

Les Journées Scientifiques, Inria, Sophia-Antipolis
le 14 juin 2017

WHERE IS **JUSTIN BIEBER** ? OR ONLINE INFLUENCE MAXIMIZATION

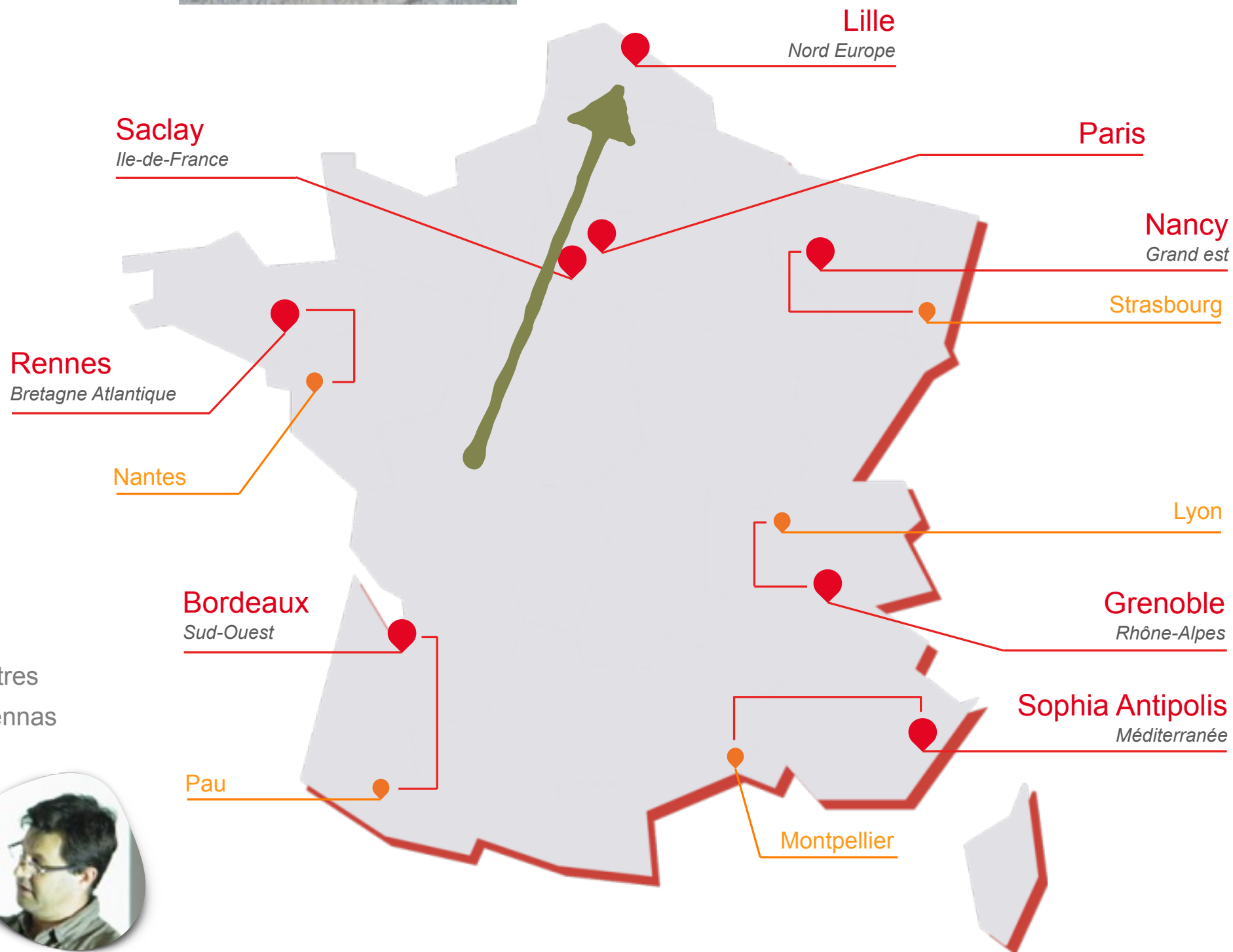
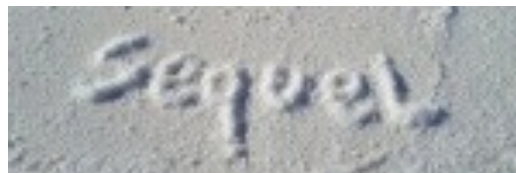
Michal Valko, SequeL, Inria Lille - Nord Europe



2015-2016
AISTAST 2016



2016-2017, under review
<https://arxiv.org/abs/1605.06593>

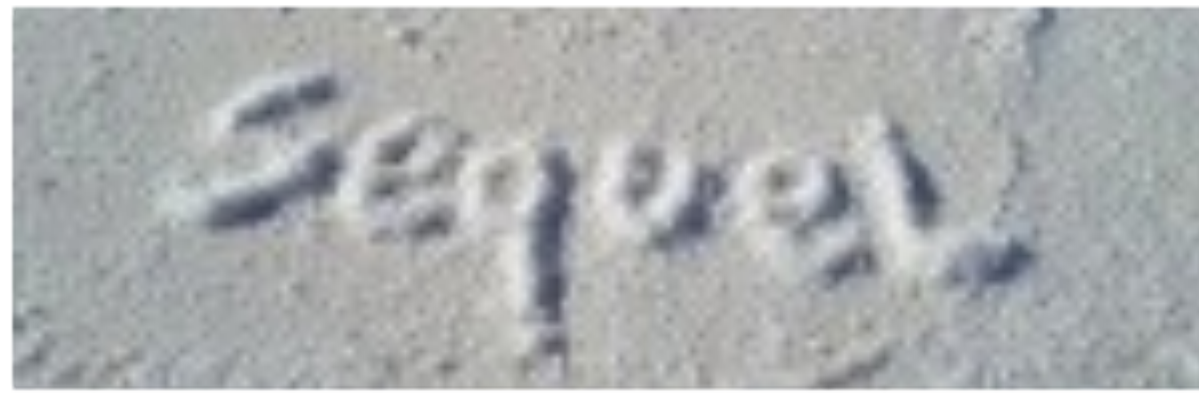


Philippe Preux
Sequel, Inria



Rémi Munos
Google DeepMind

10 YEARS



... LAST 10 YEARS AND INDUSTRY

criteo.

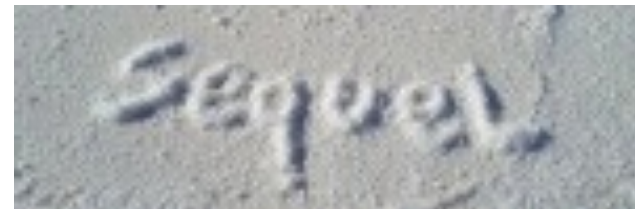
J. Mary

M. Davy

veikia
THE HEART OF RETAIL



M. Ghavamzadeh



A. Lazaric

R. Coulom
CrazyStone

R. Munos
O. Pietquin
B. Piot
G. Dulac-Arnold
A. Huang
M. Azar
JB. Grill

Facebook AI Research
PARIS



DeepMind

FOCUS ON **ONLINE** RECOMMENDER SYSTEMS

Deep Learning for Recommender systems (Strub, Mary, Gaudel, Preux)
State-of-art result on Netflix challenge + contracts with companies



sequential decision making way of thinking solutions for cold-star problem



Predictive Merchandising Technology

deep or not: recommender systems are major field of research and applications of Sequel

3 ACTIVE HIGHLIGHTS... LAST 6 MONTHS

TrailBlazer - plenary at NIPS 2016
general sample-efficient planner
Grill, Munos, Valko

Blazing the trails before beating the path: Sample-efficient Monte-Carlo planning

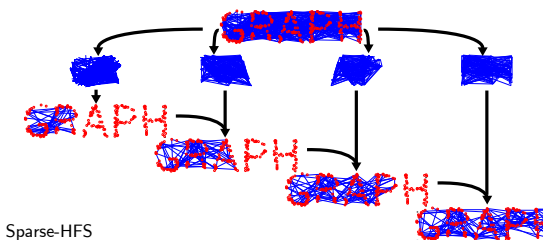
Jean-Bastien Grill *Sequel-team, INRIA Lille - Nord Europe, France*
jean-bastien.grill@inria.fr Michal Valko *Google DeepMind, UK*
michal.valko@inria.fr Rémi Munos *Google DeepMind, UK*
munos@google.com

Abstract

You are a robot and you live in a Markov decision process (MDP) with a finite or an infinite number of transitions from state-action to next states. You get beliefs and so you plan before you act. Luckily, your robotparents equipped you with a generative model to do some Monte-Carlo planning. The world is waiting for you and you have no time to waste. You want your planning to be efficient. *Sample-efficient*. Indeed, you want to exploit the possible structure of the MDP by exploring only a subset of states reachable by following near-optimal policies. You want guarantees on sample complexity that depend on a measure of the quantity of near-optimal states. You want something, that is an extension of Monte-Carlo sampling (for estimating an expectation) to problems that alternate maximization (over actions) and expectation (over next states). But you do not want to stop with exponential running time, you want something simple to implement and computationally efficient. You want it all and you want it now. You want **TrailBlazer**.



