Adaptive Influence Maximization

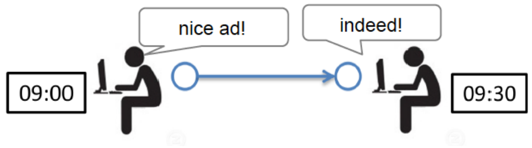
Bogdan Cautis, Silviu Maniu, Nikolaos Tziortziotis

KDD 2019, Anchorage, August 4th 2019

Social Media Advertising

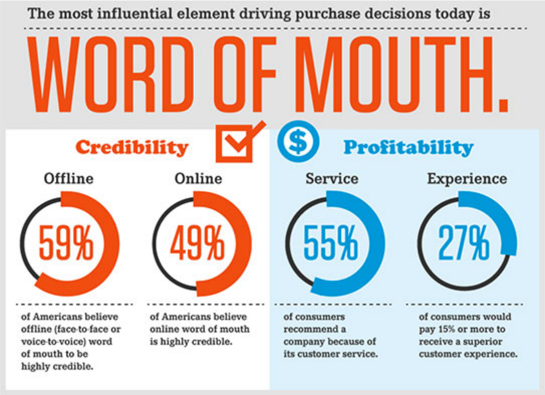
Social media advertising budgets have doubled worldwide from 2014 to 2016, reaching $30B, continuing with double-digit growth.

Word-of-mouth in Social Networks



Importance of Word-of-mouth Diffusion

Lexicon of modern marketers: word-of-mouth, social value, social whales, influencers, online social strategy, etc.

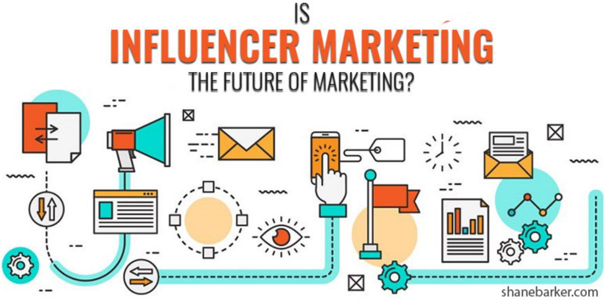


Word-of-mouth Diffusion and Influencers

And experiencing directly right now . . .

The Future of Online Marketing: Influencer Marketing

A new, highly effective, rapidly growing form of marketing on the social Web.



Influencer Marketing

Focus on influential people rather than the target market as a whole (Wikipedia).

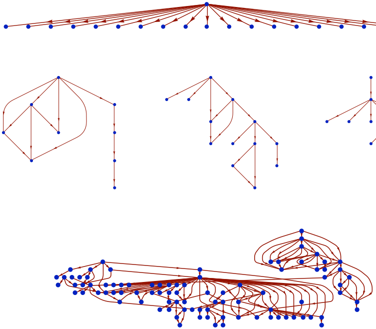
Influence Maximization (IM) [Kempe et al., 2003]

Objective

Given a promotion budget, maximize the influence spread in a social network, by the word-of-mouth effect

- Select k spread seeds in the social graph, given diffusion graph $G = (V, E)$ and a propagation model;
- Edges correspond to following relations, friendships, etc., in the social media environment

Influence Cascades



Influence Cascades

Time-ordered sequence of records indicating when a user adopted the product (was activated), starting a one or several persons [Bakshy et al., 2011]

IM Objective

- Denoting $\sigma(S)$ the influence cascade starting from a set of seeds S, the objective of IM is to solve the following problem:

$$\arg\max_{S \subseteq V, |S| \leqslant k} \mathbb{E}[|\sigma(I)|]$$

- Measuring the size of an influence cascade depends on the propagation model

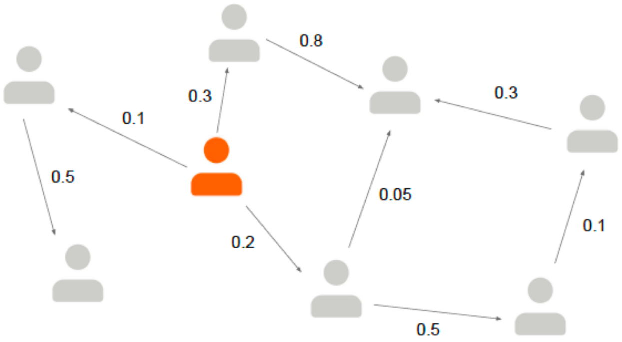
Independent Cascade (IC) Model [Kempe et al., 2003]

To each edge (u, v) from E, a probability $p(u, v)$ is associated

- at time 0 – activate seed s
- node u activated at time t – influence is propagated at $t + 1$ to neighbors v independently with probability $p(u, v)$
- once a node is activated, it cannot be deactivated / reactivated

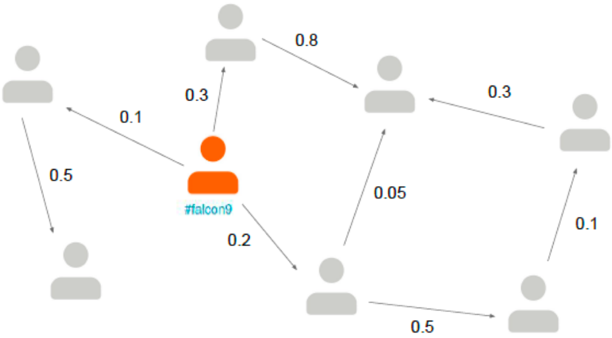
Independent Cascade (IC) Model – Example

One seed selected



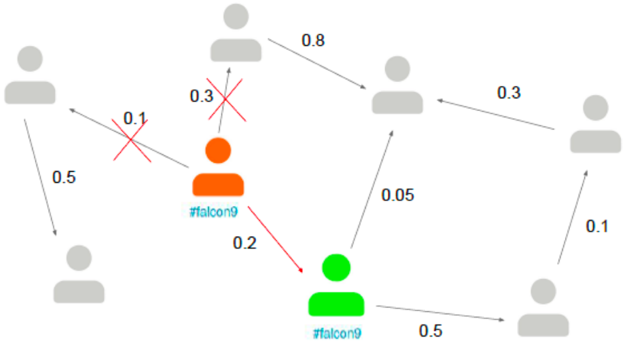
Independent Cascade (IC) Model – Example

Spread step 1



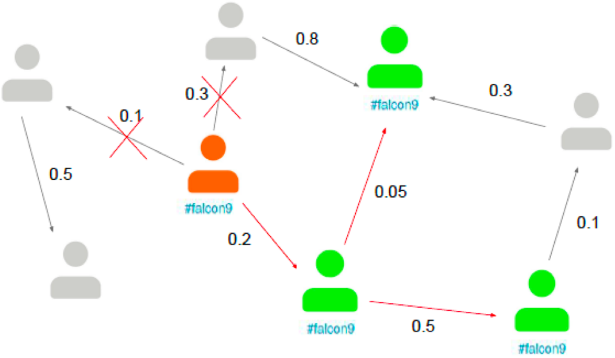
Independent Cascade (IC) Model – Example

Spread step 2



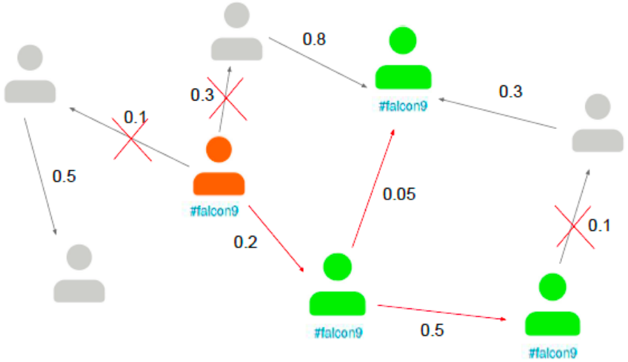
Independent Cascade (IC) Model – Example

Spread step 3



Independent Cascade (IC) Model – Example

Spread step 4



Linear Threshold (LT) Model [Kempe et al., 2003]

Similar to IC, we have weights $b(u, v)$ on each edge, but also a threshold $\theta(v) \in [0, 1]$ for each node. The LT process is as follows:

- at time 0 – activate seed s,
- at time t – all nodes active at t remain activated, and any node v is activated if:

$$\sum_{w \in N(v)} b(v, w) \geqslant \theta(v).$$

Submodularity and Approximation [Nemhauser et al., 1978]

The IM problem is known to be NP-hard, for both IC and LT.

Both LT and IC models are examples of submodular set functions, i.e., they respect:

$$\mathbb{E}\left[\sigma\left(S \cup \{v\}\right)\right] - \mathbb{E}[\sigma(S)] \geqslant \mathbb{E}[\sigma(T \cup \{v\})] - \mathbb{E}[\sigma(T)],$$

for all subsets of seeds $S \subseteq T \subseteq V$.

Submodular Set Function Optimization

The optimization problem is an instance of submodular set function optimization, known to give constant $1 - 1/e$ approximation algorithm via the greedy algorithm.

The Greedy Algorithm

ALGORITHM 1: – Greedy Submodular Maximization

Input: Graph $G(V, E)$, spread function σ, budget k

1: **Initialization:** set $S = \emptyset$
2: **for** $t = 1, \ldots, k$ **do**
3: Choose $v_t = \arg\max_{v \in E \setminus S} \mathbb{E}[\sigma(S \cup \{v\})]$
4: Update $S = S \cup \{v_t\}$
5: **end for**
6: **return** S

Adaptive Stochastic Optimization [Golovin and Krause, 2011]

- The objective of Adaptive Influence Maximization:

In practical situations, the model is known but the parameters - $p(u, v)$ and θ – are not.

The model needs to be learned adaptively and updated from priors – a case of Adaptive Optimization

Adaptivity [Golovin and Krause, 2011]

- $\phi : \mathcal{E} \rightarrow \mathcal{O}$ **realization** of the influence graph
- **Partial realization** $\psi \subseteq \mathcal{E} \times \mathcal{O}$
  - Domain: $\psi \subseteq \mathcal{E} \times \mathcal{O} \rightarrow$ set of nodes that are observed to be active through ψ
  - ψ *consistent* with ϕ: $\phi \sim \psi$
  - ψ a *sub-realisation* of ψ' ($\psi \prec \psi'$) if $\psi \subseteq \psi'$
- **Adaptive policy**: mapping π from partial realizations to nodes.
  - we write $\pi(\psi)$ for the node seeded by π under partial realization ψ
  - seeding $\pi(\psi)$ leads to partial realization $\psi' = \psi \cup (\pi(\psi), \phi(\pi(\psi)))$

Adaptive IM optimization problem

Discover policy π^* such that:

$$\pi^* \in \arg\max_{\pi} f_{avg} \triangleq \mathbb{E}_{\Phi}[f(E(\pi, \Phi), \Phi)] \quad \text{s.t.} \quad |E(\pi, \phi)| \leq k, \forall \phi$$

where $E(\pi, \phi) \subseteq \mathcal{V}$ represents the seed nodes that have been selected following policy π under realization ϕ

Adaptive Monotonicity and Submodularity

Definition: Expected Marginal Gain

The **conditional expected marginal benefit** of $v \in \mathcal{V}$, conditioned on partial realization ψ, is given as:

$$\Delta_f(v|\psi) \triangleq \mathbb{E}_\Phi\Big[f(dom(\psi) \cup \{v\}, \Phi) - f(dom(\psi), \Phi) | \Phi \sim \psi\Big].$$

Definition: Adaptive Monotonicity and Submodularity

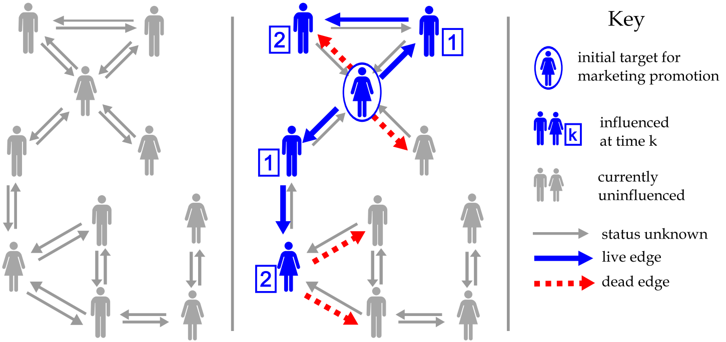
f is **adaptive monotone** *iff*, for all $v \in \mathcal{V}$ and ψ such that $\mathbb{P}(\Phi \sim \psi) > 0$, we have:

$$\Delta_f(v|\psi) \geq 0$$

f is **adaptive submodular** *iff*, for all $v \in \mathcal{V} \setminus dom(\psi')$ and $\psi \subseteq \psi'$, we have:

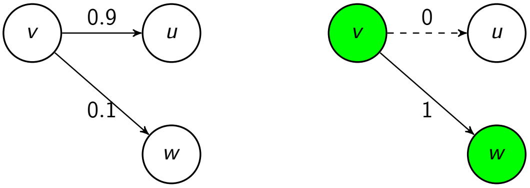
$$\Delta_f(v|\psi) \geq \Delta_f(v|\psi')$$

Adaptive Viral Marketing [Golovin and Krause, 2011]



Edge feedback model under IC propagation: given u, the realization $\phi(u)$ encodes each edge as live, dead, or unknown

Why Adaptive Influence Maximisation?



