Online learning with noisy side observations

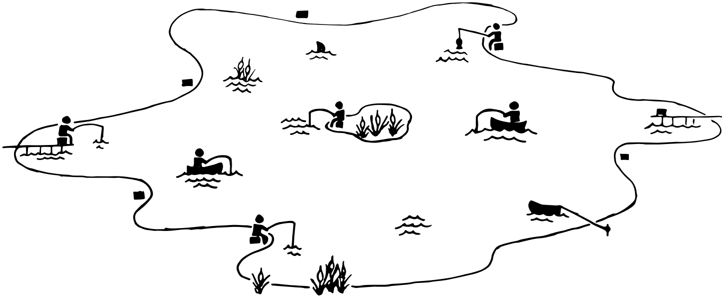
Tomáš Kocák *Inria Lille - Nord Europe, France*
Gergely Neu *DTIC, Universitat Pompeu Fabra, Barcelona, Spain*
Michal Valko *Inria Lille - Nord Europe, France*

SequeL – Inria Lille
AISTATS 2016

Cadiz, April 2016

Example

Fishing spots



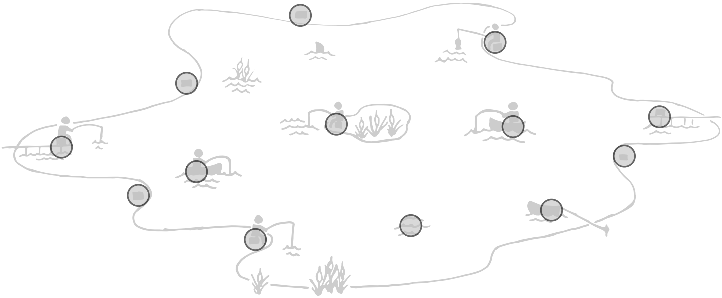
Every day:

- Choose fishing spot (at the beginning of the day)
- Maximize total number of fish caught

This example is motivated by an example of Wu et al. 2015

Example

Fishing spots



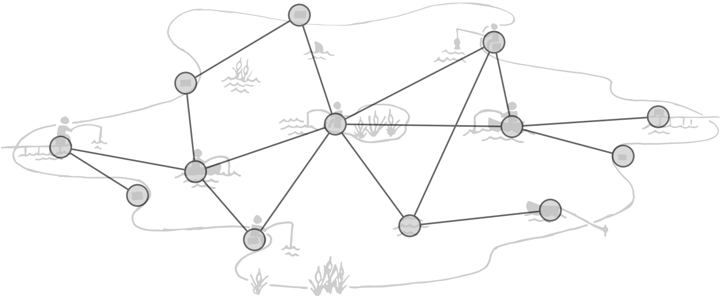
Every day:

- Choose fishing spot (at the beginning of the day)
- Maximize total number of fish caught

This example is motivated by an example of Wu et al. 2015

Problem representation

Previous representation, introduced by Mannor and Shamir 2011

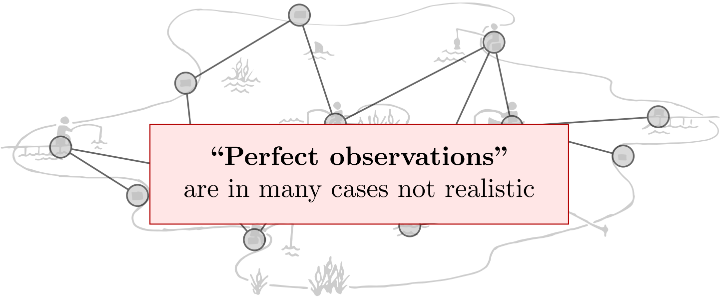


- Actions are nodes of directed **unweighted** graph
- Playing action reveals losses $c_{t,j}$ of neighbors $j \in N(I_t)$

$$c_{t,j} = \ell_{t,j}$$

Problem representation

Previous representation, introduced by Mannor and Shamir 2011



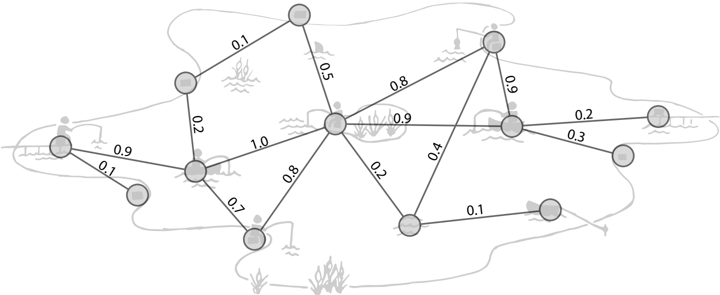
"Perfect observations"
are in many cases not realistic

