S > max, max = \bar{\sigma}_{S \cup \{v\}} - \bar{\sigma}_S$;
  - $S = S \cup \{argmax_{v \in V \setminus S} \bar{\sigma}_{S \cup \{v\}}\}$
- Return S.

Example: CELF



- Assume the probabilities of all edges is 0.5; $k = 3$;
- Initial Seed Set $S_0 = \emptyset$;
- Round 1: The same as Greedy, $S_1 = \{2\}$;
- Round 2: Prune the simulation of $v = 4,8$, $S_2 = \{2,7\}$;
- Round 3: Prune the simulation of $v = 1,3,4,5,6,8$, $S_3 = \{2,7,9\}$;
- The resultant seed set is $\{2,7,9\}$.

CELF

- For each iteration $i = 2, \ldots\ldots, k$, CELF only runs simulations for a very small portion of vertices;
- Empirically, CELF is 700 times faster than Greedy.

- Why CELF works?
  - Submodularity
    - For all $S \subseteq T \subseteq V$ and $u \in V \setminus T, \sigma(S \cup \{u\}) - \sigma(S) \geq \sigma(T \cup \{u\}) - \sigma(T)$
  - If $i < j, S_i \subset S_j$, then $\sigma(S_i \cup \{u\}) - \sigma(S_i) \geq \sigma\big(S_j \cup \{u\}\big) - \sigma(S_j)$ holds for all $u \in V \setminus S_j$.

Pros and Cons

- Pros:
  - The seed set is near optimal: $(1 - \frac{1}{e})$ is the best achievable guarantee for influence maximization;
- Cons:
  - High Computational Cost
    - CELF runs nearly a hour for a graph with 10,000 nodes and 100,000 edges;
    - Greedy Algorithm even needs nearly one day…
  - Not Scalable to Large Graphs
    - Twitter graph has 2 billion users and tens of billions edges

IM Algorithm

Heuristic-Based Methods

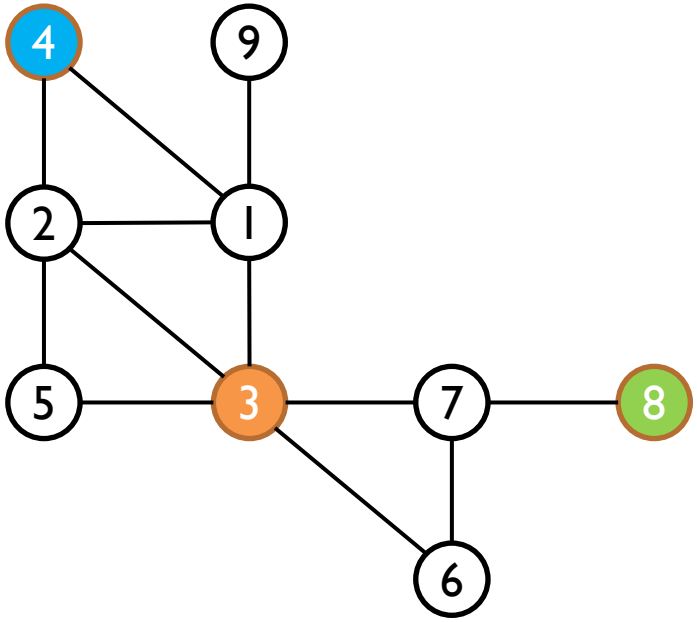
Overview

- Objective
  - Solve the efficiency and scalability problems of simulation-based approaches
- High-Level Idea
  - Transform IM to another related problem, which is often solvable in polynomial time
  - Use its solution as an approximate solution for IM

Degree Discount

- Intuition
  - Nodes with higher out-degree ≈ more influential
  - Seed nodes should not overlap with each other
- Initialize $S = \emptyset$
- Compute the out-degree of each node;
- For $i = 1, \ldots\ldots, k$ do:
  - Select the node u with highest remaining degrees into S;
  - For each vertex $v \in V \setminus S$ such that v is a neighbor of u do:
    - Discount the degree of v according to edge probability $p_{u,v}$;
- Return S.
- Time Complexity: $O(k \cdot \log|V| + E)$

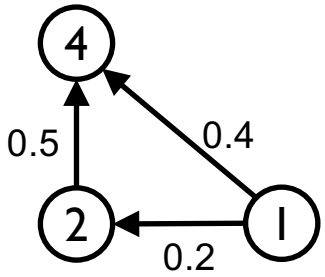
Example: Degree Discount



- Assume the probabilities of all edges is 0.1; $k = 3$;
- Initial Seed Set $S_0 = \emptyset$;
- Round 1: Select the node with the largest degree, $S_1 = \{3\}$;
- Round 2: Select the node with the largest remaining degree, $S_2 = \{3,4\}$;
- Round 3: Select the node with the largest remaining degree, $S_3 = \{3,4,8\}$;
- The resultant seed set is $\{3,4,8\}$.

