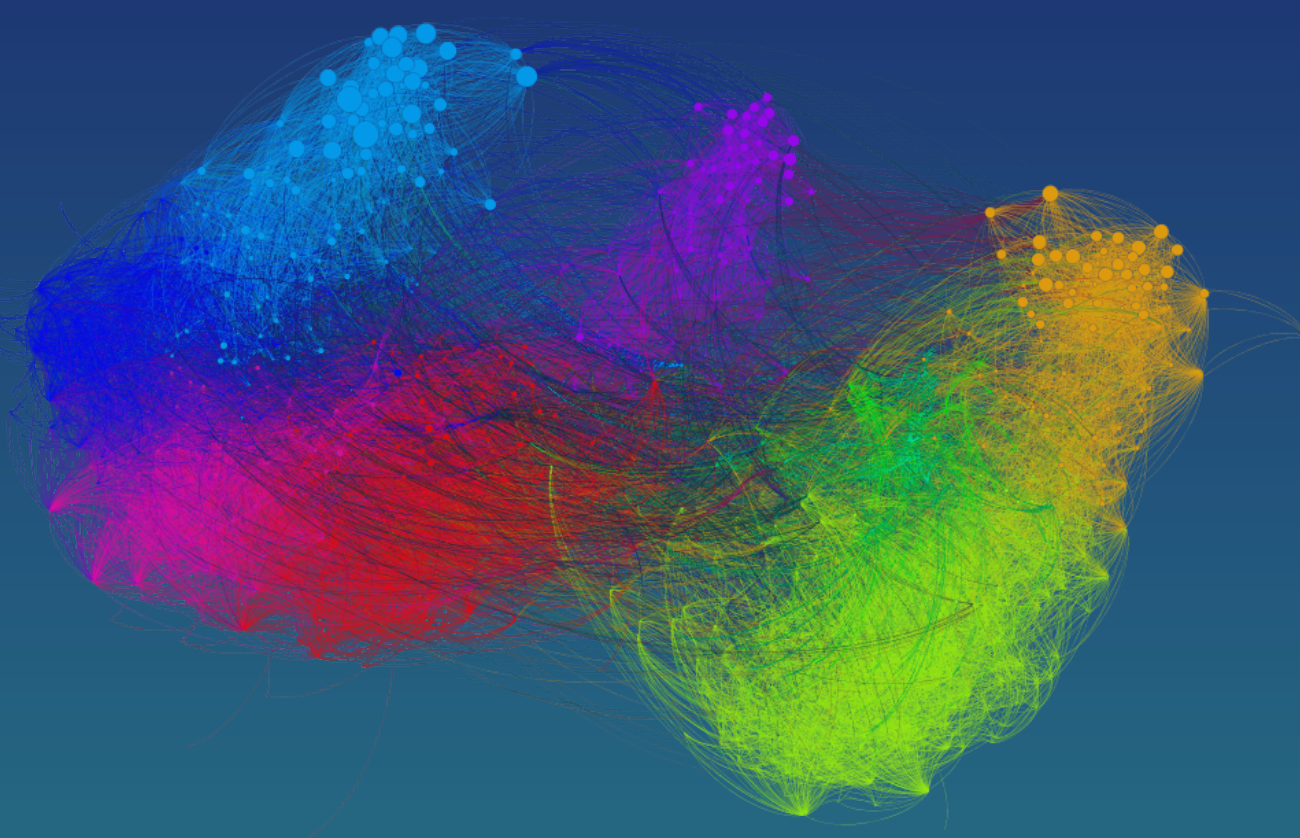
Graphs in Machine Learning

Michal Valko

Inria Lille - Nord Europe, France

TA: Daniele Calandriello

Partially based on material by: Tomáš Kocák, Nikhil Srivastava, Yiannis Koutis, Joshua Batson, Daniel Spielman



LAST LECTURE

- Scaling harmonic functions to millions of samples
- Graph Sparsification
- Spectral Sparsification

THIS LECTURE

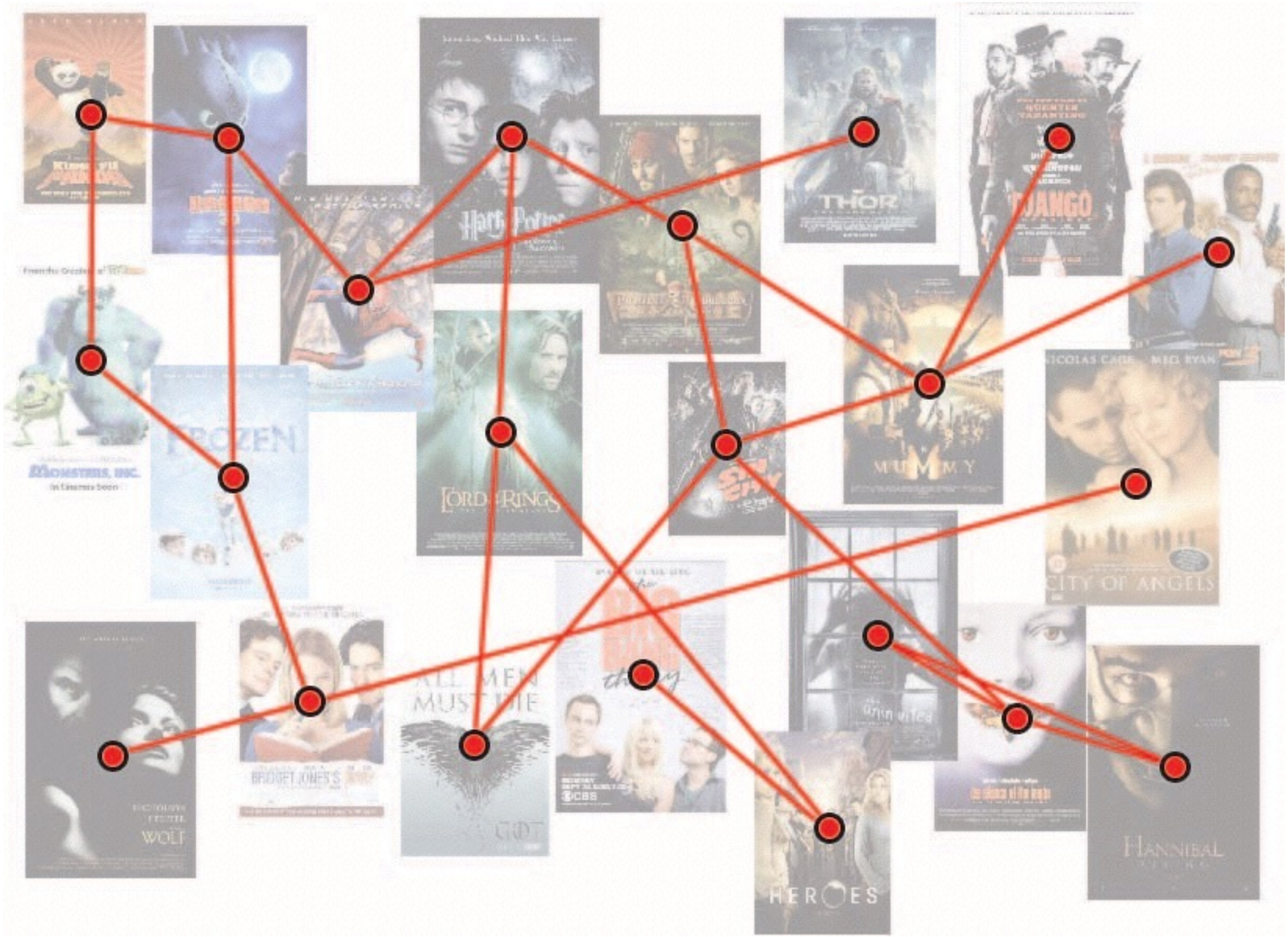
- Graph bandits
    - Spectral bandits
    - Observability graphs
    - Side information
    - Influence Maximization

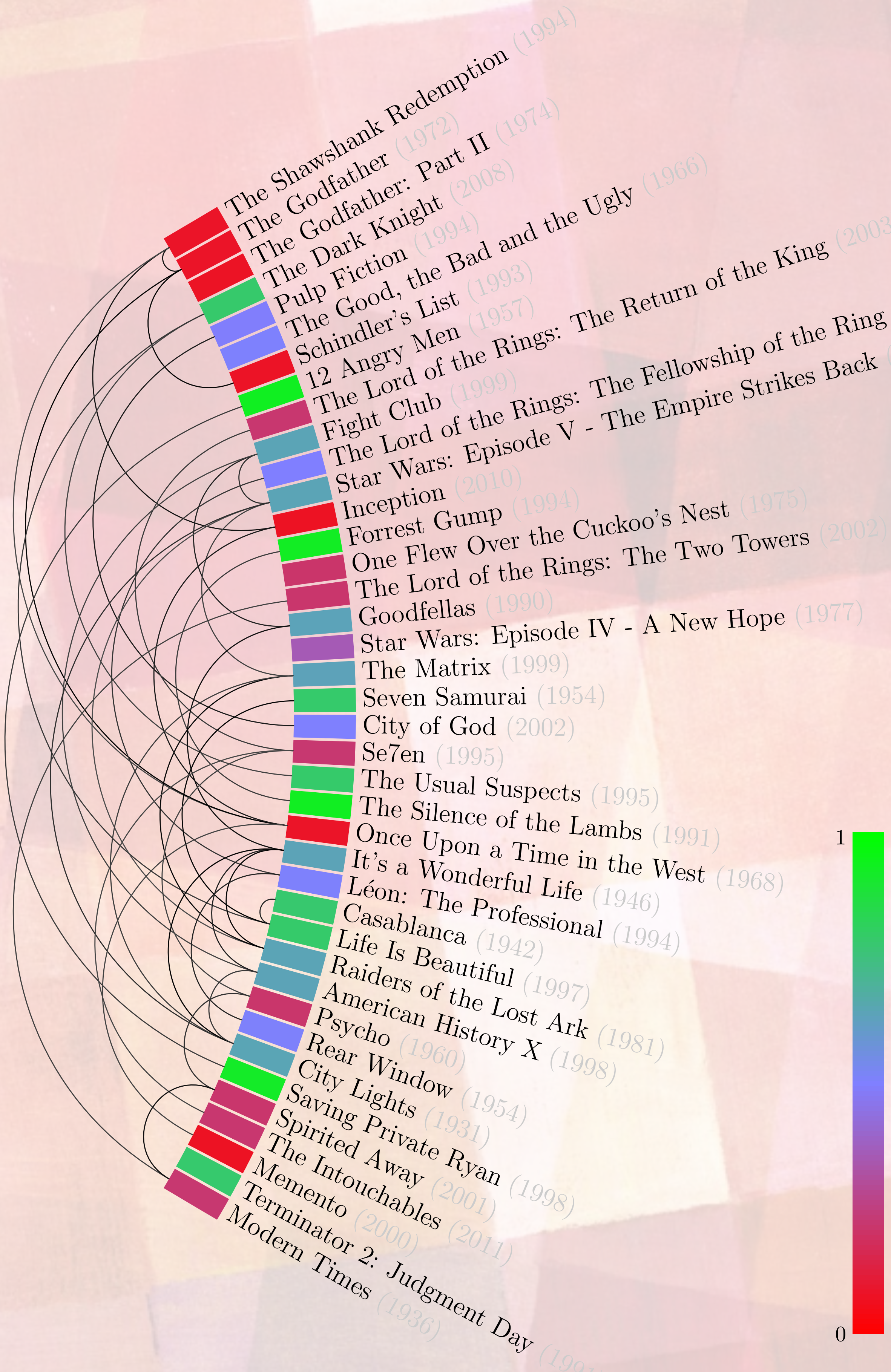
PREVIOUS LAB SESSION

- 28. 11. 2016 by Daniele.Calandriello@inria.fr
- Content (this time lecture in class + coding at home)
  - Large-scale graph construction and processing (in class)
  - Scalable algorithms:
    - Online face recognizer (to code in Matlab)
    - Iterative label propagation (to code in Matlab)
    - Graph sparsification (presented in class)
- AR: record a video with faces
- Short written report
- Questions to piazza
- *Deadline:* 12. 12. 2016 (Today!)
- http://researchers.lille.inria.fr/~calandri/teaching.html

FINAL CLASS PROJECTS

- time and formatting description on the class website
- grade: report + short presentation of the **team**
- deadlines
  - 5. 1. 2017 final report (for all projects)
  - 9. 1. 2017, presentation in class (Cournot C102)
  - alternatively Jan 2017, remote presentations (other projects)
- project report: 5-10 pages in NIPS format
- presentation: 15+5 minutes (**time it!**)
- everybody has to present
- book presentation time slot on the website
- **explicitly state the contributions**


THOR
FROZEN
HEROES
CITY OF ANGELS
HANNIBAL
ALL MEN MUST DIE



Example of a graph bandit problem

movie recommendation

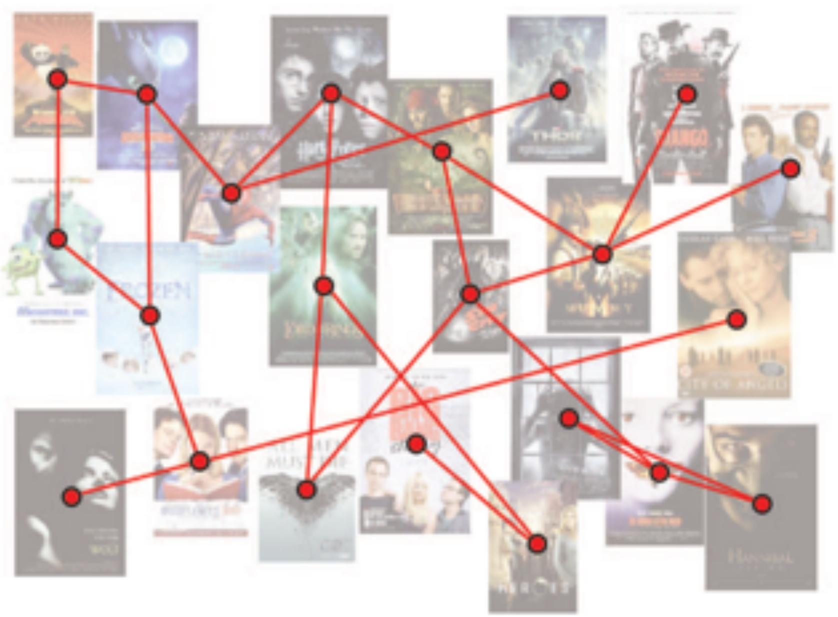
- recommend movies to a **single** user
- **goal:** maximise the sum of the ratings (minimise regret)
- good prediction after just a few steps

$$T \ll N$$

- extra information
  - ratings are **smooth** on a graph
- main question: can we learn **faster**?

GETTING REAL

Let's be lazy and ignore the structure



→

Multi-armed bandit problem!

Worst case regret (to the best fixed strategy)

$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions (N)

#rounds (T)

Matching lower bound (Auer, Cesa-Bianchi, Freund, Schapire 2002)

How big is N? Number of movies on **http://www.imdb.com/stats**: 4,003,294

Problem: Too many actions!

LEARNING FASTER

informatics mathematics

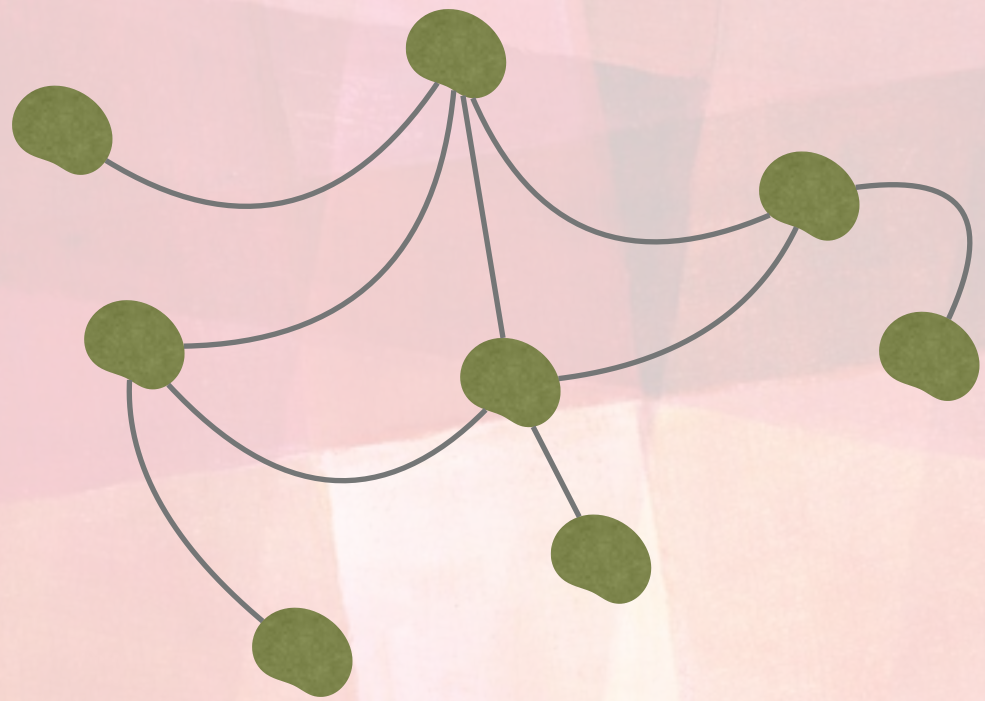
$$R_T = \mathcal{O}\left(\sqrt{NT}\right)$$

#actions

#rounds

#dimensions

- Arm independence is too strong and unnecessary
- Replace **N** with something much smaller
  - problem/instance/data dependent
  - example: linear bandits **N** to **D**
- **Today: Graph** Bandits!
  - sequential problems where **actions are nodes** on a graph
  - find strategies that replace **N** with a smaller **graph-dependent** quantity



GRAPH BANDITS: GENERAL SETUP

Every round **t** the learner

- picks a node $I_t \in [N]$
- incurs a loss ℓ_{t,I_t}
- optional feedback

The performance is total expected regret

$$R_T = \max_{i \in [N]} \mathbb{E}\left[\sum_{t=1}^{T} (\ell_{t,I_t} - \ell_{t,i})\right]$$

Specific problems differ in

1. loss
2. feedback
3. guarantees

STRUCTURES IN BANDIT PROBLEMS

GRAPHS

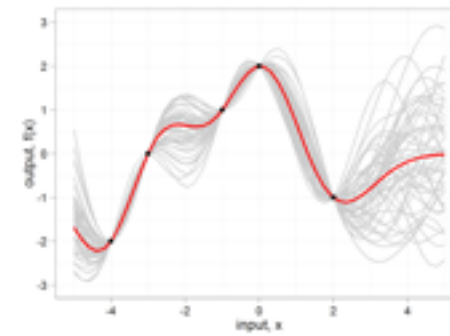
KERNELS

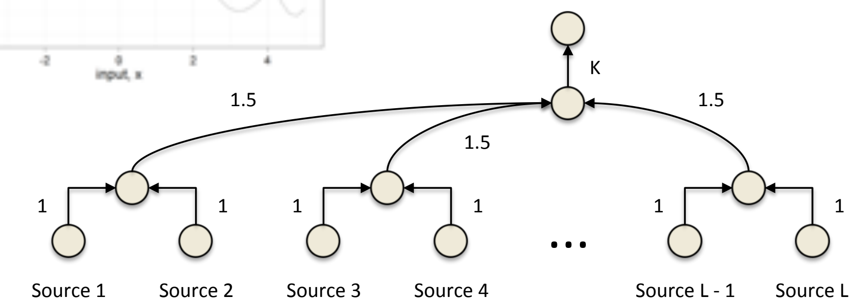
POLYMATROIDS

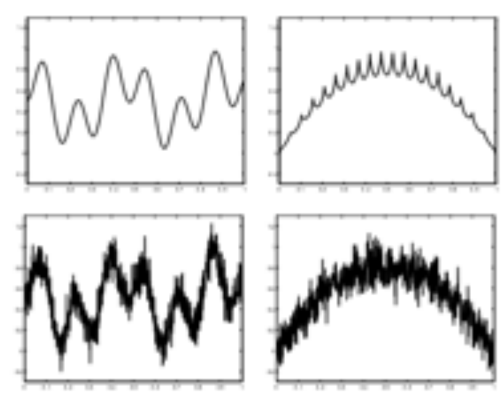
BLACK-BOX FUNCTIONS

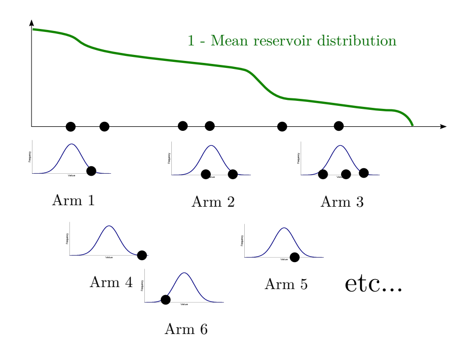
STRUCTURES WITHOUT TOPOLOGY

. . .









