APPLICATIONS OF SOCIAL DIFFUSION AND INFLUENCE ALGORITHMS IN TENCENT GAMES

National University of Singapore

Presented by
Shiqi Zhang
April, 18th, 2024

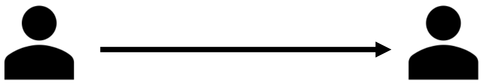
INFORMATION DIFFUSION MEETS INVITATION MECHANISM

Shiqi Zhang, Jiachen Sun, Wenqing Lin,
Xiaokui Xiao, Yiqian Huang, Bo Tang

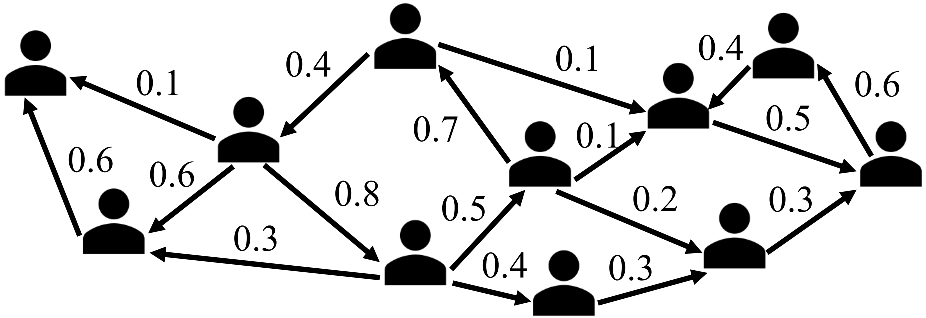
INVITATION MECHANISM

- Invitation is also everywhere in Tencent games
- The invitation behavior can cascade

HOW TO MODEL INFORMATION DIFFUSION VIA INVITATION MECHANISM?



IC model:
u independently influences
v with probability $p_{u,v}$



social network

- Traditional diffusion models ignore the conversion funnel in invitation mechanism.

CONVERSION FUNNEL

- A distillation of a user’s journey
- Describe how user behavior changes in multiple stages

