WHERE IS JUSTIN BIEBER ? OR ONLINE INFLUENCE MAXIMIZATION

Michal Valko, SequeL, Inria Lille - Nord Europe

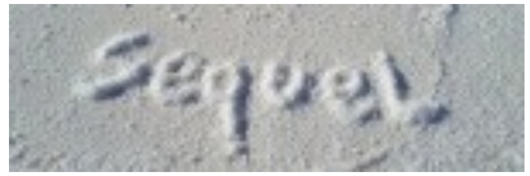

2015-2016

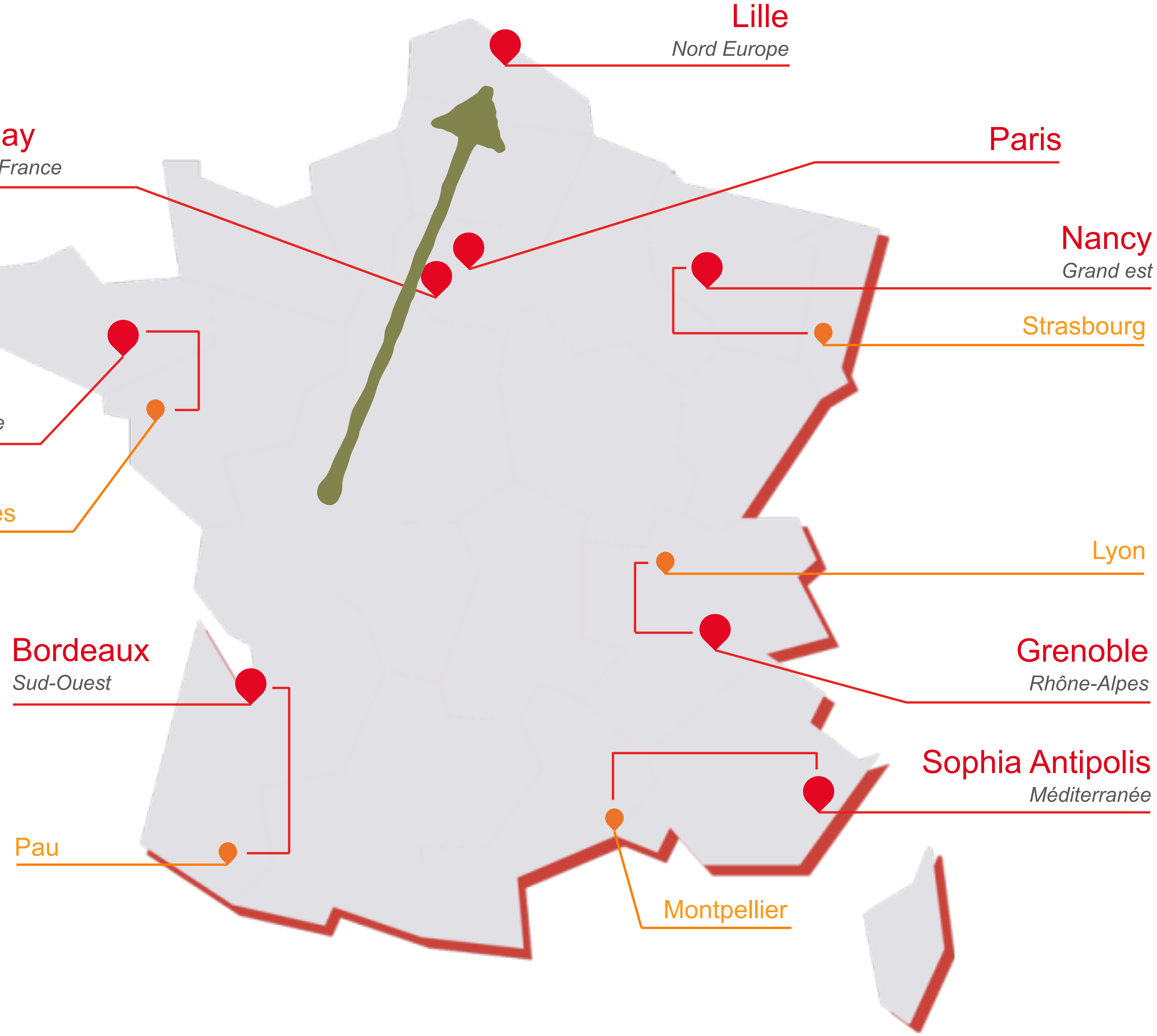
AISTAST 2016




2016-2017

NIPS 2017 accepted, to appear



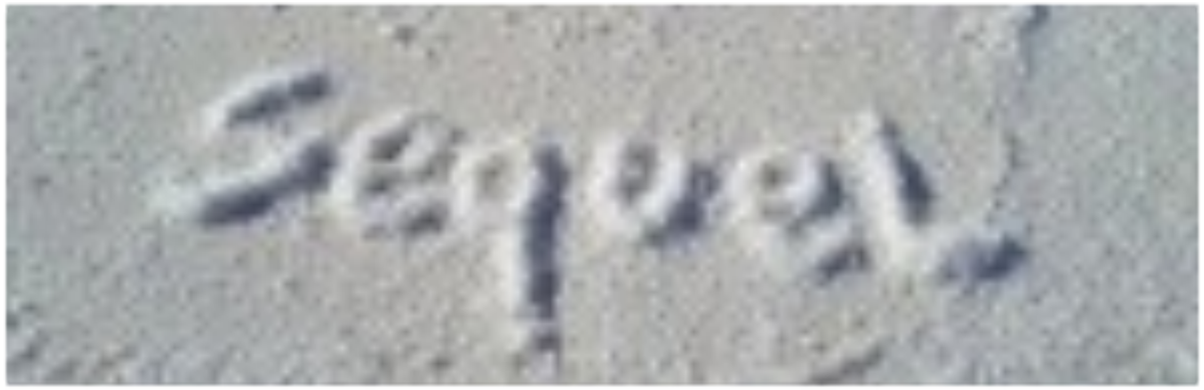


+

Philippe Preux
SequeL, Inria

Rémi Munos
Google DeepMind

10 YEARS



. . . LAST 10 YEARS AND INDUSTRY

Inria informatics mathematics

criteo

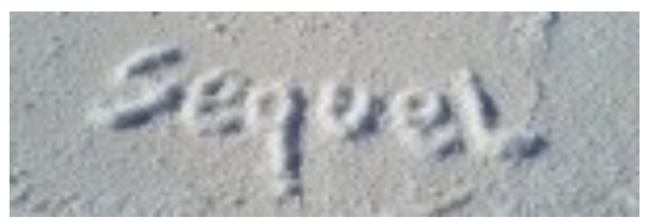
J. Mary
M. Abeille

M. Davy

vekia THE HEART OF RETAIL

Adobe

M. Ghavamzadeh



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

A. Lazaric

Facebook AI Research PARIS

R. Coulom
CrazyStone

UNBALANCE

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**

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

3 ACTIVE HIGHLIGHTS… LAST 12 MONTHS

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

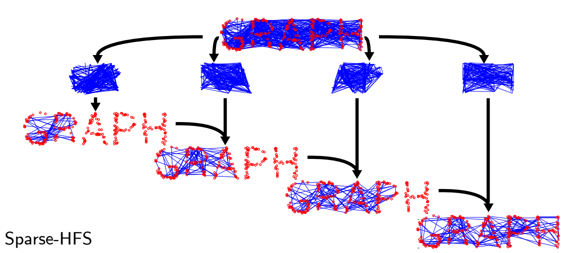

Blazing the trails before beating the path: Sample-efficient Monte-Carlo planning

Jean-Bastien Grill, Michal Valko — SequeL team, INRIA Lille - Nord Europe, France
jean-bastien.grill@inria.fr, 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 got brains and so you *plan* before you act. Luckily, your roboparents 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, NIPS 2017
Calandriello, Lazaric, Valko



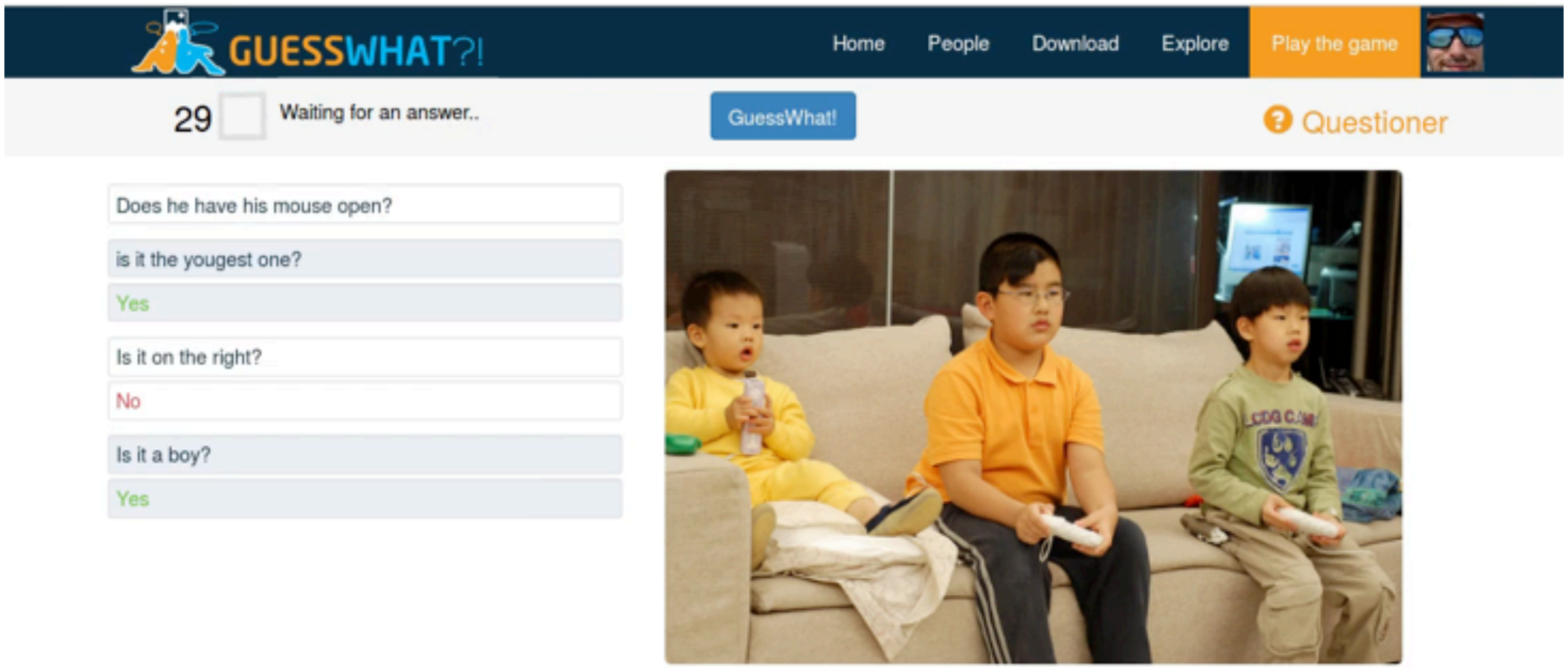
Algorithm	parkinson $n = 5,875$, $d = 20$			cpusmall $n = 8,192$, $d = 12$		
	Avg. Squared Loss	#SV	Time	Avg. Squared Loss	#SV	Time
FOGD	**0.04909** ± 0.00020	30	—	0.02577 ± 0.00050	30	—
NOGD	**0.04896** ± 0.00068	30	—	0.02559 ± 0.00024	30	—
PROS-N-KONS	0.05798 ± 0.00136	18	5.16	0.02494 ± 0.00141	20	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	—	—