SPECIFIC GRAPH BANDIT SETTINGS

smoothness
spectral bandits

$R_T = \widetilde{\mathcal{O}}\left(d\sqrt{T \ln T}\right)$

#relevant eigenvectors

side observations
on graphs

$R_T = \widetilde{\mathcal{O}}\left(\sqrt{\alpha T \ln N}\right)$

independence number

influence maximisation
revealing bandits

$R_T = \widetilde{\mathcal{O}}\left(\sqrt{r_* T D_*}\right)$

detectable dimension

noisy side observations
on graphs

$R_T = \widetilde{\mathcal{O}}\left(\sqrt{\alpha^\star T \ln N}\right)$

effective independence number

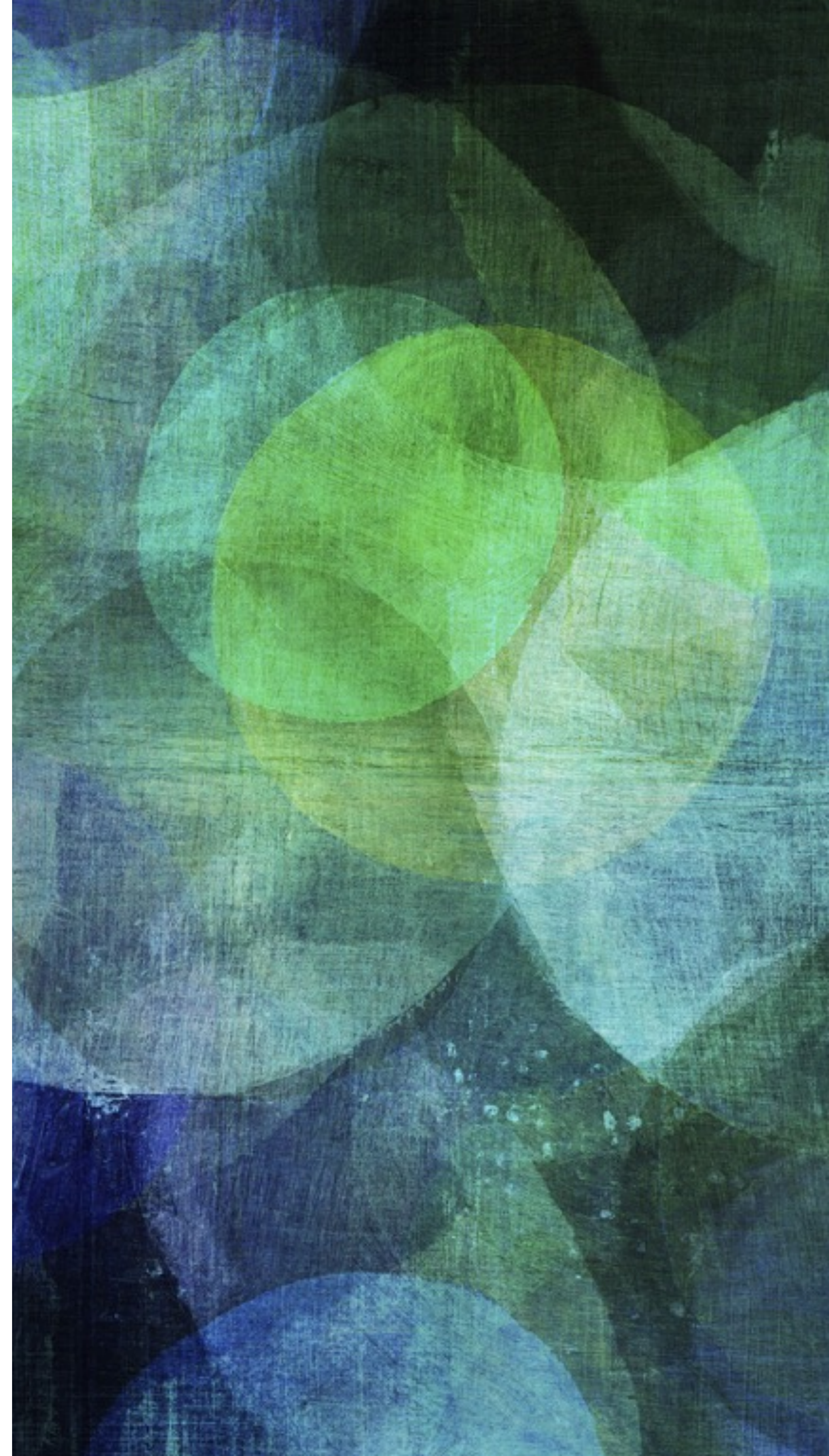
MV, Munos, Kveton, Kocák: **Spectral Bandits for Smooth Graph Functions**, ICML 2014

Kocák, MV, Munos, Agrawal: **Spectral Thompson Sampling**, AAAI 2014

Hanawal, Saligrama, MV, Munos: **Cheap Bandits**, ICML 2015

SPECTRAL BANDITS

exploiting smoothness of rewards on graphs



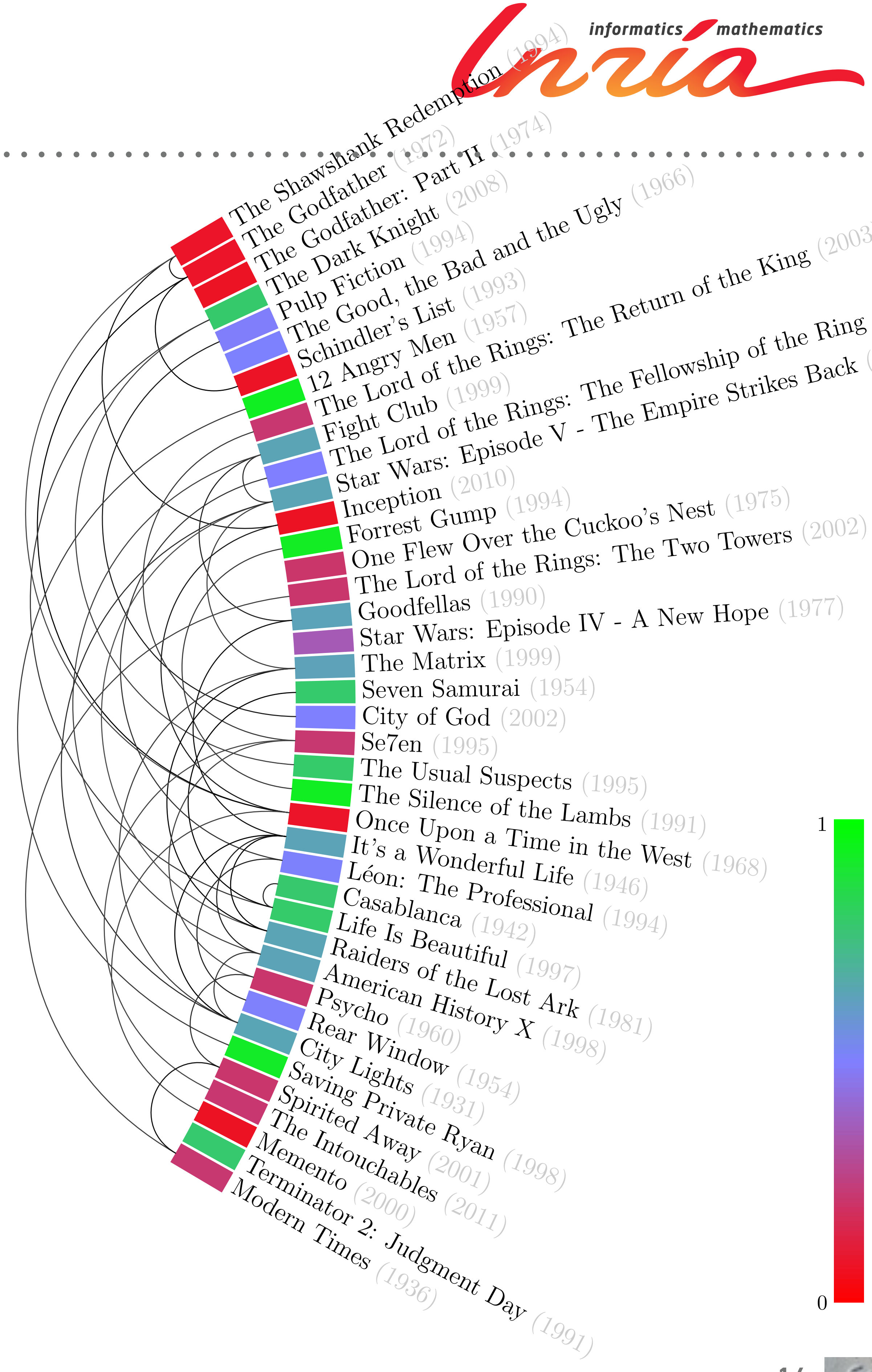
SPECTRAL BANDITS

Assumptions

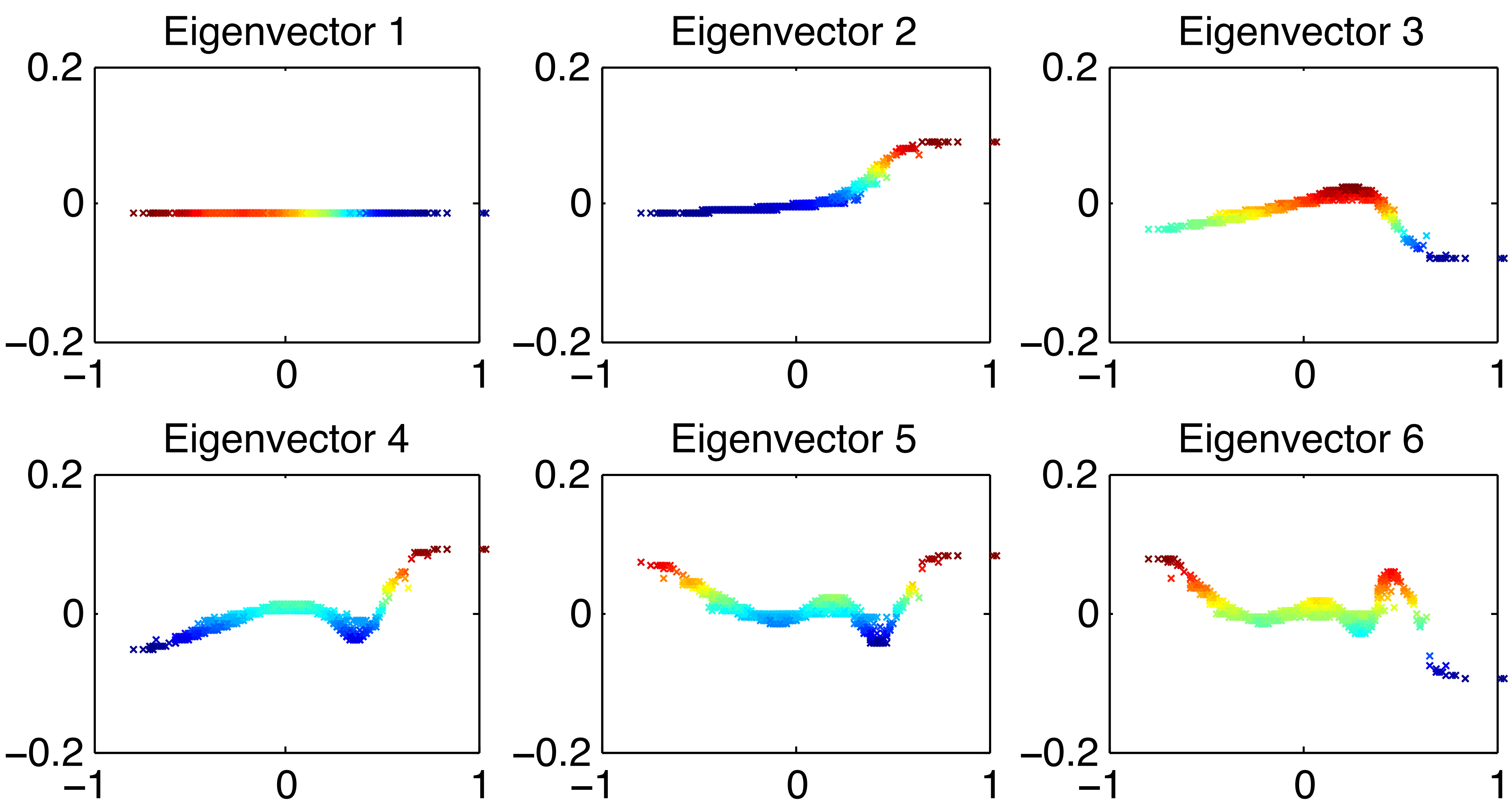
- Unknown reward function $f : V(G) \to \mathbb{R}$.
- Function f is **smooth** on a graph.
- Neighboring movies $\Rightarrow$ similar preferences.
- Similar preferences $\not\Rightarrow$ neighboring movies.

Desiderata

An algorithm useful in the case $T \ll N$!



FLIXSTER DATA



Eigenvectors from the Flixster data corresponding to the smallest few eigenvalues of the graph Laplacian projected onto the first principal component of data. Colors indicate the values.

SPECTRAL BANDIT: LEARNING SETTING

Learning setting for a bandit algorithm π

- In each time t step choose a node $\pi(t)$.
- the $\pi(t)$-th row $\mathbf{x}_{\pi(t)}$ of the matrix $\mathbf{Q}$ corresponds to the arm $\pi(t)$.
- Obtain noisy reward $r_t = \mathbf{x}_{\pi(t)}^\mathsf{T}\boldsymbol{\alpha}^* + \varepsilon_t$. Note: $\mathbf{x}_{\pi(t)}^\mathsf{T}\boldsymbol{\alpha}^* = f_{\pi(t)}$
  - ε_t is R-sub-Gaussian noise. $\forall\xi \in \mathbb{R}, \mathbb{E}[e^{\xi\varepsilon_t}] \leq \exp\left(\xi^2 R^2/2\right)$
- Minimize cumulative regret

$$R_T = T \max_a \left(\mathbf{x}_a^\mathsf{T}\boldsymbol{\alpha}^*\right) - \sum_{t=1}^{T} \mathbf{x}_{\pi(t)}^\mathsf{T}\boldsymbol{\alpha}^*.$$

Can we just use linear bandits?

LINEAR VS. SPECTRAL BANDITS

- **Linear bandit algorithms**
  - **LinUCB** (Li et al., 2010)
    - Regret bound $\approx D\sqrt{T \ln T}$
  - **LinearTS** (Agrawal and Goyal, 2013)
    - Regret bound $\approx D\sqrt{T \ln N}$

Note: D is ambient dimension, in our case N, length of x_i.
Number of actions, e.g., all possible movies $\rightarrow$ **HUGE!**

- **Spectral bandit algorithms**
  - **SpectralUCB** (Valko et al., ICML 2014)
    - Regret bound $\approx d\sqrt{T \ln T}$
    - Operations per step: $D^2 N$
  - **SpectralTS** (Kocák et al., AAAI 2014)
    - Regret bound $\approx d\sqrt{T \ln N}$
    - Operations per step: $D^2 + DN$

Note: d is effective dimension, usually much smaller than D.

