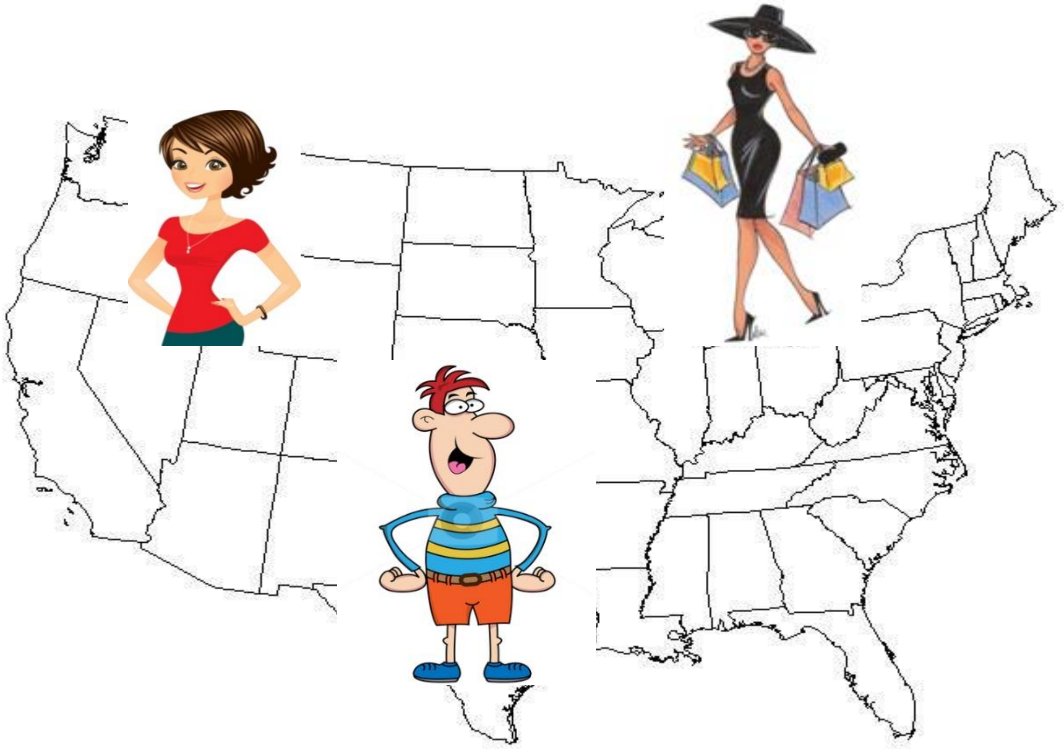
Boosting Information Spread: An Algorithmic Approach

Yishi Lin (The Chinese University of Hong Kong)
Wei Chen (Microsoft Research)
John C.S. Lui (The Chinese University of Hong Kong)

Background: Viral Marketing

➢ **Assumption**: the *word-of-mouth* effect



Whom to give free samples to maximize the purchase of the product ?

Influence Maximization
Select k seed nodes so to maximize the expected spread of influence.

Motivation

Some marketing strategies **boost** customers so that they are

- More likely to be influenced by friends, or
- More likely to influence their friends

Examples

- Customer incentive programs
- Social media advertising
- Referral marketing

Motivation: Complement the Classical IM

Boosting a user **vs.** Turning a user into an initial adopter

(e.g., coupon) (e.g., free products)

Our study: How to select users to "boost"?

IM studies: How to identify influential initial adopters?

Companies have **more flexibility** in determining where to allocate their marketing budgets

Main Contributions

Influence boosting model

- the idea of *boosting* + the *Independent Cascade* model

k-boosting problem

- NP-hard
- Non-submodular objective function

Approximation algorithms

- `PRR-Boost` / `PRR-Boost-LB`
- Approximation guarantee
- Practical efficiency

Influence Boosting Model

Social network $G = (V, E)$

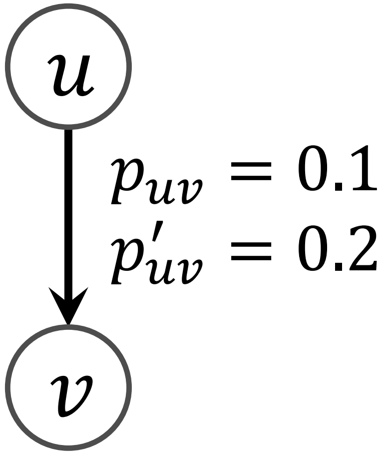
- Seed users: $\boldsymbol{S} \subseteq \boldsymbol{V}$
- Boosted users: $\boldsymbol{B} \subseteq \boldsymbol{V}$

Influence propagation

- Each "newly-influenced" node u attempts to influence its neighbor v
- If v is boosted ($v \in B$), u succeeds w.p. $p'_{uv} \geq p_{uv}$
- Otherwise, u succeeds w.p. p_{uv}

Notations

- $\sigma_S(B)$: boosted influence spread (expected influence spread)
- $\Delta_S(B) = \sigma_S(B) - \sigma_S(\emptyset)$: boost of influence spread of B



$S = \{u\}$

$B = \{v\}$

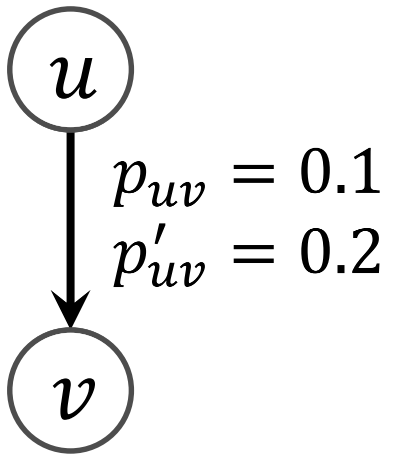
$\sigma_S(B) = 1.2$

$\Delta_S(B) = 0.1$

k-Boosting Problem

Problem

- Given graph G, budget k, seeds S
- Select a set B of k nodes so that the boost of influence spread of is maximized.



The k-boosting problem is **NP hard**. Computing $\Delta_S(B)$ is **#P hard**.

☹

The boost of influence $\Delta_S(B)$ is neither submodular nor supermodular!