PMIA

- Maximum Influence Path
  - Propagation Probability of a path $P\langle u_1, \dots, u_m \rangle$ is $pp(P) = \prod_{i=1}^{m-1} p(u_i, u_{i+1})$
- MIA (Maximum Influence Arborescence) Model
  - Maximum Influence Path (MIP): MIP_{uv} is the path with the highest propagation probability among all paths from u to v.

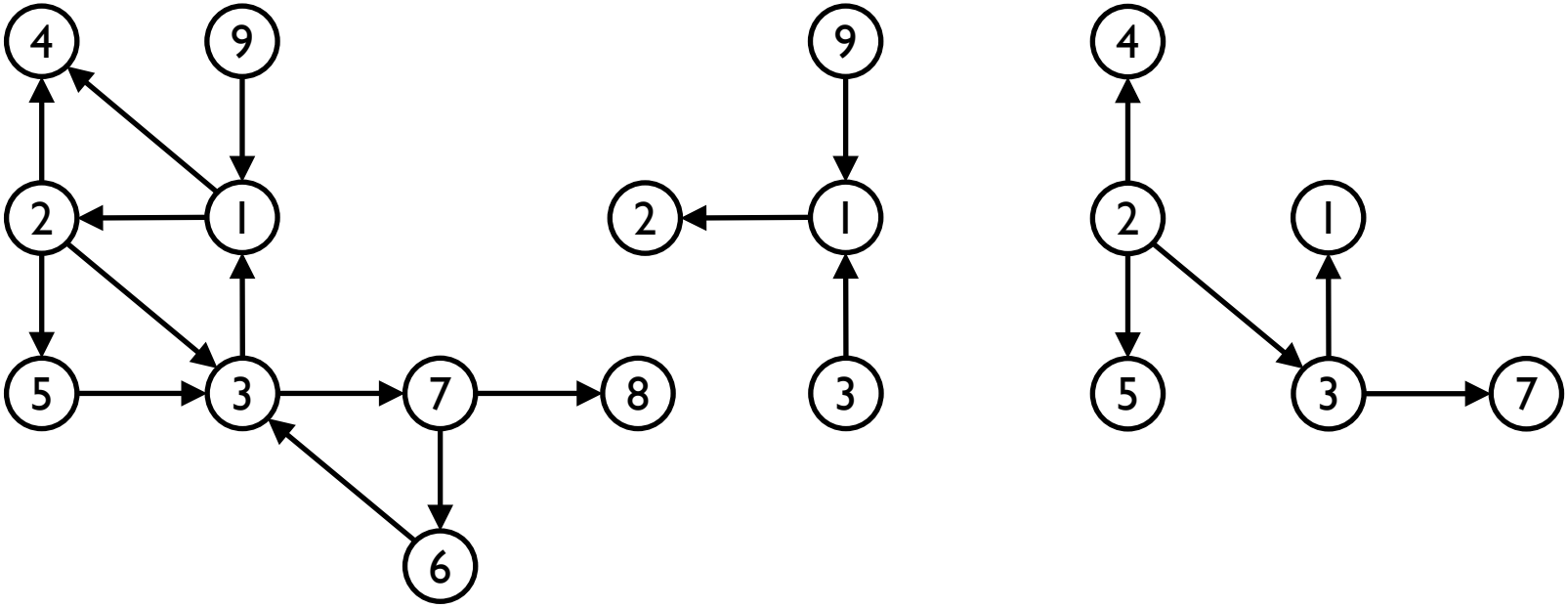


- For path $P_a\langle 1,2,4 \rangle$, $pp(P_a) = 0.2 \times 0.5 = 0.1$
- $MIP_{14} = P_b\langle 1,4 \rangle$, since $pp(P_b) = 0.4 > pp(P_a)$

PMIA

- MIA (Maximum Influence Arborescence) Model
  - Given an influence threshold θ
  - MIIA (Maximum Influence In-Arborescence)
    - $MIIA(v, \theta)$ contains all MIP to v with $pp \geq \theta$
    - $MIIA(v, \theta) = \cup_{u \in V, pp(MIP_{uv}) \geq \theta} MIP_{uv}$
  - MIOA (Maximum Influence Out-Arborescence)
    - $MIOA(v, \theta)$ contains all MIP from v with $pp \geq \theta$
    - $MIOA(v, \theta) = \cup_{u \in V, pp(MIP_{vu}) \geq \theta} MIP_{vu}$
  - **MIA** model assumes that a node v can only be influenced through its MIIA
  - MIIA and MIOA can be generated by adapting the Dijkstra algorithm for the shortest paths on graphs

