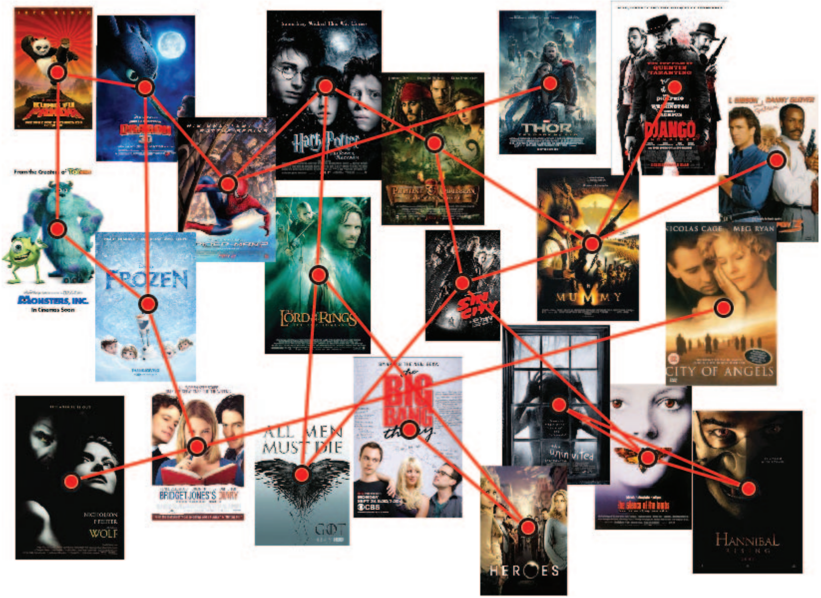
# Bandits on Graphs

*Exploiting smoothness and side observations*

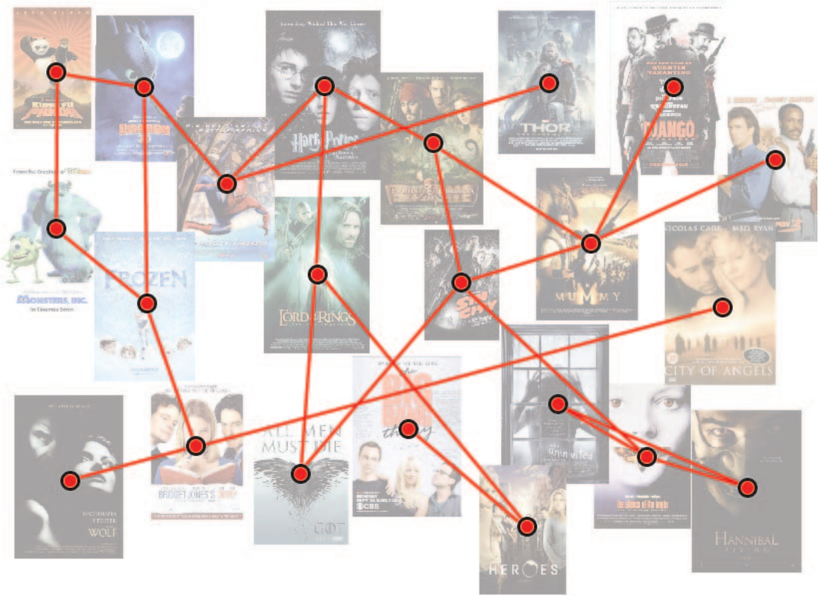
**Michal Valko** (SequeL INRIA)

*joint work with*
**Shipra Agrawal** (MSR India)
**Tomáš Kocák** (SequeL INRIA)
**Branislav Kveton** (Technicolor $\rightarrow$ Adobe)
**Rémi Munos** (SequeL INRIA/Google Deepmind)
**Gergely Neu** (SequeL INRIA)

From the Creators of
MONSTERS, INC.
In Cinemas Soon
FROZEN
Harry Potter
THOR
DJANGO
LORD OF THE RINGS
SIN CITY
MUMMY
NICOLAS CAGE MEG RYAN
CITY OF ANGELS
NICHOLSON
PFEIFFER
WOLF
BRIDGET JONES'S DIARY
ALL MEN MUST DIE
the BIG BANG theory
CBS
the uninvited
HEROES
HANNIBAL



FROZEN
MONSTERS, INC.
In Cinemas Soon
THOR
DJANGO
MUMMY
NICOLAS CAGE
MEG RYAN
CITY OF ANGELS
ALL MEN MUST DIE
BRIDGET JONES'S DIARY
WOLF
HEROES
HANNIBAL



Harry Potter
THOR
DJANGO
FROZEN
MONSTERS, INC.
LORD OF THE RINGS
NICOLAS CAGE MEG RYAN
CITY OF ANGELS
ALL MEN MUST DIE
BRIDGET JONES'S
WOLF
HEROES
HANNIBAL

# Overview

- Sequential decision making in structured settings

# Overview

- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph

# Overview

- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph
  - the graph encodes some **structural property** of the setting

# Overview

- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph
  - the graph encodes some **structural property** of the setting
  - goal: maximize the sum of the outcomes

# Overview

- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph
  - the graph encodes some **structural property** of the setting
  - goal: maximize the sum of the outcomes
  - application: recommender systems

# Overview

- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph
  - the graph encodes some **structural property** of the setting
  - goal: maximize the sum of the outcomes
  - application: recommender systems
- Exploiting **smoothness**

# Overview

- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph
  - the graph encodes some **structural property** of the setting
  - goal: maximize the sum of the outcomes
  - application: recommender systems
- Exploiting **smoothness**
  - fixed graph

# Overview

- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph
  - the graph encodes some **structural property** of the setting
  - goal: maximize the sum of the outcomes
  - application: recommender systems
- Exploiting **smoothness**
  - fixed graph
  - iid outcomes

# Overview

- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph
  - the graph encodes some **structural property** of the setting
  - goal: maximize the sum of the outcomes
  - application: recommender systems
- Exploiting **smoothness**
  - fixed graph
  - iid outcomes
  - neighboring nodes have similar outcomes

# Overview

- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph
  - the graph encodes some **structural property** of the setting
  - goal: maximize the sum of the outcomes
  - application: recommender systems
- Exploiting **smoothness**
  - fixed graph
  - iid outcomes
  - neighboring nodes have similar outcomes
- Exploiting **side observations**

# Overview

- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph
  - the graph encodes some **structural property** of the setting
  - goal: maximize the sum of the outcomes
  - application: recommender systems
- Exploiting **smoothness**
  - fixed graph
  - iid outcomes
  - neighboring nodes have similar outcomes
- Exploiting **side observations**
  - changing graph

# Overview

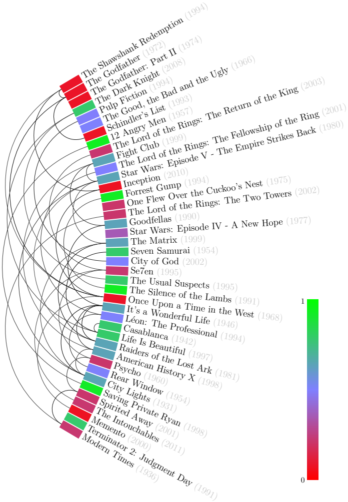
- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph
  - the graph encodes some **structural property** of the setting
  - goal: maximize the sum of the outcomes
  - application: recommender systems
- Exploiting **smoothness**
  - fixed graph
  - iid outcomes
  - neighboring nodes have similar outcomes
- Exploiting **side observations**
  - changing graph
  - non-stochastic outcomes

# Overview

- Sequential decision making in structured settings
  - we are asked to pick a node (or a few nodes) in a graph
  - the graph encodes some **structural property** of the setting
  - goal: maximize the sum of the outcomes
  - application: recommender systems
- Exploiting **smoothness**
  - fixed graph
  - iid outcomes
  - neighboring nodes have similar outcomes
- Exploiting **side observations**
  - changing graph
  - non-stochastic outcomes
  - side observations

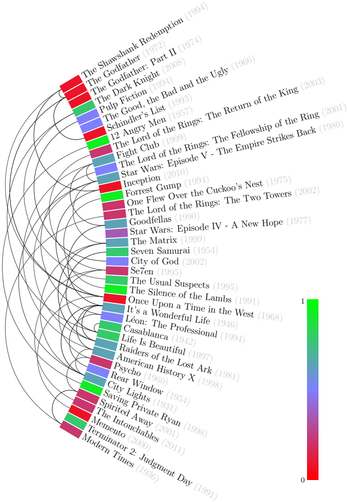
**Movie recommendation:** (in each time step)

- Recommend movies to a **single user**.



**Movie recommendation:** (in each time step)

- Recommend movies to a **single user**.
- Good prediction after a few steps ($T \ll N$).

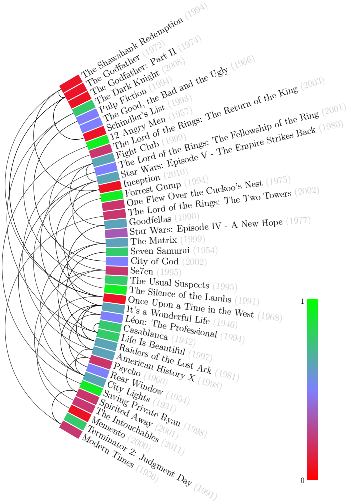


**Movie recommendation:** (in each time step)

- Recommend movies to a **single user**.
- Good prediction after a few steps ($T \ll N$).

**Goal:**

- Maximize overall reward (sum of ratings).



**Movie recommendation:** (in each time step)

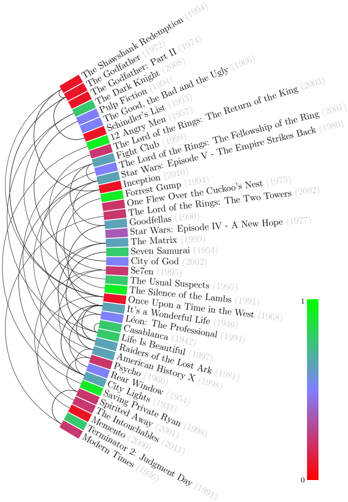
- Recommend movies to a **single user**.
- Good prediction after a few steps ($T \ll N$).

**Goal:**

- Maximize overall reward (sum of ratings).

**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.



# Smooth graph function

- Graph $G$ with vertex set $V(G) = \{1, \ldots, N\}$ and edge set $E(G)$.

# Smooth graph function

- Graph $G$ with vertex set $V(G) = \{1, \ldots, N\}$ and edge set $E(G)$.
- $f_1, \ldots, f_N$: Values of the function on the vertices of the graph.

# Smooth graph function

- Graph $G$ with vertex set $V(G) = \{1, \ldots, N\}$ and edge set $E(G)$.
- $f_1, \ldots, f_N$: Values of the function on the vertices of the graph.
- $w_{i,j}$: Weight of the edge connecting nodes $i$ and $j$.

# Smooth graph function

- Graph $G$ with vertex set $V(G) = \{1, \ldots, N\}$ and edge set $E(G)$.
- $f_1, \ldots, f_N$: Values of the function on the vertices of the graph.
- $w_{i,j}$: Weight of the edge connecting nodes $i$ and $j$.

# Smooth graph function

- Graph $G$ with vertex set $V(G) = \{1, \ldots, N\}$ and edge set $E(G)$.
- $f_1, \ldots, f_N$: Values of the function on the vertices of the graph.
- $w_{i,j}$: Weight of the edge connecting nodes $i$ and $j$.

**Smoothness of the function:**

$$S_G(f) = \frac{1}{2} \sum_{i,j \leq N} w_{i,j}(f_i - f_j)^2$$

Smaller value of $S_G(f)$, smoother the function $f$ is.

# Smooth graph function

- Graph $G$ with vertex set $V(G) = \{1, \dots, N\}$ and edge set $E(G)$.
- $f_1, \dots, f_N$: Values of the function on the vertices of the graph.
- $w_{i,j}$: Weight of the edge connecting nodes $i$ and $j$.

**Smoothness of the function:**

$$S_G(f) = \frac{1}{2} \sum_{i,j \leq N} w_{i,j}(f_i - f_j)^2$$

Smaller value of $S_G(f)$, smoother the function $f$ is.
**Examples:**

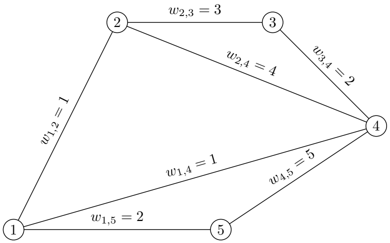
- **Complete graph:** Only constant function has smoothness 0.
- **Edgeless graph:** Every function has smoothness 0.
- **Constant function:** Smoothness 0 for every graph.

# Graph Laplacian

- $\mathcal{W}$: $N \times N$ matrix of the edge weights $w_{i,j}$.
- $\mathcal{D}$: Diagonal matrix with the entries $d_i = \sum_j w_{i,j}$.
- $\mathcal{L} = \mathcal{D} - \mathcal{W}$: Graph Laplacian.
  - Positive semidefinite matrix.
  - Diagonally dominant matrix.

**Example:**

$$\mathcal{L} = \begin{pmatrix} 4 & -1 & 0 & -1 & -2 \\ -1 & 8 & -3 & -4 & 0 \\ 0 & -3 & 5 & -2 & 0 \\ -1 & -4 & -2 & 12 & -5 \\ -2 & 0 & 0 & -5 & 7 \end{pmatrix}$$



# Smoothness of the function and Laplacian

- $\boldsymbol{f} = (f_1, \ldots, f_N)^\intercal$: Vector of function values.

# Smoothness of the function and Laplacian

- $\boldsymbol{f} = (f_1, \ldots, f_N)^\mathsf{T}$: Vector of function values.
- Let $\mathcal{L} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\mathsf{T}$ be the eigendecomposition of the Laplacian.