SPECTRAL BANDITS – EFFECTIVE DIMENSION

- **Effective dimension:** Largest d such that

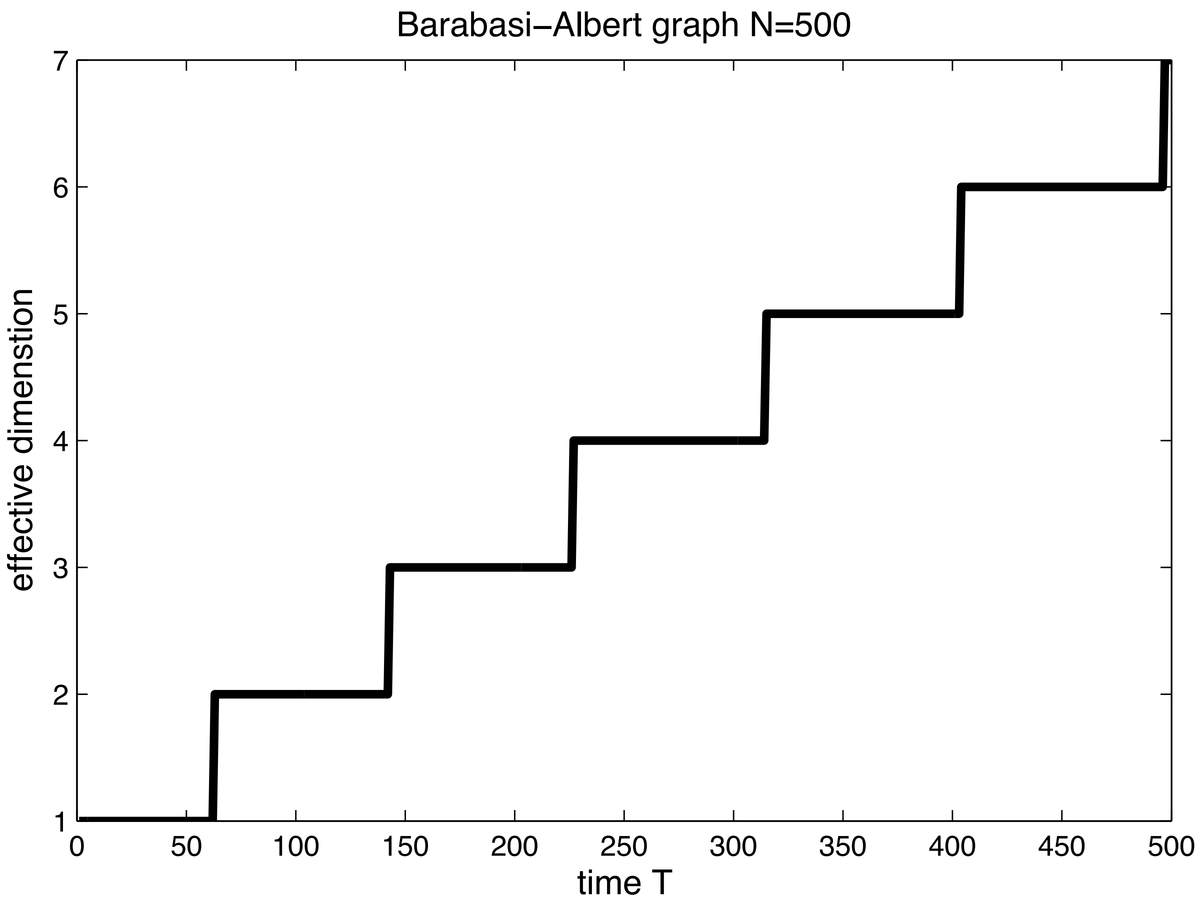
$$(d-1)\lambda_d \leq \frac{T}{\log(1+T/\lambda)}.$$

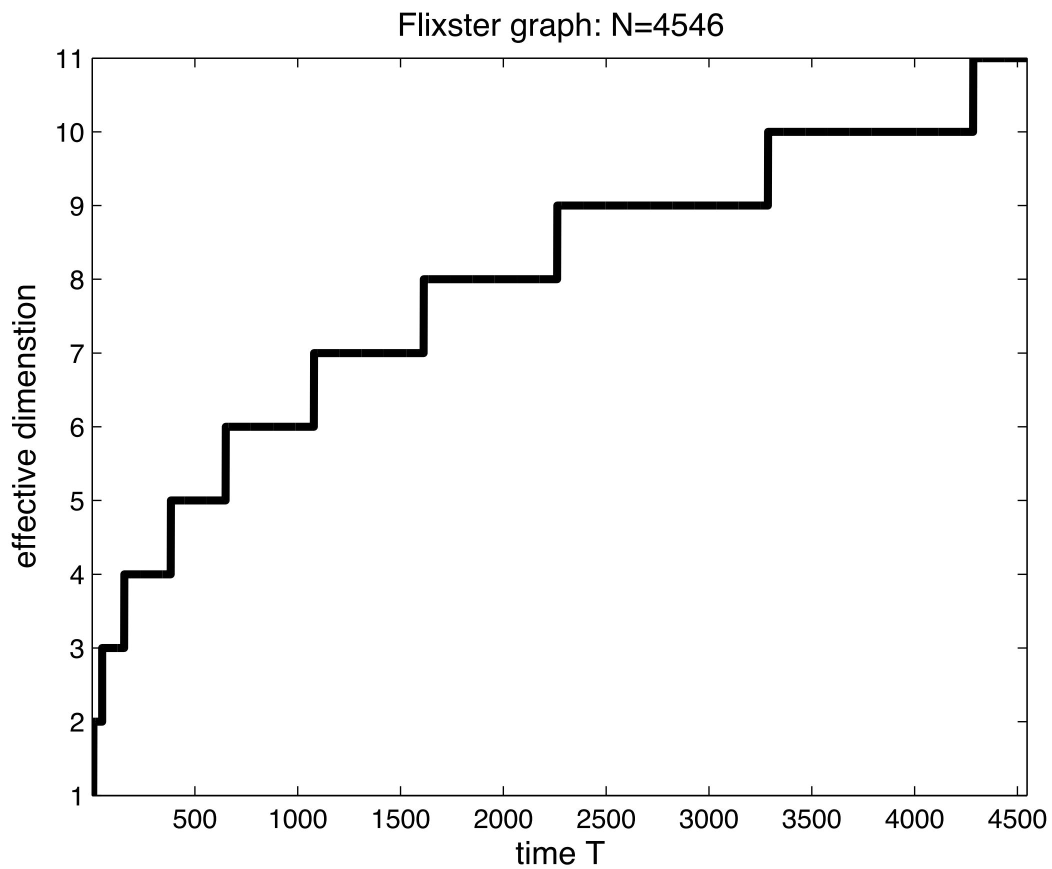
- Function of time horizon and graph properties
- λ_i : i-th smallest eigenvalue of $\mathbf{\Lambda}$.
- λ: Regularization parameter of the algorithm.

Properties:

- d is small when the coefficients λ_i grow rapidly above time.
- d is related to the number of “non-negligible” dimensions.
- Usually d is much smaller than D in real world graphs.
- Can be computed beforehand.

SPECTRAL BANDITS - EFFECTIVE DIMENSION

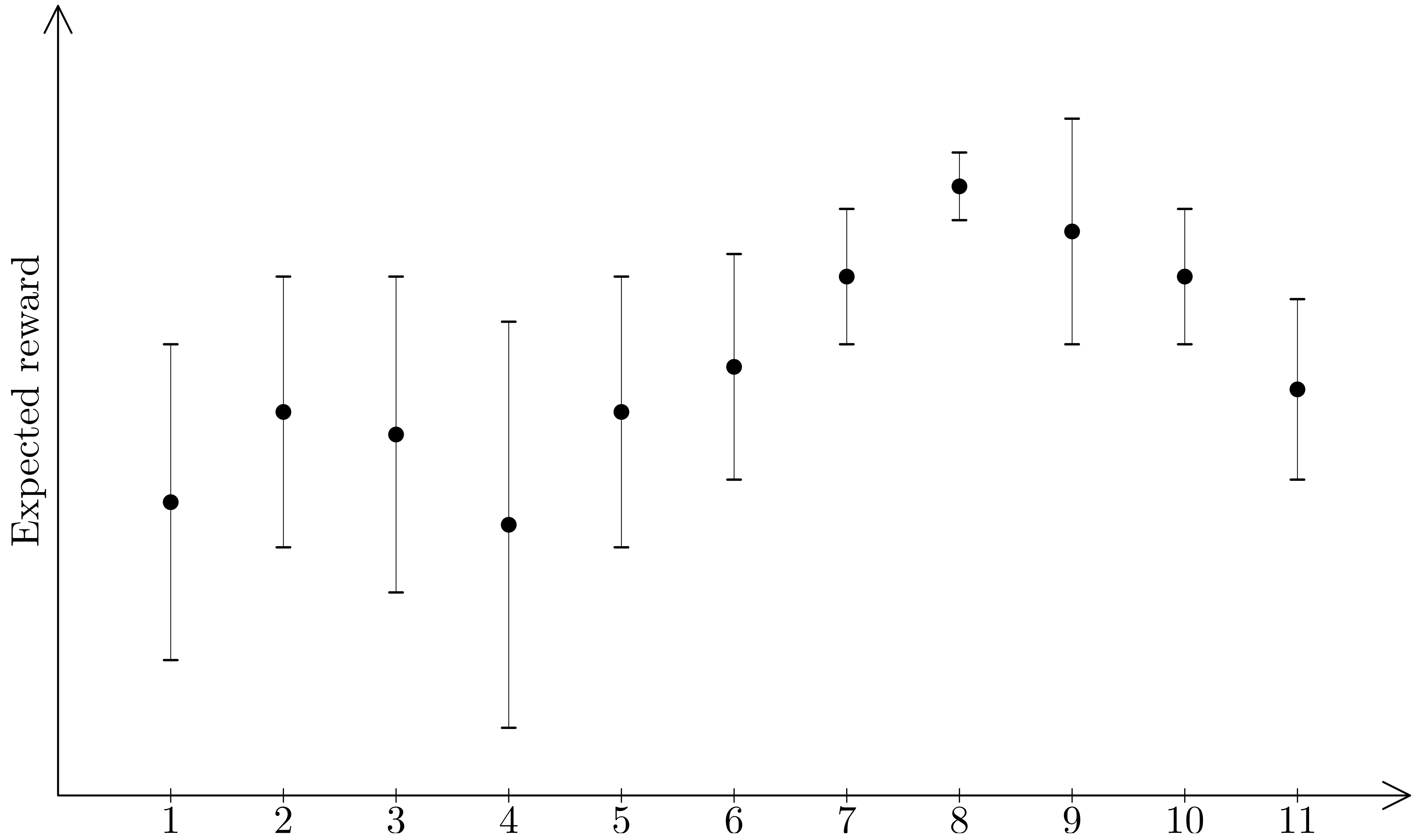




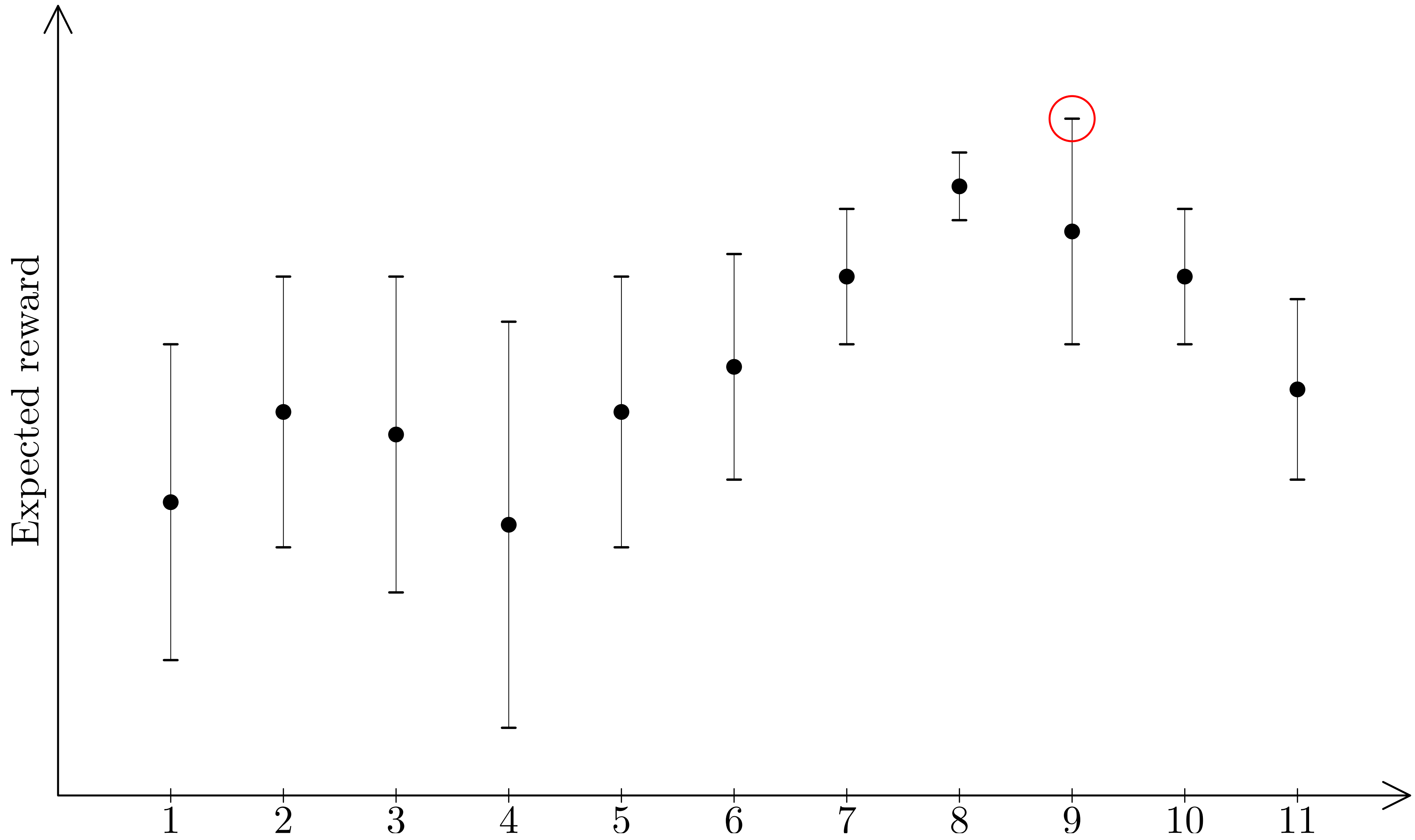
$$d \ll D$$

Note: In our setting $T < N = D$.

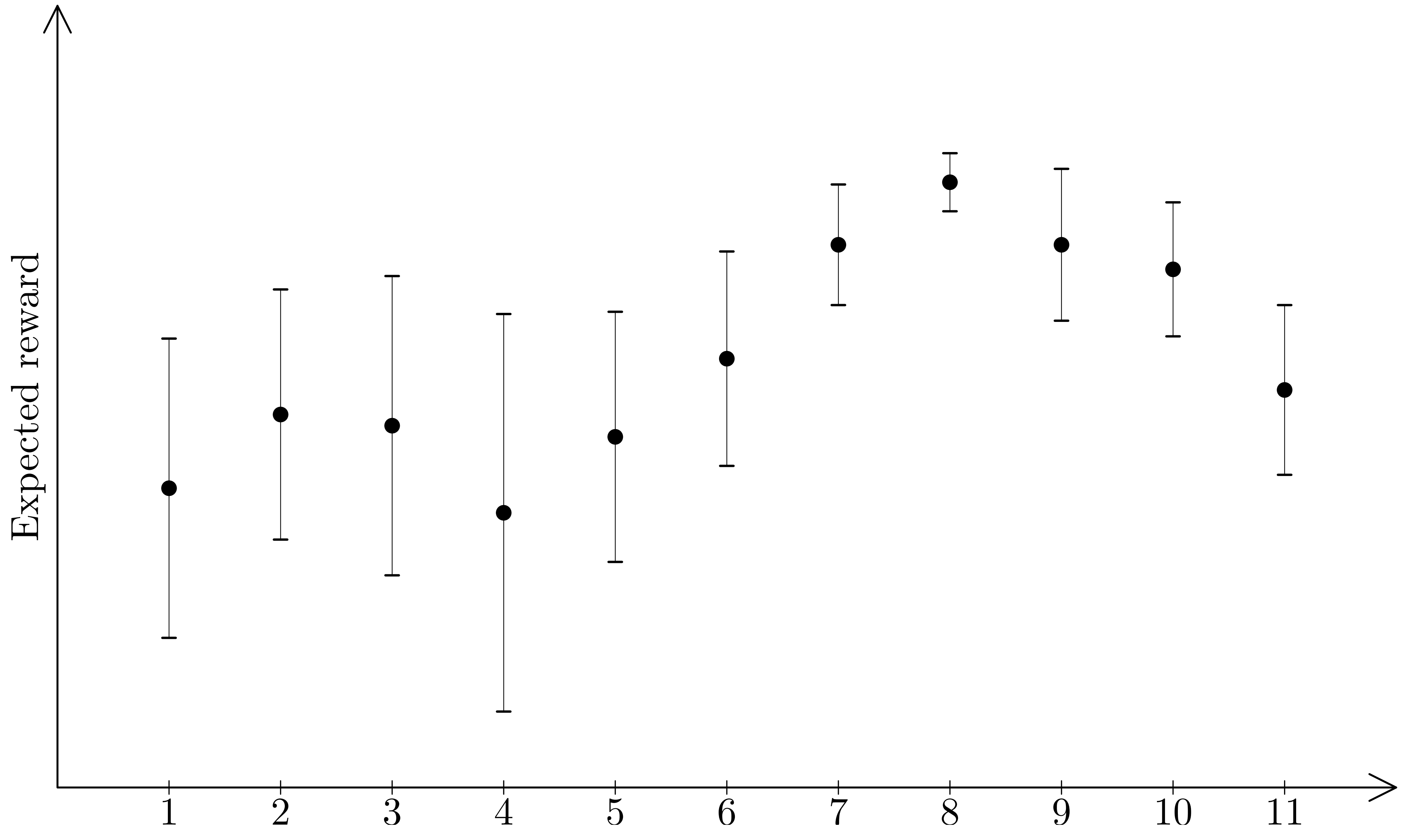
UPPER CONFIDENCE BOUND BASED ALGOS



UPPER CONFIDENCE BOUND BASED ALGOS



UPPER CONFIDENCE BOUND BASED ALGOS



SPECTRAL UCB

Given a vector of weights $\boldsymbol{\alpha}$, we define its $\boldsymbol{\Lambda}$ norm as

$$\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} = \sqrt{\sum_{k=1}^{N} \lambda_k \alpha_k^2} = \sqrt{\boldsymbol{\alpha}^\mathsf{T} \boldsymbol{\Lambda} \boldsymbol{\alpha}},$$

and fit the ratings r_v with a (regularized) least-squares estimate

$$\widehat{\boldsymbol{\alpha}}_t = \arg\min_{\boldsymbol{\alpha}} \left(\sum_{v=1}^{t} [\langle \mathbf{x}_v, \boldsymbol{\alpha} \rangle - r_v]^2 + \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}^2 \right).$$

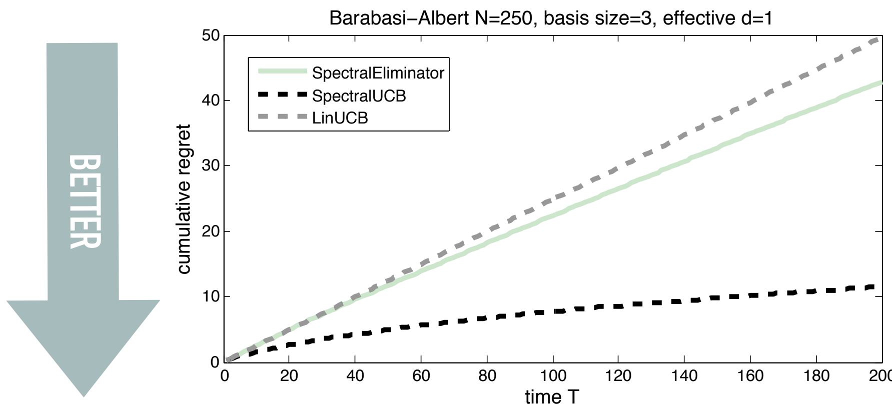
$\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}$ is a penalty for non-smooth combinations of eigenvectors.