☹☹☹

$S = \{u\}, k = 1$

$B = ?$

Our Solution: PRR-Boost/PRR-Boost-LB

Potentially Reverse Reachable Graphs (PRR-graphs)

- Estimate the boost of influence spread and its lower bound (for SA)

Sandwich Approximation (SA) strategy [1]

- Provides approximation guarantee
- Deals with the non-submodularity of objective function

State-of-the-art IM techniques [2][3]

- Sample PRR-graphs

[1] W. Lu, W. Chen, and L. V. S. Lakshmanan, "From competition to complementarity: Comparative influence diffusion and maximization," *VLDB Endow.*, vol. 9, no. 2, 2015.
[2] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization in near-linear time: A martingale approach," in *SIGMOD*, 2015.
[3] H. T. Nguyen, T. N. Dinh, and M. T. Thai, "Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks," in *SIGMOD*, 2016.

PRR-Boost: Estimating the boost of influence

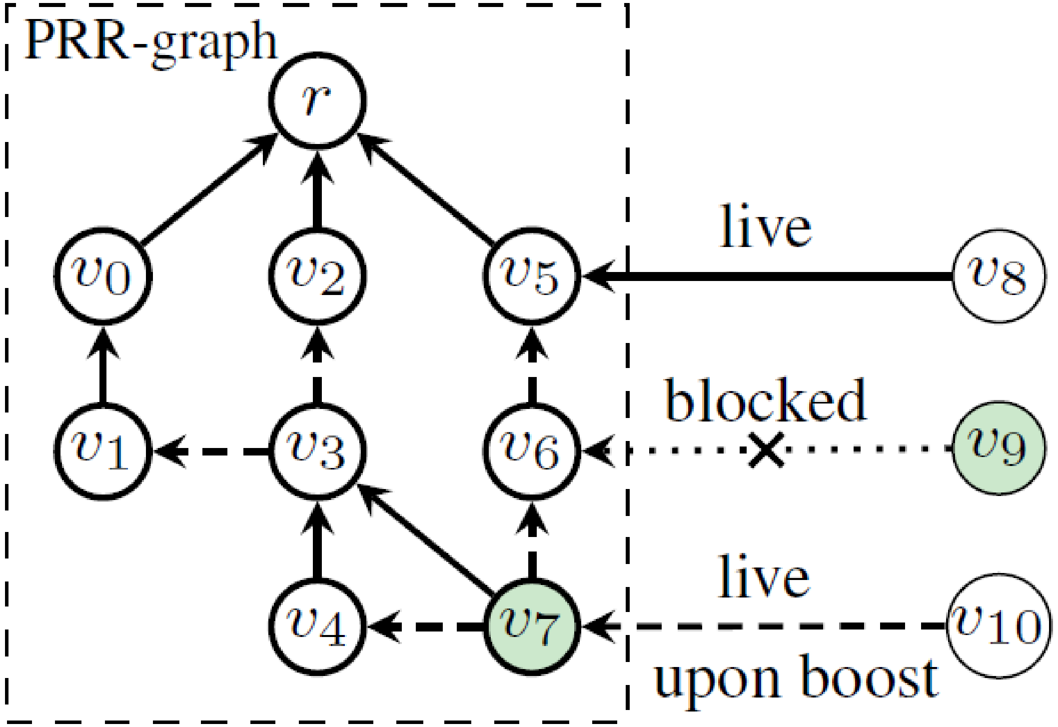
Question

How to estimate the boost of influence (the objective function)?

PRR-Boost: Estimating the boost of influence

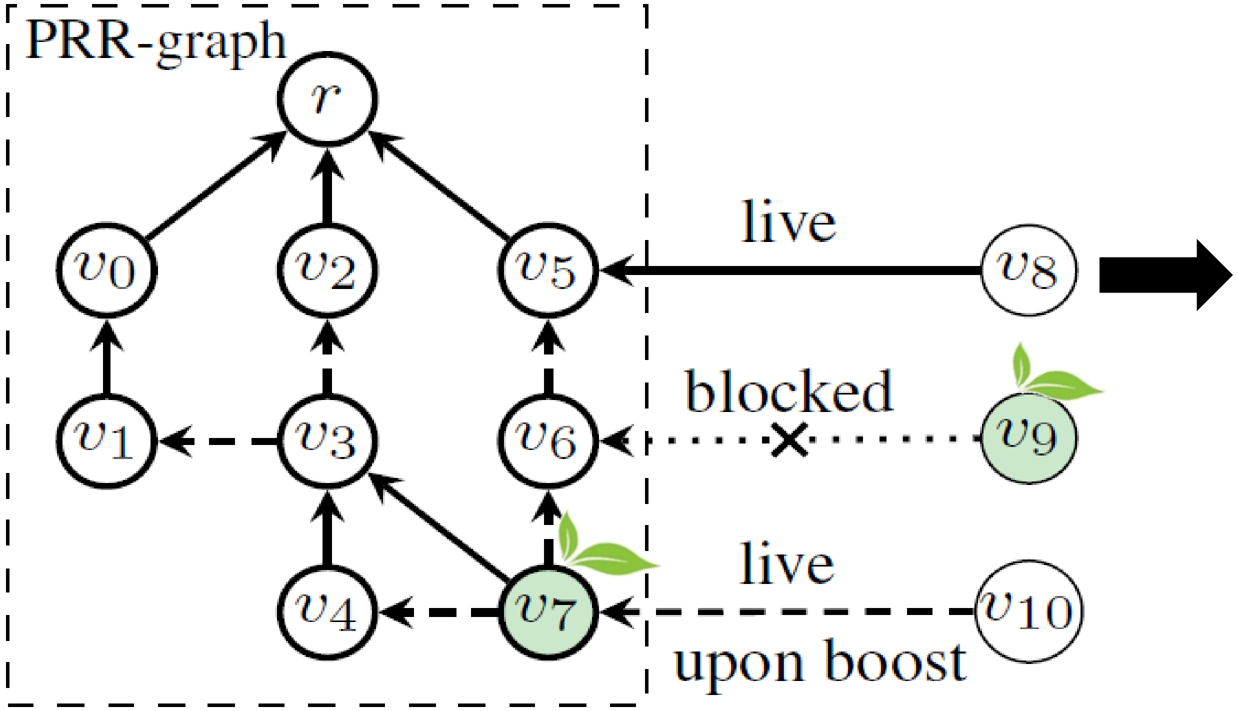
Potentially Reverse Reachable Graph (PRR-Graph)