# Smoothness of the function and Laplacian

- $\boldsymbol{f} = (f_1, \dots, f_N)^\mathsf{T}$: Vector of function values.
- Let $\mathcal{L} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\mathsf{T}$ be the eigendecomposition of the Laplacian.
  - Diagonal matrix $\mathbf{\Lambda}$ whose diagonal entries are eigenvalues of $\mathcal{L}$.

# Smoothness of the function and Laplacian

- $\boldsymbol{f} = (f_1, \ldots, f_N)^\intercal$: Vector of function values.
- Let $\mathcal{L} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\intercal$ be the eigendecomposition of the Laplacian.
  - Diagonal matrix $\boldsymbol{\Lambda}$ whose diagonal entries are eigenvalues of $\mathcal{L}$.
  - Columns of $\mathbf{Q}$ are eigenvectors of $\mathcal{L}$.

# Smoothness of the function and Laplacian

- $\boldsymbol{f} = (f_1, \dots, f_N)^\mathsf{T}$: Vector of function values.
- Let $\mathcal{L} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}$ be the eigendecomposition of the Laplacian.
  - Diagonal matrix $\boldsymbol{\Lambda}$ whose diagonal entries are eigenvalues of $\mathcal{L}$.
  - Columns of $\mathbf{Q}$ are eigenvectors of $\mathcal{L}$.
  - Columns of $\mathbf{Q}$ form a basis.

# Smoothness of the function and Laplacian

- $\boldsymbol{f} = (f_1, \ldots, f_N)^\mathsf{T}$: Vector of function values.
- Let $\mathcal{L} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}$ be the eigendecomposition of the Laplacian.
  - Diagonal matrix $\boldsymbol{\Lambda}$ whose diagonal entries are eigenvalues of $\mathcal{L}$.
  - Columns of $\mathbf{Q}$ are eigenvectors of $\mathcal{L}$.
  - Columns of $\mathbf{Q}$ form a basis.
- $\boldsymbol{\alpha}^*$: Unique vector such that $\mathbf{Q}\boldsymbol{\alpha}^* = \boldsymbol{f}$ Note: $\mathbf{Q}^\mathsf{T}\boldsymbol{f} = \boldsymbol{\alpha}^*$

# Smoothness of the function and Laplacian

- $\boldsymbol{f} = (f_1, \ldots, f_N)^\intercal$: Vector of function values.
- Let $\mathcal{L} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\intercal$ be the eigendecomposition of the Laplacian.
  - Diagonal matrix $\boldsymbol{\Lambda}$ whose diagonal entries are eigenvalues of $\mathcal{L}$.
  - Columns of $\mathbf{Q}$ are eigenvectors of $\mathcal{L}$.
  - Columns of $\mathbf{Q}$ form a basis.
- $\boldsymbol{\alpha}^*$: Unique vector such that $\mathbf{Q}\boldsymbol{\alpha}^* = \boldsymbol{f}$ Note: $\mathbf{Q}^\intercal \boldsymbol{f} = \boldsymbol{\alpha}^*$

# Smoothness of the function and Laplacian

- $\boldsymbol{f} = (f_1, \ldots, f_N)^\mathsf{T}$: Vector of function values.
- Let $\mathcal{L} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}$ be the eigendecomposition of the Laplacian.
  - Diagonal matrix $\boldsymbol{\Lambda}$ whose diagonal entries are eigenvalues of $\mathcal{L}$.
  - Columns of $\mathbf{Q}$ are eigenvectors of $\mathcal{L}$.
  - Columns of $\mathbf{Q}$ form a basis.
- $\boldsymbol{\alpha}^*$: Unique vector such that $\mathbf{Q}\boldsymbol{\alpha}^* = \boldsymbol{f}$    Note: $\mathbf{Q}^\mathsf{T}\boldsymbol{f} = \boldsymbol{\alpha}^*$

$$S_G(f) = \boldsymbol{f}^\mathsf{T}\mathcal{L}\boldsymbol{f} = \boldsymbol{f}^\mathsf{T}\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}\boldsymbol{f} = \boldsymbol{\alpha}^{*\mathsf{T}}\boldsymbol{\Lambda}\boldsymbol{\alpha}^* = \|\boldsymbol{\alpha}^*\|_{\boldsymbol{\Lambda}}^2 = \sum_{i=1}^{N} \lambda_i (\alpha_i^*)^2$$

**Smoothness and regularization:** Small value of

**(a)** $S_G(f)$   **(b)** $\boldsymbol{\Lambda}$ norm of $\boldsymbol{\alpha}^*$   **(c)** $\alpha_i^*$ for large $\lambda_i$

# Setting

**Learning setting for a bandit algorithm $\pi$**

- In each time $t$ step choose a node $\pi(t)$.

# Setting

**Learning setting for a bandit algorithm $\pi$**

- 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 $\pi(t)$.

# Setting

**Learning setting for a bandit algorithm $\pi$**

- 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: $\mathbf{x}_{\pi(t)}^{\mathsf{T}} \boldsymbol{\alpha}^* = f_{\pi(t)}$
  - $\varepsilon_t$ 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't we just use *linear bandits*?

# Solutions

- **Linear bandit algorithms** (Existing solutions)
  - **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!**

# Solutions

- **Linear bandit algorithms** (Existing solutions)
  - **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** (Our solutions)
  - **SpectralUCB** (Valko et al., ICML 2014)
    - Regret bound $\approx d\sqrt{T \ln T}$
  - **SpectralTS** (Kocák et al., AAAI 2014)
    - Regret bound $\approx d\sqrt{T \ln N}$

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

# Solutions

- **Linear bandit algorithms** (Existing solutions)
  - **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** (Our solutions)
  - **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 $D$.

# Effective dimension

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

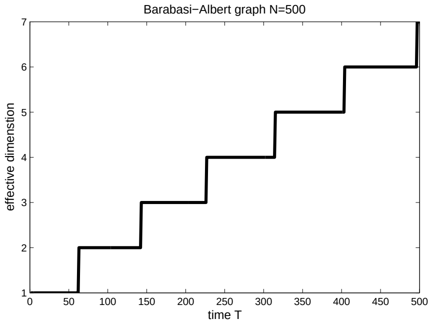
$$(d-1)\lambda_d \leq \frac{T}{\log(1 + T/\lambda)}.$$

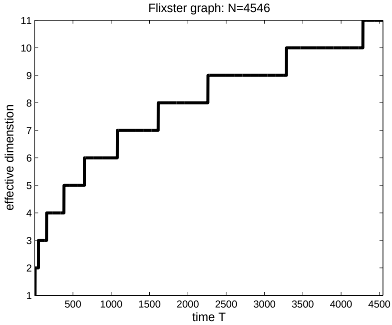
- Function of time horizon and graph properties
- $\lambda_i$ : $i$-th smallest eigenvalue of $\mathbf{\Lambda}$.
- $\lambda$: Regularization parameter of the algorithm.

**Properties:**

- $d$ is small when the coefficients $\lambda_i$ 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.

# Effective dimension vs. Ambient dimension

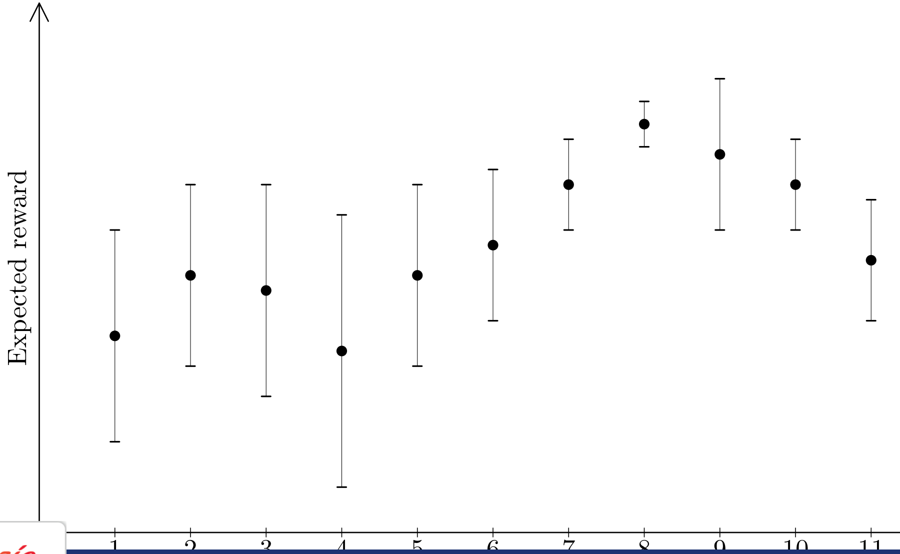


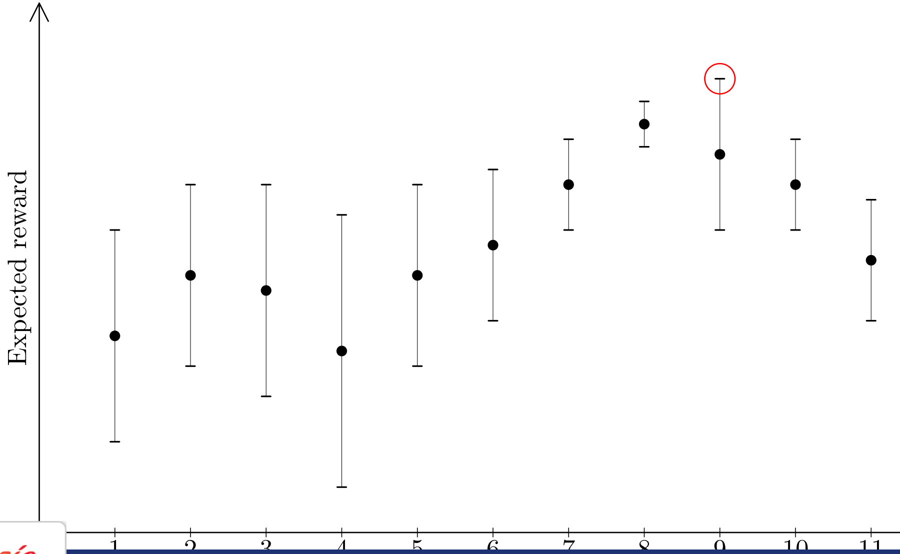

$$d \ll D$$

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

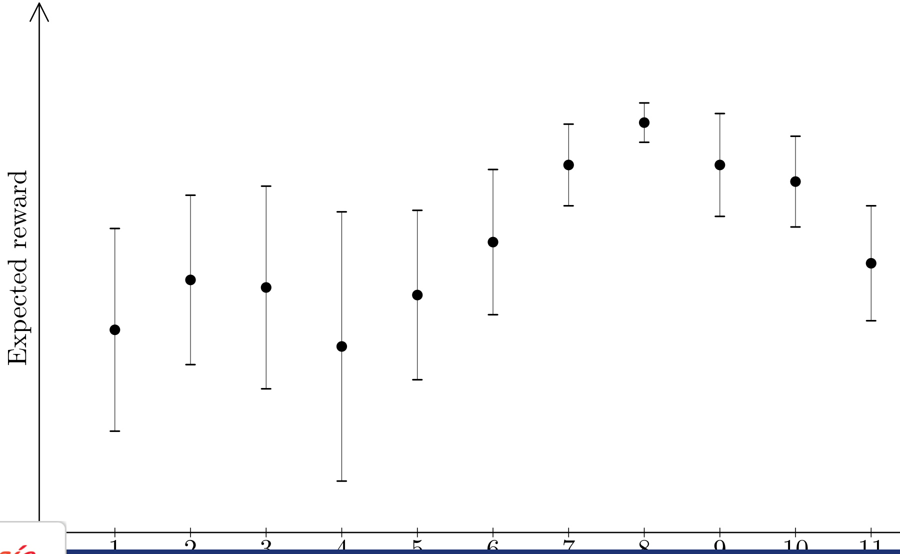
# UCB style algorithms: **Estimate**



# UCB style algorithms: **Sample**



# UCB style algorithms: **Estimate** . . .



# SpectralUCB

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

# SpectralUCB

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

# SpectralUCB

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

# SpectralUCB

**Input:**

$N$, $T$, $\{\mathbf{\Lambda}_{\mathcal{L}}, \mathbf{Q}\}$, $\lambda$, $\delta$, $R$, $C$ $\mathcal{L}$

**Run:**

$\mathbf{\Lambda} \leftarrow \mathbf{\Lambda}_{\mathcal{L}} + \lambda \mathbf{I}$

$d \leftarrow \max\{d : (d-1)\lambda_d \leq T / \ln(1 + T/\lambda)\}$

**for** $t = 1$ **to** $T$ **do**

Update the basis coefficients $\hat{\boldsymbol{\alpha}}$:

$\mathbf{X}_t \leftarrow [\mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(t-1)}]^\mathsf{T}$

$\mathbf{r} \leftarrow [r_1, \ldots, r_{t-1}]^\mathsf{T}$

$\mathbf{V}_t \leftarrow \mathbf{X}_t \mathbf{X}_t^\mathsf{T} + \mathbf{\Lambda}$

$\hat{\boldsymbol{\alpha}}_t \leftarrow \mathbf{V}_t^{-1} \mathbf{X}_t^\mathsf{T} \mathbf{r}$

$c_t \leftarrow 2R\sqrt{d \ln(1 + t/\lambda) + 2\ln(1/\delta)} + C$

$\pi(t) \leftarrow \arg\max_a \left( \mathbf{x}_a^\mathsf{T} \hat{\boldsymbol{\alpha}} + c_t \|\mathbf{x}_a\|_{\mathbf{V}_t^{-1}} \right)$

Observe the reward $r_t$

**end for**

# SpectralUCB

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

# SpectralUCB

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

# SpectralUCB regret bound

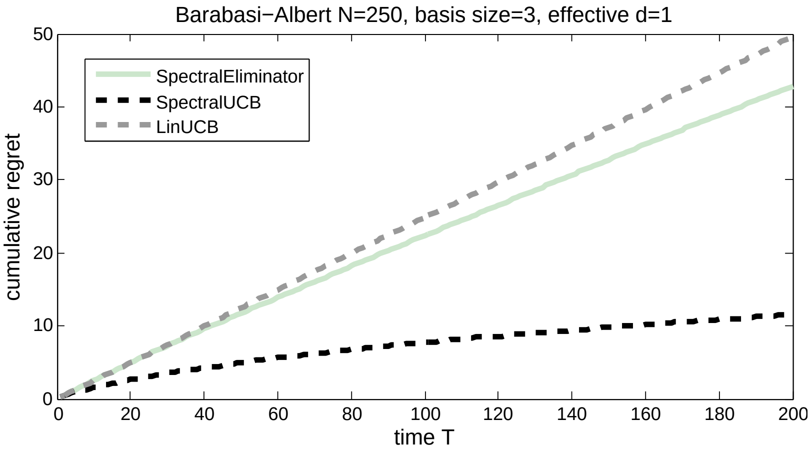
- $d$: Effective dimension.
- $\lambda$: Minimal eigenvalue of $\mathbf{\Lambda} = \mathbf{\Lambda}_{\mathcal{L}} + \lambda \mathbf{I}$.
- $C$: Smoothness upper bound, $\|\boldsymbol{\alpha}^*\|_{\mathbf{\Lambda}} \leq C$.
- $\mathbf{x}_i^\mathsf{T} \boldsymbol{\alpha}^* \in [-1, 1]$ for all $i$.