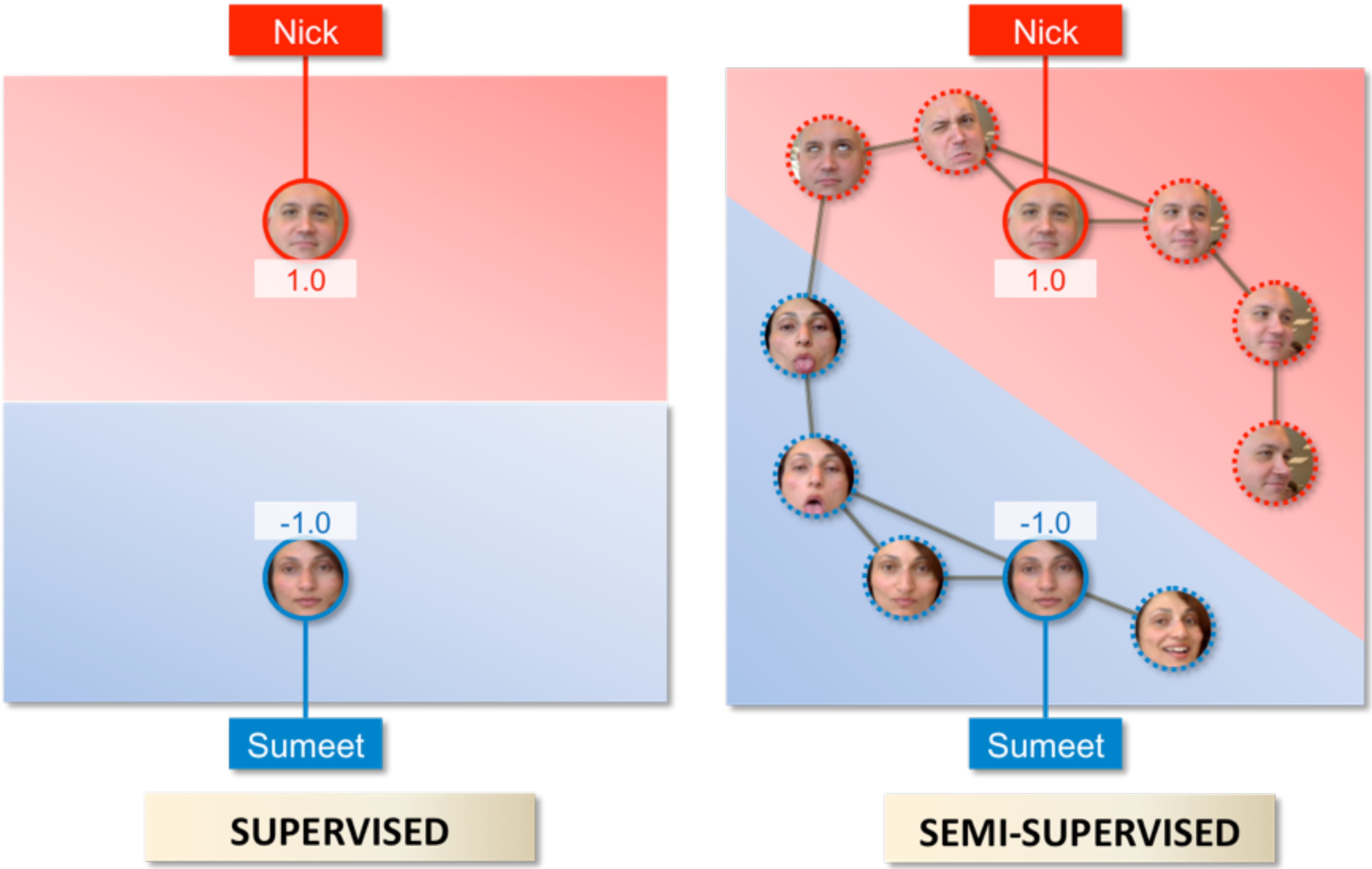
http://GuessWhat.AI
deep RL for dialogue in natural language
Strub, de Vries, Mary, Pietquin, Courville, Larochelle

GUESSWHAT.AI



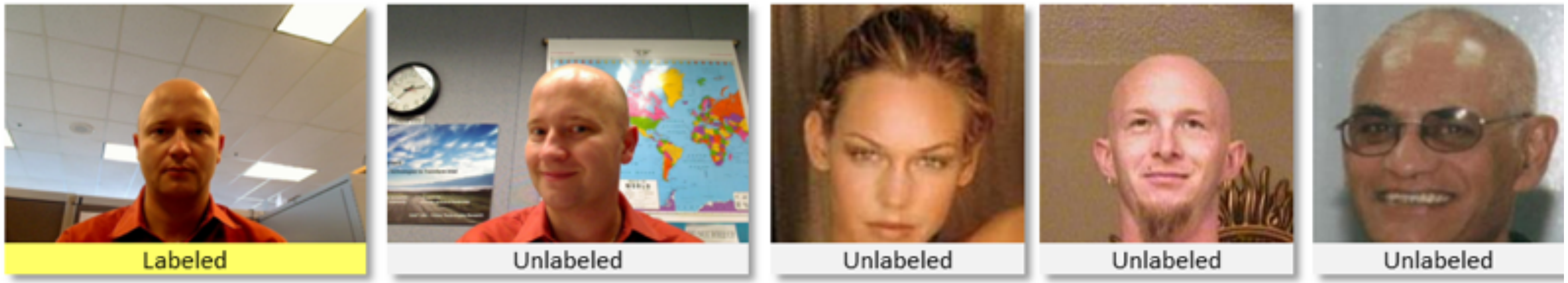
MY RESEARCH: MINIMAL FEEDBACK

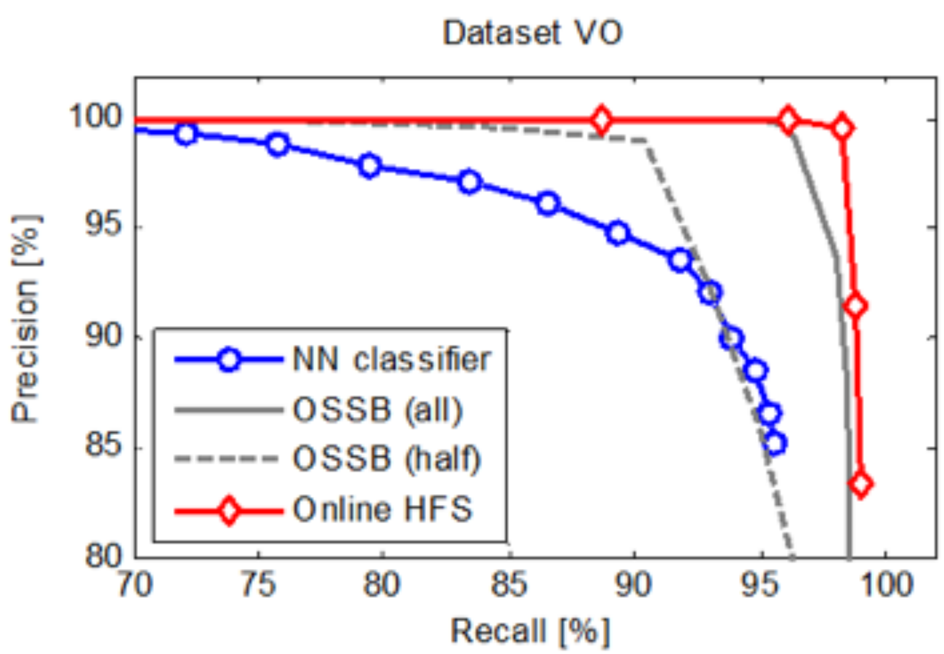
EXAMPLE 1: SEMI-SUPERVISED LEARNING

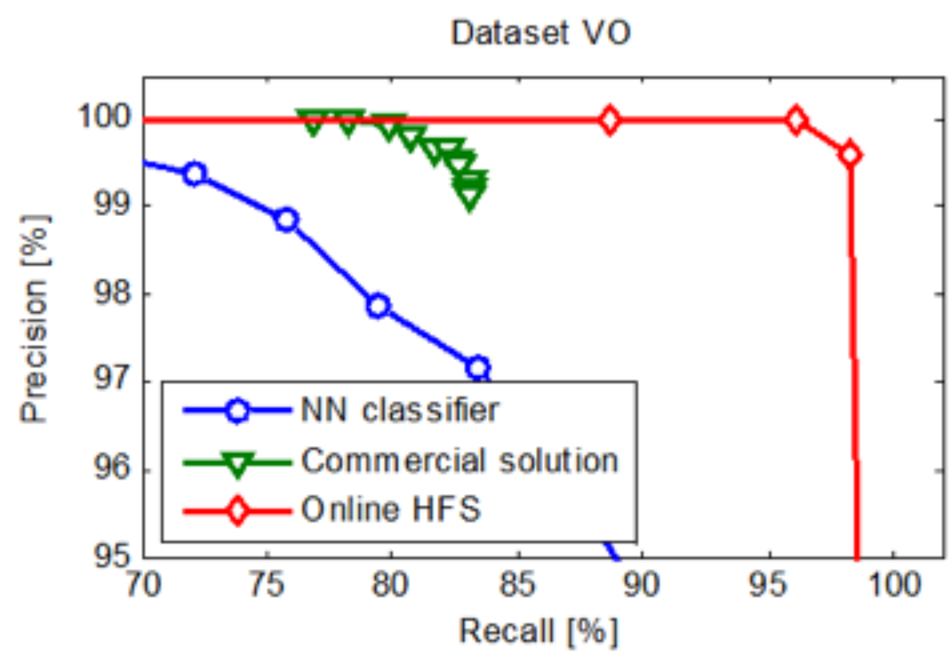


FACE RECOGNIZER THAT LEARNS ON THE FLY

- One person moves among various indoor locations
- 4 labeled examples of a person in the cubicle





