Graphs in Machine Learning

Daniele Calandriello

DeepMind Paris, France

Collaborators: Achraf Azize,
Michal Valko

Partially based on material by Michal Valko

This lecture

- Multi-Armed Bandits
- Spectral Bandits
- Influence Maximization

Bandits

learning with real-time interactions

Multi-Armed Bandits

- At each time t, the learner chooses an action A_t
- Then, it receives a reward $r_t = f(A_t) + \varepsilon_t$

Some applications: movie recommendation, clinical trials, ads etc.

Multi-Armed Bandits

- At each time t, the learner chooses an action A_t
- Then, it receives a reward $r_t = f(A_t) + \varepsilon_t$

Some applications: movie recommendation, clinical trials, ads etc.

$$\text{maximize sum of rewards} \quad \sum_{t=1}^{T} f(A_t)$$

$$\iff \text{minimize the regret} \quad R_T = \sum_{t=1}^{T} \max_a f(a) - f(A_t)$$

We care about the performance **during** learning!

Multi-Armed Bandits

- Set of arms $a \in \{1, \ldots, N\}$
- $f(a) = \mu_a$ mean reward of arm a
- $r_t = \mu_{A_t} + \varepsilon_t$ observed reward when choosing the arm A_t
- $\mu^* = \max_a f(a) = f(a^*)$ mean reward of the optimal arm a^*

The regret after T rounds is

$$R_T = T\mu^* - \sum_{t=1}^{T} \mu_{A_t}$$

where $(A_1, \ldots, A_T)$ are the actions chosen by the algorithm.

We want an algorithm that gives $R_T/T \to 0$ as $T \to \infty$!

Naive Algorithms

- $A_t =$ random action

$$\implies \mathbb{E}[R_T] = \Omega(T)$$

- $A_t =$ random action with prob. ε, action with highest empirical mean with prob. $1 - \varepsilon$

$$\implies \mathbb{E}[R_T] = \Omega(\varepsilon T)$$

Linear regret, we don't want that!

Upper Confidence Bound (UCB) Algorithm

- Build confidence intervals for the mean of each arm.
- Act *optimistically* to balance exploration and exploitation.

Let $\widehat{\mu}_a(t)$ be the empirical mean of the arm a at round t:

$$\widehat{\mu}_a(t) = \frac{1}{N_a(t)} \sum_{s=1}^{t-1} \mathbb{1}\{A_t = a\} r_t \;, \quad \text{where} \quad N_a(t) = \sum_{s=1}^{t-1} \mathbb{1}\{A_t = a\}.$$

Upper Confidence Bound (UCB) Algorithm

- Build confidence intervals for the mean of each arm.
- Act *optimistically* to balance exploration and exploitation.

Let $\widehat{\mu}_a(t)$ be the empirical mean of the arm a at round t:

$$\widehat{\mu}_a(t) = \frac{1}{N_a(t)} \sum_{s=1}^{t-1} \mathbb{1}\{A_t = a\} r_t \ , \quad \text{where} \quad N_a(t) = \sum_{s=1}^{t-1} \mathbb{1}\{A_t = a\}.$$

If the noise satisfies $\mathbb{E}[\exp(\lambda \varepsilon_t) | \mathcal{F}_t] \leq \exp(\sigma^2 \lambda^2 / 2)$ for all $\lambda \in \mathbb{R}$, then, with prob. at least $1 - \delta$, $\forall t, \forall a$,

$$|\widehat{\mu}_a(t) - \mu_a| \leq \sqrt{\frac{2\sigma^2}{N_a(t)} \log\left(\frac{N_a(t)(N_a(t)+1)N}{\delta}\right)}$$

Proof: Use Hoeffding's inequality.

Upper Confidence Bound (UCB) Algorithm

UCB Strategy: Play, in each round t, the action A_t such that

$$A_t \in \arg\max_a \left(\widehat{\mu}_a(t) + \sqrt{\frac{2\sigma^2}{N_a(t)} \log\left(\frac{N_a(t)(N_a(t)+1)N}{\delta} \right)} \right)$$

Upper Confidence Bound (UCB) Algorithm

UCB Strategy: Play, in each round t, the action A_t such that

$$A_t \in \arg\max_a \left(\widehat{\mu}_a(t) + \sqrt{\frac{2\sigma^2}{N_a(t)} \log\left(\frac{N_a(t)(N_a(t)+1)N}{\delta} \right)} \right)$$

Theorem (UCB regret bound)

With probability at least $1-\delta$, the regret of UCB is bounded as

$$R_T \le c\sqrt{\sigma^2 NT \log(NT/\delta)}$$

where c is a constant and N is the number of actions.