- Random target node r
- Random "edge status"
- Seed nodes
- Non-blocked paths from seeds to r



A sampled influence propagation process

PRR-Boost: Estimating the boost of influence



"Score" of B: $f_R(B) = \mathbb{I}(\text{influence } 0 \rightarrow 1)$

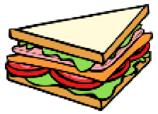
$\mathbb{E}$[**"Score" of *B*** in a random R] $=$ Pr[a random node is inactive w/o boosting and active upon boosting B]

Boost of $\boldsymbol{B}$ = $\boldsymbol{n} \cdot \mathbb{E}_R[\boldsymbol{f}_R(\boldsymbol{B})]$

Boost of $\boldsymbol{B} \approx \boldsymbol{n} \cdot \frac{\sum_R f_R(B)}{|R \text{ samples}|}$

PRR-Boost/PRR-Boost-LB: Algorithm Design

PRR-Boost $(\boldsymbol{G}, \boldsymbol{S}, \boldsymbol{k}, \boldsymbol{\epsilon}, \boldsymbol{\ell})$

1. $\ell' \leftarrow \ell \cdot (1 + \log 3 \,/\, \log n)$
2. $\mathcal{R} \leftarrow \text{SamplingLB}(G, S, k, \epsilon, \ell')$ // sampling PRR-graphs
3. $B_\mu \leftarrow \text{NodeSelectionLB}(\mathcal{R}, k)$ // maximize the lower bound of boost
4. $B_\Delta \leftarrow \text{NodeSelection}(\mathcal{R}, k)$ // maximize the boost of influence
5. $B_{sa} \leftarrow \text{argmax}_{B \in \{B_\Delta, B_\mu\}} \text{Estimation of } \Delta_S(B)$ // 
6. Return B_{sa}

PRR-Boost-LB returns $\boldsymbol{B_\mu}$

Experiments: Settings

Datasets

- Real social networks & learned influence probabilities [4]
- Boosted influence probability: $p'_{uv} = 1 - (1 - p_{uv})^{\beta}, \beta = 2$

Table 1: Statistics of datasets and seeds (all directed)

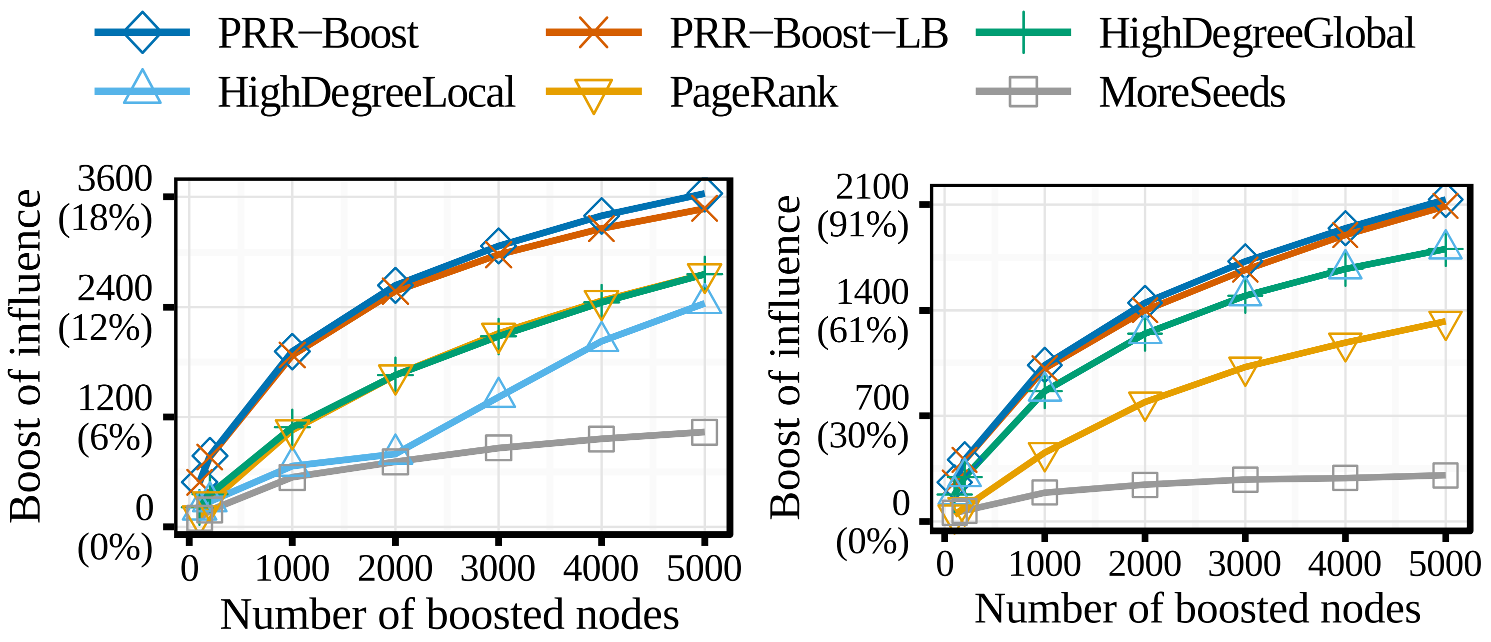
Description	Digg	Flixster	Twitter	Flickr
number of nodes (n)	28 K	96 K	323 K	1.45 M
number of edges (m)	200 K	485 K	2.14 M	2.15 M
average influence probability	0.239	0.228	0.608	0.013
influence of 50 influential seeds	2.5 K	20.4 K	85.3 K	2.3 K
influence of 500 random seeds	1.8 K	12.5 K	61.8 K	0.8 K

[4] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, "Learning influence probabilities in social networks," in *WSDM*, 2010.

