Lecture 3: Regret Minimisation in Bandits

ID 6002W: Online and Reinforcement Learning

Web-enabled M.Tech. program, IIT Madras

Suryanarayana Sankagiri

19th May, 2026

Where We Are:
Best-Arm Identification in MAB via Confidence Intervals

Hoeffding's Inequality

For Confidence Intervals

$$2\exp(-2n\epsilon^2) \leq \delta \Rightarrow \epsilon \geq \sqrt{\frac{\log(2/\delta)}{2n}}$$

With probability $\geq 1-\delta$:

$$\mu \in \left[\hat{\mu} - \sqrt{\frac{\log(2/\delta)}{2n}}, \hat{\mu} + \sqrt{\frac{\log(2/\delta)}{2n}}\right]$$

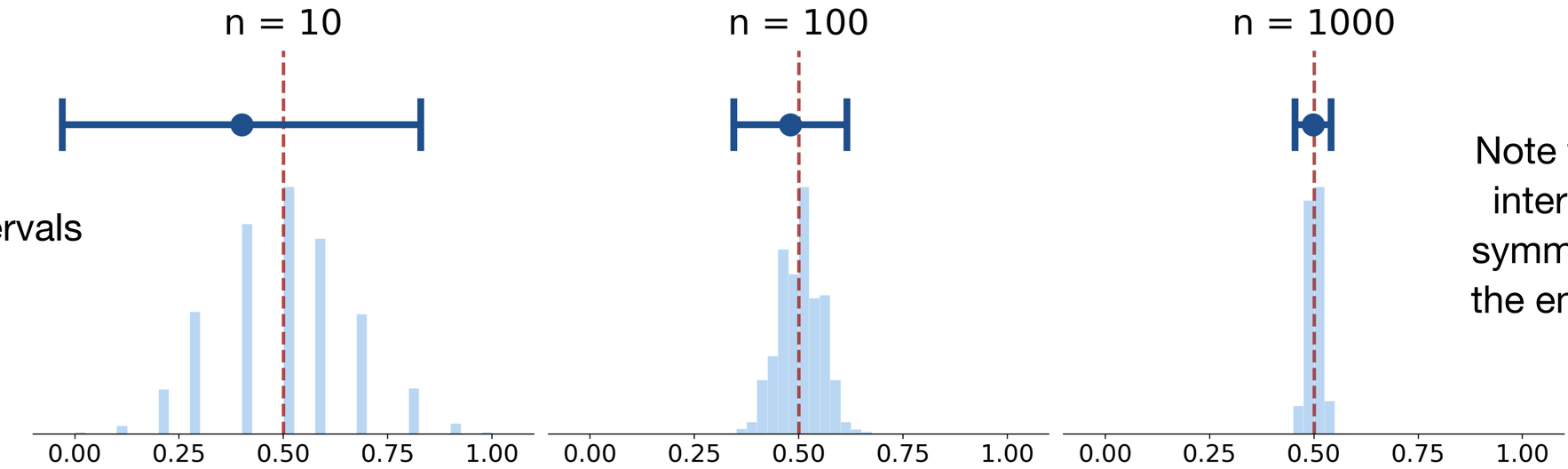
Hoeffding's Inequality

For Confidence Intervals

$$2\exp(-2n\epsilon^2) \leq \delta \Rightarrow \epsilon \geq \sqrt{\frac{\log(2/\delta)}{2n}}$$

With probability $\geq 1-\delta$:

$$\mu \in \left[\hat{\mu} - \sqrt{\frac{\log(2/\delta)}{2n}}, \hat{\mu} + \sqrt{\frac{\log(2/\delta)}{2n}}\right]$$



95 % confidence intervals

$\delta = 0.05$

Note that confidence intervals are drawn symmetrically around the empirical average

