Tracking Influential Nodes in Time-Decaying Dynamic Interaction Networks

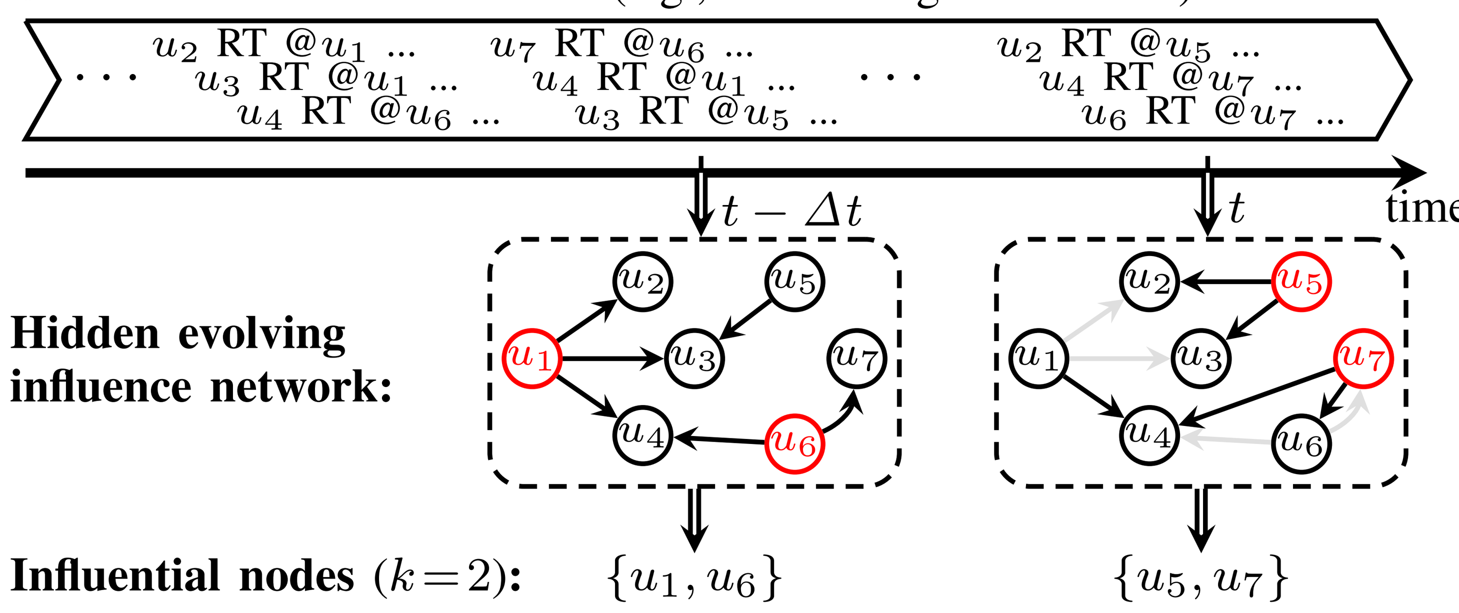
Junzhou Zhao[1] Shuo Shang[2] Pinghui Wang[3] John C.S. Lui[4] Xiangliang Zhang[1]

[1]KAUST [2]IIAI [3]XJTU [4]CUHK

Background & Motivation

- Word-of-mouth effect: one person can influence another in a social network.
- Influence maximization: selecting k seed nodes from a network to maximize the influence spread in a network.
- Dynamic influence challenge:
  - *Network structure* may change over time: e.g., 9% of all connections in Twitter are changing every month.
  - *Influence probability* on each edge may change over time: e.g., relationship strength becomes weaker/stronger.
- As a result, today's influential nodes may not be still influential tomorrow.