Squeak: Online Graph and Kernel Sparsifiers
UAI 2016, AISTATS 2017, ICML 2017, ...
Calandriello, Lazaric, Valko



Sparse-HFS

Algorithm	parkinson $n = 5,875, d = 20$			cpusmall $n = 8,192, d = 12$		
	Avg. Squared Loss	#SV	Time	Avg. Squared Loss	#SV	Time
FOGD	0.04909 ± 0.00020	30	—	0.02577 ± 0.00050	30	—
NOGD	0.04896 ± 0.00068	30	—	0.02559 ± 0.00024	30	—
PROS-N-KONS	0.05798 ± 0.00136	18	5.16	0.02494 ± 0.00141	20	7.28
CON-KONS	0.05696 ± 0.00129	18	5.21	0.02269 ± 0.00164	20	7.40
B-KONS	0.05795 ± 0.00172	18	5.35	0.02496 ± 0.00177	20	7.37
BATCH	0.04535 ± 0.00002	—	—	0.01090 ± 0.00082	—	—

Algorithm	cadata $n = 20,640, d = 8$			casp $n = 45,730, d = 9$		
	Avg. Squared Loss	#SV	Time	Avg. Squared Loss	#SV	Time
FOGD	0.04097 ± 0.00015	30	—	0.08021 ± 0.00031	30	—
NOGD	0.03983 ± 0.00018	30	—	0.07844 ± 0.00008	30	—
PROS-N-KONS	0.03095 ± 0.00110	20	18.59	0.06773 ± 0.00105	21	40.73
CON-KONS	0.02850 ± 0.00174	19	18.45	0.06832 ± 0.00315	20	40.91
B-KONS	0.03095 ± 0.00118	19	18.65	0.06775 ± 0.00067	21	41.13
BATCH	0.02202 ± 0.00002	—	—	0.06100 ± 0.00003	—	—

Algorithm	slice $n = 53,500, d = 385$			year $n = 463,715, d = 90$		
	Avg. Squared Loss	#SV	Time	Avg. Squared Loss	#SV	Time
FOGD	0.00726 ± 0.00019	30	—	0.01427 ± 0.00004	30	—
NOGD	0.02636 ± 0.00460	30	—	0.01427 ± 0.00004	30	—
DUAL-SGD	—	—	—	0.01440 ± 0.00000	100	—
PROS-N-KONS	did not complete	—	—	0.01450 ± 0.00014	149	884.82
CON-KONS	did not complete	—	—	0.01444 ± 0.00017	147	889.42
B-KONS	0.00913 ± 0.00045	100	60	0.01302 ± 0.00006	100	505.36
BATCH	0.00212 ± 0.00001	—	—	0.01147 ± 0.00001	—	—

<http://GuessWhat.AI>

deep RL for dialogue in natural language

Strub, de Vries, Mary, Pietquin, Courville, Larochelle

Dataset
It's rich

- 155,280 played games
- 821,889 questions+answers
- 66,537 images
- 134,073 objects

Download the dataset.



GUESSWHAT.AI

 **GUESSWHAT?!**

Home People Download Explore **Play the game** 

29 ☐ Waiting for an answer.. [GuessWhat!](#) [? Questioner](#)

Does he have his mouse open?

is it the yougest one?

Yes

Is it on the right?

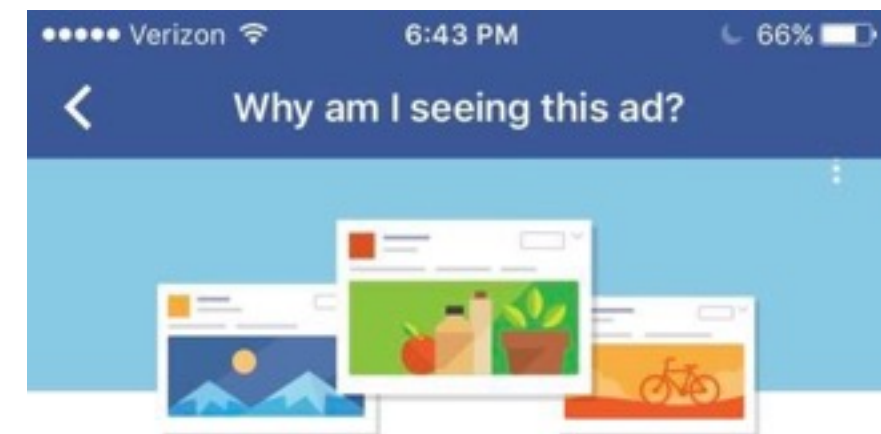
No

Is it a boy?

Yes

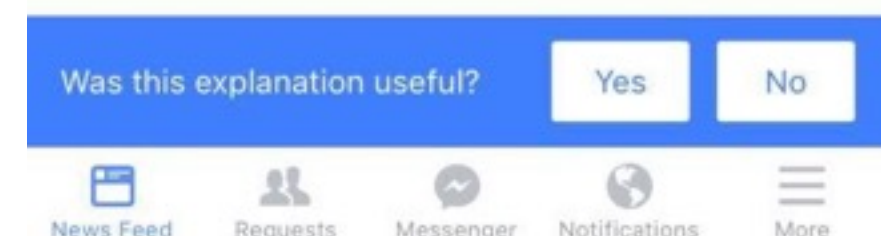


HOW TO RULE THE WORLD?

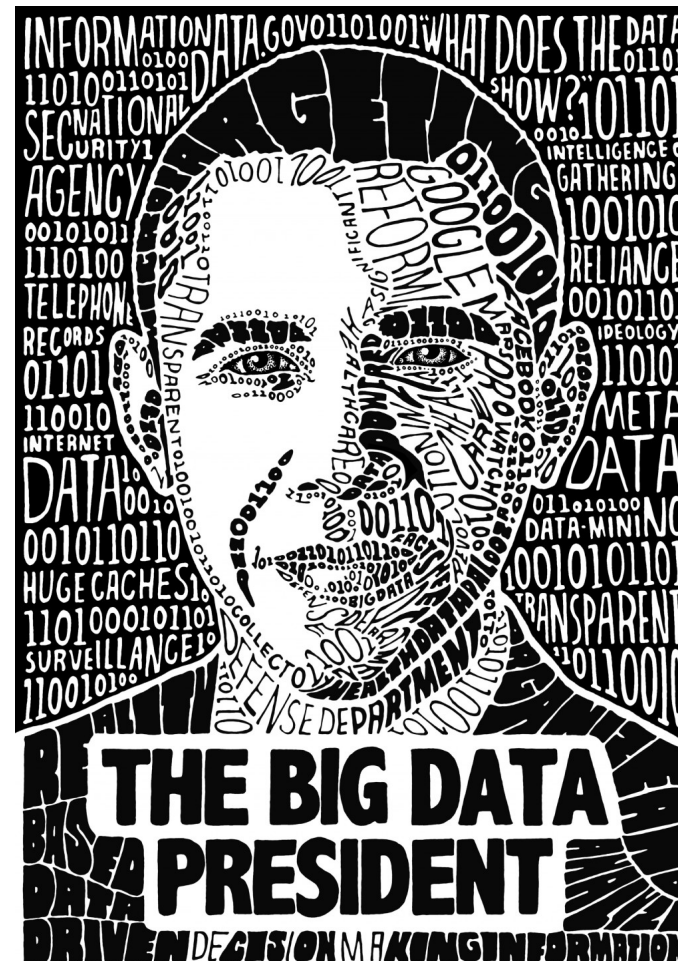


One reason you're seeing this ad is that [Donald J. Trump](#) wants to reach people who are part of an audience called **"Likely To Engage in Politics (Liberal)"**. This is based on your activity on Facebook and other apps and websites, as well as where you connect to the internet.

There may be other reasons you're seeing this ad, including that Donald J. Trump wants to reach **people ages 25 and older who live near Boston, Massachusetts**. This is information based on your Facebook profile and where you've connected to the internet.



“IA” EST DÉJÀ LÀ



https://www.washingtonpost.com/opinions/obama-the-big-data-president/2013/06/14/1d71fe2e-d391-11e2-b05f-3ea3f0e7bb5a_story.html

<https://www.technologyreview.com/s/509026/how-obamas-team-used-big-data-to-rally-voters/>