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

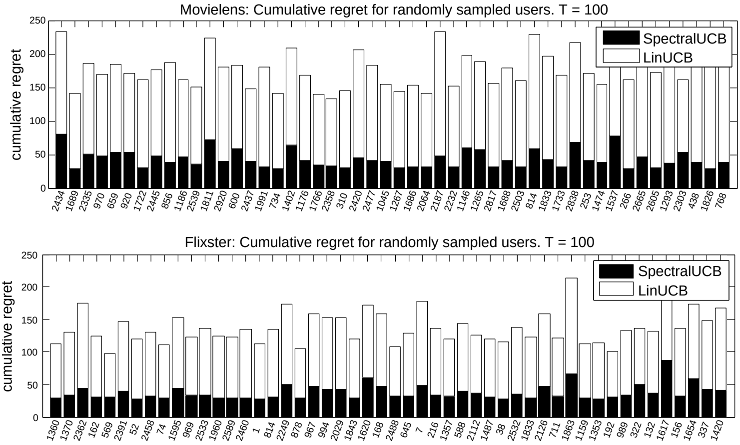
$$R_T \leq \left( 8R\sqrt{d \ln \frac{\lambda + T}{\lambda} + 2 \ln \frac{1}{\delta}} + 4C + 4 \right) \sqrt{dT \ln \frac{\lambda + T}{\lambda}}.$$

$$R_T \approx d\sqrt{T \ln T}$$

# Synthetic experiment

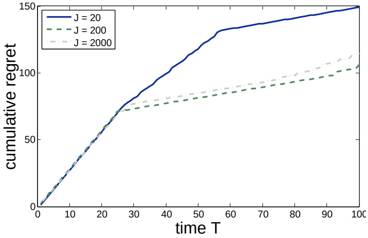


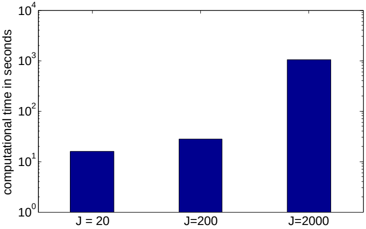
# Real world experiment



# Improving the running time: reduced eigenbasis

- **Reduced basis:** We only need first few eigenvectors.
- **Getting $J$ eigenvectors:** $\mathcal{O}(Jm \log m)$ time for $m$ edges
- Computationally less expensive, comparable performance.



# How to make it faster?

- UCB-style algorithms need to (re)-compute UCBs every $t$

# How to make it faster?

- UCB-style algorithms need to (re)-compute UCBs every $t$
- Can be a problem for large set of arms $\rightarrow D^2 N \rightarrow N^3$

# How to make it faster?

- UCB-style algorithms need to (re)-compute UCBs every $t$
- Can be a problem for large set of arms $\rightarrow D^2N \rightarrow N^3$
- Optimistic (UCB) approach vs. Thompson Sampling

# How to make it faster?

- UCB-style algorithms need to (re)-compute UCBs every $t$
- Can be a problem for large set of arms $\rightarrow D^2N \rightarrow N^3$
- Optimistic (UCB) approach vs. Thompson Sampling
  - Play the arm maximizing probability of being the best

# How to make it faster?

- UCB-style algorithms need to (re)-compute UCBs every $t$
- Can be a problem for large set of arms $\rightarrow D^2 N \rightarrow N^3$
- Optimistic (UCB) approach vs. Thompson Sampling
  - Play the arm maximizing probability of being the best
    - Sample $\tilde{\boldsymbol{\mu}}$ from the distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}, v^2\mathbf{B}^{-1})$

# How to make it faster?

- UCB-style algorithms need to (re)-compute UCBs every $t$
- Can be a problem for large set of arms $\rightarrow D^2 N \rightarrow N^3$
- Optimistic (UCB) approach vs. Thompson Sampling
  - Play the arm maximizing probability of being the best
    - Sample $\tilde{\boldsymbol{\mu}}$ from the distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}, v^2\mathbf{B}^{-1})$
    - Play arm which maximizes $\mathbf{b}^\mathsf{T}\tilde{\boldsymbol{\mu}}$ and observe reward

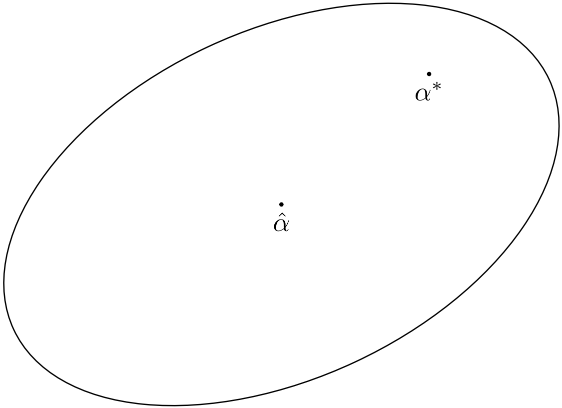
# How to make it faster?

- UCB-style algorithms need to (re)-compute UCBs every $t$
- Can be a problem for large set of arms $\rightarrow D^2 N \rightarrow N^3$
- Optimistic (UCB) approach vs. Thompson Sampling
  - Play the arm maximizing probability of being the best
    - Sample $\tilde{\boldsymbol{\mu}}$ from the distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}, v^2\mathbf{B}^{-1})$
    - Play arm which maximizes $\mathbf{b}^\mathsf{T}\tilde{\boldsymbol{\mu}}$ and observe reward
  - Compute posterior distribution according to reward received

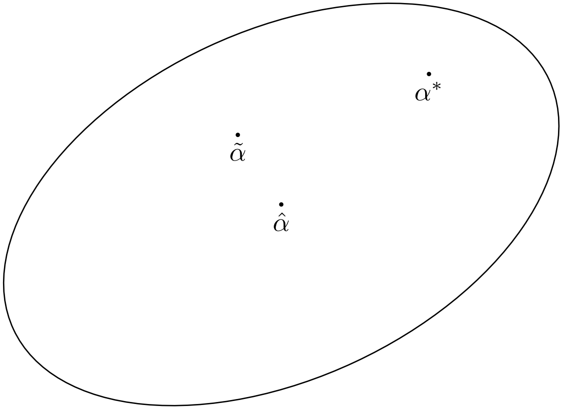
# How to make it faster?

- UCB-style algorithms need to (re)-compute UCBs every $t$
- Can be a problem for large set of arms $\rightarrow D^2N \rightarrow N^3$
- Optimistic (UCB) approach vs. Thompson Sampling
  - Play the arm maximizing probability of being the best
    - Sample $\tilde{\boldsymbol{\mu}}$ from the distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}, v^2\mathbf{B}^{-1})$
    - Play arm which maximizes $\mathbf{b}^\mathsf{T}\tilde{\boldsymbol{\mu}}$ and observe reward
  - Compute posterior distribution according to reward received
- Only requires $D^2 + DN \rightarrow N^2$ per step update

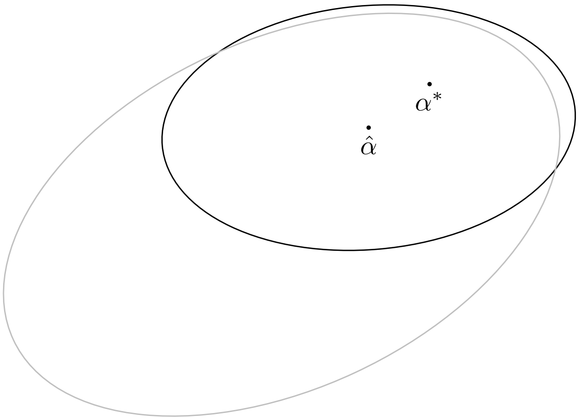
# Thomson Sampling: **Estimate**



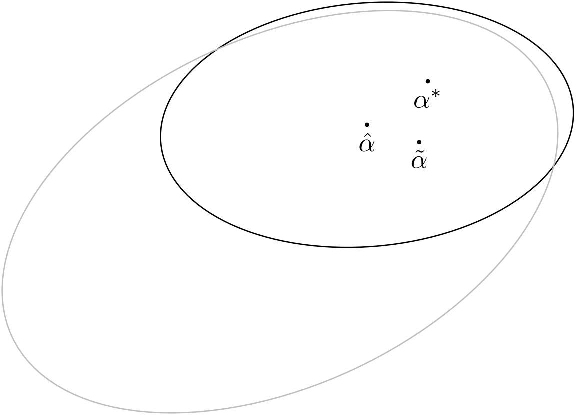
# Thomson Sampling: **Sample**



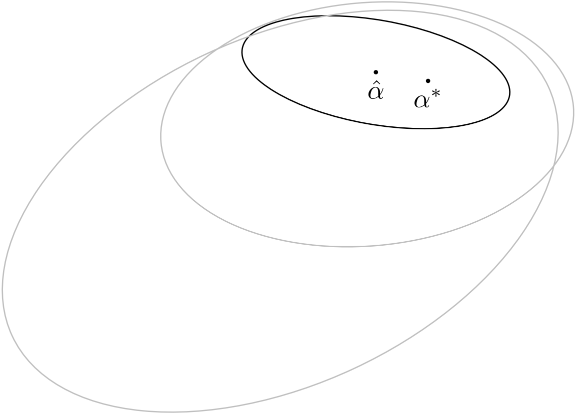
# Thomson Sampling: **Estimate**



# Thomson Sampling: **Sample**



# Thomson Sampling: **Estimate** ...



# SpectralTS algorithm

**Input:**
    $N$, $T$, $\{\mathbf{\Lambda}_{\mathcal{L}}, \mathbf{Q}\}$, $\lambda$, $\delta$, $R$, $C$
**Initialization:**
    $v = R\sqrt{6d\log((\lambda + T)/\delta\lambda)} + C$
    $\hat{\boldsymbol{\alpha}} = 0_N$
    $\boldsymbol{f} = 0_N$
    $\mathbf{V} = \mathbf{\Lambda}_{\mathcal{L}} + \lambda\mathbf{I}_N$
**Run:**
**for** $t = 1$ **to** $T$ **do**
    Sample $\tilde{\boldsymbol{\alpha}} \sim \mathcal{N}(\hat{\boldsymbol{\alpha}}, v^2\mathbf{V}^{-1})$
    $\pi(t) \leftarrow \arg\max_a \mathbf{x}_a^{\mathsf{T}}\tilde{\boldsymbol{\alpha}}$
    Observe a noisy reward $r(t) = \mathbf{x}_{\pi(t)}^{\mathsf{T}}\boldsymbol{\alpha}^* + \varepsilon_t$
    $\boldsymbol{f} \leftarrow \boldsymbol{f} + \mathbf{x}_{\pi(t)}r(t)$
    Update $\mathbf{V} \leftarrow \mathbf{V} + \mathbf{x}_{\pi(t)}\mathbf{x}_{\pi(t)}^{\mathsf{T}}$
    Update $\hat{\boldsymbol{\alpha}} \leftarrow \mathbf{V}^{-1}\boldsymbol{f}$
**end for**

# SpectralTS algorithm

**Input:**
  $N$, $T$, $\{\boldsymbol{\Lambda}_{\mathcal{L}}, \mathbf{Q}\}$, $\lambda$, $\delta$, $R$, $C$
**Initialization:**
  $v = R\sqrt{6d\log((\lambda + T)/\delta\lambda)} + C$
  $\hat{\boldsymbol{\alpha}} = 0_N$
  $\boldsymbol{f} = 0_N$
  $\mathbf{V} = \boldsymbol{\Lambda}_{\mathcal{L}} + \lambda\mathbf{I}_N$
**Run:**
**for** $t = 1$ **to** $T$ **do**
  Sample $\tilde{\boldsymbol{\alpha}} \sim \mathcal{N}(\hat{\boldsymbol{\alpha}}, v^2\mathbf{V}^{-1})$
  $\pi(t) \leftarrow \arg\max_a \mathbf{x}_a^{\mathsf{T}}\tilde{\boldsymbol{\alpha}}$
  Observe a noisy reward $r(t) = \mathbf{x}_{\pi(t)}^{\mathsf{T}}\boldsymbol{\alpha}^* + \varepsilon_t$
  $\boldsymbol{f} \leftarrow \boldsymbol{f} + \mathbf{x}_{\pi(t)}r(t)$
  Update $\mathbf{V} \leftarrow \mathbf{V} + \mathbf{x}_{\pi(t)}\mathbf{x}_{\pi(t)}^{\mathsf{T}}$
  Update $\hat{\boldsymbol{\alpha}} \leftarrow \mathbf{V}^{-1}\boldsymbol{f}$
**end for**