Experiments: Settings

- **Datasets**
  - Real social networks & learned influence probabilities [4]
  - Boosted influence: $p'_{uv} = 1 - (1 - p_{uv})^{\beta}, \beta = 2$
- **Settings**
  - Parallelization with OpenMP and executed using 8 threads
  - A Linux machine with an Intel Xeon E5620@2.4GHz CPU and 30GB memory

Quality of Solution (50 influential seeds)



(a) Flixster

$n = 96K, m = 485K$
$\bar{p} = 0.228, \sigma(S) = 20.4K$

(b) Flickr

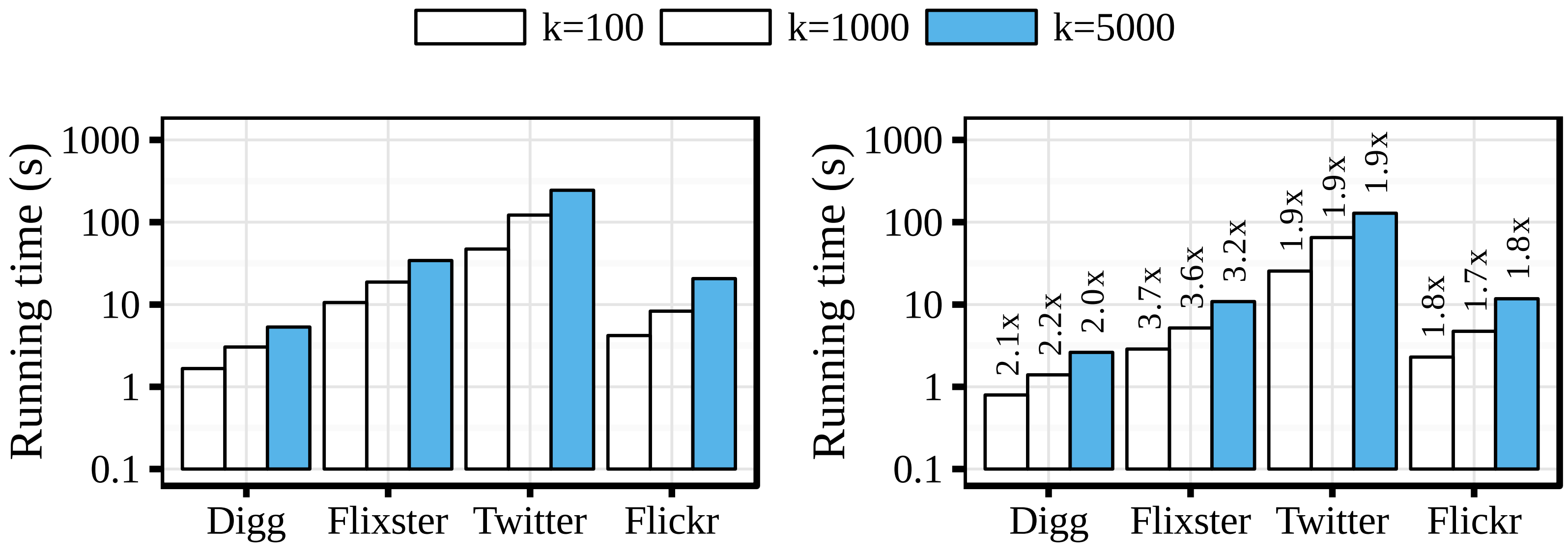
$n = 1.45M, m = 2.15M$
$\bar{p} = 0.013, \sigma(S) = 2{,}3K$

PRR-Boost
- Best quality

PRR-Boost-LB
- Slightly lower but comparable quality

Both of them
- significantly outperform other baselines

Running Time (50 influential seeds)



