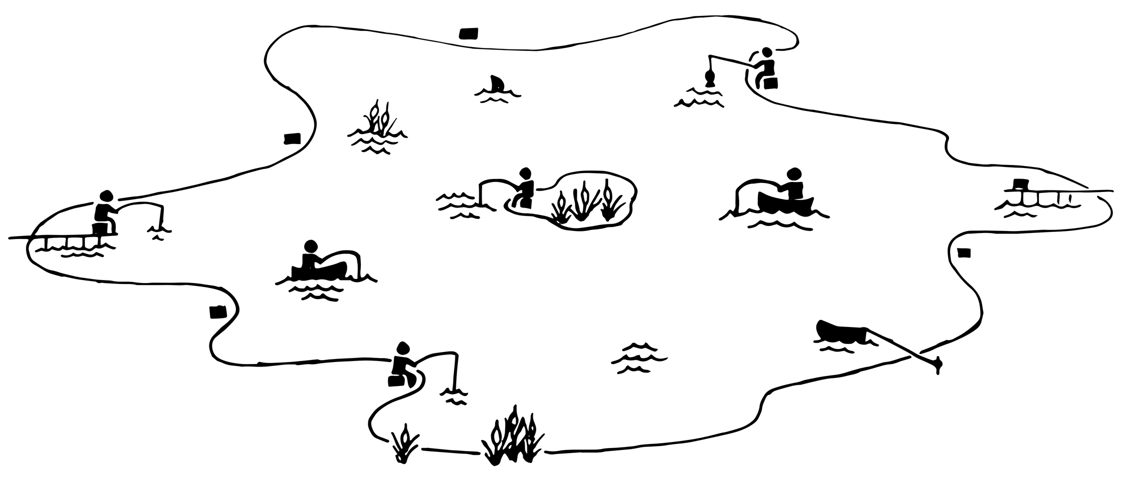
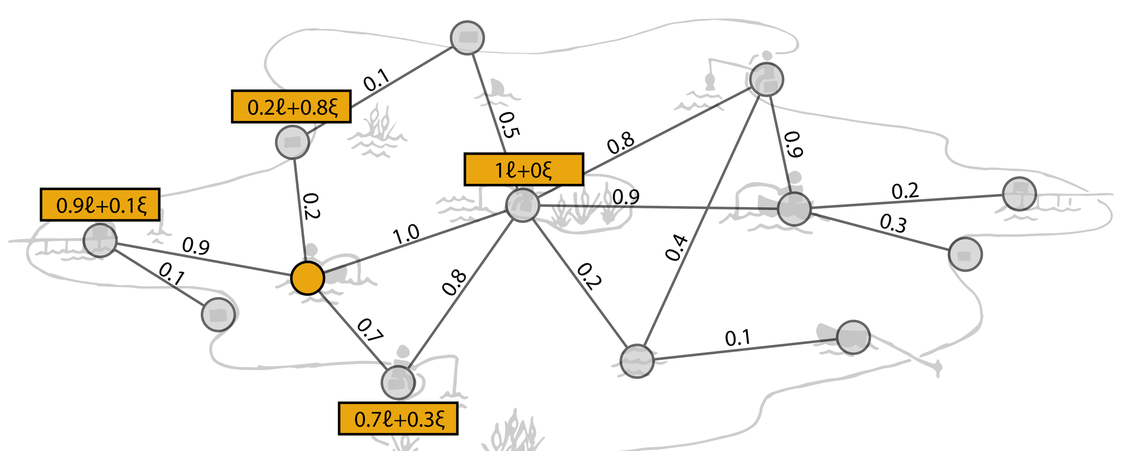
Online learning with noisy side observations

Tomas.Kocak@inria.fr, Neu.Gergely@gmail.com, and Michal.Valko@inria.fr

Motivation - Fishing spots





Setup

- **Pick** a fishing **spot** (at the beginning of the day)
- **Obtain** some amount of **fish** (at the end of the day)
- **Observe** other fishermen (**noisy** side observations)
  - Does not represent what you would get exactly
  - Different fishing lure, motivation, skill…
- **Goal:** catch as many fish as possible over the time

Side observations - prior work (Mannor and Shamir 2011)

- Playing an action reveals the losses of neighboring spots

***Noisy* side observations** - generalization

- Side observations represented by a **weighted** graph
- Playing action reveals **noisy** the losses of neighboring spots