- Actions are nodes of directed **unweighted** graph
- Playing action reveals losses $c_{t,j}$ of neighbors $j \in N(I_t)$

$$c_{t,j} = \ell_{t,j}$$

Problem representation

Problem representation in our paper

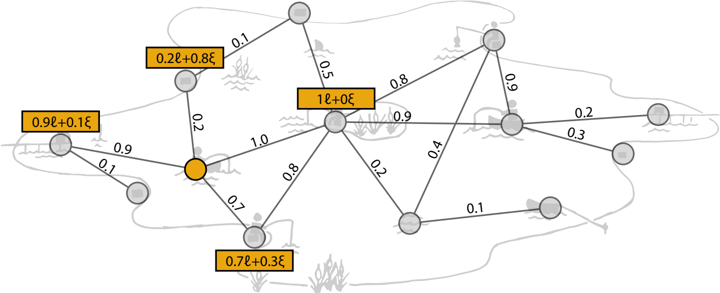


- Actions are nodes of directed weighted graph.
- Playing action reveals noisy losses $c_{t,j}$ of neighbors $j \in N(I_t)$.
  - Smaller weight $s_{t,(I_t,j)} \to$ bigger noise

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

Problem representation

Problem representation in our paper



- Actions are nodes of directed weighted graph.
- Playing action reveals noisy losses $c_{t,j}$ of neighbors $j \in N(I_t)$.
  - Smaller weight $s_{t,(I_t,j)} \rightarrow$ bigger noise

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

Framework

Description and goals

Learning process

- N actions (nodes of a graph)
- T rounds:
  - Environment sets losses for actions
  - Environment choses a graph structure (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
  - Learner observes noisy losses $c_{t,j}$ of the neighbors $j \in N(I_t)$

Goal of the learner

Minimizing cumulative regret

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

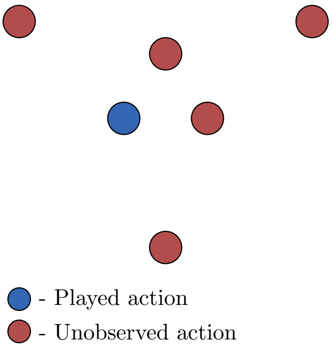
Regret bounds - special cases

Unweighted graphs

Edgeless graph (Bandit problem)

- No side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{NT})$

Edgeless graph



Regret bounds - special cases

Unweighted graphs

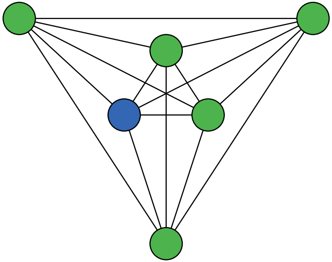
Edgeless graph (Bandit problem)

- No side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{NT})$

Complete graph (Full information)

- All side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{T})$

Complete graph



- Played action
- Observed action

Regret bounds - special cases

Unweighted graphs

Edgeless graph (Bandit problem)

- No side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{NT})$

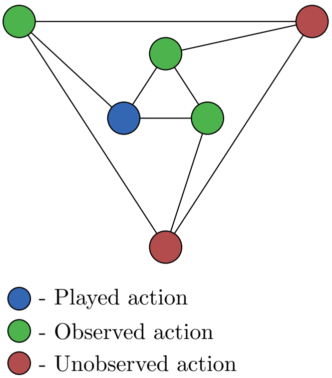
Complete graph (Full information)

- All side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{T})$

General unweighted graph (Mannor and Shamir 2011)

- Some side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{\alpha T})$
- α - independence number

General graph



Regret bounds - special cases

Unweighted graphs

Edgeless graph (Bandit problem)

- No side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{NT})$

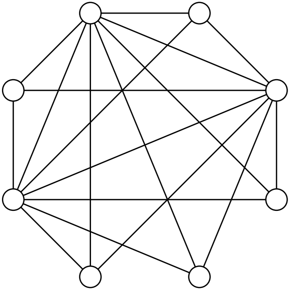
Complete graph (Full information)

- All side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{T})$

General unweighted graph
(Mannor and Shamir 2011)

- Some side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{\alpha T})$
- α - independence number

Independence number



- Size of the largest independence set
  - Not connected nodes
- Example above: $\alpha = 5$

Regret bounds - special cases

Unweighted graphs

Edgeless graph (Bandit problem)

