Online Influence Maximization under Independent Cascade Model with Semi-Bandit Feedback

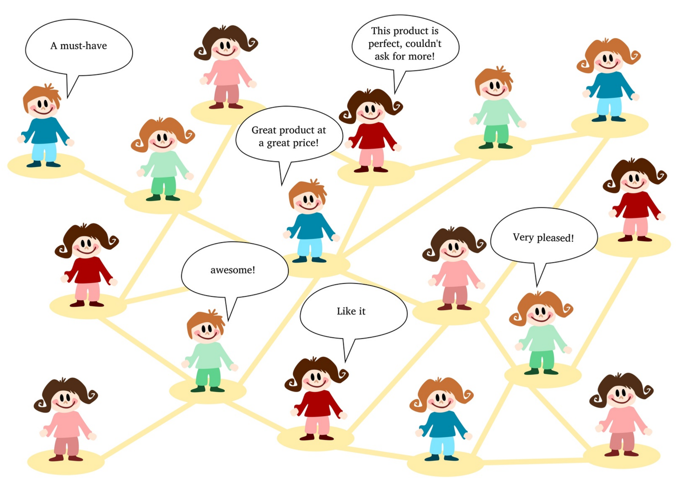
Zheng Wen [1] Branislav Kveton [1] Michal Valko [2] Sharan Vaswani [3]
[1] Adobe Research [2]SequeL team, INRIA Lille - Nord Europe [3]University of British Columbia

Overview

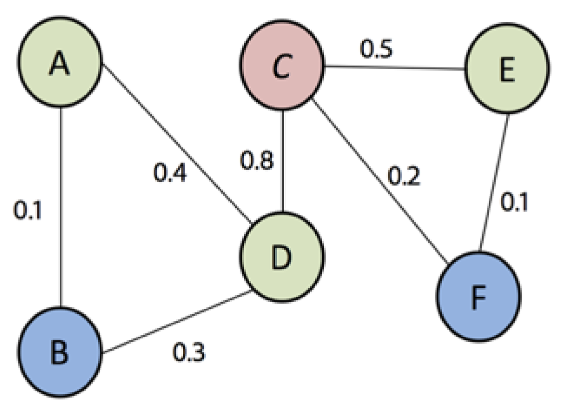
- We study the online influence maximization (OIM) problem in social networks under the independent cascade model with edge-level semi-bandit feedback
- We propose a LinUCB-based algorithm, referred to as `IMLinUCB`
  - `IMLinUCB` permits linear generalization, and is both statistically and computationally efficient for large-scale problems
  - We also propose an approach to construct features based on `node2vec`
- We have derived regret bounds for `IMLinUCB` when linear generalization is perfect
  - Our bounds reflect the topology of the network and the activation probabilities of its edges
  - Experiments show that in several representative graph topologies, the regret of `IMLinUCB` scales as suggested by our upper bounds
- Demonstrate experiment results in a subgraph of Facebook network

Influence Maximization under Independent Cascade (IC) Model

- This problem is characterized by $(\mathcal{G}, K, \bar{w})$
  - $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed graph with $L = |\mathcal{V}|$ nodes and $|\mathcal{E}|$ edges
  - $K \leq L$ is the cardinality of source nodes
  - $\bar{w} : \mathcal{E} \to [0, 1]$ encodes the activation probabilities of edges
- The influence diffuses according to the IC model
  - **Initialization:** source nodes are influenced
  - Each influenced node has a single chance to activate its currently uninfluenced neighbors
  - The successes of these activations are independent Bernoulli r.v. with mean specified by $\bar{w}$



(a) Information diffusion in a social network



(b) Input for an IM problem

- **Influence maximization:**

$$\max_{\mathcal{S}: |\mathcal{S}|=K} f(\mathcal{S}, \bar{w})$$

  - $f(\mathcal{S}, \bar{w})$ is the expected number of influenced nodes when the source node set is $\mathcal{S}$ and the edge activation probabilities are $\bar{w}$
  - NP-hard, but efficient approximation algorithms (referred to as oracles) exist
  - For any $\alpha, \gamma \in [0, 1]$, we say that `ORACLE` is an (α, γ)-approximation oracle if for any $\bar{w}$,

$$f(\mathcal{S}^*, \bar{w}) \geq \gamma f(\mathcal{S}^{\text{opt}}, \bar{w})$$

with probability at least α, where $\mathcal{S}^{\text{opt}}$ is the optimal solution and $\mathcal{S}^* = \texttt{ORACLE}(\mathcal{G}, K, \bar{w})$