Example: MIIA & MIOA



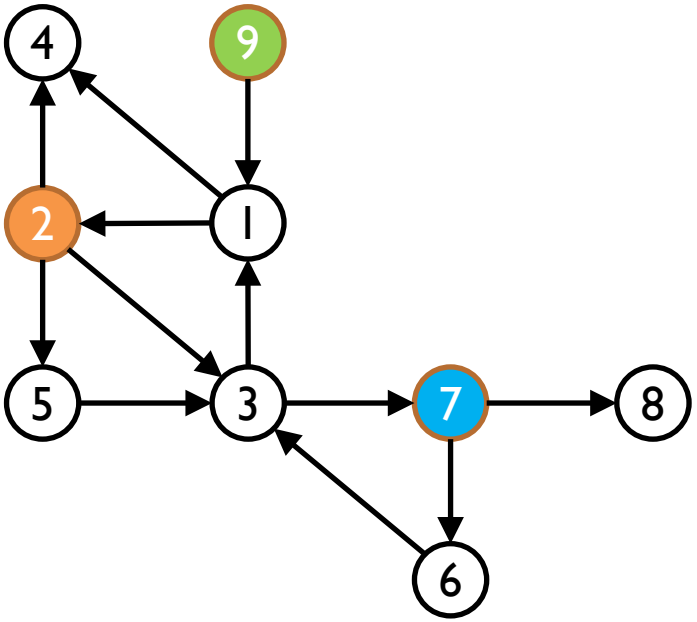
(a) Graph $G = (V, E)$ (b) MIIA of node 2 (c) MIOA of node 2

- Assume the probabilities of all edges is 0.5; $\boldsymbol{\theta = 0.2}$.
- Then, we show the MIIA and MIOA of node 2 in (b) and (c).

The PMIA Algorithm

- The Influence Spread of S in the MIA model
  - $\sigma_M(S) = \sum_{v \in V} ap(v, S, MIIA(v, \theta))$, where $ap(v, S, MIIA(v, \theta))$ is the active probability of v through paths in $MIIA(v, \theta)$ from S
- Influence Maximization in the MIA model is also NP-Hard
- A similar greedy algorithm guarantees a $(1 - \frac{1}{e})$ approximation

Example: MIA Algorithm



- Assume the probabilities of all edges is $0.5; k = 3; \theta = 0.2$
- Initial Seed Set $S_0 = \emptyset$;
- Round 1: $\sigma_M(\{2\}) = 3, S_1 = \{2\}$;
- Round 2: $\sigma_M(\{2,7\}) = 4.875, S_2 = \{2,7\}$;
- Round 3: $\sigma_M(\{2,7,9\}) = 6.5, S_3 = \{2,7,9\}$;
- The resultant seed set is $\{2,7,9\}$.

Pros and Cons

- Pros:
  - Heuristic methods are usually orders of magnitude faster than simulation-based method;
  - Usually, the influence spread of seeds selected by heuristic methods is nearly equivalent ($\geq 90\%$) to those of simulation-based method.
- Cons:
  - Heuristic methods cannot provide the solution guarantees theoretically.
  - In some extreme cases, heuristic methods may provide undesired bad solutions.