- No side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{NT})$

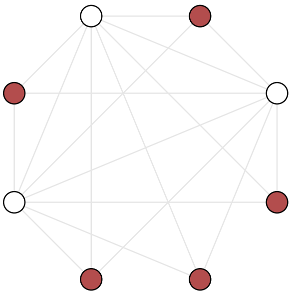
Complete graph (Full information)

- All side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{T})$

General unweighted graph
(Mannor and Shamir 2011)

- Some side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{\alpha T})$
- α - independence number

Independence number



- Size of the largest independence set
  - Not connected nodes
- Example above: $\alpha = 5$

Regret bounds - special cases

Unweighted graphs

Edgeless graph (Bandit problem)

- No side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{NT})$

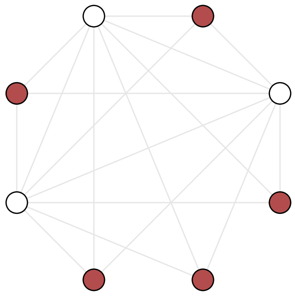
Complete graph (Full information)

- All side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{T})$

General unweighted graph (Mannor and Shamir 2011)

- Some side observations
- Regret bound of $\widetilde{\mathcal{O}}(\sqrt{\alpha T})$
- α - independence number

Independence number



- Size of the largest independence set
  - Not connected nodes
- Example above: $\alpha = 5$

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

Exp3 type algorithm template

In every round:

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

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

- **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** $\hat{\ell}_{t,i}$ (using observability graph)
  - Loss estimates define the algorithm
  - Loss estimates compensate for lack of side observations

Exp3 type algorithm template

In every round:

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

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

- **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** $\hat{\ell}_{t,i}$ (using observability graph)

Question:

What are "good" loss estimates?

Loss estimates

Desired property of loss estimates:

- Good loss approximation (unbiased estimates)

$$\mathbb{E}\left[\hat{\ell}_{t,i}\right] \approx \ell_{t,i}$$

Loss estimates

Desired property of loss estimates:

- Good loss approximation (unbiased estimates)

$$\mathbb{E}\left[\hat{\ell}_{t,i}\right] \approx \ell_{t,i}$$

We can use noisy side observations:

$$\mathbb{E}\left[c_{t,i}\right] = \underbrace{\mathbb{E}\left[s_{t,(I_t,i)}\ell_{t,j}\right]}_{\left(\sum_{j=1}^{N} s_{t,(j,i)}p_{t,j}\right)\ell_{t,i}} + \underbrace{\mathbb{E}\left[(1 - s_{t,(I_t,i)})\xi_{t,i}\right]}_{0}$$

Loss estimates

Desired property of loss estimates:

- Good loss approximation (unbiased estimates)

$$\mathbb{E}\left[\hat{\ell}_{t,i}\right] \approx \ell_{t,i}$$

We can use noisy side observations:

$$\mathbb{E}\left[c_{t,i}\right] = \underbrace{\mathbb{E}\left[s_{t,(I_t,i)}\ell_{t,j}\right]}_{\left(\sum_{j=1}^{N} s_{t,(j,i)}p_{t,j}\right)\ell_{t,i}} + \underbrace{\mathbb{E}\left[(1 - s_{t,(I_t,i)})\xi_{t,i}\right]}_{0}$$

Loss estimates

$$\hat{\ell}_{t,i} = \frac{c_{t,i}}{\sum_{j=1}^{N} s_{t,(j,i)}p_{t,j}}$$

Loss estimates

First attempt

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

Pros:

- Unbiased estimates (good approximation of real losses)

Cons:

- Unreliable observations are included (small weights)
- Large variance of estimates
- No theoretical guaranties!

Loss estimates

First attempt

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

Pros:

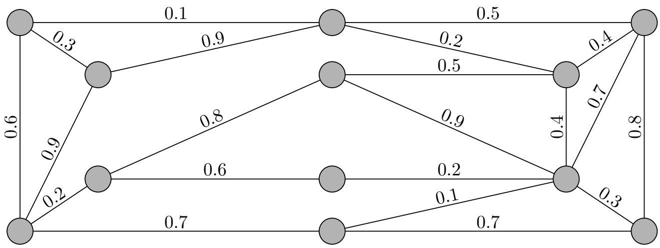
- Unbiased estimates (good approximation of real losses)

Cons:

- Unreliable observations are included (small weights)
- Large variance of estimates
- No theoretical guaranties!