# SpectralTS algorithm

**Input:**
$N$, $T$, $\{\mathbf{\Lambda}_\mathcal{L}, \mathbf{Q}\}$, $\lambda$, $\delta$, $R$, $C$
**Initialization:**
$v = R\sqrt{6d\log((\lambda+T)/\delta\lambda)} + C$
$\hat{\boldsymbol{\alpha}} = 0_N$
$\boldsymbol{f} = 0_N$
$\mathbf{V} = \mathbf{\Lambda}_\mathcal{L} + \lambda\mathbf{I}_N$
**Run:**
**for** $t = 1$ **to** $T$ **do**
Sample $\tilde{\boldsymbol{\alpha}} \sim \mathcal{N}(\hat{\boldsymbol{\alpha}}, v^2\mathbf{V}^{-1})$
$\pi(t) \leftarrow \arg\max_a \mathbf{x}_a^\mathsf{T}\tilde{\boldsymbol{\alpha}}$
Observe a noisy reward $r(t) = \mathbf{x}_{\pi(t)}^\mathsf{T}\boldsymbol{\alpha}^* + \varepsilon_t$
$\boldsymbol{f} \leftarrow \boldsymbol{f} + \mathbf{x}_{\pi(t)} r(t)$
Update $\mathbf{V} \leftarrow \mathbf{V} + \mathbf{x}_{\pi(t)}\mathbf{x}_{\pi(t)}^\mathsf{T}$
Update $\hat{\boldsymbol{\alpha}} \leftarrow \mathbf{V}^{-1}\boldsymbol{f}$
**end for**

# SpectralTS algorithm

**Input:**
    $N$, $T$, $\{\mathbf{\Lambda}_{\mathcal{L}}, \mathbf{Q}\}$, $\lambda$, $\delta$, $R$, $C$
**Initialization:**
    $v = R\sqrt{6d\log((\lambda + T)/\delta\lambda)} + C$
    $\hat{\boldsymbol{\alpha}} = 0_N$
    $\boldsymbol{f} = 0_N$
    $\mathbf{V} = \mathbf{\Lambda}_{\mathcal{L}} + \lambda \mathbf{I}_N$
**Run:**
**for** $t = 1$ **to** $T$ **do**
    Sample $\tilde{\boldsymbol{\alpha}} \sim \mathcal{N}(\hat{\boldsymbol{\alpha}}, v^2\mathbf{V}^{-1})$
    $\pi(t) \leftarrow \arg\max_a \mathbf{x}_a^{\mathsf{T}}\tilde{\boldsymbol{\alpha}}$
    Observe a noisy reward $r(t) = \mathbf{x}_{\pi(t)}^{\mathsf{T}}\boldsymbol{\alpha}^* + \varepsilon_t$
    $\boldsymbol{f} \leftarrow \boldsymbol{f} + \mathbf{x}_{\pi(t)} r(t)$
    Update $\mathbf{V} \leftarrow \mathbf{V} + \mathbf{x}_{\pi(t)}\mathbf{x}_{\pi(t)}^{\mathsf{T}}$
    Update $\hat{\boldsymbol{\alpha}} \leftarrow \mathbf{V}^{-1}\boldsymbol{f}$
**end for**

# SpectralTS algorithm

**Input:**
$N$, $T$, $\{\boldsymbol{\Lambda}_{\mathcal{L}}, \mathbf{Q}\}$, $\lambda$, $\delta$, $R$, $C$
**Initialization:**
$v = R\sqrt{6d\log((\lambda + T)/\delta\lambda)} + C$
$\hat{\boldsymbol{\alpha}} = 0_N$
$\boldsymbol{f} = 0_N$
$\mathbf{V} = \boldsymbol{\Lambda}_{\mathcal{L}} + \lambda \mathbf{I}_N$
**Run:**
**for** $t = 1$ **to** $T$ **do**
Sample $\tilde{\boldsymbol{\alpha}} \sim \mathcal{N}(\hat{\boldsymbol{\alpha}}, v^2\mathbf{V}^{-1})$
$\pi(t) \leftarrow \arg\max_a \mathbf{x}_a^{\mathsf{T}}\tilde{\boldsymbol{\alpha}}$
Observe a noisy reward $r(t) = \mathbf{x}_{\pi(t)}^{\mathsf{T}}\boldsymbol{\alpha}^* + \varepsilon_t$
$\boldsymbol{f} \leftarrow \boldsymbol{f} + \mathbf{x}_{\pi(t)} r(t)$
Update $\mathbf{V} \leftarrow \mathbf{V} + \mathbf{x}_{\pi(t)}\mathbf{x}_{\pi(t)}^{\mathsf{T}}$
Update $\hat{\boldsymbol{\alpha}} \leftarrow \mathbf{V}^{-1}\boldsymbol{f}$
**end for**

# SpectralTS regret bound

- $d$: Effective dimension.
- $\lambda$: Minimal eigenvalue of $\mathbf{\Lambda} = \mathbf{\Lambda}_{\mathcal{L}} + \lambda\mathbf{I}$.
- $C$: 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 **SpectralTS** is with probability $1 - \delta$ bounded as

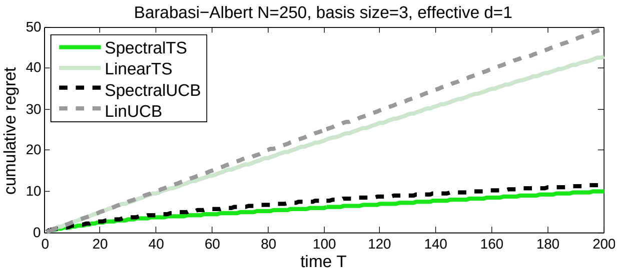
$$\mathcal{R}_T \leq \frac{11g}{p}\sqrt{\frac{4+4\lambda}{\lambda}dT\log\frac{\lambda+T}{\lambda}} + \frac{1}{T} + \frac{g}{p}\left(\frac{11}{\sqrt{\lambda}}+2\right)\sqrt{2T\log\frac{2}{\delta}},$$

where $p = 1/(4e\sqrt{\pi})$ and

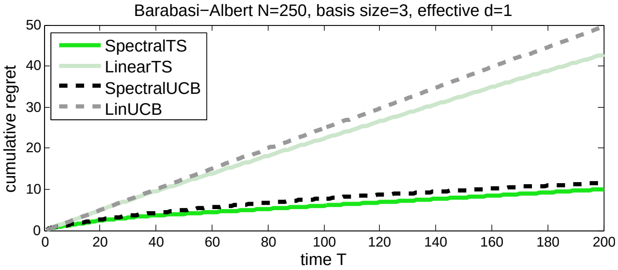
$$g = \sqrt{4\log TN}\left(R\sqrt{6d\log\left(\frac{\lambda+T}{\delta\lambda}\right)} + C\right) + R\sqrt{2d\log\left(\frac{(\lambda+T)T^2}{\delta\lambda}\right)} + C.$$

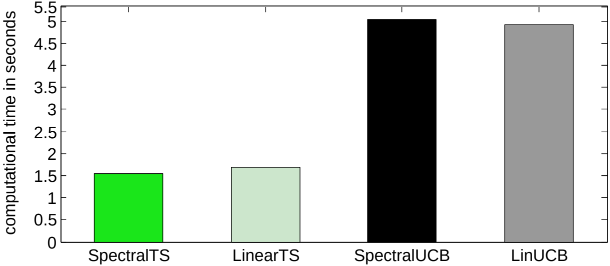
$$R_T \approx d\sqrt{T\log N}$$

# Synthetic experiment



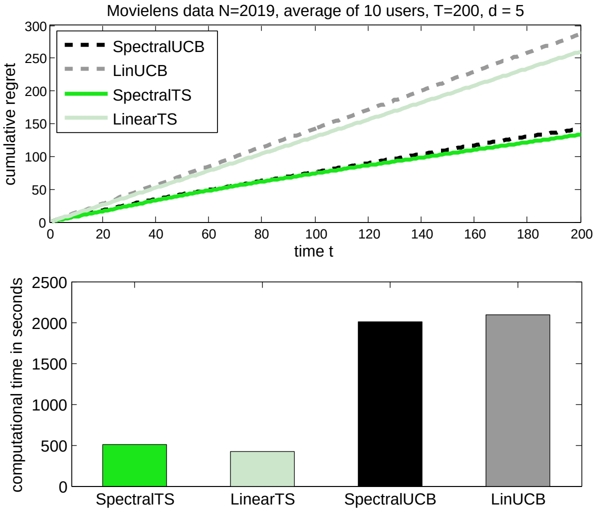
# Synthetic experiment



# Real world experiment

MovieLens dataset of 6k users who rated one million movies.



# Spectral Bandits Summary

- New spectral bandit setting **(for smooth graph functions)**.

# Spectral Bandits Summary

- New spectral bandit setting **(for smooth graph functions)**.
- **SpectralUCB**
  - Regret bound $\approx d\sqrt{T \ln T}$

# Spectral Bandits Summary

- New spectral bandit setting **(for smooth graph functions)**.
- **SpectralUCB**
  - Regret bound $\approx d\sqrt{T \ln T}$
- **SpectralTS**
  - Regret bound $\approx d\sqrt{T \ln N}$
  - Computationally more efficient.

# Spectral Bandits Summary

- New spectral bandit setting **(for smooth graph functions)**.
- **SpectralUCB**
  - Regret bound $\approx d\sqrt{T \ln T}$
- **SpectralTS**
  - Regret bound $\approx d\sqrt{T \ln N}$
  - Computationally more efficient.
- **SpectralEliminator**
  - Regret bound $\approx \sqrt{dT \ln T}$
  - Side result: **LinearEliminator** with $\mathcal{O}(\sqrt{DT \ln T})$ regret for (contextual) linear bandits.

# Spectral Bandits Summary

- New spectral bandit setting **(for smooth graph functions)**.
- **SpectralUCB**
  - Regret bound $\approx d\sqrt{T \ln T}$
- **SpectralTS**
  - Regret bound $\approx d\sqrt{T \ln N}$
  - Computationally more efficient.
- **SpectralEliminator**
  - Regret bound $\approx \sqrt{dT \ln T}$
  - Side result: **LinearEliminator** with $\mathcal{O}(\sqrt{DT \ln T})$ regret for (contextual) linear bandits.
- Bounds scale with **effective dimension** $d \ll D$.

# Spectral Bandits Summary

- New spectral bandit setting **(for smooth graph functions)**.
- **SpectralUCB**
  - Regret bound $\approx d\sqrt{T \ln T}$
- **SpectralTS**
  - Regret bound $\approx d\sqrt{T \ln N}$
  - Computationally more efficient.
- **SpectralEliminator**
  - Regret bound $\approx \sqrt{dT \ln T}$
  - Side result: **LinearEliminator** with $\mathcal{O}(\sqrt{DT \ln T})$ regret for (contextual) linear bandits.
- Bounds scale with **effective dimension** $d \ll D$.

# Exploiting side observations

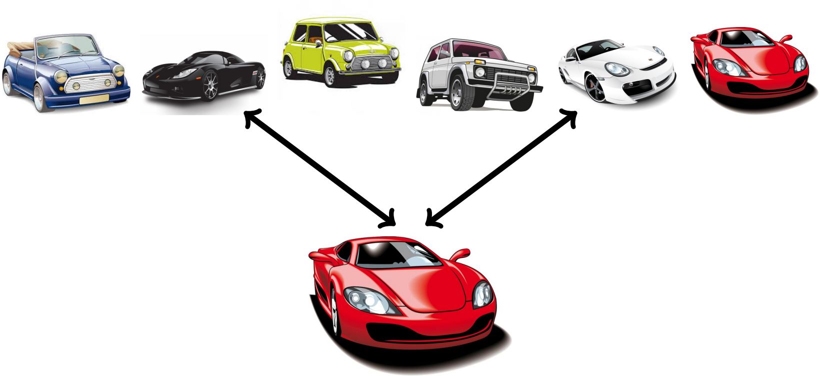
## Example 1: undirected observations

# Exploiting side observations

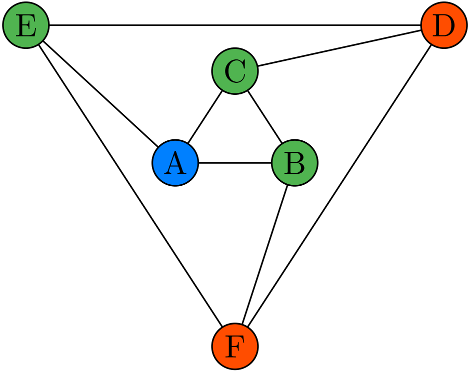
## Example 1: undirected observations

# Exploiting side observations

## Example 1: undirected observations



## Example 1: Graph Representation



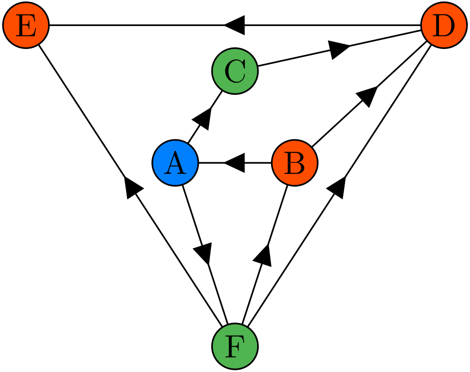
# Example 2: Directed observation




# Example 2: Directed observation


# Example 2: Directed observation

## Example 2



## Learning setting

**In each time step** $t = 1, \ldots, T$

- **Environment (adversary):**
  - Privately assigns losses to actions
  - Generates an observation graph

