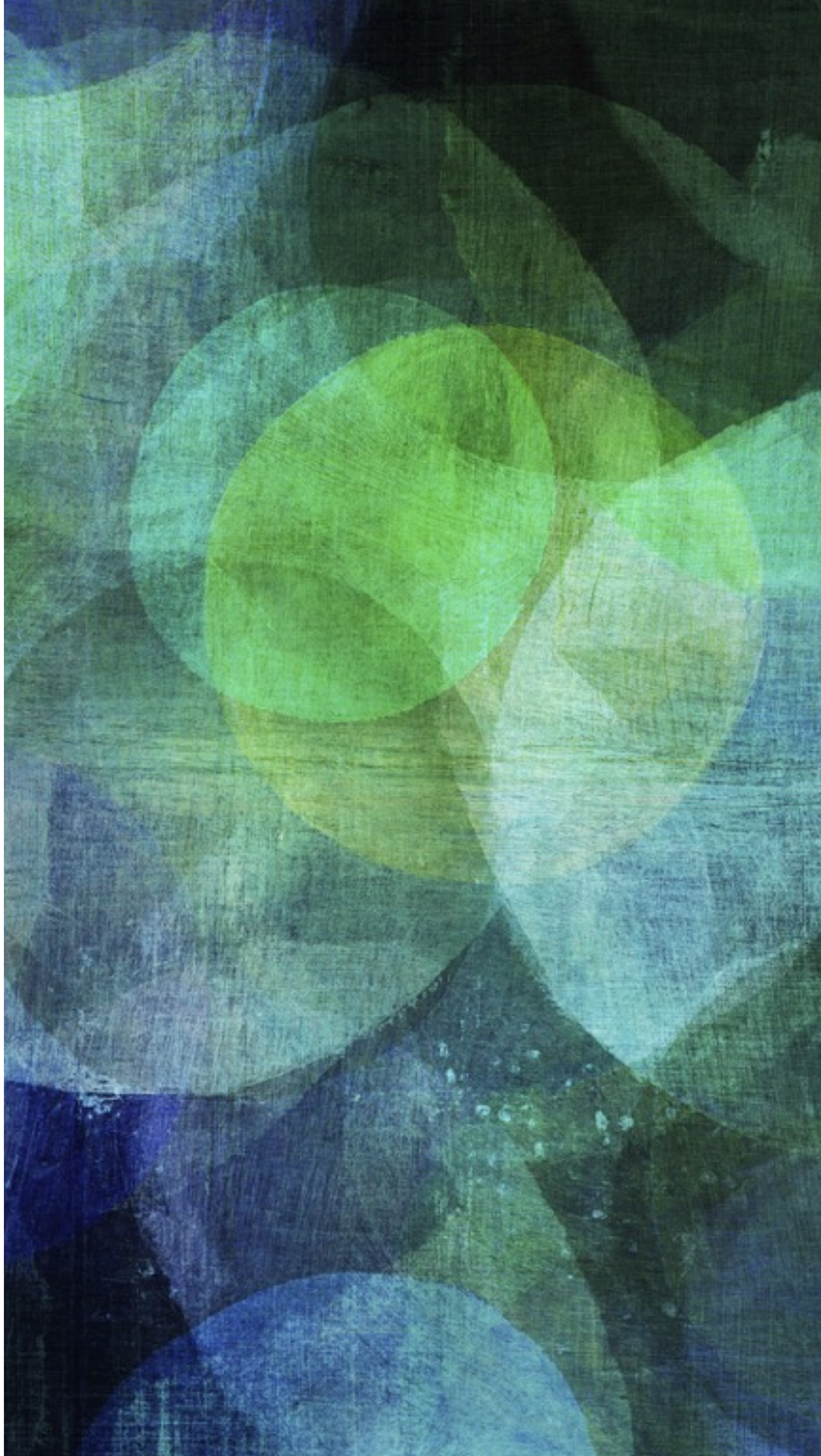
Carpentier, MV: **Revealing Graph Bandits for Maximising Local Influence**, AISTATS 2016

Wen, Kveton, MV: **Influence Maximization with Semi-Bandit Feedback**, (arXiv:1605.06593)

INFLUENCE MAXIMISATION

looking for the influential nodes **while** exploring the graph



Influence the influential!