Loss estimates

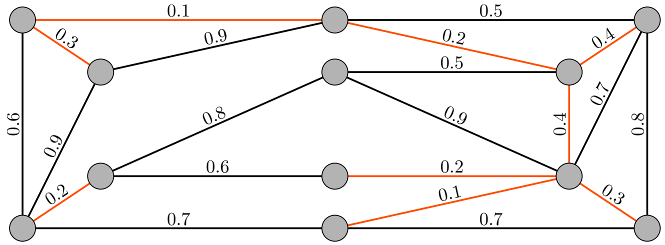
Second attempt - thresholding (Exp3-IXt algorithm)



- Small weights are not reliable

Loss estimates

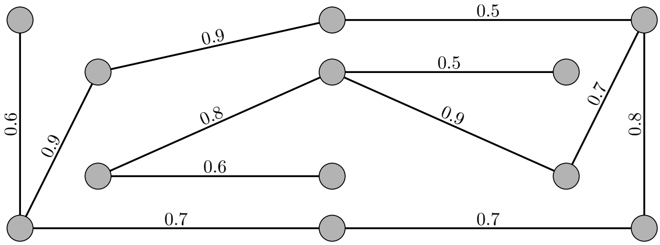
Second attempt - thresholding (Exp3-IXt algorithm)



- Small weights are not reliable

Loss estimates

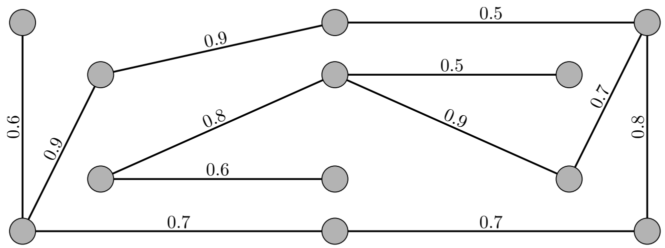
Second attempt - thresholding (Exp3-IXt algorithm)



- Small weights are not reliable

Loss estimates

Second attempt - thresholding (EXP3-IXT algorithm)



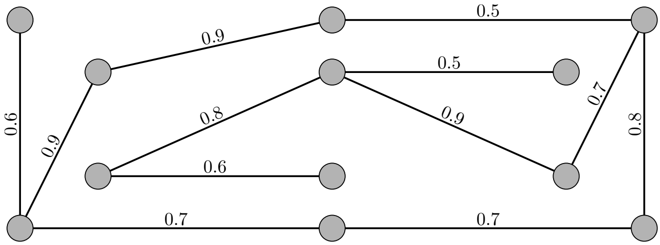
- Small weights are not reliable

Loss estimates for thresholded weights:

$$\hat{\ell}_{t,i} = \frac{c_{t,i}\mathbb{I}\{s_{t,(I_t,i)} \geq \varepsilon\}}{\sum_{j=1}^{N} s_{t,(j,i)}p_{t,j}\mathbb{I}\{s_{t,(j,i)} \geq \varepsilon\}}$$

Loss estimates

Second attempt - thresholding (Exp3-IXt algorithm)



- Small weights are not reliable

Loss estimates for thresholded weights:

$$\hat{\ell}_{t,i} = \frac{c_{t,i}\mathbb{I}\{s_{t,(I_t,i)} \geq \varepsilon\}}{\sum_{j=1}^{N} s_{t,(j,i)} p_{t,j} \mathbb{I}\{s_{t,(j,i)} \geq \varepsilon\}}$$

Loss estimates

Second attempt - Exp3-IXt algorithm

$$\hat{\ell}_{t,i} = \frac{c_{t,i}\mathbb{I}\{s_{t,(I_t,i)} \geq \varepsilon\}}{\sum_{j=1}^{N} s_{t,(j,i)} p_{t,j} \mathbb{I}\{s_{t,(j,i)} \geq \varepsilon\}}$$

Theorem (Regret bound of Exp3-IXt)

When tuned properly, Exp3-IXt *has regret bound of*

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

$\alpha(\varepsilon)$ - independence number of thresholded graph

- Delete all the edges with weight smaller than ε
- Compute independence number (ignoring weights)

Loss estimates

Second attempt - Exp3-IXt algorithm

$$\hat{\ell}_{t,i} = \frac{c_{t,i}\mathbb{I}\{s_{t,(I_t,i)} \geq \varepsilon\}}{\sum_{j=1}^{N} s_{t,(j,i)} p_{t,j} \mathbb{I}\{s_{t,(j,i)} \geq \varepsilon\}}$$

Theorem (Regret bound of Exp3-IXt)

When tuned properly, Exp3-IXt *has regret bound of*

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

$\alpha(\varepsilon)$ - independence number of thresholded graph

- Delete all the edges with weight smaller than ε
- Compute independence number (ignoring weights)