UCB - Proof of the Regret Bound (sketch)

Recall that $R_T = T\mu^* - \sum_{t=1}^{T} \mu_{A_t} = \sum_{t=1}^{T}(\mu^* - \mu_{A_t})$.

Let $g(n) \stackrel{\text{def}}{=} \frac{1}{n} \log\left(\frac{n(n+1)}{\delta} N\right)$. Then, with prob. at least $1-\delta$,

$$
\begin{aligned}
\mu^* - \mu_{A_t} &\leq \widehat{\mu}_{a^*}(t) + \sqrt{2\sigma^2 g(N_{a^*}(t))} - \widehat{\mu}_{A_t}(t) + \sqrt{2\sigma^2 g(N_{A_t}(t))} && (1)\\
&\leq \widehat{\mu}_{A_t}(t) + \sqrt{2\sigma^2 g(N_{A_t}(t))} - \widehat{\mu}_{A_t}(t) + \sqrt{2\sigma^2 g(N_{A_t}(t))} && (2)\\
&= 2\sqrt{2\sigma^2 g(N_{A_t}(t))} = 2\sqrt{\frac{2\sigma^2}{N_{A_t}(t)} \log\left(\frac{N_{A_t}(t)(N_{A_t}(t)+1)}{\delta} N\right)}
\end{aligned}
$$

where (1) comes from the confidence intervals and (2) comes from the definition of the UCB algorithm.

UCB - Proof of the Regret Bound (sketch)

Consequently, with probability at least $1-\delta$ ($\lesssim$ omits constants)

$$R_T \lesssim \boxed{\sigma\sqrt{\log\left(\frac{NT}{\delta}\right)}\sum_{t=1}^{T}\sqrt{\frac{1}{N_{A_t}(t)}}} = \sigma\sqrt{\log\left(\frac{NT}{\delta}\right)}\sum_{a=1}^{N}\sum_{t=1}^{T}\mathbb{1}_{A_t=a}\sqrt{\frac{1}{N_a(t)}}$$

UCB - Proof of the Regret Bound (sketch)

Consequently, with probability at least $1 - \delta$ ($\lesssim$ omits constants)

$$R_T \lesssim \boxed{\sigma \sqrt{\log\left(\frac{NT}{\delta}\right)} \sum_{t=1}^{T} \sqrt{\frac{1}{N_{A_t}(t)}}} = \sigma \sqrt{\log\left(\frac{NT}{\delta}\right)} \sum_{a=1}^{N} \sum_{t=1}^{T} \mathbb{1}_{A_t = a} \sqrt{\frac{1}{N_a(t)}}$$

$$= \sigma \sqrt{\log(NT/\delta)} \sum_{a=1}^{N} \sum_{t=1}^{T} (N_a(t+1) - N_a(t)) \sqrt{\frac{1}{N_a(t)}}$$

$$= \sigma \sqrt{\log(NT/\delta)} \sum_{a=1}^{N} \sum_{t=1}^{T} \int_{N_a(t)}^{N_a(t+1)} \sqrt{\frac{1}{N_a(t)}} \mathrm{d}x$$

$$\lesssim \sigma \sqrt{\log(NT/\delta)} \sum_{a=1}^{N} \sqrt{N_a(T+1)} \leq \sqrt{\log(NT/\delta)} \sqrt{N} \sqrt{\sum_{a=1}^{N} N_a(T+1)}$$

$$= \boxed{\sqrt{NT \log(NT/\delta)}}.$$

Lower Bound

In a worst-case scenario, what is the best regret we can achieve?

Theorem ([Aue+02])

For any number of actions $N \geq 2$, there exists a bandit such that

$$\mathbb{E}[R_T] \geq \frac{1}{20} \min\left(\sqrt{NT}, T\right)$$

$\implies$ UCB matches the lower bound, up to constants and logarithmic terms!

Questions:

- What happens if $N > T$?
- Can you think of problems where $N > T$?

