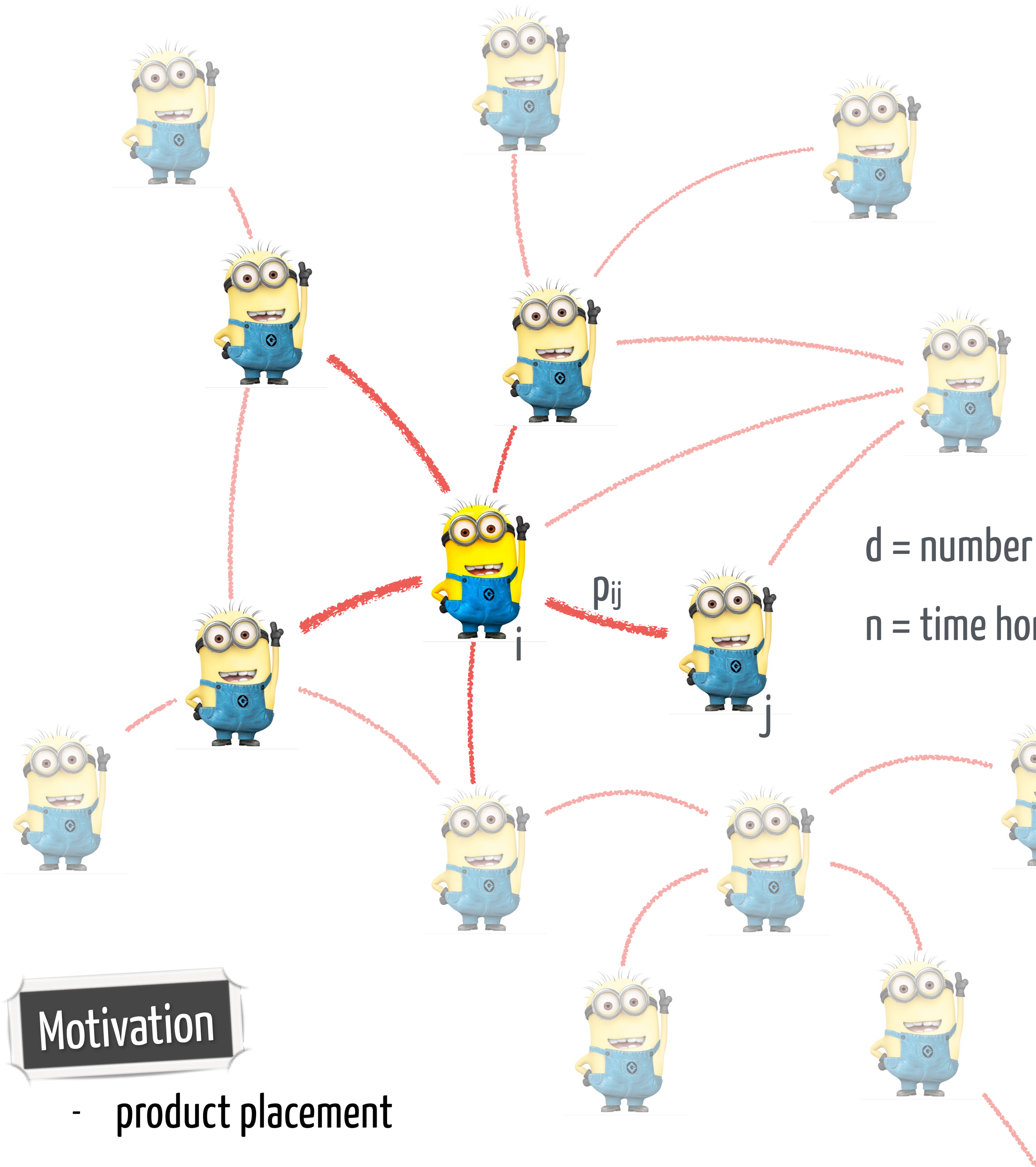
Revealing graph bandits for maximizing local influence

ALEXANDRA CARPENTIER and MICHAL VALKO

carpentier@math.uni-potsdam.de and michal.valko@inria.fr

Universität Potsdam



d = number of nodes

n = time horizon

Motivation

- product placement
- information campaign
- local model of influence and unknown graphs
- information gathered through "likes" or promotional codes