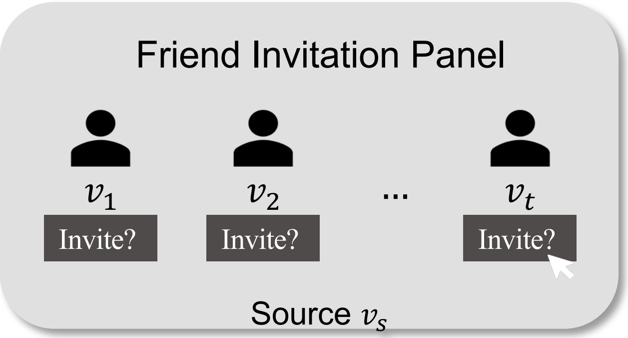
Awareness: business outreach

Adoption: platform activeness

Action: e-commerce revenue

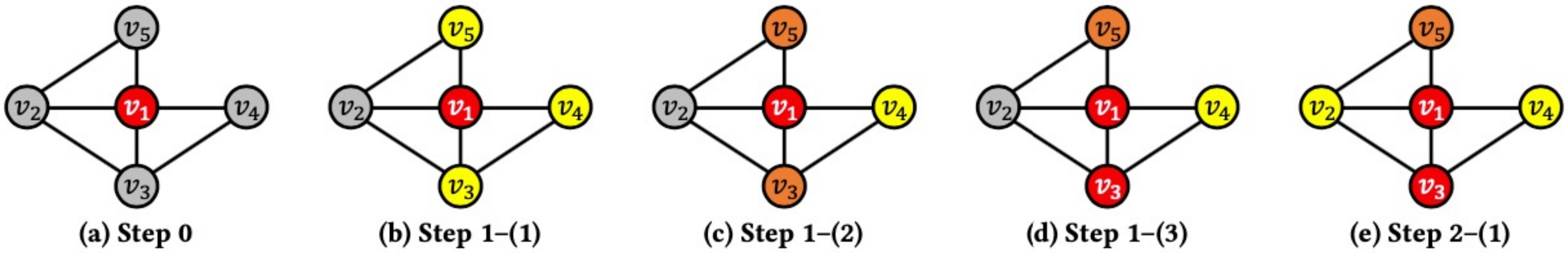
EVERY STAGE MATTERS

INVITATION CONVERSION FUNNEL



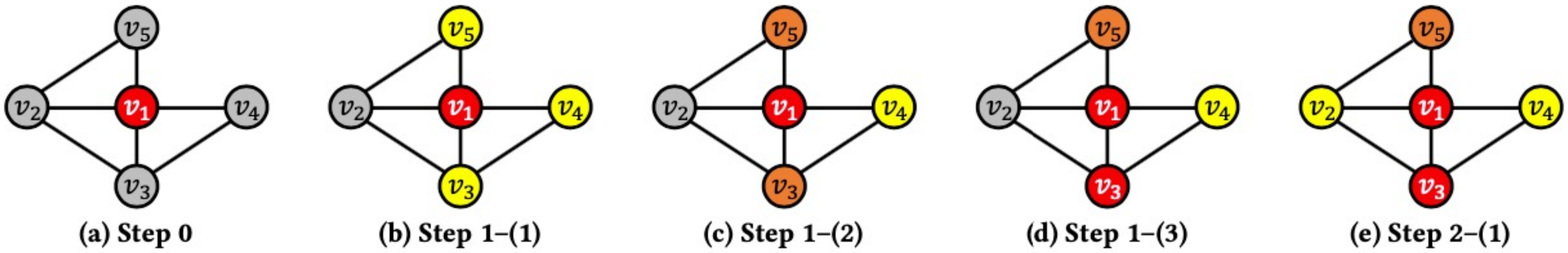
- Conversion funnel of a user

ICI: INDEPENDENT CASCADE WITH INVITATION MODEL



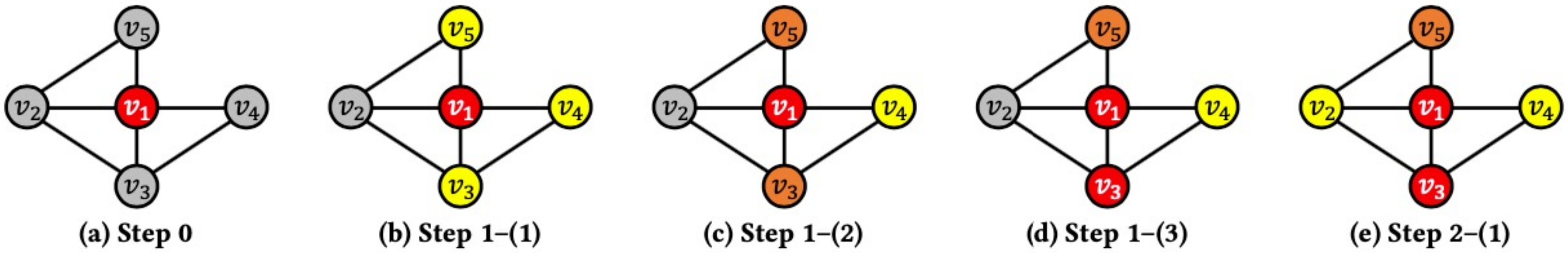
- User roles:
  - Inactive state: uninformed (grey)
  - Active states: inviter (red), invitee (yellow), acceptor (orange)

ICI: INDEPENDENT CASCADE WITH INVITATION MODEL



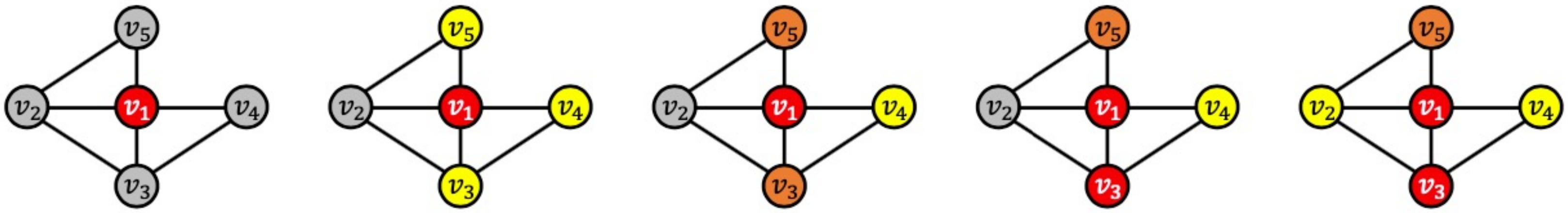
- Given the seeds, a diffusion instance unfolds in discrete steps
  - At step 0
    - all seeds → initial inviters
    - others → uninformed

ICI: INDEPENDENT CASCADE WITH INVITATION MODEL



- Given the seeds, a diffusion instance unfolds in discrete steps
  - At the subsequent step
    - new inviter v_i has a probability $p_{i,j}$ to invite the uninformed friend v_j
    - if v_j becomes an invitee, it has a probability β to be an acceptor
    - if v_j becomes an acceptor, it has a probability γ to be an inviter

ICI: INDEPENDENT CASCADE WITH INVITATION MODEL



(a) Step 0 **(b) Step 1–(1)** **(c) Step 1–(2)** **(d) Step 1–(3)** **(e) Step 2–(1)**

- Given the seeds, a diffusion instance unfolds in discrete steps
  - Stops when no new inviters exist

APPLICATION: CASCADE ESTIMATION

- Objective:
  - Estimate the number of influenced users from a given seed set S
- Solution:
  - Given a diffusion model M, estimate the average number of influenced users from S under M by T simulations
- Model M
  - Our proposal: ICI (treat acceptors as influenced users)
  - 6 competitors: IC, CT-IC, IC-N, LT, LT-C, F-TM (by defaults)

APPLICATION: CASCADE ESTIMATION

Table 1: Dataset statistics ($K = 10^3$, $M = 10^6$).

