A Budgeted Framework to Model a Multi-round Competitive Influence Maximization Problem

Nadia Niknami, Jie Wu

Computer and Information Science Department
Temple University
Outline

- Models of influence
  - Linear Threshold
  - Independent Cascade

- Influence maximization problem Algorithm

- Competitive Influence maximization problem
- Our Approach
- Experiments
  - Data and setting
  - Results
What is Social Influence?

- Social network plays a fundamental role as a medium for the spread of **INFLUENCE** among its members
  - Opinions, ideas, information, innovation…

- **Nodes**: Social actors (individuals or organizations)
- **Links**: Social relations

- Social influence occurs when one's opinions, emotions, or behaviors are affected by others, intentionally or unintentionally.

- Influential persons often have many friends.
Influential Persons

- Number of friends: Node degree
- Famous persons
- Betweenness centrality

- Direct Marketing takes the “word-of-mouth” effects to significantly increase profits
Influence Maximization Problem

- Influence spread of node set $S$: $\sigma(S)$
  - expected number of active nodes at the end of diffusion process, if set $S$ is the initial active set.
- Problem Definition (by Kempe et al., 2003):
  (Influence Maximization). Given a directed and edge-weighted social graph $G = (V, E, p)$, a diffusion model $m$, and an integer $k \leq |V|$, find a set $S \subseteq V, |S| = k$, such that the expected influence spread $\sigma(S)$ is maximum.
Influence Maximization Problem

• Diffusion Model
  • Linear Threshold (LT)
    • A node has random threshold $\theta$
    • A node becomes active when at least $\theta$ fraction of its neighbors are active
  • Independent Cascade (IC)
    • When node $v$ becomes active, it has a single chance of activating each currently inactive neighbor $w$.
    • The activation attempt succeeds with probability $p_{vw}$
Competitive Influence Maximization

- First, parties identify the most influential nodes of the network.
- Then they compete over only these influential nodes by the amount of budget they allocate to each node.
Multi-round Diffusion

• In a multi-stage CIM problem, competitors need to select seed nodes simultaneously in each of the sequence stages.

• 1) Nod-Node influence competition
• 2) Link-Link influence competition
• 3) Node-Link influence competition
A Budgeted Framework to Model a Multi-round Competitive Influence Maximization Problem

- We integrate seed selection and budget allocation into the RL model.

- Selecting Seed Nodes and Propagation Model
  - Most Reliable Influence Path (MRIP)
    - Fixed weight
    - Dynamic weight

- Reinforcement Learning
Algorithm 2 Finding seed set by MRIP

1: \( S \leftarrow \emptyset \)
2: for all \( u \in V \) do
3: \( w'(u) \leftarrow 0 \)
4: for \( u \in V \) do
5: Construct \( \mathcal{T}_u \) via Alg. 3
6: for each leaf \( v \) in reverse \( \mathcal{T}_u \) do
7: \( z \leftarrow \text{parent}(v) \)
8: while \( v \neq u \) do
9: Compute \( w'(z) = w'(z) + R(v) \times w(z) \)
10: \( v \leftarrow z \)
11: \( z \leftarrow \text{parent}(v) \)
12: new seed \( \leftarrow \arg \max_{u \in V/\mathcal{T}_u} w'(u) \)
13: \( S \leftarrow S \cup \{\text{new seed}\} \)
14: \( V_A \leftarrow \text{Activated nodes by new seed node} \)
15: Constructing \( \mathcal{G}' \) with vertex set \( V - V_A \)
16: Recalculate \( \mathcal{T} \) and \( w' \) in \( \mathcal{G}' \)

Algorithm 3 Computing \( \mathcal{T}_u \)

Require: \( \mathcal{G}(V,E,P) \), source node \( u \)
1: \( A = \{u\}, R(u) = 1 \)
2: while \( A \neq V \) do
3: Find node \( v \in N(A) \) and \( v \in V - A \) such that
4: \( R'(v) = \max_{s,v} R(s) \times p(s,v) \)
5: \( R(v) = R'(v) \)
6: \( A = A \cup \{v\} \)
7: Set \( s \) as the parent of \( v \) in spanning tree \( \mathcal{T}_u \)
8: return \( \mathcal{T}_u \)
Most Reliable Influence Path (MRIP)

\[ v_2 \]

\[ v_2, v_8 \]

\[ v_2, v_8, v_7 \]

\[ v_2, v_8, v_7, v_1 \]

\[ v_2, v_8, v_7, v_1, v_3 \]

\[ v_2, v_8, v_7, v_1, v_3, v_5 \]

\[ v_2, v_8, v_7, v_1, v_3, v_5, v_6 \]

\[ v_2, v_8, v_7, v_1, v_3, v_5, v_6, v_4 \]

\[ v_2, v_8, v_7, v_1, v_3, v_5, v_6, v_4, v_9 \]