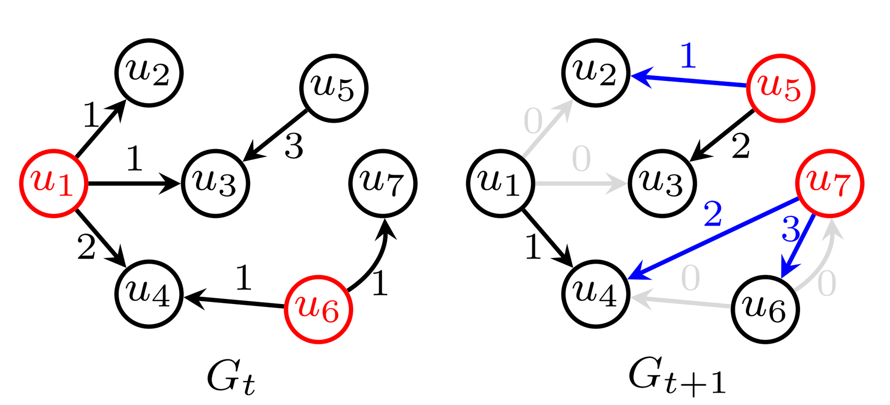
Observed node interactions: (e.g., re-tweeting interactions)



TDN Model

- We propose a time-decaying dynamic interaction network (TDN) model, which is a general way to model node interaction streams in a network.
- TDN is represented as a continuously evolving dynamic graph $G_t = (V_t, E_t)$.
- Each edge $e \triangleq (u, v, \tau)$ represents a *node interaction*: user u influenced user v at time τ, e.g., v retweeted u's tweet in Twitter.
- Each edge e can survive for l_e time units. $l_e > 0$ is referred to as e's *lifetime*.
  - Lifetime decreases over time.
  - If lifetime becomes zero, the edge expires and no longer exists.
  - If edges of a node all expire, the node expires.
- At any time t, all survival edges E_t and nodes V_t form a graph G_t.

$e \triangleq (u, v, \tau)$	$l_\tau(e)$
$e_1 = (u_1, u_2, t)$	1
$e_2 = (u_1, u_3, t)$	1
$e_3 = (u_1, u_4, t)$	2
$e_4 = (u_5, u_3, t)$	3
$e_5 = (u_6, u_4, t)$	1
$e_6 = (u_6, u_7, t)$	1
$e_7 = (u_5, u_2, t+1)$	1
$e_8 = (u_7, u_4, t+1)$	2
$e_9 = (u_7, u_6, t+1)$	3



Tracking Influential Nodes over TDNs

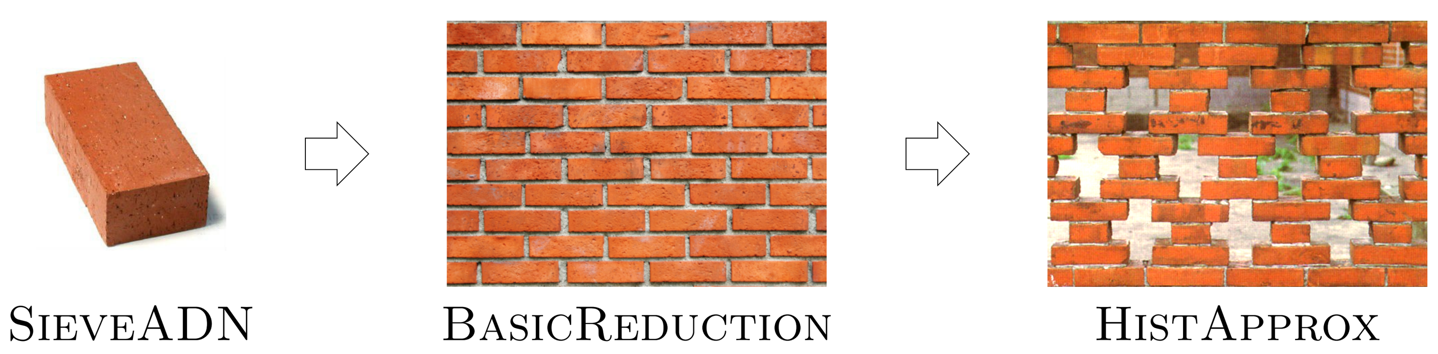
- Influence spread of a set of nodes S in G_t:

$$f_t(S) \triangleq |\{v\colon \text{node } v \text{ is reachable from } S \text{ in } G_t\}|$$