(a) Graph network (b) True world at time $t = 2$

Non-Adaptive Influence Maximisation

- Seed set: $\mathcal{S} = \{v, w\}$
- Total number of influenced nodes: 2

Adaptive Influence Maximisation

- Seed set: $\mathcal{S} = \{v, u\}$
- Total number of influenced nodes: 3

Adaptive Greedy

ALGORITHM 2: – Adaptive Greedy

Input: Graph $G(V, E)$, distribution $p(\phi)$ and utility function f, budget k

Initialization: set $S = \emptyset$, $\psi = \emptyset$
for $t = 1, \dots, k$ **do**
    Choose $v_t = \arg\max_{v \in E \setminus I} \Delta(e|\psi) = \mathbb{E}[f(S \cup \{v\}, \Phi) - f(S, \Phi)|\Phi \sim \psi]$
    Update $S = S \cup \{v_t\}$
    Observe $\Phi(v_t)$
    Update $\psi = \psi \cup \{(v_t, \Phi(v_t))\}$
end for
return S

Adaptive Greedy

Theorem

Since in the IC model with full-adoption feedback the influence function is adaptive monotone and adaptive submodular, the adaptive greedy algorithm is a $(1 - \frac{1}{e})$ approximation of the adaptive optimal policy.

The Multi-Armed Bandit View

Another way to see the problem is to consider that each node is an arm in a multi-armed bandit environment.

Setting:

- m arms each having random variable X_i (reward for arm i) having expectation $\mu_i \in [0, 1]$
- arms are "pulled" in T rounds, giving reward $R_i(t)$
- the measure of interest for multi-armed bandits algorithms is the regret R_t, i.e., the difference between always choosing the optimal arm (X_i^*) and the given algorithm:

$$\mathrm{Reg}_t = \mathbb{E}\left[\sum_{i=1}^{t} R^*(i)\right] - \mathbb{E}\left[\sum_{i=1}^{t} R(i)\right]$$

Huge literature on bandit algorithms, regret bounds in various settings (stochastic, adversarial, linear, combinatorial) [Lattimore and Szepesvári, 2019]

Setting and Bandit Feedback

Goal

Learn the set of "best influencers" in a social network by repeatedly interacting with it, by online IM campaigns.

Why MAB: may begin with no knowledge, at each step choose seeds that improve our knowledge (explore) or seeds that yield better spread.

- full-bandit feedback: only the number of activated nodes is revealed after each IM run
- edge semi-bandit feedback: all live edges are revealed (as in [Lei et al., 2015, Vaswani and Lakshmanan, 2016])
- node semi-bandit feedback: the activated nodes are revealed (as in [Vaswani and Lakshmanan, 2016, Lagrée et al., 2017, Lagrée et al., 2018])

Node-Level Feedback vs. Edge-Level Feedback

(Full) edge-level feedback

After a node (batch) is seeded, we can observe the status of each edge exiting an active node

(Full) node-level feedback

After a node (batch) is seeded, we can observe the status of each node (active / inactive)

Node-Level Feedback vs. Edge-Level Feedback

(Full) edge-level feedback

After a node (batch) is seeded, we can observe the status of each edge exiting an active node

- ☺ Most of the literature relies on this kind of feedback
- ☺ May be realistic in micro-blogging scenarios (tweet / retweet)
- ☹ Not very realistic in many other scenarios (e.g., purchase, share, like)

(Full) node-level feedback

After a node (batch) is seeded, we can observe the status of each node (active / inactive)

Node-Level Feedback vs. Edge-Level Feedback

(Full) edge-level feedback

After a node (batch) is seeded, we can observe the status of each edge exiting an active node

- ☺ Most of the literature relies on this kind of feedback
- ☺ May be realistic in micro-blogging scenarios (tweet / retweet)
- ☹ Not very realistic in many other scenarios (e.g., purchase, share, like)

(Full) node-level feedback

After a node (batch) is seeded, we can observe the status of each node (active / inactive)

- ☺ Realistic for most scenarios, more general
- ☹ Less studied in the literature (leads to credit assignment problems)

Combinatorial Multi-Armed Bandits (CMAB) [Chen et al., 2013]

Super-arms

In each round, a super-arm consisting of a subset of the m arms $S \subseteq 2^m$ is selected (combinatorial)
Then the outcomes of all arms in S are revealed (in some cases, the outcomes of some other arms are revealed)

The reward of a super-arm R_{S_t} depends only on the expected reward vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)$ and the arms in S

No access to the "real world" but to an oracle depending on $\boldsymbol{\mu}$ (or an estimation thereof); we assume it is an (α, β)-approximation oracle

$$\mathrm{Reg}_{\boldsymbol{\mu},\alpha,\beta}(t) = t \cdot \alpha \cdot \beta \cdot \mathrm{opt}_{\boldsymbol{\mu}} - \mathbb{E}\left[\sum_{i=1}^{t} R_{\boldsymbol{\mu}}(S_i)\right]$$

CUCB Algorithm [Chen et al., 2013]

ALGORITHM 3: – CUCB

Input: Arms $[m]$, Oracle algorithm

1: Maintain T_i – total number of times arm i has been played, the estimated mean $\hat{\mu}_i$
2: For each arm i, play an arbitrary super-arm $S \in \mathcal{S}$ such that $i \in S$ and update T_i and $\hat{\mu}_i$
3: $t \leftarrow m$
4: **while** true **do**
5: $\quad t \leftarrow t + 1$
6: $\quad$ Set each $\bar{\mu}_i = \hat{\mu}_i + \sqrt{\frac{3 \ln t}{2 T_i}}$
7: $\quad S = \text{Oracle}(\bar{\mu}_1, \ldots, \bar{\mu}_m)$
8: $\quad$ Play S and update each T_i and $\hat{\mu}_i$
9: **end while**

Based on the UCB (Upper Confidence Bound) algorithm – "optimism in the face of uncertainty"

CMAB and Influence Maximization [Chen et al., 2013]

Applying to influence maximization:

- arms are the edges in the graph $G(V, E)$ having expected probability p_{uv}
- the super-arm is a set of edges outgoing from at most k nodes
- the edges in the super-arm reveal if they are activated; but also other edges can reveal their outcome due to the influence spread – edge feedback
- the oracle is the classic IM algorithm using the estimated $\hat{\boldsymbol{\mu}}$; it is an $(1 - 1/e - \epsilon, 1 - 1/|E|)$-approximation

CUCB Regret for Influence Maximization

The CUCB regret is bounded by:

$$\text{Reg}(T) \leqslant \sum_{i \in E, \Delta^i_{\min} > 0} \frac{12 V^2 E^2 \ln T}{\Delta^i_{\min}} + \left(\frac{\pi^2}{2} + 1\right) E \Delta_{\max}$$

IMLinUCB: a LinUCB-like Algorithm [Wen et al., 2017]

IC semi-bandit algorithm (ICSB) - edge semi-bandit feedback

Known diffusion graph, unknown activation probabilities $w(e)$, but a linear generalisation: for each edge e there exists a d-dimensional known feature vector x_e s.t. $w(e)$ is well approximated by $x_e^T \theta^*$, where $\theta^* \in \mathcal{R}^d$ is an unknown coefficient vector that must be learned.

ALGORITHM 4: `IMLinUCB: Influence Maximisation Linear UCB`

Input: $G, k,$ `ORACLE`, feature vector x_e's, parameters $\sigma, c > 0$

1: **Initialization:** $B_0 \leftarrow 0 \in \mathbb{R}^d$, $M_0 \leftarrow \boldsymbol{I} \in \mathbb{R}^{d \times d}$

2: **for** $t = 1, 2, \ldots, n$ **do**

3: $\quad \bar{\theta}_{t-1} \leftarrow \sigma^{-2} M_{t-1}^{-1} B_{t-1}$, $U_t(e) \leftarrow \mathrm{Proj}_{[0,1]}\left(x_e \bar{\theta}_{t-1} + c\sqrt{x_e^\top M_{t-1}^{-1} x_e} \right), \forall e \in E$

4: $\quad$ choose $S_t \in$ `ORACLE`(G, k, U_t), and observe the edge-level semi-bandit feedback

5: $\quad$ update statistics:

6: $\quad\quad$ (a) Initialize: $M_t \leftarrow M_{t-1}$ and $B_t \leftarrow B_{t-1}$

7: $\quad\quad$ (b) for all observed $e \in E$, update $M_t \leftarrow M_t + \sigma^{-2} x_e x_e^\top$, $B_t \leftarrow B_t + x_e w_t(e)$

8: **end for**

Note: w/o features (tabular case) it reduces to CUCB [Chen et al., 2013].

Regret Analysis

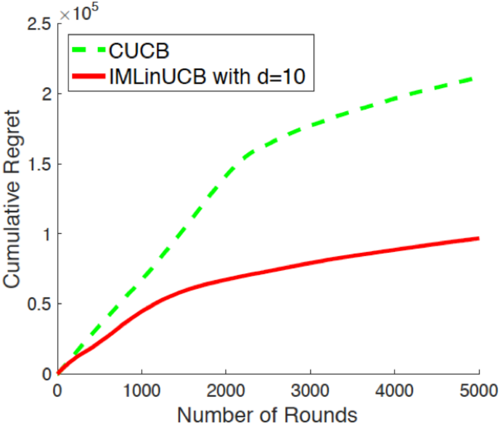
- Regret: accumulated loss in reward (spread) because of the lack of knowledge of the activation probabilities.
- η-scaled regret: $R_t^\eta = f(\mathcal{S}^{opt}) - \frac{1}{\eta} f(\mathcal{S}_t)$: e.g., $\eta = \alpha\gamma$, when the offline IM oracle is an (α, γ) approximation

Main Result

$$R_n^{\alpha\gamma} \leq \tilde{\mathcal{O}}\left((|\mathcal{V}| - k)\,|\mathcal{E}|^{\frac{3}{2}}\sqrt{n}/(\alpha\gamma)\right)$$

Experiments - Comparison with CUCB

Facebook graph, $|\mathcal{V}| = 0.3k$, $|\mathcal{E}| = 5k$, comparing with optimal (full-knowledge) strategy, IM oracle is TIM, $k = 5000$, $d = 10$



(b) Subgraph of Facebook network

Experimental Study of CMAB: Influence Maximization With Bandits [Vaswani and Lakshmanan, 2015]

Edge feedback: Same setting as [Chen et al., 2013],

Node feedback: challenge is updating the mean estimate for the activation probability of each edge, as any of the active parents may be responsible for activating a given node.

- MLE-based approach: similar to learning offline, from cascades (timestamped activations)
- frequentist approach: assuming low influence probabilities, hence few active parents, chose for attribution one parent randomly

Generic CMAB

ALGORITHM 5: CMAB framework for IM

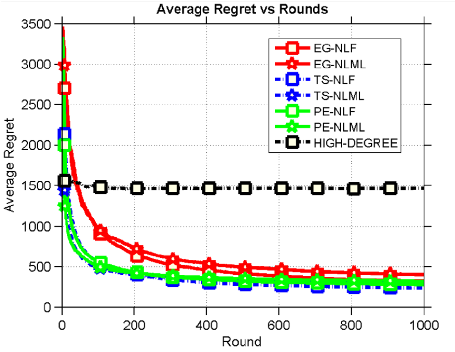
Input: G, k, feedback mechanism M, algorithm $\mathcal{A}$

Initialize $\vec{\hat{\mu}}$
$T_i = 0, \forall i$
`IS-EXPLOIT` is a boolean set by alg $\mathcal{A}$
if `IS-EXPLOIT` **then**
    $E_S = \texttt{EXPLOIT}(G, \vec{\hat{\mu}}, O, k)$
else
    $E_S = \texttt{EXPLORE}(G, k)$
end if
Play the superarm E_S, and observe the diffusion cascade c
$\vec{\hat{\mu}} = \texttt{UPDATE}(c, M)$

- instantiated with CUCB, ϵ-greedy, Thompson Sampling, pure exploitation.