(a) PRR-Boost

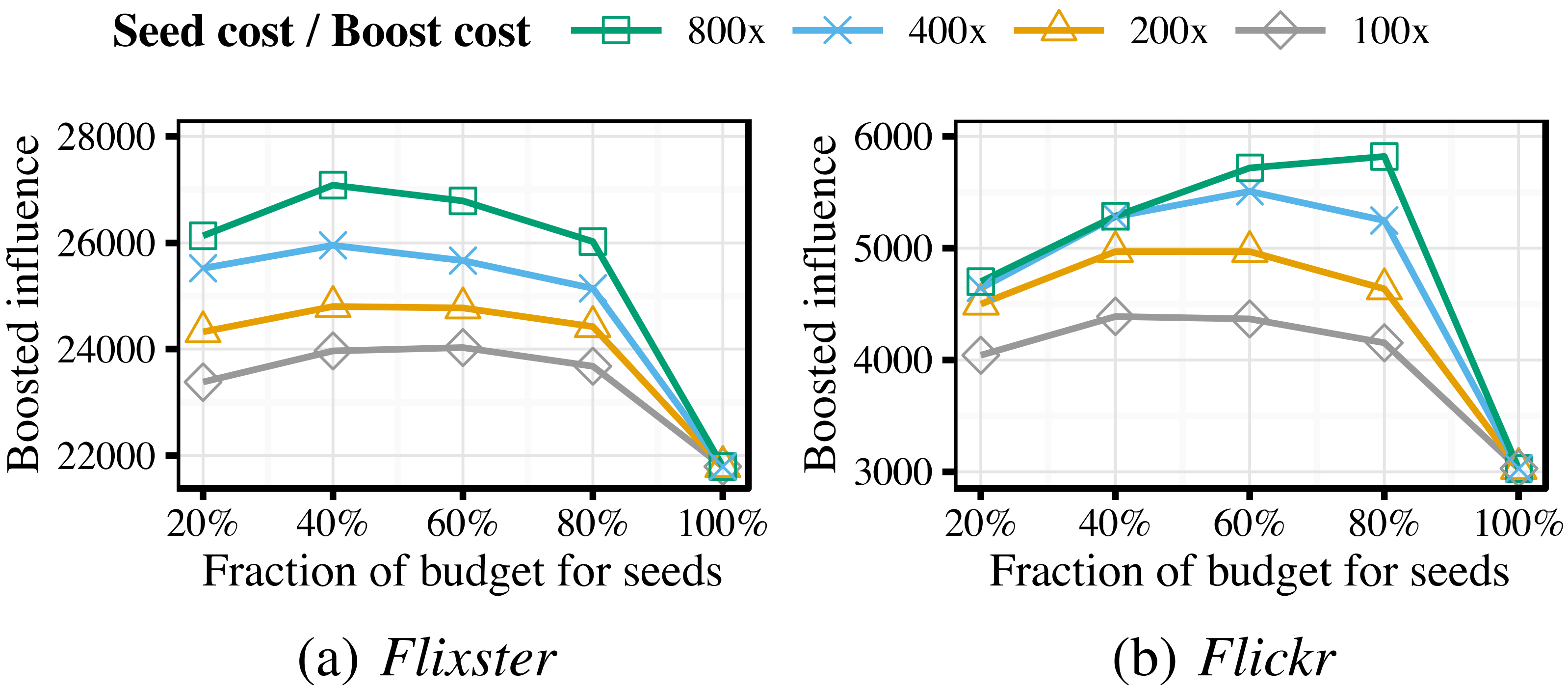
(b) PRR-Boost-LB

Time increases with *k*
- # of PRR-graphs ↑

PRR-Boost
- Efficient

PRR-Boost-LB
- Faster
- Effective & Efficient

More Experiments: Budget Allocation



Steps

1. Select seeds
2. Select boost users

Take away messages

1. Our study complements the IM studies.
2. Budget allocation problem!

Setting: We assume that we can target 100 users as seed nodes with all the budget.

Conclusion

The k-boosting problem

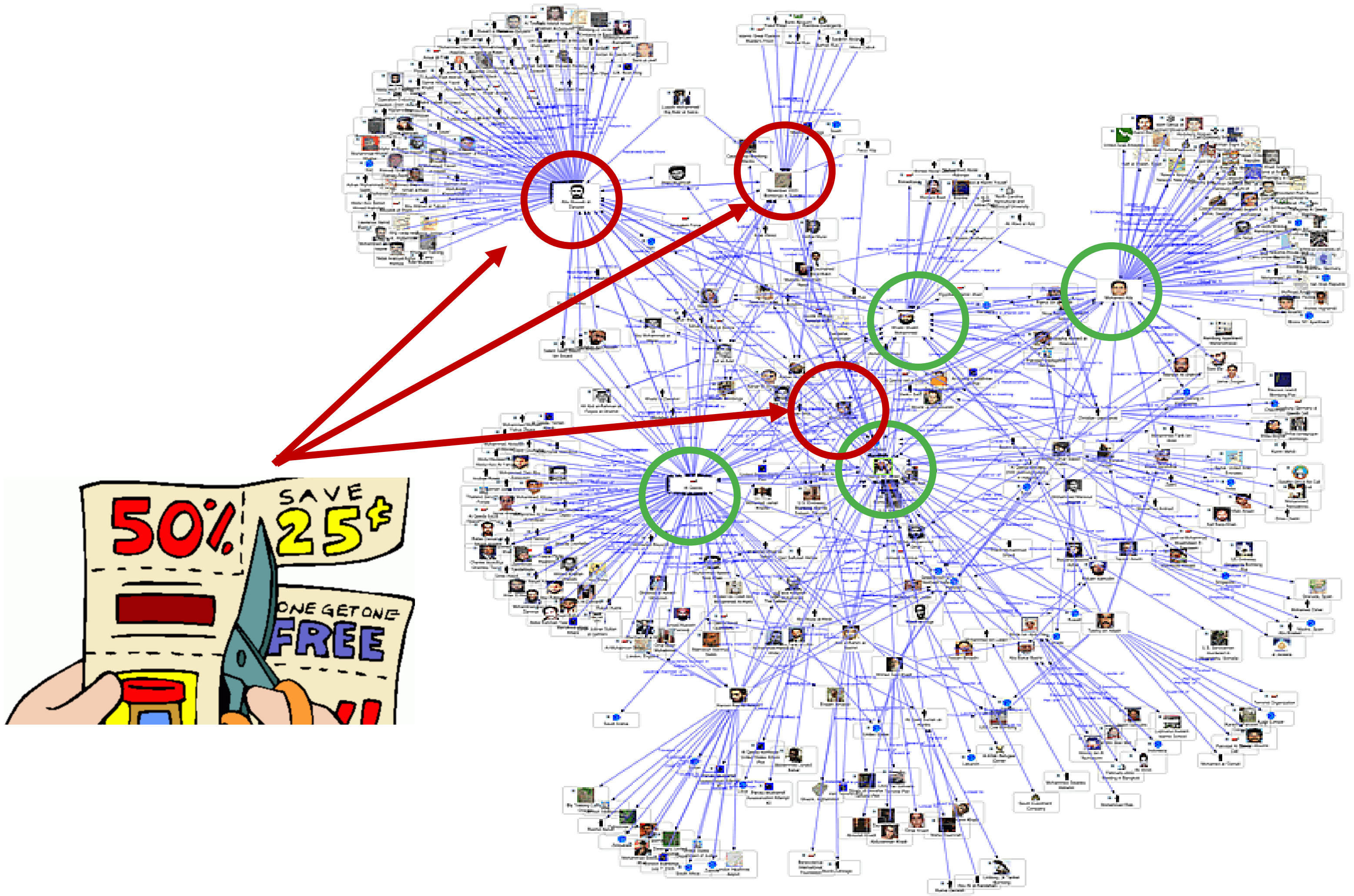
- Influence boosting model
- NP-hard & non-submodular objective function