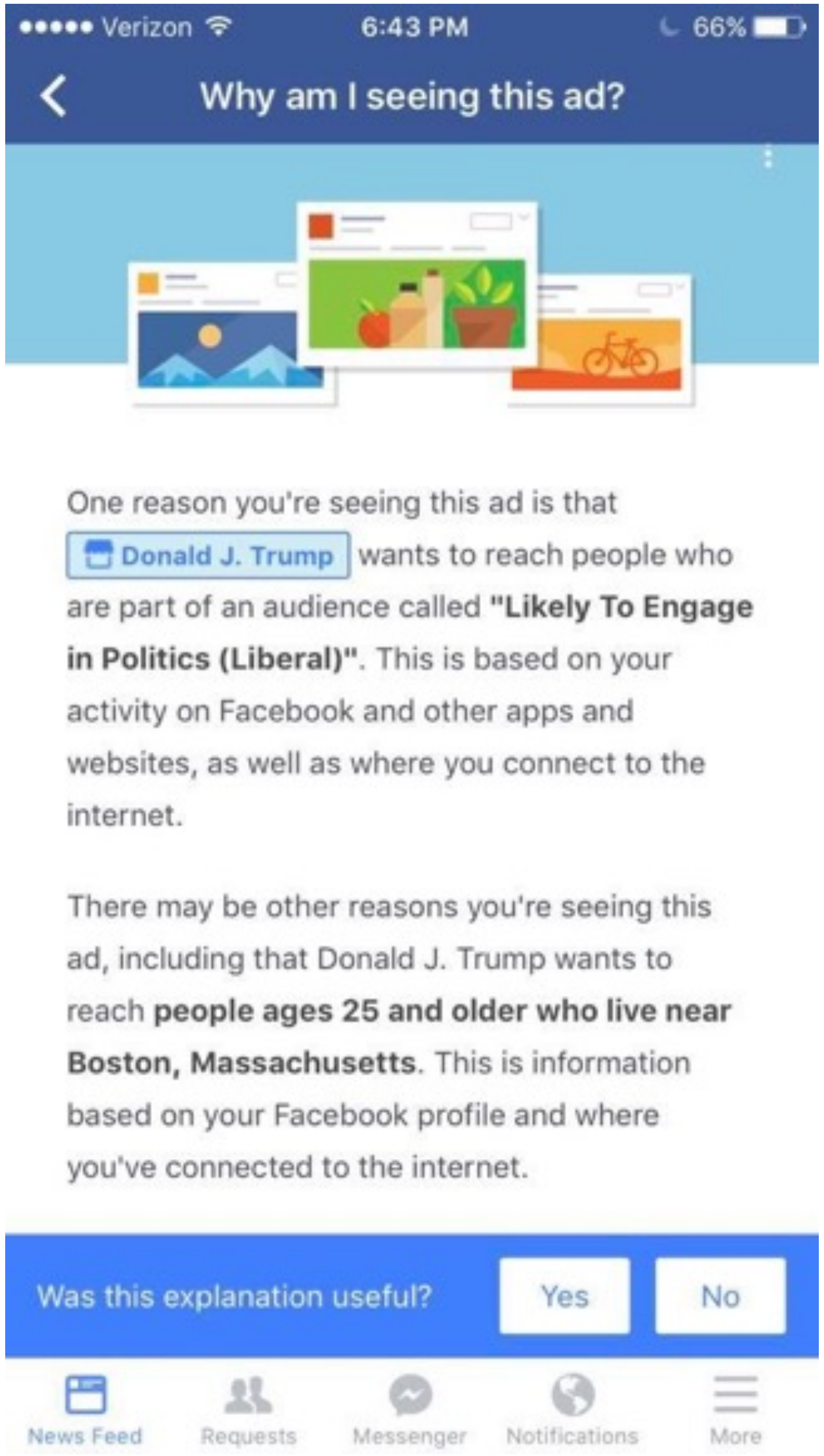


Online HFS outperforms OSSB (even when the weak learners are chosen using future data)

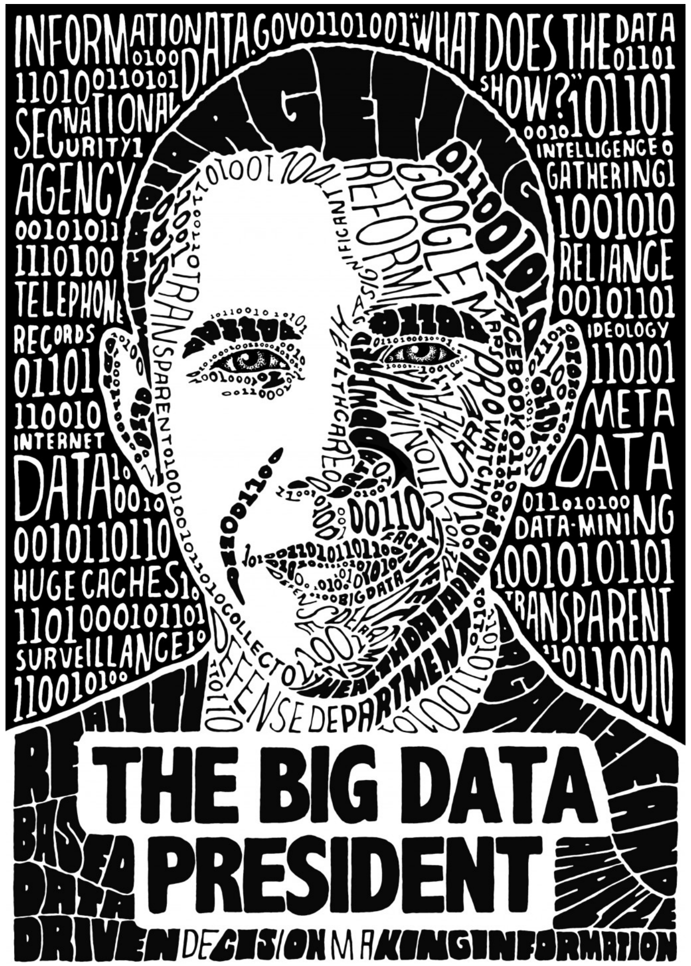
Online HFS yields better results than a commercial solution at 20% of the computational cost

FINAL PRODUCT

HOW TO RULE THE WORLD?



“IA” EST DÉJÀ LÀ



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

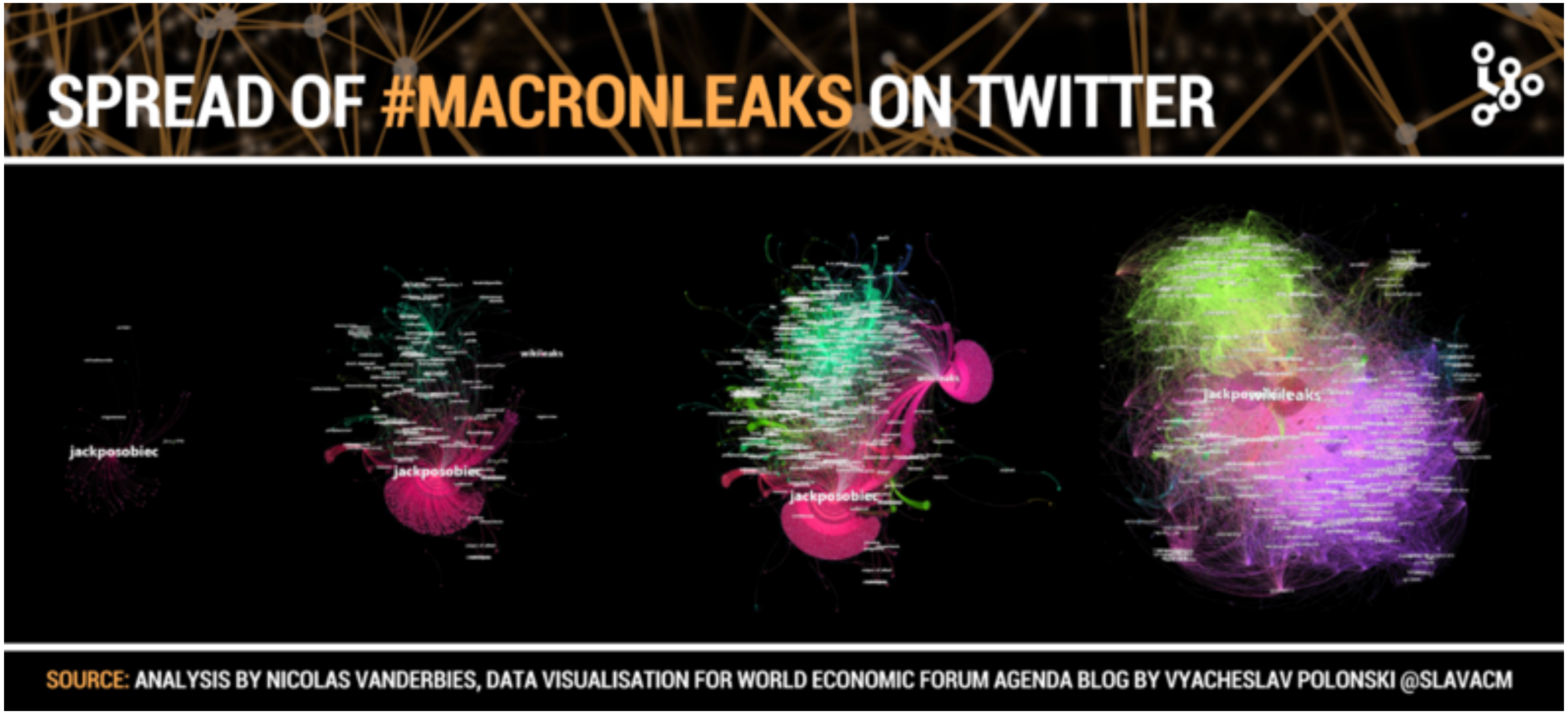
https://www.technologyreview.com/s/509026/how-obamas-team-used-big-data-to-rally-voters/