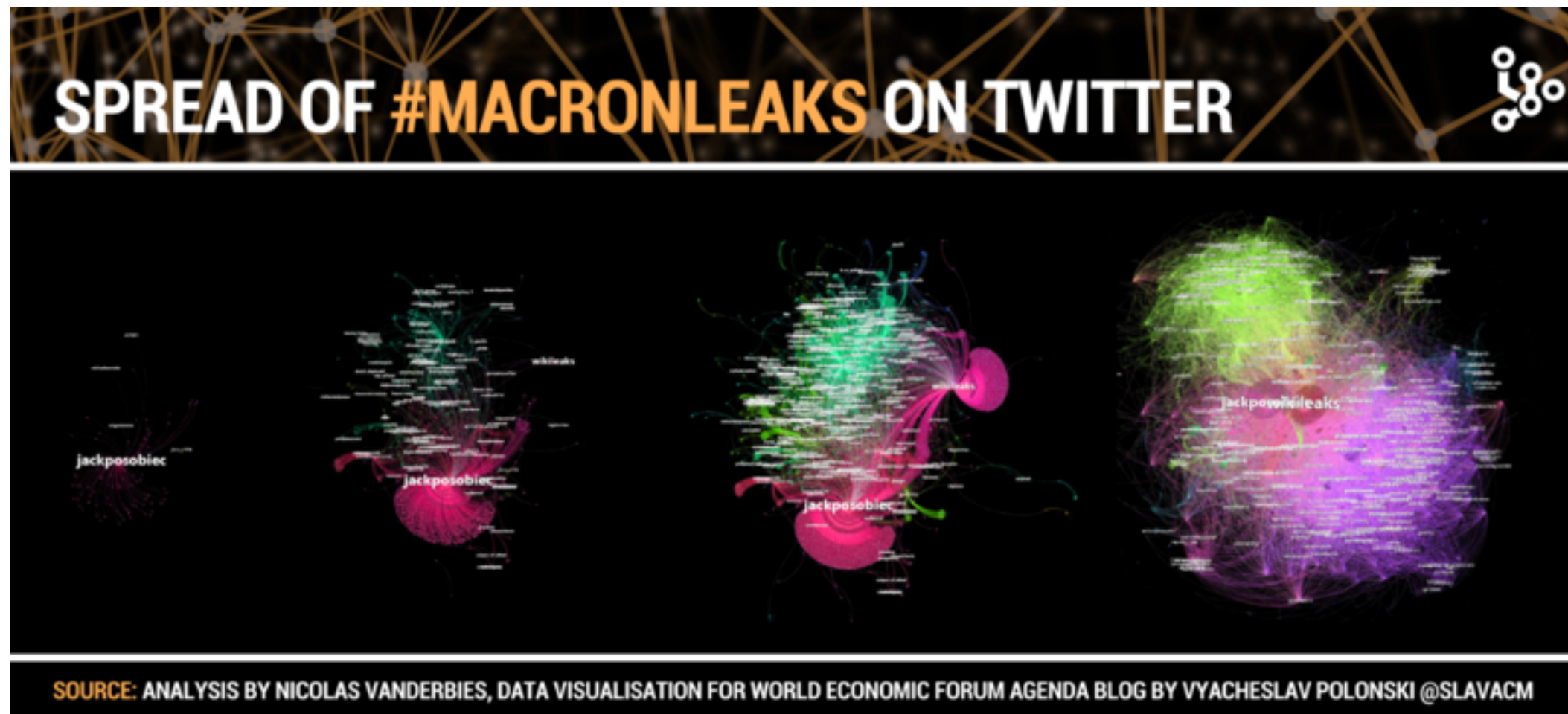
Talk of Rayid Ghaniy: https://www.youtube.com/watch?v=gDM1GuszM_U

INSOUMISE OU ENRACINÉE ?

.....

Le "big data" ou la recette secrète du succès d'Emmanuel Macron?

<https://www.rts.ch/info/sciences-tech/8580821-le-big-data-ou-la-recette-secrete-du-succes-d-emmanuel-macron-.html>



HOW TO RULE THE WORLD?

Influence the influential!



JULY 18, 2016

Religion



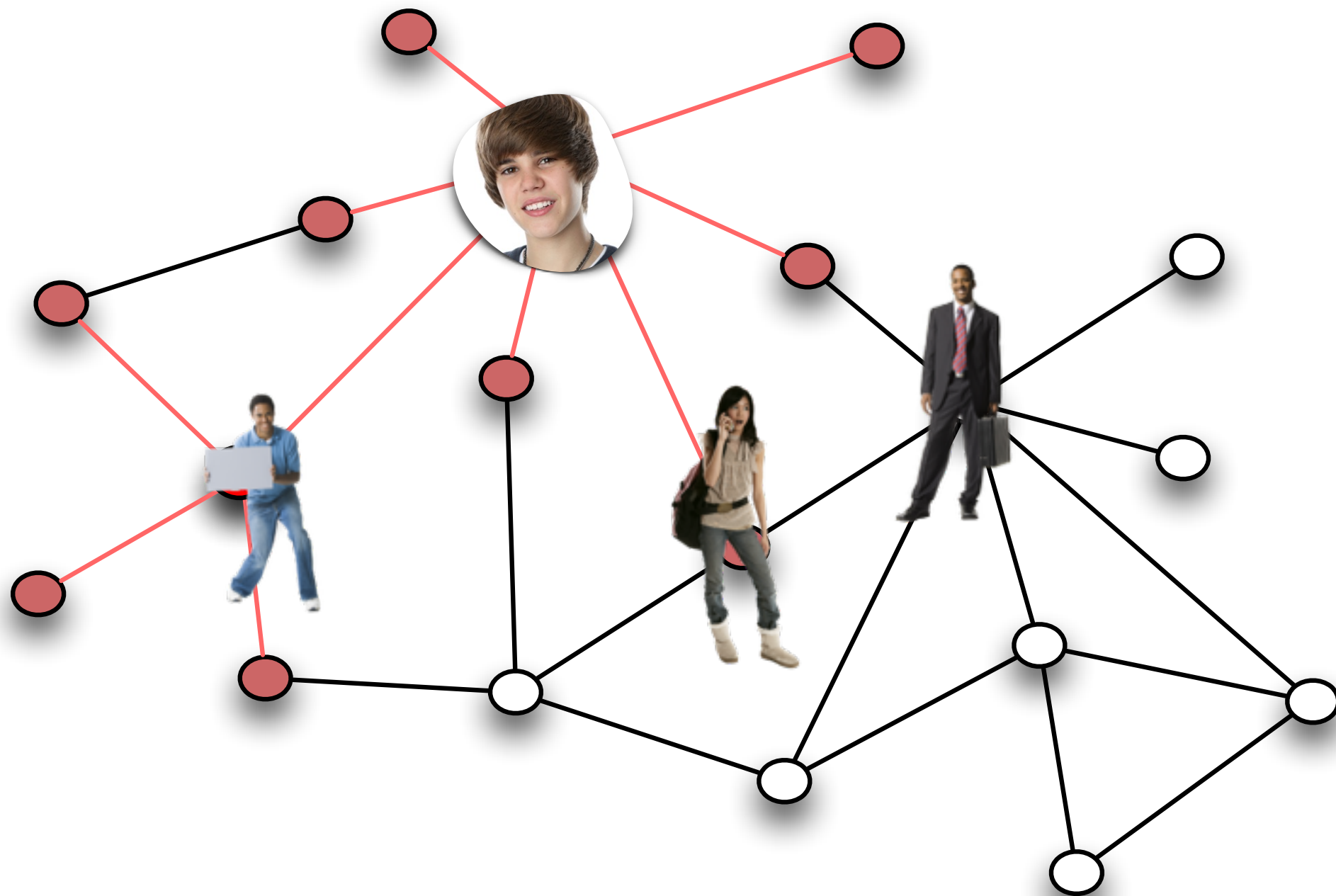
March 26, 2017

Politics



September 1, 2009

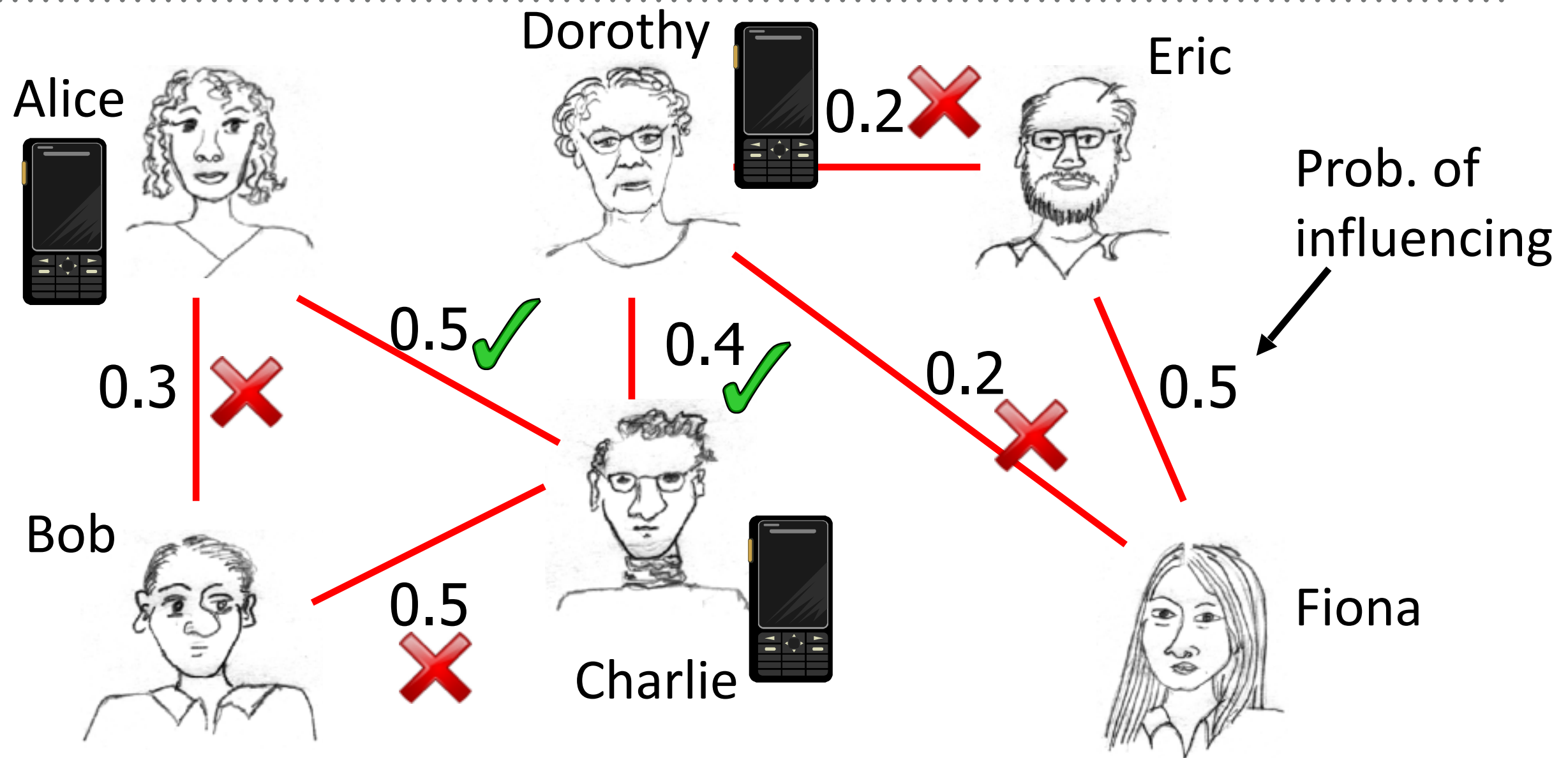
Culture



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

EXAMPLE: INFLUENCE IN SOCIAL NETWORKS

[KEMPE, KLEINBERG, TARDOS KDD '03]