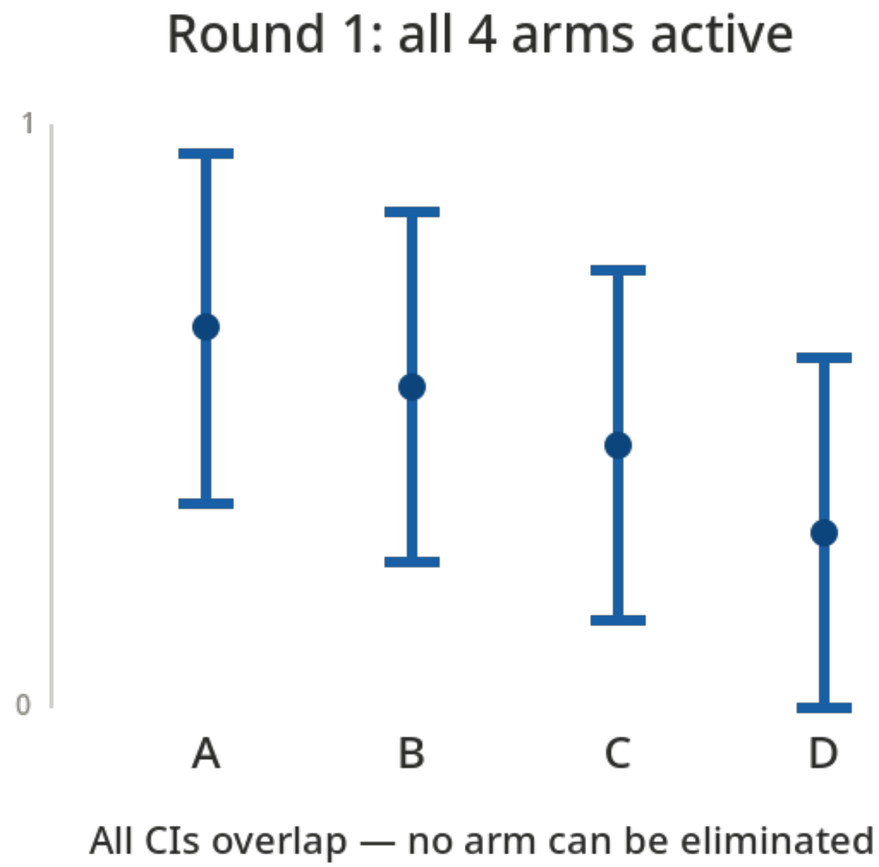
Successive Elimination

The Key Idea

We have K arms. We want to find the best one with the following strategy.

Maintain a set of **active arms** (initially, all K).
At each round t

- pull every active arm once
- Update *empirical means*, *confidence intervals*



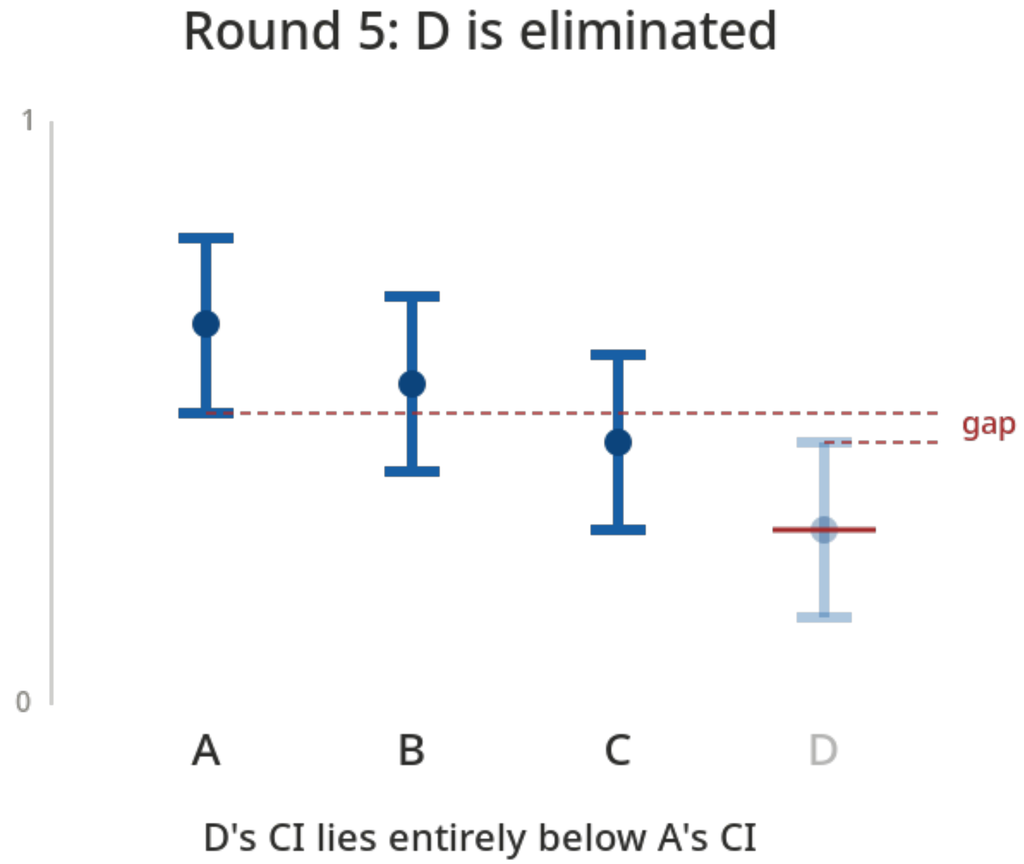
Successive Elimination

The Key Idea

We have K arms. We want to find the best one with the following strategy.