## Learning setting

**In each time step** $t = 1, \ldots, T$

- **Environment (adversary):**
  - Privately assigns losses to actions
  - Generates an observation graph
    - Undirected / Directed

## Learning setting

**In each time step** $t = 1, \ldots, T$

- **Environment (adversary):**
  - Privately assigns losses to actions
  - Generates an observation graph
    - Undirected / Directed
    - Disclosed / Not disclosed

## Learning setting

**In each time step** $t = 1, \ldots, T$

- **Environment (adversary):**
  - Privately assigns losses to actions
  - Generates an observation graph
    - Undirected / Directed
    - Disclosed / Not disclosed
- **Learner:**
  - Plays action $I_t \in [N]$
  - Obtain loss $\ell_{t,I_t}$ of action played
  - Observe losses of neighbors of $I_t$

## Learning setting

**In each time step $t = 1, \ldots, T$**

- **Environment (adversary):**
  - Privately assigns losses to actions
  - Generates an observation graph
    - Undirected / Directed
    - Disclosed / Not disclosed
- **Learner:**
  - Plays action $I_t \in [N]$
  - Obtain loss $\ell_{t,I_t}$ of action played
  - Observe losses of neighbors of $I_t$
    - Graph: disclosed

## Learning setting

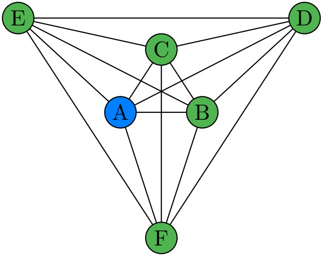
**In each time step** $t = 1, \ldots, T$

- **Environment (adversary):**
  - Privately assigns losses to actions
  - Generates an observation graph
    - Undirected / Directed
    - Disclosed / Not disclosed
- **Learner:**
  - Plays action $I_t \in [N]$
  - Obtain loss $\ell_{t,I_t}$ of action played
  - Observe losses of neighbors of $I_t$
    - Graph: disclosed
- **Performance measure:** Total expected regret

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

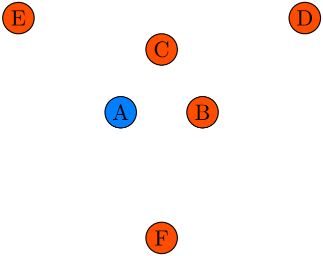
**Full Information setting**

- Pick an action (e.g. action A)
- Observe losses of all actions
- $R_T = \widetilde{\mathcal{O}}(\sqrt{T})$



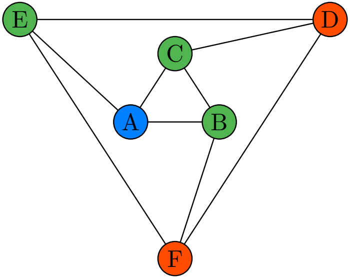

**Bandit setting**

- Pick an action (e.g. action A)
- Observe loss of a chosen action
- $R_T = \widetilde{\mathcal{O}}(\sqrt{NT})$



**Side observation (Undirected case)**

- Pick an action (e.g. action A)
- Observe losses of neighbors

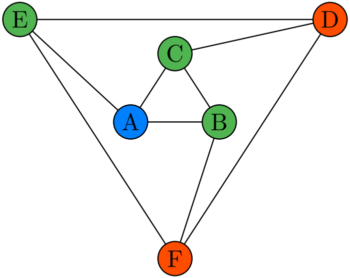


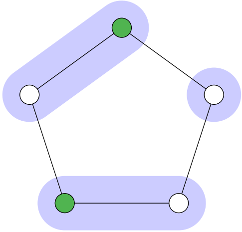
**Side observation (Undirected case)**

- Pick an action (e.g. action A)
- Observe losses of neighbors

**Mannor and Shamir** (ELP algorithm)

- Need to know graph
- Clique decomposition ($c$ cliques)
- $R_T = \widetilde{\mathcal{O}}(\sqrt{cT})$



**Side observation (Undirected case)**

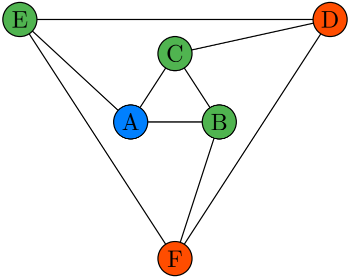
- Pick an action (e.g. action A)
- Observe losses of neighbors

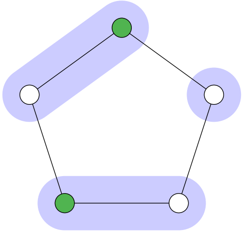
**Mannor and Shamir** (ELP algorithm)

- Need to know graph
- Clique decomposition ($c$ cliques)
- $R_T = \widetilde{\mathcal{O}}(\sqrt{cT})$

**Alon, Cesa-Bianchi, Gentile, Mansour**

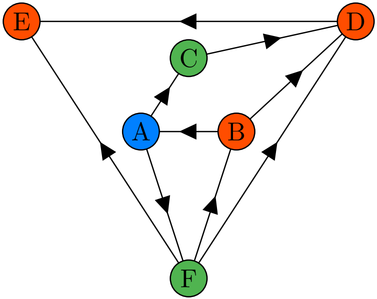
- No need to know graph
- Independence set of $\alpha$ actions
- $R_T = \widetilde{\mathcal{O}}(\sqrt{\alpha T})$



## Side observation (Directed case)

- Pick an action (e.g. action A)
- Observe losses of neighbors

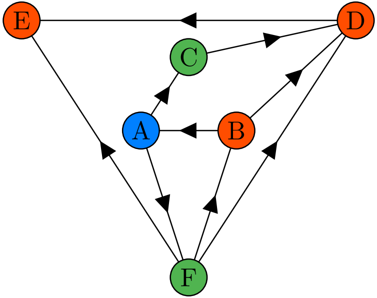


**Side observation (Directed case)**

- Pick an action (e.g. action A)
- Observe losses of neighbors

**Alon, Cesa-Bianchi, Gentile, Mansour**

- **Exp3-DOM**
- Need to know graph
- Need to find dominating set
- $R_T = \widetilde{\mathcal{O}}(\sqrt{\alpha T})$



## Side observation (Directed case)

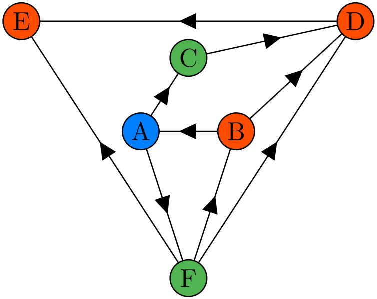
- Pick an action (e.g. action A)
- Observe losses of neighbors

**Alon, Cesa-Bianchi, Gentile, Mansour**

- **Exp3-DOM**
- Need to know graph
- Need to find dominating set
- $R_T = \widetilde{\mathcal{O}}(\sqrt{\alpha T})$

**Our solution: Exp3-IX**

- No need to know graph
- $R_T = \widetilde{\mathcal{O}}(\sqrt{\alpha T})$



# Exp3 algorithms in general

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

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

- **Play action** $I_t$ such that

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

- **Update loss estimates** (using observability graph)

# Exp3 algorithms in general

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

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

- **Play action** $I_t$ such that

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

- **Update loss estimates** (using observability graph)

**How the algorithms approach to bias variance tradeoff?**

## Bias variance tradeoff approaches

- Approach of previous algorithms – Mixing
  - Bias sampling distribution $\mathbf{p}_t$ over actions
    - $\mathbf{p}_t' = (1-\gamma)\mathbf{p}_t + \gamma \mathbf{s}_t$ – mixed distribution
    - $\mathbf{s}_t$ – probability distribution which supports exploration
  - Loss estimates $\hat{\ell}_{t,i}$ are unbiased
- Approach of our algorithm – Implicit eXploration (**IX**)
  - Bias loss estimates $\hat{\ell}_{t,i}$
    - Biased loss estimates $\implies$ biased weights
    - Biased weights $\implies$ biased probability distribution
  - No need for mixing

## Mannor and Shamir - ELP algorithm

- $\mathbb{E}[\hat{\ell}_{t,i}] = \ell_{t,i}$ – unbiased loss estimates
- $p'_{t,i} = (1-\gamma)p_{t,i} + \gamma s_{t,i}$ – bias by mixing
- $\mathbf{s}_t = \{s_{t,1}, \ldots, s_{t,N}\}$ – probability distribution over the action set

$$\mathbf{s}_t = \arg\max_{\mathbf{s}_t} \left[ \min_{j \in [N]} \left( s_{t,j} + \sum_{k \in N_{t,j}} s_{t,k} \right) \right] = \arg\max_{\mathbf{s}_t} \left[ \min_{j \in [N]} q_{t,j} \right]$$

- $q_{t,j}$ – probability that loss of $j$ is observed according to $\mathbf{s}_t$

# Mannor and Shamir - ELP algorithm

- $\mathbb{E}[\hat{\ell}_{t,i}] = \ell_{t,i}$ – unbiased loss estimates
- $p'_{t,i} = (1-\gamma)p_{t,i} + \gamma s_{t,i}$ – bias by mixing
- $\mathbf{s}_t = \{s_{t,1}, \ldots, s_{t,N}\}$ – probability distribution over the action set

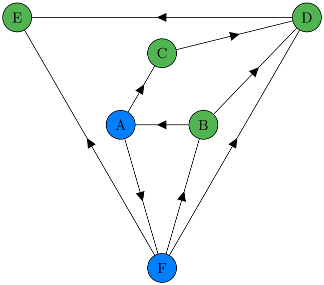
$$\mathbf{s}_t = \arg\max_{\mathbf{s}_t} \left[ \min_{j\in[N]} \left( s_{t,j} + \sum_{k\in N_{t,j}} s_{t,k} \right) \right] = \arg\max_{\mathbf{s}_t} \left[ \min_{j\in[N]} q_{t,j} \right]$$

- $q_{t,j}$ – probability that loss of $j$ is observed according to $\mathbf{s}_t$
- **Computation of $\mathbf{s}_t$**
  - Graph needs to be disclosed
  - Solving simple linear program
- Needs to know graph before playing an action
- Graphs can be only undirected

## Alon, Cesa-Bianchi, Gentile, Mansour - Exp3-DOM

- $\mathbb{E}[\hat{\ell}_{t,i}] = \ell_{t,i}$ – unbiased loss estimates
- $p'_{t,i} = (1-\gamma)p_{t,i} + \gamma s_{t,i}$ – bias by mixing
- $\mathbf{s}_t = \{s_{t,1}, \ldots, s_{t,N}\}$ – probability distribution over the action set

$$s_{t,i} = \begin{cases} \frac{1}{r} & \text{if } i \in R;\ |R| = r \\ 0 & \text{otherwise.} \end{cases}$$




- $R$ – dominating set of $r$ elements
- $\mathbf{s}_t$ – uniform distribution over $R$
- Needs to know graph beforehand
- Graphs can be directed

## Previous algorithms - loss estimates

$$\hat{\ell}_{t,i} = \begin{cases} \ell_{t,i}/o_{t,i} & \text{if } \ell_{t,i} \text{ is observed} \\ 0 & \text{otherwise.} \end{cases}$$

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

## **Previous algorithms** - loss estimates

$$\hat{\ell}_{t,i} = \begin{cases} \ell_{t,i}/o_{t,i} & \text{if } \ell_{t,i} \text{ is observed} \\ 0 & \text{otherwise.} \end{cases}$$

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

## **Exp3-IX** - loss estimates

$$\hat{\ell}_{t,i} = \begin{cases} \ell_{t,i}/(o_{t,i} + \gamma) & \text{if } \ell_{t,i} \text{ is observed} \\ 0 & \text{otherwise.} \end{cases}$$

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

- No mixing!