- Influential Nodes Tracking Problem:
  - Given TDN G_t evolving over time t, and budget k,
  - Want to find $S_t^* \subseteq V_t$ with $|S_t^*| \le k$ s.t.

$$S_t^* = \arg\max_{S \subseteq V_t \wedge |S| \le k} f_t(S).$$

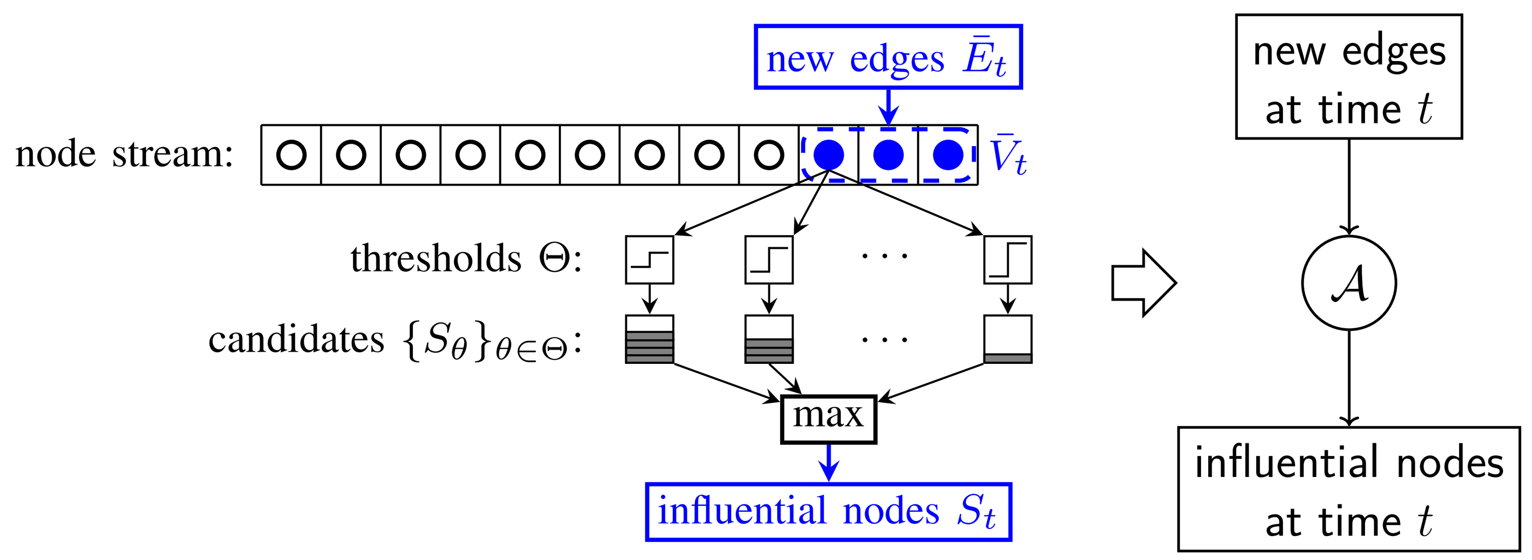
Overview of Our Algorithms



Algorithm	Update Time	Memory	Approximate Ratio
SieveADN	$O(b\gamma\epsilon^{-1}\log k)$	$O(k\epsilon^{-1}\log k)$	$1/2-\epsilon$
BasicReduction	$O(Lb\gamma\epsilon^{-1}\log k)$	$O(Lk\epsilon^{-1}\log k)$	$1/2-\epsilon$
HistApprox	$O(b\gamma\epsilon^{-2}\log^2 k)$	$O(k\epsilon^{-2}\log^2 k)$	$1/3-\epsilon$
Greedy	$O(k\|V_t\|\gamma)$	$O(\|V_t\|)$	$1-e^{-1}$