Maintain a set of **active arms** (initially, all K).
At each round t

- pull every active arm once
- Update *empirical means*, *confidence intervals*
- Eliminate any arm whose confidence interval lies *entirely below* another arm's interval



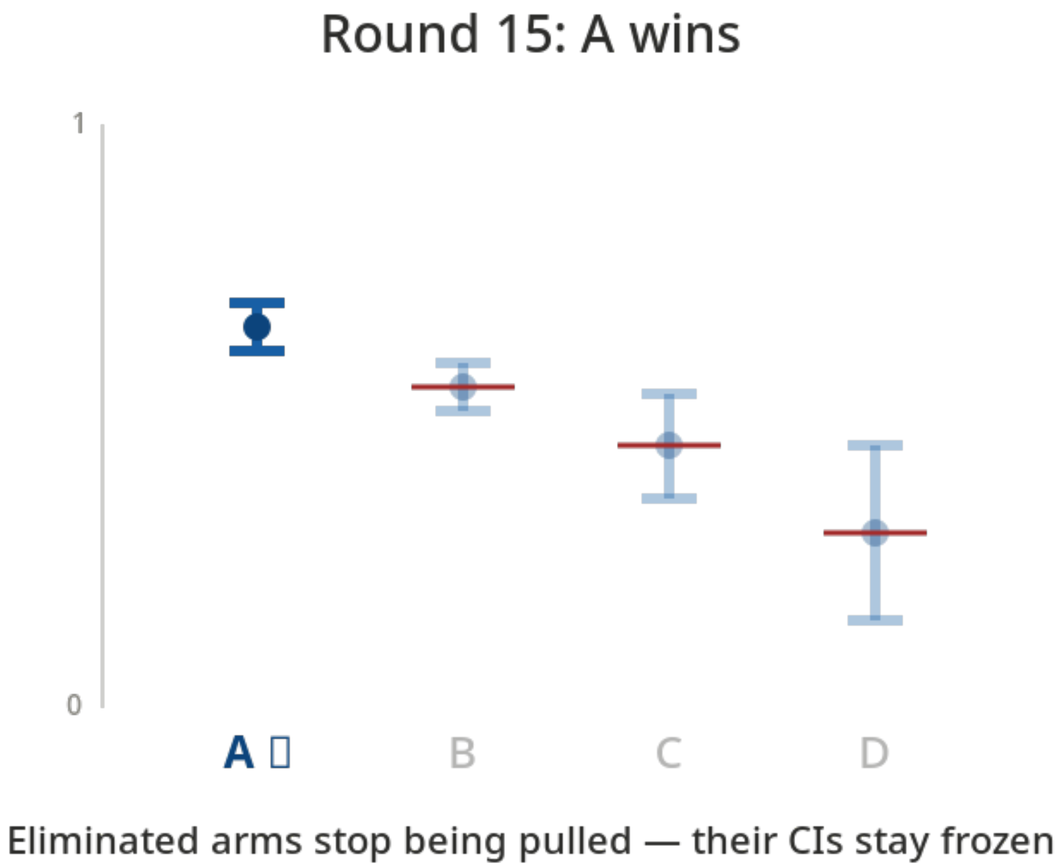
Successive Elimination

The Key Idea

We have K arms. We want to find the best one with the following strategy.

Maintain a set of **active arms** (initially, all K).
At each round t

- pull every active arm once
- Update *empirical means*, *confidence intervals*
- Eliminate any arm whose confidence interval lies *entirely below* another arm's interval
- Stop when only one arm remains



Goal For Today:
A Gentle Introduction to Regret Minimisation

Regret

A metric for long-term performance

Recall the definition of regret from Lecture 1:

- If arm distributions were known, then we can always pull arm i^*
  - Get expected reward $T\mu^*$
- Regret = what we missed out on.

$$R^{\text{actual}}(T) = T\mu^* - \sum_{t=1}^{T} X_t$$

Regret is a random quantity, because both rewards and actions are random

Expected Regret

Cleaner than actual regret

- To simplify analysis, we use expected regret in analysing algorithms:

$$R(T) = \mathbb{E}[R^{\text{actual}}(T)] = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} X_t\right]$$