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{S}\rvert$	Spread	Type
TXG-A	$153.0K$	$2.3M$	$10.3K$	$12.8K$	Invitation
TXG-B	$155.5K$	$2.5M$	$4.9K$	$12.6K$	Invitation
TXG-C	$155.9K$	$2.5M$	$4.4K$	$11.0K$	Invitation
TXG-D	$133.9K$	$2.1M$	$12.2K$	$76.4K$	Invitation
Diggs	$279.6K$	$1.5M$	$0.6K$	$8.1K$	Vote
Twitter	$456.6K$	$12.5M$	$27.0K$	$38.7K$	Retweet

Table 2: The RMSE of estimating overall spreads ($\times 10^3$).

Model	***TXG-A***	***TXG-B***	***TXG-C***	***TXG-D***	***Diggs***	***Twitter***
IC	40.6	32.7	32.7	39.7	40.9	13.2
CT-IC	*20.9*	*8.3*	*8.1*	*22.9*	30.8	42.0
IC-N	23.4	14.8	14.9	23.8	*22.0*	*76.7*
LT	97.1	100.0	101.7	88.6	59.6	227.4
LT-C	69.6	71.9	73.6	63.2	42.7	161.1
F-TM	103.1	112.0	113.4	92.2	120.6	241.6
ICI	**11.2**	**1.7**	**2.1**	**13.4**	**7.2**	**37.1**

- ICI outperforms all competitors across all test datasets in terms of RMSE

APPLICATION: DIFFUSION PREDICTION

- Objective:
  - Predict if each user is (directly/indirectly) influenced by a given seed set S
- Solution:
  - Given a diffusion model M, prediction is

$$\hat{y}_i = \frac{\text{\# times that } v_i \text{ is influenced from S under M over T}}{\text{T simulations}}$$

- Model M
  - Our proposal: ICI (treat acceptors as influenced users)
  - 6 competitors: IC, CT-IC, IC-N, LT, LT-C, F-TM (by defaults)

APPLICATION: DIFFUSION PREDICTION

Table 3: The AUC (%) and MAP (%) of different models in diffusion prediction.

Model		IC	CT-IC	IC-N	LT	LT-C	F-TM	IC+	ICI
TXG-A	**AUC**	82.11±0.08	79.30±0.10	82.36±0.10	78.29±0.03	77.77±0.07	77.32±0.17	*82.58±0.12*	**83.36±0.06**
	MAP	20.07±0.13	18.35±0.08	20.34±0.12	16.51±0.23	16.15±0.19	18.99±0.19	*20.69±0.05*	**20.71±0.12**
TXG-B	**AUC**	81.96±0.05	80.76±0.05	83.06±0.11	74.17±0.04	73.98±0.10	75.95±0.17	*83.30±0.15*	**84.43±0.10**
	MAP	19.48±0.06	20.13±0.06	21.05±0.11	12.41±0.12	12.37±0.14	16.10±0.24	*21.54±0.18*	**22.05±0.15**
TXG-C	**AUC**	82.26±0.09	81.23±0.07	83.35±0.13	73.56±0.06	73.28±0.07	75.06±0.17	*83.56±0.13*	**84.90±0.08**
	MAP	18.82±0.12	19.42±0.08	20.43±0.16	11.10±0.21	10.89±0.09	13.83±0.20	*20.81±0.11*	**21.41±0.09**
TXG-D	**AUC**	78.20±0.04	74.30±0.11	78.47±0.08	78.12±0.04	77.11±0.08	75.57±0.21	*78.35±0.06*	**78.98±0.07**
	MAP	20.04±0.04	16.43±0.06	20.03±0.03	20.03±0.08	19.14±0.18	20.01±0.14	*20.08±0.04*	**20.11±0.02**
Diggs	**AUC**	86.65±0.03	82.03±0.04	87.58±0.06	87.82±0.02	87.83±0.03	**90.18±0.05**	88.06±0.03	*89.67±0.06*
	MAP	10.19±0.02	7.25±0.01	11.52±0.12	11.85±0.08	12.02±0.06	**26.21±0.14**	12.23±0.03	*15.95±0.22*
Twitter	**AUC**	70.39±0.04	72.37±0.04	72.88±0.03	69.91±0.03	69.29±0.05	68.80±0.06	*76.62±0.04*	**77.97±0.04**
	MAP	15.97±0.03	19.12±0.04	18.27±0.06	14.35±0.04	14.59±0.06	15.40±0.04	*21.17±0.03*	**22.40±0.05**

- ICI outperforms all competitors on all test datasets but Diggs

APPLICATION: FRIEND RANKING

- Objective: recommend existing friends for players to improve engagement
- Solution:
  - Compute each friend's influence spread under IC/ICI model
  - Rank friends based on their spread in descending order
  - Select the top k friends to recommend
- Competitor: Intimacy
  - Rank friends based on the number of historical interactions with the player
  - Select the top k friends to recommend

APPLICATION: FRIEND RANKING

- Performance on social lottery events of one Tencent RPG game

Metrics	**ICI**	**IC**	**Intimacy**
Invitation Rate	**9.60%**	6.24%	7.98%
Pay Rate	**35.15%**	32.91%	26.71%