Influence Maximization Semi-Bandit

- Also characterized by $(\mathcal{G}, K, \bar{w})$, but $\bar{w}$ is unknown to the agent
- **Goal:** maximize the expected cumulative reward in n rounds
- **Protocol at each time t:**
  - Agent adaptively chooses a source node set $\mathcal{S}_t$, based on its prior information and past observations
  - Influence diffuses from $\mathcal{S}_t$ according to the IC model
  - Agent receives a reward = (the number of influenced nodes)
  - Agent observes the success/failure of each activation attempt from all influenced nodes (edge-level semi-bandit feedback)

Linear Generalization

- # of edges in real-world social networks tends to be in millions or even billions
- Efficient learning in such cases requires exploiting generalization models
- **This paper:** we assume a linear generalization model for $\bar{w}$
  - $\bar{w}(e) \approx x_e^T \theta^*$ for all $e \in \mathcal{E}$
  - $x_e \in \Re^d$ is a known feature vector for edge e, and $\theta^* \in \Re^d$ is an unknown coefficient vector
  - "Linear generalization is perfect" iff $\bar{w}(e) = x_e^T \theta^* \ \forall e \in \mathcal{E}$
  - Standard assumption in linear bandit literature
- Use $X \in \Re^{|\mathcal{E}| \times d}$ to denote the feature matrix
  - **Tabular case:** $X = I \in \Re^{|\mathcal{E}| \times |\mathcal{E}|}$

`IMLinUCB` Algorithm

- **Input:** $\mathcal{G}$, K, `ORACLE`, feature vector x_e's, and algorithm parameters $\sigma, c > 0$
- **Initialization:** $B_0 \leftarrow 0 \in \Re^d$, $M_0 \leftarrow I \in \Re^{d \times d}$
- For $t = 1, 2, \ldots, n$:
  - Set $\bar{\theta}_{t-1} \leftarrow \sigma^{-2} M_{t-1}^{-1} B_{t-1}$ and the UCBs as

$$U_t(e) \leftarrow \text{Proj}_{[0,1]} \left(x_e^T \bar{\theta}_{t-1} + c\sqrt{x_e^T M_{t-1}^{-1} x_e} \right)$$

  for all $e \in \mathcal{E}$
  - Choose $\mathcal{S}_t \in \texttt{ORACLE}(\mathcal{G}, K, U_t)$, and observe the edge-level semi-bandit feedback
  - Update statistics: (a) Initialize $M_t \leftarrow M_{t-1}$ and $B_t \leftarrow B_{t-1}$ (b) For all observed edges $e \in \mathcal{E}$, update $M_t \leftarrow M_t + \sigma^{-2} x_e x_e^T$ and $B_t \leftarrow B_t + x_e \mathbf{w}_t(e)$

Maximum Observed Relevance C_*

- A novel complexity metric reflecting both (1) the topology of the network and (2) the activation probabilities of its edges
- **Edge-node relevance:** under given source node set $\mathcal{S} \subseteq \mathcal{V}$, an edge $e \in \mathcal{E}$ is relevant to a node $v \in \mathcal{V} \setminus \mathcal{S}$ if $\exists$ a path p from a source node $s \in \mathcal{S}$ to v s.t. (1) $e \in p$ and (2) p does not contain another source node.
- Define $N_{\mathcal{S},e}$ as the number of nodes edge e is relevant to under source node set $\mathcal{S}$, and $P_{\mathcal{S},e} \triangleq \mathbb{P}(e \text{ is observed} \mid \mathcal{S})$
- **Maximum observed relevance** C_* is defined as the maximum (over $\mathcal{S}$) 2-norm of $N_{\mathcal{S},e}$'s weighted by $P_{\mathcal{S},e}$'s,

$$C_* \triangleq \max_{\mathcal{S}: |\mathcal{S}|=K} \sqrt{\sum_{e \in \mathcal{E}} N_{\mathcal{S},e}^2 P_{\mathcal{S},e}} \overset{(a)}{\leq} \max_{\mathcal{S}: |\mathcal{S}|=K} \sqrt{\sum_{e \in \mathcal{E}} N_{\mathcal{S},e}^2} \overset{(b)}{\leq} (L - K)\sqrt{|\mathcal{E}|},$$

where (a) is a topology-based upper bound (see the table below), and (b) is a size-based upper bound. Both bounds are far from tight if $\bar{w}(e)$'s are small.