# Analysis of Exp3 algorithms in general

- Evolution of $W_{t+1}/W_t$

$$\frac{1}{\eta}\log\frac{W_{t+1}}{W_t} = \frac{1}{\eta}\log\left(1-\eta\sum_{i=1}^{N}p_{t,i}\hat{\ell}_{t,i}+\frac{\eta^2}{2}\sum_{i=1}^{N}p_{t,i}(\hat{\ell}_{t,i})^2\right),$$

$$\sum_{i=1}^{N}p_{t,i}\hat{\ell}_{t,i} \leq \left[\frac{\log W_t}{\eta}-\frac{\log W_{t+1}}{\eta}\right]+\frac{\eta}{2}\sum_{i=1}^{N}p_{t,i}(\hat{\ell}_{t,i})^2$$

- Taking expectation and summing over time

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N}p_{t,i}\hat{\ell}_{t,i}\right]-\mathbb{E}\left[\sum_{t=1}^{T}\hat{\ell}_{t,k}\right] \leq \mathbb{E}\left[\frac{\log N}{\eta}\right]+\mathbb{E}\left[\frac{\eta}{2}\sum_{t=1}^{T}\sum_{i=1}^{N}p_{t,i}(\hat{\ell}_{t,i})^2\right]$$

# Regret bound of Exp3-IX

$$\underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\right]}_{A} - \underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\hat{\ell}_{t,k}\right]}_{B} \leq \mathbb{E}\left[\frac{\log N}{\eta}\right] + \underbrace{\mathbb{E}\left[\frac{\eta}{2}\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2\right]}_{C}$$

**Lower bound of A** (using definition of loss estimates)

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\right] \geq \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\ell_{t,i}\right] - \mathbb{E}\left[\gamma\sum_{t=1}^{T}\sum_{i=1}^{N}\frac{p_{t,i}}{o_{t,i}+\gamma}\right]$$

**Lower bound of B** (optimistic loss estimates: $\mathbb{E}[\hat{\ell}] < \mathbb{E}[\ell]$)

$$-\mathbb{E}\left[\sum_{t=1}^{T}\hat{\ell}_{t,k}\right] \geq -\mathbb{E}\left[\sum_{t=1}^{T}\ell_{t,k}\right]$$

**Upper bound of C** (using definition of loss estimates)

$$\mathbb{E}\left[\frac{\eta}{2}\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2\right] \leq \mathbb{E}\left[\frac{\eta}{2}\sum_{t=1}^{T}\sum_{i=1}^{N}\frac{p_{t,i}}{o_{t,i}+\gamma}\right]$$

## Regret bound of Exp3-IX

$$R_T \leq \frac{\log N}{\eta} + \left(\frac{\eta}{2} + \gamma\right) \sum_{t=1}^{T} \mathbb{E}\left[\sum_{i=1}^{N} \frac{p_{t,i}}{o_{t,i} + \gamma}\right]$$

$$R_T \approx \mathcal{O}\left(\sqrt{\log N \sum_{t=1}^{T} \mathbb{E}\left[\sum_{i=1}^{N} \frac{p_{t,i}}{o_{t,i} + \gamma}\right]}\right)$$

## Regret bound of Exp3-IX

$$R_T \leq \frac{\log N}{\eta} + \left(\frac{\eta}{2} + \gamma\right) \sum_{t=1}^{T} \mathbb{E}\left[\sum_{i=1}^{N} \frac{p_{t,i}}{o_{t,i} + \gamma}\right]$$

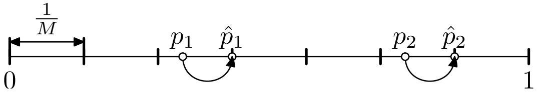
$$R_T \approx \mathcal{O}\left(\sqrt{\log N \sum_{t=1}^{T} \mathbb{E}\left[\sum_{i=1}^{N} \frac{p_{t,i}}{o_{t,i} + \gamma}\right]}\right)$$

## Graph lemma

- Graph $G$ with $V(G) = \{1, \ldots, N\}$
- $d_i^-$ – in-degree of vertex $i$
- $\alpha$ – independence set of $G$
- Turán's Theorem + induction

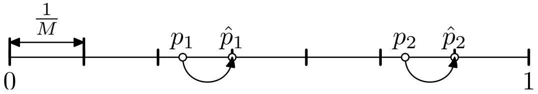
$$\sum_{i=1}^{N} \frac{1}{1 + d_i^-} \leq 2\alpha \log\left(1 + \frac{N}{\alpha}\right)$$

## Discretization




$$\sum_{i=1}^{N} \frac{p_{t,i}}{o_{t,i} + \gamma} = \sum_{i=1}^{N} \frac{p_{t,i}}{p_{t,i} + \sum_{j \in N_i^-} p_{t,j} + \gamma} \leq \sum_{i=1}^{N} \frac{\hat{p}_{t,i}}{\hat{p}_{t,i} + \sum_{j \in N_i^-} \hat{p}_{t,j}} + 2$$

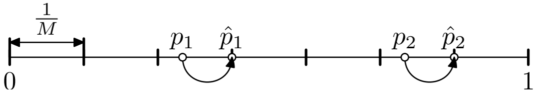
## Discretization




$$\sum_{i=1}^{N} \frac{p_{t,i}}{o_{t,i}+\gamma} = \sum_{i=1}^{N} \frac{p_{t,i}}{p_{t,i} + \sum_{j \in N_i^-} p_{t,j} + \gamma} \leq \sum_{i=1}^{N} \frac{\hat{p}_{t,i}}{\hat{p}_{t,i} + \sum_{j \in N_i^-} \hat{p}_{t,j}} + 2$$

**Note:** we set $M = \lceil N^2/\gamma \rceil$

## Discretization



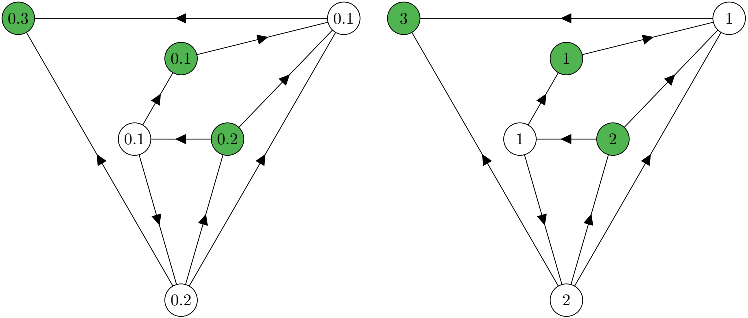

$$\sum_{i=1}^{N} \frac{p_{t,i}}{o_{t,i} + \gamma} = \sum_{i=1}^{N} \frac{p_{t,i}}{p_{t,i} + \sum_{j \in N_i^-} p_{t,j} + \gamma} \leq \sum_{i=1}^{N} \frac{\hat{p}_{t,i}}{\hat{p}_{t,i} + \sum_{j \in N_i^-} \hat{p}_{t,j}} + 2$$

**Note:** we set $M = \lceil N^2/\gamma \rceil$

$$\sum_{i=1}^{N} \frac{\hat{p}_{t,i}}{\hat{p}_{t,i} + \sum_{j \in N_i^-} \hat{p}_{t,j}}$$

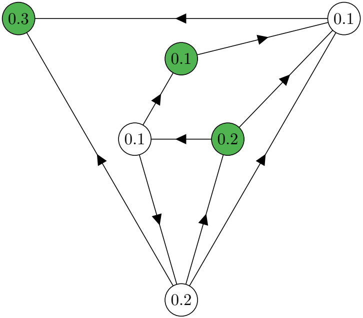
$$\sum_{i=1}^{N} \frac{\hat{p}_{t,i}}{\hat{p}_{t,i} + \sum_{j \in N_i^-} \hat{p}_{t,j}}$$

**Example:** let $M = 10$



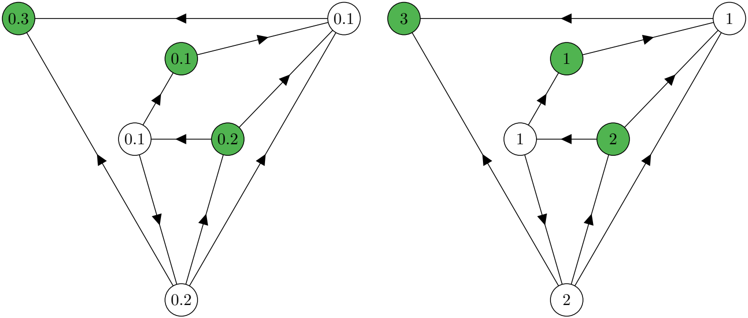
$$\sum_{i=1}^{N} \frac{M\hat{p}_{t,i}}{M\hat{p}_{t,i} + \sum_{j \in N_i^-} M\hat{p}_{t,j}}$$

**Example:** let $M = 10$



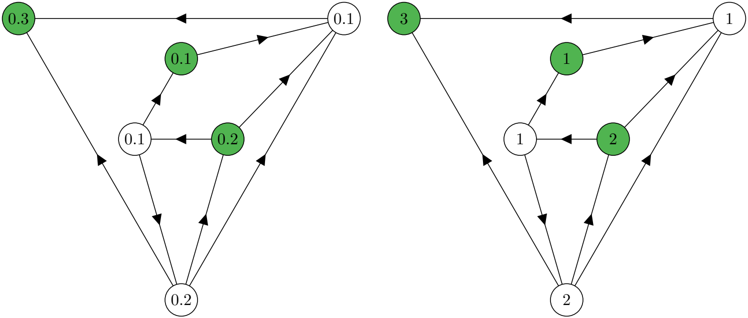
$$\sum_{i=1}^{N} \frac{M\hat{p}_{t,i}}{M\hat{p}_{t,i} + \sum_{j\in N_i^-} M\hat{p}_{t,j}} = \sum_{i=1}^{N} \sum_{k\in C_i} \frac{1}{1+d_k^-}$$

**Example:** let $M = 10$



$$\sum_{i=1}^{N} \frac{M\hat{p}_{t,i}}{M\hat{p}_{t,i} + \sum_{j\in N_i^-} M\hat{p}_{t,j}} = \sum_{i=1}^{N} \sum_{k\in C_i} \frac{1}{1+d_k^-} \leq 2\alpha \log\left(1 + \frac{M+N}{\alpha}\right)$$

**Example:** let $M = 10$



**Exp3-IX** regret bound

$$R_T \leq \frac{\log N}{\eta} + \left(\frac{\eta}{2} + \gamma\right) \sum_{t=1}^{T} \mathbb{E}\left[2\alpha_t \log\left(1 + \frac{\lceil N^2/\gamma \rceil + N}{\alpha_t}\right) + 2\right]$$

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

**Exp3-IX** regret bound

$$R_T \leq \frac{\log N}{\eta} + \left(\frac{\eta}{2} + \gamma\right) \sum_{t=1}^{T} \mathbb{E}\left[2\alpha_t \log\left(1 + \frac{\lceil N^2/\gamma \rceil + N}{\alpha_t}\right) + 2\right]$$

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

# Next step

**Exp3-IX** regret bound

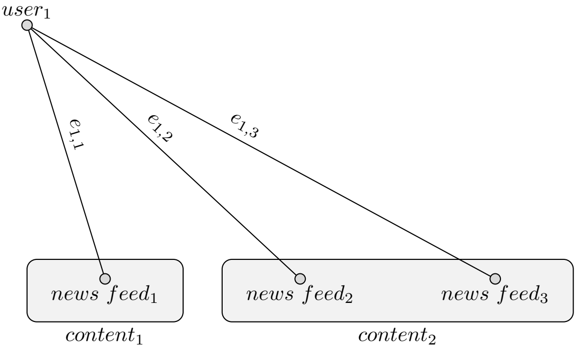
$$R_T \leq \frac{\log N}{\eta} + \left(\frac{\eta}{2} + \gamma\right) \sum_{t=1}^{T} \mathbb{E}\left[2\alpha_t \log\left(1 + \frac{\lceil N^2/\gamma \rceil + N}{\alpha_t}\right) + 2\right]$$

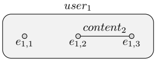
$$R_T = \widetilde{\mathcal{O}}\left(\sqrt{\overline{\alpha}\, T \log(N)}\right)$$

# Next step

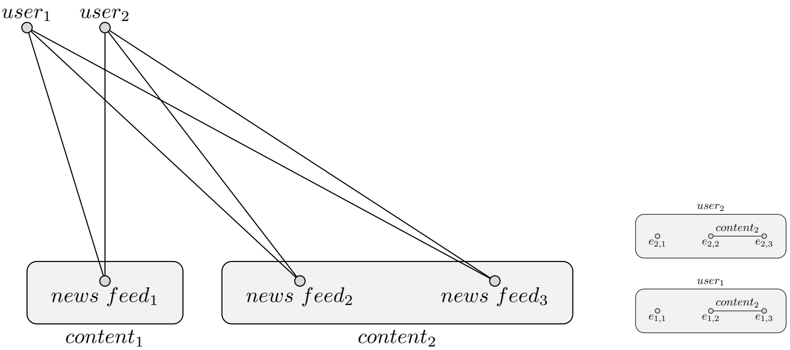
Generalization of the setting to **combinatorial actions**

# Example


$user_1$
$e_{1,1}$
$e_{1,2}$
$e_{1,3}$
$news\ feed_1$
$news\ feed_2$
$news\ feed_3$
$content_1$
$content_2$

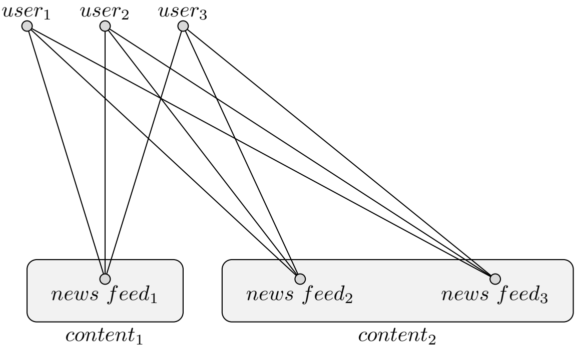

$user_1$
$content_2$
$e_{1,1}$
$e_{1,2}$
$e_{1,3}$

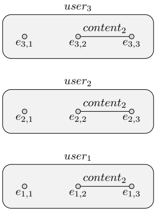
# Example



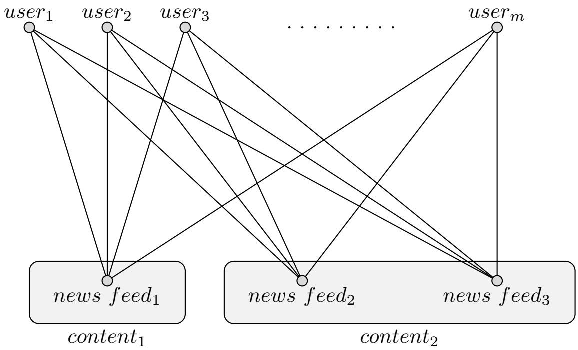
$user_1$
$user_2$
$news\ feed_1$
$content_1$
$news\ feed_2$
$news\ feed_3$
$content_2$
$user_2$
$content_2$
$e_{2,1}$
$e_{2,2}$
$e_{2,3}$
$user_1$
$content_2$
$e_{1,1}$
$e_{1,2}$
$e_{1,3}$

# Example

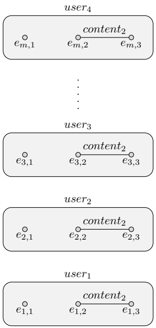

$user_1$
$user_2$
$user_3$
$news\ feed_1$
$news\ feed_2$
$news\ feed_3$
$content_1$
$content_2$