Metrics	**ICI**	**IC**	**Intimacy**
Invitation Rate	**17.89%**	16.85%	16.15%
Pay Rate	**30.91%**	24.53%	29.80%

APPLICATION: KOL SELECTION

- Objective: identify k influencers to maximize the event outreach
- Solution:
  - Treat IC/ICI as the diffusion model
  - Invoke the greedy algorithm of influence maximization to select k seeds
- Competitor: degree
  - Select k players with the largest degree centrality

APPLICATION: KOL SELECTION

- Performance on viral marketing events of one Tencent battle royale game

Metrics	ICI	IC	Degree
Spread Increment	**2286**	1923	843
Invition Rate	**46.20%**	39.64%	32.44%

SUMMARY

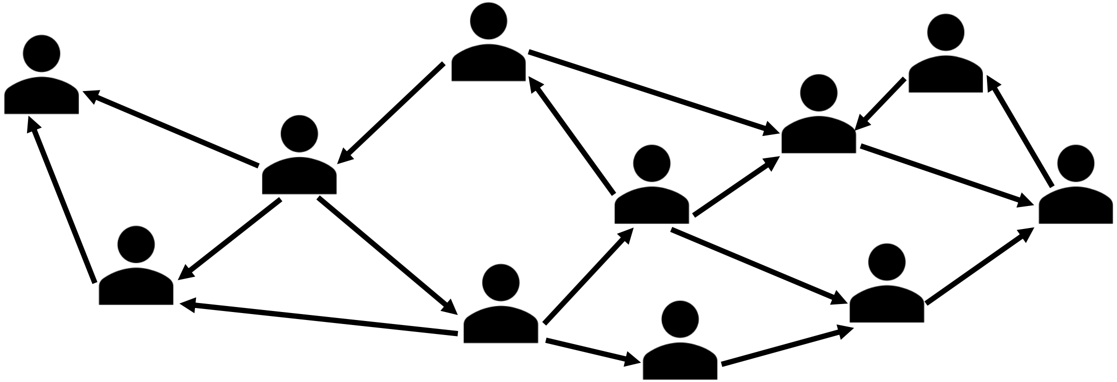
- ICI: a new diffusion model considering invitation mechanism
- Better performance on cascade estimation and diffusion prediction
- Have been deployed to friend ranking and KOL selection on Tencent gaming platforms

CAPACITY CONSTRAINED INFLUENCE MAXIMIZATION IN SOCIAL NETWORKS

Shiqi Zhang, Yiqian Huang, Jiachen Sun, Wenqing Lin
Xiaokui Xiao, Bo Tang

INFLUENCE MAXIMIZATION (IM)

followee can influence follower

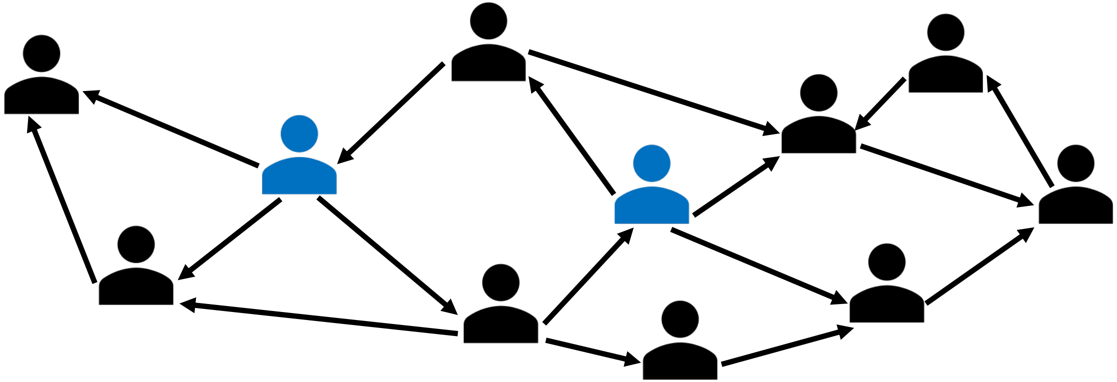


social network

- Motivated by viral marketing in social networks

INFLUENCE MAXIMIZATION (IM)

initial adopter:
paid by the merchant

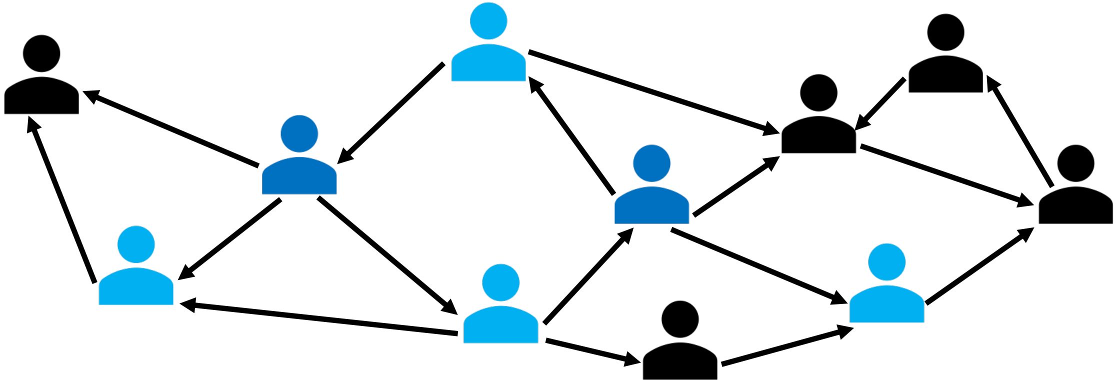


social network

- Motivated by viral marketing in social networks
  - pay *k* individuals

INFLUENCE MAXIMIZATION (IM)