Approximation Algorithm

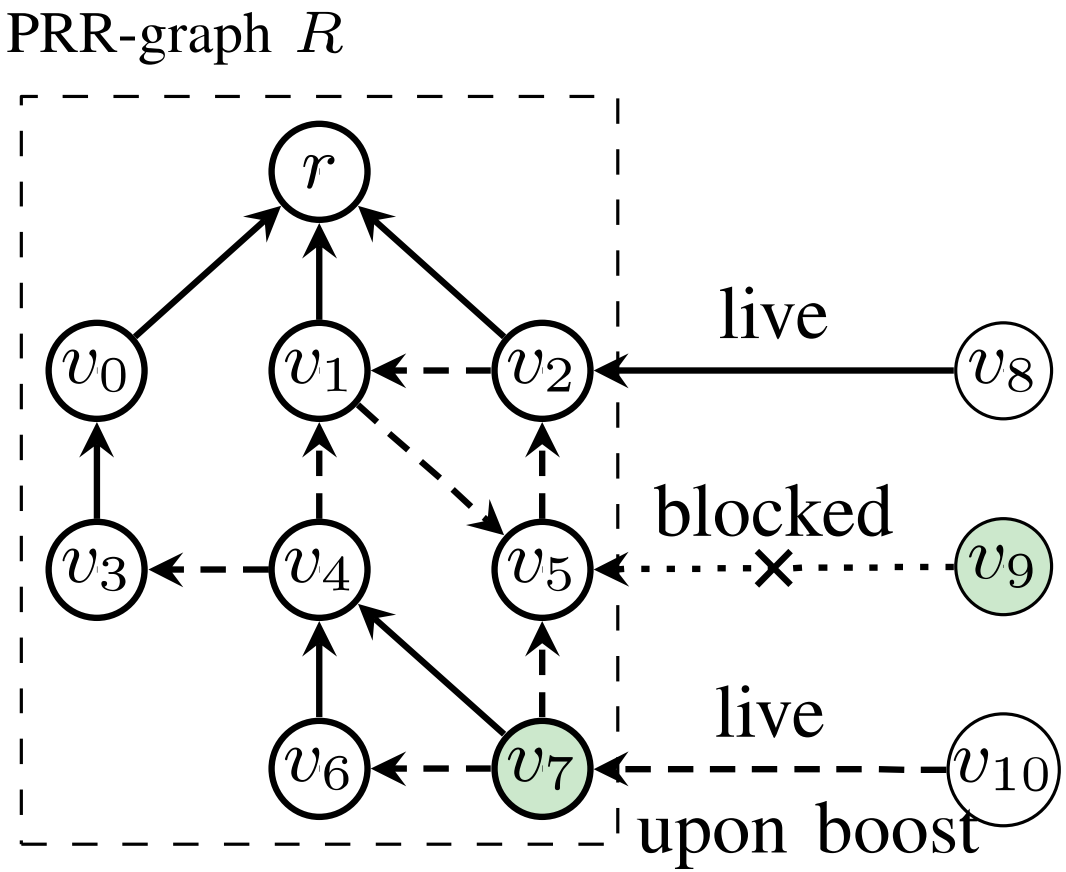
- `PRR-Boost/PRR-Boost-LB` = PRR-graphs + other techniques
- Approximation ratio: $\left(1 - 1/e - \epsilon\right) \cdot \frac{\mu\left(B^{OPT}\right)}{\Delta_S\left(B^{OPT}\right)}$
- Practical efficiency:
  - PRR-Boost: $O\left(\frac{EPT}{OPT_\mu} \cdot k \cdot (k + \ell) \cdot (n + m) \log n \cdot \epsilon^{-2}\right)$
  - PRR-Boos-LB: $O\left(\frac{EPT}{OPT_\mu} \cdot (k + \ell) \cdot (n + m) \log n \cdot \epsilon^{-2}\right)$

Thank you!

Motivation

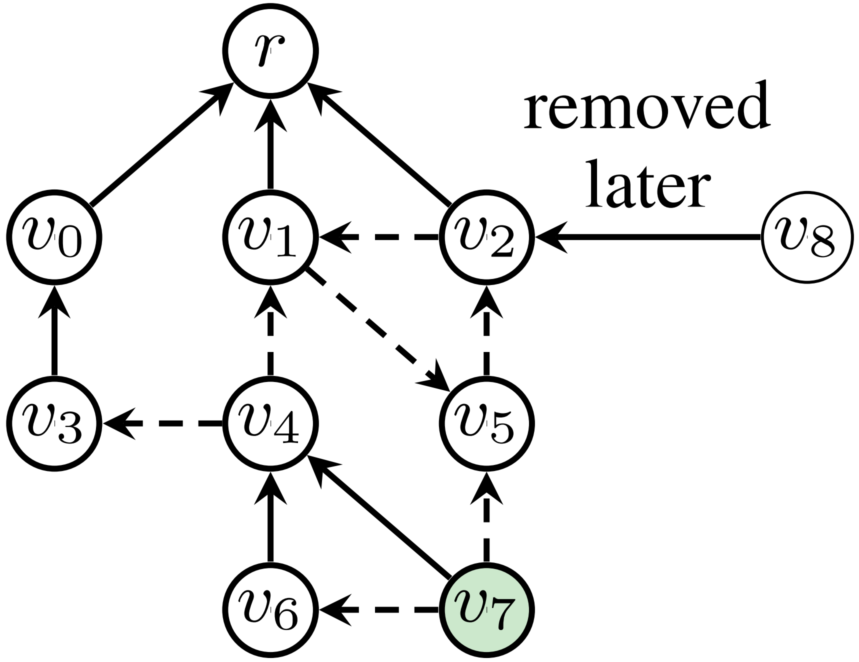


Potentially Reverse Reachable Graphs: Definition

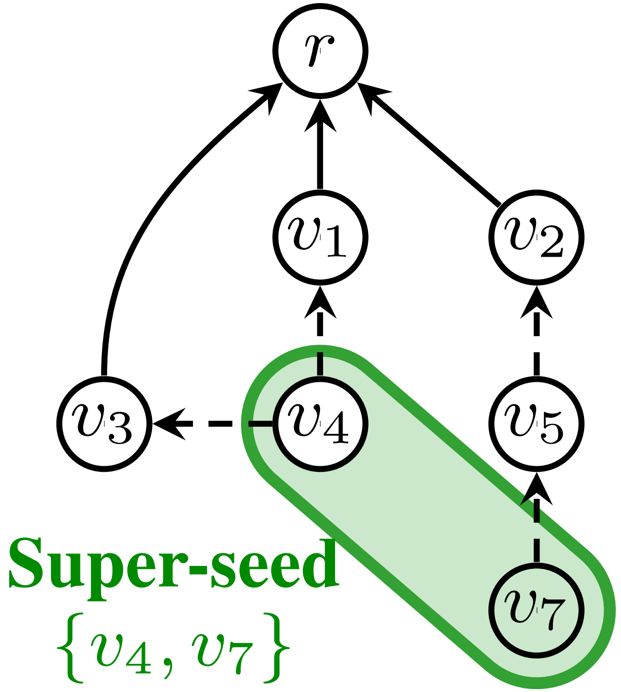


- Estimating the boost
  - $f_R(\emptyset) = 0$
  - $f_R(\{v_1\}) = 1$
  - $f_R(\{v_3\}) = 1$
  - $f_R(\{v_2, v_5\}) = 1$
- Critical nodes
  - $C_R = \{v_1, v_3\}$
- Estimating the lower bound
  - $\mu(B) = \mathbb{I}(B \cap C_R \neq \emptyset)$