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

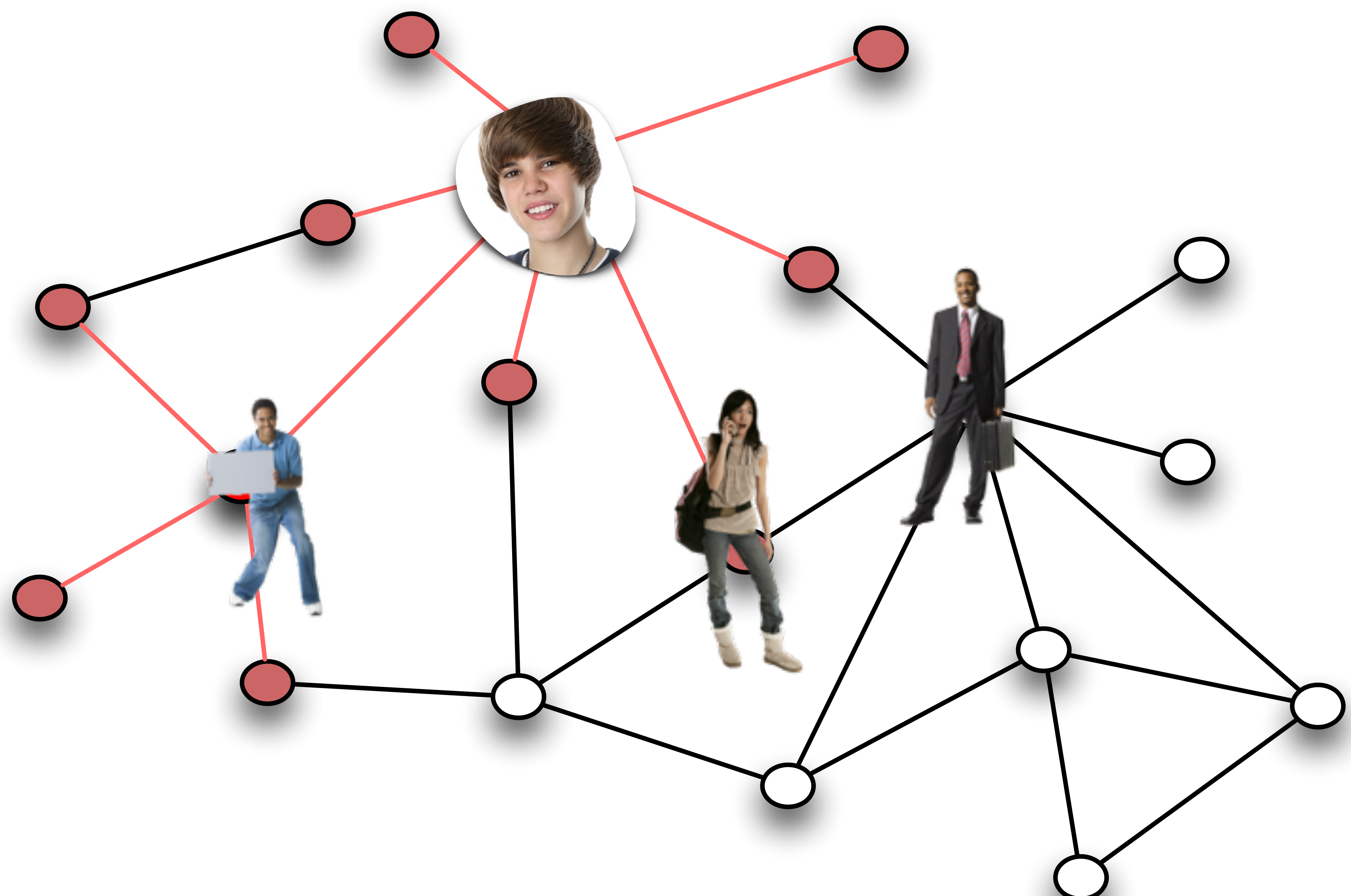
Religion

March 26, 2017

Politics

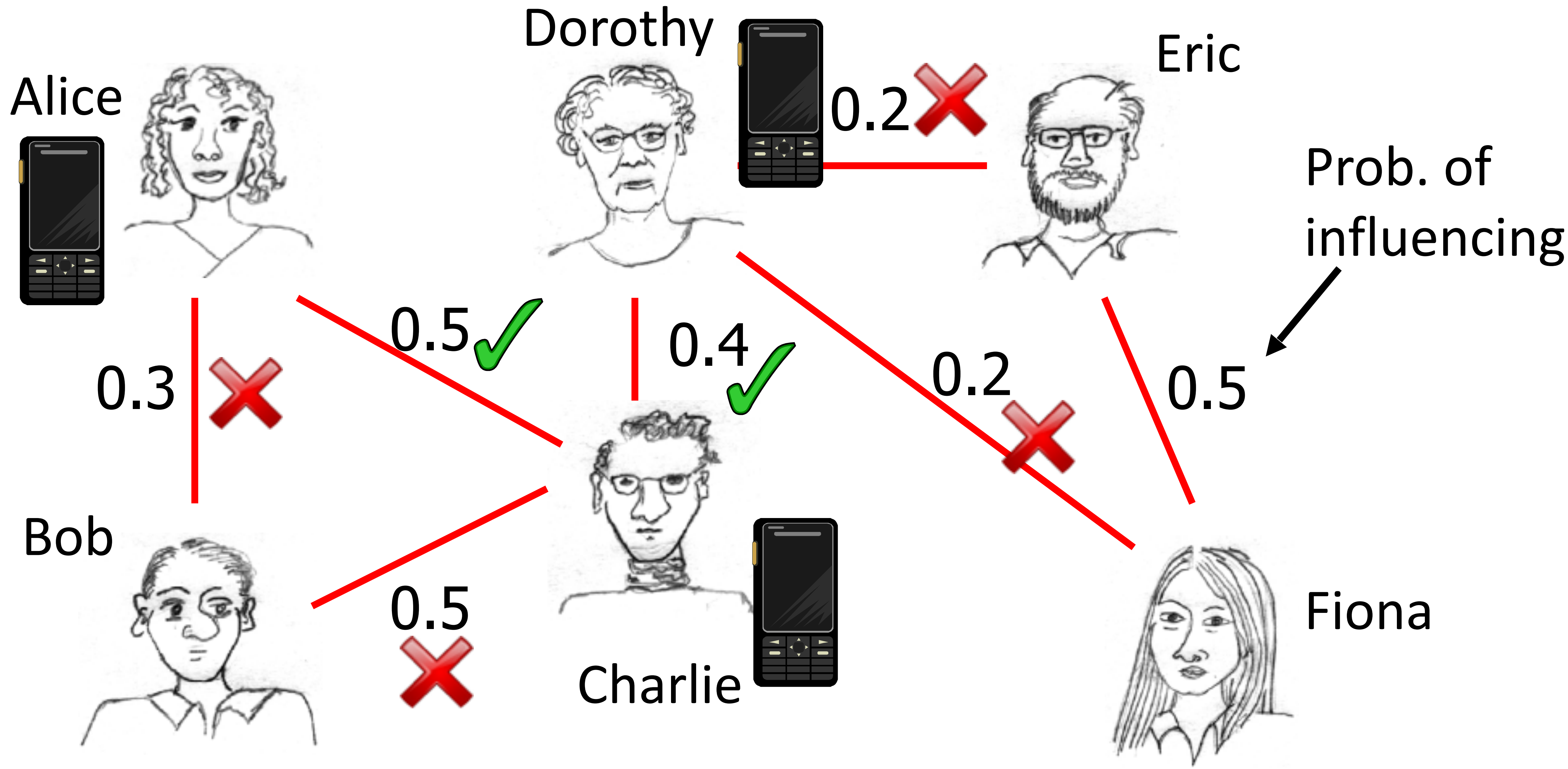
September 1, 2009

Culture



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

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

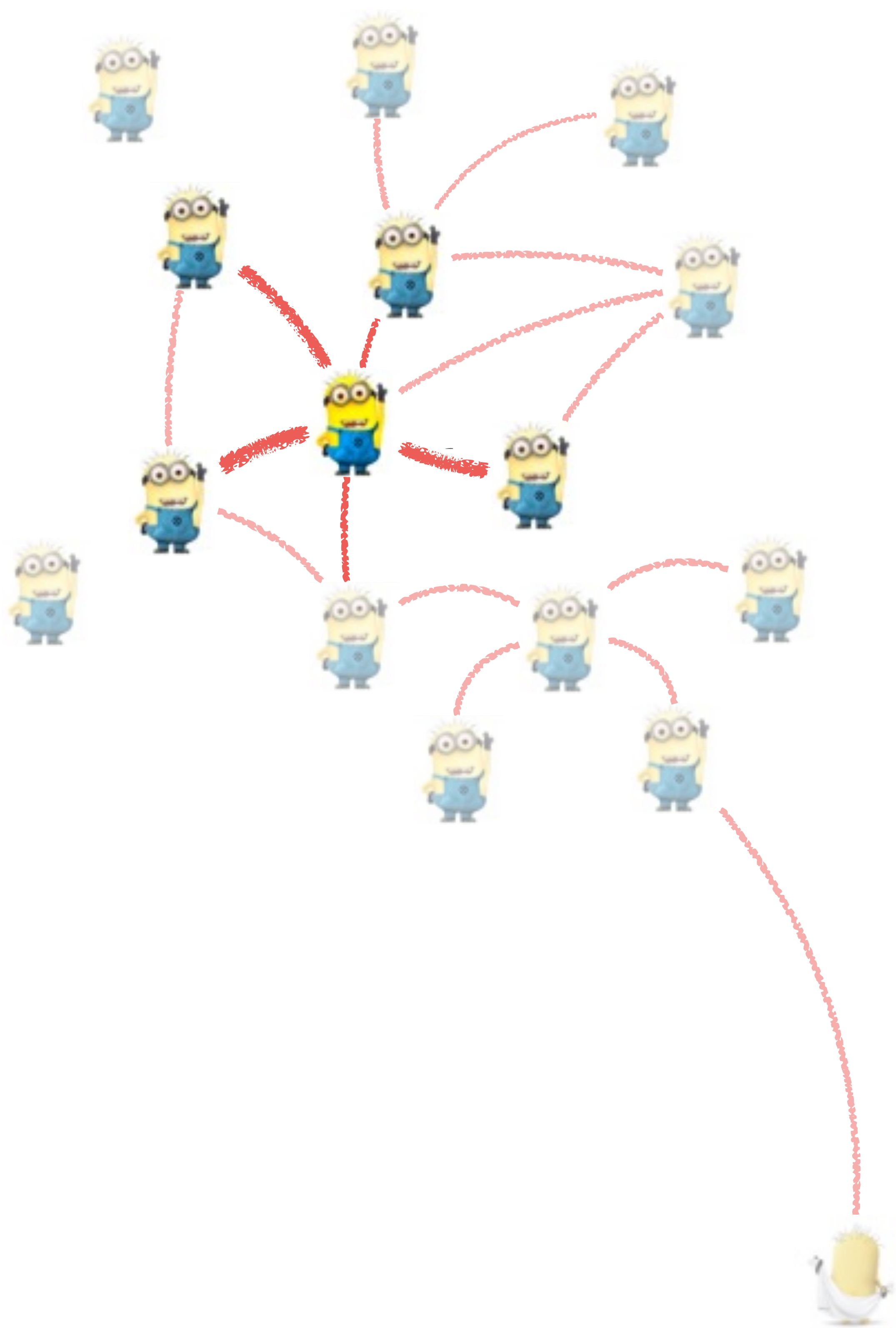


Who should get free cell phones?