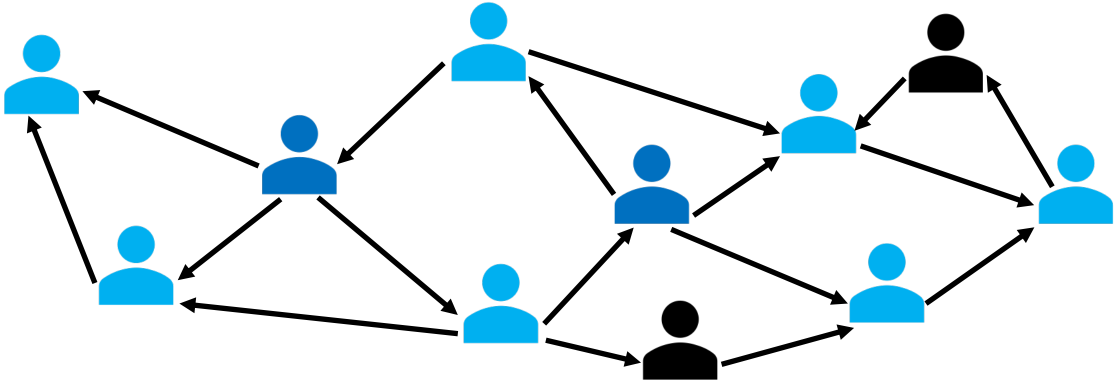
subsequent adopter:
influenced by friends



social network

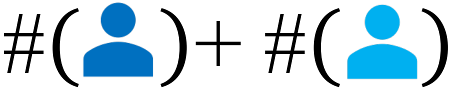
- Motivated by viral marketing in social networks
    - hope word-of-mouth promotes the given product

INFLUENCE MAXIMIZATION (IM)

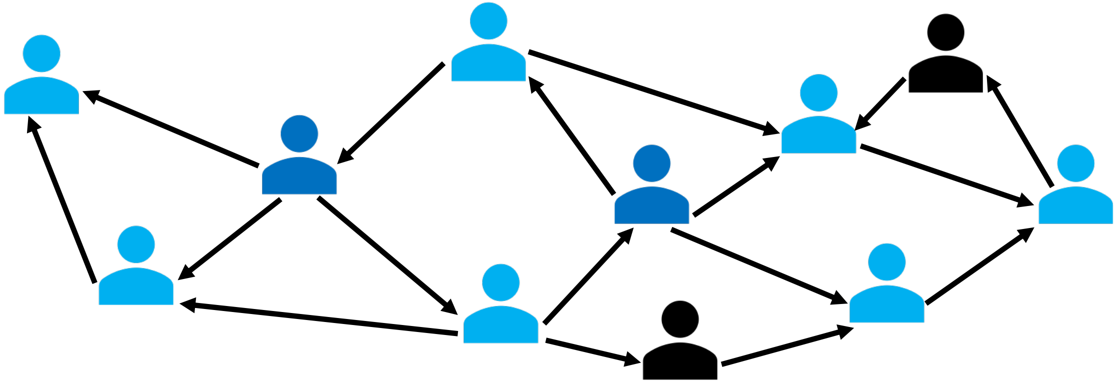


- Motivated by viral marketing in social networks
  - hope word-of-mouth promotes the given product
  - create a *cascade of influence*

INFLUENCE MAXIMIZATION (IM)



influence spread

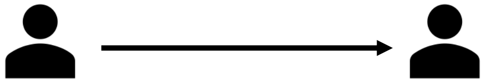


social network

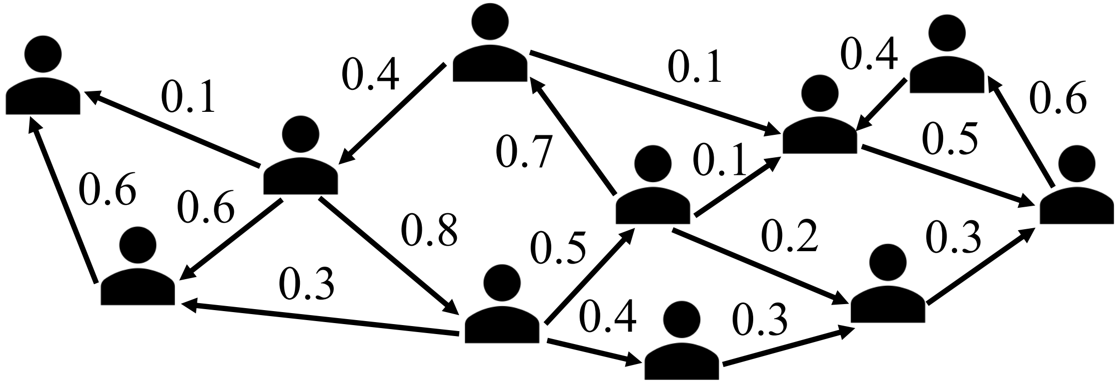
- Problem: how to pick *k* individuals for the merchant such that the eventual influence spread is maximized?