Node Feedback Experiments - Flixster Example [Vaswani and Lakshmanan, 2015]

Flixster graph, $|\mathcal{V}| = 29k$, $|\mathcal{E}| = 300k$, WIC activation scores

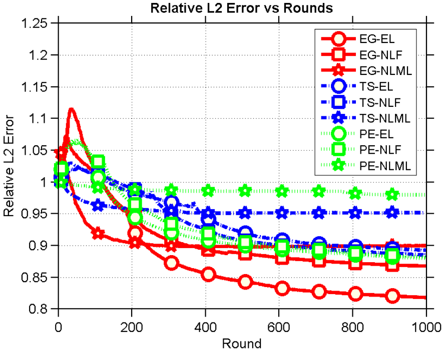


(a) Flixster

Note: CUCB omitted in the plot as it performs poorly, being biased towards exploring edges not triggered often → low rate of regret decrease

Node- vs. Edge-Level Feedback [Vaswani and Lakshmanan, 2015]

Flixster graph, $|\mathcal{V}| = 29k$, $|\mathcal{E}| = 300k$, WIC activation scores

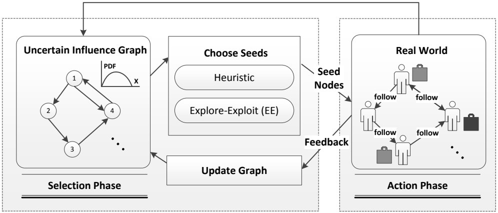


Online Influence Maximization [Lei et al., 2015]

Online Influence Maximization (OIM) framework:

- model the influence graph as having probabilities with priors on them, e.g., $p(u, v) \sim \text{Beta}(\alpha_{uv}, \beta_{uv})$
- for a budget of $k \times N$ seeds, run N rounds in which k seeds are activated, and feedback is gathered
- similar edge feedback to CMAB: a set of activated edges and the set of edges failing to be activated

OIM Framework [Lei et al., 2015]



ALGORITHM 6: – OIM Framework

Input: trials N, budget k, uncertain influence graph G

1: $A \leftarrow \emptyset$
2: **for** $n = 1$ **to** N **do**
3: $\quad S_n \leftarrow \texttt{Choose}(G, k)$
4: $\quad (A_n, F_n) \leftarrow \texttt{RealWorld}(S_n)$
5: $\quad A \leftarrow A \cup A_n$
6: $\quad \texttt{Update}(G, F_n)$
7: **end for**
8: **return** $(S_i)_{n=1\dots N}$, A

OIM – Algorithms [Lei et al., 2015]

There are several ways to implement `Choose` in an explore-exploit manner:

- ϵ-greedy approaches: explore with ϵ probability, exploit otherwise
- Upper Confidence bounds on the edges' distributions
- Exponentiated Gradient in which explore probabilities are dynamically updated

OIM – Updating the Model [Lei et al., 2015]

The model: the uncertain influence graphs modelled with (Beta) distributions on its probabilities

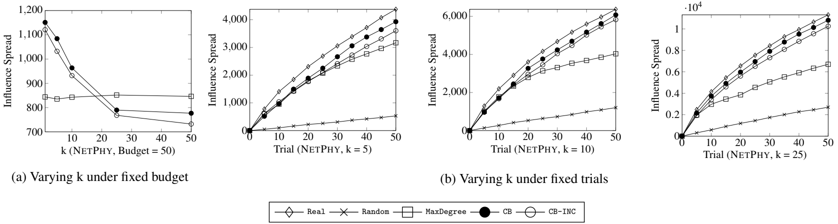
Can update:

- locally: e.g., using the conjugate prior properties of the Beta distribution:

$$\text{Beta}(\alpha_{uv}, \beta_{uv}) \to \text{Beta}(\alpha_{uv} + 1, \beta_{uv})$$

  in case of successful edge activation
- globally: assuming probabilities follow a global (in the graph) distribution: regression / MLE based on all previous feedback

OIM – Results [Lei et al., 2015]



(a) Varying k under fixed budget

(b) Varying k under fixed trials

Online Influencer Marketing [Lagrée et al., 2017, Lagrée et al., 2018]

Online and adaptive influence maximization:

- Influence campaign: multiple consecutive rounds spreading the same type of information
- Goal is to reach / activate as many users as possible
- Assuming a known set of spread seed candidates (the influencers), but no diffusion model

In each round:

- select some influencers from which a new spread starts
- the diffusion happens, observe activated nodes, but not the diffusion process itself
- influencers may be re-seeded throughout a campaign

Influence Persistence

A campaign with multiple rounds, diffusing the same post or different posts with the same semantics

- people may pass along the information several times, but "adopting" the concept rewards only once (e.g., in politics)
- brand fanatics, e.g., Star Wars, Apple, etc
- advertisement in users' feeds (e.g., Twitter), people may transfer / like the content several times during the campaign

Persistence

A node can be activated several times at different trials, but it is counted only once.

Motivation for Persistence

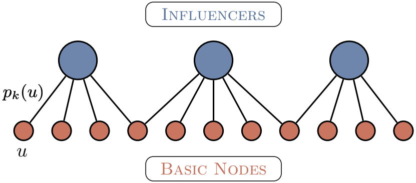
- Directly motivated by influencer marketing
- More realistic at many levels: no assumption regarding the diffusion model, simple feedback, IM via influencers
- Clear algorithmic interest: learn parameters on influencers (their potential) instead of diffusion edges –> large scale
- Independent influence campaigns with relatively short timespan

OIMP Formally [Lagrée et al., 2017, Lagrée et al., 2018]

- $[K] := \{1, \ldots, K\}$, set of influencers up for selection, N rounds, L influencers to be selected at each round
- Each influencer is connected to an unknown and potentially large base (its support, $A_k \subseteq V$) of basic nodes
- $p_k(u)$: each basic node u has an unknown activation probability by influencer k
- Influence process: when influencer k is selected, each basic node from A_k is sampled for activation
- Feedback: all activated basic nodes
- Reward: all newly activated basic nodes

$$\textit{Objective}: \quad \underset{I_n \subseteq [K], |I_n| = L, \forall 1 \leqslant n \leqslant N}{\arg\max} \mathbb{E} \left| \bigcup_{1 \leqslant n \leqslant N} S(I_n) \right|$$

OIMP Solution [Lagrée et al., 2018]



- Key difference w.r.t. classic MABs: no constant optimal seed set, selection at one trial depends on previous activations; we must follow an adaptive policy
- Algorithm GT-UCB: explore-exploit strategy using the Good-Turing estimator
- UCB-type algorithm: rely on upper confidence bounds on the estimator of remaining spread potential of an influencer

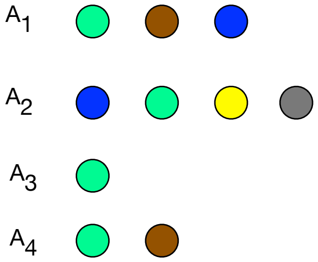
Good-Turing Estimator

Main idea: how to estimate the remaining spread for an influencer without knowing the model?

Good-Turing Estimator

Estimating the number of unique items left in a random process (e.g., species estimation, code breaking)

- estimated as the frequency of items encountered only once – hapaxes



Applying Good-Turing to OIMP [Lagrée et al., 2017, Lagrée et al., 2018]

For each influencer we need to estimate the remaining potential:

$$R_k(t) := \sum_{u \in A_k} \mathbb{1}\left\{ u \notin \bigcup_{i=1}^{t} S(i) \right\} p_k(u)$$

In the case of OIMP, we use the Good-Turing estimator as the frequency of nodes influenced only once:

$$\hat{R}_k(t) := \frac{1}{n_k(t)} \sum_{u \in A_k} U_k(u,t) \prod_{l \neq k} Z_l(u,t)$$

UCB index

We can plug this in a UCB algorithm by computing, for each influencer, the index:

$$b_k(t) = \hat{R}_k(t) + \left(1+\sqrt{2}\right)\sqrt{\frac{\hat{\lambda}_k(t)\log(4t)}{n_k(t)}} + \frac{\log(4t)}{3n_k(t)}$$

The GT-UCB Algorithm [Lagrée et al., 2017, Lagrée et al., 2018]

ALGORITHM 7: – GT-UCB ($L = 1$)

Input: Set of influencers $[K]$, time budget N

1: **Initialization:** play each influencer $k \in [K]$ once, observe the spread $S_{k,1}$, set $n_k = 1$
2: **for** $t = K + 1, \ldots, N$ **do**
3: Compute $b_k(t)$ for every influencer k
4: Choose $k(t) = \arg\max_{k \in [K]} b_k(t)$
5: Play influencer $k(t)$ and observe spread $S(t)$
6: Update statistics of influencer $k(t)$: $n_{k(t)}(t+1) = n_{k(t)}(t) + 1$ and $S_{k,n_k(t)} = S(t)$.
7: **end for**
8: **return** W

GT-UCB Theoretical Analysis [Lagrée et al., 2018]

Theorem: Good-Turing Deviation

With probability at least $1 - \delta$, for $\lambda = \sum_{u \in A} p(u)$ and $\beta_n := (1 + \sqrt{2}) \sqrt{\frac{\lambda \log(4/\delta)}{n}} + \frac{1}{3n} \log \frac{4}{\delta}$, the following holds:

$$-\beta_n - \frac{\lambda}{n} \leq R_n - \hat{R}_n \leq \beta_n.$$

GT-UCB Waiting Time [Lagrée et al., 2018]

Waiting Time

Let $\lambda_k = \sum_{u \in A_k} p(u)$ denote the expected number of activations obtained by the first call to influencer k. For $\alpha \in (0, 1)$, the *waiting time* $T_{UCB}(\alpha)$ of GT-UCB represents the round at which the remaining potential of *each* influencer k is smaller than $\alpha\lambda_k$. Formally,

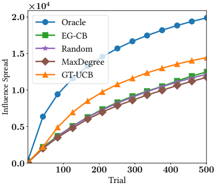
$$T_{UCB}(\alpha) := \min\{t : \forall k \in [K], R_k(t) \leq \alpha\lambda_k\}.$$

Theorem: GT-UCB Waiting Time

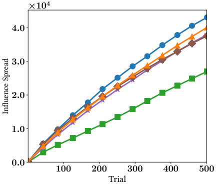
Let $\lambda^{\min} := \min_{k\in[K]} \lambda_k$ and let $\lambda^{\max} := \max_{k\in[K]} \lambda_k$. Assuming that $\lambda^{\min} \geq 13$, for any $\alpha \in \left[\frac{13}{\lambda^{\min}}, 1\right]$, if we define $\tau^* := T^*\left(\alpha - \frac{13}{\lambda^{\min}}\right)$, with probability at least $1 - \frac{2K}{\lambda^{\max}}$ the following holds:

$$T_{\mathrm{UCB}}(\alpha) \leq \tau^* + K\lambda^{\max} \log(4\tau^* + 11K\lambda^{\max}) + 2K.$$

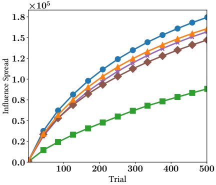
OIMP Regret [Lagrée et al., 2018]



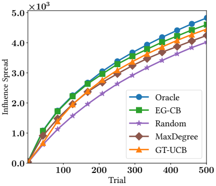
(a) HepPh (WC – $L = 1$)



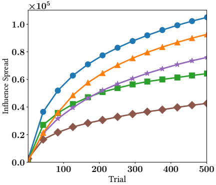
(b) DBLP (WC – $L = 1$)



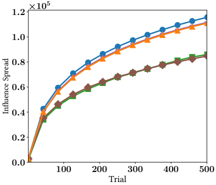
(c) DBLP (WC – $L = 10$)



(d) HepPh (TV – $L = 1$)

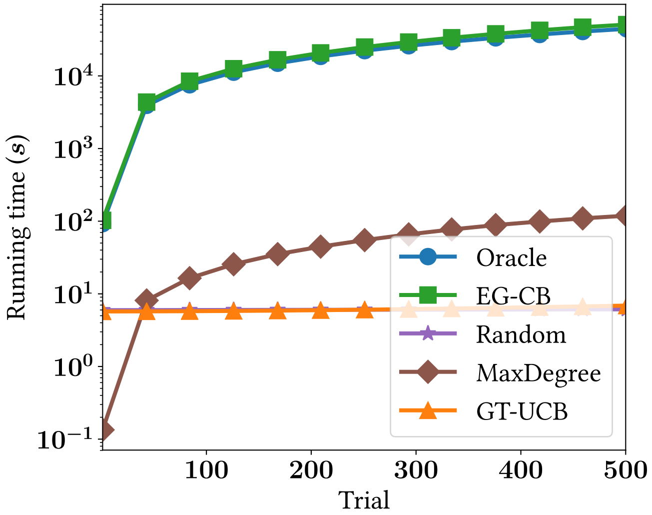


(e) DBLP (TV – $L = 1$)