Effective independence number

Effective independence number (setting ε to optimal value)

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

Effective independence number

Effective independence number (setting ε to optimal value)

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

Regret bound of Exp3-IXt

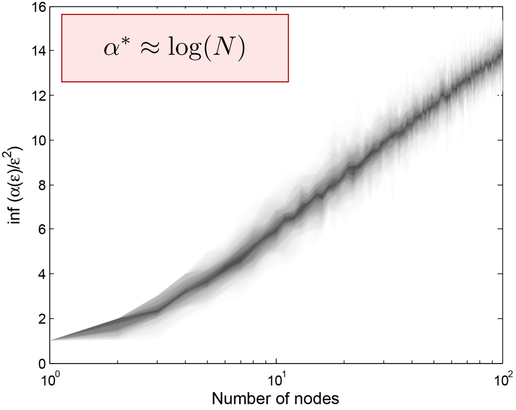
For optimal value of ε, EXP3-IXT has regret bound of

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

Effective independence number

α^* can be much smaller then N

Example: complete graph with uniformly distributed weights

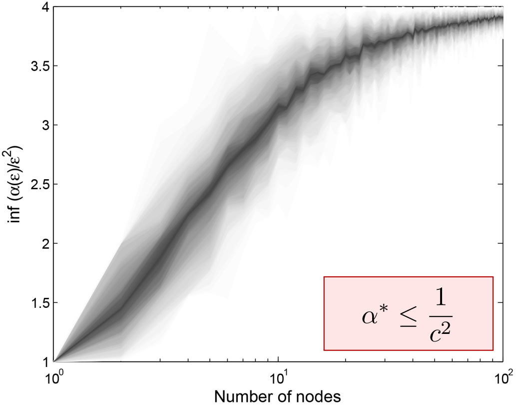


$U(0,1)$ weights

Effective independence number

α^* can be much smaller then N

Example: complete graph with uniformly distributed weights



$U(c, 1)$ weights

Effective independence number

Question: How do we set ε?

Effective independence number

Question: How do we set ε?

Answer: Finding optimal ε is **hard**

- Necessary to know whole graph
- Computing $\alpha(\varepsilon)$ is NP hard

Loss estimates

Third attempt - Exp3-WIX algorithm

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

Loss estimates

Third attempt - Exp3-WIX algorithm

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

Loss estimates

Third attempt - Exp3-WIX algorithm

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

Properties of the estimates

- **Unbiased estimates**
- **Smaller variance**. Multiplying by $s_{t,(I_t,i)}$:
  - Pulls estimate towards zero when weight is small
  - Decreasing variance of noise

Weighted graph

Third attemp - regret bound of Exp3-WIX algorithm

Theorem (Regret bound of Exp3-WIX)

When tuned properly, Exp3-WIX *algorithm has regret bound of*

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

Weighted graph

Third attemp - regret bound of EXP3-WIX algorithm

Theorem (Regret bound of EXP3-WIX)

When tuned properly, EXP3-WIX *algorithm has regret bound of*

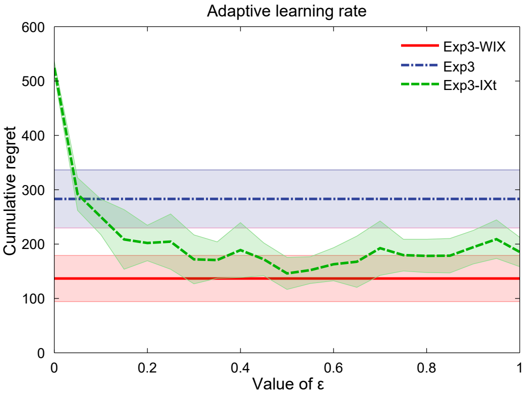
$$\mathbb{E}\left[R_t\right] = \widetilde{\mathcal{O}}\left(\sqrt{\alpha^* T}\right)$$

- **Advantages** over EXP3-IXT algorithm
  - No thresholding (using all side observations)
  - Algorithm does not need to know the best ε **(NP-hard)**
  - **Regret bound of order** $\sqrt{\alpha^* T}$

Experiments

Empirical performance

- **Exp3** - basic algorithm which ignores all side observations
- **Exp3-IXt** - thresholded algorithm (needs to set ε)
- **Exp3-WIX** - proposed algorithm



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, setting of Mannor and Shamir 2011)
  - Related lower-bound (Wu et al. 2015) for a stochastic setting with Gaussian noise

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

Thank you!

Tomorrow session: **Poster 10**

Tomáš Kocák
tomas.kocak@inria.fr
sequel.lille.inria.fr