Spectral Bandits

f is **smooth** on a graph

Spectral Bandits

Can we learn when $N > T$ by using *similarity graphs*?

Assumptions:

- Each action $a \in \{1, \dots, N\}$ is a node in a graph $\mathcal{G}$.
- If two actions a, b are similar, their mean rewards $f(a), f(b)$ are close.

Notation:

- $\mathcal{V}$: set of nodes (actions) $= \{1, \dots, N\}$.
- $\mathcal{W}$: $N \times N$ similarity matrix, $\mathcal{D}$: $N \times N$ degree matrix.
- $\mathcal{L} = \mathcal{D} - \mathcal{W}$: graph Laplacian.
- $\{\lambda_k^{\mathcal{L}}, q_k\}_{k=1}^{N}$: eigenvalues and eigenvectors of $\mathcal{L}$.
- $\mathcal{L} = Q\Lambda_{\mathcal{L}}Q^{\mathsf{T}}$: eigendecomposition of $\mathcal{L}$.

Spectral Bandits

Let $f_{\boldsymbol{\alpha}} : \mathcal{V} \to \mathbb{R}$ be the mean reward function. At each round t:

- The learner chooses an action (node) A_t;
- It receives a reward $r_t = f_{\boldsymbol{\alpha}}(A_t) + \varepsilon_t$

Let $\Lambda = \Lambda_{\mathcal{L}} + \lambda I$. We make the smoothness assumption:

$$f_{\boldsymbol{\alpha}}(a) = \sum_{k=1}^{N} \alpha_k (q_k)_a = x_a^{\mathsf{T}} \boldsymbol{\alpha} \quad \text{such that } \|\boldsymbol{\alpha}\|_{\Lambda}^2 = \sum_{k=1}^{N} \lambda_i \alpha_i^2 \leq C$$

Spectral Bandits

Let $f_{\boldsymbol{\alpha}} : \mathcal{V} \to \mathbb{R}$ be the mean reward function. At each round t:

- The learner chooses an action (node) A_t;
- It receives a reward $r_t = f_{\boldsymbol{\alpha}}(A_t) + \varepsilon_t$

Let $\Lambda = \Lambda_{\mathcal{L}} + \lambda I$. We make the smoothness assumption:

$$f_{\boldsymbol{\alpha}}(a) = \sum_{k=1}^{N} \alpha_k (q_k)_a = x_a^{\mathsf{T}} \boldsymbol{\alpha} \quad \text{such that } \|\boldsymbol{\alpha}\|_{\Lambda}^2 = \sum_{k=1}^{N} \lambda_i \alpha_i^2 \leq C$$

Question:

- Why does it mean that $f_{\boldsymbol{\alpha}}$ is smooth on the graph?

SpectralUCB

UCB strategy: estimate mean reward ($\boldsymbol{\alpha}$) and add confidence bound.

How to estimate α? Linear regression with graph regularization!

$$\widehat{\boldsymbol{\alpha}}_t = \arg\min_{\omega \in \mathbb{R}^N} \left(\sum_{s=1}^{t-1} \left(x_{A_s}^\mathsf{T} \omega - r_s \right)^2 + \|\omega\|_\Lambda^2 \right) = V_t^{-1} X_t^\mathsf{T} r$$

SpectralUCB

UCB strategy: estimate mean reward ($\boldsymbol{\alpha}$) and add confidence bound.

How to estimate α? Linear regression with graph regularization!

$$\widehat{\boldsymbol{\alpha}}_t = \arg\min_{\omega \in \mathbb{R}^N} \left(\sum_{s=1}^{t-1} \left(x_{A_s}^\mathsf{T} \omega - r_s \right)^2 + \|\omega\|_\Lambda^2 \right) = V_t^{-1} X_t^\mathsf{T} r$$

where

- $X_t = \left[x_{A_1}, \ldots, x_{A_{t-1}} \right]^\mathsf{T}$
- $r = [r_1, \ldots, r_{t-1}]^\mathsf{T}$
- $V_t = X_t X_t^\mathsf{T} + \Lambda$ (recall that $\Lambda = \Lambda_\mathcal{L} + \lambda I$)

SpectralUCB

How to obtain an upper confidence bound?