<table>
<thead>
<tr>
<th>( A )</th>
<th>N(A)</th>
<th>( R(s) )</th>
<th>( p(s, v) )</th>
<th>( R(v) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( { v_2 } )</td>
<td>( v_1 )</td>
<td>( 1 \cdot 0.2 = 0.2 )</td>
<td>( 1*0.4=0.4 )</td>
<td>( R(v_8) )</td>
</tr>
<tr>
<td>( { v_2, v_8 } )</td>
<td>( v_1 )</td>
<td>( 1 \cdot 0.2 = 0.2 )</td>
<td>( 0.4 \cdot 0.1 = 0.04 )</td>
<td>( R(v_7) )</td>
</tr>
<tr>
<td>( { v_2, v_8, v_7 } )</td>
<td>( v_1 )</td>
<td>( 1 \cdot 0.2 = 0.2 )</td>
<td>( 0.4 \cdot 0.1 = 0.04 )</td>
<td>( R(v_1) )</td>
</tr>
<tr>
<td>( { v_2, v_8, v_7, v_1 } )</td>
<td>( v_1 )</td>
<td>( 0.4 \cdot 0.1 = 0.04 )</td>
<td>( 0.2 \cdot 0.5 = 0.1 )</td>
<td>( R(v_3) )</td>
</tr>
<tr>
<td>( { v_2, v_8, v_7, v_1, v_3 } )</td>
<td>( v_1 )</td>
<td>( 0.4 \cdot 0.1 = 0.04 )</td>
<td>( 0.2 \cdot 0.4 = 0.08 )</td>
<td>( R(v_5) )</td>
</tr>
<tr>
<td>( { v_2, v_8, v_7, v_1, v_3, v_5 } )</td>
<td>( v_1 )</td>
<td>( 0.4 \cdot 0.1 = 0.04 )</td>
<td>( 0.2 \cdot 0.4 = 0.08 )</td>
<td>( 1 \cdot 0.1 = 0.1 )</td>
</tr>
<tr>
<td>( { v_2, v_8, v_7, v_1, v_3, v_5, v_6 } )</td>
<td>( v_1 )</td>
<td>( 0.4 \cdot 0.1 = 0.04 )</td>
<td>( 0.2 \cdot 0.4 = 0.08 )</td>
<td>( 1 \cdot 0.1 = 0.1 )</td>
</tr>
<tr>
<td>( { v_2, v_8, v_7, v_1, v_3, v_5, v_6, v_4 } )</td>
<td>( v_4 )</td>
<td>( 1 \cdot 0.1 = 0.1 )</td>
<td>( 0.3 \cdot 0.4 = 0.12 )</td>
<td>( R(v_4) )</td>
</tr>
<tr>
<td>( { v_2, v_8, v_7, v_1, v_3, v_5, v_6, v_4, v_9 } )</td>
<td>( v_4 )</td>
<td>( 1 \cdot 0.1 = 0.1 )</td>
<td>( 0.12 \cdot 0.2 = 0.024 )</td>
<td>( R(v_4) )</td>
</tr>
<tr>
<td>( { v_2, v_8, v_7, v_1, v_3, v_5, v_6, v_4, v_9 } )</td>
<td>( v_9 )</td>
<td>( 0.12 \cdot 0.2 = 0.024 )</td>
<td>( R(v_9) )</td>
<td></td>
</tr>
</tbody>
</table>
In RL, the agent keeps interacting with the environment to find the optimal policy $\pi$ to maximize his expected accumulated rewards.

- **State**
- **Action**
State

1) Number of inactive nodes
2) Summation of degrees of all inactive nodes
3) Maximum degree among all inactive nodes
4) Summation of the weight of the edges for which both vertices are inactive
5) Summation of the inactive out-edge weight for nodes which are the neighbors of player 1
6) Maximum sum of the inactive out-edge weight of a node among all nodes
7) Ratio of budgets
8) Weight of nodes in case of reachability

Action:

1) Selecting a new seed node and
2) feeding a node in case of tie.
1) Evaluation on Budget Setting
2) Evaluation Based on Different Topologies
3) Evaluation on Edge-weight Setting
   - Case 1: uniform probabilities p on each edge
   - Case 2: weight of the edges in the range of [0.1, 0.4] and [0.4, 0.7]. In addition, the weight for edges are randomly sampled from the normal distribution of U (0, 0.2) and U (0, 1).

4) Evaluation on Different Competing Strategies:
Thank you

Nadia.niknami@temple.edu