(f) DBLP (TV – $L = 10$)

OIMP Execution Time [Lagrée et al., 2018]



Model Independent IM [Vaswani et al., 2017]

- Goal: wide applicability by an IM problem formulation based on pairwise reachability probabilities (as in [Lagrée et al., 2018])
  - all stochasticity in the diffusion model $\mathcal{D}$ encoded in a random diffusion vector w $\rightarrow$ each diffusion has a corresponding w sampled from an underlying distribution $\mathcal{P}$
  - online IM: marketer choses seed set $\mathcal{S}$, nature samples $w \sim \mathcal{P}$
  - activated nodes in a diffusion are completely determined by the seed set $\mathcal{S}$ (from a known graph) and $\mathcal{D}(w)$ (unknown)
- Surrogate objective function: based on maximum reachability
- Pairwise influence feedback: observe each node activation along with the seed node responsible for it (note: weaker than edge-level feedback)

Surrogate Objective Function

- for any pair of nodes u, v, the pairwise reachability from u to v, $p^*_{u,v}$, is the probability that v is activated if u is the only seed node
- for a seed set $\mathcal{S}$, $f(\mathcal{S}, v, p^*) = \max_{u \in S} p^*_{u,v}$ is the maximal pairwise reachability from $\mathcal{S}$ to v
- surrogate IM objective function:

$$f(\mathcal{S}, p^*) = \sum_{v \in \mathcal{V}} (\mathcal{S}, v, p^*)(monotone\ and\ submodular)$$

- goal:

$$\tilde{\mathcal{S}} = \arg\max_{\mathcal{S}} f(\mathcal{S}, p^*)$$

(shown to be bounded by below by $1/K$ wrt the optimal IM solution)
- finding $\tilde{\mathcal{S}}$ remains hard, greedy $(1 - 1/e)$ approximation instead
- given p^* (or learning it online as in [Vaswani et al., 2017]), we can obtain an approximate solution for the IM problem w/o knowing the diffusion model

Linear Generalisation

$O(n^2)$ parameters $\rightarrow$ $O(dn)$ parameters

Linear generalisation: for each seed u and node v there exists two d-dimensional feature vectors, x_v (known) and θ_u^* (unknown) s.t. $p^*(u, v)$ is well approximated by $x_v^T \theta_u^*$ (i.e., $\theta_u^* \in \mathcal{R}^d$ are the unknown coefficient vectors that must be learned)

Another LinUCB-like Algorithm

ALGORITHM 8: Diffusion Independent `LinUCB` (`DILinUCB`)

Input: G, C, oracleORACLE, target feature matrix $X \in \mathbb{R}^{d \times n}$, parameters $c, \lambda, \sigma > 0$

Initialize: $\Sigma_{u,0} \leftarrow\in \lambda \boldsymbol{I}_d, b_{u,0} \leftarrow 0, \hat{\theta}_{u,0} \leftarrow 0, \forall v \in V$, and `UCB` $\bar{p}_{u,v}, \forall u, v \in V$

for $t = 1, 2, \ldots, T$ **do**

  Choose $S_t \leftarrow \texttt{ORACLE}(G, C, \hat{p})$

  for $u \in S_t$ **do**

    Get pairwise influence feedback $y_{u,t}$

    $b_{u,t} \leftarrow b_{u,t-1} + X y_{u,t}$

    $\Sigma_{u,t} \leftarrow \Sigma_{u,t-1} + \sigma^{-2} X X^\top$

    $\hat{\theta}_{u,t} \leftarrow \sigma^{-2} \Sigma_{u,t}^{-1} b_{u,t}$

    $\bar{p}_{u,v} \leftarrow \texttt{Proj}_{[0,1]} \left[\langle \hat{\theta}_{u,t} x_v \rangle + c \|x_v\|_{\Sigma_{u,t}^{-1}} \right], \forall v \in V$

  end for

  for $u \notin S_t$ **do**

    $b_{u,t} = b_{u,t-1}$

    $\Sigma_{u,t} = \Sigma_{u,t-1}$

  end for

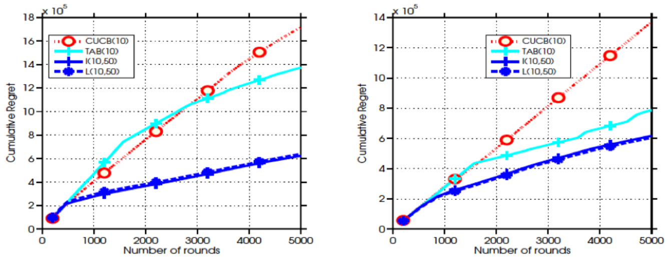
end for

$$R^{\rho\alpha}(T) \leq \frac{2c}{\rho\alpha} n^{\frac{3}{2}} \sqrt{\frac{dKT \log\left(1 + \frac{nT}{d\lambda\sigma^2}\right)}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)}} + \frac{1}{\rho}.$$

Experiments

Some notes:

- reachability from a source to target nodes should be a smooth graph function
- also smoothness assumptions for source features $||\theta^*_{u_1} - \theta^*_{u_2}||_2$ should be "small" if u_1 and u_2 are adjacent $\rightarrow$ Laplacian regularization)



(a) IC Model (b) LT Model

Full Feedback

Full feedback

Activating a seed node at time t, we observe the *entire* propagation in graph

Full Feedback

Full feedback

Activating a seed node at time t, we observe the *entire* propagation in graph

- ☺ Utility function f is adaptive monotone and submodular
- ☹ Not very realistic model
- ☹ Potentially huge delay

Adaptivity Revisited in [Vaswani and Lakshmanan, 2016]

- $\psi_t : \mathcal{V} \rightarrow \{0, 1\}$ **realisation** / **network state** of the influence graph, i.e., set of active nodes at t
- **adaptive policy**: mapping π_k from network states ψ_t to (set of) nodes (empty set included) under budget k
- we write $\pi_k(\psi_t)$ for the node(s) seeded by π_k at $t + 1$ under the network state ψ_t at time t
- seeding $\pi_k(\psi_t)$ leads to the network state $\psi_{t+1} = \psi_t \cup \{\pi_k(\psi_t)\}$
- $f(\pi_k)$ denotes the spread achieved by π_k in a possible world

Offline Policies [Vaswani and Lakshmanan, 2016]

Offline policies

Focus on offline policies, with the objective to maximise in average $f(\pi_k)$ over some candidate possible worlds (the training set). (Note: simply says we can sample possible worlds, as G is known, and we can design the policy offline)

Adaptive IM Optimization Problem

Find the optimal $\pi_{opt,k}$ such that the performance $f(\pi_{opt,k})$ is maximised in average (over the candidate possible worlds).

Equivalence node-level feedback / edge-level feedback

If the diffusion process is allowed to terminate after every seeding step, node-level feedback is equivalent to edge-level feedback w.r.t. marginal gain computation $\rightarrow$ the expected spread function remains adaptive submodular and adaptive monotone.

Main Results in [Vaswani and Lakshmanan, 2016]

How well $\pi_{GA,k}$ (greedy adaptive, sequential) and $\pi_{GNA,k}$ (greedy non adaptive) may do compared to $\pi_{OA,k}$ (optimal adaptive, sequential) ?

Greedy approximations

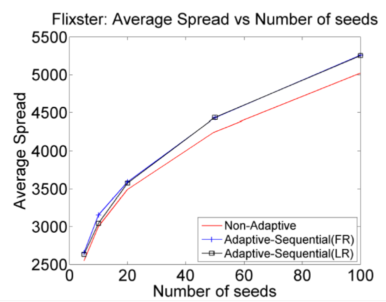
- $f(\pi_{GA,k}) \geq \left(1 - e^{-\frac{1}{\gamma}}\right) \times f(\pi_{OA,k})$
- $f(\pi_{GNA,k}) \geq (1 - \frac{1}{e})^2 \times f(\pi_{OA,k})$

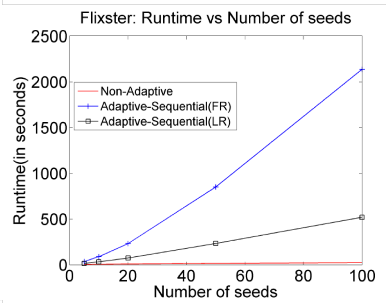
for $\gamma = \left(\frac{e}{e-1}\right)^2$.

Note: assuming perfect marginal gain computation.

Experiments

- 100 possible worlds, spread results averaged over them
- adaptive TIM (RR sets regenerated lazily / LR or eagerly / FR after each seeding step)





Adaptivity Gaps [Chen and Peng, 2019]

Key question: under full-adoption feedback, to what extent an adaptive policy might outperform a non adaptive one ?

Adaptivity gap

For a graph $G = (L, V, p)$, budget k, let $OPT_N(G, k)$ (resp. $OPT_A(G, k)$) the spread of the optimal non-adaptive (resp. adaptive) policy. The adaptivity gap is defined as follows:

$$sup_{G,k} \frac{OPT_A(G, k)}{OPT_N(G, k)}$$

Upper Bounds

Theorem: in-arborescence

When the underline influence graph is an in-arborescence, the adaptivity gap for the IM problem in the IC model with full adoption feedback is at most $\frac{2e}{e-1}$.

Theorem: out-arborescence

When the underline influence graph is an out-arborescence, the adaptivity gap for the IM problem in the IC model with full adoption feedback is at most 2.

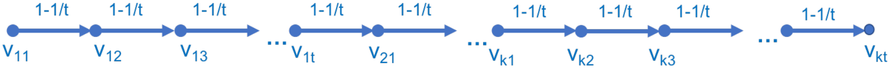
Theorem: bipartite

When the underline influence graph is bipartite (one-directional), the adaptivity gap for the IM problem in the IC model with full adoption feedback is at most $\frac{2e}{e-1}$.

Lower Bound

Theorem: bipartite

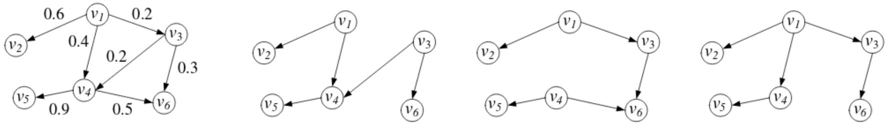
The adaptivity gap for the IM problem in the IC model with full adoption feedback is at least $\frac{e}{e-1}$.



Open question

Adaptivity gap upper bounds for general graphs under full-adoption feedback.