Lemma ([kocak2020spectral])

With probability at least $1-\delta$*, for any* $x \in \mathbb{R}^N$ *and* $t \geq 1$*,*

$$|x^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t - x^\mathsf{T}\boldsymbol{\alpha}| \leq \|x\|_{V_t^{-1}} \left(\sqrt{2\sigma^2 \log\left(\frac{|V_t|^{1/2}}{\delta|\Lambda|^{1/2}} \right)} + C \right)$$

SpectralUCB

How to obtain an upper confidence bound?

Lemma ([kocak2020spectral])

With probability at least $1-\delta$*, for any* $x \in \mathbb{R}^N$ *and* $t \geq 1$*,*

$$|x^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t - x^\mathsf{T}\boldsymbol{\alpha}| \leq \|x\|_{V_t^{-1}} \left(\sqrt{2\sigma^2 \log\left(\frac{|V_t|^{1/2}}{\delta|\Lambda|^{1/2}}\right)} + C \right)$$

Questions:

- When does it make sense to take $x_a = [0, \ldots, 1, \ldots, 0]$ (non-zero at a-th coordinate)? Hint: think of when $\Lambda_\mathcal{L} = 0$.
- In this case, interpret the inequality above for $x = x_a$. Can you relate it to Hoeffding's inequality?

SpectralUCB

How to obtain an upper confidence bound?

Lemma ([kocak2020spectral])

*Let d be **the effective dimension** and $t \leq T + 1$. Then,*

$$\log\left(\frac{|V_t|}{|\Lambda|}\right) \leq d \log\left(1 + \frac{T}{K\lambda}\right)$$

where K is the number of non-zero eigenvalues of the Laplacian $\Lambda_{\mathcal{L}}$.

SpectralUCB Strategy: Play, in each round t, the action A_t:

$$A_t \in \arg\max_a \left(x_a^{\mathsf{T}} \widehat{\alpha}_t + c\|x_a\|_{V_t^{-1}}\right)$$

where $c = \sigma\sqrt{2d\log\left(1 + \frac{T}{K\lambda}\right) + 8\log\left(\frac{1}{\delta}\right)} + C$.

SpectralUCB

Theorem (SpectralUCB regret bound)

Let d be the **effective dimension** *and λ be the minimum eigenvalue of Λ. If $\|\boldsymbol{\alpha}\|_\Lambda \leq C$, and for all a, $|x_a^\mathsf{T}\boldsymbol{\alpha}| \leq 1$, then, with probability at least $1 - \delta$,*

$$R_T = T \max_a (x_a^\mathsf{T}\boldsymbol{\alpha}) - \sum_{t=1}^{T} x_{A_t}^\mathsf{T}\boldsymbol{\alpha} \leq \widetilde{\mathcal{O}}\left(d\sqrt{T}\right),$$

where $\widetilde{\mathcal{O}}(\cdot)$ omits logarithmic factors.

Question:

- Under which condition on d is SpectralUCB better than UCB?

SpectralUCB - Effective Dimension d

Definition 1: Take d as

$$d = \left\lceil \frac{\max_{t_1,\ldots,t_N : \sum_{t_i} = T} \log \prod_{i=1}^{N} \left(1 + \frac{t_i}{\lambda_i}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil$$

Definition 2: Take $\widetilde{d}$ as the largest integer in $\{1, \ldots, N\}$ such that

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

We can show that $d \leq 2\widetilde{d}$ [**kocak2020spectral**].

SpectralUCB - Effective Dimension d

Definition 1: Take d as

$$d = \left\lceil \frac{\max_{t_1,\ldots,t_N:\sum_{t_i}=T} \log \prod_{i=1}^{N}\left(1+\frac{t_i}{\lambda_i}\right)}{\log\left(1+\frac{T}{K\lambda}\right)} \right\rceil$$

Definition 2: Take $\widetilde{d}$ as the largest integer in $\{1,\ldots,N\}$ such that

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

We can show that $d \leq 2\widetilde{d}$ [**kocak2020spectral**].

Questions: Find examples of graphs where

- $\widetilde{d} \approx N$;
- $\widetilde{d} \ll N$.

Lower Bound for Spectral Bandits

Theorem ([kocak2020spectral])

For any T and d, there exists a problem with effective dimension d and time horizon T such that $\mathbb{E}[R_T] = \Omega(\sqrt{dT})$.

Proof idea: Build a graph with d blocks (or clusters), such that each block behaves as a single action.
Then, the problem behaves as a "classic" d-armed bandit, whose lower bound is $\Omega(\sqrt{dT})$.