Problem formalization

Learning process

- N **actions** (nodes of a graph)
- T **rounds:**
  - Environment (adversary) **sets losses** for actions
  - Environment **choses a graph** (not disclosed)
  - Learner **picks an action** I_t to play
  - Learner **incurs the loss** ℓ_{t,I_t} of the action I_t
  - Learner **observes graph** (second neighborhood of I_t)
  - Learner **observes noisy losses** $c_{t,j}$ of neighbors $j \in N(I_t)$

where

$$c_{t,j} = s_{t,(I_t,j)}\ell_{t,j} + (1 - s_{t,(I_t,j)})\xi_{t,j}$$

- $s_{t,(I_t,j)}$: weight of the edge from node I_t to node j at time t
- $\ell_{t,j}$: loss of the action j at time t
- $\xi_{t,j}$: zero mean noise such that $|\xi_{t,j}| \le R$

Goal of the learner: minimizing cumulative regret R_t defined as

$$R_t = \underbrace{\sum_{t=1}^{T} \ell_{t,I_t}}_{\text{learner}} - \underbrace{\min_{j\in[N]} \left[\sum_{t=1}^{T} \ell_{t,j}\right]}_{\text{best action}}$$

Exp3-type algorithm template

- **Compute exponential weights** using loss estimates $\widehat{\ell}_{s,i}$

$$w_{t,i} = \exp\left(-\eta_t \sum_{s=1}^{t-1} \widehat{\ell}_{s,i}\right)$$

- **Create a probability distribution** such that $p_{t,i} \propto w_{t,i}$
- **Play action** I_t such that

$$\mathbb{P}(I_t = i) = p_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{N} w_{t,j}}$$

- **Create loss estimates** (using observability graph)
  - The definition of $\widehat{\ell}$ defines an Exp3-type algorithm

Loss estimates of algorithms

Desired property of loss estimates: $\mathbb{E}\left[\widehat{\ell}_{t,i}\right] \approx \ell_{t,i}$

Typical estimates: $\widehat{\ell}_{t,i} = \frac{c_{t,i}\mathbb{1}\{\text{arm } i \text{ is observed}\}}{\mathbb{P}\{\text{arm } i \text{ is observed}\}}$

Graphs without weights (known algorithms)

Exp3 (edgeless graph)

$$\widehat{\ell}_{t,i} = \frac{\ell_{t,i}}{p_{t,i}}\mathbb{1}\{\text{arm } i \text{ is observed}\}$$

Hedge (full graph)

$$\widehat{\ell}_{t,i} = \frac{\ell_{t,i}}{1}\mathbb{1}\{\text{arm } i \text{ is observed}\}$$

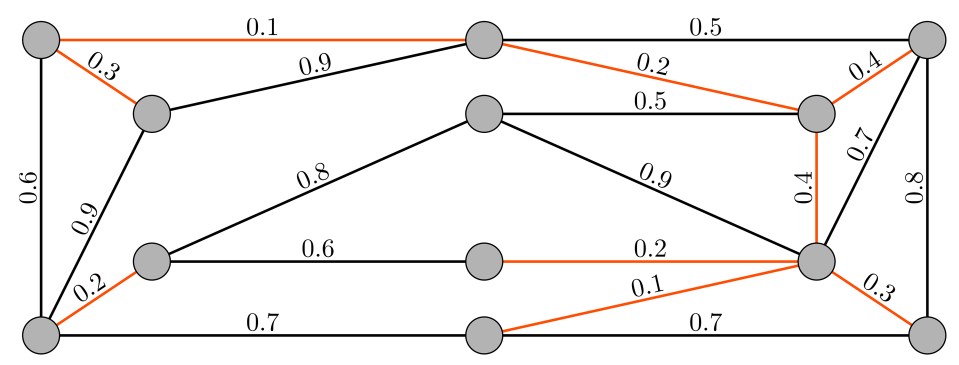
Exp3-IX (general graph)

$$\widehat{\ell}_{t,i} = \frac{\ell_{t,i}}{\sum_{i\in N(j)} p_{t,j} + \gamma_t}\mathbb{1}\{\text{arm } i \text{ is observed}\}$$

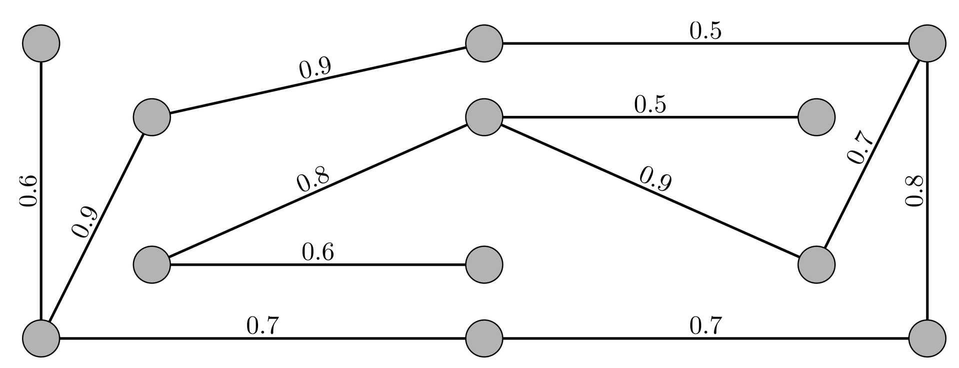
Graphs with weights (new algorithms)