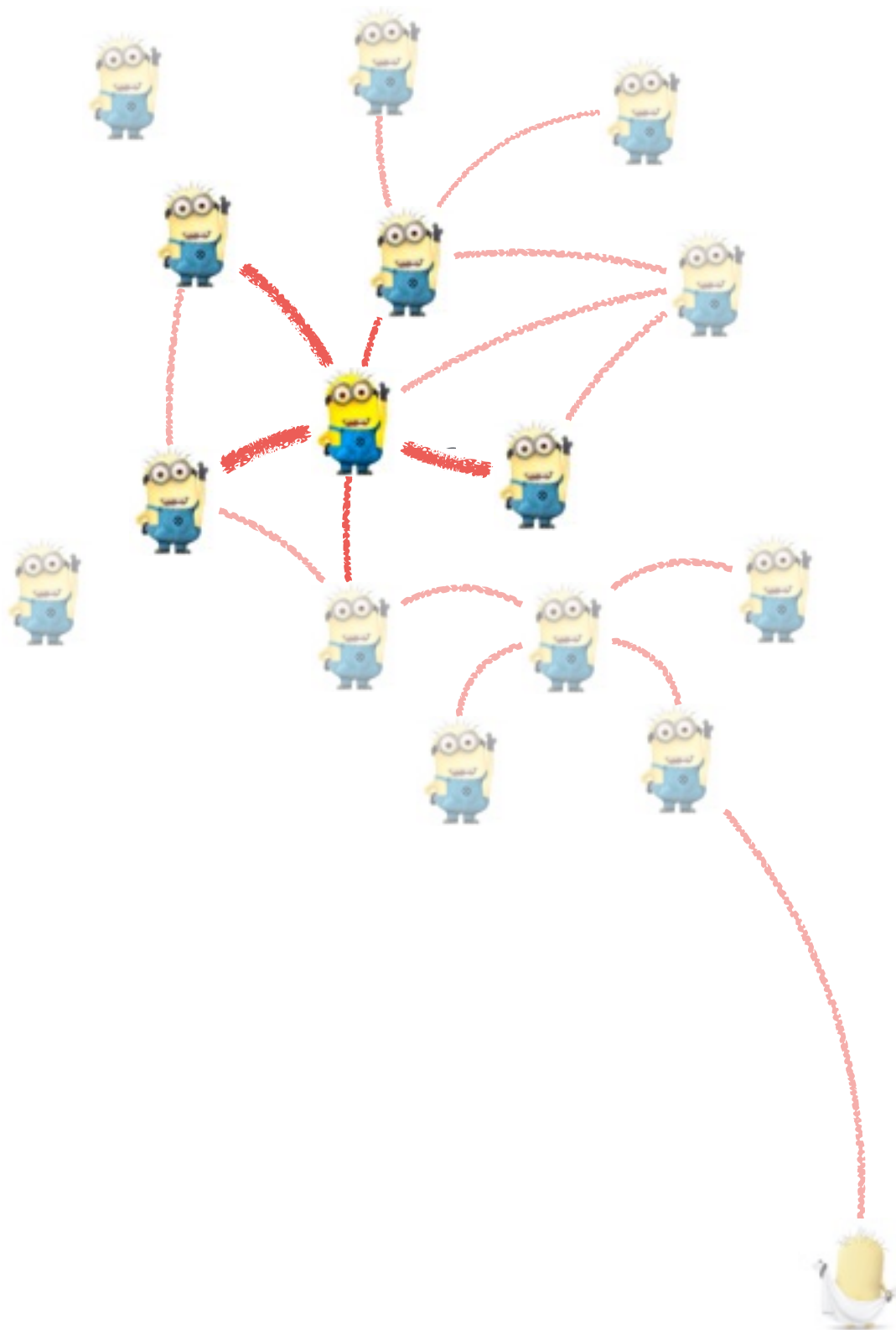


Who should get free cell phones?

$V = \{\text{Alice}, \text{Bob}, \text{Charlie}, \text{Dorothy}, \text{Eric}, \text{Fiona}\}$

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

MAXIMIZING INFLUENCE



Product placement

- ▶ dispatch few to sell more
- ▶ target influential people

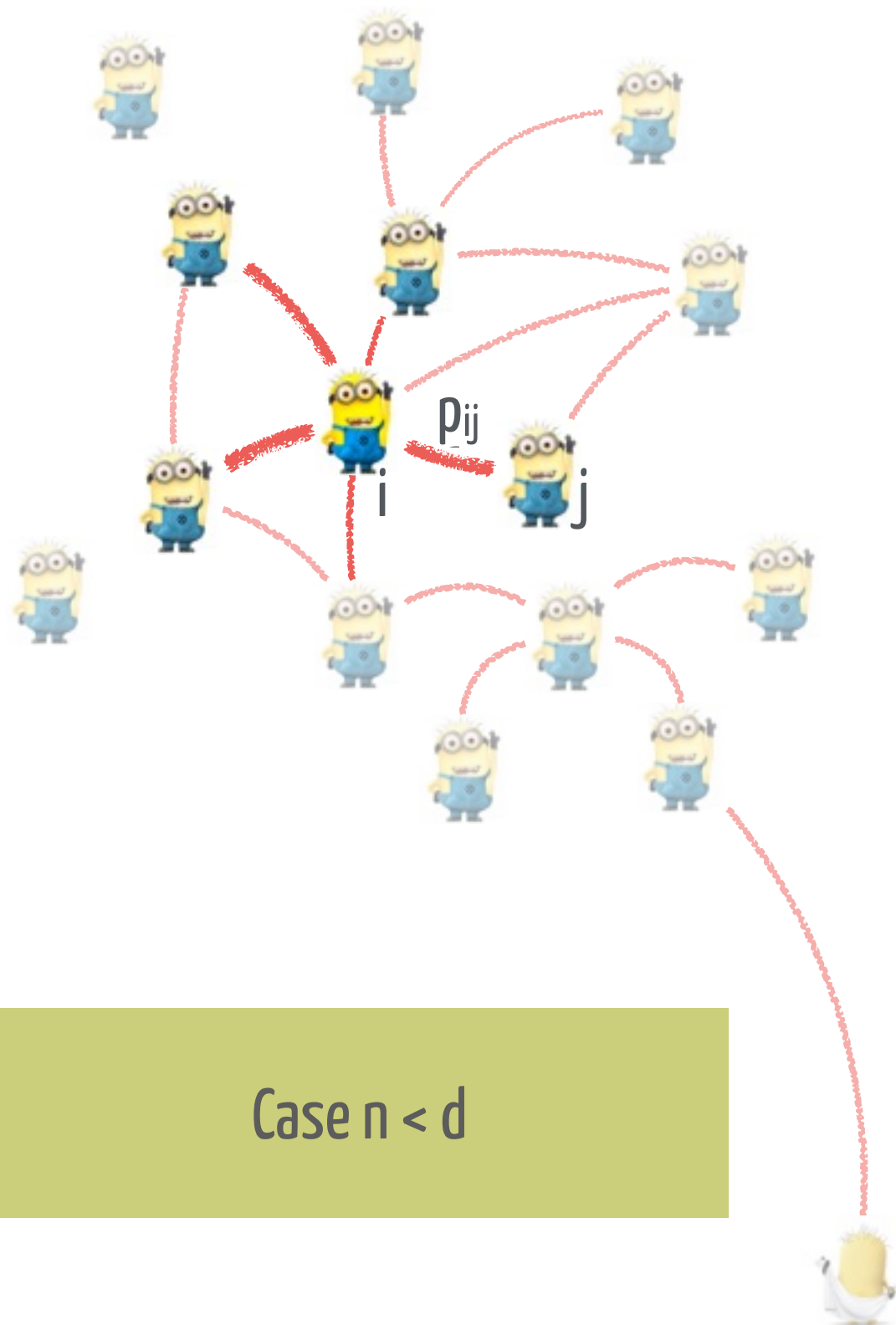
Gathering the information?

- ▶ likes on FB
- ▶ promotional codes

Unknown graphs

- ▶ all prior work needed to know the **graph**
- ▶ here: **provably learning faster without it**

REVEALING BANDITS FOR **LOCAL** INFLUENCE



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

In each time step $t = 1, \dots, n$

learner picks a node k_t

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

Select influential people = Find the strategy maximising

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

Why this is a **bandit problem**?