Detectable dimension

- number of nodes we can efficiently extract in less than n rounds
- function D controls number of nodes given a gap

$$D(\Delta) \stackrel{\text{def}}{=} |\{i \le d : r^\circ_\star - r^\circ_i \le \Delta\}|$$

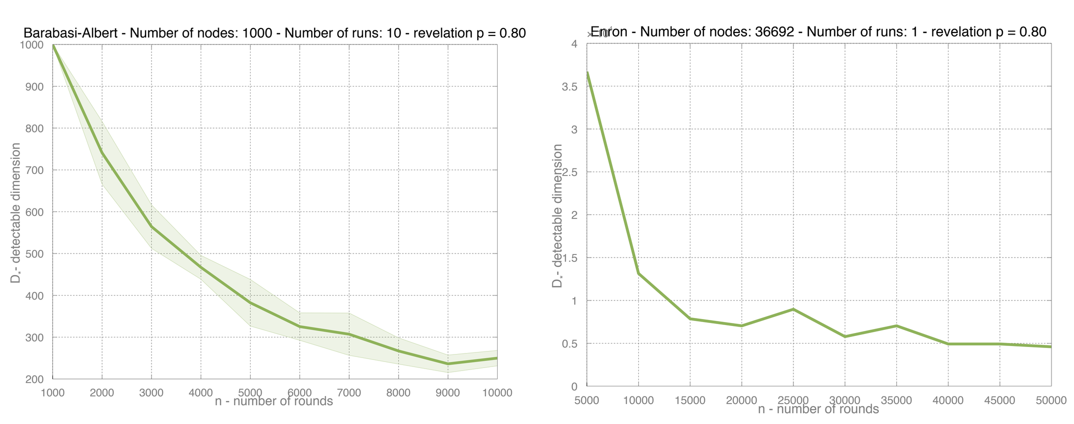
- $D(r) = d$ for $r \ge r_\star$ and $D(0)$ = number of most influenced nodes
- **Detectable dimension** $D_\star = D(\Delta_\star)$
- Detectable gap $\Delta_\star$ constants coming from the analysis and the Bernstein inequality

$$\Delta_\star \stackrel{\text{def}}{=} 16\sqrt{\frac{r^\circ_\star d\log(nd)}{T_\star}} + \frac{80d\log(nd)}{T_\star}$$

- Detectable horizon $T_\star$, smallest integer s.t. $T_\star r^\circ_\star \ge \sqrt{D_\star n r^\circ_\star}$
- Equivalently: $D_\star$ corresponding to smallest $T_\star$ such that

$$T_\star r^\circ_\star \ge \sqrt{D\left(16\sqrt{\frac{r^\circ_\star d\log(nd)}{T_\star}} + \frac{80d\log(nd)}{T_\star}\right) n r^\circ_\star}$$

- For (easy, structured) star graphs $D_\star = 1$ even for small n (big gain)
- For (difficult) empty graphs $D_\star = d$ even for large n (no gain)
- In general: $D_\star$ roughly decreases with n and it is small when D decreases quickly
- For n large enough $D_\star$ is the number of the most influences nodes
- Example: $D_\star$ for Barabási–Albert model & Enron graph as a function of n



Setting

Unknown $(p_{ij})_{ij}$ — (symmetric) probability of influences

In each time step t = 1, ..., n

learner picks a node k_t

environment reveals the set of influenced node S_{k_t}

Select influential people = Find the strategy maximising

$$L_n = \sum_{t=1}^{n} |S_{k_t,t}|$$

Our solution

BAndit REvelator: 2-phase algorithm

- global exploration phase
  - super-efficient exploration
  - linear regret — needs to be short!
  - extracts $D_\star$ nodes
- bandit phase
  - uses a minimax-optimal bandit algorithm
  - GraphMOSS is a little brother of MOSS
  - has a "square root" regret on $D_\star$ nodes
- **$D_\star$ realizes the optimal trade-off!**
  - different from exploration/exploitation tradeoff