SPECTRAL UCB

Input:
$N,\ T,\ \{\mathbf{\Lambda_L}, \mathbf{Q}\},\ \lambda,\ \delta,\ R,\ C$
Run:
$\mathbf{\Lambda} \leftarrow \mathbf{\Lambda_L} + \lambda \mathbf{I}$
$d \leftarrow \max\{d : (d-1)\lambda_d \leq T / \ln(1 + T/\lambda)\}$
for $t = 1$ **to** T **do**
Update the basis coefficients $\widehat{\boldsymbol{\alpha}}$:
$\mathbf{X}_t \leftarrow [\mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(t-1)}]^\mathsf{T}$
$\mathbf{r} \leftarrow [r_1, \ldots, r_{t-1}]^\mathsf{T}$
$\mathbf{V}_t \leftarrow \mathbf{X}_t \mathbf{X}_t^\mathsf{T} + \mathbf{\Lambda}$
$\widehat{\boldsymbol{\alpha}}_t \leftarrow \mathbf{V}_t^{-1} \mathbf{X}_t^\mathsf{T} \mathbf{r}$
$c_t \leftarrow 2R\sqrt{d \ln(1 + t/\lambda) + 2\ln(1/\delta)} + C$
$\pi(t) \leftarrow \arg\max_a \left(\mathbf{x}_a^\mathsf{T} \widehat{\boldsymbol{\alpha}} + c_t \|\mathbf{x}_a\|_{\mathbf{V}_t^{-1}} \right)$
Observe the reward r_t
end for

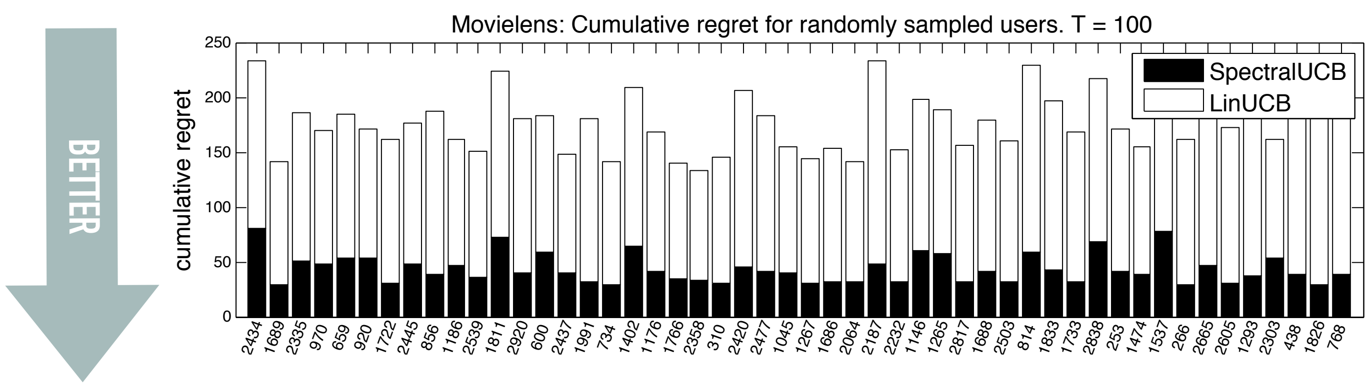
SPECTRALUCB REGRET BOUND

- d: Effective dimension.
- λ: Minimal eigenvalue of $\mathbf{\Lambda} = \mathbf{\Lambda_L} + \lambda \mathbf{I}$.
- C: Smoothness upper bound, $\|\boldsymbol{\alpha}^*\|_{\mathbf{\Lambda}} \leq C$.
- $\mathbf{x}_i^{\mathsf{T}} \boldsymbol{\alpha}^* \in [-1, 1]$ for all i.

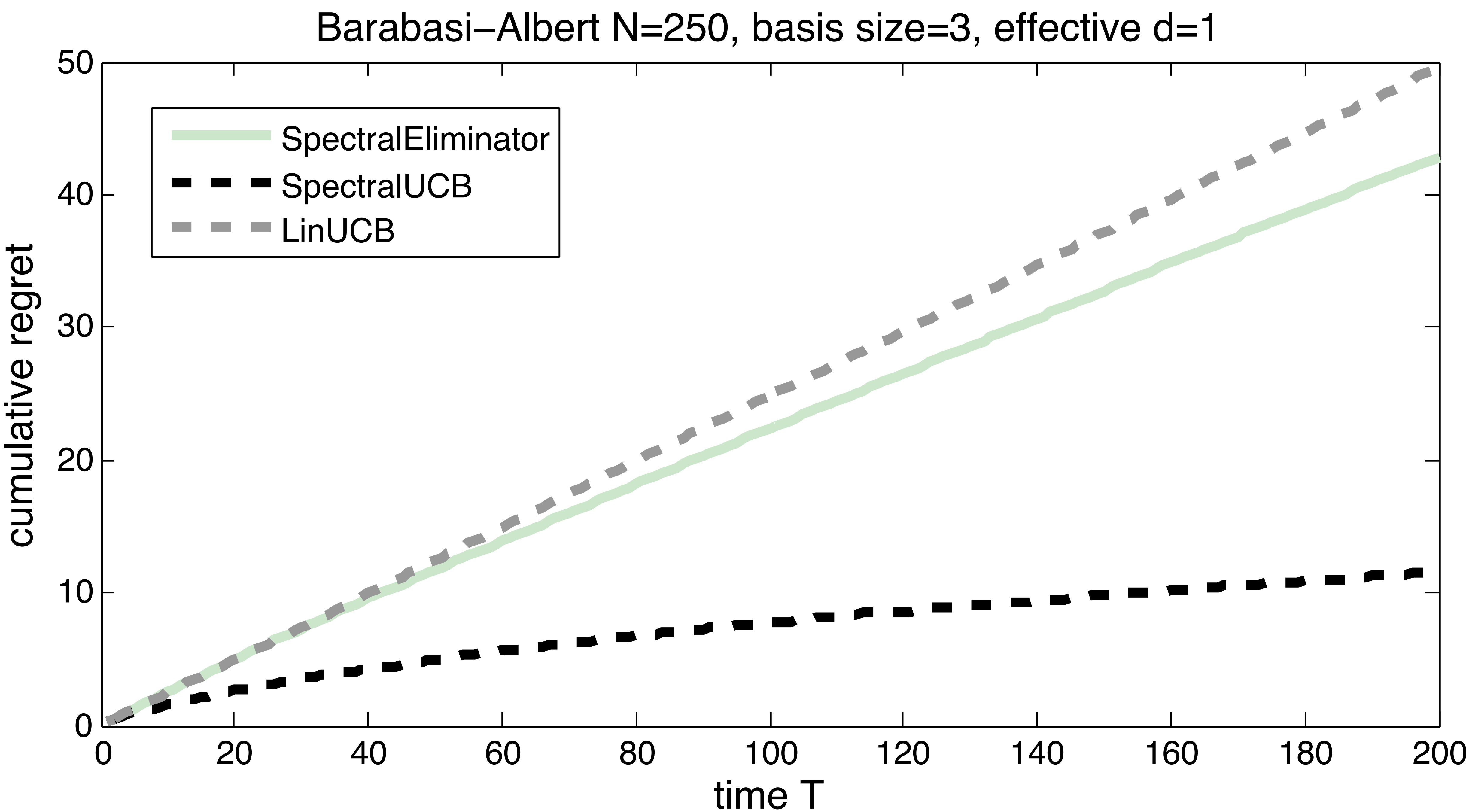


The **cumulative regret** R_T of **SpectralUCB** is with probability $1 - \delta$ bounded as

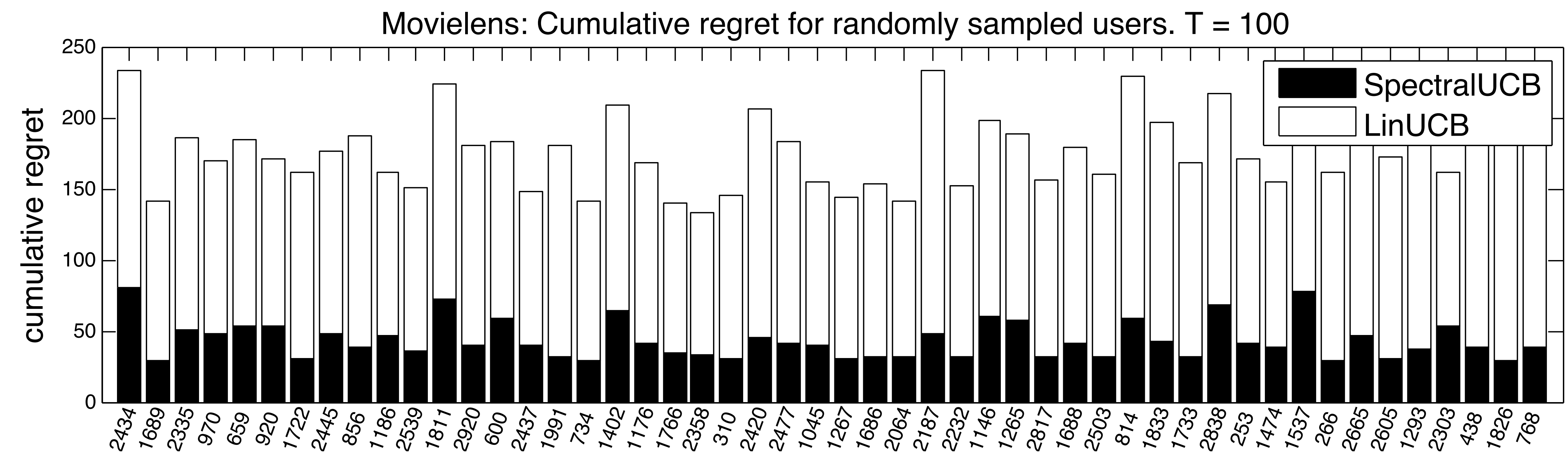
$$R_T \leq \left(8R\sqrt{d \ln \frac{\lambda + T}{\lambda} + 2 \ln \frac{1}{\delta}} + 4C + 4\right) \sqrt{dT \ln \frac{\lambda + T}{\lambda}}.$$

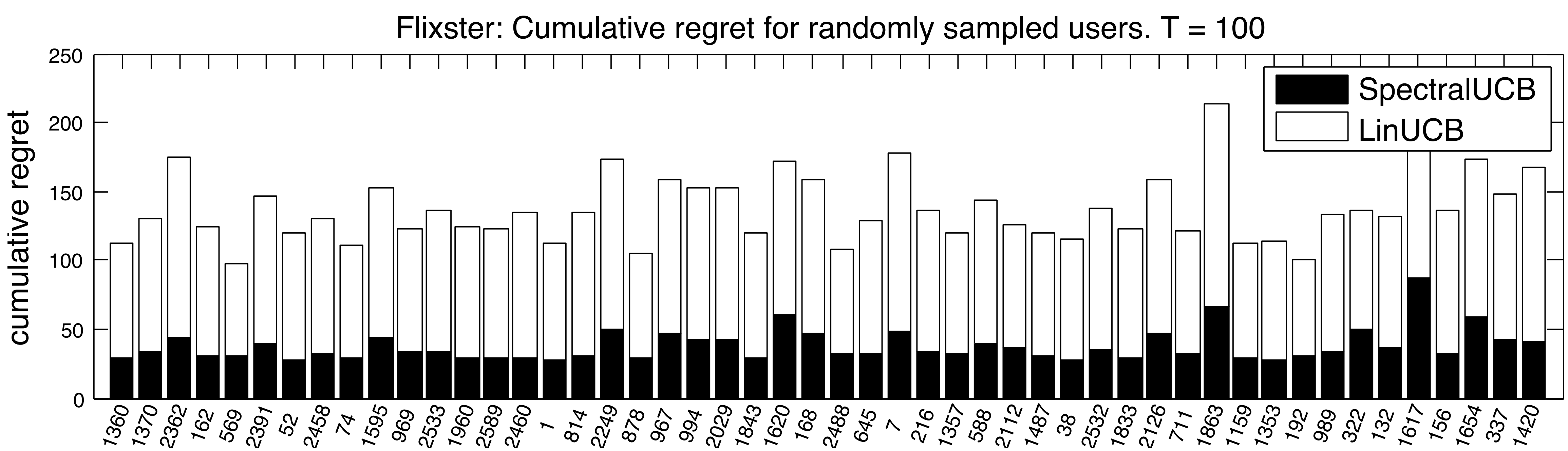


SPECTRAL UCN ON BA GRAPH



SPECTRAL UCN ON REAL DATA



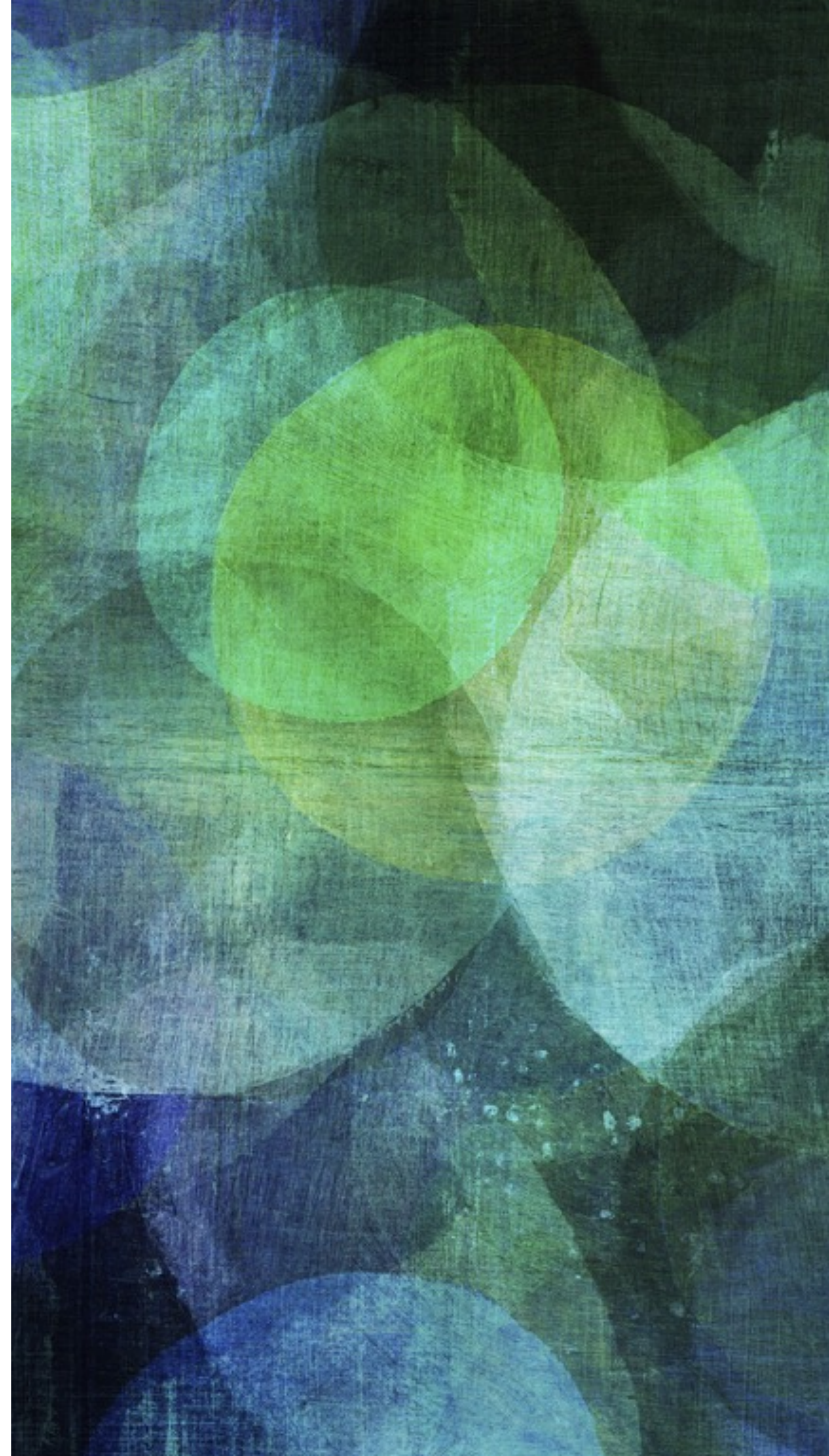


Kocák, Neu, MV, Munos: **Efficient learning by implicit exploration in bandit problems with side observations**, NIPS 2014

Kocák, Neu, MV: **Online learning with Erdos-Rényi side-observation graphs**
UAI 2016 (to appear)