Potentially Reverse Reachable Graphs: Generation



(a) Results of phase I

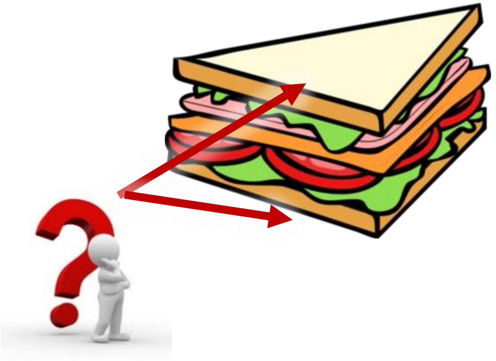


(b) Results of phase II

PRR-Boost: Sandwich Approximation

Goal: to tackle the non-submodularity of $\Delta_S(\cdot)$

Sandwich Approximation (SA) strategy



Submodular UB: $\mu(B)$

$\Delta_S(B)$

Submodular LB: $\nu(B)$

$$\xrightarrow{\text{greedy}} \left.\begin{matrix} B_\mu \\ B_\Delta \\ B_\nu \end{matrix}\right\} B_{sa} = \operatorname{argmax}_{B \in \{B_\Delta, B_\mu, B_\nu\}} \Delta_S(B)$$

- Theoretical guarantee:

$$\Delta_S(B_{sa}) \geq \max\left\{\frac{\Delta_S(B_\nu)}{\nu(B_\nu)}, \frac{\mu(B^{OPT})}{\Delta_S(B^{OPT})}\right\} \cdot \left(1 - \frac{1}{e} - \epsilon\right) \cdot OPT$$

Remarks

- Proposed by Lu, Wei et al. in “From competition to complementarity: comparative influence diffusion and maximization.” (*VLDB’15)*

PRR-Boost: Main Results

PRR-Boost:

PRR-Boost-LB: same bound, much faster

Practical performance guarantee

- $\Delta_S(B_{sa}) \geq \left(1 - \frac{1}{e} - \epsilon\right) \cdot \frac{\mu(B^{OPT})}{\Delta_S(B^{OPT})} \cdot OPT$, w.p. at least $1 - \mathrm{n}^{-\ell}$
- The approximate ratio is good if the lower bound is tight
- Experiments show that the lower bound is tight

Practical efficiency

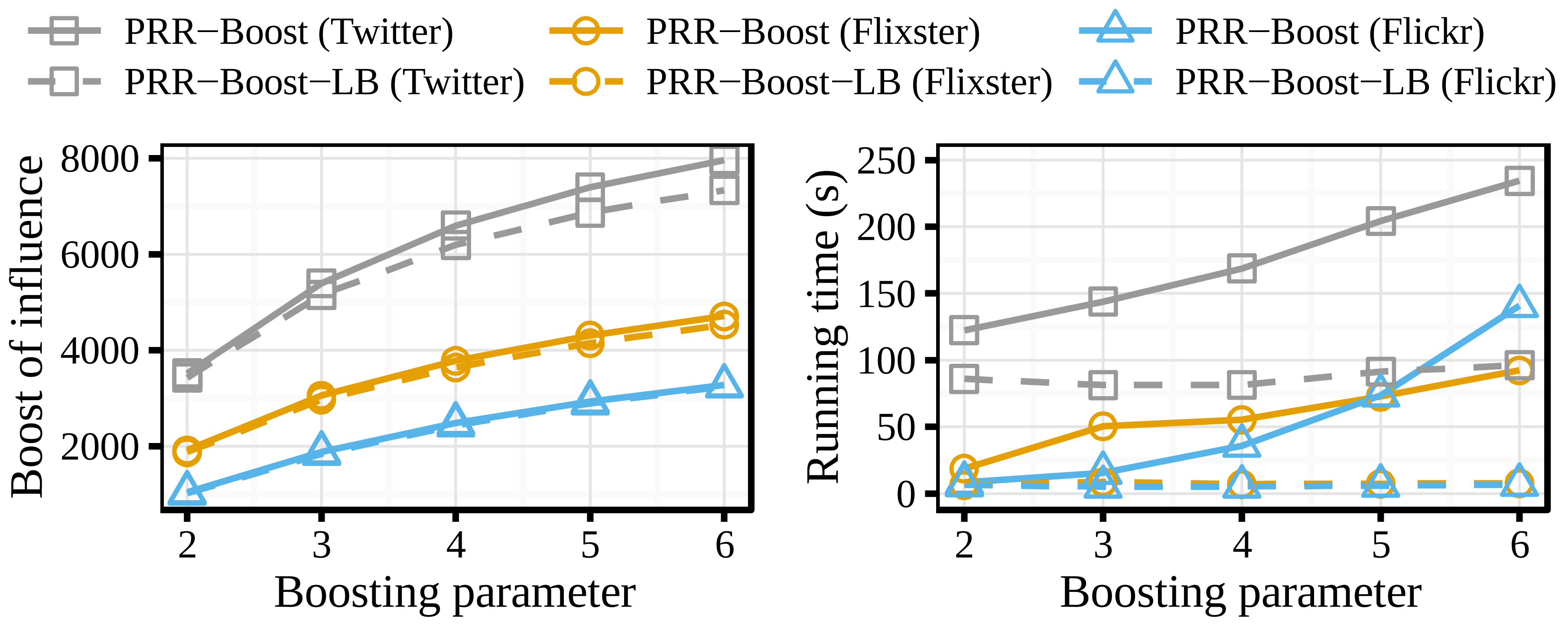
- $O\left(\frac{EPT}{OPT_\mu} \cdot k \cdot (k + \ell) \cdot (n + m) \log n \cdot \epsilon^{-2}\right)$
- EPT: the expected time to construct a PRR-graph
- OPT_μ: the optimum solution for maximizing μ