Case $n < d$

What are **bandits** anyway?

The number of expected influences of node k is by definition

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

Oracle strategy always selects the best

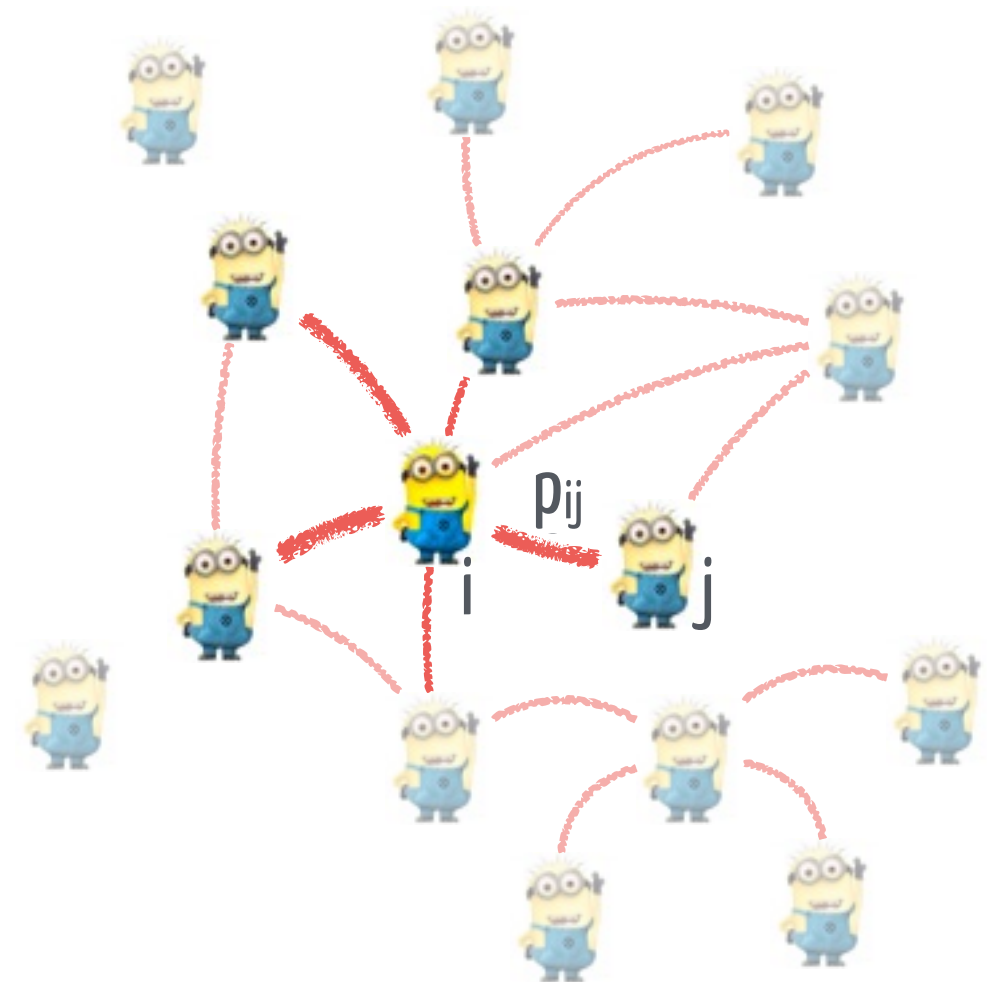
$$k^* = \arg \max_k \mathbb{E} \left[\sum_{t=1}^n |S_{k,t}| \right] = \arg \max_k n r_k$$