Effective Algorithms for Adaptive Influence Maximization



(a) A social network (b) Possible world w_1 (c) Possible world w_2 (d) Possible world w_3

Figure: A social network and three of its possible worlds $w \sim \mathcal{W}$

- Select k seed nodes in r batches of equal size $b = k/r$
- We observe the influence prop. in w for r rounds in total, once after the selection of each batch
- Our objective is to select r seed set $S_1, S_2, \ldots, S_r$, to maximize the expected influence spread over the choices of $w \sim \mathcal{W}$ (see fig above)
- The *full-feedback* model is adopted
- If $b = k, (i.e., r = 1)$, we resort to the standard IM task

AdaptGreedy efficient algorithm [Han et al., 2018]

Given any non-adaptive IM algorithm able to identify a size-b seed set S_i for the i^{th} residue graph G_i, such that:

$$\mathbb{E}[f_{G_i}(S_i)] \geq (c - \xi_i)\mathrm{OPT}_b(G_i),$$

`AdaptGreedy` achieves a provable approximation guarantee represented by ξ, where:

- $\mathbb{E}[f_{G_i}(S_i)]$ is the expected spread of S_i on G_i
- residue graph G_i is generated by removing from G_{i-1} those nodes that are influenced by S_{i-1}, with $G_1 = G$
- $\mathrm{OPT}_b(G_i)$ is the maximum spread of any size-b seed set on G_i
- $c = 1$ if $b = 1$ and $c = 1 - 1/e$ otherwise

ALGORITHM 9: `AdaptGreedy`

Input: G, k (budget) , r (number of batches)
Output: Seed set $S_1, \ldots, S_r$ (adaptively selected)

$b \leftarrow k/r$ (number of seeds selected at each round)
$G_1 \leftarrow G$
if $r == k$ **then**
    $c \leftarrow 1$
else
    $c \leftarrow 1 - 1/e$
end if
for i = 1 **to** r **do**
    Identify a size-b seed set S_i from G_i, such that:
    $\mathbb{E}[f_{G_i}(S_i)] \geq (c - \xi)\mathrm{OPT}_b(G_i)$
    Observe influence of S_i in G_i
    $G_{i+1} \leftarrow$ Remove all nodes from G_i influenced by S_i
end for
return $S_1, \ldots, S_r$

AdaptGreedy Performance Guarantees

Theorem

Let $\mathcal{G}$ be the set of all possible choices of G_i. Let $\mathbb{P}[\xi_i|G_1,\ldots,G_i]$ be the probability that S_i achieves an approximation ratio of $c-\xi_i$ conditioned on the event that the first i residue graphs are $G_1,\ldots,G_i$, and

$$\xi = \frac{1}{r}\sum_{i=1}^{r}\sum_{G_1\in\mathcal{G}_1,\ldots,G_i\in\mathcal{G}_i}(\xi_i\cdot\mathbb{P}[\xi_i|G_1,\ldots,G_i]\cdot\mathbb{P}[G_1,\ldots,G_i])$$

Then, *the approximation guarantees* of `AdaptGreedy` is at least:

$$\begin{cases} 1-exp(\xi-1), & \text{if } b=1,\\ 1-exp\left(\xi-1+\frac{1}{e}\right), & \text{if } otherwise \end{cases}$$

EPIC: IM with expected approximation

Reverse reachable sets (RR-sets)

An *RR-set* R of G is generated by:

1. First select a node $v \in V$ uniformly at random,
2. Then take the nodes that can reach v in a random graph generated by independently removing each edge $e \in E$ with probability $1 - p(e)$

Then, we get that:

$$\mathbb{E}[f_G(S)] = |V| \underbrace{Cov_{\mathcal{R}}(S)/|\mathcal{R}|}_{\triangleq F_{\mathcal{R}}(S)}$$

where $Cov_{\mathcal{R}}(S)$ denotes the number of RR-sets in CR that overlaps S.

EPIC general framework

1. Start from a small number of *RR-sets*
2. Iteratively increase the *RR-set* number until a satisfactory solution is satisfied

ALGORITHM 10: `EPIC` Algorithm

Input: $G_i, \epsilon_i, \delta_i, b$

Output: Seed set S_i (i^{th} batch)

$\gamma_{i,1} = \frac{\epsilon_i}{6}, \gamma_{i,3} = \frac{\epsilon_i}{2}, \gamma_{i,2} = \frac{\epsilon_i - \gamma_{i,1} - c\gamma_{i,3}}{1+\gamma_{i,1}}$

$\mathcal{Y}_1 = \frac{(4e-8)(1+\gamma_{i,1})(1+\gamma_{i,2})}{\gamma_{i,3}^2} \ln(3/\delta_i)$

$T_{max} = \frac{(8+2\epsilon_i)n_i}{b\epsilon_i^2}\left(\ln\frac{2}{\delta_i} + \ln\binom{n_i}{b}\right), \omega = \left\lceil \log_2\left(\frac{T_{max}}{\mathcal{Y}_1}\right)\right\rceil$

$\mathcal{Y}_2 = 1 + \frac{(4e-8)(1+\gamma_{i,2})}{\gamma_{i,2}^2} \ln\frac{3\omega}{\delta_i}$

Generate a set $\mathcal{R}_1$ of $\mathcal{Y}_1$ random RR sets

repeat

  $\langle S_i, F_{\mathcal{R}_1}(S_i)\rangle \leftarrow$ `MaxCover`$(\mathcal{R}_1, b)$

  if $|\mathcal{R}_1| \cdot F_{\mathcal{R}_1}(S_i) \geq \mathcal{Y}_1$ **then**

    Generate $|\mathcal{R}_1|$ random RR sets in $\mathcal{R}_2$

    Calculate $F_{\mathcal{R}_2}(S_i)$ of S_i in $\mathcal{R}_2$

    if $|\mathcal{R}_2| \cdot F_{\mathcal{R}_2}(S_i) \geq \mathcal{Y}_2$ **and** $F_{\mathcal{R}_1}(S_i) \leq (1+\gamma_{i,1})F_{\mathcal{R}_2}(S_i)$ **then**

      return S_i

    end if

  end if

  $\mathcal{R}_1 = \mathcal{R}_1 \cup \mathcal{R}_2$

until $|\mathcal{R}_1| \geq T_{max}$

return S_i

ALGORITHM 11: `MaxCover` Algorithm

Input: A set $\mathcal{R}$ of random RR set, b

Output: S_i, and the fraction of RR sets in $\mathcal{R}$ covered by S_i

1: $S_i = \emptyset$
2: **for** i = 1 **to** b **do**
3: $\quad v \in \arg\max_{u \in V} Cov_{\mathcal{R}}(S_i \cup \{u\}) - Cov_{\mathcal{R}}(S_i)$
4: $\quad S_i \leftarrow S_i \cup \{v\}$
5: **end for**
6: **return** $\langle S_i, Cov_{\mathcal{R}}(S_i)/|\mathcal{R}|\rangle$

EPIC Performance Guarantees

Theorem

With a probability of at least $1 - \delta_i$, `EPIC` returns a seed set S_i satisfying

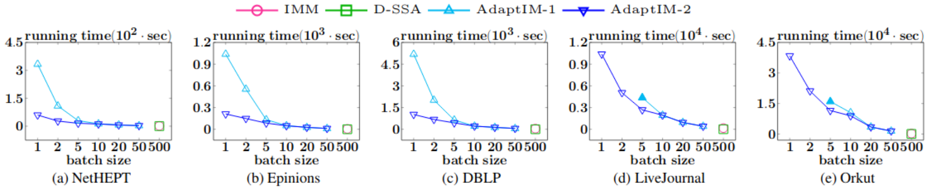
$$\mathbb{E}[f_{G_i}(S_i)] \geq (c - \epsilon_i)\mathrm{OPT}_b(G_i)$$

for any G_i. In addition, the expected time complexity of `EPIC` is

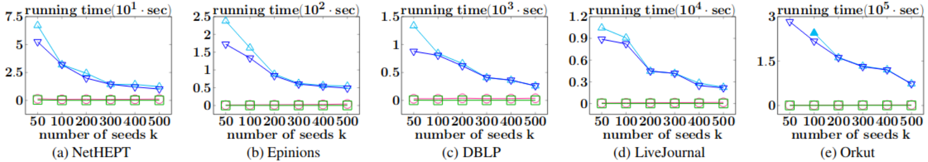
$$O\left(\left(b\log(n_i) + \log\left(\frac{1}{\delta_i}\right)\right)(m_i + n_i)/\epsilon^2\right)$$

where m_i and n_i are the numbers of nodes and edges of G_i, respectively.

Empirical Analysis: Running Time Vs. Seed and Batch size

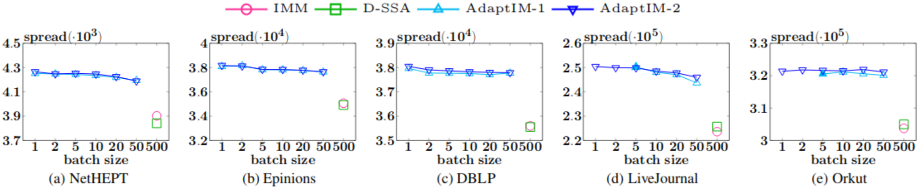


Running time vs. batch size

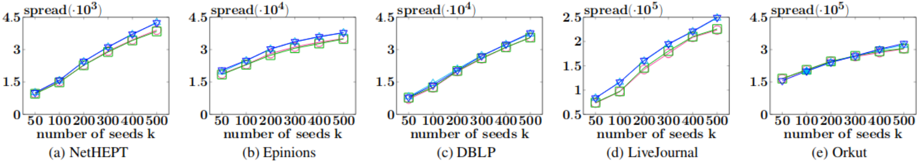


Running time vs. seed size

Empirical Analysis: Spread Vs. Seed and Batch size



Spread vs. batch size



Spread vs. seed size

Full feedback Vs. Partial feedback

Full feedback

Activating a seed node at time t, we observe the *entire* propagation in graph

Partial feedback

Activating a seed node at time t, we observe the propagation in graph for d time slots:

Full feedback Vs. Partial feedback

Full feedback

Activating a seed node at time t, we observe the *entire* propagation in graph

- ☺ Utility function f is adaptive monotone and submodular
- ☹ Not very realistic model
- ☹ Potentially huge delay

Partial feedback

Activating a seed node at time t, we observe the propagation in graph for d time slots:

Full feedback Vs. Partial feedback

Full feedback

Activating a seed node at time t, we observe the *entire* propagation in graph

- ☺ Utility function f is adaptive monotone and submodular
- ☹ Not very realistic model
- ☹ Potentially huge delay

Partial feedback

Activating a seed node at time t, we observe the propagation in graph for d time slots:

- ☺ Allows us to select to select seed nodes at any intermediate stage
- ☹ Utility function f is **NOT** adaptive submodular

Adaptive IM with Partial Feedback [Yuan and Tang, 2017]

The next seed is selected *iff* the following condition is satisfied:

$$\frac{f(\mathcal{S}|\psi_{[r]})}{|V \setminus O_{[r]}|} \geq \alpha$$

where,

- $\alpha \in [0, 1]$: control parameter
  - $\alpha = 1$: full-feedback
  - $\alpha = 0$: zero-feedback (standard IM)
- $\psi_{[r]}$: observations made at round r
- $O_{[r]}$: set of nodes whose activation probability is zero at round r.

Uniform cost

The node with the maximum expected marginal gain given existing seeds $\mathcal{S}$ and partial realization $\psi_{[r]}$ is selected as seed node at each round:

$$v = \arg\max_{u \in V \setminus \mathcal{S}} \Delta_f(u|\psi_{[r]})$$

ALGORITHM 12: α-Greedy policy π^u

Input: $\mathcal{G}, B, 0 \leq \alpha \leq 1$

Output: $\mathcal{S}$

$\mathcal{S} \leftarrow \emptyset$; $r \leftarrow 0$

$v = \arg\max_{u \in V \setminus \mathcal{S}} \Delta_f(u|\psi_{[r]})$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{v\}$; $B \leftarrow B - 1$

while $B \geq 0$ **do**

  $r \leftarrow r + 1$

  if $\frac{f(\mathcal{S}|\psi_{[r]})}{|V \setminus O_{[r]}|} \geq \alpha$ **then**

    $v = \arg\max_{u \in V \setminus \mathcal{S}} \Delta_f(u|\psi_{[r]})$

    $\mathcal{S} \leftarrow \mathcal{S} \cup \{v\}$; $B \leftarrow B - 1$

  else

    wait one time slot; update $\psi_{[r]}$

  end if

end while

return $\mathcal{S}$ (final set of influenced nodes)

Adaptive IM with Partial Feedback [Yuan and Tang, 2017]

The next seed is selected *iff* the following condition is satisfied:

$$\frac{f(\mathcal{S}|\psi_{[r]})}{|V \setminus O_{[r]}|} \geq \alpha$$

where,

- $\alpha \in [0, 1]$: control parameter
  - $\alpha = 1$: full-feedback
  - $\alpha = 0$: zero-feedback (standard IM)
- $\psi_{[r]}$: observations made at round r
- $O_{[r]}$: set of nodes whose activation probability is zero at round r.