EXISTING SOLUTIONS FOR IM

- Adopt a stochastic model M to simulate the influence propagation
  - i.e., under what condition will a user be influenced
- Example: Independent Cascade (IC) model



IC model:
u independently influences v
with probability $p_{u,v}$



social network

EXISTING SOLUTIONS FOR IM

- Adopt a stochastic model M to simulate the propagation
  - i.e., under what condition will a user be influenced
- Generate samples of the social network based on M
- Identify k influencers by using
  - the samples to estimate the spread
  - and the greedy algorithm over samples

LIMITATION #1 OF IM

- Conventional IM
  - considers the cost factor
  - ignores individual's capacity for spending efforts on consuming the promoting content
- User's capacity
  - is crucial as it determines the adoption of the product
  - is limited on online platforms
    - e.g., while playing e-games with friends

LIMITATION #2 OF IM

- Conventional IM
  - assumes influencers unconditionally become initial adopters
- Observation from real-world scenarios
  - influencers tend to be the friends of initial adopters

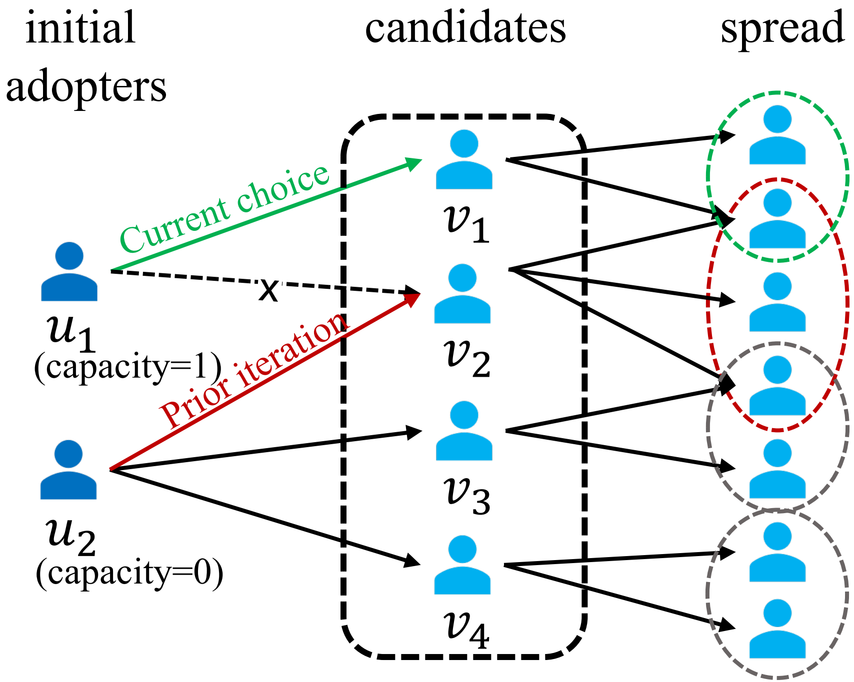
source: https://zootopia.fandom.com/wiki

CAPACITY CONSTRAINED IM (CIM)

- Input
  - social network G and stochastic model M
  - d initial adopters and capacity constant k
- Output
  - k influential friends (seeds) for each of d initial adopters
- Objective
  - maximize the spread of the set of all selected seeds
- CIM is NP-hard