$user_3$
$content_2$
$e_{3,1}$ $e_{3,2}$ $e_{3,3}$
$user_2$
$content_2$
$e_{2,1}$ $e_{2,2}$ $e_{2,3}$
$user_1$
$content_2$
$e_{1,1}$ $e_{1,2}$ $e_{1,3}$

# Example

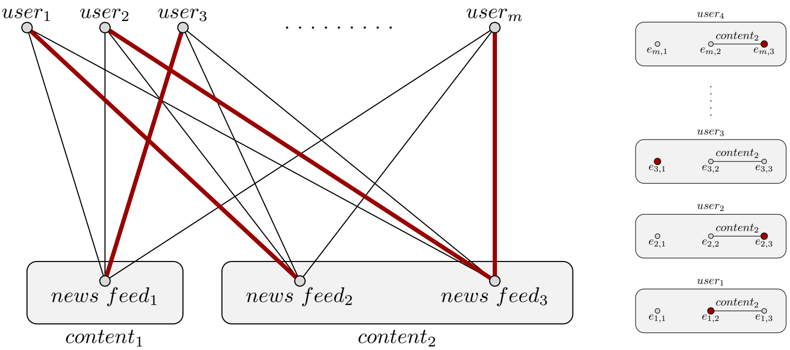


$user_1$
$user_2$
$user_3$
$user_m$
$news\ feed_1$
$news\ feed_2$
$news\ feed_3$
$content_1$
$content_2$

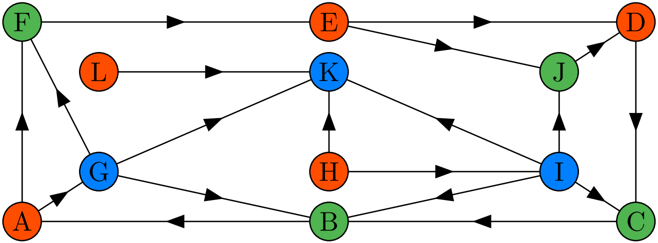



$user_4$
$content_2$
$e_{m,1}$
$e_{m,2}$
$e_{m,3}$
$user_3$
$content_2$
$e_{3,1}$
$e_{3,2}$
$e_{3,3}$
$user_2$
$content_2$
$e_{2,1}$
$e_{2,2}$
$e_{2,3}$
$user_1$
$content_2$
$e_{1,1}$
$e_{1,2}$
$e_{1,3}$

# Example




- Play $m$ out of $N$ nodes (combinatorial structure)
- Obtain losses of all played nodes
- Observe losses of all neighbors of played nodes




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

## FPL-IX algorithm

- **Draw perturbation** $Z_{t,i} \sim \mathrm{Exp}(1)$ for all $i \in [N]$

- **Play "the best" action** $\mathbf{V}_t$ according to total loss estimate $\widehat{\mathbf{L}}_{t-1}$ and perturbation $\mathbf{Z}_t$

$$\mathbf{V}_t = \arg\min_{\mathbf{v}\in\mathcal{S}} \mathbf{v}^\mathsf{T} \left(\eta_t \widehat{\mathbf{L}}_{t-1} - \mathbf{Z}_t\right)$$

- **Compute loss estimates**

$$\hat{\ell}_{t,i} = \ell_{t,i} K_{t,i} \mathbb{1}\{\ell_{t,i} \text{ is observed}\}$$

- $K_{t,i}$: geometric random variable with

$$\mathbb{E}\left[K_{t,i}\right] = \frac{1}{o_{t,i} + (1 - o_{t,i})\gamma}$$

## **FPL-IX** - regret bound

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

## Side Observation Summary

- **I**mplicit e**X**ploration idea
- **New algorithm for simple actions - Exp3-IX**
  - Using implicit exploration idea
  - Same regret bound as previous algorithm
  - No need to know graph before an action is played
  - Computationally efficient
- **New combinatorial setting with side observations**
- **Algorithm for combinatorial setting - FPL-IX**
- **Future directions**
  - No need to know graph after an action is played
  - Stochastic side observations
  - Random graph models
  - Exploiting the communities

**Michal Valko**
michal.valko@inria.fr
sequel.lille.inria.fr

**Sylvester's determinant theorem:**

$$|\mathbf{A} + \mathbf{x}\mathbf{x}^\intercal| = |\mathbf{A}||\mathbf{I} + \mathbf{A}^{-1}\mathbf{x}\mathbf{x}^\intercal| = |\mathbf{A}|(1 + \mathbf{x}^\intercal\mathbf{A}^{-1}\mathbf{x})$$

**Goal:**

- Upperbound determinant $|\mathbf{A} + \mathbf{x}\mathbf{x}^\intercal|$ for $\|\mathbf{x}\|_2 \leq 1$
- Upperbound $\mathbf{x}^\intercal\mathbf{A}^{-1}\mathbf{x}$

**Sylvester's determinant theorem:**

$$|\mathbf{A} + \mathbf{x}\mathbf{x}^\intercal| = |\mathbf{A}||\mathbf{I} + \mathbf{A}^{-1}\mathbf{x}\mathbf{x}^\intercal| = |\mathbf{A}|(1 + \mathbf{x}^\intercal\mathbf{A}^{-1}\mathbf{x})$$

**Goal:**

- Upperbound determinant $|\mathbf{A} + \mathbf{x}\mathbf{x}^\intercal|$ for $\|\mathbf{x}\|_2 \leq 1$
- Upperbound $\mathbf{x}^\intercal\mathbf{A}^{-1}\mathbf{x}$

$$\mathbf{x}^\intercal\mathbf{A}^{-1}\mathbf{x} = \mathbf{x}^\intercal\mathbf{Q}\boldsymbol{\Lambda}^{-1}\mathbf{Q}^\intercal\mathbf{x} = \mathbf{y}^\intercal\boldsymbol{\Lambda}^{-1}\mathbf{y} = \sum_{i=1}^{N}\lambda_i y_i^2$$

**Sylvester's determinant theorem:**

$$|\mathbf{A} + \mathbf{x}\mathbf{x}^\mathsf{T}| = |\mathbf{A}||\mathbf{I} + \mathbf{A}^{-1}\mathbf{x}\mathbf{x}^\mathsf{T}| = |\mathbf{A}|(1 + \mathbf{x}^\mathsf{T}\mathbf{A}^{-1}\mathbf{x})$$

**Goal:**

- Upperbound determinant $|\mathbf{A} + \mathbf{x}\mathbf{x}^\mathsf{T}|$ for $\|\mathbf{x}\|_2 \leq 1$
- Upperbound $\mathbf{x}^\mathsf{T}\mathbf{A}^{-1}\mathbf{x}$

$$\mathbf{x}^\mathsf{T}\mathbf{A}^{-1}\mathbf{x} = \mathbf{x}^\mathsf{T}\mathbf{Q}\boldsymbol{\Lambda}^{-1}\mathbf{Q}^\mathsf{T}\mathbf{x} = \mathbf{y}^\mathsf{T}\boldsymbol{\Lambda}^{-1}\mathbf{y} = \sum_{i=1}^{N} \lambda_i y_i^2$$

- $\|\mathbf{y}\|_2 \leq 1$.
- $\mathbf{y}$ is a canonical vector.
- $\mathbf{x} = \mathbf{Q}\mathbf{y}$ is an eigenvector of $\mathbf{A}$.

**Corollary**:
Determinant $|\mathbf{V}_T|$ of $\mathbf{V}_T = \mathbf{\Lambda} + \sum_{t=1}^T \mathbf{x}_t\mathbf{x}_t^\mathsf{T}$ is maximized when all $\mathbf{x}_t$ are aligned with axes.

$$|\mathbf{V}_T| \leq \max_{\sum t_i = T} \prod(\lambda_i + t_i)$$

$$\ln \frac{|\mathbf{V}_T|}{|\mathbf{\Lambda}|} \leq \max_{\sum t_i = T} \sum \ln\left(1 + \frac{t_i}{\lambda_i}\right)$$

$$\ln \frac{|\mathbf{V}_T|}{|\mathbf{\Lambda}|} \leq \sum_{i=1}^{d} \ln\left(1 + \frac{T}{\lambda}\right) + \sum_{i=d+1}^{N} \ln\left(1 + \frac{t_i}{\lambda_{d+1}}\right)$$

$$\leq d \ln\left(1 + \frac{T}{\lambda}\right) + \frac{T}{\lambda_{d+1}}$$

$$\leq 2d \ln\left(1 + \frac{T}{\lambda}\right)$$

$$\boldsymbol{f}^\mathsf{T}\mathcal{L}\boldsymbol{f} = \frac{1}{2}\sum_{i,j\leq N} w_{i,j}(f_i - f_j)^2 = S_G(f)$$

**Proof:**

$$\begin{aligned}
\boldsymbol{f}^\mathsf{T}\mathcal{L}\boldsymbol{f} &= \boldsymbol{f}^\mathsf{T}\mathcal{D}\boldsymbol{f} - \boldsymbol{f}^\mathsf{T}\mathcal{W}\boldsymbol{f} = \sum_{i=1}^{N} d_i f_i^2 - \sum_{i,j\leq N} w_{i,j} f_i f_j \\
&= \frac{1}{2}\left(\sum_{i=1}^{N} d_i f_i^2 - 2\sum_{i,j\leq N} w_{i,j} f_i f_j + \sum_{j=1}^{N} d_i f_j^2\right) = \frac{1}{2}\sum_{i,j\leq N} w_{i,j}(f_i - f_j)^2
\end{aligned}$$

# SpectralUCB analysis sketch

- Derivation of the confidence ellipsoid for $\hat{\boldsymbol{\alpha}}$ with probability $1-\delta$.
  - Using analysis of OFUL (Abbasi-Yadkori et al., 2011)

$$|x^\mathsf{T}(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*)| \leq \|x\|_{\mathbf{V}_t^{-1}} \left( R\sqrt{2\ln\left(\frac{|V_t|^{1/2}}{\delta|\boldsymbol{\Lambda}|^{1/2}}\right)} + C \right)$$

- Regret in one time step: $r_t = \mathbf{x}_*^\mathsf{T}\boldsymbol{\alpha}^* - \mathbf{x}_{\pi(t)}^\mathsf{T}\boldsymbol{\alpha}^* \leq 2c_t\|\mathbf{x}_{\pi(t)}\|_{\mathbf{V}_t^{-1}}$
- Cumulative regret:

$$R_T = \sum_{t=1}^T r_t \leq \sqrt{T\sum_{t=1}^T r_t^2} \leq 2(\ c_T + 1)\sqrt{2T\ln\frac{|\mathbf{V}_T|}{|\boldsymbol{\Lambda}|}}$$

# SpectralUCB analysis sketch

- Derivation of the confidence ellipsoid for $\hat{\boldsymbol{\alpha}}$ with probability $1-\delta$.
  - Using analysis of OFUL (Abbasi-Yadkori et al., 2011)

$$|x^\mathsf{T}(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}^*)| \leq \|x\|_{\mathbf{V}_t^{-1}} \left( R\sqrt{2\ln\left(\frac{|V_t|^{1/2}}{\delta|\boldsymbol{\Lambda}|^{1/2}}\right)} + C \right)$$

- Regret in one time step: $r_t = \mathbf{x}_*^\mathsf{T}\boldsymbol{\alpha}^* - \mathbf{x}_{\pi(t)}^\mathsf{T}\boldsymbol{\alpha}^* \leq 2c_t\|\mathbf{x}_{\pi(t)}\|_{\mathbf{V}_t^{-1}}$
- Cumulative regret:

$$R_T = \sum_{t=1}^T r_t \leq \sqrt{T\sum_{t=1}^T r_t^2} \leq 2(c_T + 1)\sqrt{2T\ln\frac{|\mathbf{V}_T|}{|\boldsymbol{\Lambda}|}}$$

# SpectralUCB analysis sketch

- Derivation of the confidence ellipsoid for $\hat{\boldsymbol{\alpha}}$ with probability $1-\delta$.
  - Using analysis of OFUL (Abbasi-Yadkori et al., 2011)

$$|x^\mathsf{T}(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}^*)| \leq \|x\|_{\mathbf{V}_t^{-1}} \left( R\sqrt{2\ln\left(\frac{|V_t|^{1/2}}{\delta|\boldsymbol{\Lambda}|^{1/2}}\right)} + C \right)$$

- Regret in one time step: $r_t = \mathbf{x}_*^\mathsf{T}\boldsymbol{\alpha}^* - \mathbf{x}_{\pi(t)}^\mathsf{T}\boldsymbol{\alpha}^* \leq 2c_t\|\mathbf{x}_{\pi(t)}\|_{\mathbf{V}_t^{-1}}$
- Cumulative regret:

$$R_T = \sum_{t=1}^T r_t \leq \sqrt{T\sum_{t=1}^T r_t^2} \leq 2(c_T+1)\sqrt{2T\ln\frac{|\mathbf{V}_T|}{|\boldsymbol{\Lambda}|}}$$

- Upperbound for $\ln(|\mathbf{V}_t|/|\boldsymbol{\Lambda}|)$

$$\ln\frac{|\mathbf{V}_t|}{|\boldsymbol{\Lambda}|} \leq \ln\frac{|\mathbf{V}_T|}{|\boldsymbol{\Lambda}|} \leq 2d\ln\left(\frac{\lambda+T}{\lambda}\right)$$

# SpectralTS analysis sketch

**Divide arms into two groups**

- $\Delta_i = \mathbf{b}_*^\mathsf{T}\boldsymbol{\mu} - \mathbf{b}_i^\mathsf{T}\boldsymbol{\mu} \leq g\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}$ arm $i$ is **unsaturated**
- $\Delta_i = \mathbf{b}_*^\mathsf{T}\boldsymbol{\mu} - \mathbf{b}_i^\mathsf{T}\boldsymbol{\mu} > g\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}$ arm $i$ is **saturated**