Regret of SpectralUCB - Proof Idea

Let $a^* = \arg\max_a(x_a^\mathsf{T}\boldsymbol{\alpha})$ and recall that $R_T = \sum_{t=1}^{T}\left(x_{a^*}^\mathsf{T}\boldsymbol{\alpha} - x_{A_t}^\mathsf{T}\boldsymbol{\alpha}\right)$.

We have, with probability at least $1-\delta$,

$$
\begin{aligned}
x_{a^*}^\mathsf{T}\boldsymbol{\alpha} - x_{A_t}^\mathsf{T}\boldsymbol{\alpha} &\leq x_{a^*}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t + c\|x_{a^*}\|_{V_t^{-1}} - x_{A_t}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t + c\|x_{A_t}\|_{V_t^{-1}} \\
&\leq x_{A_t}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t + c\|x_{A_t}\|_{V_t^{-1}} - x_{A_t}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t + c\|x_{A_t}\|_{V_t^{-1}} \\
&= 2c\|x_{A_t}\|_{V_t^{-1}}
\end{aligned}
$$

Also, by assumption, we have $|x_a^\mathsf{T}\boldsymbol{\alpha}| \leq 1$ for all a, which gives us

$$x_{a^*}^\mathsf{T}\boldsymbol{\alpha} - x_{A_t}^\mathsf{T}\boldsymbol{\alpha} \leq 2.$$

Regret of SpectralUCB - Proof Idea

Consequently,

$$
\begin{aligned}
R_T &\leq \sum_{t=1}^{T} \min\left(2, 2c\|x_{A_t}\|_{V_t^{-1}}\right) \\
&\leq (2+2c)\sqrt{T}\sqrt{\sum_{t=1}^{T} \min\left(1, c\|x_{A_t}\|^2_{V_t^{-1}}\right)} \\
&\ldots \\
&\leq (2+2c)\sqrt{2T\log\left(\frac{|V_{T+1}|}{|\Lambda|}\right)} \\
&\leq (2+2c)\sqrt{2dT\log\left(1+\frac{T}{K\lambda}\right)}.
\end{aligned}
$$

The result follows from the definition of c, which is $\widetilde{\mathcal{O}}\left(\sqrt{d}\right)$.

Influence Maximization

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

Influence Maximization

Now, the reward is the number of **influenced neighbors**

- **Unknown** p_{ij}: probability that i influences j
- At each time t:
  - Choose a node A_t;
  - Observe a set of influenced nodes $S_{A_t,t}$

Select influential nodes = **Find the strategy maximizing**

$$L_T = \sum_{t=1}^{T} |S_{A_t,t}|$$

Influence Maximization

Now, the reward is the number of **influenced neighbors**

- **Unknown** p_{ij}: probability that i influences j
- At each time t:
  - Choose a node A_t;
  - Observe a set of influenced nodes $S_{A_t,t}$

Select influential nodes = **Find the strategy maximizing**

$$L_T = \sum_{t=1}^{T} |S_{A_t,t}|$$

Questions:

- Why is this a bandit problem?
- Can we simply apply UCB?

Performance Criterion

- Number of expected influences of node k:

$$\mu_k = \mathbb{E}[|S_{k,t}|] = \sum_{j=1}^{N} p_{k,j}$$

- An optimal strategy would always select the best node:

$$k^* = \arg\max_k \mathbb{E}\left[\sum_{t=1}^{T} |S_{k,t}|\right] = \arg\max_k \mu_k$$

- Expected regret of an adaptive strategy **unaware** of p_{ij}:

$$\mathbb{E}[R_T] = \mathbb{E}[L_T^*] - \mathbb{E}[L_T]$$

where $L_T^* = T\mu^* = T\mu_{k^*}$.

Baseline: GraphMOSS

- **Only $|S_{A_t,t}|$ is observed** (not the set $S_{A_t,t}$).
- MOSS = **M**inimax **O**ptimal **S**trategy in the **S**tochastic case
  - Strategy is similar to UCB.
  - Improved regret (minimax) when compared to UCB (no log factor).

Baseline: GraphMOSS

- **Only $|S_{A_t,t}|$ is observed** (not the set $S_{A_t,t}$).
- MOSS = **M**inimax **O**ptimal **S**trategy in the **S**tochastic case
  - Strategy is similar to UCB.
  - Improved regret (minimax) when compared to UCB (no log factor).