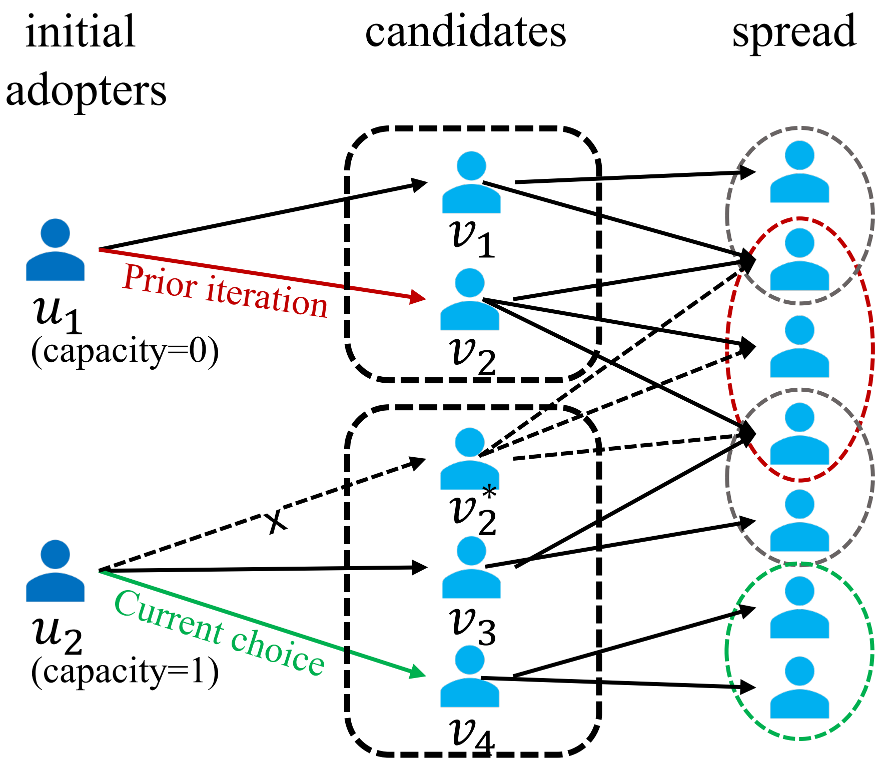
IDEA OF GREEDY ALGORITHMS

- MG-Greedy: select a user v from candidates as the next if
  - adding v to current seed set yields the largest lift
  - and, existing a v's friend in initial adopters that remains capacity
- $\frac{1}{2}$-approximate if known spread



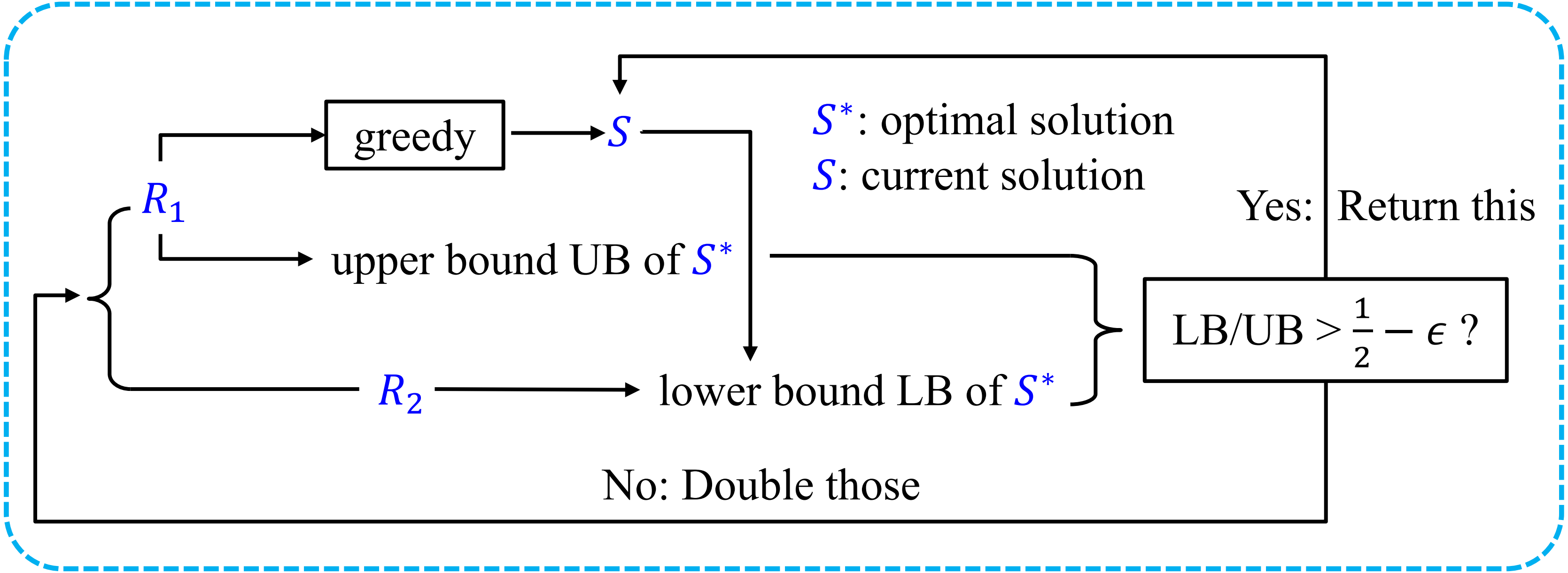
IDEA OF GREEDY ALGORITHMS

- RR-Greedy: select in a round-robin manner
  - choose an initial adopter u remaining capacity
  - select from u's candidates
  - add v to current seed set if yielding the largest lift
- $\geq \frac{1}{2}$-approximate if known spread



SCALABLE IMPLEMENTATIONS

- Borrow OPIM-C framework in SIGMOD’18
- Generate two equal-size sets of samples: R_1 and R_2



SCALABLE IMPLEMENTATIONS

- Borrow OPIM-C framework in SIGMOD'18
- Redesign each parameter by rigorous theoretical analysis
- Implementations:
  - MG-OPIM: greedy = MG-Greedy
  - RR-OPIM: greedy = RR-Greedy
  - RR-OPIM+: RR-OPIM with an optimized UB
- Result: $(\frac{1}{2} - \epsilon)$-approximate in near-linear running time