Non-uniform cost

The node with the maximum expected marginal gain given existing seeds $\mathcal{S}$ and partial realization $\psi_{[r]}$ is selected as seed node at each round:

$$v = \arg\max_{u \in V \setminus \mathcal{S}} \frac{\Delta_f(u|\psi_{[r]})}{c_u}$$

ALGORITHM 13: α-Greedy policy with non-uniform cost π^{nu}

Input: $\mathcal{G}, B, 0 \leq \alpha \leq 1$

Output: $\mathcal{S}$

$\mathcal{S} \leftarrow \emptyset$; $r \leftarrow 0$

$v = \arg\max_{u \in V \setminus \mathcal{S}} \frac{\Delta(u|\psi_{[r]})}{c_u}$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{v\}$; $B \leftarrow B - c_v$

while $B \geq 0$ **do**

  $r \leftarrow r + 1$

  if $\frac{f(\mathcal{S}|\psi_{[r]})}{|V \setminus O_{[r]}|} \geq \alpha$ **then**

    $v = \arg\max_{u \in V \setminus \mathcal{S}} \frac{\Delta(u|\psi_{[r]})}{c_u}$

    if $B - c_v < 0$ **then**

      break

    else

      $\mathcal{S} \leftarrow \mathcal{S} \cup \{v\}$; $B \leftarrow B - c_v$

    end if

  else

    wait one time slot; update $\psi_{[r]}$

  end if

end while

return $\mathcal{S}$ (final set of influenced nodes)

Adaptive IM with Partial Feedback Guarantees

Theorem: Performance Bound of π^u (uniform cost)

The expected cascade of policy π^u under the IC model is bounded by:

$$f(\pi^u) \geq \alpha \left(1 - e^{-\frac{1}{\alpha}}\right) f(\pi^*).$$

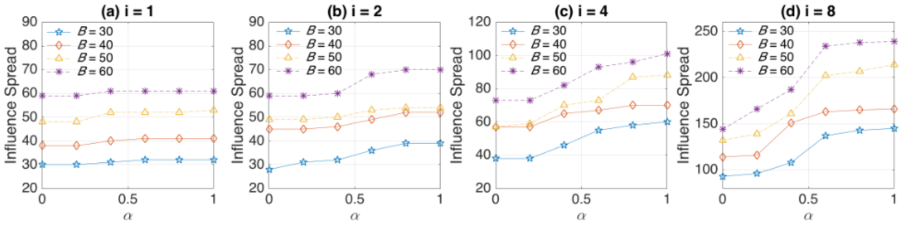
Under *full-feedback* model ($\alpha = 1$), we get: $f(\pi^u) \geq \underbrace{(1 - 1/e)}_{\simeq 63\%} f(\pi^*)$.

Theorem: Performance Bound of π^{nu} (non-uniform cost)

The expected cascade of policy π^{nu} under the IC model is bounded by:

$$f(\pi^{nu}) \geq \alpha \left(1 - e^{-\frac{1}{\alpha}\frac{B-\bar{c}}{B}}\right) f(\pi^*), \quad \text{where } \bar{c} \triangleq \max_{u \in V} c_u.$$

Empirical analysis



Experimental setup

- NetHEPT network ($|V| = 15233, |E| = 62774$)
- Edge influence probability is randomly assigned: $i \times \{0.01, 0.001\}$
- Budget B ranges from 30 to 60
- The cost of each node is randomly assigned from $[1, 10]$

Full Feedback Vs. Myopic Feedback

Full feedback

Activating a seed node at time t, we observe the *entire* propagation in graph

Myopic feedback

Activating a seed node at time t, we only observe the status (active or not) of the neighbors of the seed nodes at time $t+1$

Full Feedback Vs. Myopic Feedback

Full feedback

Activating a seed node at time t, we observe the *entire* propagation in graph

- ☺ Utility function f is adaptive monotone and submodular
- ☹ Not very realistic model
- ☹ Potentially huge delay

Myopic feedback

Activating a seed node at time t, we only observe the status (active or not) of the neighbors of the seed nodes at time $t + 1$

Full Feedback Vs. Myopic Feedback

Full feedback

Activating a seed node at time t, we observe the *entire* propagation in graph

- ☺ Utility function f is adaptive monotone and submodular
- ☹ Not very realistic model
- ☹ Potentially huge delay

Myopic feedback

Activating a seed node at time t, we only observe the status (active or not) of the neighbors of the seed nodes at time $t + 1$

- ☺ Realistic model
- ☹ Utility function f is **NOT** adaptive submodular

Myopic Adaptive Influence Maximisation [Salha et al., 2018]

Modified utility function

Given a finite *horizon* T, the proposed utility function is defined as:

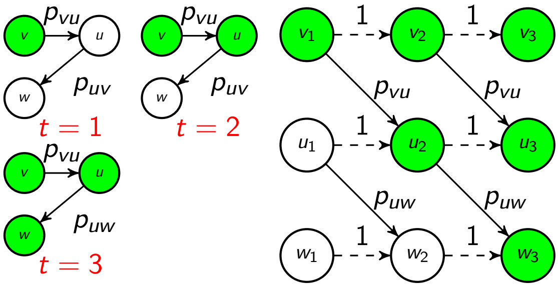
$$\tilde{f}(\mathcal{S}, \phi) \triangleq \sum_{t=1}^{T} |\sigma_t(\mathcal{S}, \phi)|,$$

where $\sigma_t(\mathcal{S}, \phi)$ represents the set of active nodes at time t.

Modified IC model

Each active node has multiple opportunities to influence its inactive neighbors.

Layered Graph Representation - $\mathcal{G}^L$



Lemma

For seed set $\mathcal{S}$ (with time indices) and realization ϕ, it holds that:

$$\tilde{f}_{\mathcal{G}}(\mathcal{S}, \phi) = f_{\mathcal{G}^L}(\mathcal{S}, \phi)$$

Representation Analysis

Definition: Time function

Time function $\mathcal{T} : \Psi \rightarrow \{1, \ldots, T\}$ returns, for a particular ψ, the largest time index from observed nodes and edges, and 1 if $\psi = \emptyset$

Definition: Marginal gain

The **marginal gain of choosing** v **as a seed node**, having observed ψ with $\mathcal{T}(\psi) = t$, and for the ground truth realization ϕ of the network, is:

$$\delta_{\phi}(v|\psi) \triangleq \tilde{f}_{\mathcal{G}}(dom(\psi) \cup \{v_t\}, \phi) - \tilde{f}_{\mathcal{G}}(dom(\psi), \phi).$$

Representation Analysis

Lemma: Marginal gain

The marginal gain of choosing v as a seed node on $\mathcal{G}^L$, under partial realization ψ with $\mathcal{T}(\psi) = t$, is given by:

$$\delta_\phi(v|\psi) = f_{\mathcal{G}^L}([\mathcal{L}_t \cap dom(\psi)] \cup \{v_t\}, \phi) - f_{\mathcal{G}^L}(\mathcal{L}_t \cap dom(\psi), \phi).$$

Lemma: Submodularity property

- For partial realizations $\psi \subseteq \psi'$ with $\mathcal{T}(\psi) = \mathcal{T}(\psi') = t$ and any $v \in V$, we get $\delta_\phi(v|\psi) \geq \delta_\phi(v|\psi')$.
- For partial realizations $\psi \subseteq \psi'$ with $\mathcal{T}(\psi) < \mathcal{T}(\psi')$ and any $v \in V \setminus dom(\psi')$, we get $\delta_\phi(v|\psi) \geq 1 + \delta_\phi(v|\psi')$.

Myopic Adaptive Greedy Strategy Guarantees

Optimization Problem:

$$\pi^* \in \arg\max_{\pi} \tilde{f}_{avg}(\pi) \triangleq \mathbb{E}_{\Phi}[\tilde{f}_{\mathcal{G}}(E(\pi, \Phi), \Phi)] \quad \text{s.t.} \quad |E(\pi, \phi)| \leq k, \forall \phi.$$

Theorem: Performance Bound

Adaptive greedy policy π^{g} obtains at least $(1 - 1/e)$ of the value of the best policy for the AIM problem under the *modified* IC model with myopic feedback:

$$\tilde{f}_{avg}(\pi^{g}) \geq \underbrace{(1 - 1/e)}_{\simeq 63\%} \tilde{f}_{avg}(\pi^*).$$

ALGORITHM 14: Myopic adaptive greedy policy

Input: $\mathcal{G}, T$

$\psi \leftarrow \emptyset,\ \mathcal{S} \leftarrow \emptyset$
for t = 1 **to** T **do**
    Compute $\Delta_{\tilde{f}}(v|\psi), \forall v \in \mathcal{V} \setminus \mathcal{S}$
    Select $v^* \in \arg\max_{v \in \mathcal{V} \setminus \mathcal{S}} \Delta_{\tilde{f}}(v|\psi)$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{v^*\}$
    Update ψ observing (one-step) myopic feedback
    $\mathcal{S} \leftarrow \mathcal{S} \cup dom(\psi)$
end for
return $\mathcal{S}$ (final set of influenced nodes)

Modified IC Hypotheses

Lemma: Utility function $\tilde{f}$ under standard IC model

The utility function $\tilde{f}$ is not adaptive submodular under the *standard* IC model with myopic feedback.

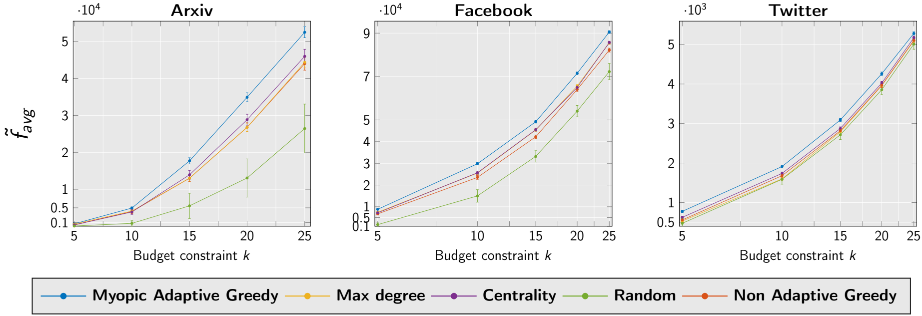
Lemma: Non-Progressive Adaptive Submodular IM

Forcing active nodes to remain active throughout the process constitutes a *necessary condition* to verify the adaptive submodularity property of:

i) $\tilde{f}_{\mathcal{G}}$ in the modified IC model with myopic feedback;

ii) $f_{\mathcal{G}}$ in the standard IC model with full-adoption feedback.

Empirical Results



Adaptivity Gaps under myopic Feedback [Peng and Chen, 2019]

Key question: under myopic feedback, to what extent an adaptive policy might outperform a non adaptive one ?

Adaptivity gap

For all graphs $G = (L, V, p)$, budgets k, let $OPT_N(G, k)$ (resp. $OPT_A(G, k)$) the spread of the optimal non-adaptive (resp. adaptive) policy. The adaptivity gap is defined as follows:

$$sup_{G,k} \frac{OPT_A(G, k)}{OPT_N(G, k)}$$

Adaptivity Gap: Lower and Upper Bounds [Peng and Chen, 2019]

Theorem (Upper bound)

Under the IC model with myopic feedback, the adaptivity gap for the influence maximization problem is at most 4.

Theorem (Lower bound)

Under the IC model with myopic feedback, the adaptivity gap for the influence maximization problem is at least $\frac{e}{e-1}$.

Greedy vs. Optimal Adaptive Policy [Peng and Chen, 2019]

Theorem

Both greedy and adaptive greedy are $\frac{1}{4}(1-\frac{1}{3})$-approximate to the optimal adaptive policy under the IC model with myopic feedback. (conjecture from [Golovin and Krause, 2011]).

Theorem

The approximation ratio for greedy and adaptive greedy is no better than $\frac{e^2+1}{(e+1)^2} \approx 0.606$ w.r.t. the optimal adaptive policy under the IC model with myopic feedback.

Note: $\frac{e^2+1}{(e+1)^2} \approx 0.606 < (1-\frac{1}{e}) \approx 0.632$.

Theorem

Under the IC model with myopic feedback the approximation ratio of adaptive greedy is at most that of the non-adaptive greedy.

General Feedback

General feedback

Activating a seed node at time t, we observe the propagation in graph for d steps, for $d \in \mathbb{N} \cup \{\infty\}$ and fixed:

- Allows to select seed nodes at predefined intermediate stages
- Recall utility function f is **NOT** adaptive submodular unless $d \neq \infty$
  - $d = 1$ represents the myopic feedback model
  - $d = \infty$ represents the full (adoption) feedback model

Adaptive IM with General Feedback [Tong and Wang, 2019]

(k, d)-AIM

Given a budget k, and an observation stage of d steps,

- repeat the following: select one seed node, wait for d rounds of diffusion, and observe the diffusion ...
- ... until k nodes are selected
- wait for final diffusion to end, output number of activated nodes

Policy search

Policy

A policy π maps a status (S, ϕ) to a set of nodes to be seeded, for S denoting the set of current active nodes and ϕ being a *realization* giving the live/dead state of edges that have been observed.

Objective

For k and d given, find a policy π such that the expected number of active nodes, denoted $F(\pi, k, d)$, is maximized.