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

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

MAXIMIZING INFLUENCE



Product placement

- dispatch few to sell more
- target influential people

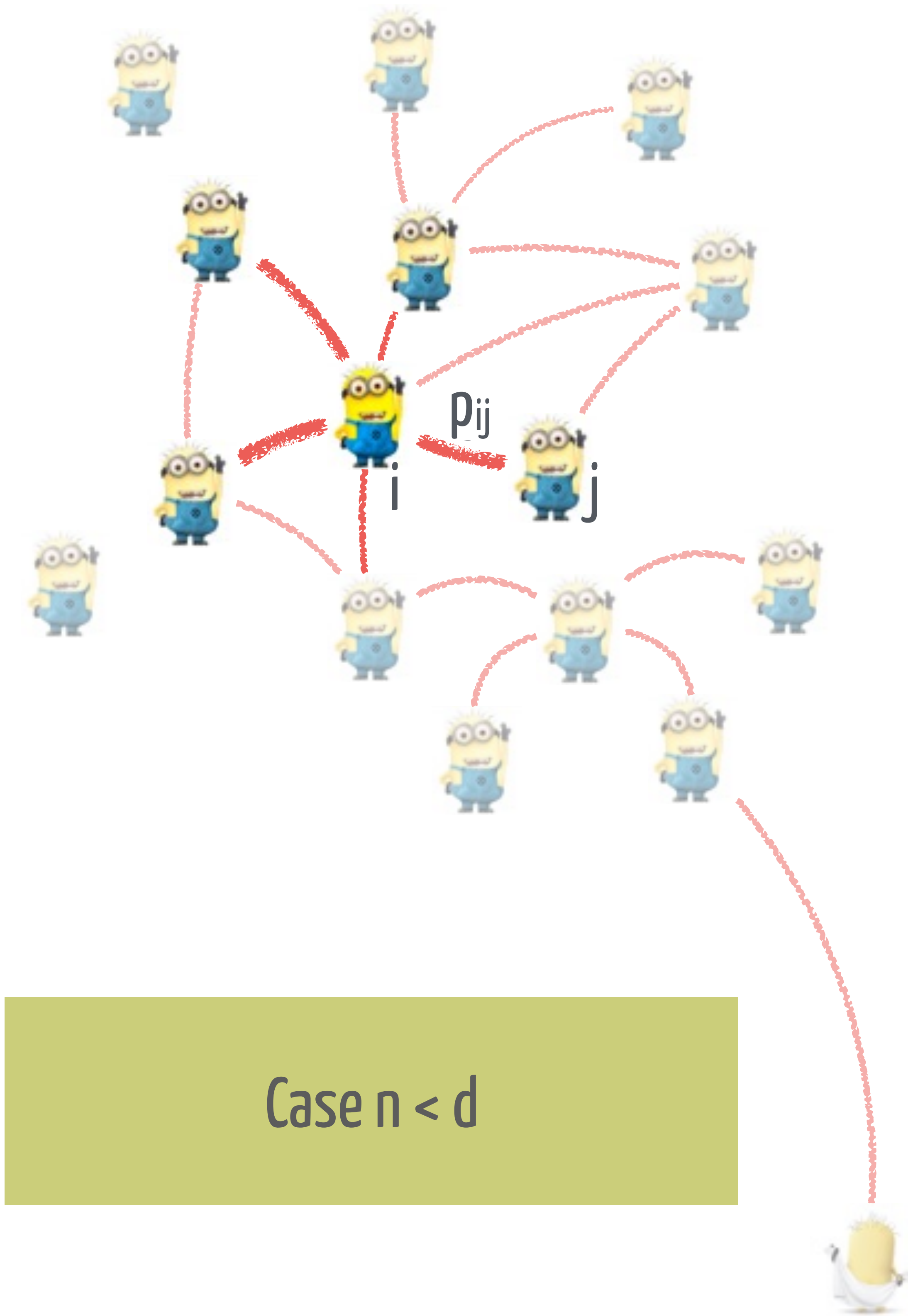
Gathering the information?

- 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, ..., 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?

PERFORMANCE CRITERION

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

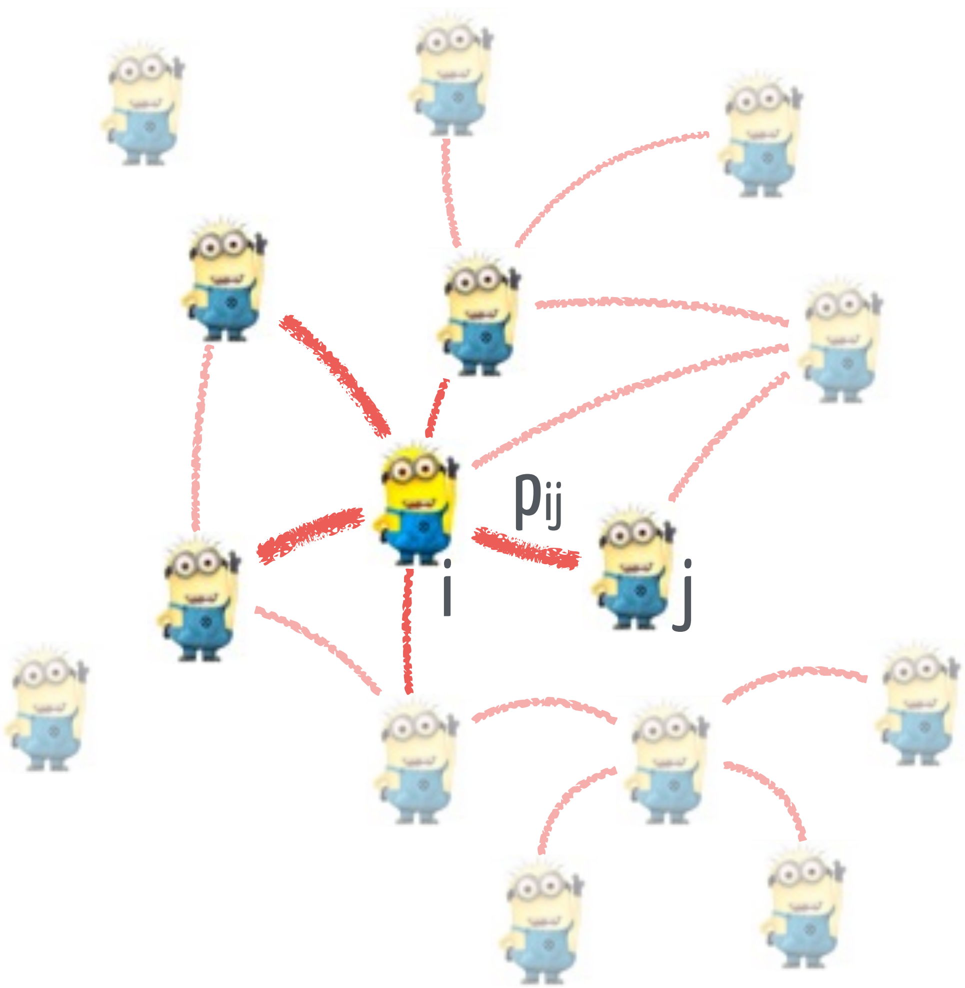
$$r_k = \mathbb{E}\left[|S_{k,t}|\right] = \sum_{j \le d} p_{k,j}$$

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 reward of the oracle strategy