IM Algorithm

Sampling-Based Method

Motivation

- Can we have an algorithm that is

Efficient/Scalable

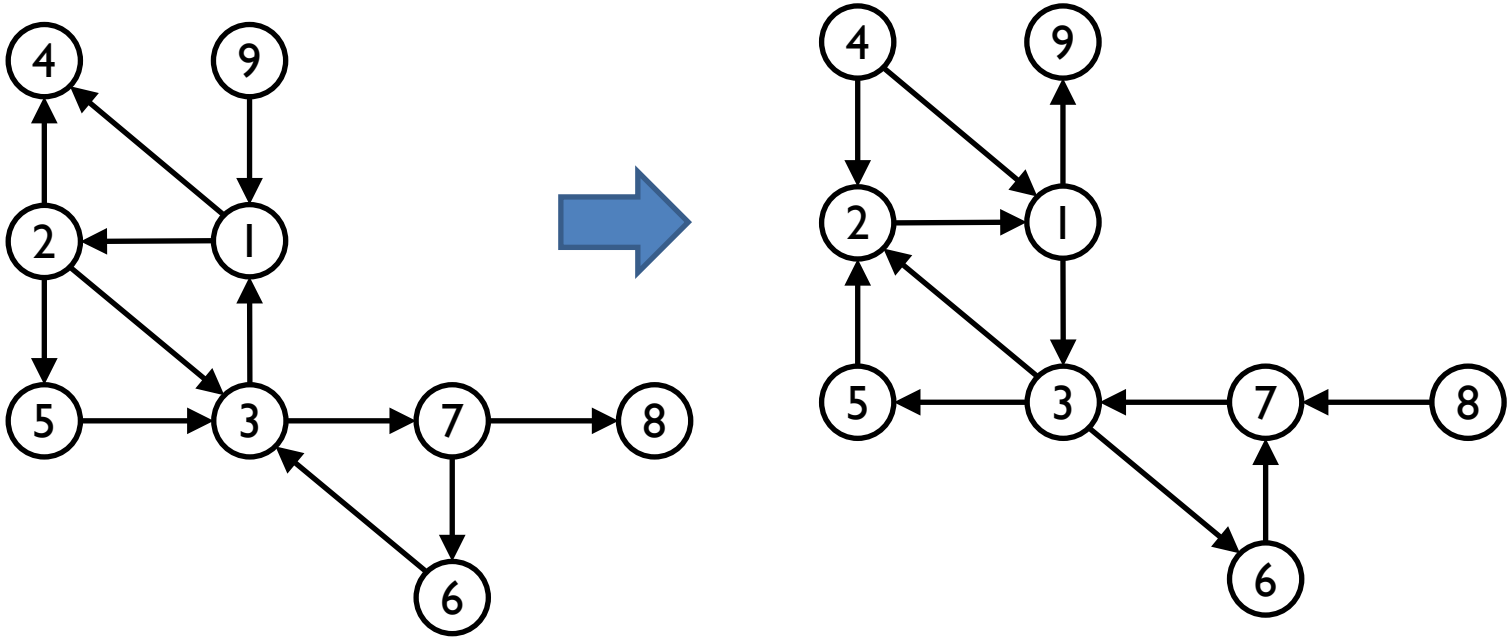
and

Theoretically Bounded

RIS: Reverse Influence Sampling

- Generate reverse graph $G^T = (V, E^T)$
  - If $e(i \to j)$ exists in E, then $e(j \to i)$ exists in E^T
- Choose a node $u \in V$ uniformly at random;
- Run simulation under diffusion model to generate a sketch;
- The total size of all sketches is $\Theta(k \cdot (|V| + |E|) \cdot \log|V| \cdot \varepsilon^{-3})$.

Example: Reverse Sampling



G G^T

Sample Sketches:

2(6 1 3) 9 7 8

2(6 9 1 3 7 5) 6 4(2 9 1)

……

RIS: Reverse Influence Sampling

- Phase 2: Build Seed Set
  - Degree discount on all sampled sketches

  - Input: seed size k, sketches H
  - Initialize $S = \emptyset$
  - For $i = 1, \ldots, k$ do
    - Add $\boldsymbol{v_i}$ with the highest degree in H into S
    - Remove v_i and all incident edges from H
  - Return S

Example: Build Seed Set

Seed Set: $S = \{2,7,9\}$

Node	Round 1	Round 2	Round 3
1	434	130	123
2	510	×	×
3	460	160	80
4	150	65	62
5	393	136	102
6	366	210	103
7	348	219	×
8	154	134	77
9	353	210	208

TIM & IMM

- Reverse Influence Sampling (RIS)
  - scalable and efficient in theory
  - still incurs significant computational cost in practice
  - Time Complexity: $\Theta(k \cdot (|V| + |E|) \cdot log|V| \cdot \varepsilon^3)$
- Two-Phase Influence Maximization (TIM)
  - Control the number of random RR sets
  - Phase 1: Parameter Estimation
    - Minimize the number of random RR sets by maintaining the upper bound of the optimal influence spread
  - Phase 2: Node Selection (the same as RIS)
  - Time Complexity: $O((k + l) \cdot (|V| + |E|) \cdot log|V| \cdot \varepsilon^2)$
  - Empirically, 2-3 orders of magnitude faster than RIS

TIM & IMM

- IMM: Influence Maximization with Martingale
  - Improve the Parameter Estimation phase of TIM by a tighter estimation of OPT and eliminating redundant random sets
  - Improve the Node Selection phase of TIM by a martingale approach
  - Support Continuous Time Independent Cascade (CT-IC) model
  - Retains the time complexity and approximation ratio of TIM
  - Empirically, one order of magnitude faster