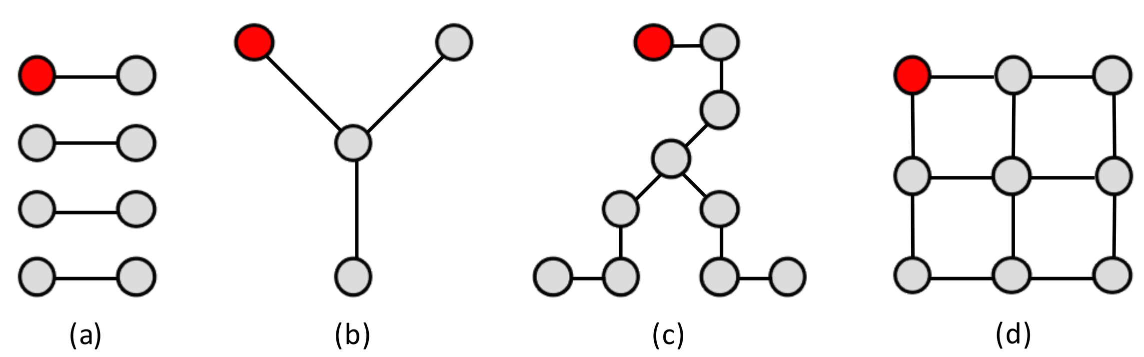


Figure: **a**. Bar graph. **b**. Star graph. **c**. Ray graph. **d**. Grid graph.

topology	bar graph	star graph	ray graph	tree graph	grid graph	complete graph
C_*	$\mathcal{O}(\sqrt{K})$	$\mathcal{O}(L\sqrt{K})$	$\mathcal{O}(L^{\frac{5}{4}}\sqrt{K})$	$\mathcal{O}(L^{\frac{3}{2}})$	$\mathcal{O}(L^{\frac{3}{2}})$	$\mathcal{O}(L^2)$
regret bound	$\tilde{\mathcal{O}}(d\sqrt{Kn})$	$\tilde{\mathcal{O}}(Ld\sqrt{Kn})$	$\tilde{\mathcal{O}}(dL^{\frac{5}{4}}\sqrt{Kn})$	$\tilde{\mathcal{O}}(dL^{\frac{3}{2}}\sqrt{n})$	$\tilde{\mathcal{O}}(dL^{\frac{3}{2}}\sqrt{n})$	$\tilde{\mathcal{O}}(dL^2\sqrt{n})$

Table: C_* and regret bounds for different topologies.

Regret Bounds

- **Scaled cumulative regret:** for any $\eta > 0$, the scaled cumulative regret is defined as

$$R^\eta(n) \triangleq \sum_{t=1}^n \mathbb{E}\left[\eta f(\mathcal{S}^{\text{opt}}, \bar{w}) - f(\mathcal{S}_t, \bar{w})\right]$$

  - When $\eta = 1$, $R^\eta(n)$ reduces to the standard cumulative regret $R(n)$
- **Theorem 1:** Assume that (1) $\bar{w}(e) = x_e^T \theta^*$ for all $e \in \mathcal{E}$ and (2) `ORACLE` is an (α, γ)-approximation algorithm, then if we apply `IMLinUCB` with properly chosen σ and c, we have

$$R^{\alpha\gamma}(n) \leq \tilde{\mathcal{O}}\left(dC_*\sqrt{n}\right) \leq \tilde{\mathcal{O}}\left(d(L-K)\sqrt{|\mathcal{E}|n}\right)$$