Experiments: Compression Ratio

Table 2: Memory usage and compression ratio (influential seeds). Numbers in parentheses are additional memory usage for boostable PRR-graphs.

k	Dataset	PRR-Boost		PRR-Boost-LB
		Compression Ratio	Memory (GB)	Memory (GB)
100	Digg	1810.32 / 2.41 = 751.79	0.07 (0.01)	0.06 (0.00)
	Flixster	3254.91 / 3.67 = 886.90	0.23 (0.05)	0.19 (0.01)
	Twitter	14343.31 / 4.62 = 3104.61	0.74 (0.07)	0.69 (0.02)
	Flickr	189.61 / 6.86 = 27.66	0.54 (0.07)	0.48 (0.01)
5000	Digg	1821.21 / 2.41 = 755.06	0.09 (0.03)	0.07 (0.01)
	Flixster	3255.42 / 3.67 = 886.07	0.32 (0.14)	0.21 (0.03)
	Twitter	14420.47 / 4.61 = 3125.37	0.89 (0.22)	0.73 (0.06)
	Flickr	189.08 / 6.84 = 27.64	0.65 (0.18)	0.50 (0.03)

Experiments: Effects of the Boosting Parameter



(a) Boost of *influence* (b) Running time

Fig. 7: Effects of the boosting parameter (influential seeds, k = 1000).

Experiments: Approx. Ratio $(1 - 1/e - \epsilon) \cdot \frac{\mu(B^{OPT})}{\Delta_S(B^{OPT})}$

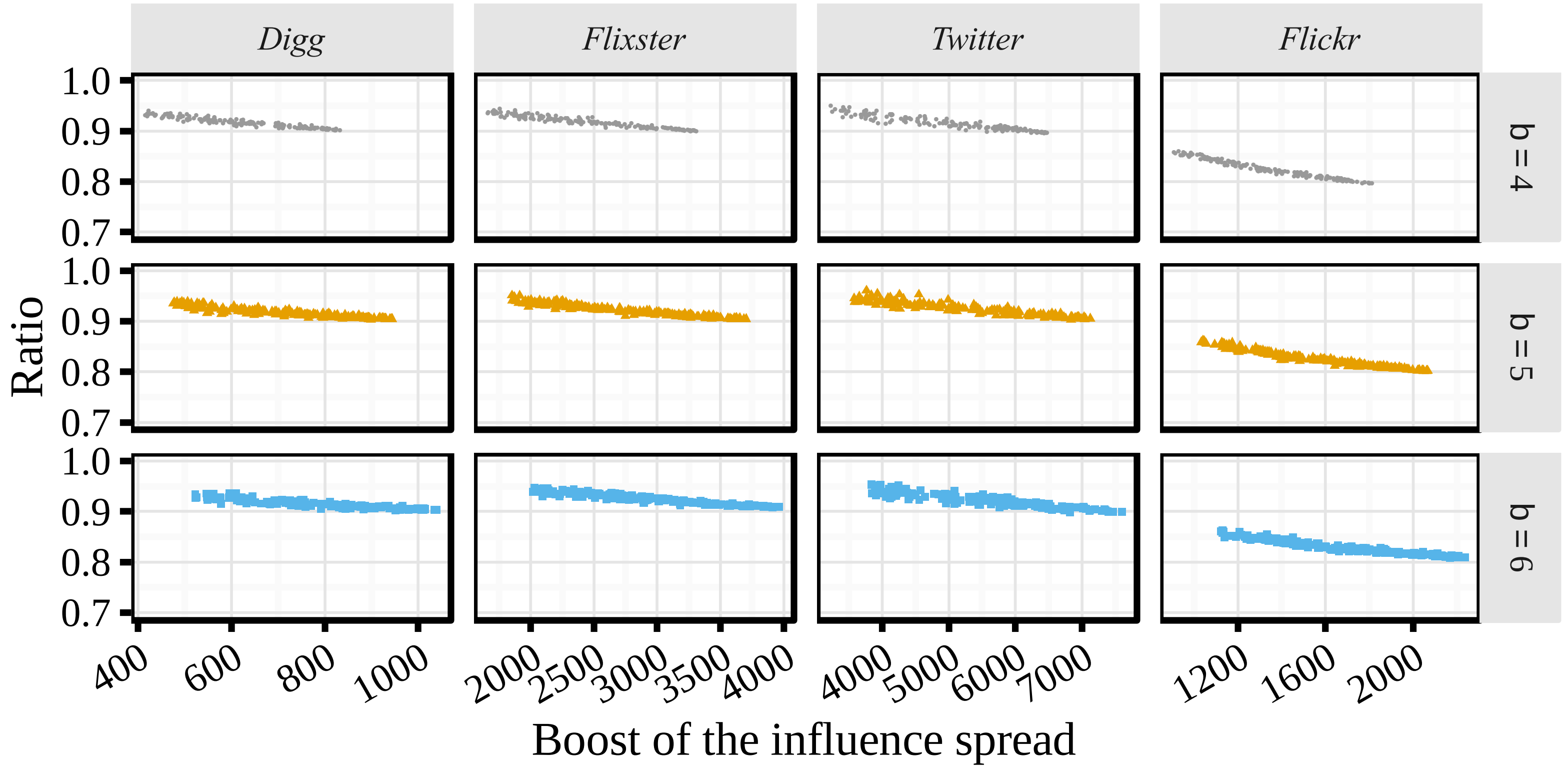


Fig. 8: Sandwich Approximation with varying boosting parameter: $\frac{\mu(B)}{\Delta_S(B)}$ (influential seeds, k = 1000).