GraphMOSS Strategy: Play each arm twice, then

$$A_t \in \arg\max_a \left(\widehat{\mu}_a(t) + 2\widehat{\sigma}_a(t)\sqrt{g(N_a(t))} + 2g(N_a(t))\right), \quad \text{for } t > 2N$$

where $g(n) = \frac{1}{n}\max\left(\log\left(\frac{T}{nN}\right), 0\right)$.

- The expected regret of GraphMOSS satisfies:

$$\mathbb{E}[R_T] \leq c \min\left(\mu^* T, \mu^* N + \sqrt{\mu^* NT}\right)$$

- Again, what if $N \geq T$?

Back to the real setting

- Can we do better by observing the set $S_{A_t,t}$?
  - Not in a worst-case scenario!
  - The minimax optimal rate is still the same ($\sqrt{NT}$).
- Hard cases:
  - When there are many isolated nodes.
  - When being influenced is not correlated to being influential.

Back to the real setting

- Can we do better by observing the set $S_{A_t,t}$?
  - Not in a worst-case scenario!
  - The minimax optimal rate is still the same ($\sqrt{NT}$).
- Hard cases:
  - When there are many isolated nodes.
  - When being influenced is not correlated to being influential.
- **How to measure the difficulty of the problem?**
  - Can we find a quantity D^* that replaces N in the regret bound?

Detectable Dimension D^*

Let the *dual influence* of a node k be defined as

$$\mu_k^\circ = \sum_{j=1}^{N} p_{j,k}$$

and $\mu_*^\circ = \max_k \mu_k^\circ$.

The function D gives the number of nodes corresponding to a gap Δ:

$$D(\Delta) = |\{i : \mu_*^\circ - \mu_i^\circ \leq \Delta\}|$$

Detectable Dimension D^*

Let the *dual influence* of a node k be defined as

$$\mu_k^\circ = \sum_{j=1}^{N} p_{j,k}$$

and $\mu_*^\circ = \max_k \mu_k^\circ$.

The function D gives the number of nodes corresponding to a gap Δ:

$$D(\Delta) = |\{i : \mu_*^\circ - \mu_i^\circ \leq \Delta\}|$$

- $D(\Delta) = N$ if $\Delta \geq \mu_*^\circ$.
- $D(0) =$ number of most influenced nodes.

Detectable Dimension D^*

Let $T_* > 1$ be the smallest integer such that

$$T_* \mu_*^\circ \geq \sqrt{\boxed{D\left(16\sqrt{\frac{\mu_*^\circ N \log(NT)}{T_*}} + \frac{144 N \log(NT)}{T_*}\right)} T \mu_*^\circ}$$

Then, D^* is defined as:

$$D^* = D\left(16\sqrt{\frac{\mu_*^\circ N \log(NT)}{T_*}} + \frac{144 N \log(NT)}{T_*}\right)$$

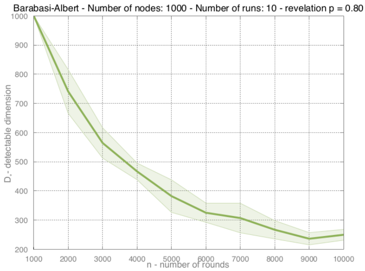
Questions:

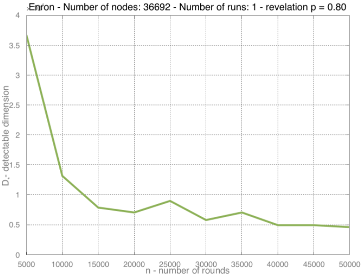
- What is D^* for a star graph?
- What is D^* for an empty graph?

Detectable Dimension D^*

$$D^* = D\left(16\sqrt{\frac{\mu_*^\circ N \log(NT)}{T_*}} + \frac{144 N \log(NT)}{T_*}\right)$$

- As $T \to \infty$, D^* converges to the number of most influenced nodes.
- The graph structure is helpful when D decreases quickly with T.