Adaptive IM with General Feedback [Tong and Wang, 2019]

(π, k, d)-process

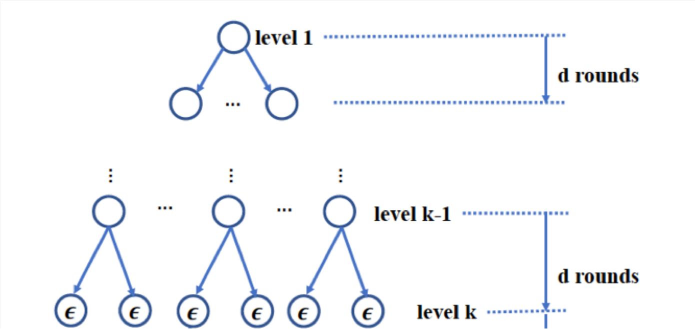
Given a budget k, and an observation stage of d steps,

- starting with status $(S, \phi) = (\varnothing, \phi_\varnothing)$
- repeat the following step k times:
  - select and activate seed node $\pi(S, \phi)$
  - wait for and observe d rounds of diffusion
  - update S as set of current active nodes
  - update ϕ as current realization
- wait for final diffusion to end, output number of activated nodes

Decision Tree

Decision tree

An adaptive seeding process can be seen as a decision tree, where node = seed set, edge = status.



Greedy Policy

Greedy policy π_g

Given a status (S, ϕ), the greedy policy π_g selects the node that maximizes the marginal gain conditioned on (S, ϕ):

$$\pi_g(S, \phi) = \arg\max_{v} \Delta f_\infty(S, v, \phi)$$

where

- S denotes the set of current active nodes
- ϕ is the *realization* i.e. state of edges that have been observed
- $\Delta f_\infty(S, v, \phi) = \sum_{\phi \prec \psi, \psi \in \Psi} Pr[\psi|\phi] \times \Delta_\infty(S, v, \psi)$ is the expected marginal profit after diffusion terminates ($d = \infty$), Ψ = full realisations (possible worlds)
- $\Delta_\infty(S, v, \psi) = |Active_\infty(S \cup \{v\}, \psi)| - |Active_\infty(S, \psi)|$ is the marginal increase due to v after diffusion terminates ($d = \infty$)

Regret Ratio

Given a status (S, ϕ), suppose we need to select one seed maximizing the number of active nodes after t rounds (bounded time horizon t)

- Option 1: seed immediately based on (S, ϕ), to achieve a marginal profit $\max_v \Delta f_\infty(S, v, \phi)$
- Option 2: wait for diffusion to terminate, reaching some possible status (S_*, ϕ_*) and then select v by

$$\arg\max_v \Delta f_\infty(S_*, v, \phi_*),$$

to achieve a marginal profit

$$\sum_{(S_*,\phi_*)} Pr[\phi_*|\phi] \times \max \Delta f_\infty(S_*, v, \phi_*)$$

(t, d)-regret ratio for (S, ϕ)

Regret ratio $\alpha(S, \phi) = \frac{result\ \ of\ \ option\ \ 2}{result\ \ of\ \ option\ \ 1}$

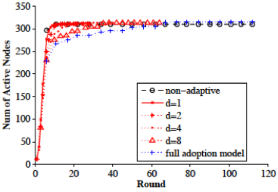
Main Result in [Tong and Wang, 2019]

For each policy π, we have that

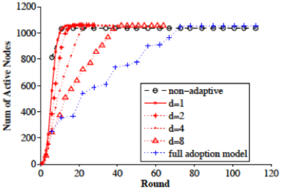
$$F(\pi_g, k, d) \geq (1 - e^{-1/\alpha}) \times F(\pi, k, d)$$

where $\alpha = \max_{(S,\phi)} \alpha(S, \phi)$ over all (S, ϕ) in the (π_g, k, d)-process / corresponding decision tree.

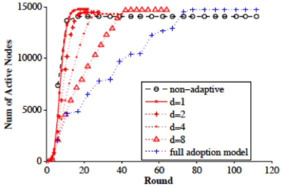
Empirical Analysis - Different Feedback Models (d)



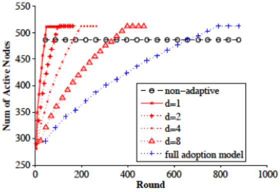
(a) Higgs with $k = 5$



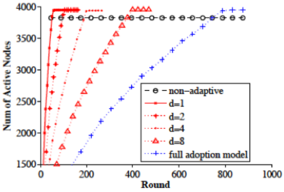
(b) Hepph with $k = 5$



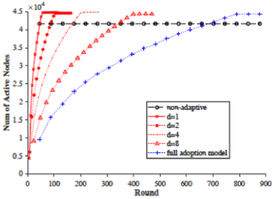
(c) DBLP with $k = 5$



(d) Higgs with $k = 50$



(e) Hepph with $k = 50$



(f) DBLP with $k = 50$

Multi-Round Influence Maximization [Sun et al., 2018]

An advertiser's marketing campaign may contain multiple rounds to promote one product [a]

- ✓ **Non-adaptive MRIM**: determine the seed sets for all rounds at the beginning
- ✓ **Adaptive MRIM**: select seed sets adaptively based on the propagation in the previous rounds

[a] KDD 2018: https://www.youtube.com/watch?v=FzDId0_78b0

Triggering Diffusion Model

✓ Discrete time diffusion model $t = 0, 1, \ldots$

✓ At time $t = 0$:

- Seed set $\mathcal{S}_0$ is selected
- Each node $v \in V$ selects a random *triggering set* $T(v)$ according to some distribution over subsets of its **in**-neighbors

✓ At time $t \geq 1$:

- An inactive node v becomes active if at least one node in $T(v)$ is active at $t - 1$

✓ The diffusion ends when no more nodes activated in a time steps.

Triggering diffusion model $\equiv$ to propagation in *live-edge graph*

Given sets $\{T(v)\}_{v \in V}$, we get the *live-edge graph* $L = (V, E(L))$:

☞ $E(L) = \{(u, v) | v \in V, u \in T(v)\}$ (*live* edges)

Multi-Round Triggering (MRT) diffusion model

- MRT includes T *independent rounds*, r
- At each round $r \in [T]$ diffusion starts from a separate seed set S_r
- $\mathcal{S} \triangleq \{(v, r) | v \in S_r\}$ represents the seed set at round r
- The diffusion at each round follows the standard *triggering* model
- The **budget** at each round is equal to k

Influence spread in MRT model

$$\rho(\mathcal{S}) = \rho(\cup_{r=1}^{T} \mathcal{S}_r) \triangleq \mathbb{E}\left[\left|\bigcup_{r=1}^{T} \Gamma(L_r, S_t)\right|\right]$$

where $\Gamma(L_r, S_t)$ is the active nodes at the end of round r.

✓ The expectation is over the distribution of live-edge graphs $L_1, \ldots, L_T$.

Non-Adaptive MRIM optimization task

Problem formulation

Given:

i) Graph $\mathcal{G} = (V, E)$

ii) **Triggering set** distribution for every node

iii) Number of **rounds** T

iv) Each-round **budget** k

our **objective** is to find seed set $\mathcal{S}^*$ such that:

$$\mathcal{S}^* = \mathcal{S}_1^* \cup \mathcal{S}_2^* \cup \cdots \cup \mathcal{S}_T^* = \underset{\mathcal{S}:|\mathcal{S}_t| \leq k, \forall r \in [T]}{\arg\max} \rho(S)$$

- ✓ Find the T seed sets all at once before the propagation starts
- ✓ Classical IM is a special case of MRIM with $T = 1$

Cross-Round setting

Let $\mathcal{V}_r = \{(v, r) | v \in V\}$ (all possible nodes at round r) and $\mathcal{V} \triangleq \bigcup_{r=1}^{T} \mathcal{V}_r$

Cross-Round Greedy Policy

1. Candidate space $\mathcal{C} = \mathcal{V}$
2. At every (greedy) time step:
    - Pick $(v, r) \in \mathcal{C}$ with the maximum gain without replacement
    - **IF** budget of round r exhausts, $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{V}_r$

Theorem: Performance bound

For every $\epsilon > 0$ and $\ell > 0$, with probability at least $1 - 1/n^{\ell}$, the output $\mathcal{S}^0$ of `CR-Greedy` satisfies:

$$\rho(\mathcal{S}^0) \geq \left(\frac{1}{2} - \epsilon\right) \rho(\mathcal{S}^*),$$

if $R = \lceil 31k^2 T^2 n \log(3kn^{\ell+1})/\epsilon^2 \rceil$ as input.

ALGORITHM 15: CR-Greedy: Cross-Round Greedy Algorithm

Input: $\mathcal{G}, T, k, R$ (triggering set distributions)
Output: $\mathcal{S}^0$

1: $\mathcal{S}^0 \leftarrow \emptyset; \mathcal{C} \leftarrow \mathcal{V}$
2: $c_1, c_2, \ldots, c_t \leftarrow 0$
3: **for** $i = 1$ **to** kT **do**
4: $\quad \forall (v, r) \in \mathcal{C} \setminus \mathcal{S}^0$, estimate $\rho(\mathcal{S}^0 \cup \{(v, r)\})$ simulating diffusion process R times
5: $\quad (v_i, r_i) \leftarrow \arg\max_{(v,r)\in\mathcal{C}\setminus\mathcal{S}^0} \hat{\rho}(\mathcal{S}^0 \cup \{(v, r)\})$
6: $\quad \mathcal{S}^0 \leftarrow \mathcal{S}^0 \cup \{(v_i, r_i)\}; c_{r_i} \leftarrow c_{r_i} + 1$
7: $\quad$ **if** $c_{r_i} \geq k$ **then**
8: $\quad\quad \mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{V}_{r_i}$
9: $\quad$ **end if**
10: **end for**
11: **return** $\mathcal{S}^0$

Within-Round setting

Let $\mathcal{V}_r = \{(v, r) | v \in V\}$ (all possible nodes at round r) and $\mathcal{V} \triangleq \bigcup_{r=1}^{T} \mathcal{V}_r$

Within-Round Greedy Policy

1. Seed nodes are selected by **round**-by-**round**
2. Only after selected all k seed nodes at round r, we greedily select seed nodes for the next round $r + 1$.

Theorem: Performance bound

For every $\epsilon > 0$ and $\ell > 0$, with probability at least $1 - 1/n^{\ell}$, the output $\mathcal{S}^0$ of `WR-Greedy` satisfies:

$$\rho(\mathcal{S}^0) \geq \left(1 - e^{-\left(1-\frac{1}{e}\right)} - \epsilon\right) \rho(\mathcal{S}^*),$$

if $R = \lceil 31k^2 n \log(2kn^{\ell+1} T)/\epsilon^2 \rceil$ as input.

ALGORITHM 16: WR-Greedy: Within-Round Greedy Algorithm

Input: $\mathcal{G}, T, k, R$ (triggering set distributions)
Output: $\mathcal{S}^0$

$\mathcal{S}^0 \leftarrow \emptyset; \mathcal{C} \leftarrow \mathcal{V}$
for $r = 1$ **to** T **do**
    for $i = 1$ **to** k **do**
        $\forall (v, r) \in \mathcal{C} \setminus \mathcal{S}^0$, estimate $\rho(\mathcal{S}^0 \cup \{(v, r)\})$ simulating diffusion process R times
        $(v, r) \leftarrow \arg\max_{(v,r) \in \mathcal{C} \setminus \mathcal{S}^0} \hat{\rho}(\mathcal{S}^0 \cup \{(v, r)\})$
        $\mathcal{S}^0 \leftarrow \mathcal{S}^0 \cup \{(v, r)\}$
    end for
end for
return $\mathcal{S}^0$

CR-Greedy Vs. WR-Greedy

Performance Guarantee - Approximation ratio

- CR-Greedy: $(\frac{1}{2} - \epsilon)$
- WR-Greedy: $0.46 - \epsilon$

Running Time

- The running time of WR-Greedy is improved by a factor of T^3, compared to CR-Greedy

Adaptive Multi-Round Influence Maximization

✓ Let S_r to be the seeds selected at round r, then (S_r, r) is called *item*

Utility function

$$f(\{(S_1, 1), \ldots, (S_r, r)\} | \phi) \triangleq \left| \bigcup_{i=1}^{r} \Gamma(L_i^{\phi}, S_i) \right|,$$

where L_i^{ϕ} is the live-edge graph of round i.

Adaptive Multi-Round IM problem

Discover best policy π^* such that:

$$\pi^* = \arg\max_{\pi \in \Pi_{T,k}} f_{avg}(\pi) = \mathbb{E}_{\Phi}[f(E(\pi, \Phi), \Phi)],$$

with $E(\pi, \Phi)$ to be the set of items selected under policy π.

Adaptive Multi-Round Influence Maximization

Theorem: Performance bound

For every $\epsilon > 0$ and $\ell > 0$, with probability at least $1 - 1/n^\ell$, the policy π^{ag} satisfies:

$$f_{avg}(\pi^{ag}) \geq \left(1 - e^{-\left(1-\frac{1}{e}\right)} - \epsilon\right) f_{avg}(\pi^*),$$

if $R = \lceil 31k^2 n \log(2kn^{\ell+1}T)/\epsilon^2 \rceil$ as input.