Pros and Cons

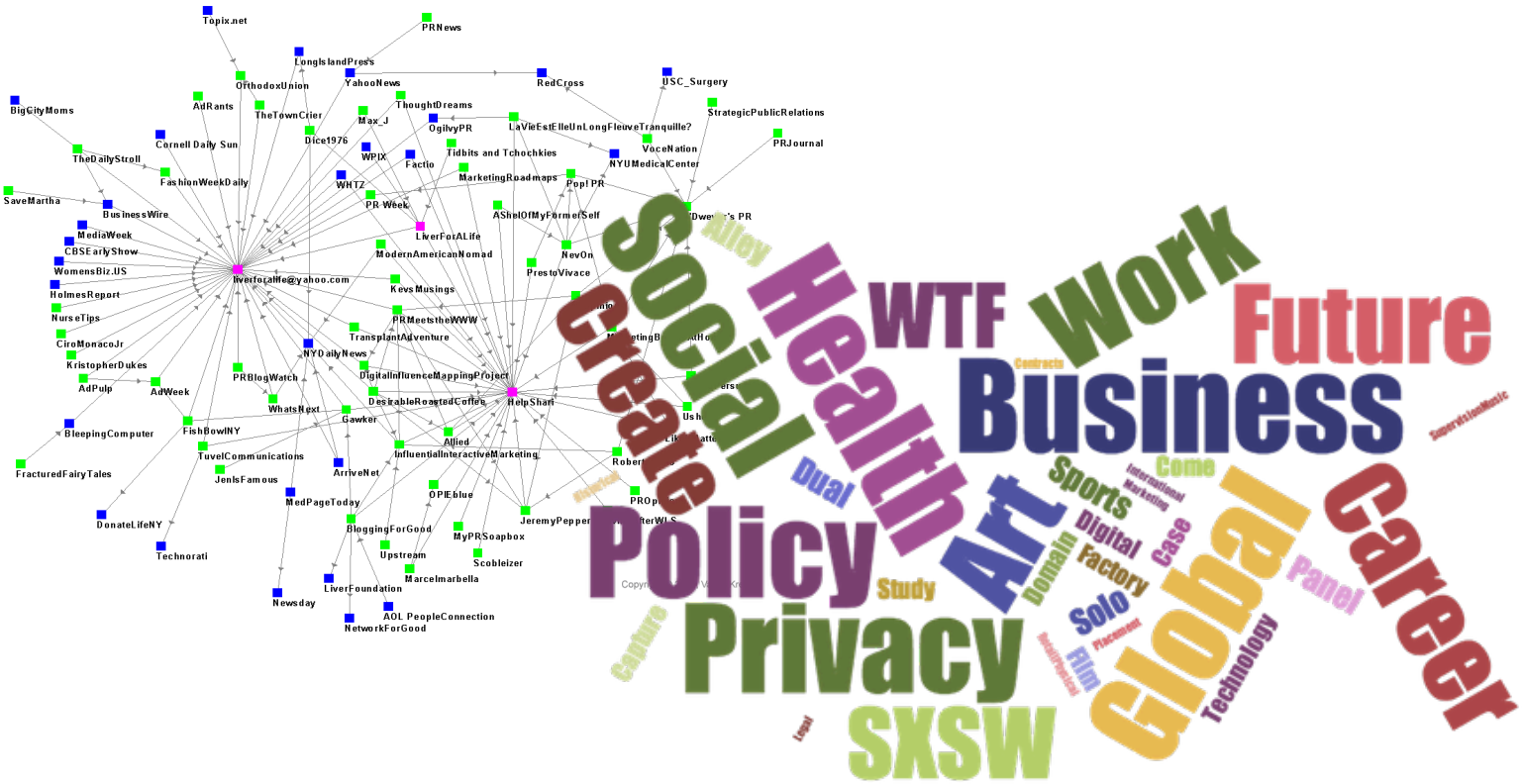
- Pros:
  - Efficient and Scalable: near linear to $|V| + |E|$
  - Theoretical Guarantee: $1 - \frac{1}{e} - \varepsilon$
  - Applicable to different diffusion models: IC, LT, Trigger, Continuous Time-IC
- Cons:
  - Handling dynamic updates on graphs

Outline

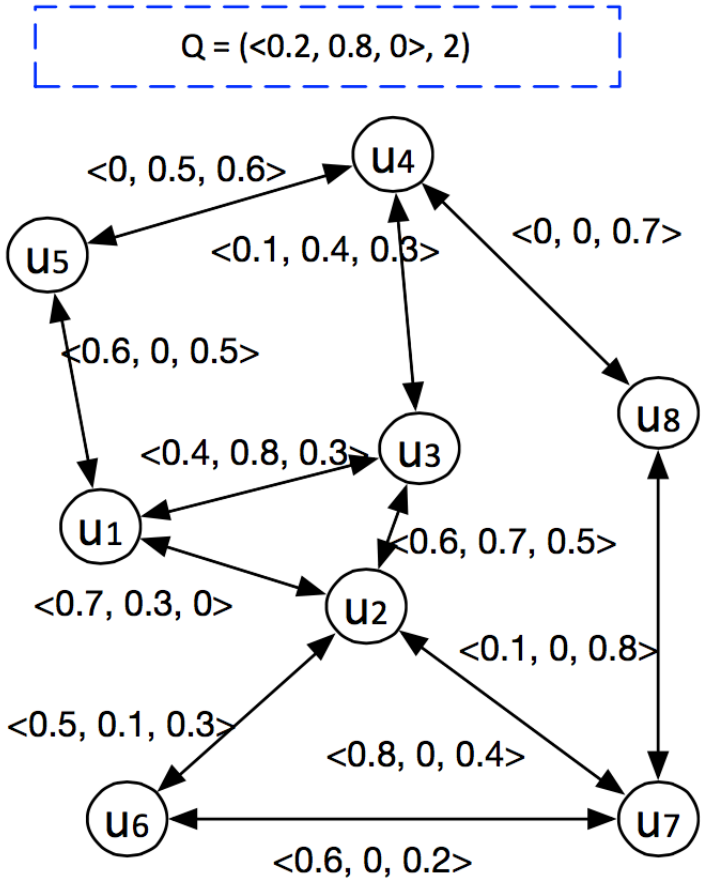
- Diffusion Models
- Algorithms
  - Simulation-Based
  - Heuristic-Based
  - Sampling-Based
- **Topic-Aware IM**
- Summary and Future Directions