Algorithm: BARE

We have D^*, a measure of difficulty. **But what's the algorithm?**

BARE: **Ba**ndit **Re**velator, a two-phase algorithm [CV16]

- **global exploration phase**
  - Efficient exploration: sample random nodes
  - Linear regret $\implies$ needs to be short
  - Extracts D^* nodes
- **bandit phase**
  - Uses a minimax-optimal bandit algorithm: GraphMOSS

Algorithm: BARE

We have D^*, a measure of difficulty. **But what's the algorithm?**

BARE: **Ba**ndit **Re**velator, a two-phase algorithm [CV16]

- **global exploration phase**
  - Efficient exploration: sample random nodes
  - Linear regret $\implies$ needs to be short
  - Extracts D^* nodes
- **bandit phase**
  - Uses a minimax-optimal bandit algorithm: GraphMOSS

Intuition:

- First, learn the **set of most influenced nodes**;
- Then, run a bandit algorithm on these nodes.

The regret is small **if the set of most influenced nodes contains the most influential nodes!**

Algorithm: BARE

- **Input:** N (number of nodes), T (time horizon)
- **Initialization:**
  - $t \leftarrow 1, \quad \widehat{\mu_k^\circ}(t) \leftarrow 0, \quad \widehat{\sigma_*}(t) \leftarrow N, \quad \widehat{D}_*(t) \leftarrow N$
- **Global exploration:** while
  $t\left(\widehat{\sigma_*}(t) - 4\sqrt{N\log(NT)/t}\right) \leq \sqrt{\widehat{D}_*(t)T}$
  - Influence a note A_t at random and observe $S_{A_t,t}$;
  - $\widehat{\mu_k^\circ}(t+1) \leftarrow \frac{t}{t+1}\widehat{\mu_k^\circ}(t) + \frac{N}{t+1}S_{A_t,t}(k)$
  - $\widehat{\sigma_*}(t+1) \leftarrow \max_{k'} \sqrt{\widehat{r_{k'}^\circ}(t+1) + 8N\log(NT)/(t+1)}$
  - $w_*(t+1) \leftarrow \widehat{\sigma_*}(t+1)\sqrt{\frac{N\log(NT)}{t+1}} + \frac{40N\log(NT)}{t+1}$
  - $\widehat{D}_*(t+1) \leftarrow \left|\left\{k : \max_{k'} \widehat{\mu_{k'}^\circ}(t+1) - \widehat{\mu_k^\circ}(t+1) \leq w_*(t+1)\right\}\right|$
  - $t \leftarrow t+1$
- **Bandit phase:** run GraphMOSS on the $\widehat{D}_*(t)$ chosen nodes.

BARE: Regret Bound

Let $\mathcal{D}^\circ = \{i : \mu_i^\circ = \max_k \mu_k^\circ\}$ be the set of most influenced nodes.

The influential-influenced gap is defined as:

$$\varepsilon^* = \mu^* - \max_{k \in \mathcal{D}^\circ} \mu_k$$

Theorem ([CV16])

The expected regret of BARE satisfies

$$\mathbb{E}[R_T] \leq c \min\left(\mu^* T, D^* \mu^* + \sqrt{\mu^* D^* T} + \boxed{T\varepsilon^*}\right)$$

BARE: Regret Bound

Let $\mathcal{D}^\circ = \{i : \mu_i^\circ = \max_k \mu_k^\circ\}$ be the set of most influenced nodes.

The influential-influenced gap is defined as:

$$\varepsilon^* = \mu^* - \max_{k \in \mathcal{D}^\circ} \mu_k$$

Theorem ([CV16])

The expected regret of BARE satisfies

$$\mathbb{E}[R_T] \le c \min\left(\mu^* T, D^* \mu^* + \sqrt{\mu^* D^* T} + \boxed{T\varepsilon^*}\right)$$

Question:

- For which types of graphs is $\varepsilon^* = 0$?

Conclusion

Conclusion

Multi-armed bandits

- Model several interactive learning scenarios.
- Regret bounds depend on the number of possible actions N.
- If $N > T$, we can't learn anything, **unless we have more structure!**

Spectral bandits

- The reward function is smooth on a graph;
- The problem complexity depends on **an effective dimension** d.

Influence maximization

- The observations are richer: **set of influenced nodes**.
- The complexity becomes a function of a **detectable dimension** D^*.
- We still need correlation between being influential and being influenced.

Daniele Calandriello

dcalandriello@google.com

ENS Paris-Saclay, MVA 2022/2023

https://sites.google.com/view/daniele-calandriello/