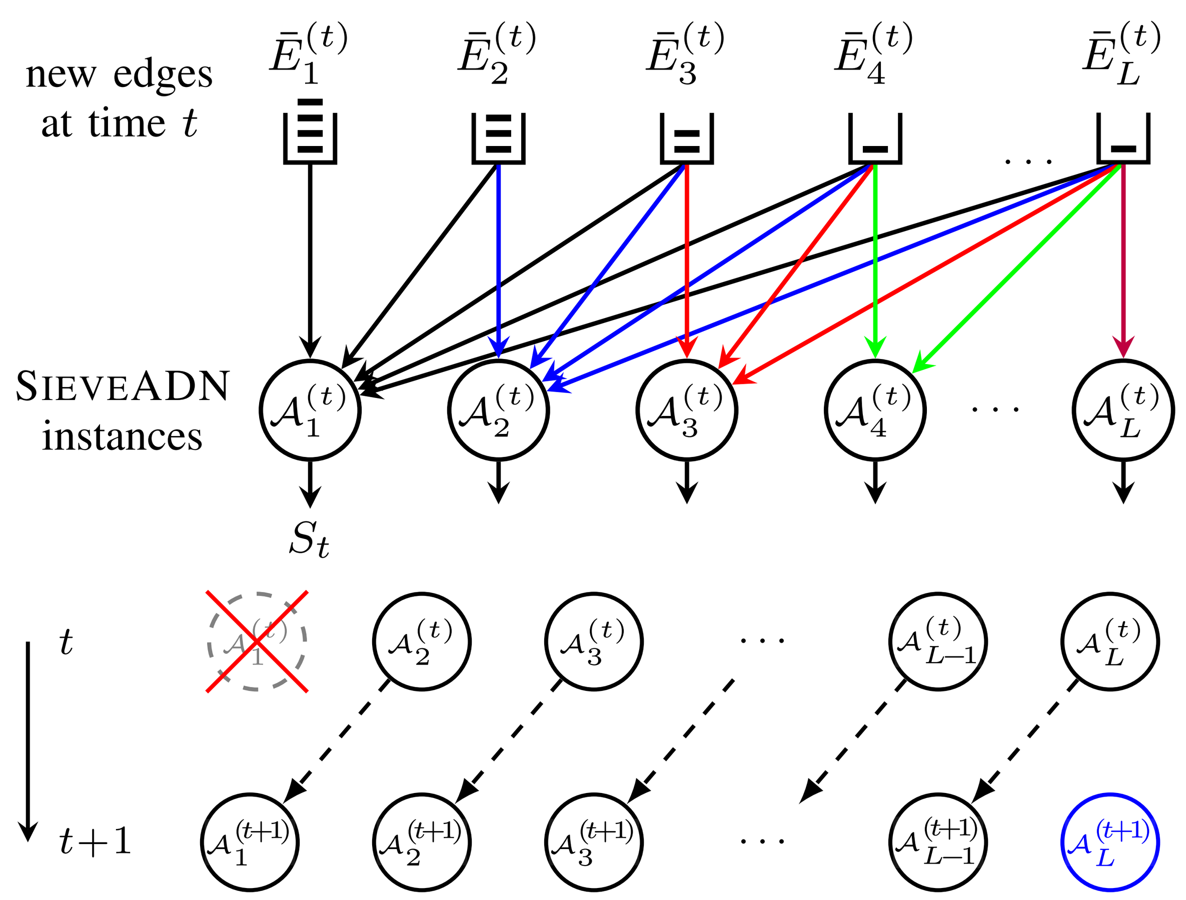
SieveADN

- Addition-only dynamic networks (ADNs): every edge has an infinite lifetime.
- SieveADN is adapted from SieveStreaming [1].