Topic-Aware Information Diffusion

Influence network → Topics/Interests → Applications



Topic-Aware IC Model

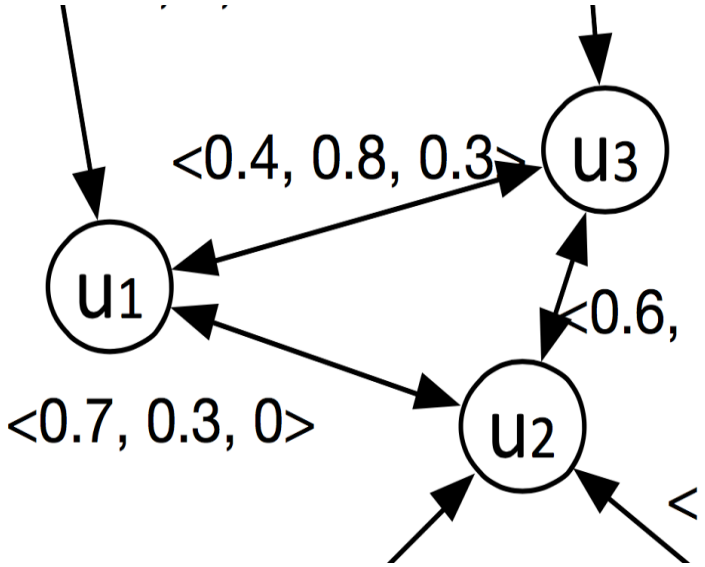


Edge	Topic 1	Topic 2	Topic 3
(1, 2)	0.7	0.3	0
(1, 3)	0.4	0.8	0.3
(1, 5)	0.6	0	0.5
(2, 3)	0.6	0.7	0.5
(2, 6)	0.5	0.1	0.3
(2, 7)	0.8	0	0.4
(3, 4)	0.1	0.4	0.3
(4, 5)	0	0.5	0.6
(4, 8)	0	0	0.7
(6, 7)	0.6	0	0.2
(7, 8)	0.1	0	0.8