JULY 18, 2016

Religion

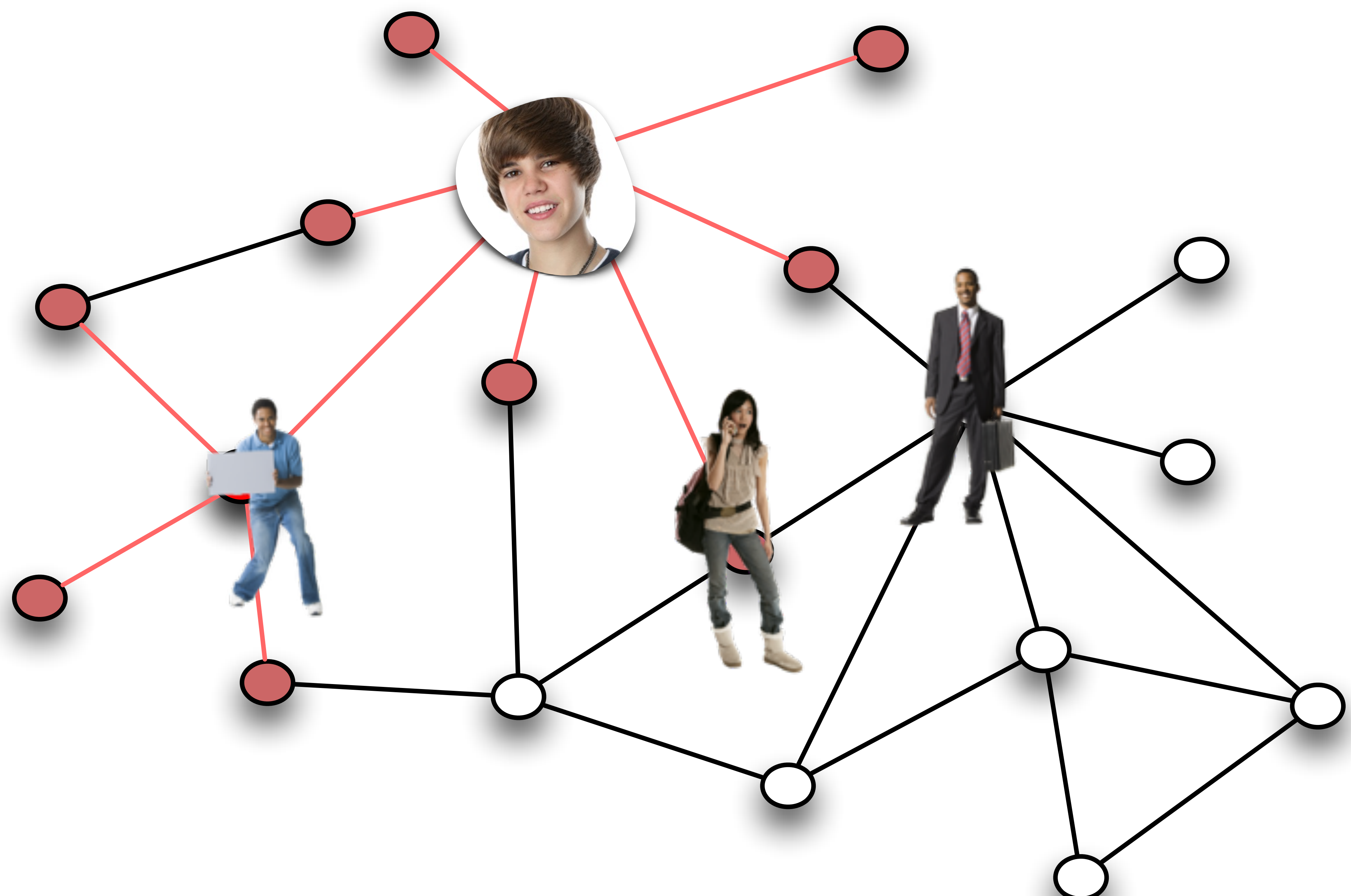
March 26, 2017

Politics

September 1, 2009

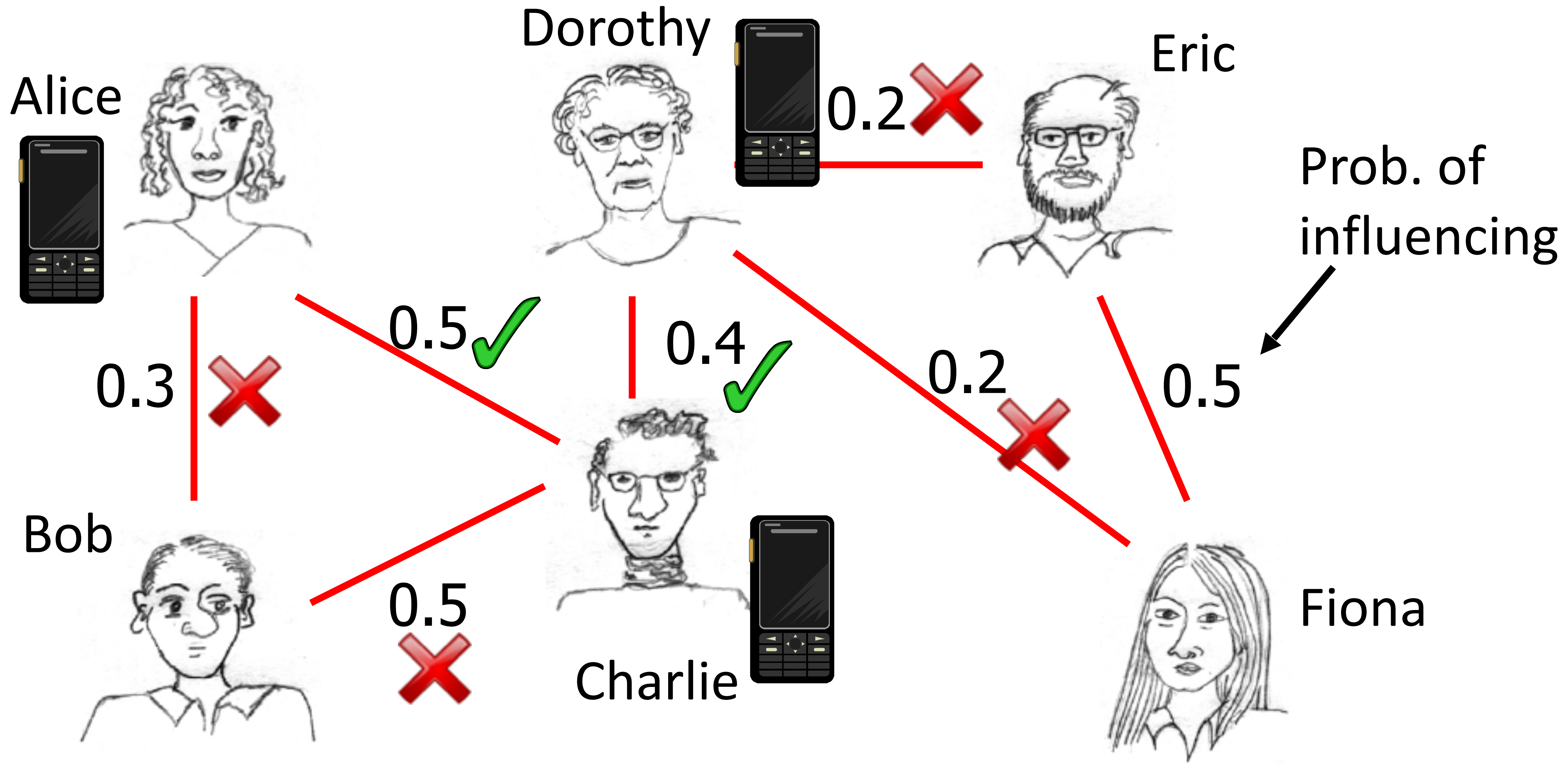
Culture

338 ET 200



$$F(S) = \text{spread}$$

EXAMPLE: INFLUENCE IN SOCIAL NETWORKS [KEMPE, KLEINBERG, TARDOS KDD '03]