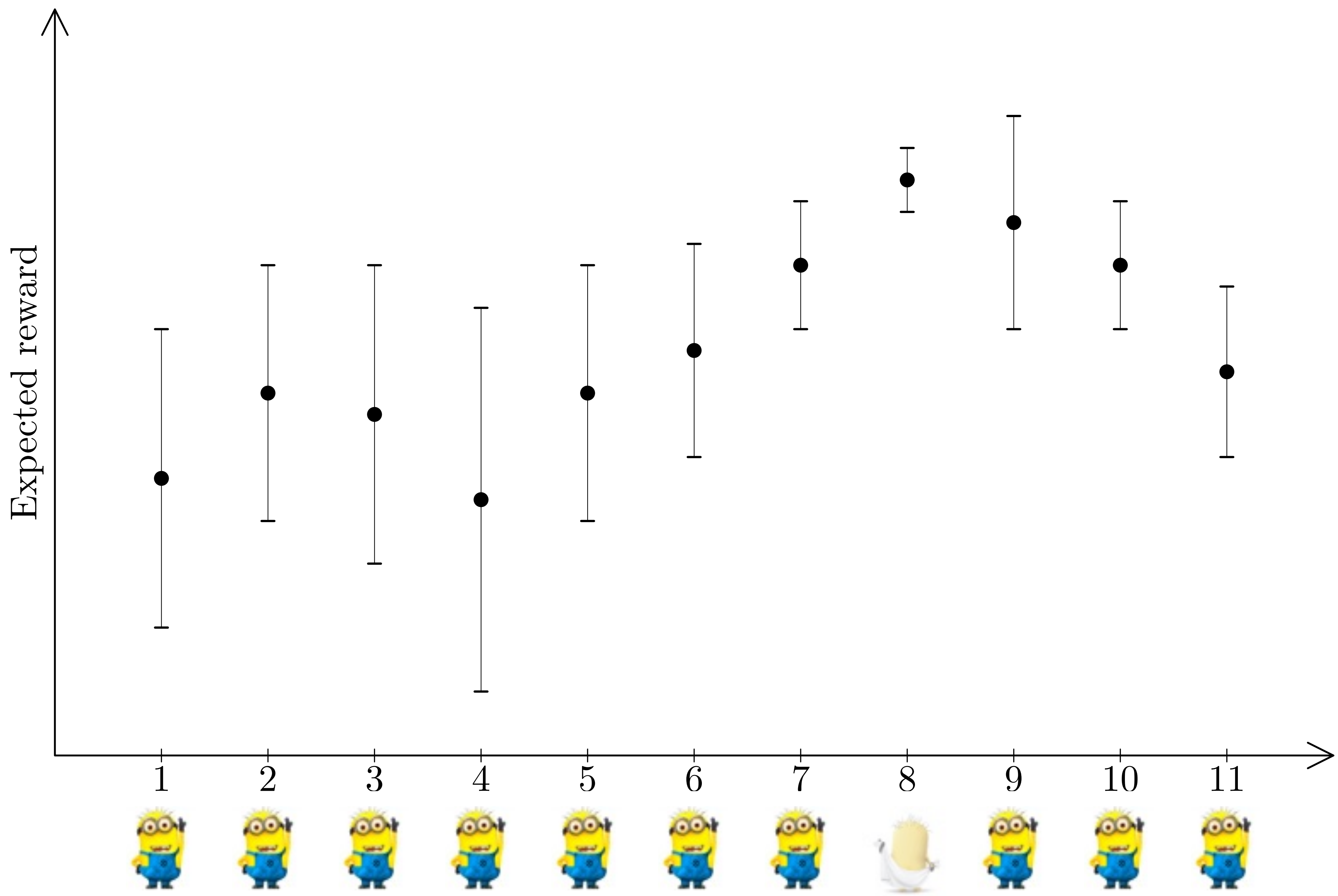
$$\mathbb{E}\left[L_n^\star\right] = n r_\star$$



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

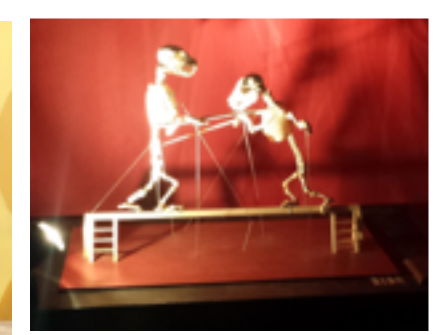
$$\mathbb{E}\left[R_n\right] = \mathbb{E}\left[L_n^\star\right] - \mathbb{E}\left[L_n\right]$$

UPPER CONFIDENCE BOUND BASED ALGOS

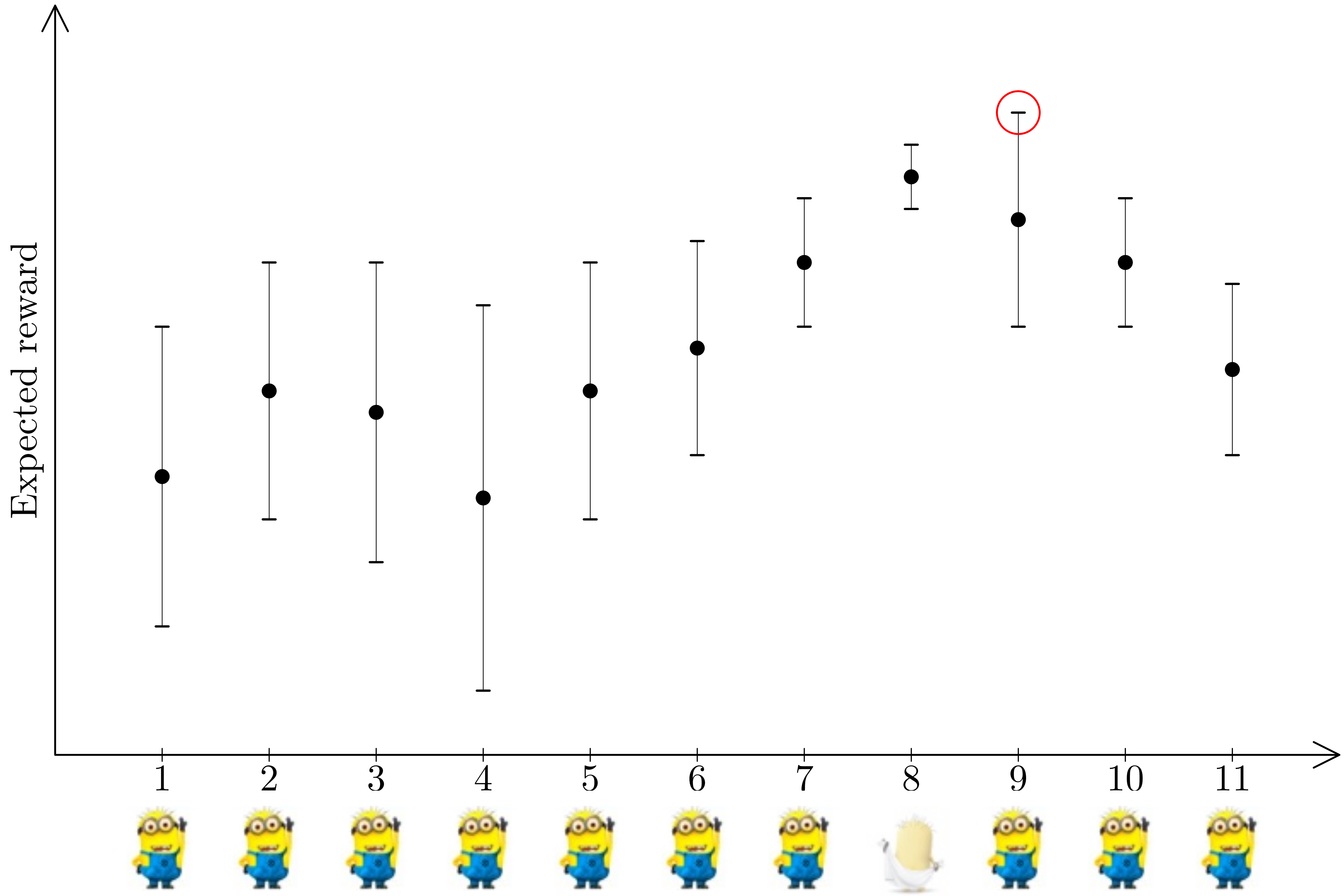


MULTI-ARM BANDITS IN CAFÉ CULTURE

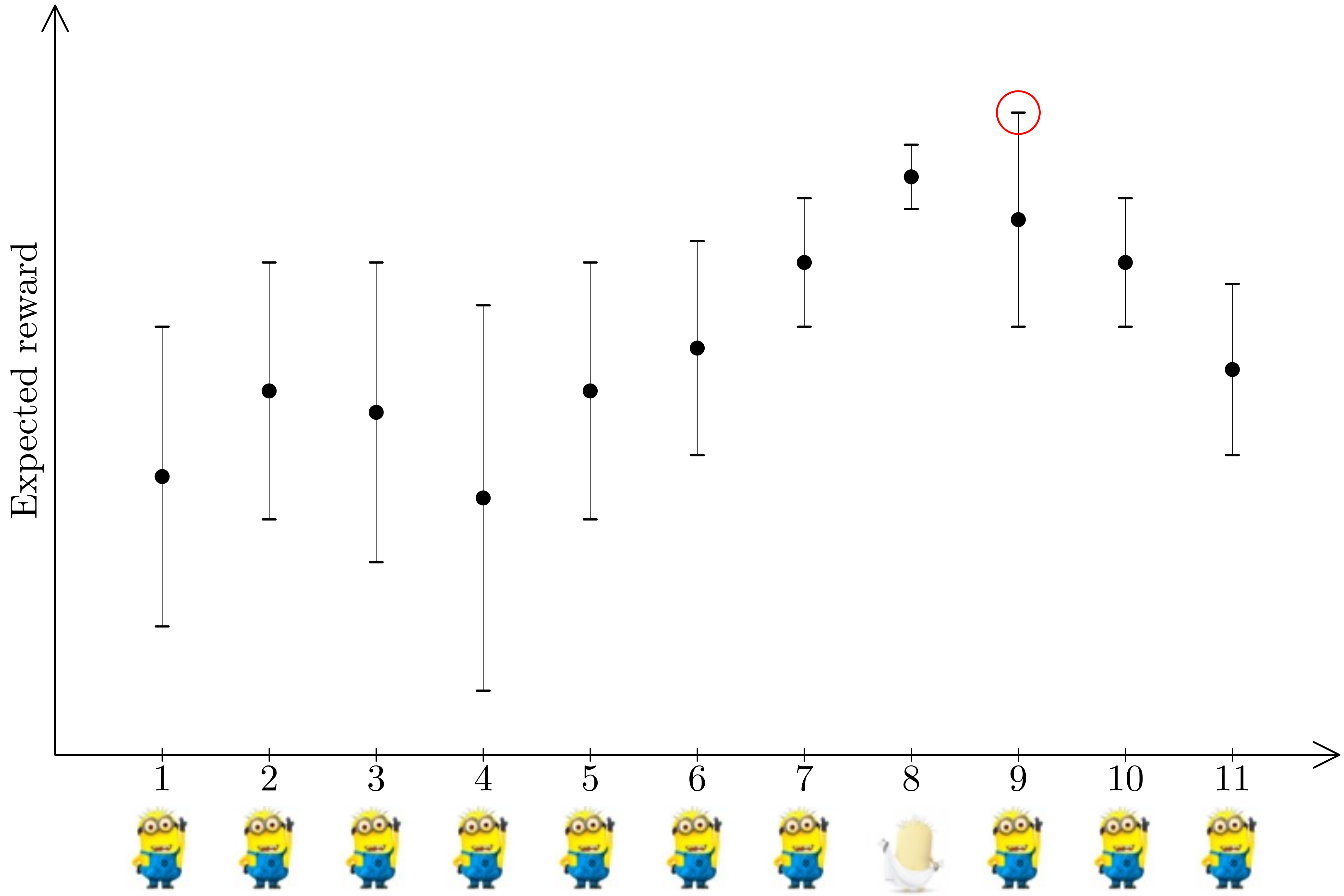
Video recorded March 30th, 2015, 13h50,
Université de Lille, Susie & the Piggy Bones Band