DATASETS

- Various public datasets
  - *DNC*
  - *Blog*
  - *Twitch*
  - *Orkut*
  - *Twitter*
- Tencent game dataset
  - *TXG* (with ground-truth spread)

Dataset statistics ($K = 10^3, M = 10^6, B = 10^9$)

Name	**#nodes (n)**	**#edges (m)**
DNC	$0.9K$	$24.2K$
Blog	$10.3K$	$668.0K$
Twitch	$168.1K$	$13.6M$
TXG	$243.4K$	$11.8M$
Orkut	$3.1M$	$234.2M$
Twitter	$41.7M$	$2.9B$

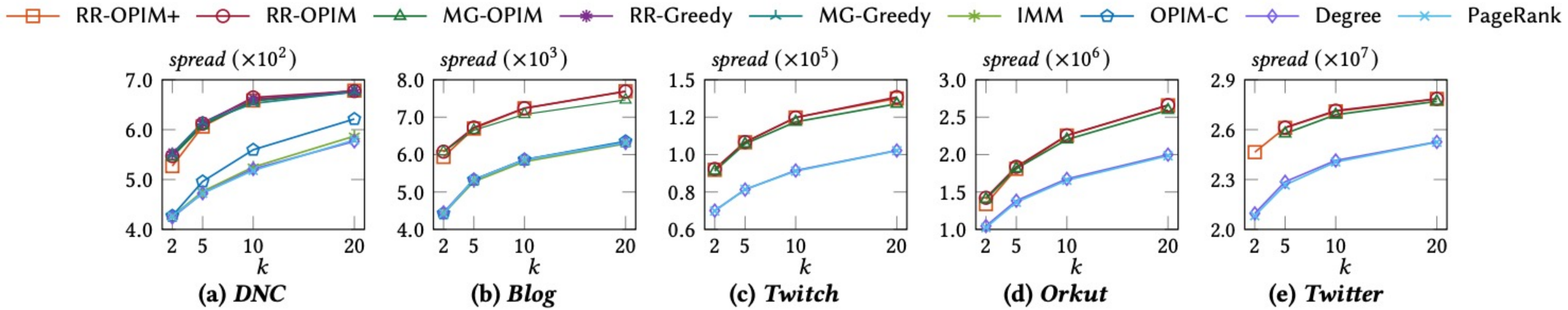
APPROACHES

- Local competitors: independently select k friends for each initial adopter
  - based on a heuristic score: Degree, PageRank
  - based on a SOTA IM solver: IMM, OPIM-C
- Greedy solutions
  - MG-Greedy, RR-Greedy
- Scalable implementations
  - MG-OPIM, RR-OPIM, RR-OPIM+

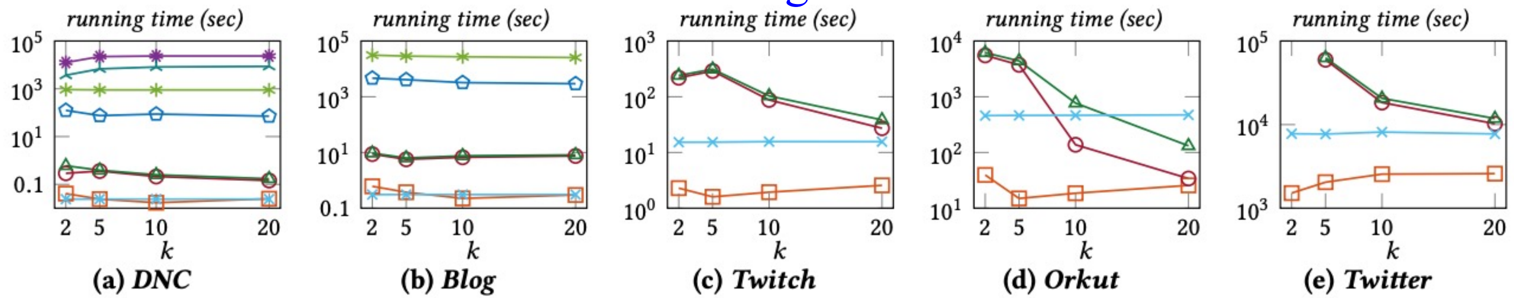
PERFORMANCE ON PUBLIC DATASETS

- Final solution RR-OPIM+ outperforms all solutions

Large spread



Low running time



ACTUAL SPREAD ON TENCENT GAME

- Final solution RR-OPIM+ outperforms
  - all solutions on TXG during offline evaluation

Solution	RR-OPIM+	MG-OPIM	RR-OPIM	Degree	PageRank
Spread	**1,632**	1,625	1,609	1,488	1,471

  - control group on a battle royale game during online deployment

Solution	Treatment	Control
Spread	**60.69K**	58.28K

SUMMARY

- CIM: a new problem for real-world viral marketing
- MG/RR-Greedy: effective greedy algorithms for CIM
- RR-OPIM+: scalable greedy implementation for CIM
- Have been deployed on Tencent gaming platforms

THANK YOU!