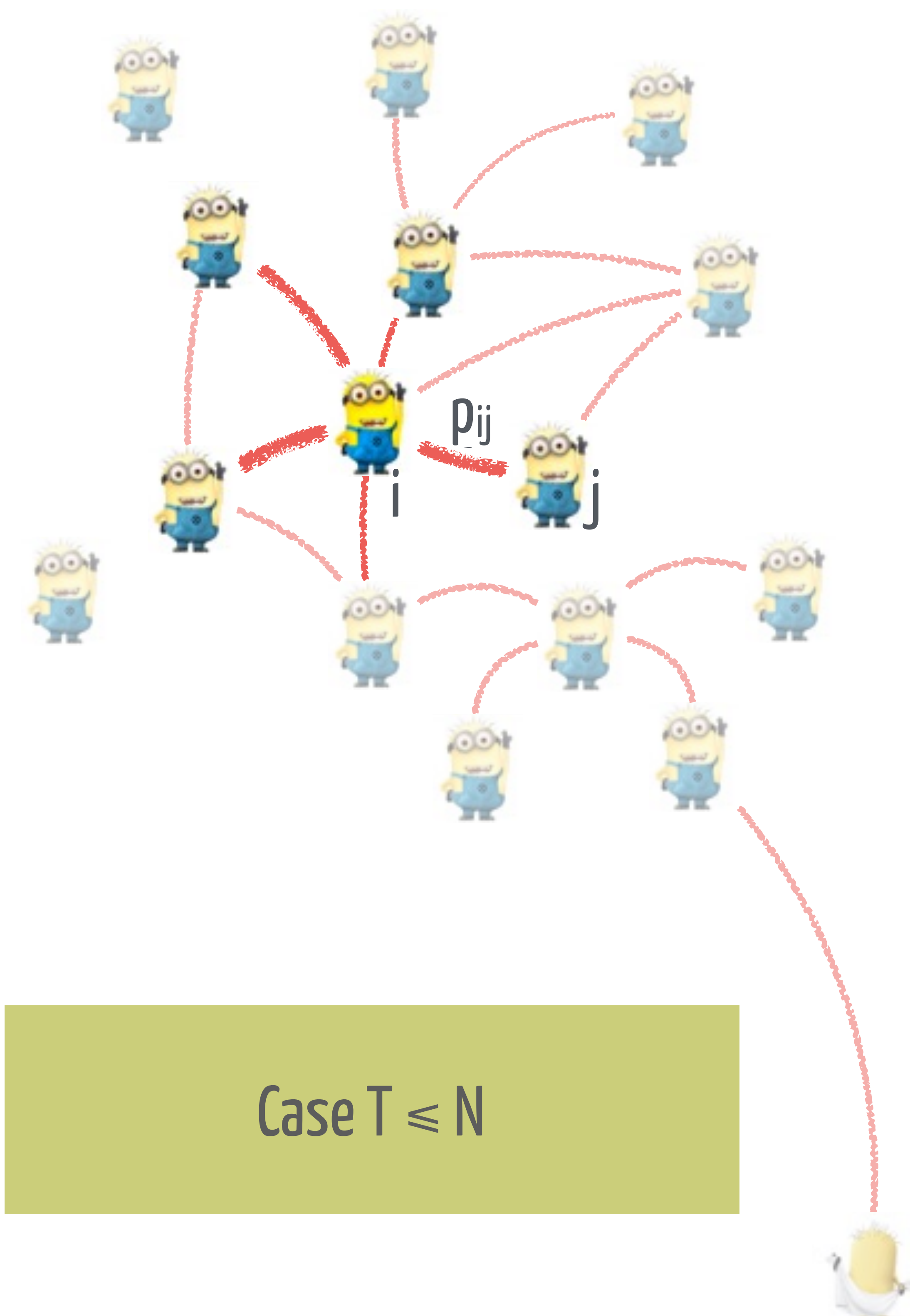


Who should get free cell phones?

V = {Alice,Bob,Charlie,Dorothy,Eric,Fiona}

F(A) = Expected number of people influenced when targeting A

REVEALING BANDITS FOR LOCAL INFLUENCE



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

In each time step t = 1, …., T

learner picks a node k_t

environment **reveals** the set of influenced node S_{k_t}

Select influential people = Find the strategy maximising

$$L_T = \sum_{t=1}^{T} |S_{k_t,t}|$$

Why this is a bandit problem?