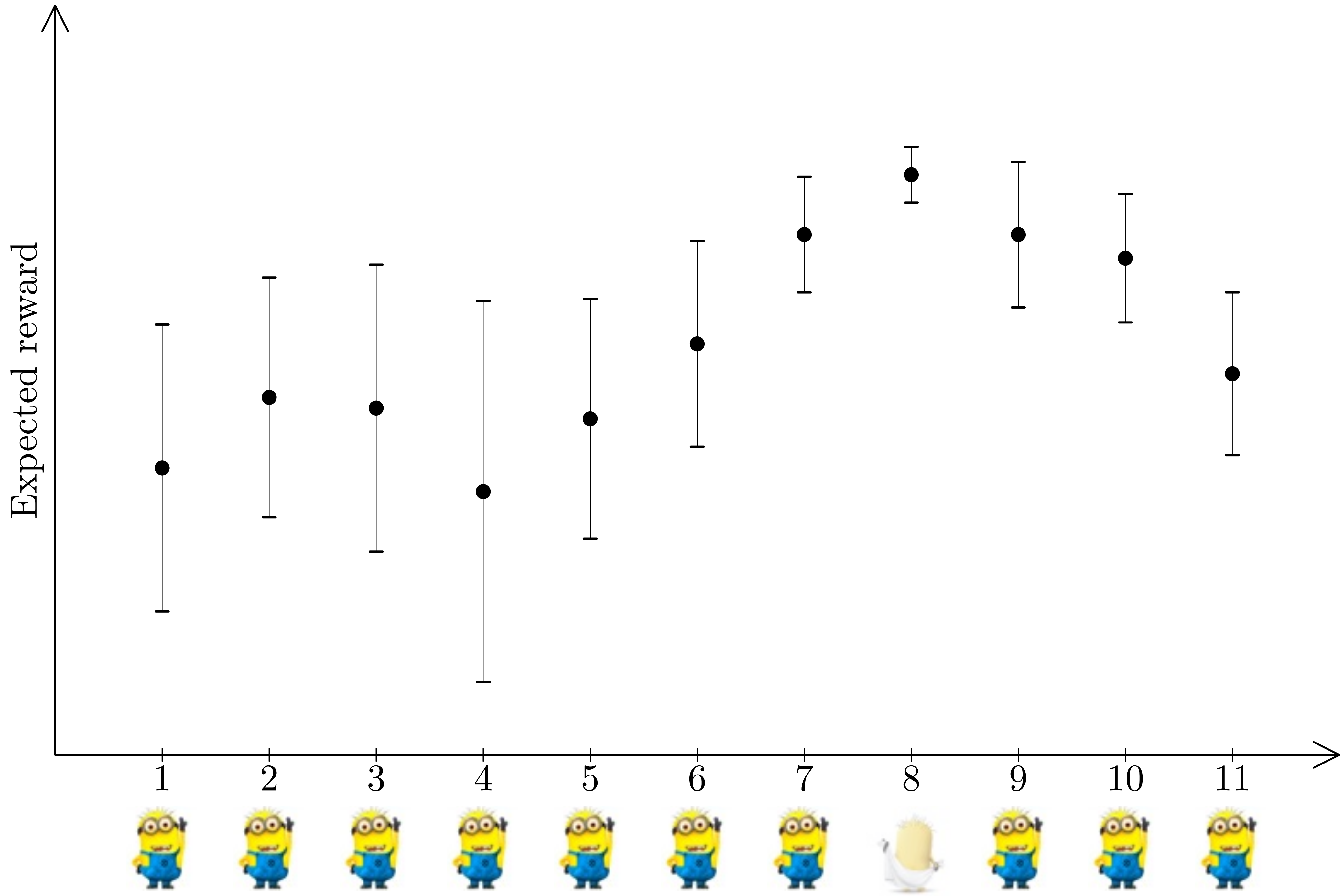
UPPER CONFIDENCE BOUND BASED ALGOS



UPPER CONFIDENCE BOUND BASED ALGOS



UPPER CONFIDENCE BOUND BASED ALGOS



EMPIRICAL RESULTS

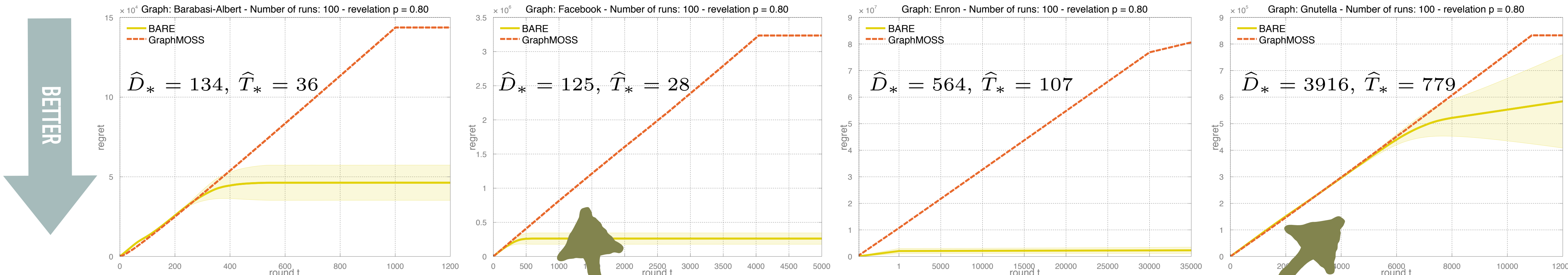
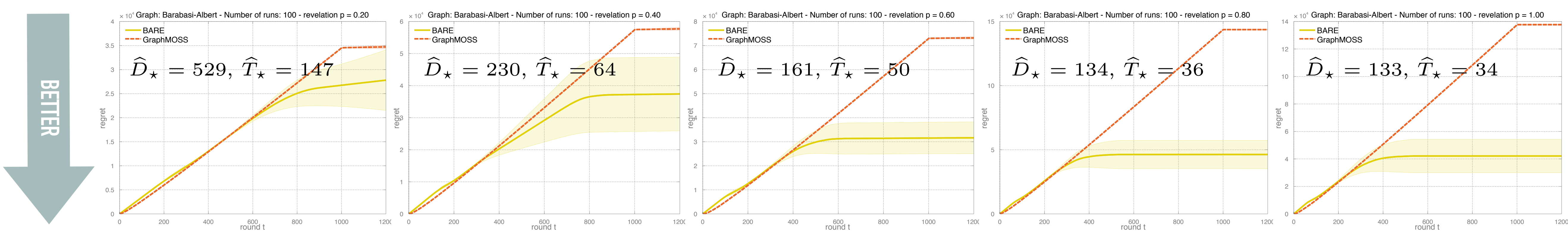


Figure 1: *Left*: Barabási-Albert. *Middle left*: Facebook. *Middle right*: Enron. *Right*: Gnutella.

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



Varying a (constant) probability of influence

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)
    - http://arxiv.org/abs/1605.06593 (Wen, Kveton, Valko, Vaswani, NIPS 2017)
    - IMLinUCB with linear parametrization of edge weights
    - Regret analysis for **general graphs**

CHALLENGES AND SOLUTIONS

- **Already the offline problem is NP hard**
  - solution: **approximation/randomized algorithms**

seed set

seed size

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

- **Lots of edges**
  - lots of parameters to learn, if we want to scale, we need to reduce this complexity
  - solution: **linear approximation of probabilities**
- **Combinatorial size of possible seed-sets**
  - Combinatorial Bandits: IMLinUCB
- **Understanding what's going on?**
  - known analyses VERY loose (e.g., scaling with $1/p_{min}$, or only assymptotic)

APPROXIMATION ORACLE

- the optimal offline solution

seed size

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

- the oracle solution that is γ-optimal with probability at least α

$$\mathcal{S}^* = \texttt{ORACLE}(\mathcal{G}, K, \overline{w})$$

- γ-optimal

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

- γ-optimal with probability at least α

$$\mathbb{E}\left[f(\mathcal{S}^*,\overline{w})\right] \geq \alpha\gamma f(\mathcal{S}^{\text{opt}},\overline{w})$$

- Our problem is a triple:

unknown to the agent

$$(\mathcal{G}, K, \overline{w})$$

topology

seed size

LINEAR GENERALIZATION

— learning the only network (weights) is VERY impractical

$$\rho \triangleq \max_{e \in \mathcal{E}} \left| \overline{w}(e) - x_e^\mathsf{T} \theta^* \right|$$

this is small

true weights

linear approximation

— by choosing the **d**imension (size of θ^*) we can reduce this complexity

— if we do not want to lose generality we set **d** to the number of edges

ALGORITHM AND PERFORMANCE MEASURE

Algorithm 1 `IMLinUCB`: Influence Maximization Linear UCB

Input: graph $\mathcal{G}$, source node set cardinality K, oracle `ORACLE`, feature vector x_e's, and algorithm parameters $\sigma, c > 0$,

Initialization: $B_0 \leftarrow 0 \in \Re^d$, $\mathbf{M}_0 \leftarrow I \in \Re^{d \times d}$