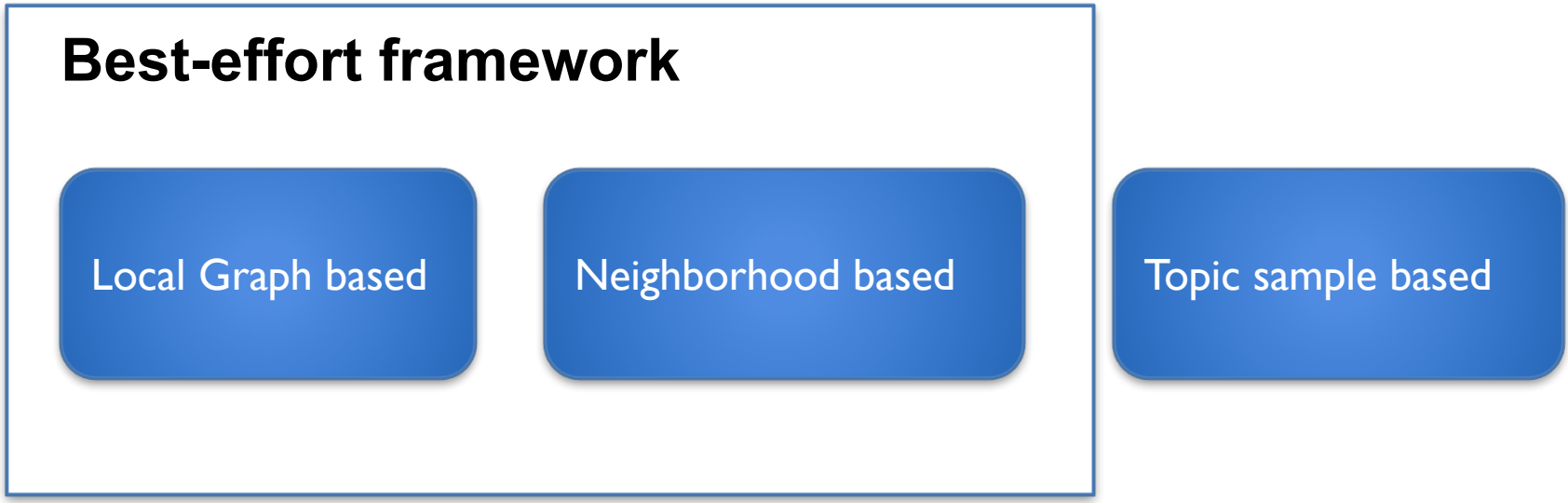
Challenges of Topic-Aware IM

- Challenges
  - Diffusion graph is query-dependent
  - Given a query, a naïve approach has to generate a graph by computing probability at each edge

- Given query <0.2, 0.8, 0>
- Influence from u1 to u2 = 0.7*0.2 + 0.3*0.8 + 0 = **0.38**



Best-Effort Approach

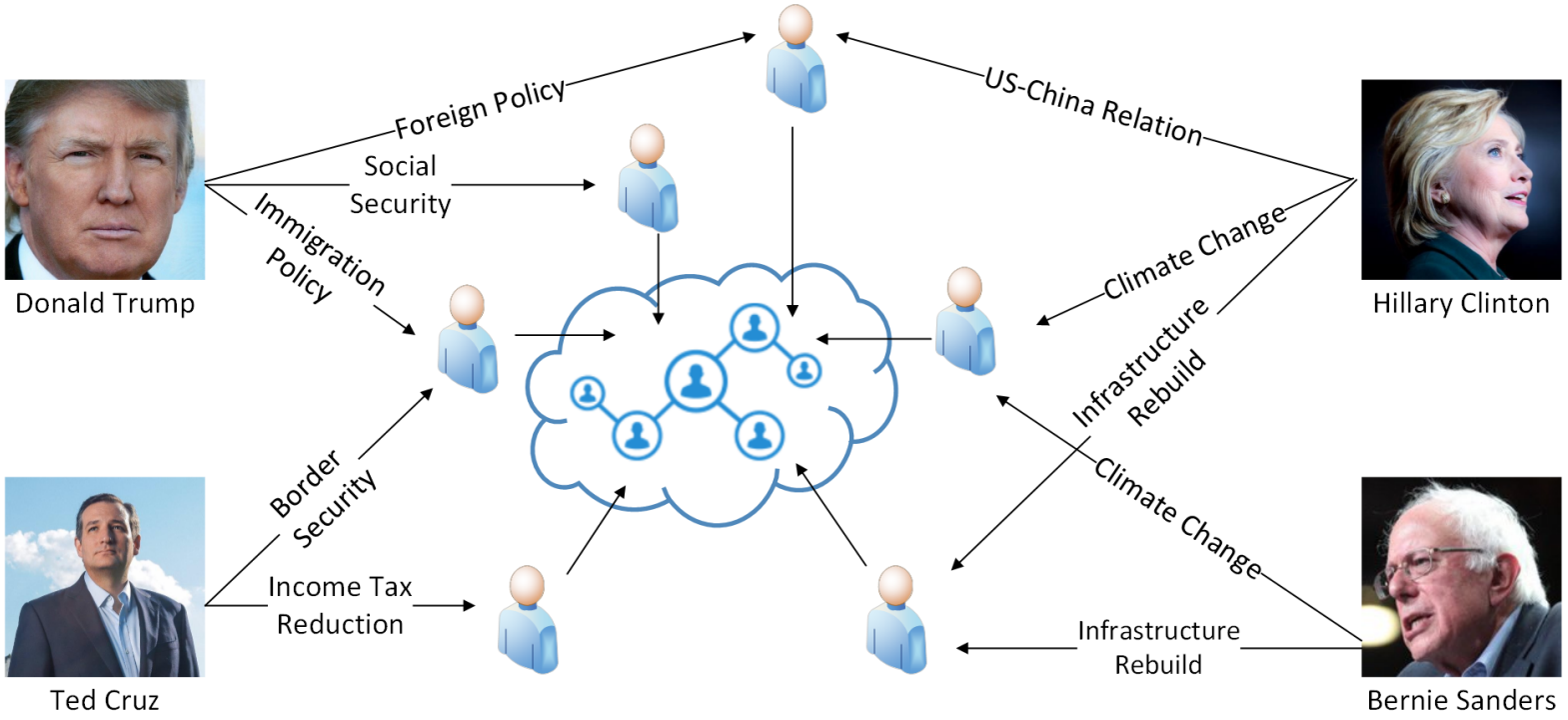


- Best-effort approach
- Focus on largely affected local vertices
- Substitute other insignificant vertices with constants

- Best-effort approach
- More accurate upper bound
- Use in/out neighborhood as upper bound