Case T ⩽ N

PERFORMANCE CRITERION

The number of expected influences of node **k** is by definition

$$r_k = \mathbb{E}\left[|S_{k,t}|\right] = \sum_{j \leq N} p_{k,j}$$

Oracle strategy always selects the best

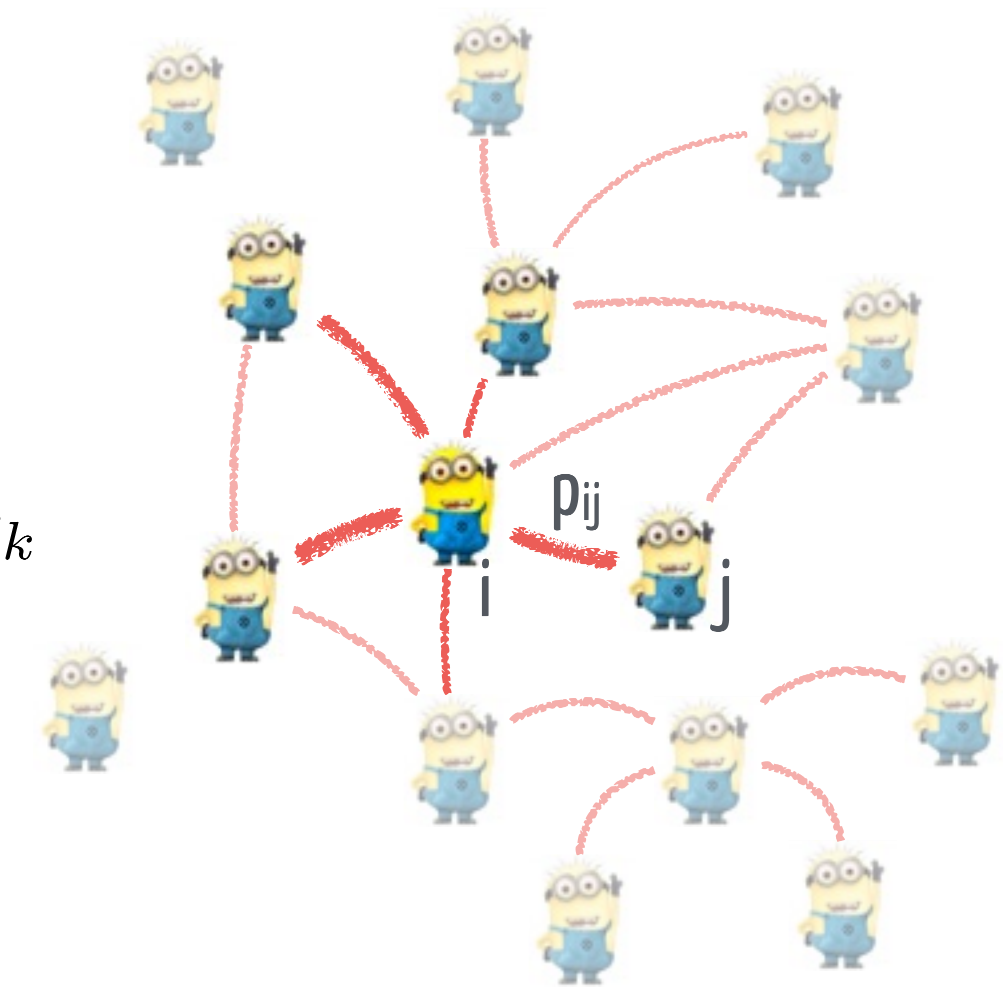
$$k^\star = \arg\max_k \mathbb{E}\left[\sum_{t=1}^{T} |S_{k,t}|\right] = \arg\max_k T r_k$$

Expected regret of the oracle strategy

$$\mathbb{E}\left[L_T^\star\right] = T r_\star$$

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

$$\mathbb{E}\left[R_T\right] = \mathbb{E}\left[L_T^\star\right] - \mathbb{E}\left[L_T\right]$$



BASELINE

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

$$\mathbb{E}\left[R_T\right] \leq U \min\left(r_\star T, r_\star N + \sqrt{r_\star T N}\right)$$

each node at least once

unlearnable case T ≤ N

- matching lower bound

Crash course on stochastic bandits?