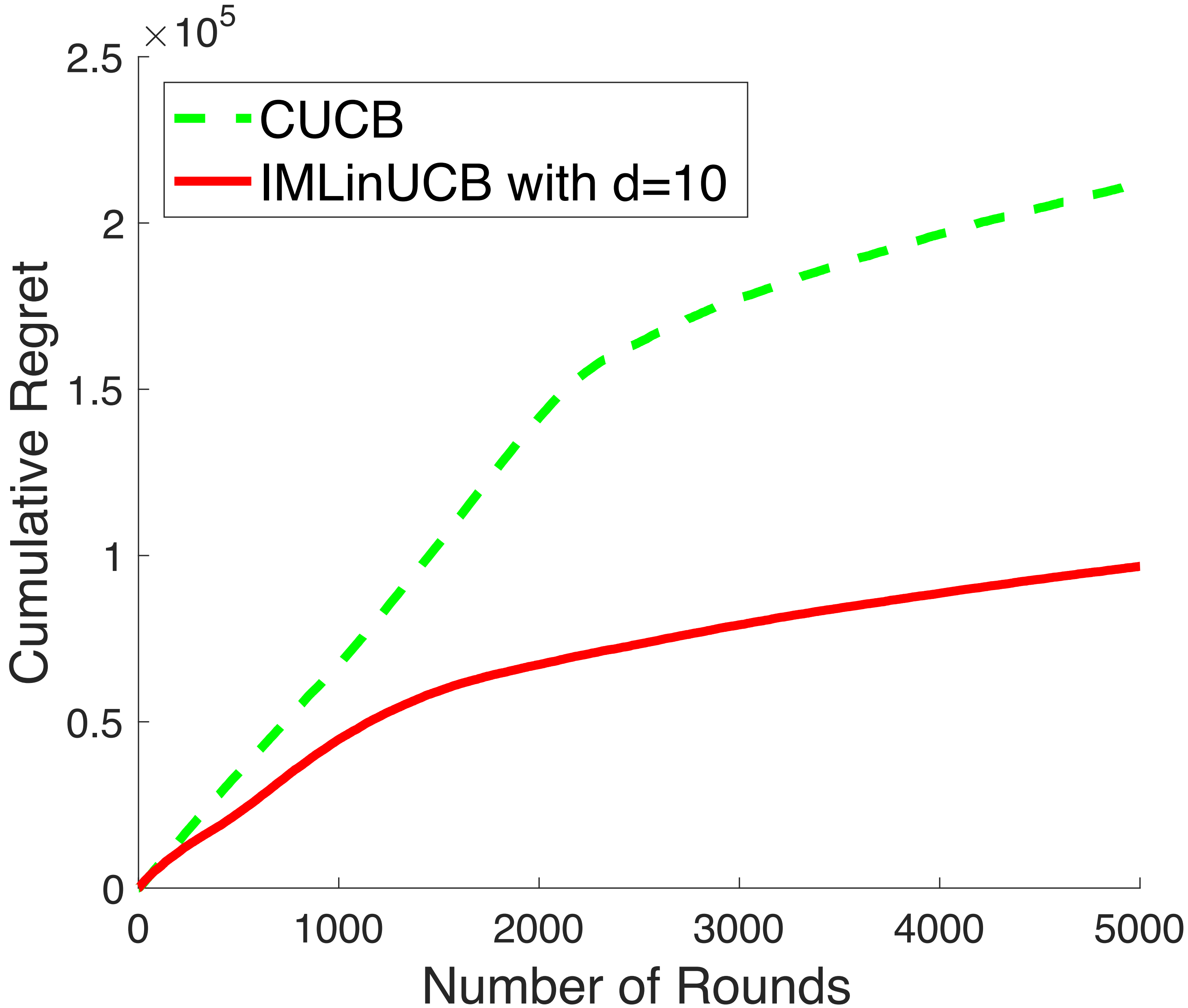
for $t = 1, 2, \dots, n$ **do**

1. set $\bar{\theta}_{t-1} \leftarrow \sigma^{-2}\mathbf{M}_{t-1}^{-1}B_{t-1}$ and the UCBs as $U_t(e) \leftarrow \mathrm{Proj}_{[0,1]}\left(x_e^\top \bar{\theta}_{t-1} + c\sqrt{x_e^\top \mathbf{M}_{t-1}^{-1} x_e}\right)$ for all $e \in \mathcal{E}$
2. choose $\mathcal{S}_t \in$ `ORACLE`$(\mathcal{G}, K, U_t)$, and observe the edge-level semi-bandit feedback
3. update statistics:
   (a) initialize $\mathbf{M}_t \leftarrow \mathbf{M}_{t-1}$ and $B_t \leftarrow B_{t-1}$
   (b) for all observed edges $e \in \mathcal{E}$, update $\mathbf{M}_t \leftarrow \mathbf{M}_t + \sigma^{-2}x_e x_e^\top$ and $B_t \leftarrow B_t + x_e \mathbf{w}_t(e)$

$$R^\eta(n) = \sum_{t=1}^{n} \mathbb{E}\left[R_t^\eta\right]$$

$$R_t^\eta = f(\mathcal{S}^{\mathrm{opt}}, \mathbf{w}_t) - \frac{1}{\eta} f(\mathcal{S}_t, \mathbf{w}_t)$$

FACEBOOK EXPERIMENT

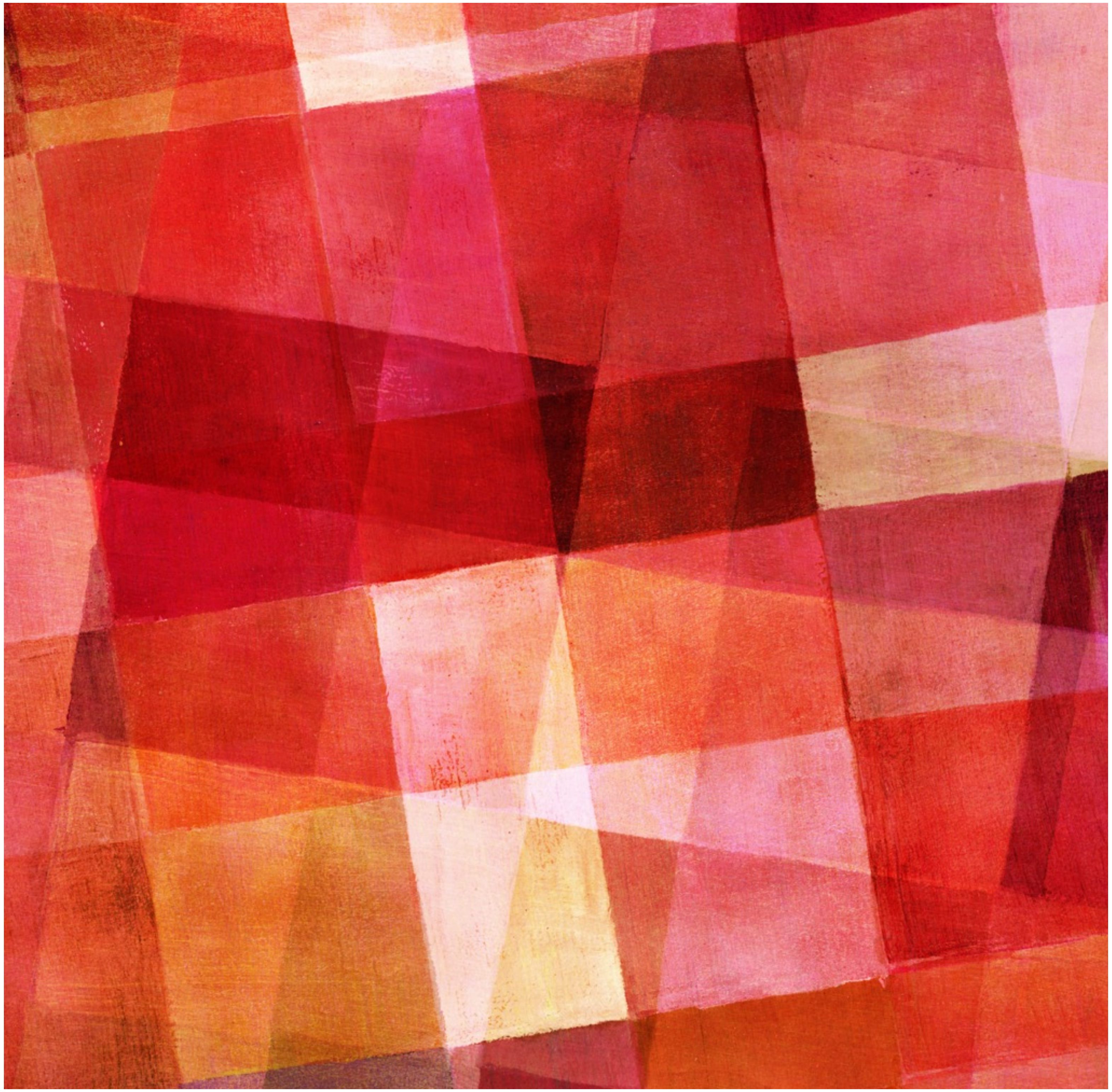


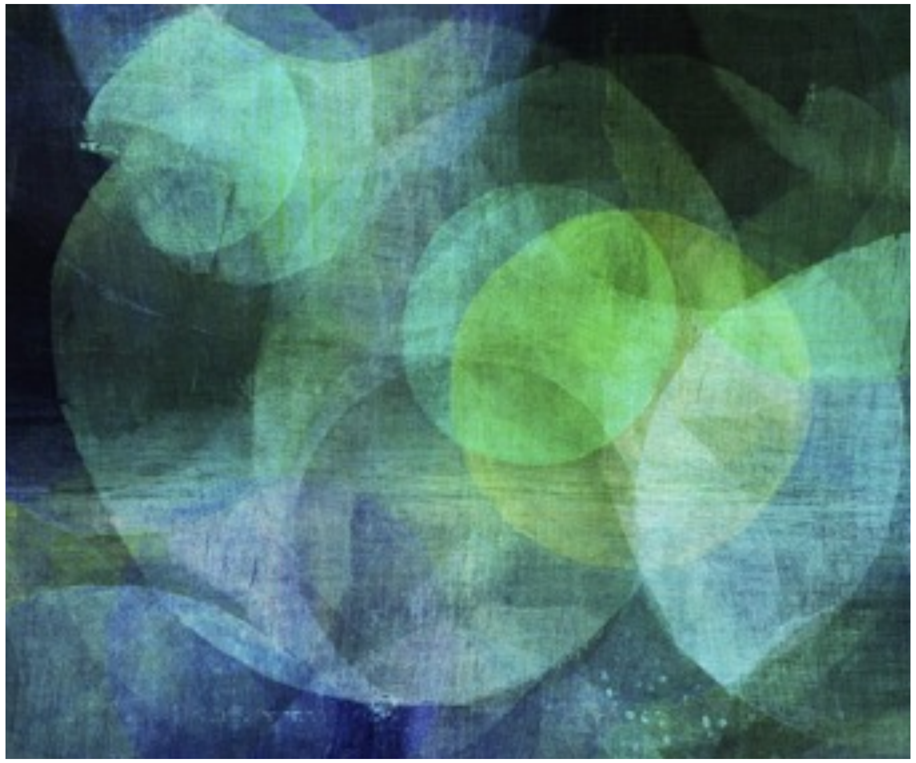
- real Facebook (a small subgraph)
- weights from U(0,0.1)
- **nodetovec** with d=10
  - imperfect
- K = 10
- CUCB with no linear generalisation

What is next?

MAIS OÙ SE CACHE-T-IL ?

photo(c) Julia@SequeL





ExtraLearn

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