- Approximation
- Premature termination with performance guarantee
- Pareto upper bound sampling

How to discover "Selling-Points"



Outline

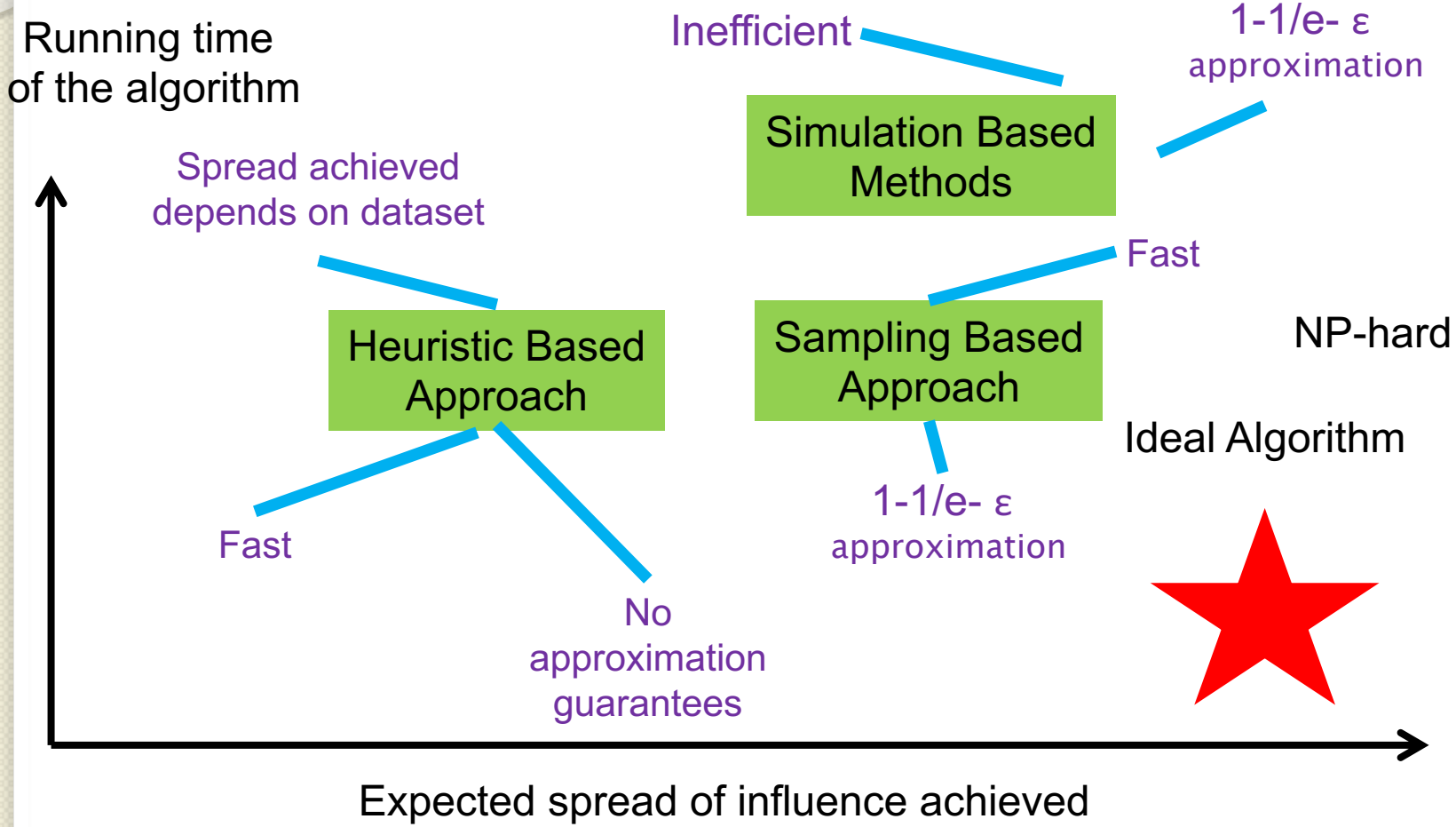
- Diffusion Models
- Algorithms
  - Simulation-Based
  - Heuristic-Based
  - Sampling-Based
- Topic-Aware IM
- **Summary and Future Directions**

Summary

- The IM problem
  - Selecting k users such that by activating them, the expected spread of influence is maximized.
- Computational Complexity
  - The IM problem is NP-hard
  - Computing influence spread is #P-hard
- Diffusion Model
  - IC, LT, CT-IC, Triggering
  - Properties of influence spread
    - Monotonicity & Submodularity

Summary

- The IM algorithms



Future Directions

- Better sampling strategies
- Influence maximization under dynamic social graph & streaming action logs
- Budget-aware influence maximization
- Influence maximization for recommendation

Thanks

Q & A

Homepage: http://iir.ruc.edu.cn/~fanj/