GRAPHMOSS FOR THE RESTRICTED SETTING

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 $\forall k \leq d$

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

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

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

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 $\forall k \leq d$

end for

BACK TO THE REAL SETTING

- Can we actually do better?
  - Well, not really.....
  - Minimax optimal rate is still the same
- But the bad cases are somehow pathological
  - isolated nodes
  - uncorrelated being influenced and being influential
  - Barabási–Albert etc tell us that the real-world graphs are not like that
- Let's think of some measure of difficulty
  - to define some non-degenerate cases
  - ideas?

DETECTABLE DIMENSION

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

$$D(\Delta) = |\{i \le N : r_\star^\circ - r_i^\circ \le \Delta\}|$$

- D(r) = N for r≥ r* and D(0) = number of most influenced nodes
- **Detectable dimension D*** = D(Δ*)
- Detectable gap Δ* constants coming from the analysis and the Bernstein inequality

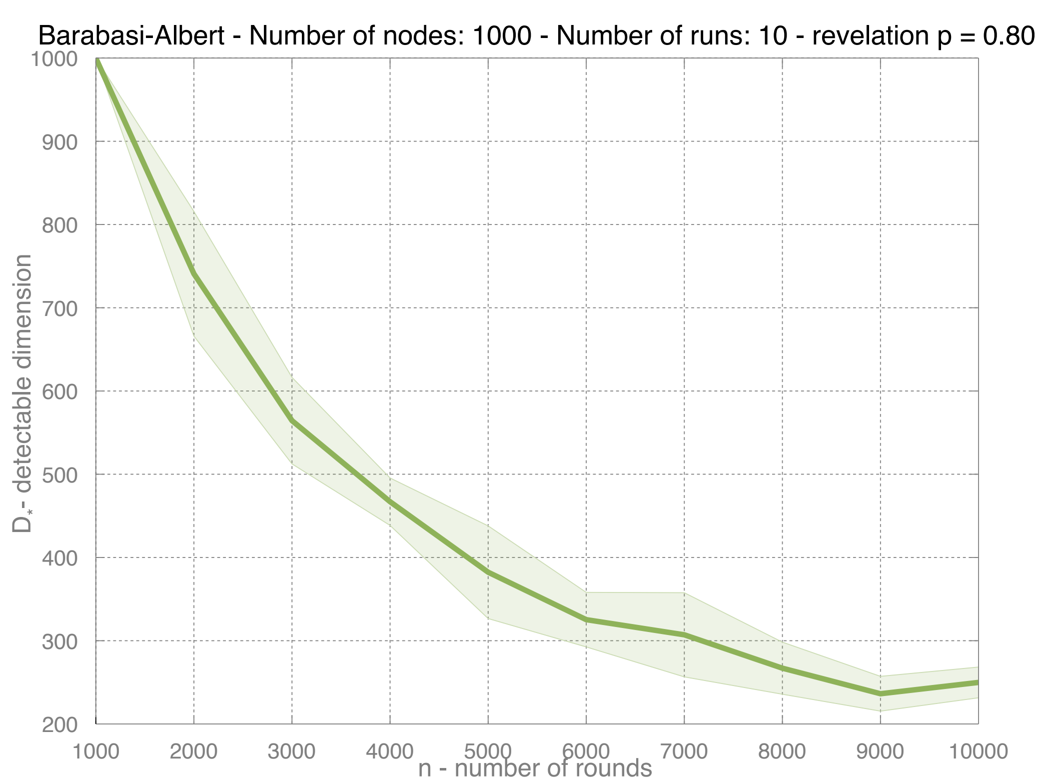
$$\Delta_\star = 16\sqrt{\frac{r_\star^\circ N \log(TN)}{T_\star}} + \frac{144 N \log(TN)}{T_\star}$$