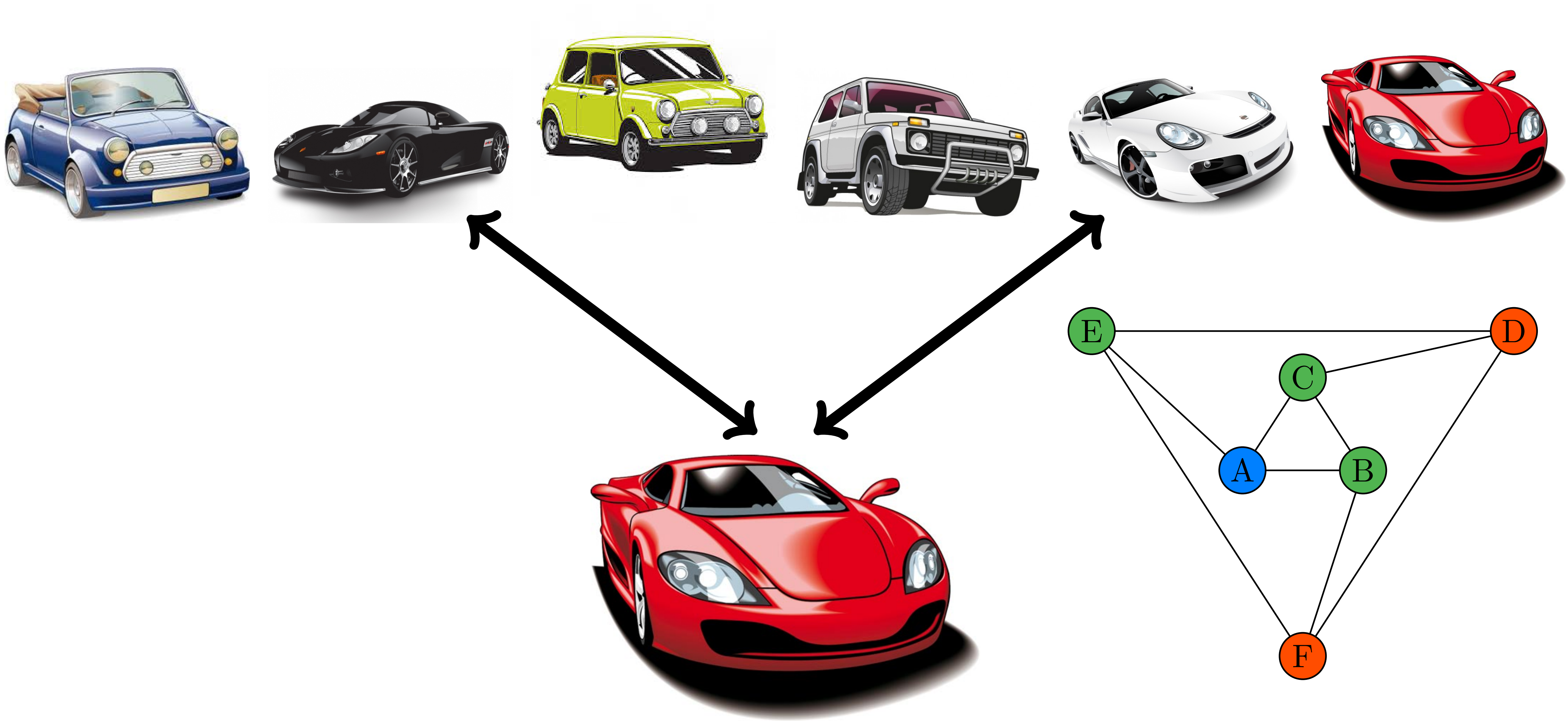
GRAPH BANDITS WITH SIDE OBSERVATIONS

exploiting free observations from neighbouring nodes

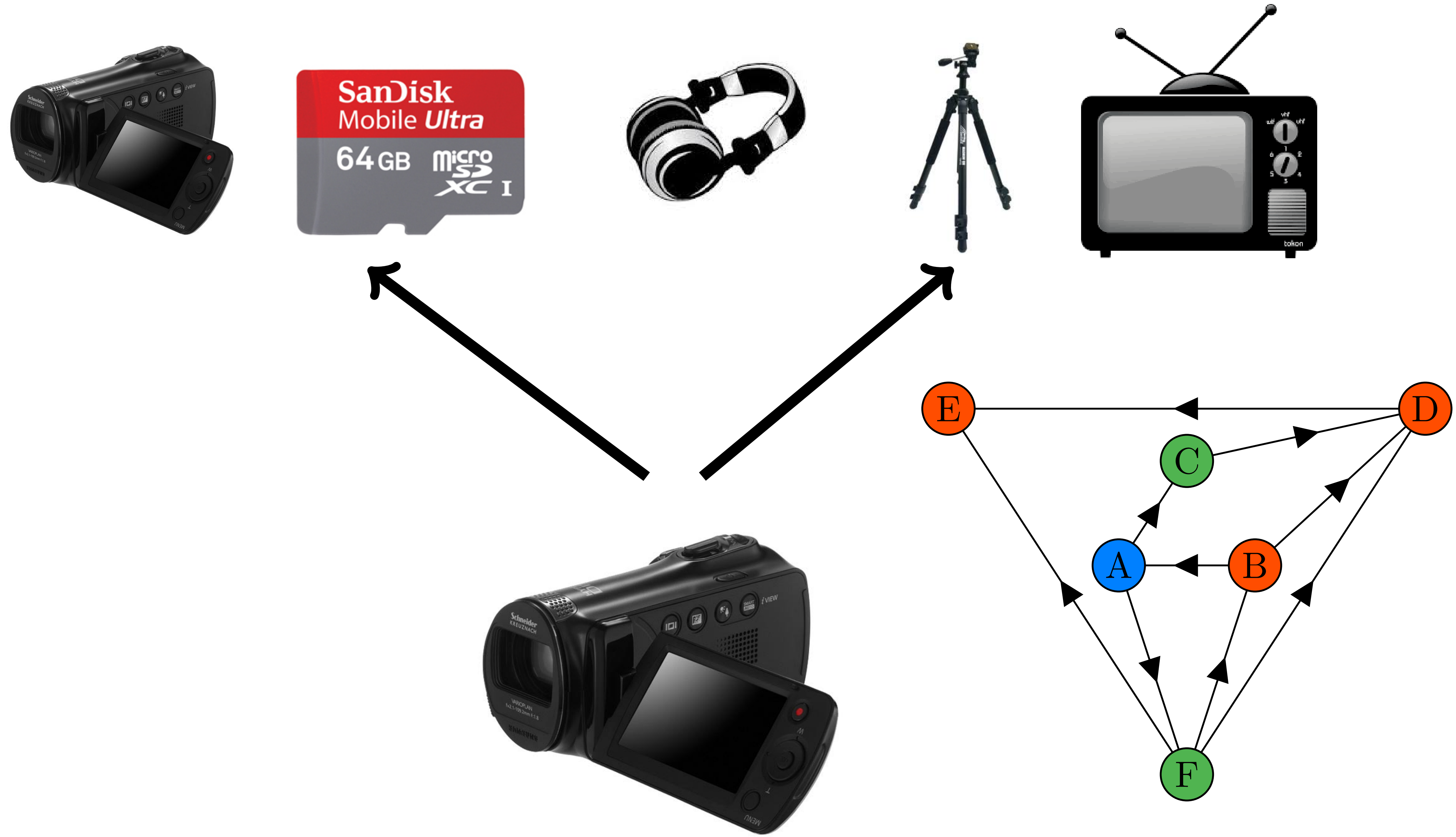


SIDE OBSERVATIONS: UNDIRECTED

Example 1: undirected observations



Example 2: Directed observation



SIDE OBSERVATIONS – AN INTERMEDIATE GAME

Full-information

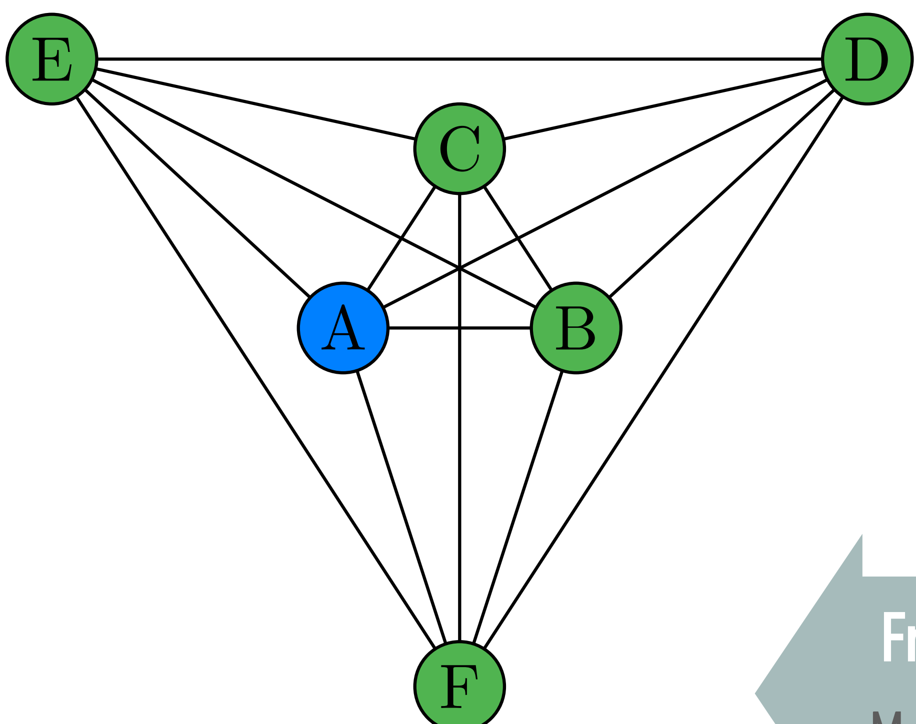
- observe losses of **all** actions
- example: Hedge

$$R_T = \widetilde{\mathcal{O}}(\sqrt{T})$$

Bandits

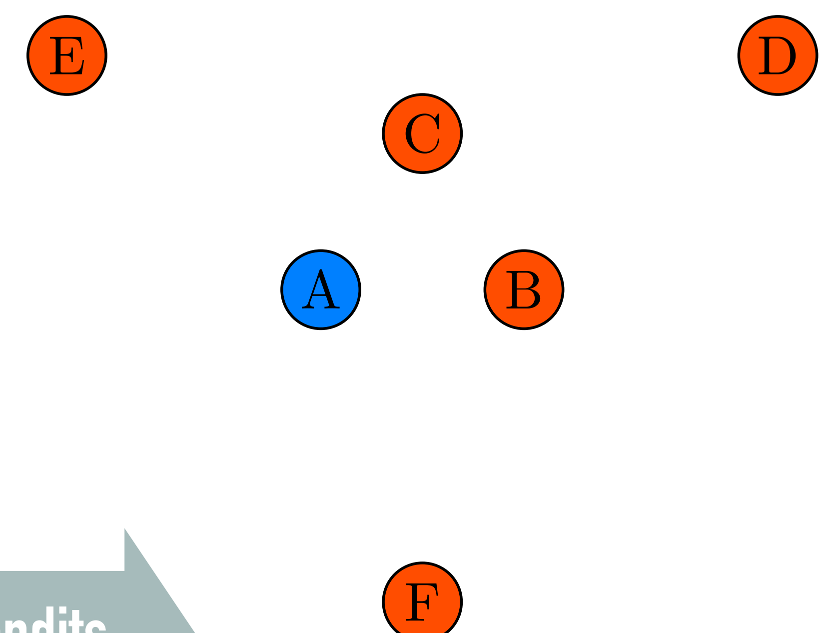
- observe losses of **the chosen** action
- example: EXP3

$$R_T = \widetilde{\mathcal{O}}(\sqrt{NT})$$



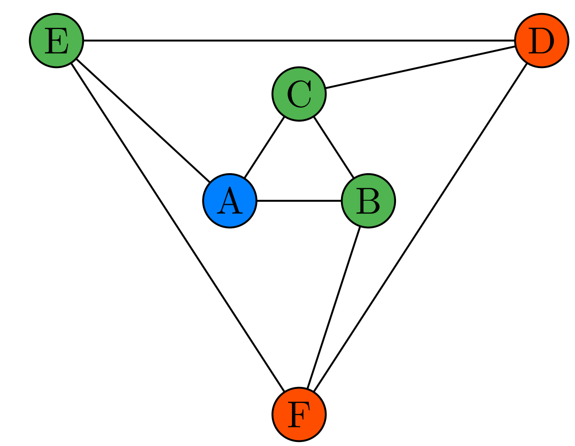
From Experts to Bandits

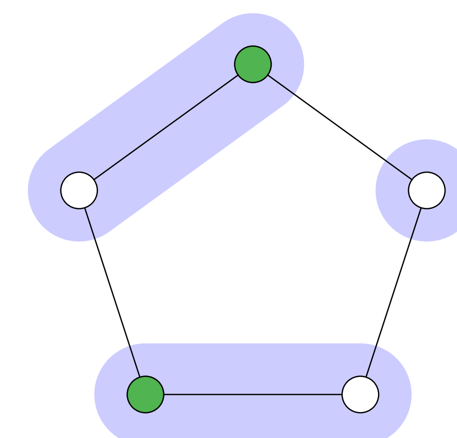
Mannor and Shamir 2011

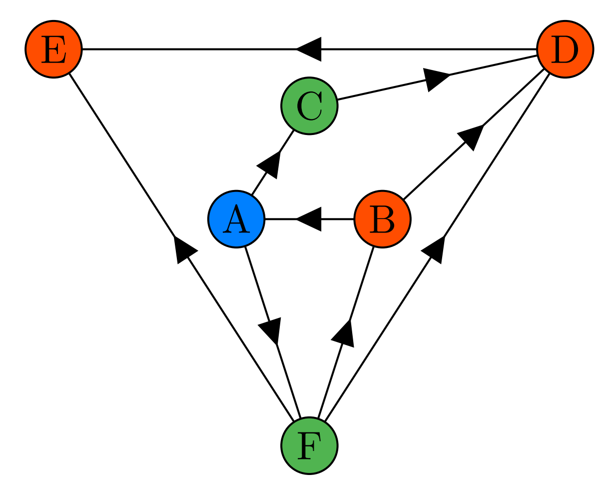


KNOWLEDGE OF OBSERVATION GRAPHS

- ELP (Mannor and Shamir 2011)
  - **EXP3** - with **"LP balanced exploration"**
  - undirected $O(\sqrt{(\alpha T)})$ ✅ – needs to know G_t
  - directed case $O(\sqrt{(cT)})$ – needs to know G_t
- EXP3-SET (Alon, Cesa-Bianchi, Gentile, Mansour, 2013)
  - undirected $O(\sqrt{(\alpha T)})$ ✅ does not need to know G_t ✅
- EXP3-DOM (Alon, Cesa-Bianchi, Gentile, Mansour, 2013)
  - directed $O(\sqrt{(\alpha T)})$ ✅ – need to know G_t
  - **calculates dominating set**







EXP3-IX: IMPLICIT EXPLORATION

Algorithm 1 EXP3-IX

Input: Set of actions $\mathcal{S} = [d]$,
  parameters $\gamma_t \in (0,1)$, $\eta_t > 0$ for $t \in [T]$.
for $t = 1$ **to** T **do**
  $w_{t,i} \leftarrow (1/d)\exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for $i \in [d]$
  An adversary privately chooses losses $\ell_{t,i}$ for $i \in [d]$ and generates a graph G_t
  $W_t \leftarrow \sum_{i=1}^{d} w_{t,i}$
  $p_{t,i} \leftarrow w_{t,i}/W_t$
  Choose $I_t \sim \boldsymbol{p}_t = (p_{t,1},\ldots,p_{t,d})$
  Observe graph G_t
  Observe pairs $\{i, \ell_{t,i}\}$ for $(I_t \to i) \in G_t$
  $o_{t,i} \leftarrow \sum_{(j\to i)\in G_t} p_{t,j}$ for $i \in [d]$
  $\hat{\ell}_{t,i} \leftarrow \frac{\ell_{t,i}}{o_{t,i}+\gamma_t}\mathbb{1}_{\{(I_t\to i)\in G_t\}}$ for $i \in [d]$
end for

Benefits of the **implicit exploration**

- **no need to know the graph before**
- no need to estimate dominating set
- no need for doubling trick
- no need for aggregation

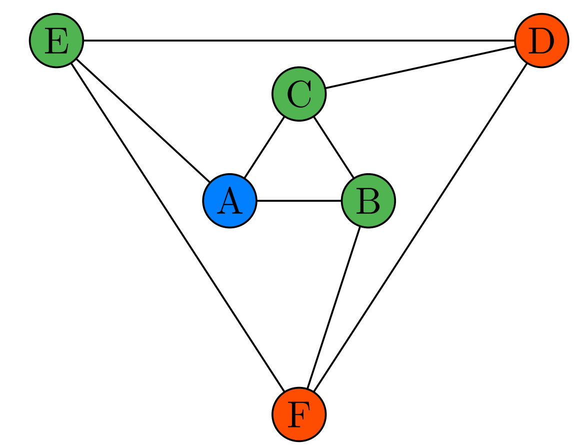
$$R_T = \widetilde{\mathcal{O}}\left(\sqrt{\alpha T \ln N}\right)$$

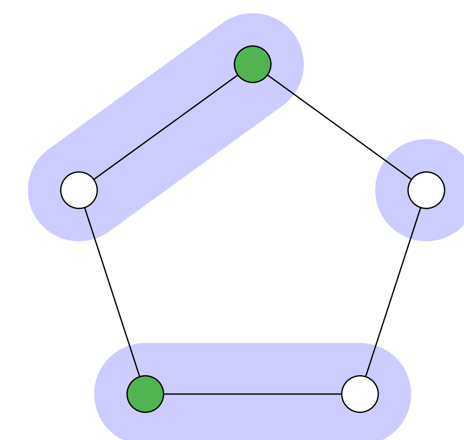
Optimistic bias for the loss estimates

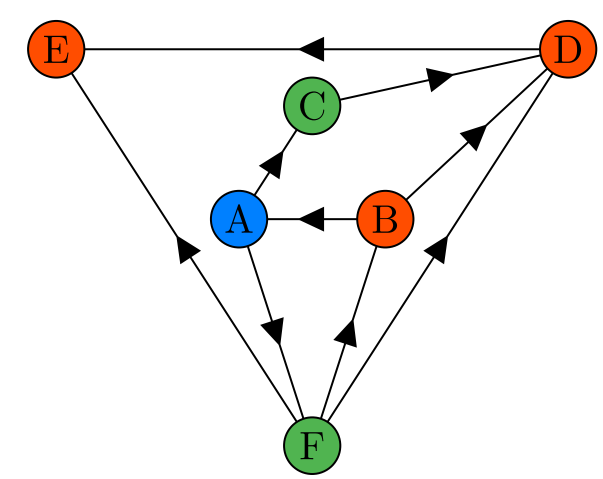
$$\mathbb{E}[\hat{\ell}_{t,i}] = \frac{\ell_{t,i}}{o_{t,i}+\gamma} o_{t,i} + 0(1-o_{t,i}) = \ell_{t,i} - \ell_{t,i}\frac{\gamma}{o_{t,i}+\gamma} \leq \ell_{t,i}$$

FOLLOW UPS

- EXP3-IX (Kocák, Neu, MV, Munos, 2014)
  - directed $O(\sqrt{\alpha T})$ ✅ does not need to know G_t ✅
- EXP3.G (Alon, Cesa-Bianchi, Dekel, Koren, 2015)
  - directed $O(\sqrt{\alpha T})$ ✅ does not need to know G_t ✅
  - mixes uniform distribution
  - more general algorithm for settings **beyond bandits**
  - high-probability bound
- Neu 2015: high-probability bound for EXP3-IX

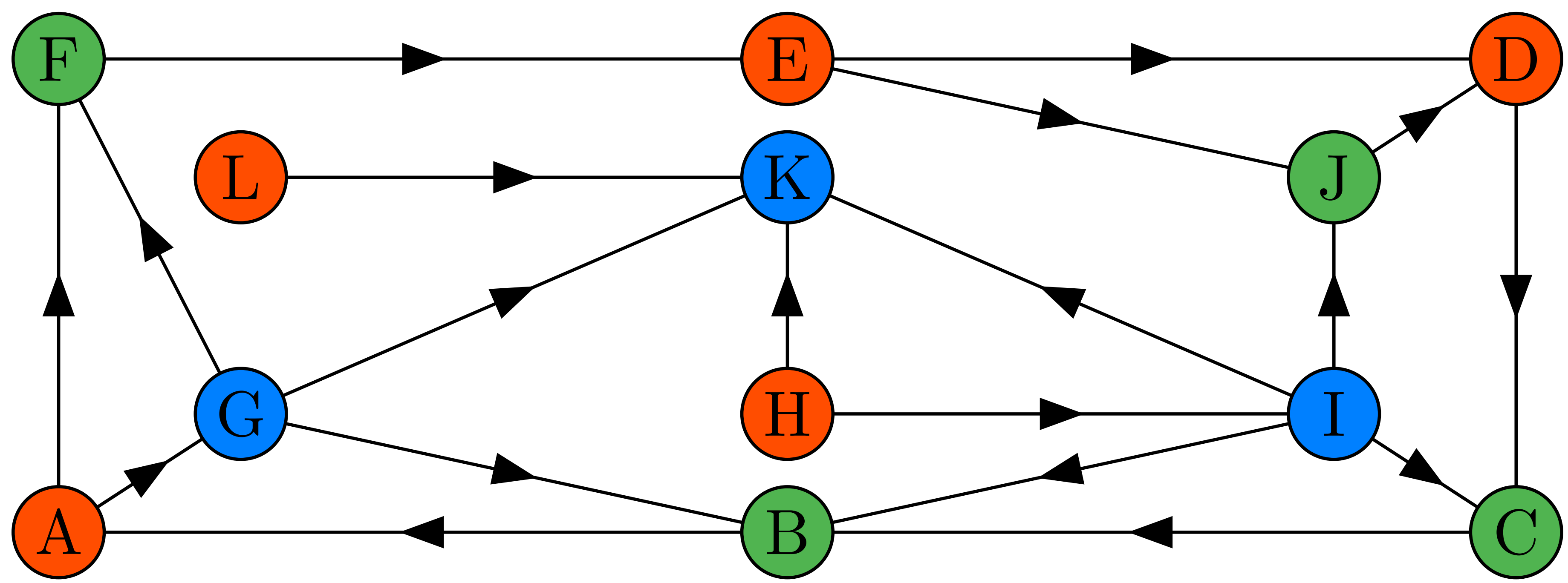






COMPLEX GRAPH ACTIONS

Example: online shortest path semi-bandits with observing traffic on the side streets