Expected **reward** of the oracle strategy

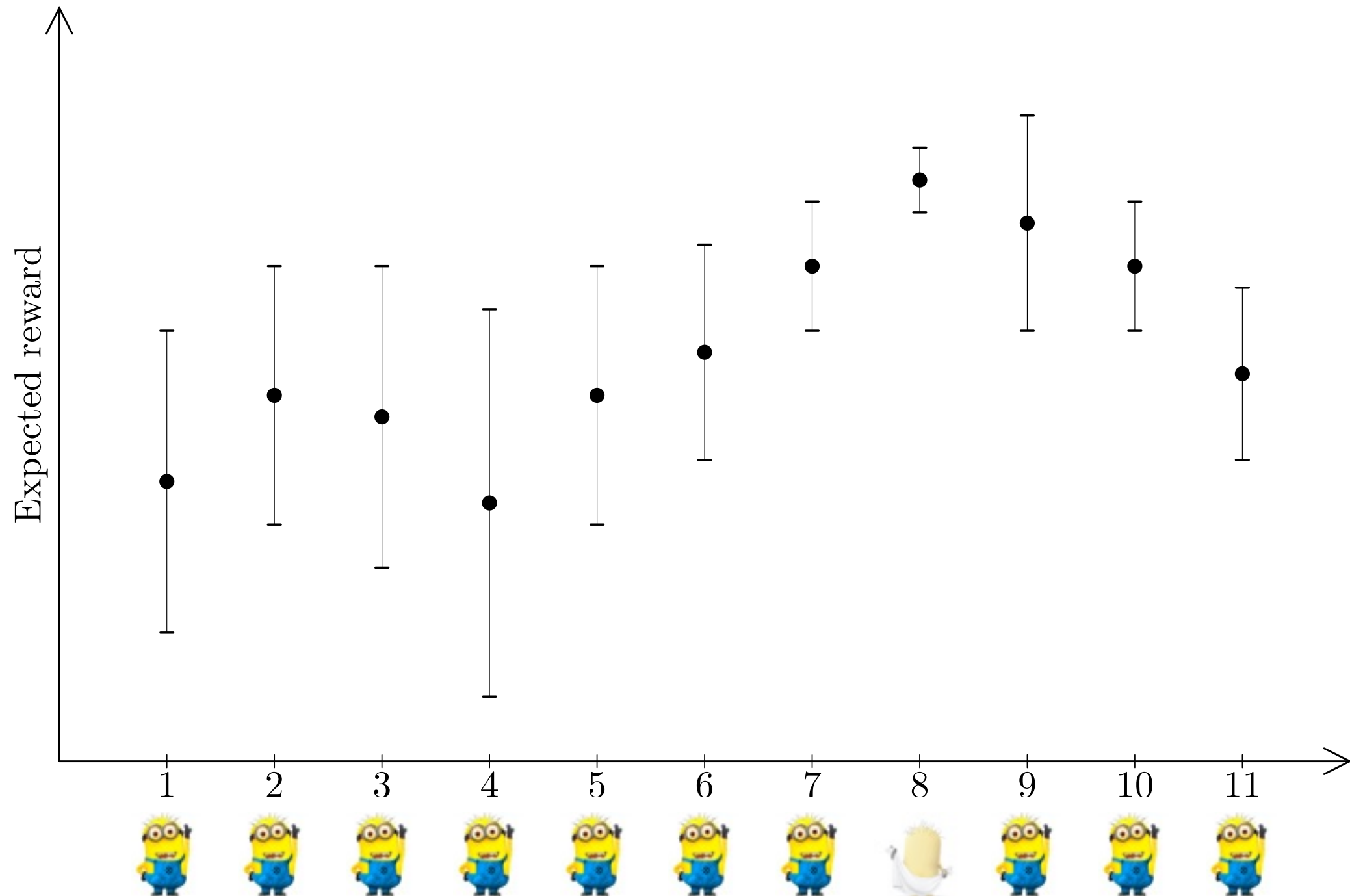
$$\mathbb{E} [L_n^*] = n r_*$$

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

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



UPPER CONFIDENCE BOUND BASED ALGOS



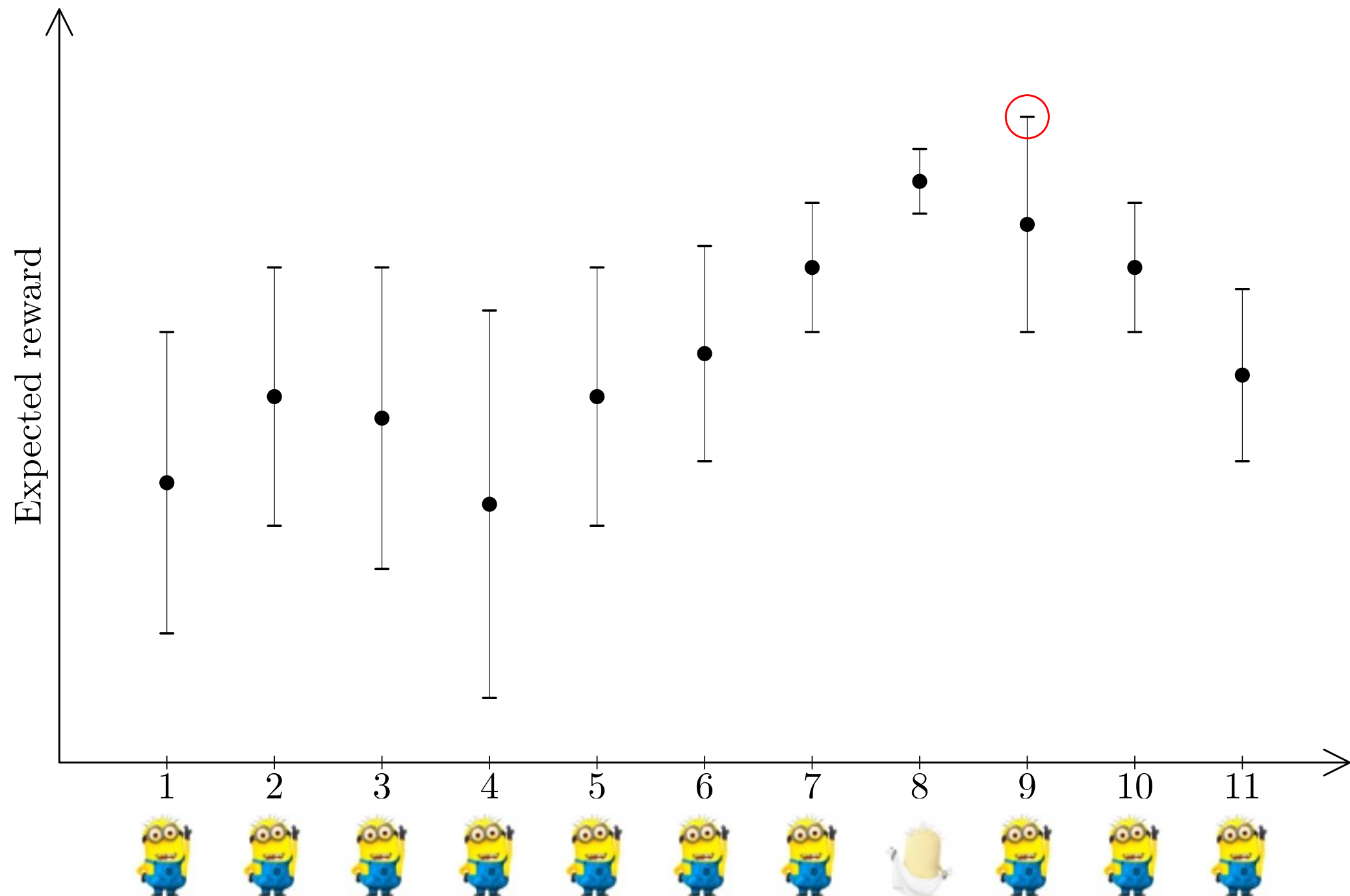
MULTI-ARM BANDITS IN CAFÉ CULTURE



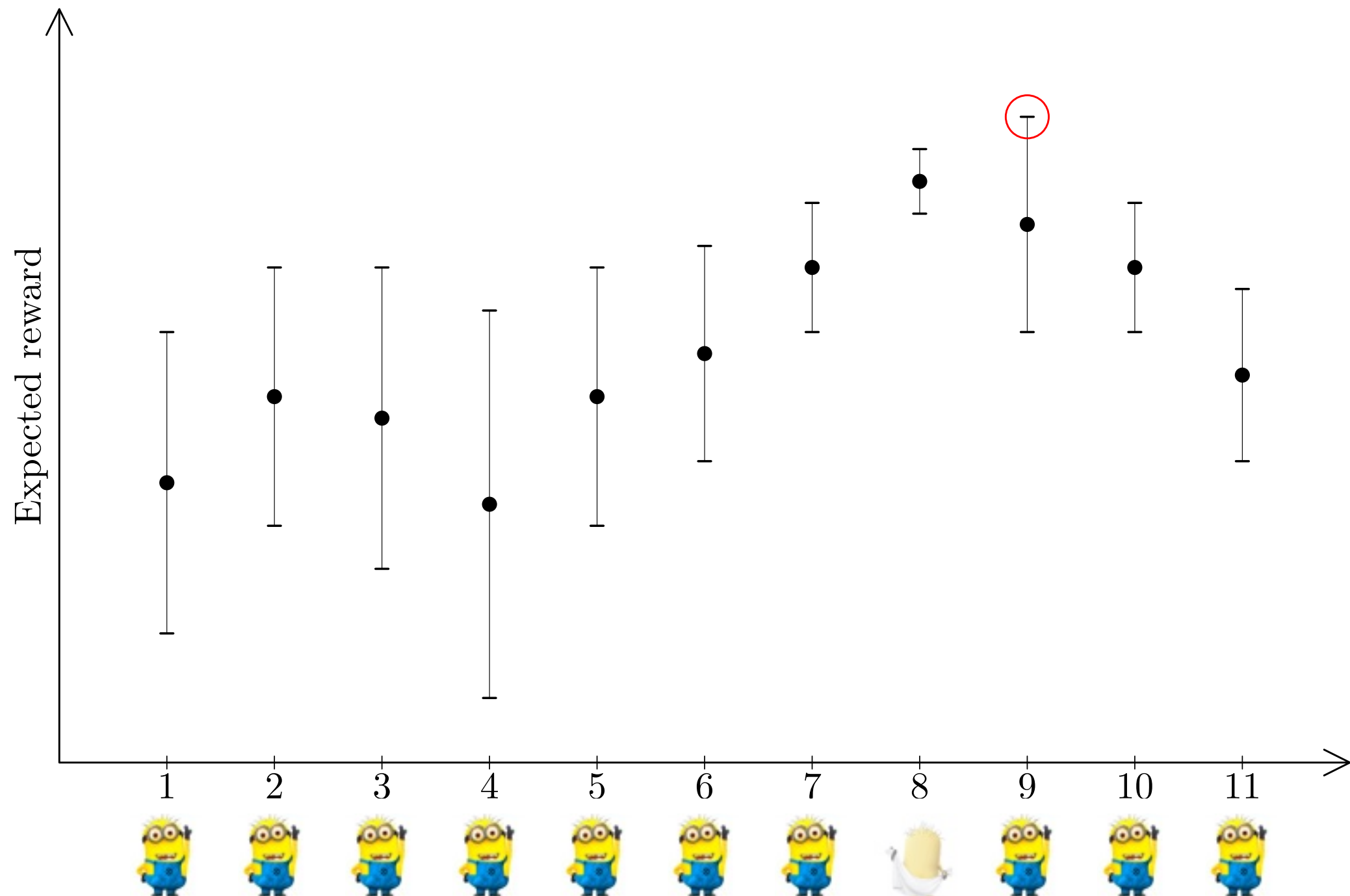
Video recorded **March 30th, 2015, 13h50**,
Université de Lille, Susie & the Piggy Bones Band



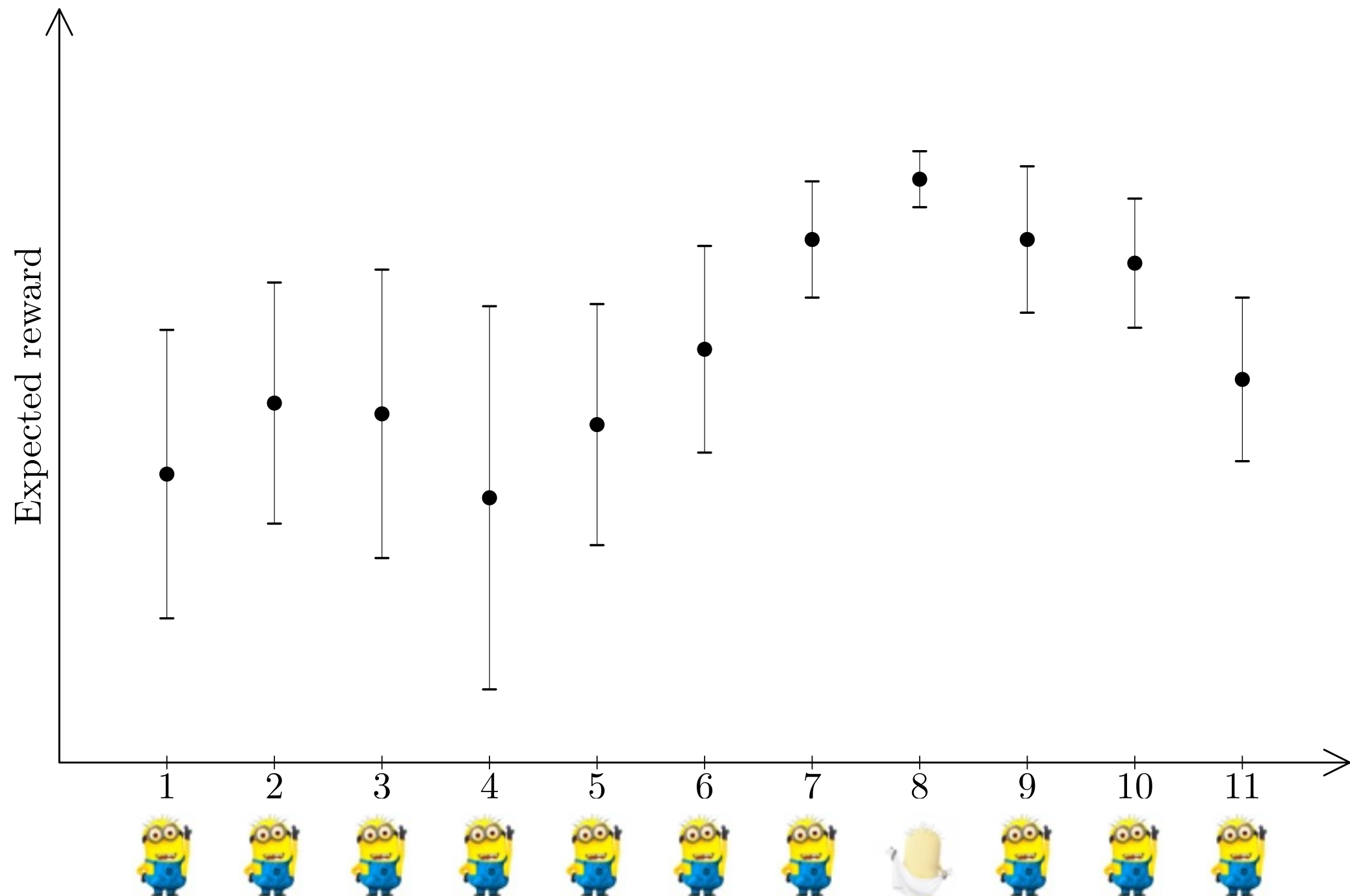
UPPER CONFIDENCE BOUND BASED ALGOS



UPPER CONFIDENCE BOUND BASED ALGOS



UPPER CONFIDENCE BOUND BASED ALGOS



- ▶ number of nodes we can efficiently extract in less than n rounds

- ▶ function D controls number of nodes given a gap

$$D(\Delta) \stackrel{\text{def}}{=} |\{i \leq d : r_{\star}^{\circ} - r_i^{\circ} \leq \Delta\}|$$