# SpectralTS analysis sketch

**Divide arms into two groups**

- $\Delta_i = \mathbf{b}_*^\mathsf{T}\boldsymbol{\mu} - \mathbf{b}_i^\mathsf{T}\boldsymbol{\mu} \leq g\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}$ arm $i$ is **unsaturated**
- $\Delta_i = \mathbf{b}_*^\mathsf{T}\boldsymbol{\mu} - \mathbf{b}_i^\mathsf{T}\boldsymbol{\mu} > g\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}$ arm $i$ is **saturated**

**Saturated arm**

- Small standard deviation $\rightarrow$ accurate regret estimate.
- **High regret** on playing the arm $\rightarrow$ **Low probability** of picking

# SpectralTS analysis sketch

**Divide arms into two groups**

- $\Delta_i = \mathbf{b}_*^\mathsf{T}\boldsymbol{\mu} - \mathbf{b}_i^\mathsf{T}\boldsymbol{\mu} \leq g\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}$ arm $i$ is **unsaturated**
- $\Delta_i = \mathbf{b}_*^\mathsf{T}\boldsymbol{\mu} - \mathbf{b}_i^\mathsf{T}\boldsymbol{\mu} > g\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}$ arm $i$ is **saturated**

**Saturated arm**

- Small standard deviation $\rightarrow$ accurate regret estimate.
- **High regret** on playing the arm $\rightarrow$ **Low probability** of picking

**Unsaturated arm**

- **Low regret** bounded by a factor of standard deviation
- **High probability** of picking

# SpectralTS analysis sketch

- Confidence ellipsoid for estimate $\hat{\boldsymbol{\mu}}$ of $\boldsymbol{\mu}$ (with probability $1 - \delta/T^2$)
  - Using analysis of OFUL algorithm (Abbasi-Yadkori et al., 2011)

$$|\mathbf{b}_i^\mathsf{T}\hat{\boldsymbol{\mu}} - \mathbf{b}_i^\mathsf{T}\boldsymbol{\mu}| \leq \left(R\sqrt{2\log\left(\frac{|\mathbf{B}_T|^{1/2}T^2}{|\boldsymbol{\Lambda}|^{1/2}\delta}\right)} + C\right)\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}$$

# SpectralTS analysis sketch

- Confidence ellipsoid for estimate $\hat{\boldsymbol{\mu}}$ of $\boldsymbol{\mu}$ (with probability $1 - \delta/T^2$)
  - Using analysis of OFUL algorithm (Abbasi-Yadkori et al., 2011)

$$|\mathbf{b}_i^{\mathsf{T}}\hat{\boldsymbol{\mu}} - \mathbf{b}_i^{\mathsf{T}}\boldsymbol{\mu}| \leq \left( R\sqrt{2\log\left(\frac{|\mathbf{B}_T|^{1/2}T^2}{|\mathbf{\Lambda}|^{1/2}\delta}\right)} + C \right) \|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}$$

- Our key result coming from spectral properties of $\mathbf{B}_t$.

$$\log\frac{|\mathbf{B}_t|}{|\mathbf{\Lambda}|} \leq 2d\log\left(1 + \frac{T}{\lambda}\right)$$

# SpectralTS analysis sketch

- Confidence ellipsoid for estimate $\hat{\boldsymbol{\mu}}$ of $\boldsymbol{\mu}$ (with probability $1 - \delta/T^2$)
  - Using analysis of OFUL algorithm (Abbasi-Yadkori et al., 2011)

$$|\mathbf{b}_i^\mathsf{T}\hat{\boldsymbol{\mu}} - \mathbf{b}_i^\mathsf{T}\boldsymbol{\mu}| \leq \left(R\sqrt{2\,d\log\left(\frac{(\lambda+T)T^2}{\delta\lambda}\right)} + C\right)\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}} = \ell\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}$$

- Our key result coming from spectral properties of $\mathbf{B}_t$.

$$\log\frac{|\mathbf{B}_t|}{|\boldsymbol{\Lambda}|} \leq 2d\log\left(1 + \frac{T}{\lambda}\right)$$

# SpectralTS analysis sketch

- Confidence ellipsoid for estimate $\hat{\boldsymbol{\mu}}$ of $\boldsymbol{\mu}$ (with probability $1 - \delta/T^2$)
  - Using analysis of OFUL algorithm (Abbasi-Yadkori et al., 2011)

$$|\mathbf{b}_i^{\mathsf{T}}\hat{\boldsymbol{\mu}} - \mathbf{b}_i^{\mathsf{T}}\boldsymbol{\mu}| \leq \left(R\sqrt{2\,d\log\left(\frac{(\lambda + T)T^2}{\delta\lambda}\right)} + C\right)\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}} = \ell\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}$$

- Our key result coming from spectral properties of $\mathbf{B}_t$.

$$\log\frac{|\mathbf{B}_t|}{|\boldsymbol{\Lambda}|} \leq 2d\log\left(1 + \frac{T}{\lambda}\right)$$

- Concentration of sample $\tilde{\boldsymbol{\mu}}$ around mean $\hat{\boldsymbol{\mu}}$ (with probability $1 - 1/T^2$)
  - Using concentration inequality for Gaussian random variable.

$$|\mathbf{b}_i^{\mathsf{T}}\tilde{\boldsymbol{\mu}} - \mathbf{b}_i^{\mathsf{T}}\hat{\boldsymbol{\mu}}| \leq \left(R\sqrt{6d\log\left(\frac{\lambda + T}{\delta\lambda}\right)} + C\right)\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}\sqrt{4\log(TN)} = v\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}\sqrt{4\log(TN)}$$

# SpectralTS analysis sketch

**Define** $\text{regret}'(t) = \text{regret}(t) \cdot \mathbb{1}\{|\mathbf{b}_i^\mathsf{T}\hat{\boldsymbol{\mu}}(t) - \mathbf{b}_i^\mathsf{T}\boldsymbol{\mu}| \le \ell \|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}\}$

$$\text{regret}'(t) \le \frac{11g}{p}\|\mathbf{b}_{a(t)}\|_{\mathbf{B}_t^{-1}} + \frac{1}{T^2}$$

# SpectralTS analysis sketch

**Define** $\mathrm{regret}'(t) = \mathrm{regret}(t) \cdot \mathbb{1}\{|\mathbf{b}_i^\intercal \hat{\boldsymbol{\mu}}(t) - \mathbf{b}_i^\intercal \boldsymbol{\mu}| \leq \ell \|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}\}$

$$\mathrm{regret}'(t) \leq \frac{11g}{p} \|\mathbf{b}_{a(t)}\|_{\mathbf{B}_t^{-1}} + \frac{1}{T^2}$$

**Super-martingale** (i.e. $\mathbb{E}[Y_t - Y_{t-1} | \mathcal{F}_{t-1}] \leq 0$)

$$X_t = \mathrm{regret}'(t) - \frac{11g}{p} \|\mathbf{b}_{a(t)}\|_{\mathbf{B}_t^{-1}} - \frac{1}{T^2}$$

$$Y_t = \sum_{w=1}^{t} X_w.$$

$(Y_t;\ t = 0, \ldots, T)$ is a **super-martingale** process w.r.t. history $\mathcal{F}_t$.

# SpectralTS analysis sketch

**Define** $\text{regret}'(t) = \text{regret}(t) \cdot \mathbb{1}\{|\mathbf{b}_i^\mathsf{T}\hat{\boldsymbol{\mu}}(t) - \mathbf{b}_i^\mathsf{T}\boldsymbol{\mu}| \leq \ell\|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}\}$

$$\text{regret}'(t) \leq \frac{11g}{p}\|\mathbf{b}_{a(t)}\|_{\mathbf{B}_t^{-1}} + \frac{1}{T^2}$$

**Super-martingale** (i.e. $\mathbb{E}[Y_t - Y_{t-1}|\mathcal{F}_{t-1}] \leq 0$)

$$X_t = \text{regret}'(t) - \frac{11g}{p}\|\mathbf{b}_{a(t)}\|_{\mathbf{B}_t^{-1}} - \frac{1}{T^2}$$

$$Y_t = \sum_{w=1}^{t} X_w.$$

$(Y_t;\ t = 0, \ldots, T)$ is a **super-martingale** process w.r.t. history $\mathcal{F}_t$.

**Azuma-Hoeffding inequality for super-martingale**, w. p. $1 - \delta/2$:

$$\sum_{t=1}^{T} \text{regret}'(t) \leq \frac{11g}{p}\sum_{t=1}^{T}\|\mathbf{b}_{a(t)}\|_{\mathbf{B}_t^{-1}} + \frac{1}{T} + \frac{g}{p}\left(\frac{11}{\sqrt{\lambda}} + 2\right)\sqrt{2T\ln\frac{2}{\delta}}$$

# SpectralTS analysis sketch

**Define** $\text{regret}'(t) = \text{regret}(t) \cdot \mathbb{1}\{|\mathbf{b}_i^\mathsf{T}\hat{\boldsymbol{\mu}}(t) - \mathbf{b}_i^\mathsf{T}\boldsymbol{\mu}| \leq \ell \|\mathbf{b}_i\|_{\mathbf{B}_t^{-1}}\}$

$$\text{regret}'(t) \leq \frac{11g}{p}\|\mathbf{b}_{a(t)}\|_{\mathbf{B}_t^{-1}} + \frac{1}{T^2}$$

**Super-martingale** (i.e. $\mathbb{E}[Y_t - Y_{t-1}|\mathcal{F}_{t-1}] \leq 0$)

$$X_t = \text{regret}'(t) - \frac{11g}{p}\|\mathbf{b}_{a(t)}\|_{\mathbf{B}_t^{-1}} - \frac{1}{T^2}$$

$$Y_t = \sum_{w=1}^{t} X_w.$$

$(Y_t;\ t = 0, \dots, T)$ is a **super-martingale** process w.r.t. history $\mathcal{F}_t$.

**Azuma-Hoeffding inequality for super-martingale**, w. p. $1 - \delta/2$:

$$\sum_{t=1}^{T} \text{regret}'(t) \leq \frac{11g}{p} \sum_{t=1}^{T} \|\mathbf{b}_{a(t)}\|_{\mathbf{B}_t^{-1}} + \frac{1}{T} + \frac{g}{p}\left(\frac{11}{\sqrt{\lambda}} + 2\right)\sqrt{2T \ln \frac{2}{\delta}}$$

# Backup: SpectralEliminator pseudocode

**Input:**

$N$ : the number of nodes, $T$ : the number of pulls

$\{\mathbf{\Lambda}_{\mathcal{L}}, \mathbf{Q}\}$ spectral basis of $\mathcal{L}$

$\lambda$ : regularization parameter

$\beta, \{t_j\}_j^J$ parameters of the elimination and phases

$A_1 \leftarrow \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$.

**for** $j = 1$ **to** $J$ **do**

$\mathbf{V}_{t_j} \leftarrow \gamma \mathbf{\Lambda}_{\mathcal{L}} + \lambda \mathbf{I}$

**for** $t = t_j$ **to** $\min(t_{j+1} - 1, T)$ **do**

Play $\mathbf{x}_t \in A_j$ with the largest width to observe $r_t$:

$\mathbf{x}_t \leftarrow \arg\max_{\mathbf{x} \in A_j} \|\mathbf{x}\|_{\mathbf{V}_t^{-1}}$

$\mathbf{V}_{t+1} \leftarrow \mathbf{V}_t + \mathbf{x}_t \mathbf{x}_t^\mathsf{T}$

**end for**

Eliminate the arms that are not promising:

$\hat{\boldsymbol{\alpha}}_t \leftarrow \mathbf{V}_t^{-1}[\mathbf{x}_{t_j}, \ldots, \mathbf{x}_t][r_{t_j}, \ldots, r_t]^\mathsf{T}$

$A_{j+1} \leftarrow \left\{ \mathbf{x} \in A_j, \langle \hat{\boldsymbol{\alpha}}_t, \mathbf{x} \rangle + \|\mathbf{x}\|_{V_t^{-1}} \beta \geq \max_{\mathbf{x} \in A_j} \left[ \langle \hat{\boldsymbol{\alpha}}_t, \mathbf{x} \rangle - \|\mathbf{x}\|_{\mathbf{V}_t^{-1}} \beta \right] \right\}$

**end for**

# Backup: SpectralEliminator analysis

**SpectralEliminator**

- Divide time into sets $(t_1 = 1 \leq t_2 \leq \dots)$ to introduce independence for Azuma-Hoeffding inequality and observe $R_T \leq \sum_{j=0}^{J}(t_{j+1} - t_j)\big[\langle \mathbf{x}^* - \mathbf{x}_t, \hat{\alpha}_j\rangle + (\|\mathbf{x}^*\|_{\mathbf{V}_j^{-1}} + \|\mathbf{x}_t\|_{\mathbf{V}_j^{-1}})\beta\big]$
- Bound $\langle \mathbf{x}^* - \mathbf{x}_t, \hat{\alpha}_j\rangle$ for each phase
- No bad arms: $\langle \mathbf{x}^* - \mathbf{x}_t, \hat{\alpha}_j\rangle \leq (\|\mathbf{x}^*\|_{\mathbf{V}_j^{-1}} + \|\mathbf{x}_t\|_{\mathbf{V}_j^{-1}})\beta$
- By algorithm: $\|\mathbf{x}\|^2_{\mathbf{V}_j^{-1}} \leq \frac{1}{t_j - t_{j-1}} \sum_{s=t_{j-1}+1}^{t_j} \|\mathbf{x}_s\|^2_{\mathbf{V}_{s-1}^{-1}}$
- $\sum_{s=t_{j-1}+1}^{t_j} \min\left(1, \|\mathbf{x}_s\|^2_{\mathbf{V}_{s-1}^{-1}}\right) \leq \log \frac{|\mathbf{V}_j|}{|\Lambda|}$