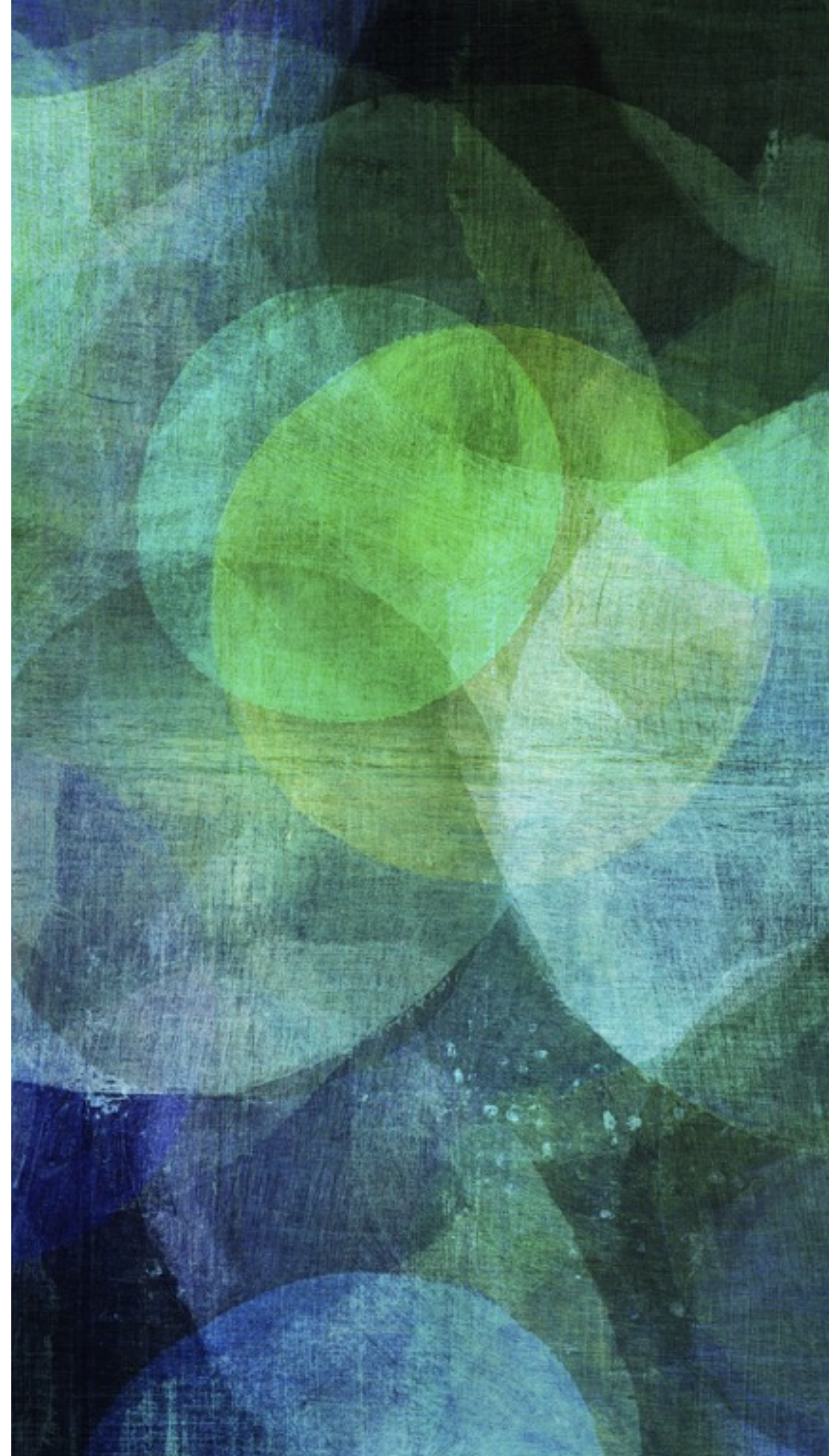
- Play action $\mathbf{V}_t \in S \subset \{0,1\}^N$, $\|\mathbf{v}\|_1 \leq m$ from all $\mathbf{v} \in S$
- Obtain losses $\mathbf{V}_t^\mathsf{T} \boldsymbol{\ell}_t$
- Observe additional losses according to the graph

$$R_T = \widetilde{\mathcal{O}}\left(m^{3/2} \sqrt{\sum_{t=1}^{T} \alpha_t} \right) = \widetilde{\mathcal{O}}\left(m^{3/2} \sqrt{\overline{\alpha} T} \right)$$

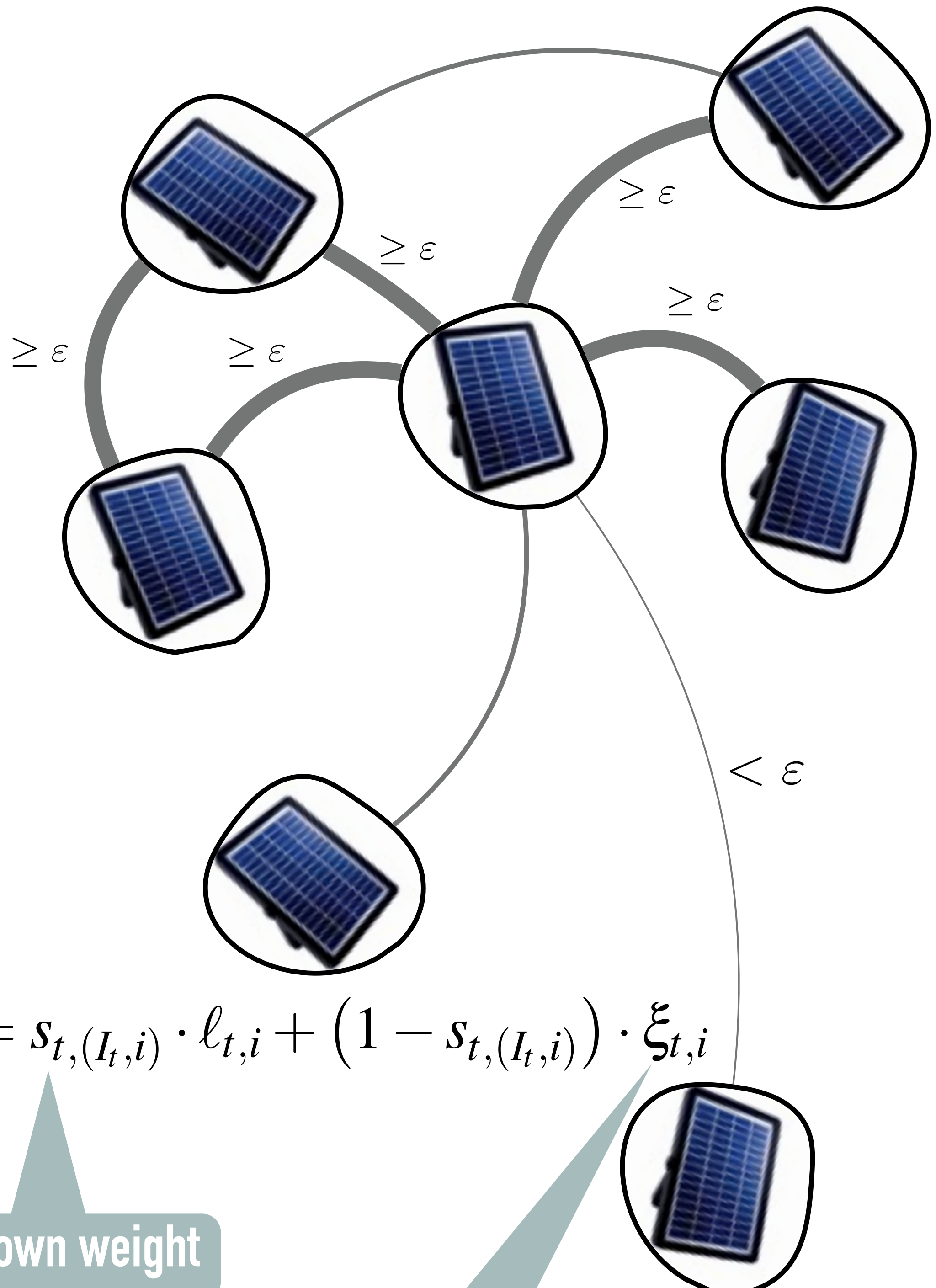
Kocák, Neu, MV: **Online learning with noisy side observations**, AISTATS 2016

GRAPH BANDITS WITH NOISY SIDE OBSERVATIONS

exploiting side observations that can be perturbed by certain level of noise



NOISY SIDE OBSERVATIONS



$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

known weight

zero-mean noise

Want: only reliable information!

1) If we know the perfect cutoff ε

- reliable: use as exact
- unreliable: rubbish

then we can improve over pure bandit setting!

2) Treating noisy observation induces bias

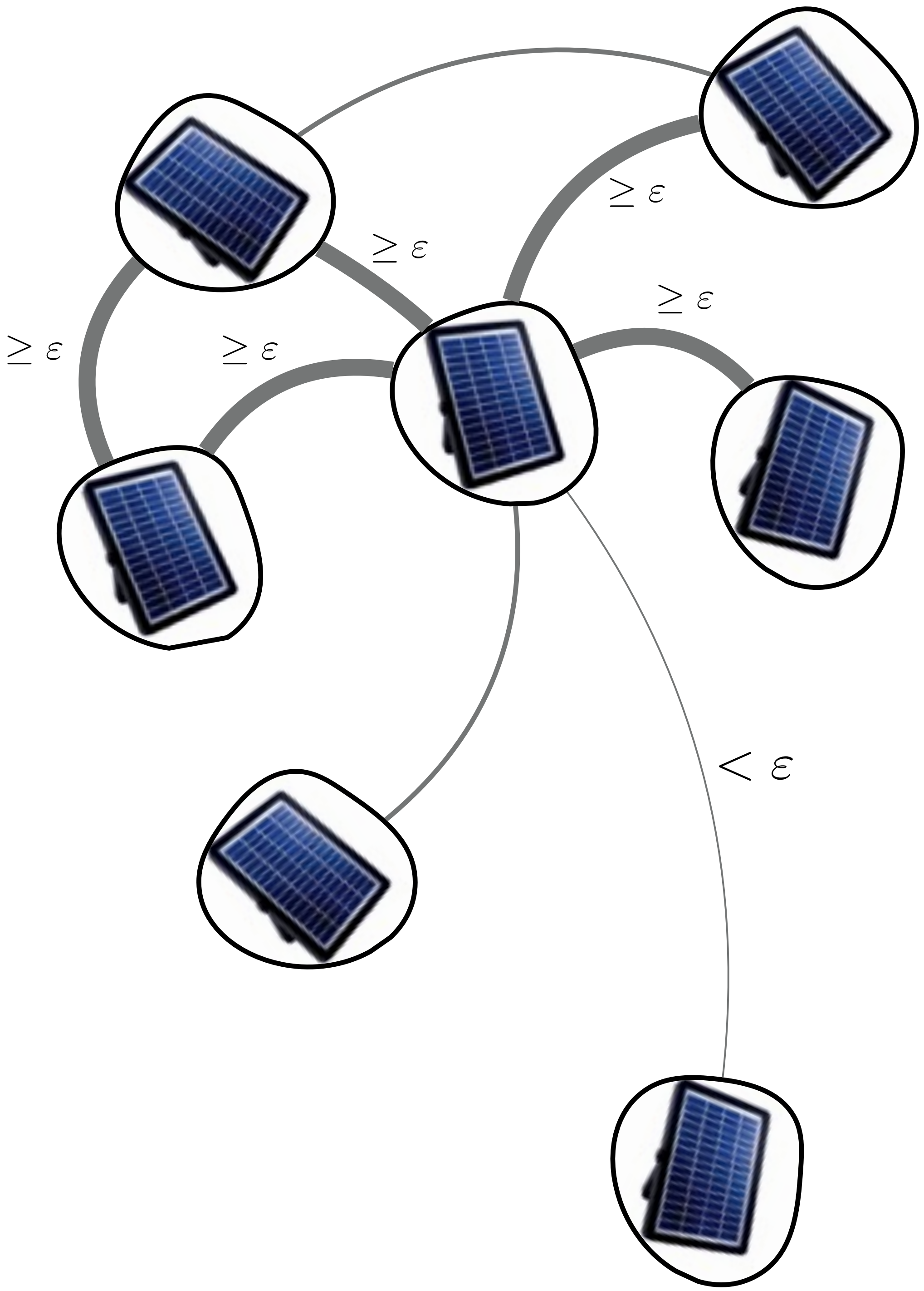
What can we hope for?

$$\tilde{\mathcal{O}}\left(\sqrt{1T}\right) \leq \qquad\qquad \leq \tilde{\mathcal{O}}\left(\sqrt{NT}\right)$$

effective independence number

Can we learn without knowing either ε or α^*?

NOISY SIDE OBSERVATIONS



Threshold estimate $R_T = \widetilde{\mathcal{O}}\left(\sqrt{\alpha^\star T}\right)$

$$\widehat{\ell}^{(\mathrm{T})}_{t,i} = \frac{c_{t,i}\mathbb{I}_{\{s_{t,(I_t,i)} \geq \varepsilon_t\}}}{\sum_{j=1}^{N} p_{t,j} s_{t,(j,i)} \mathbb{I}_{\{s_{t,(j,i)} \geq \varepsilon_t\}} + \gamma_t}$$

WIX estimate $R_T = \widetilde{\mathcal{O}}\left(\sqrt{\alpha^\star T}\right)$

$$\widehat{\ell}_{t,i} = \frac{s_{t,(I_t,i)} \cdot c_{t,i}}{\sum_{j=1}^{N} p_{t,j} s^2_{t,(j,i)} + \gamma_t}$$

Since $\alpha^* \leq \alpha(1)/1 \leq N$

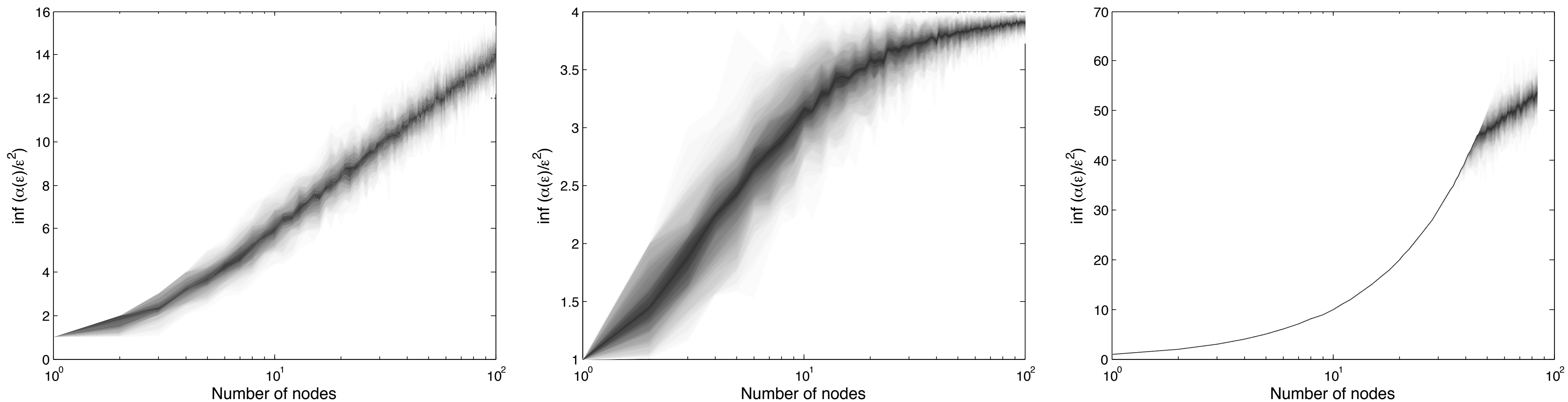
incorporating noisy observations does not hurt

$$\widetilde{\mathcal{O}}\left(\sqrt{\alpha^\star T}\right) \leq \widetilde{\mathcal{O}}\left(\sqrt{N T}\right)$$

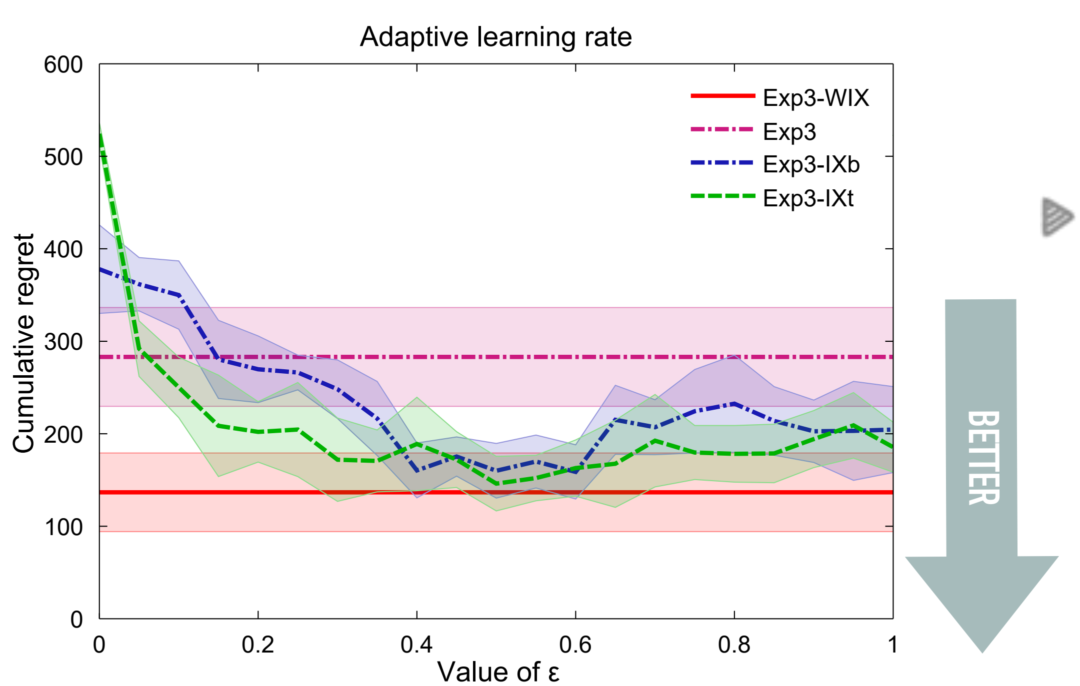
But how much does it help?

EMPIRICAL RESULTS

EMPIRICAL α^* FOR RANDOM GRAPHS WITH IID WEIGHTS



(a) $U(0,1)$ weights (b) $U(\frac{1}{2},1)$ weights (c) $U(0,\frac{1}{2})$ weights



- **special case:** if s_{ij} is either 0 or ε than $\alpha^* = \alpha/\varepsilon^2$
  - For this special case, there is a matches $\Theta(\sqrt{(\alpha T)}/\varepsilon)$ by Wu, György, Szepesvári, 2015.