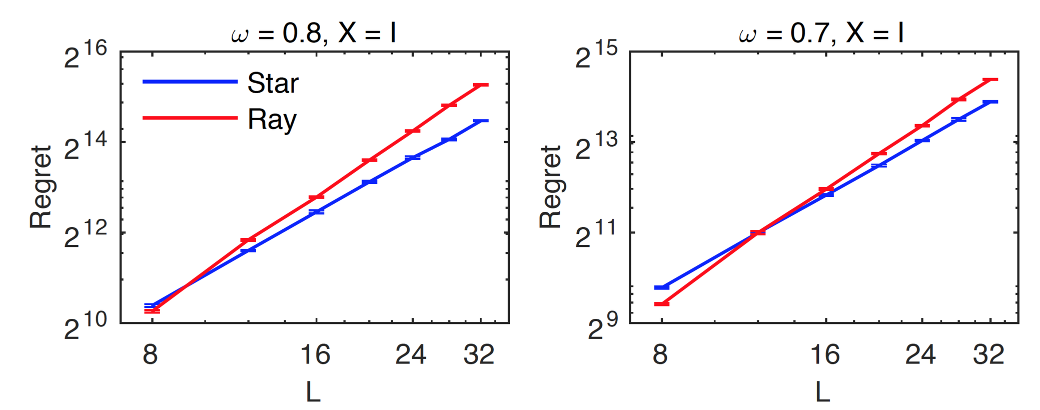
- See the left table for topology-dependent regret bounds
- **Tabular case:** $X = I$ and hence $d = |\mathcal{E}|$
- For bar graph with $K = 1$, $R(n) = \tilde{\mathcal{O}}(d\sqrt{n})$
  - Matches the known regret bound of `LinUCB`

Experiment 1: Stars and Rays

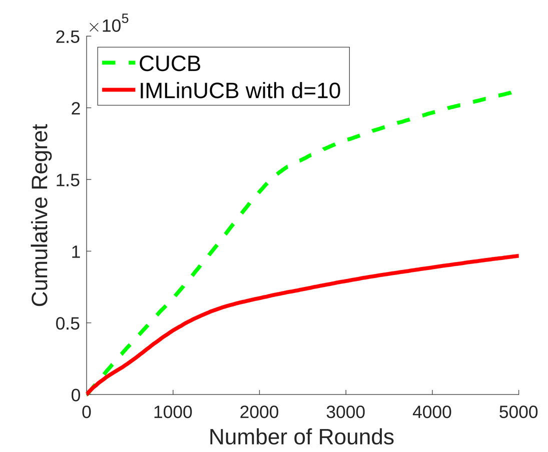
- We validate that when applying `IMLinUCB` to stars and rays, the regret's dependence on L is as predicted by Theorem 1
- **Setting:** tabular case, $K = 1$, $n = 10^4$, and $\bar{w}(e) = w \ \forall e$
  - The IM problem can be solved exactly
  - The exponent of L is estimated by linear regression in the log-log space of L and regret

topology	Theorem 1's prediction	estimation ($w = 0.8$)	estimation ($w = 0.7$)
star	$R(n) = \tilde{\mathcal{O}}(L^2)$	$R(n) = \mathcal{O}(L^{2.040})$	$R(n) = \mathcal{O}(L^{2.056})$
ray	$R(n) = \tilde{\mathcal{O}}(L^{\frac{9}{4}})$	$R(n) = \mathcal{O}(L^{2.488})$	$R(n) = \mathcal{O}(L^{2.467})$

Table: Validation of regret's dependence on L in stars and rays.



(a) Stars and Rays



(b) Subgraph of Facebook network

Experiment 2: Subgraph of Facebook Network

- Consider a subgraph of Facebook network from Snap datasets [Leskovec and Krevl 2014]
  - **Setting:** $L = 327$, $|\mathcal{E}| = 5038$, $n = 5000$, $K = 10$, and $d = 10$
  - $\bar{w}(e)$'s are independently sampled from $U(0, 0.1)$
  - Choose `ORACLE` as the IM algorithm proposed in [Tang et al. 2014]
  - Regret is measured against $\mathcal{S}^* = \texttt{ORACLE}(\mathcal{G}, K, \bar{w})$
- **Feature construction approach:**
  - First use `node2vec` algorithm to generate a node feature in $\Re^d$ for each node $v \in \mathcal{V}$
  - Then for each edge e, we generate x_e as the element-wise product of node features of the two nodes connected to e
  - Linear generalization is imperfect in this case
- **Baseline:** CUCB [Chen et al. 2013]
  - Another UCB-like algorithm that does not exploit generalization models
- **Experiment result:** dramatic regret reduction of `IMLinUCB` by exploiting even imperfect linear generalization