Guarantees

Upper bound on the regret of BARE

$$\mathbb{E}[R_n] \le C\min\left(r_\star n, D_\star r_\star + \sqrt{r_\star n D_\star}\right)$$

Matching lower bound

Algorithm

GraphMOSS

Input
d: the number of nodes
n: time horizon

Initialization
Sample each arm twice
Update $\widehat{r}_{k,2d}$, $\widehat{\sigma}_{k,2d}$, and $T_{k,2d} \leftarrow 2$, for $\forall k \le d$

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

$C_{k,t} \leftarrow 2\widehat{\sigma}_{k,t}\sqrt{\frac{\max(\log(n/(dT_{k,t})),0)}{T_{k,t}}} + \frac{2\max(\log(n/(dT_{k,t})),0)}{T_{k,t}}$, for $\forall k \le d$

$k_t \leftarrow \arg\max_k \widehat{r}_{k,t} + C_{k,t}$

Sample node k_t and receive $|S_{k_t,t}|$

Update $\widehat{r}_{k,t+1}$, $\widehat{\sigma}_{k,t+1}$, and $T_{k,t+1}$, for $\forall k \le d$

end for

Definitions

The number of expected influences of node k is by definition

$$r_k = \mathbb{E}[|S_{k,t}|] = \sum_{j\le d} p_{k,j}$$

Oracle strategy always selects the best

$$k^\star = \arg\max_k \mathbb{E}\left[\sum_{t=1}^{n}|S_{k,t}|\right] = \arg\max_k n r_k$$

Expected regret of the oracle strategy

$$\mathbb{E}[L^\star_n] = n r_\star$$

Expected regret of any adaptive strategy unaware of $(p_{ij})_{ij}$

$$\mathbb{E}[R_n] = \mathbb{E}[L^\star_n] - \mathbb{E}[L_n]$$

Baseline

- We **only** receive |S| instead of S
- Can be mapped to **multi-arm** bandits
  - rewards are 0, ..., d
  - variance bounded with r_{k_t}
- We adapt **MOSS** to GraphMOSS
- Regret upper bound of GraphMOSS

$$\mathbb{E}[R_n] \le U\min\left(r_\star n, r_\star d + \sqrt{r_\star n d}\right)$$

- matching lower bound

Algorithm

BARE - BAndit REvelator

Input
d: the number of nodes
n: time horizon

Initialization
$T_{k,t} \leftarrow 0$, for $\forall k \le d$
$\widehat{r^\circ_{k,t}} \leftarrow 0$, for $\forall k \le d$
$t \leftarrow 1$, $\widehat{T}_\star \leftarrow 0$, $\widehat{D}_{\star,t} \leftarrow d$, $\widehat{\sigma}_{\star,1} \leftarrow d$

Global exploration phase

while $t\left(\widehat{\sigma}_{\star,t} - 4\sqrt{d\log(dn)/t}\right) \le \sqrt{\widehat{D}_{\star,t} n}$ **do**

Influence a node at random (choose k_t uniformly at random) and get $S_{k_t,t}$ from this node

$\widehat{r^\circ_{k,t+1}} \leftarrow \frac{t}{t+1}\widehat{r^\circ_{k,t}} + \frac{d}{t+1}S_{k_t,t}(k)$

$\widehat{\sigma}_{\star,t+1} \leftarrow \max_{k'}\sqrt{\widehat{r^\circ_{k',t+1}} + 8d\log(nd)/(t+1)}$

$w_{\star,t+1} \leftarrow 8\widehat{\sigma}_{\star,t+1}\sqrt{\frac{d\log(nd)}{t+1}} + \frac{24d\log(nd)}{t+1}$

$\widehat{D}_{\star,t+1} \leftarrow \left|\left\{k : \max_{k'}\widehat{r^\circ_{k',t+1}} - \widehat{r^\circ_{k,t+1}} \le w_{\star,t+1}\right\}\right|$

$t \leftarrow t+1$

end while

$\widehat{T}_\star \leftarrow t.$

Bandit phase

Run minimax-optimal bandit algorithm on the $\widehat{D}_{\star,\widehat{T}_\star}$ chosen nodes (e.g., Algorithm 1)

Experiments