Exp3-IXt (general graphs with weights and thresholding)

- Use only **"reliable"** observations with low noise
- Delete all the edges with weights smaller than ε



↓



$$\widehat{\ell}_{t,i} = \frac{s_{t,(I_t,i)}\ell_{t,i} + (1 - s_{t,(I_t,j)})\xi_{t,i}}{\sum_{j=1}^{N} s_{t,(j,i)}p_{t,j} + \gamma_t}$$

Exp3-WIX (general graph with weights)

$$\widehat{\ell}_{t,i} = \frac{s_{t,(I_t,i)}\left[s_{t,(I_t,i)}\ell_{t,i} + (1 - s_{t,(I_t,j)})\xi_{t,i}\right]}{\sum_{j=1}^{N} s_{t,(j,i)}^{2}p_{t,j} + \gamma_t}$$

Theoretical guarantees

Exp3-IXt

- **Regret upper-bound**

$$\mathbb{E}\left[R_t\right] = \widetilde{\mathcal{O}}\left(\sqrt{\frac{\alpha(\varepsilon)}{\varepsilon^2}T}\right)$$

- – Needs to know the "best" ε to optimize regret bound
- – Needs to know whole graph to find the best Exp3

Exp3-WIX (main result)

- **Regret upper-bound**

$$\mathbb{E}\left[R_t\right] = \widetilde{\mathcal{O}}\left(\sqrt{\min_{\varepsilon\in[0,1]}\frac{\alpha(\varepsilon)}{\varepsilon^2}T}\right)$$

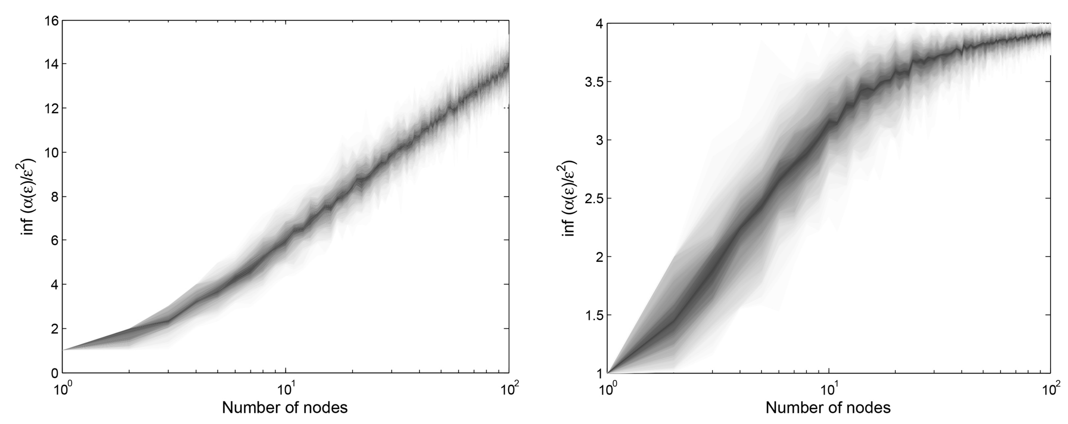
- + Does not need to set any threshold ε
- + Does not need to compute any independence numbes α
- Regret bound as good as the best bound of Exp3-IXt

Effective independence number

Effective independence number α^* is

$$\alpha^* = \min_{\varepsilon\in[0,1]}\frac{\alpha(\varepsilon)}{\varepsilon^2}$$

