Tracking Influential Nodes in Time-Decaying Dynamic Interaction Networks

Junzhou Zhao¹, Shuo Shang², Pinghui Wang³, John C.S. Lui⁴, Xiangliang Zhang¹
¹KAUST ²IIAI ³JTU ⁴CUHK

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., 90% 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.

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 $e \in \overrightarrow{E}(u, v, t)$ represents a node interaction: user $u$ influenced user $v$ at time $t$, e.g., if user $a$ retweeted user $b$’s tweet in Twitter.
- Each edge $e$ can survive for $t$, time units. $t > 0$ 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 $G_t$.

Tracking Influential Nodes over TDNs
- Influence spread of a set of nodes $S$ in $G_t$: $f(S) = \{v: \text{node } v \text{ is reachable from } S \text{ in } G_t\}$
- Inflouential Nodes Tracking Problem:
  - Given TDN $G_t$, evolving over time $t$, and budget $k$.
  - Want to find $S^* \subseteq V_t$ with $|S^*| \leq k$, such that $S^* = \arg \max_{S \subseteq V_t, |S| \leq k} f(S)$.

Overview of Our Algorithms
- **SieveADN**
- **BasicReduction**
- **HistApprox**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Update Time</th>
<th>Memory</th>
<th>Approximate Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>SieveADN</td>
<td>$O(k\log k)$</td>
<td>$O(k\log k)$</td>
<td>$1/2 - \epsilon$</td>
</tr>
<tr>
<td>BasicReduction</td>
<td>$O(k\log^2 k)$</td>
<td>$O(k\log^2 k)$</td>
<td>$1/2 - \epsilon$</td>
</tr>
<tr>
<td>HistApprox</td>
<td>$O(k\log^2 k)$</td>
<td>$O(k\log^2 k)$</td>
<td>$1/3 - \epsilon$</td>
</tr>
<tr>
<td>Greedy</td>
<td>$O(k\log^2 k)$</td>
<td>$O(k\log^2 k)$</td>
<td>$1 - \epsilon$</td>
</tr>
</tbody>
</table>

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., $t \leq L$.
- Let $E_t$ denote the new edges arrived at time $t$, and let $E_t = \overrightarrow{E}(t)$.
- BasicReduction maintains the SieveADN instances $\{A_t^0\}_{t=1}^L$ to process edges $\{E_t^0\}_{t=1}^L$ in parallel.
- Property: $A_t^i$ always processed all the edges in $G_t$.

HistApprox
- Use a histogram to approximate a “curve”.
- Requires only $O(e^{1/\log k})$ indices, i.e., $|S| = O(e^{1/\log k})$.

Experiments

<table>
<thead>
<tr>
<th>Data</th>
<th># of nodes</th>
<th># of interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brightkite (users/places)</td>
<td>51,901</td>
<td>12,785</td>
</tr>
<tr>
<td>Gowalla (users/places)</td>
<td>107,092/1</td>
<td>280,969</td>
</tr>
<tr>
<td>Twitter-Higgs</td>
<td>36,478</td>
<td>6,442,892</td>
</tr>
<tr>
<td>Twitter-HK</td>
<td>36,478</td>
<td>6,442,892</td>
</tr>
<tr>
<td>StackOverflow-c2q</td>
<td>1,627,655</td>
<td>15,664,641</td>
</tr>
<tr>
<td>StackOverflow-c2a</td>
<td>1,627,761</td>
<td>17,533,034</td>
</tr>
</tbody>
</table>

HistApprox
- Solution quality and efficiency of HistApprox

References