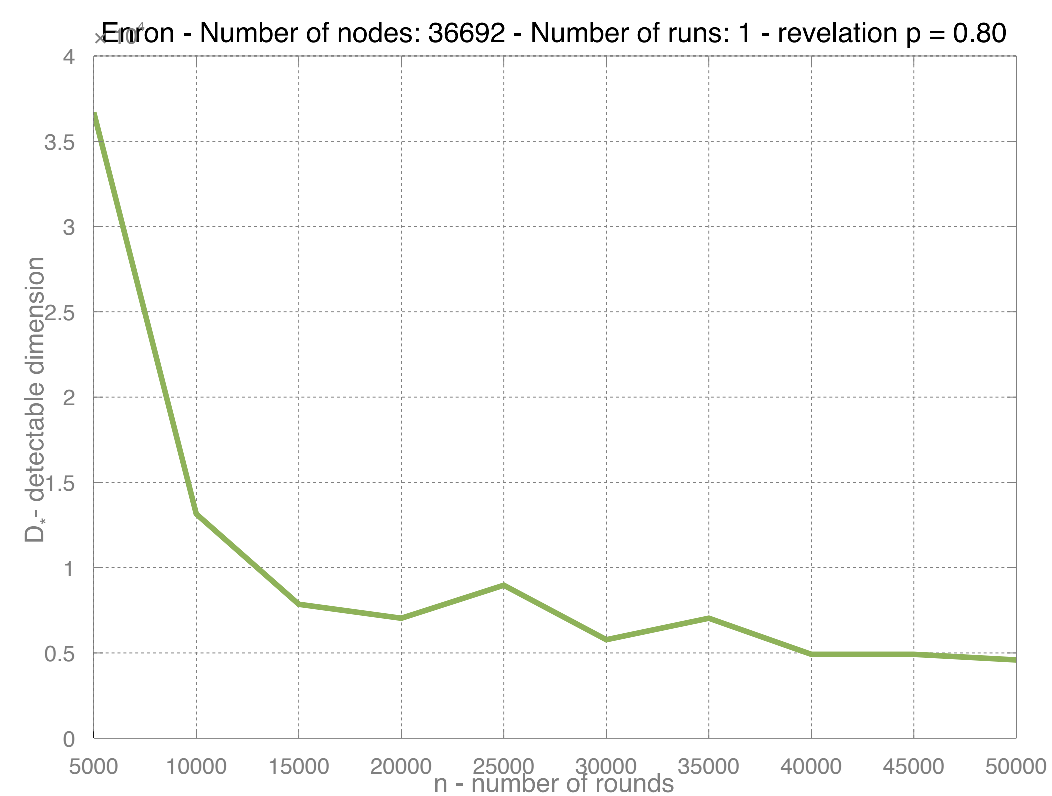
- Detectable horizon T*, smallest integer s.t. $T_\star r_\star^\circ \ge \sqrt{D_\star T r_\star^\circ},$
- Equivalently: D* corresponding to smallest T* such that

$$T_\star r_\star^\circ \ge \sqrt{D\left(16\sqrt{\frac{r_\star^\circ N \log(TN)}{T_\star}} + \frac{144 N \log(TN)}{T_\star}\right) T r_\star^\circ}$$

HOW DOES D* BEHAVE?

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





BARE SOLUTION

BAndit REvelator: 2-phase algorithm

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

Algorithm

BARE - BAndit REvelator

Input

d: the number of nodes

n: time horizon

Initialization

$T_{k,t} \leftarrow 0$, for $\forall k \leq d$

$\widehat{r^\circ_{k,t}} \leftarrow 0$, for $\forall k \leq d$

$t \leftarrow 1$, $\widehat{T}_\star \leftarrow 0$, $\widehat{D}_{\star,t} \leftarrow d$, $\widehat{\sigma}_{\star,1} \leftarrow d$

Global exploration phase

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

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

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

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

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

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

$t \leftarrow t+1$

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)