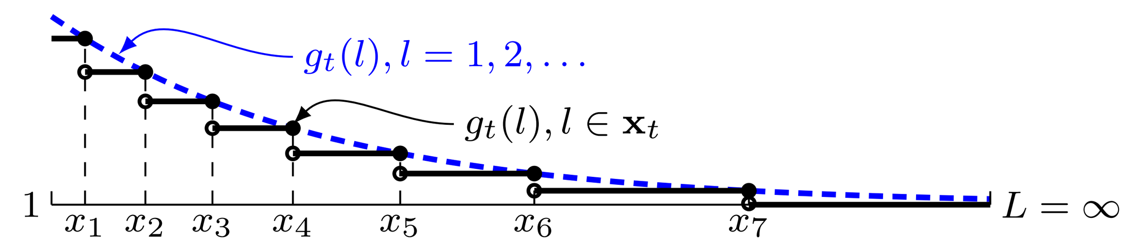
BasicReduction

- Assume lifetime is upper bounded by L, i.e., $l_e \le L$.
- Let $\bar{E}_t$ denote the new edges arrived at time t, and let $\bar{E}_t = \cup_{l=1}^{L} \bar{E}_l^{(t)}$.
- BasicReduction maintains l SieveADN instances $\{\mathcal{A}_l^{(t)}\}_{l=1}^{L}$ to process edges $\{\bar{E}_l^{(t)}\}_{l=1}^{L}$ in parallel.
- Property: $\mathcal{A}_1^{(t)}$ always processed all the edges in G_t.



HistApprox

- Use a histogram to approximate a "curve".
- Requires only $O(\epsilon^{-1}\log k)$ indices, i.e., $|\mathbf{x}_t| = O(\epsilon^{-1}\log k)$.

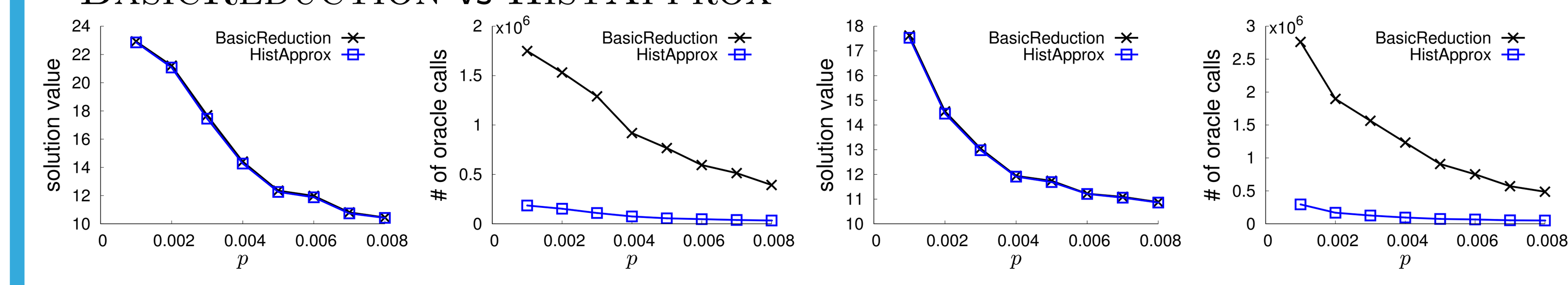


Experiments

- Data

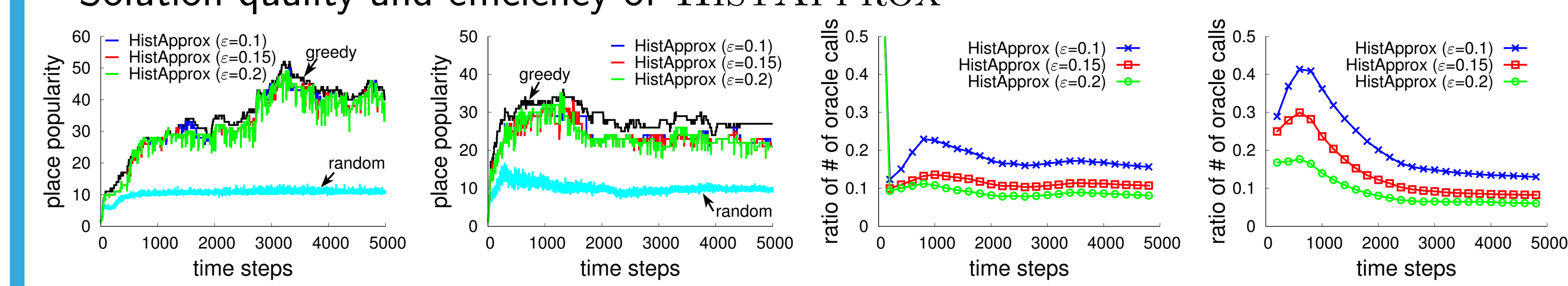
Interaction Data	# of nodes	# of interactions
Brightkite (users/places)	51,406/772,966	4,747,281
Gowalla (users/places)	107,092/1,280,969	6,442,892
Twitter-Higgs	304,198	555,481
Twitter-HK	49,808	2,930,439
StackOverflow-c2q	1,627,635	13,664,641
StackOverflow-c2a	1,639,761	17,535,031