Running time

Total running time for T−round `AdaGreedy`: $\mathcal{O}(k^3 \ell T n^2 m \log(nT)/\epsilon^2)$

ALGORITHM 17: `AdaGreedy`: Adaptive Greedy for Round r

Input: $\mathcal{G}, T, k, R$ (triggering set distributions), A_{r-1} active node set by round $r-1$

Output: S_r, A_r

1: $S_r \leftarrow$ `MC-Greedy`(G, A_{r-1}, k, R)
2: Observe the propagation of S_r
3: Update activated nodes A_r
4: **return** $(S_r, r), A_r$

Maximizing the expected marginal gain $\Delta((S_r, r)|\psi)$

$$\equiv$$

Weighted influence maximization task in which we treat nodes in A_{r-1} with weight 0 and other nodes with weight 1

Comparing Strategies

Non-adaptive Strategies

- `SG`: Select Tk seed nodes using greedy alg, then allocates the first k as S_1, and so on
- `SG-R`: Select k seed nodes, and reuse the same k seeds at each round
- `CR-Greedy`: Cross round greedy algorithm
- `CR-IMM`: Cross round using IMM algorithm [Tang et al., 2015]
- `WR-Greedy`: Within round using greedy algorithm
- `WR-IMM`: Within round using IMM algorithm

Adaptive Strategies

- `AdaGreedy`: Adaptive greedy algorithm
- `AdaIMM`: Adaptive based on IMM algorithm

Empirical Analysis: Influence Spread on NetHEPT

Method/Simulations	Round				
	1	2	3	4	5
SG (R = 10000)	290.1 [288.8, 291.4]	505.7 [504.0, 507.3]	688.6 [686.6, 690.4]	868.2 [866.2, 870.2]	1027.3 [1025.2, 1029.4]
SG-R (R = 10000)	289.5 [288.2, 290.8]	516.3 [514.6, 518.0]	714.0 [712.0, 716.0]	884.9 [882.7, 887.1]	1042.0 [1039.7, 1044.2]
E-WR-Greedy (R = 10000)	290.7 [289.4, 292.0]	528.9 [527.2, 530.6]	738.8 [736.9, 740.8]	930.2 [928.0, 932.3]	1097.6.9 [1095.3, 1099.8]
WR-IMM (R = 10000)	290.9 [289.7, 292.3]	532.8 [531.1, 534.5]	745.3 [743.2, 747.3]	930.1 [928.0, 932.2]	1093.1 [1090.8, 1095.3]
CR-Greedy (R = 10000)	267.8 [266.5, 269.1]	528.7 [527.2, 530.4]	730.4 [728.5, 732.4]	938.5 [933.7, 937.8]	1121.3 [1119.0, 1123.5]
CR-IMM (R = 10000)	283.0 [281.7, 284.2]	517.4 [515.7, 519.2]	721.9 [720.0, 723.9]	931.6 [929.4, 933.7]	1129.7 [1127.7, 1131.9]
AdaGreedy (R = 150)	288.3 [276.7, 299.7]	533.4 [519.4, 547.3]	758.1 [743.6, 772.7]	960.1 [943.9, 976.3]	1141.5 [1123.7, 1160.0]
AdaIMM (R = 150)	**291.8** [281.3, 302.4]	**544.4** [531.6, 557.2]	**761.8** [746.6, 776.9]	**965.8** [949.7, 982.0]	**1146.3** [1129.1, 1163.5]

"High Energy Physics Theory" section of arXiv from 1991 to 2003:
$|V| = 15,233$, $|E| = 62,774$

Empirical Analysis: Influence Spread on Flixster

Method/Simulations	Round				
	1	2	3	4	5
SG (R = 10000)	558.8 [557.3, 560.3]	936.2 [934.5, 937.9]	1200.3 [1198.4, 1202.2]	1437.9 [1435.9, 1439.9]	1631.5 [1629.5, 1633.6]
SG-R (R = 10000)	**559.8** [558.3, 561.3]	949.2 [947.4, 951.0]	1262.6 [1260.6, 1264.5]	1530.3 [1528.2, 1532.4]	1764.9 [1762.7, 1767.0]
E-WR-Greedy (R = 10000)	557.8 [556,3 559.2]	976.5 [974.8, 978,3]	1304.2 [1302.2, 1306.1]	1587.8 [1585.8, 1580.8]	1840.0 [1838.0, 1842.1]
WR-IMM (R = 10000)	558.1 [556.7, 559.6]	967.5 [965.7, 969.3]	1306.9 [1306.9, 1308.9]	1599.1 [1597.1, 1601.1]	1836.4 [1834.3, 1838.5]
CR-Greedy (R = 10000)	519.9 [518.4, 521.5]	948.6 [946.7, 950.5]	1295.7 [1293.7, 1297.7]	1593.5 [1591.4, 1595.5]	1863.8 [1861.7, 1865.9]
CR-IMM (R = 10000)	521.7 [521.7, 523.2]	935.8 [933.1, 937.0]	1275.3 [1273.3, 1277.3]	1585.9 [1583.8, 1588.0]	1865.1 [1863.1, 1867.3]
AdaGreedy (R = 100)	557.8 [539.8, 580.5]	977.8 [956.2, 999.1]	1307.7 [1291.1, 1324.3]	1605.2 [1588.1, 1622.3]	1861.8 [1845.3, 1878.3]
AdaIMM (R = 100)	555.5 [542.3, 568.6]	**977.9** [962.9, 993.0]	**1317.2** [1300.8, 1333.5]	**1613.2** [1594.2, 1632.1]	**1872.5** [1853.0, 1891.9]

Social movie discovery service[1]: $(|V| = 29,357, |E| = 212,614)$

[1] www.flixster.com

Adaptive IM in Summary

- ☺ Adaptive policies can bring important benefits
- ☺ May be more realistic / closer to real-life diffusion scenarios
- ☺ No other alternatives in bandit settings
- ☹ Harder to design and analyse
- ☹ Sometimes properties such as adaptive submodularity no longer exploitable
- ☹ May be slower

Open Issues in Bandit AIM Setting

- Other bandit approaches besides LinUCB (e.g., Thompson Sampling-based)
- Other feedback models (full-bandit)
- Dependency on IM-Oracles

Open Issues in Full-Knowledge Setting (1)

Some key generic questions:

- When an adaptive policy might outperform a non adaptive one ?
- By how much an adaptive policy may outperform a non adaptive one ?

Can be addressed in ...

- Theory: adaptivity gaps → some are not yet tight (e.g., myopic observations), others are yet to be established (e.g., full-adoption feedback for general graphs)
- Practice: adaptivity gains → e.g., how adaptive greedy relates to non-adaptive greedy, are there other algorithms besides greedy exhibiting a better gain ?

Open Issues in Full-Knowledge Setting (2)

Other (more general) models besides IC and studied feedback types (myopic, full, partial / general feedback)

- E.g, the edges we get to observe may depend on the context / status $\rightarrow$ diffusion (maximize spread) vs. feedback (maximize observations) trade-off when seeding nodes
- Privacy issues limiting observations
- Finite time horizon $\rightarrow$ leading to adaptivity in the seeding batches (seed later to observe more, but lose rounds . . .)
- Beyond round by round: e.g., seeding stages triggered by events
- Other diffusion models (e.g., LT, general LT/IC), continuous-time models

Practical applicability

How to bring the theory closer to the practical needs of marketing / information diffusion scenarios ?

- Generalisation models are necessary in bandit IM problems; context too
- May need more flexible bandit formulations: e.g., volatile bandits, ways to learn both the graph structure and activation probabilities
- Model independence may be beneficial in both bandit and full-knowledge problems
- Scalable algorithms for spread estimation
- Gain from going adaptive especially when imperfect marginal spread estimations $\rightarrow$ how to capture that tradeoff

Thank You

Capsule Bio: Bogdan Cautis

- Bogdan Cautis, Professor in CS at University of Paris-Sud 11
- Received a PhD in 2007 from INRIA France, was Associate Professor at Telecom ParisTech between 2007 and 2013, visiting research at Huawei Noah's Ark Lab HK between 2015 and 2017
- Doing research in the broad areas of data management and data mining, publishing regularly in top tier conferences (ICDM, WWW, KDD, IJCAI, ECML/PKDD, SDM, CIKM, ICDE, VLDB, SIGMOD, PODS, ICDT, etc) and journals (TODS, JCSS, TKDD, TKDE, Springer DAMI, etc)
- Homepage: `https://www.lri.fr/~cautis/`

Capsule Bio: Silviu Maniu

- Associate Professor in CS at University of Paris-Sud 11
- Received a PhD in 2012 from Télécom Paris, was Postdoctoral Researcher at University of Hong Kong between 2012 and 2014 and Researcher at Huawei's Noah's Ark Lab between 2014 and 2015.
- Research focused on uncertain and social data management and mining
- Homepage: `http://silviu.maniu.info/`

Capsule Bio: Nikolaos Tziortziotis

- Research Scientist at Tradelab Programmatic platform, France
- Received a PhD in 2015 from the Department of Computer Science & Engineering of the University of Ioannina, Greece. was a researcher at University of Paris-Sud 11 (2018), and postdoctoral researcher at École Polytechnique (2015–2018).
- Research interests span the broad areas of machine learning and data mining, with focus on reinforcement learning, Bayesian learning, and real-time bidding.
- Homepage: `https://ntziortziotis.github.io/`

References I

Bakshy, E., Hofman, J. M., Mason, W. A., and Watts, D. J. (2011).
Everyone's an influencer: Quantifying influence on Twitter.
In *WSDM*.

Chen, W. and Peng, B. (2019).
On adaptivity gaps of influence maximization under the independent cascade model with full adoption feedback.
CoRR, abs/1907.01707.

Chen, W., Wang, Y., and Yuan, Y. (2013).
Combinatorial multi-armed bandit: General framework, results and applications.
In *ICML*.

References II

Golovin, D. and Krause, A. (2011).
Adaptive submodularity: Theory and applications in active learning and stochastic optimization.
J. Artif. Int. Res., 42(1):427–486.

Han, K., Huang, K., Xiao, X., Tang, J., Sun, A., and Tang, X. (2018).
Efficient algorithms for adaptive influence maximization.
PVLDB, 11(9).

Kempe, D., Kleinberg, J., and Tardos, E. (2003).
Maximizing the spread of influence through a social network.
In *SIGKDD*, pages 137–146. ACM.

Lagrée, P., Cappé, O., Cautis, B., and Maniu, S. (2017).
Effective large-scale online influence maximization.
In *ICDM*.

References III

Lagrée, P., Cappé, O., Cautis, B., and Maniu, S. (2018).
Algorithms for online influencer marketing.
ACM Trans. Knowl. Discov. Data, 13(1).

Lattimore, T. and Szepesvári, C. (2019).
Bandit Algorithms.
Cambridge University Press.

Lei, S., Maniu, S., Mo, L., Cheng, R., and Senellart, P. (2015).
Online influence maximization.
In *SIGKDD*.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978).
An analysis of approximations for maximizing submodular set functions—i.
Mathematical Programming, 14(1):265–294.

References IV

Peng, B. and Chen, W. (2019).
Adaptive influence maximization with myopic feedback.
CoRR, abs/1905.11663.

Salha, G., Tziortziotis, N., and Vazirgiannis, M. (2018).
Adaptive submodular influence maximization with myopic feedback.
In *ASONAM*.

Sun, L., Huang, W., Yu, P. S., and Chen, W. (2018).
Multi-round influence maximization.
In *KDD*.

Tang, Y., Shi, Y., and Xiao, X. (2015).
Influence maximization in near-linear time: A martingale approach.
In *SIGMOD*.

References V

Tong, G. and Wang, R. (2019).
Adaptive influence maximization under general feedback models.
CoRR, abs/1902.00192.

Vaswani, S., Kveton, B., Wen, Z., Ghavamzadeh, M., Lakshmanan, L. V. S., and Schmidt, M. (2017).
Model-independent online learning for influence maximization.
In *ICML*.

Vaswani, S. and Lakshmanan, L. V. S. (2015).
Influence maximization with bandits.
CoRR, abs/1503.00024.

Vaswani, S. and Lakshmanan, L. V. S. (2016).
Adaptive influence maximization in social networks: Why commit when you can adapt?
CoRR, abs/1604.08171.

References VI

Wen, Z., Kveton, B., Valko, M., and Vaswani, S. (2017).
Online influence maximization under independent cascade model with semi-bandit feedback.
In *NIPS*.

Yuan, J. and Tang, S. (2017).
No time to observe: Adaptive influence maximization with partial feedback.
In *IJCAI*.