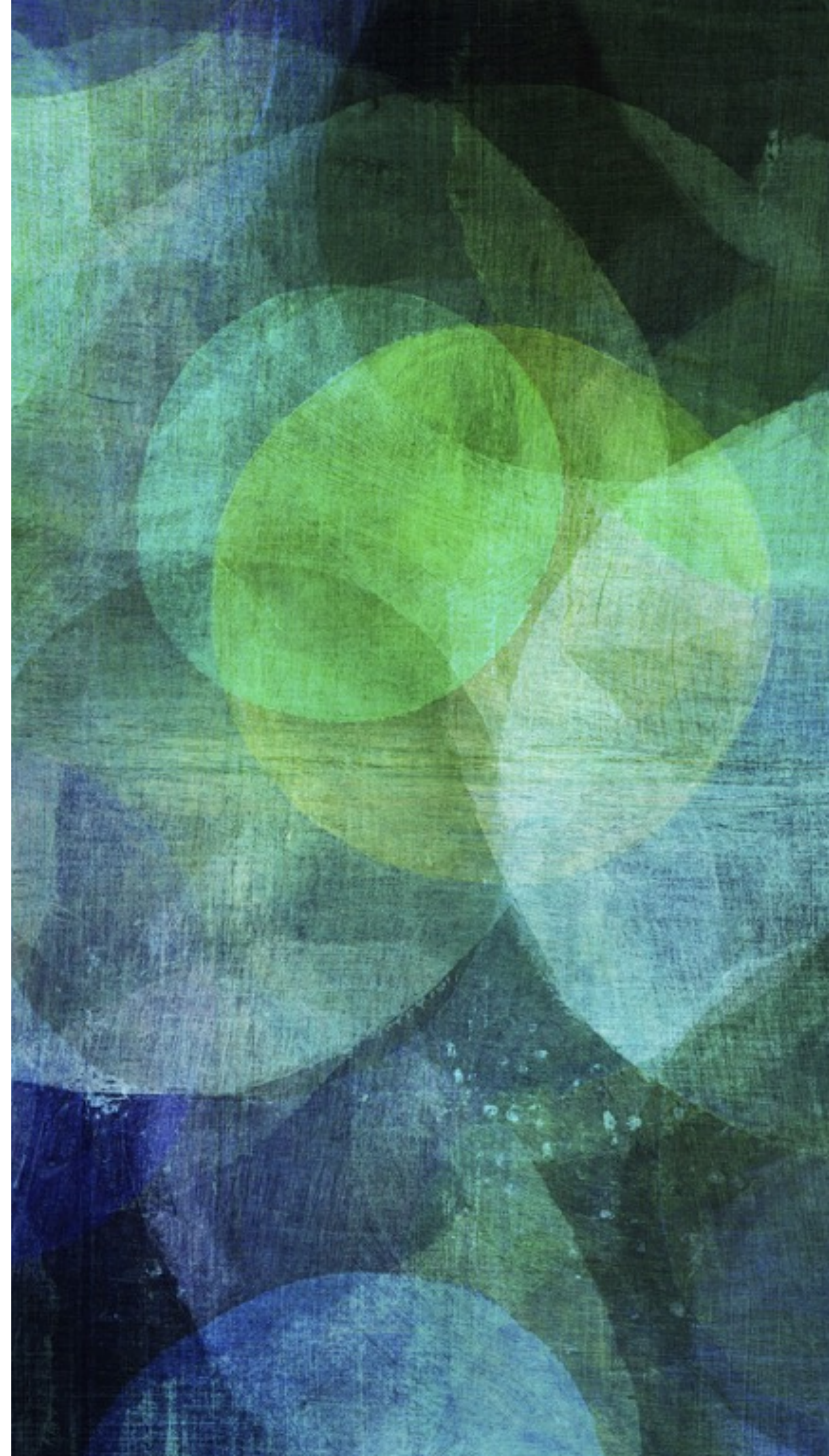
NEW DIRECTIONS: UNKNOWN GRAPHS!

- **Learning on the graph while learning the graph?**
  - **most of algorithms require (some) knowledge of the graph**
  - not always available to the learner
- Question: Can we learn faster without knowing the graphs?
  - example: social network provider has little incentive to reveal the graphs to advertisers
- Answer: **Cohen, Hazan, and Koren:** Online learning with **feedback** graphs without the graphs (ICML June 19-24, 2016)
  - **NO!** (in general we cannot, but possible in the stochastic case)
- Coming up next:
  - **Erdös-Rényi side observation graphs** (UAI June 25-26, 2016)
  - **Influence Maximisation** (AISTATS 2016 + https://arxiv.org/abs/1605.06593)

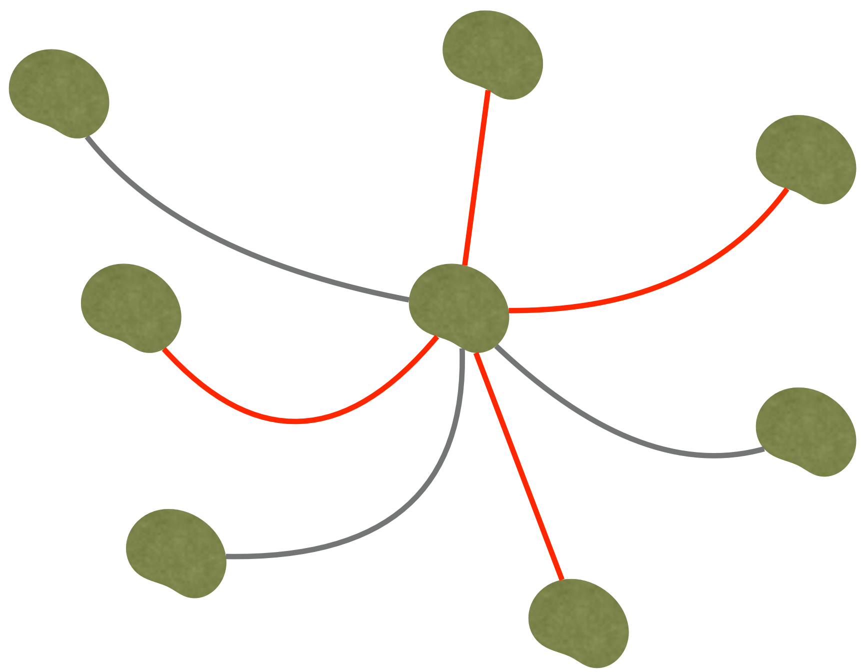
Kocák, Neu, MV: **Online learning with Erdos-Rényi side-observation graphs**
UAI 2016 (to appear)

GRAPH BANDITS WITH ERDÖS-RÉNYI OBSERVATIONS

side observations from graph generators



PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



Every round **t** the learner

- picks a node I_t
- suffers loss for I_t
- receives feedback
  - for I_t
  - for every other node with probability r_t

is loss of i observed?

true loss

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

probability of picking i

probability of side observation

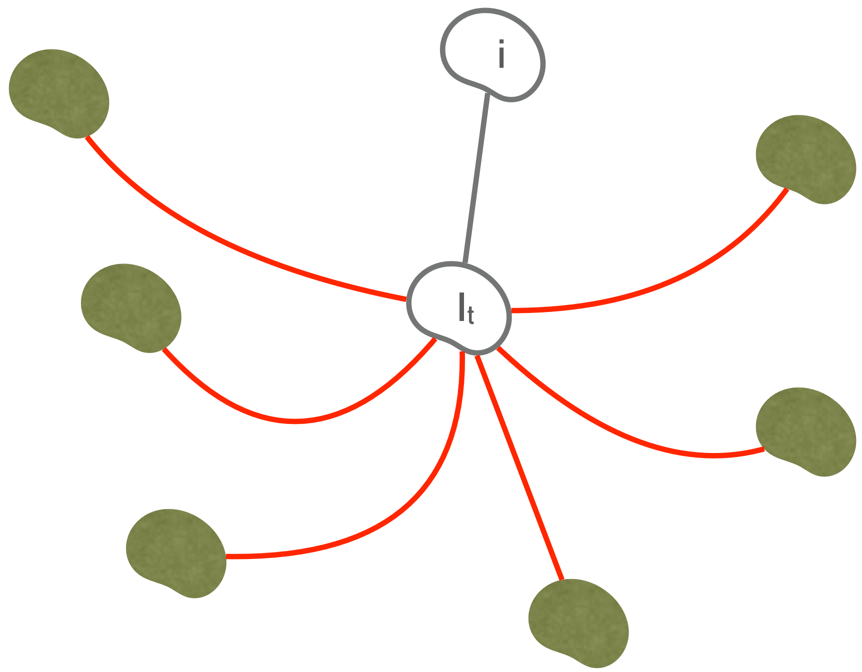
Regret of Exp3-SET (Alon et al. 2013):

$$\mathcal{O}\left(\sqrt{\textstyle\sum_t (1/r_t)(1 - (1 - r_t)^N)\log N}\right)$$

How to estimate r_t in every round when it is changing?

How to estimate losses without the knowledge of r_t?

PROTOCOL FOR ERDÖS-RÉNYI GRAPHS



- N-2 samples from Bernoulli(r_t) ... R(k)
- N-2 samples from p_{ti} ... P(k)
- O'(k) = P(k) + (1-P(k))R(k)
- G_{ti} = min{k : O'(k) = 1} U {N-1}
- E[G_{ti}] ≈ 1/(p_{ti} + (1-p_{ti})r_t)

$$\widehat{\ell}_{t,i} = G_{t,i} O_{t,i} \ell_{t,i}$$

is loss of i observed?

true loss

$$\widehat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t}$$

probability of picking i

probability of side observation

If r_t ≥ (log T)/(2N-2) then

$$\mathcal{O}\left(\sqrt{\log N \sum_{t=1}^{T} \frac{1}{r_t}}\right)$$

Lower bound (Alon et al. 2013) $\Omega(\sqrt{T/r})$

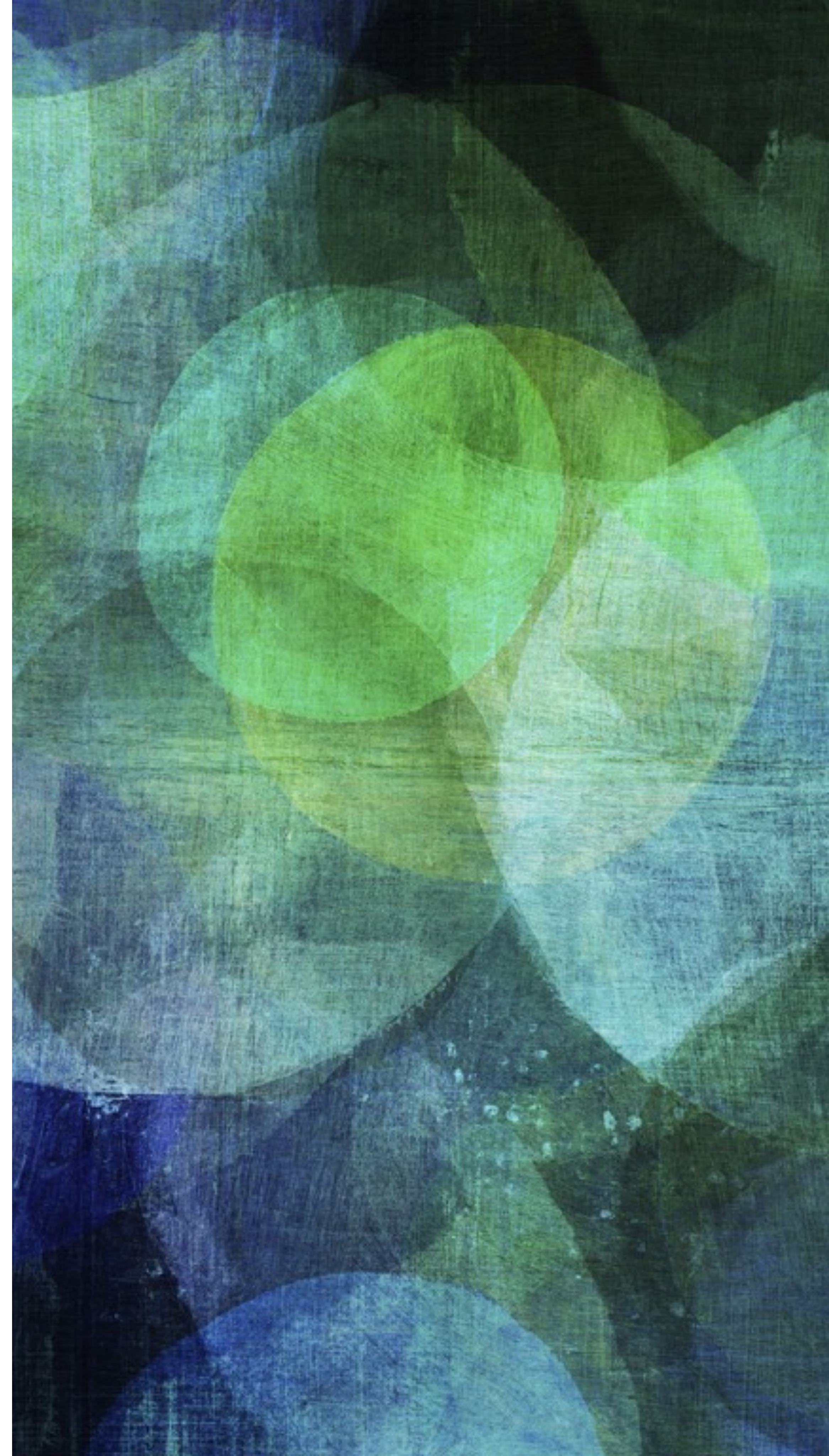
Get rid of r_t ≥ (log T)/(2N-2)?

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

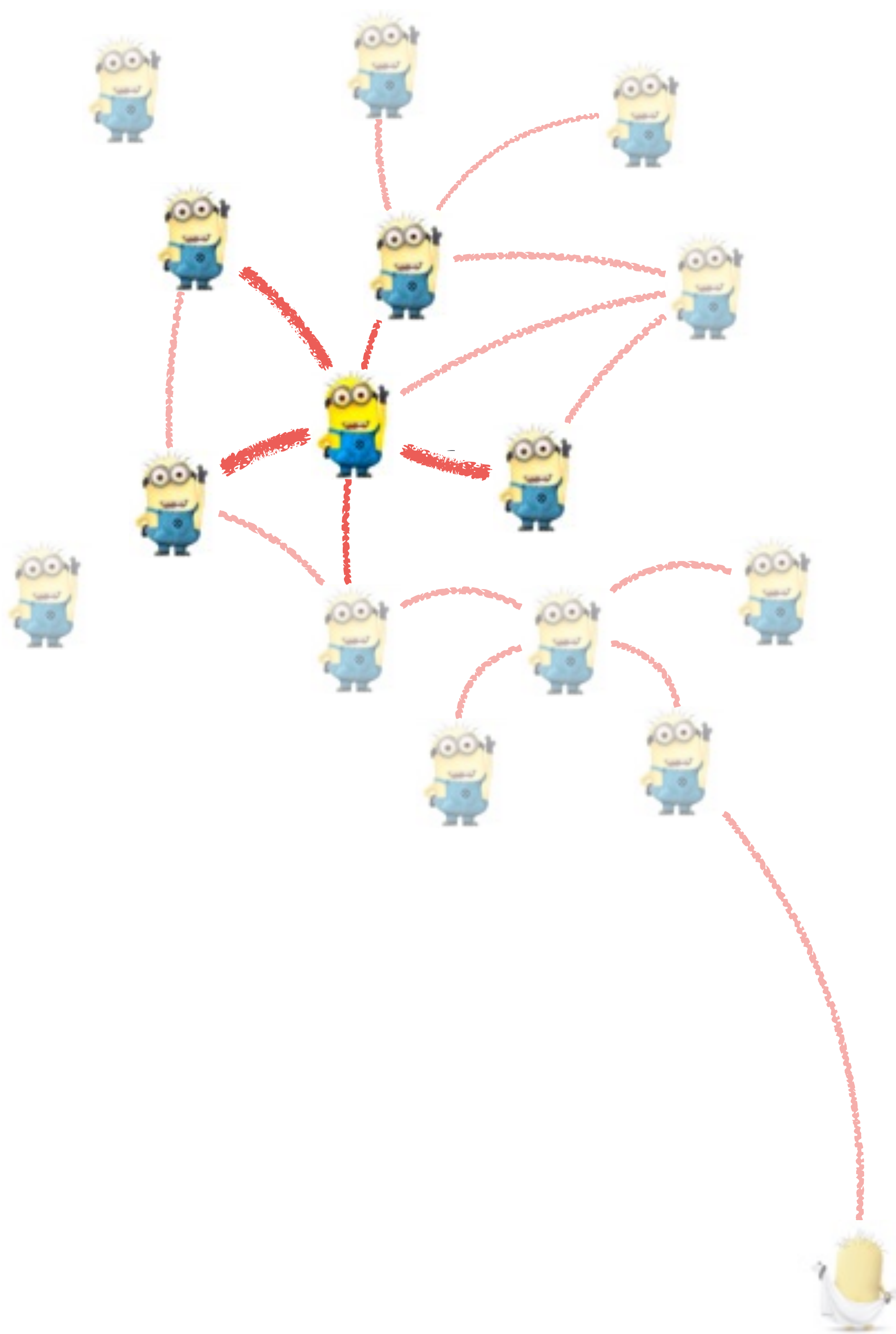
Wen, Kveton, MV: **Influence Maximization with Semi-Bandit Feedback,** (arXiv:1605.06593)

INFLUENCE MAXIMISATION

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



MAXIMIZING INFLUENCE



Product placement

- dispatch few to sell more
- target influential people

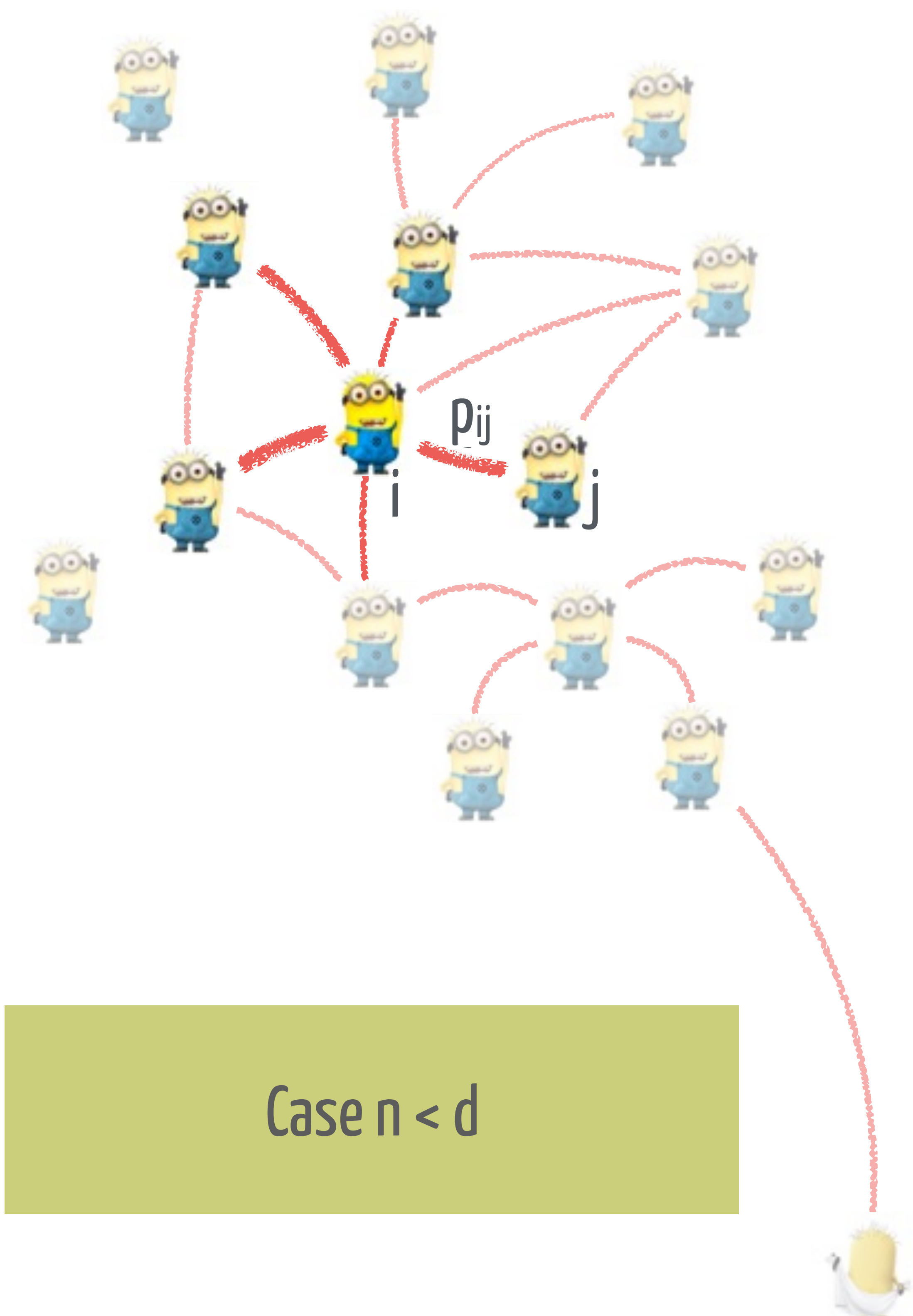
Gathering the information?