- BasicReduction vs HistApprox

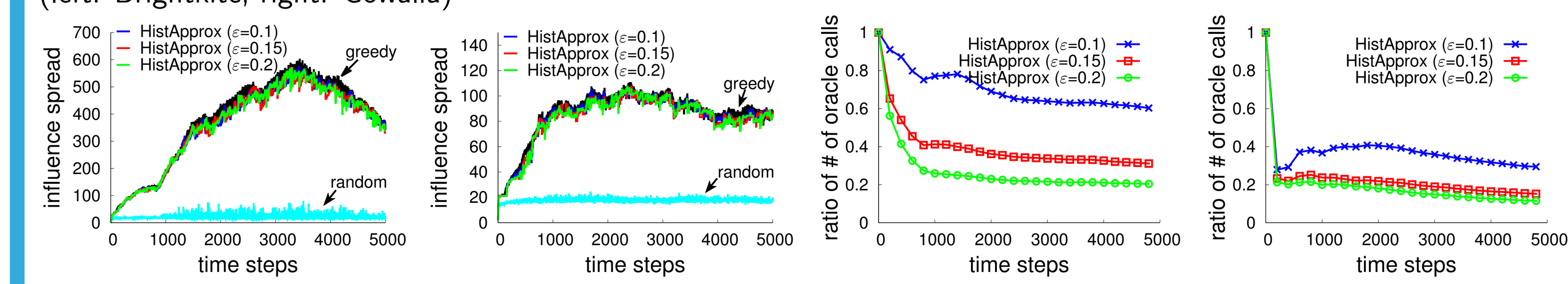


(left: Brightkite, right: Gowalla)

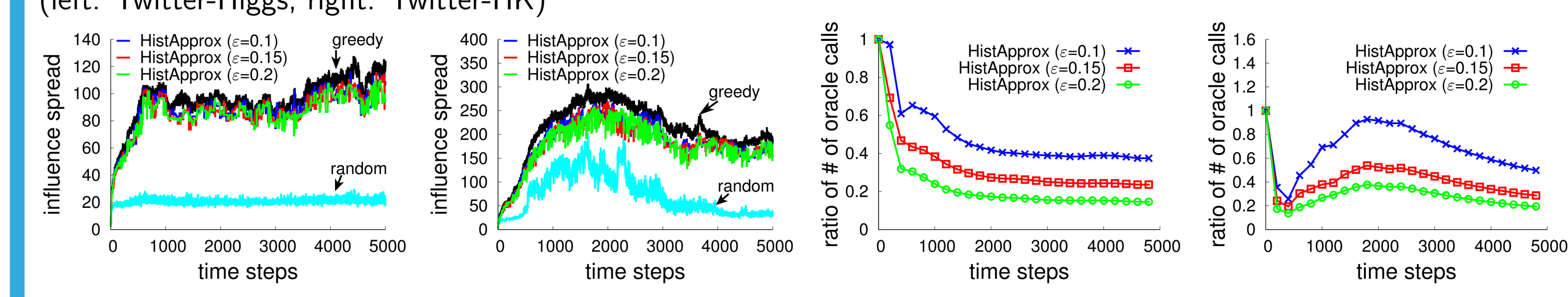
- Solution quality and efficiency of HistApprox



(left: Brightkite, right: Gowalla)



(left: Twitter-Higgs, right: Twitter-HK)



(left: StackOverflow-c2q, right: StackOverflow-c2a)

References

[1] A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause. Streaming submodular maximization: Massive data summarization on the fly. In *ACM SIGKDD*, pages 671–680, New York, New York, USA, 2014.