EMPIRICAL RESULTS

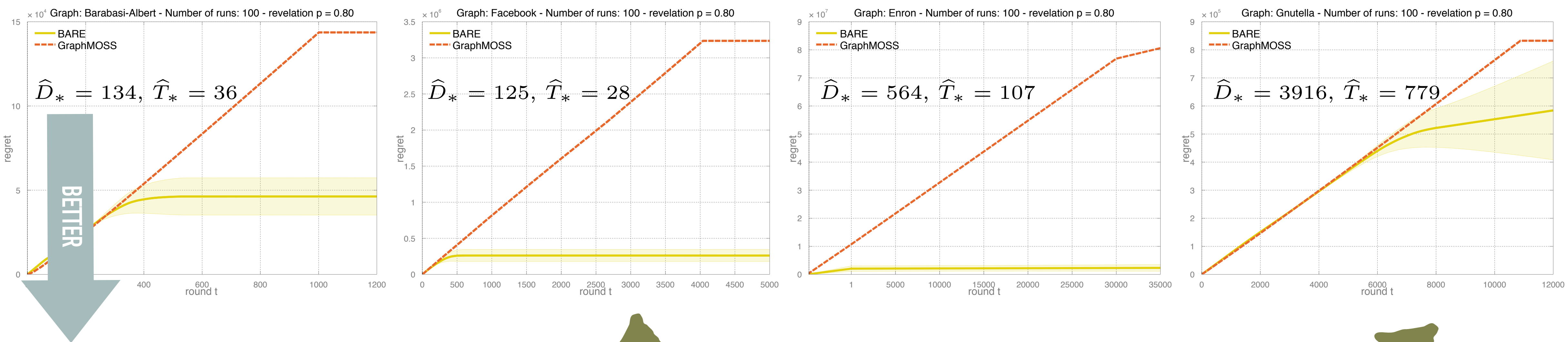
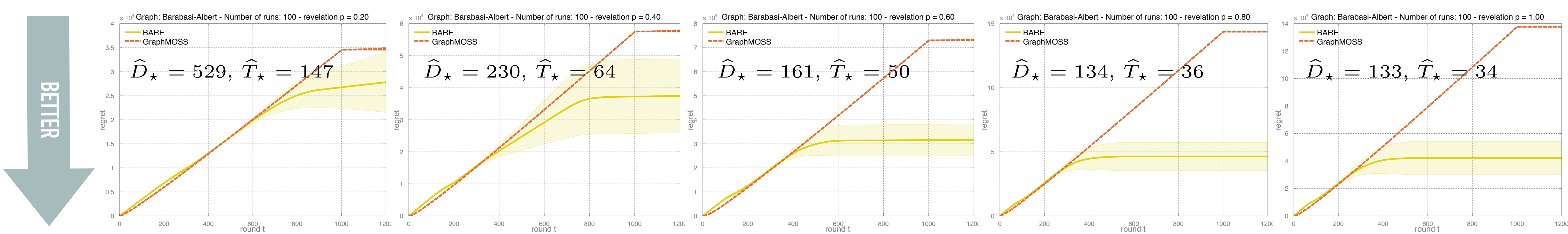


Figure 1: *Left*: Barabási-Albert. *Middle left*: Facebook. *Middle right*: Enron. *Right*: Gnutella.

Enron and Facebook **vs.** Gnutella (decentralised)



Varying a (constant) probability of influence

REVEALING BANDITS: WHAT DO YOU MEAN?

- Ignoring the structure again? $\widetilde{\mathcal{O}}\left(\sqrt{r_* T N}\right)$ — reward of the best node
- **BAndit REvelator:** 2-phase algorithm
- **global** exploration phase
  - super-efficient exploration
  - linear regret — needs to be short!
  - extracts D_* nodes
- **bandit** phase
  - uses a minimax-optimal bandit algorithm (GraphMOSS)
  - has a "square root" regret on D_* nodes
- D_* realizes the optimal trade-off !
  - different from exploration/exploitation tradeoff

Regret of BARE

$$\mathcal{O}\left(\sqrt{r_\star T D_\star}\right)$$

- **D*** - detectable dimension (depends on T and the structure)
  - **good case**: star-shaped graph can have D* = 1
  - **bad case:** a graph with many small cliques.
  - **the worst case:** all nodes are disconnected except 2

NEXT: GLOBAL INFLUENCE MODELS

- Kempe, Kleinberg, Tárdos, 2003, 2015: **Independence Cascades**, Linear Threshold models
  - **global** and **multiple-source** models
- Different feed-back models
  - **Full bandit** (only the number of influenced nodes)
  - **Node-level semi-bandit** (identities of influenced nodes)
  - **Edge-level semi-bandit** (identities of influenced edges)
    - Wen, Kveton, Valko, Vaswani, **NIPS 2017**
    - IMLinUCB with linear parametrization of edge weights
    - Regret analysis for **general graphs, cascading model, and multiple-sources**