- 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

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

What 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

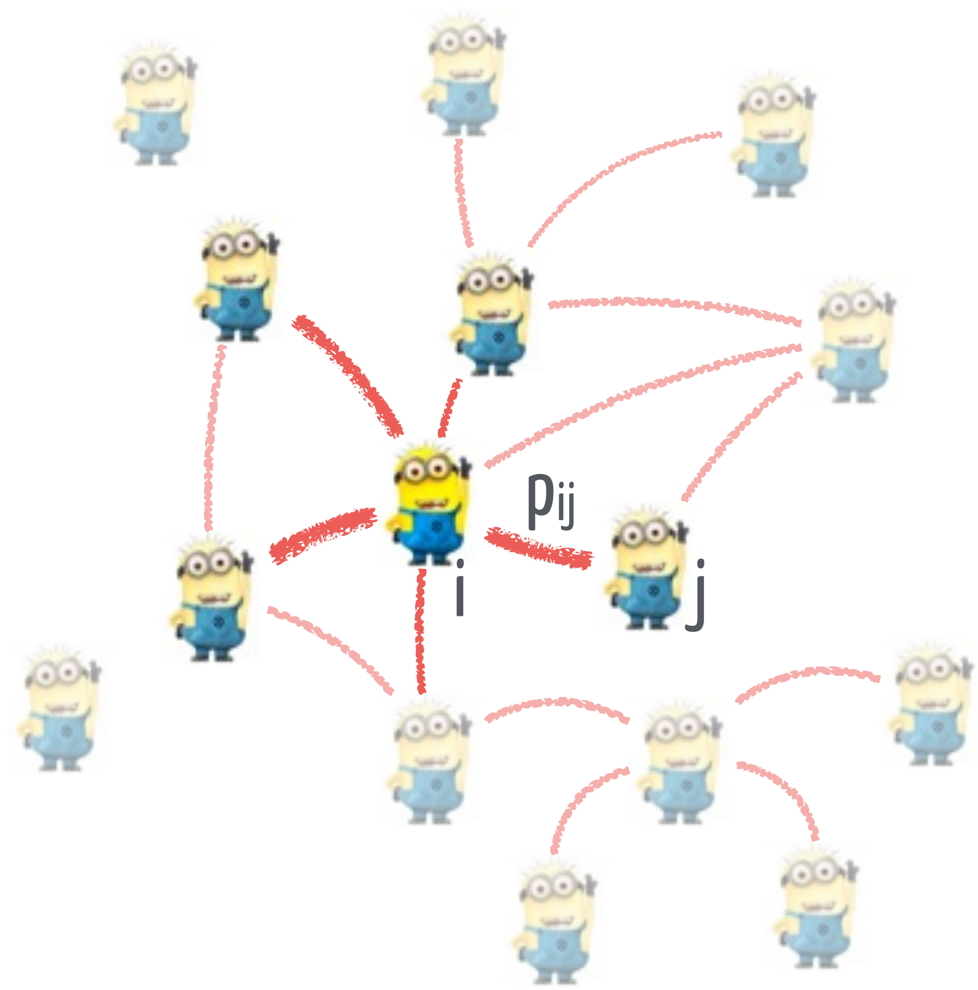
$$r_k = \mathbb{E}\left[|S_{k,t}|\right] = \sum_{j \le d} p_{k,j}$$

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 regret of the oracle strategy

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



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

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

BASELINE

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

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

each node at least once

unlearnable case n ≤ d

- matching lower bound

Crash course on stochastic bandits?

GRAPHMOSS FOR THE RESTRICTED SETTING

informatics mathematics Inria

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

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

each node at least once

unlearnable case n ≤ d

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

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

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

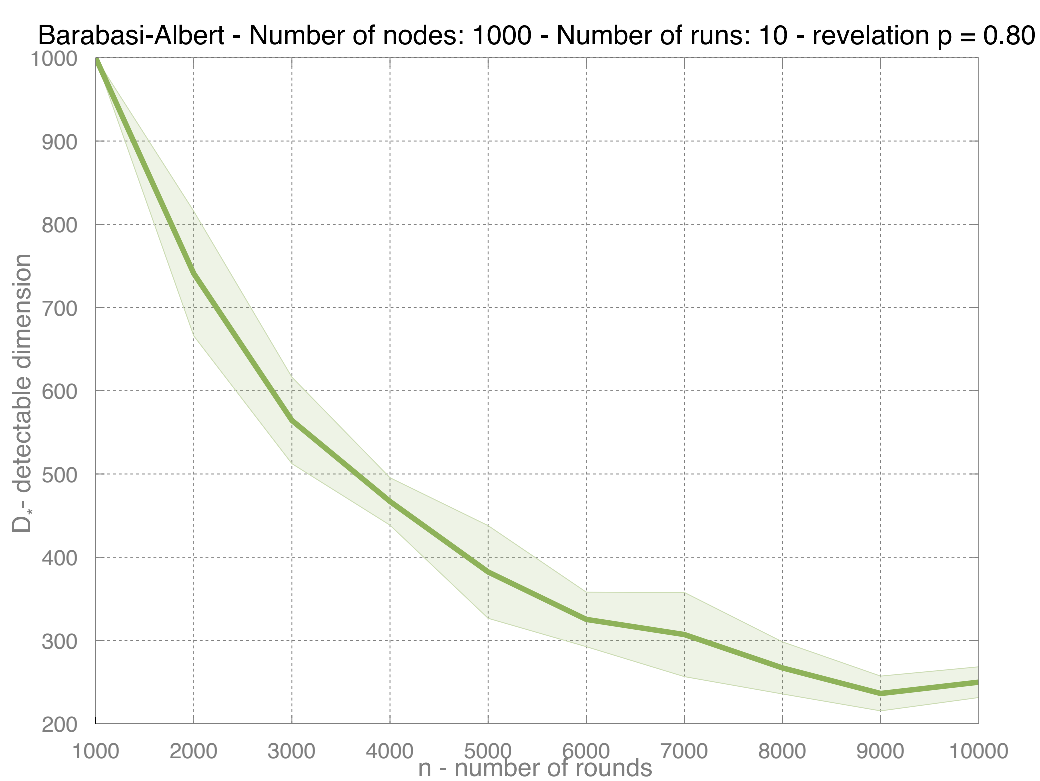
$$\Delta_\star \stackrel{\text{def}}{=} 16\sqrt{\frac{r_\star^\circ d \log(nd)}{T_\star}} + \frac{80 d \log(nd)}{T_\star}$$

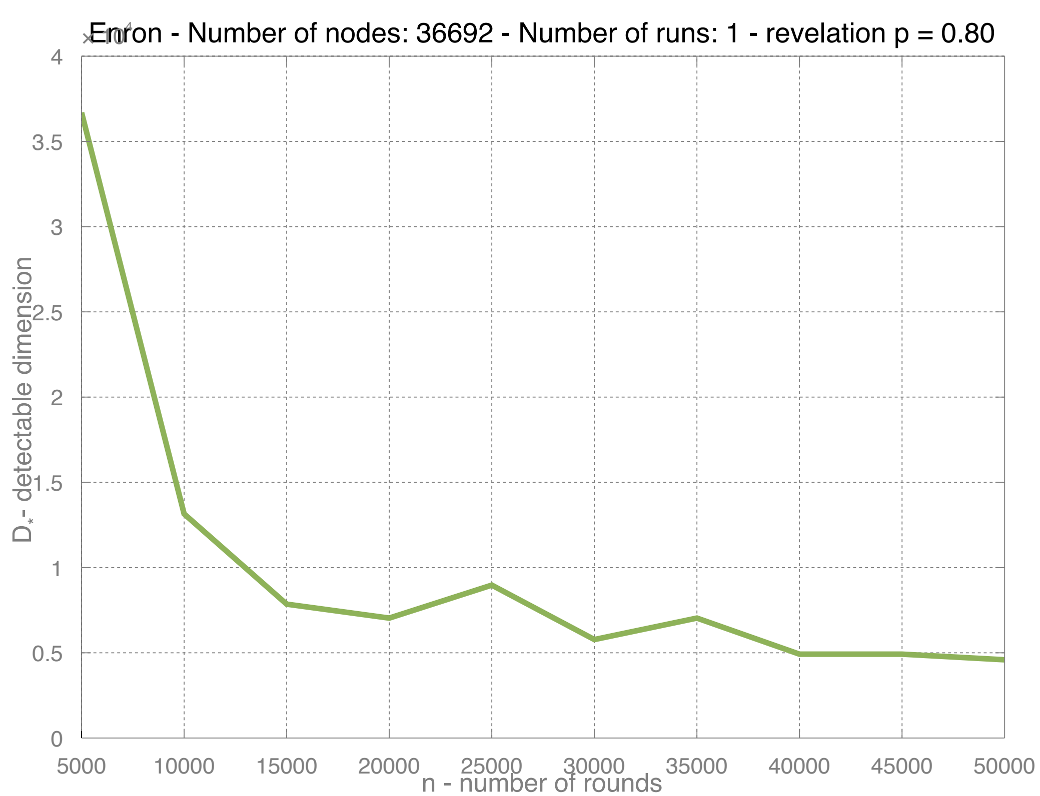
- Detectable horizon T_*, smallest integer s.t. $T_\star r_\star^\circ \geq \sqrt{D_\star n r_\star^\circ}$
- Equivalently: D_* corresponding to smallest T_* such that

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

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





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

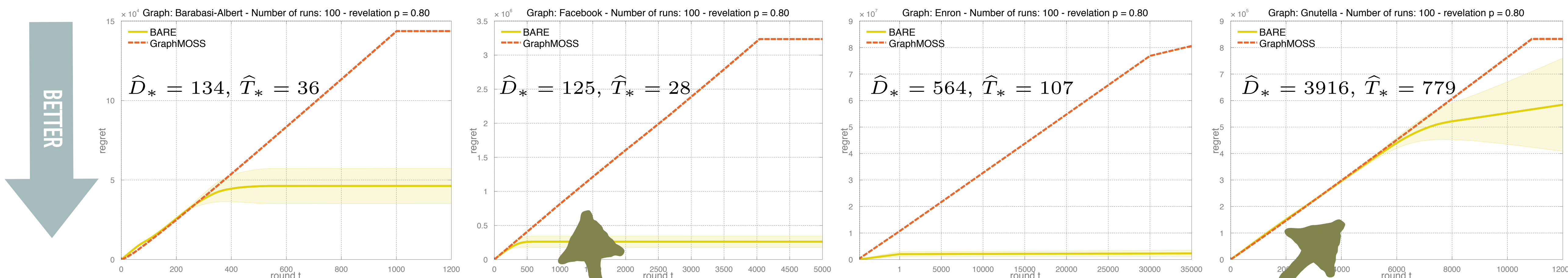
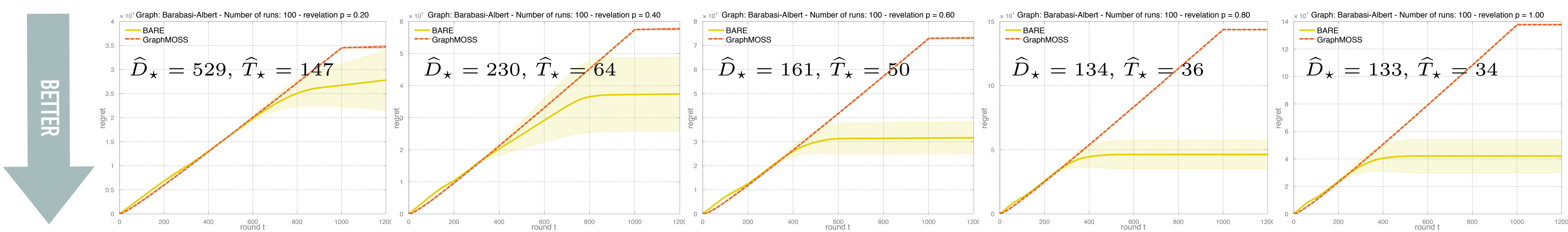


Figure 1: *Left*: Barabási-Albert. *Middle left*: Facebook. *Middle right*: Enron. *Right*: Gnutella.

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



Varying a (constant) probability of influence

REVEALING BANDITS: WHAT DO YOU MEAN?

- Ignoring the structure again? $\tilde{\mathcal{O}}\left(\sqrt{r_* T N}\right)$
  - 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

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

- **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)
    - http://arxiv.org/abs/1605.06593 (Wen, Kveton, MV)
    - IMLinUCB with linear parametrization of edge weights
    - Regret analysis for subset of graphs (forests, ...)

Advanced Learning for Text and Graph Data

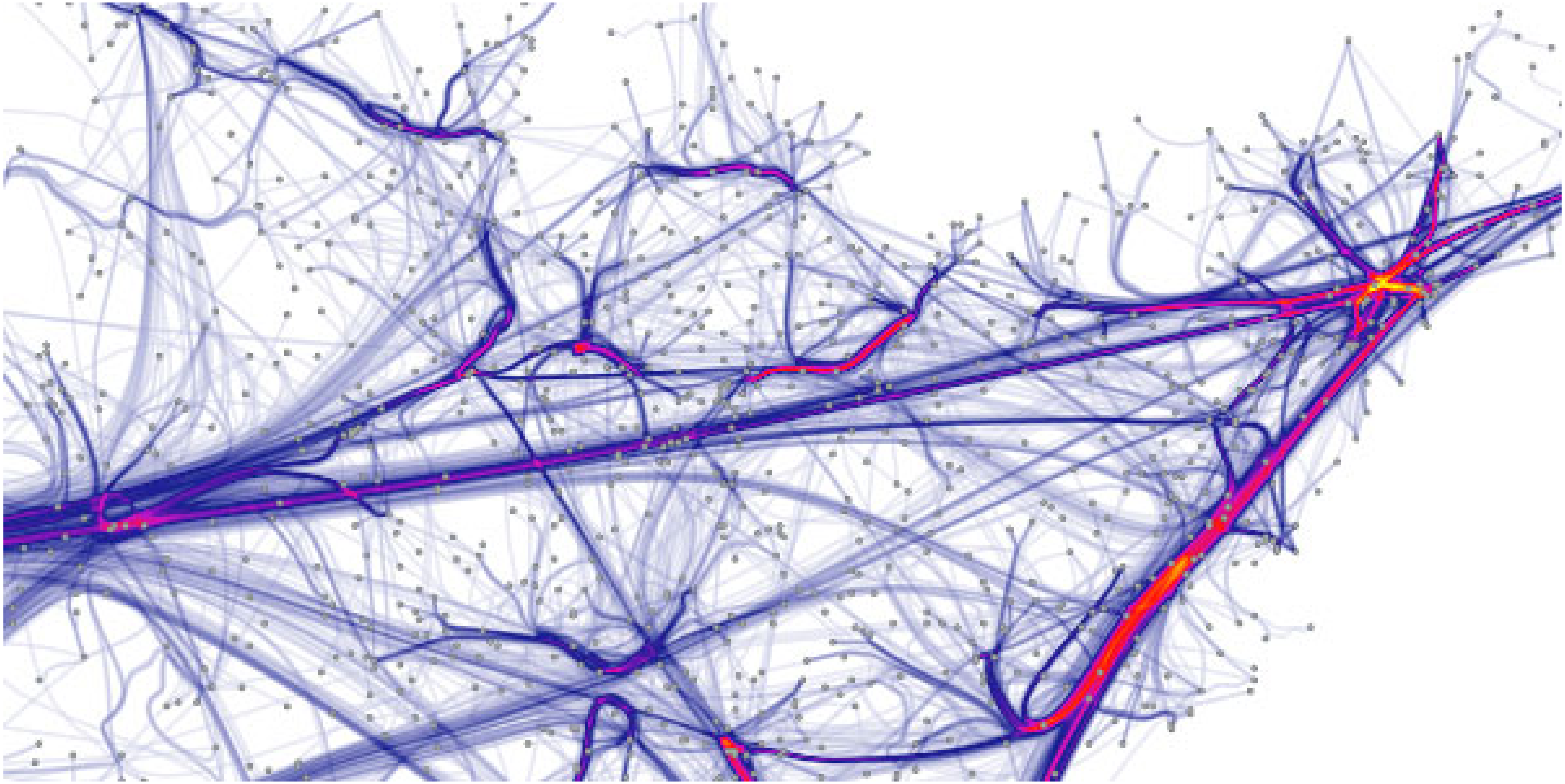
Time: Spring term **4 lectures** and **3 Labs**
Place: Polytechnique / Amphi Sauvy
Lecturer 1: Michalis Vazirgiannis (Polytechnique)
Lecturer 2: Yassine Faihe (Hewlett-Packard - Vertica)

ALTeGraD follows after **Graphs in ML**
The two graph courses are coordinated to be complementary.

Some of covered graph topics not covered in this course

- Ranking algorithms and measures (Kendal Tau, NDCG)
- Advanced graph generators
- Community mining, advanced graph clustering
- Graph degeneracy (k-core & extensions)
- Privacy in graph mining

THANK YOU!



Michal Valko
michal.valko@inria.fr
ENS Paris-Saclay, MVA 2016/2017
SequeL team, Inria Lille — Nord Europe
https://team.inria.fr/sequel/