- ▶ $D(r) = d$ for $r \geq r_{\star}$ 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

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

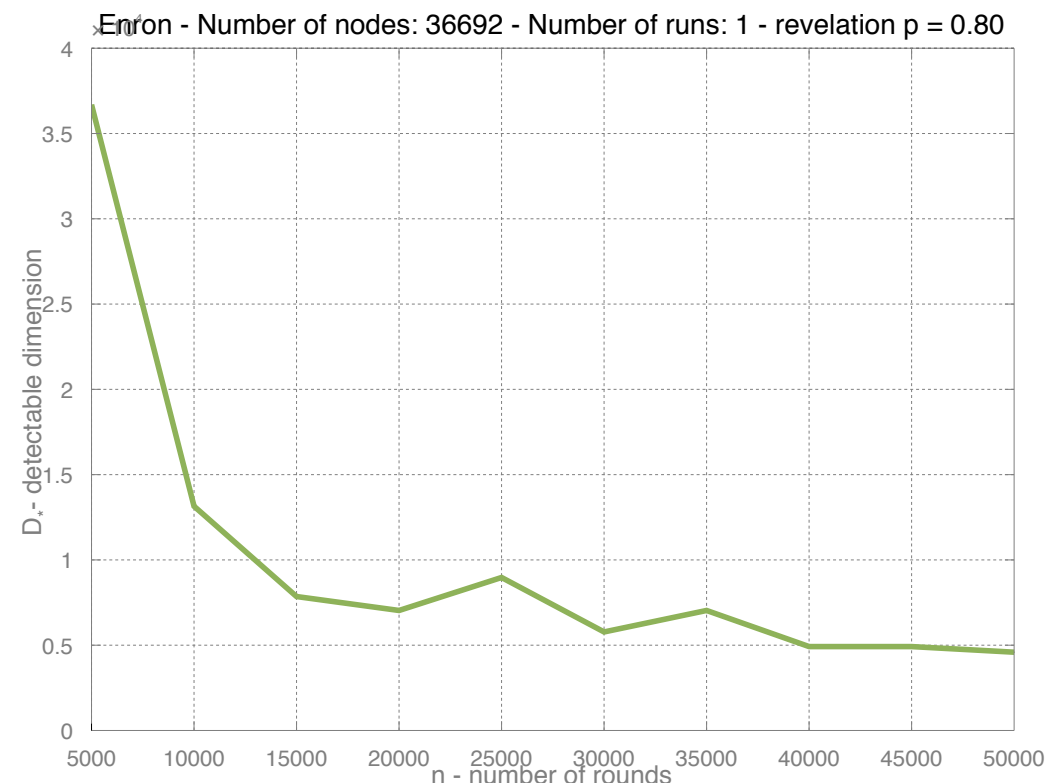
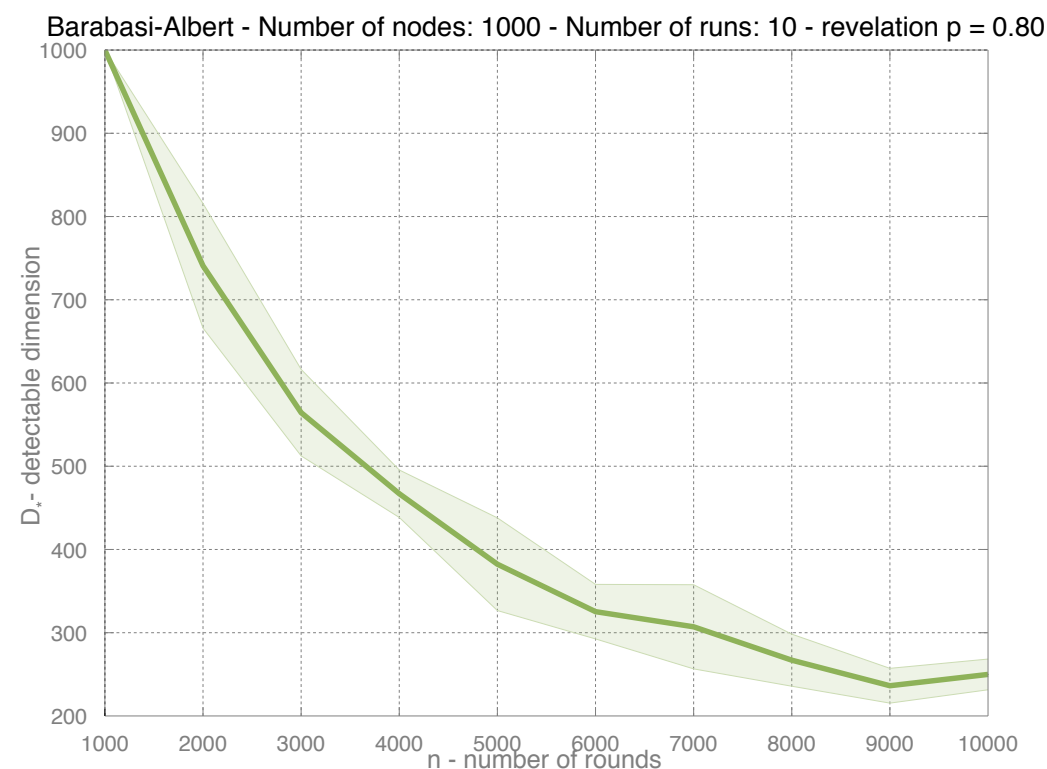
- ▶ Detectable horizon $T_{\star}$, smallest integer s.t. $T_{\star} r_{\star}^{\circ} \geq \sqrt{D_{\star} n r_{\star}^{\circ}}$

- ▶ Equivalently: $D_{\star}$ corresponding to smallest $T_{\star}$ such that

$$T_{\star} r_{\star}^{\circ} \geq \sqrt{D \left(16 \sqrt{\frac{r_{\star}^{\circ} d \log(nd)}{T_{\star}}} + \frac{80 d \log(nd)}{T_{\star}} \right) n r_{\star}^{\circ}}$$

HOW DOES D^* BEHAVE?

- ▶ For (easy, structured) **star** graphs $D^* = 1$ even for small n (**big gain**)
- ▶ For (difficult) **empty** graphs $D^* = d$ even for large n (**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 n



**Input** d : the number of nodes n : time horizon**Initialization** $T_{k,t} \leftarrow 0$, for $\forall k \leq d$ $\widehat{r_{k,t}^o} \leftarrow 0$, 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

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

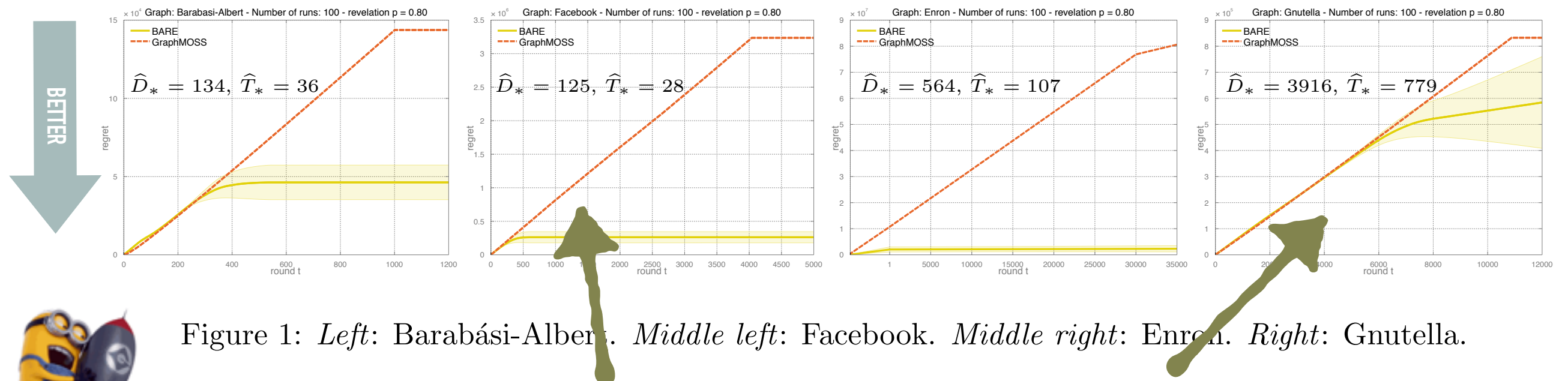
$$\widehat{\sigma}_{\star,t+1} \leftarrow \max_{k'} \sqrt{\widehat{r_{k',t+1}^o} + 8d \log(nd)/(t+1)}$$