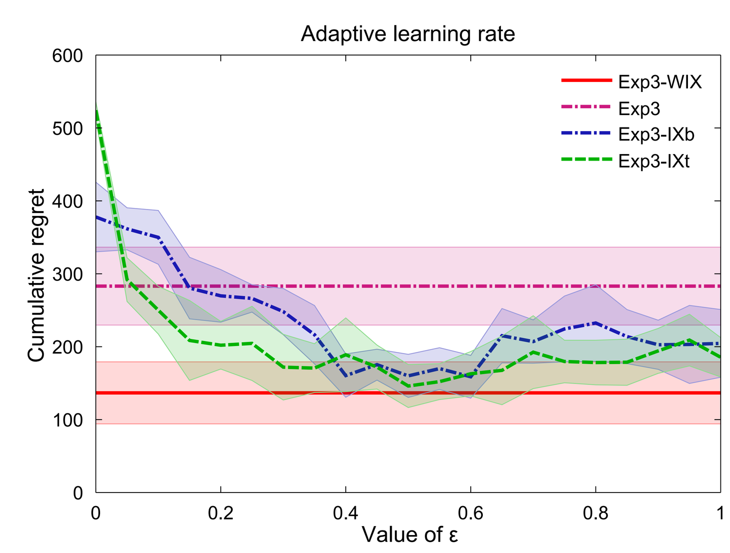
The empirical value of α^* uniformly random weights



$U(0,1)$ weights $\quad$ $U(\frac{1}{2},1)$ weights

Empirical results

- **Exp3** - basic algorithm which ignores all side observations
- **Exp3-WIX** - our proposed algorithm
- **Exp3-IXt** - thresholded algorithm (needs to set ε)
- **Exp3-IXb** - algorithm ignores noise (no guaranties)



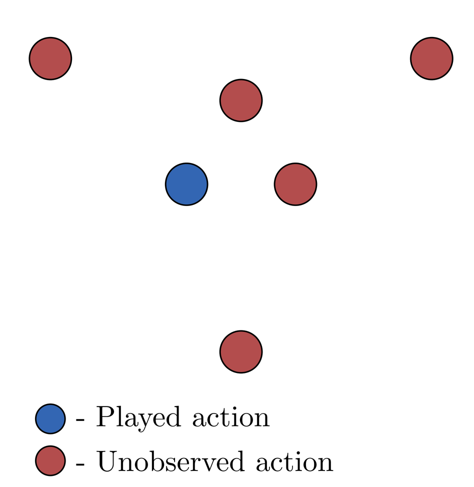
Conclusion

- **New setting** with noisy side observations
- Introduction of **effective independence number** α^*
- **Exp3-WIX** algorithm for the setting
  - Does not need to threshold
  - Does not need to know whole graph
  - Regret bound of order $\sqrt{\alpha^* T}$
- **Open questions:**
  - Is the effective independence number "right quantity?"
  - Is there a matching lower-bound for Exp3-WIX?
    - Upper-bound of Exp3-WIX matches lower-bound for some cases (e.g., bandits, full information, and setting of Mannor and Shamir 2011)
  - Related lower-bound (Wu et al. 2015) for a stochastic setting with Gaussian noise

$$R_t = \Omega\left(\sqrt{\frac{\alpha}{\varepsilon^2}T}\right)$$

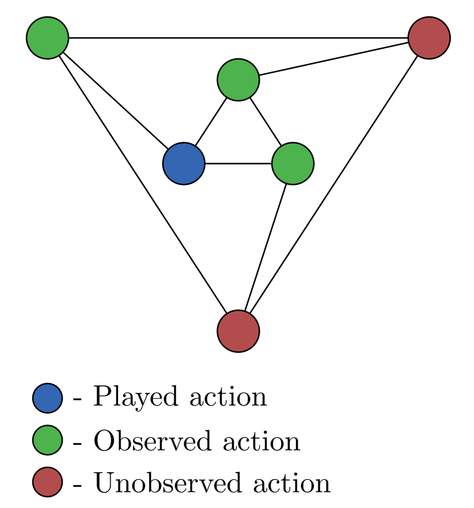
Special cases of the framework

Edgeless graphs - Bandit setting



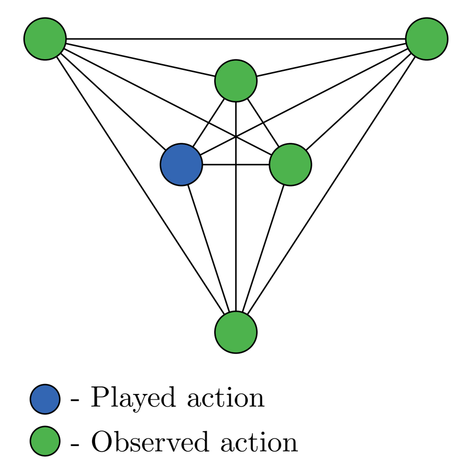
- No side observations
- Regret bound of order

General graphs - Mannor and Shamir 2011



- Some losses observed (without noise)
- Regret bound of order

Complete graphs - Full information setting



- All losses observed (without noise)
- Regret bound of order

$$\sqrt{NT} \quad \ge \quad \sqrt{\alpha T} \quad \ge \quad \sqrt{T}$$