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

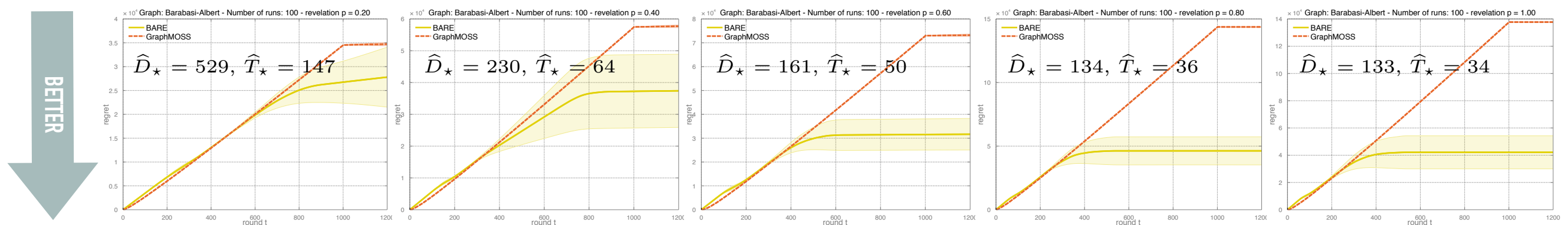
$$\widehat{D}_{\star,t+1} \leftarrow \left| \left\{ k : \max_{k'} \widehat{r_{k',t+1}^o} - \widehat{r_{k,t+1}^o} \leq w_{\star,t+1} \right\} \right|$$
 $t \leftarrow t + 1$ **end while** $\widehat{T}_\star \leftarrow t$.**Bandit phase**

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

EMPIRICAL RESULTS



Enron and Facebook vs. Gnutella (decentralised)



REVEALING BANDITS: WHAT DO YOU MEAN?

- ▶ Ignoring the structure?
- ▶ **BA**ndit **RE**velator: 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

$$\mathcal{O}(\sqrt{r_{\star} n d})$$

reward of the
best node

Regret of BARE

$$\mathcal{O}(\sqrt{r_{\star} n D_{\star}})$$

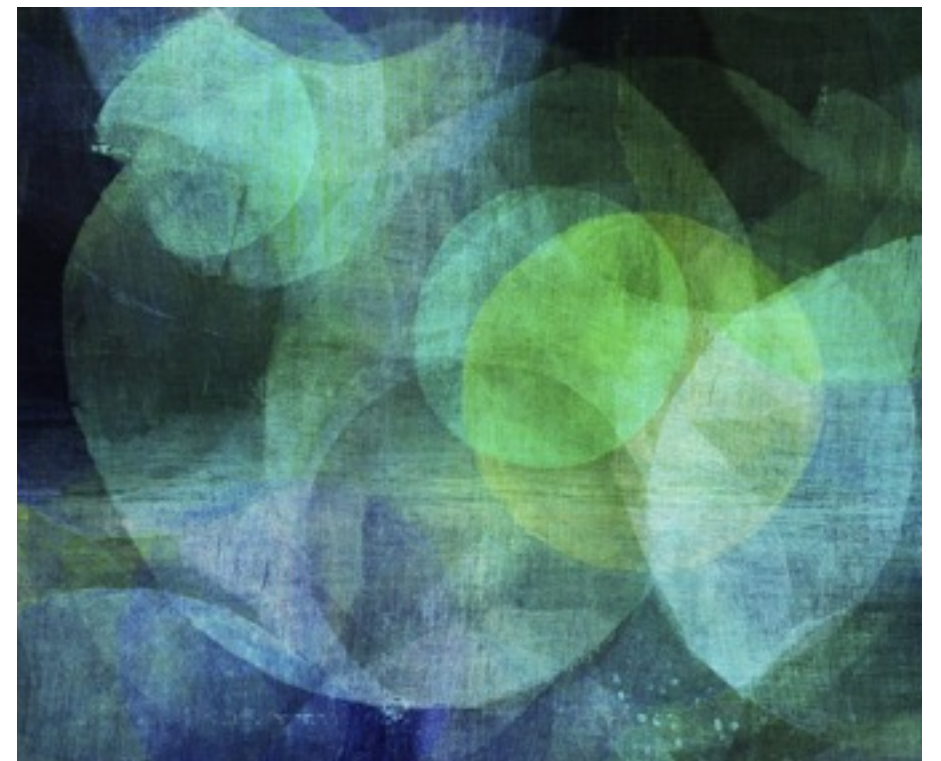
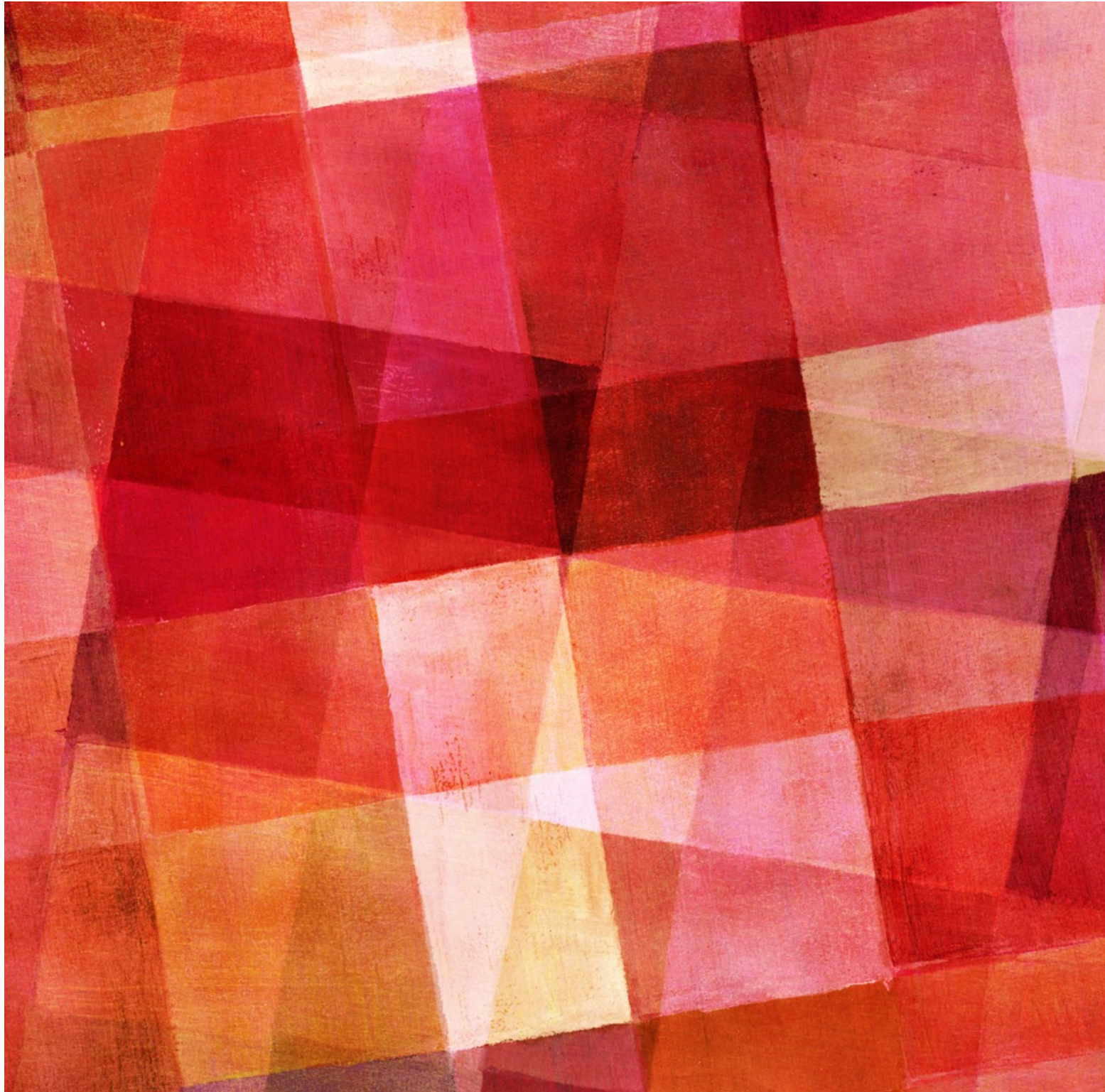
- ▶ **D*** - detectable dimension
(depends on n and the structure)
 - **good case**: star-shaped graph
can have $D^* = 1$
 - **bad case**: a graph with many
small cliques.
 - **the worst case**: all nodes are
disconnected except 2

NEXT: **GLOBAL** INFLUENCE MODELS

- ▶ Kempe, Kleinberg, Tárdoš, 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)
 - <http://arxiv.org/abs/1605.06593> (Wen, Kveton, Valko, Vaswani, under review)
 - IMLinUCB with linear parametrization of edge weights
 - Regret analysis for **general graphs**

MAIS OÙ SE CACHE-T-IL ?





ExtraLearn

Michal Valko, SequeL, Inria Lille - Nord Europe, michal.valko@inria.fr
<http://researchers.lille.inria.fr/~valko/hp/>