- Henceforth, we will call this quantity as <u>regret</u>

Expected Regret

Cleaner than actual regret

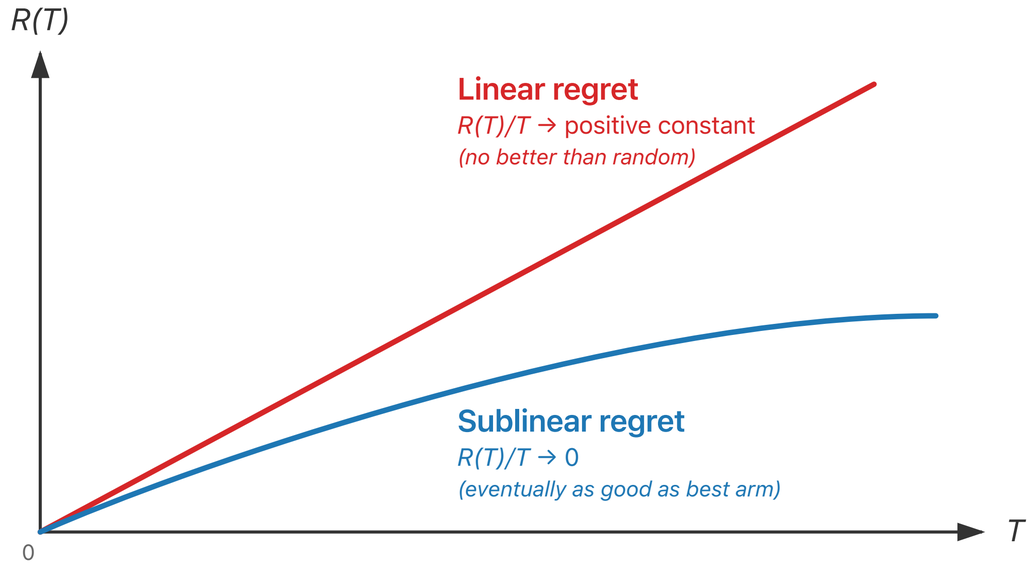
- To simplify analysis, we use expected regret in analysing algorithms:

$$R(T) = \mathbb{E}[R^{\text{actual}}(T)] = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} X_t\right]$$

- Henceforth, we will call this quantity as <u>regret</u>

A good algorithm has sublinear regret!

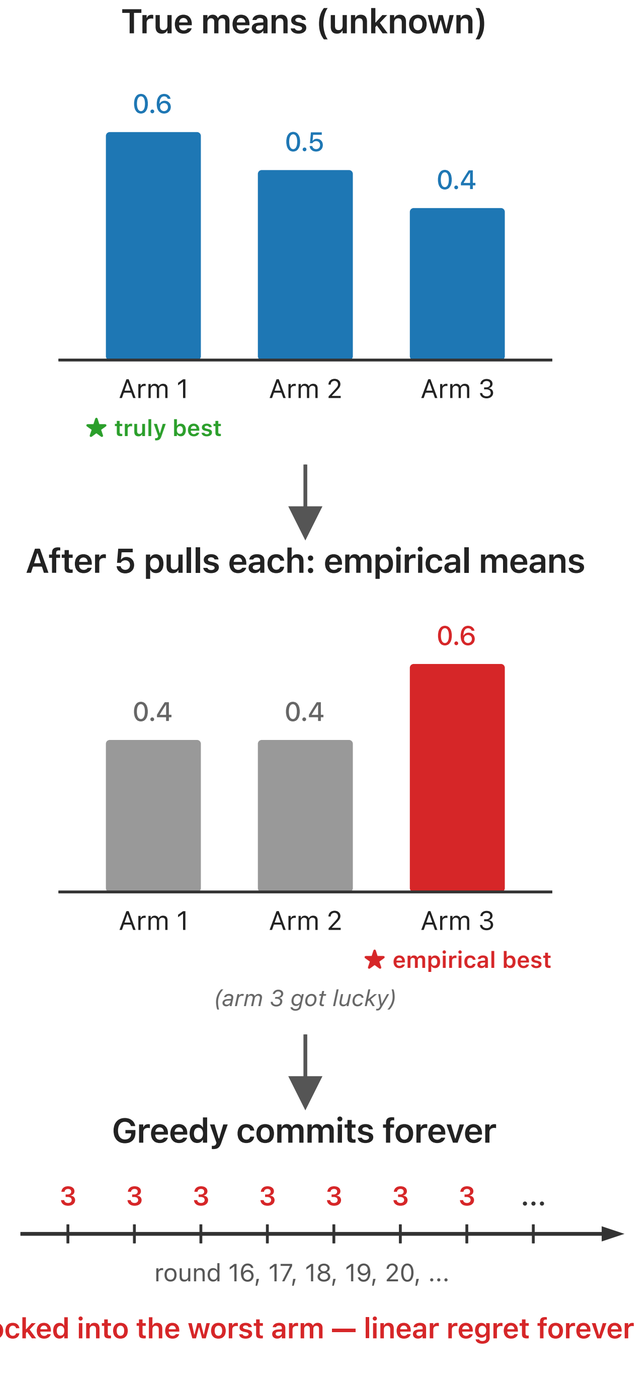
- Equivalently, per-round regret vanishes as T grows
- Algorithm eventually behaves like best arm



Two Naive Strategies

Pure Greedy

- Pull each arm a few times (n)
  - Calculate empirical mean of each arm $\hat{\mu}_i = (\Sigma_{t:A_t=i} X_t)/n$
  - Forever pull the empirically best arm: $\arg \max_{i \in \{1,2,\ldots,K\}} \hat{\mu}_i$



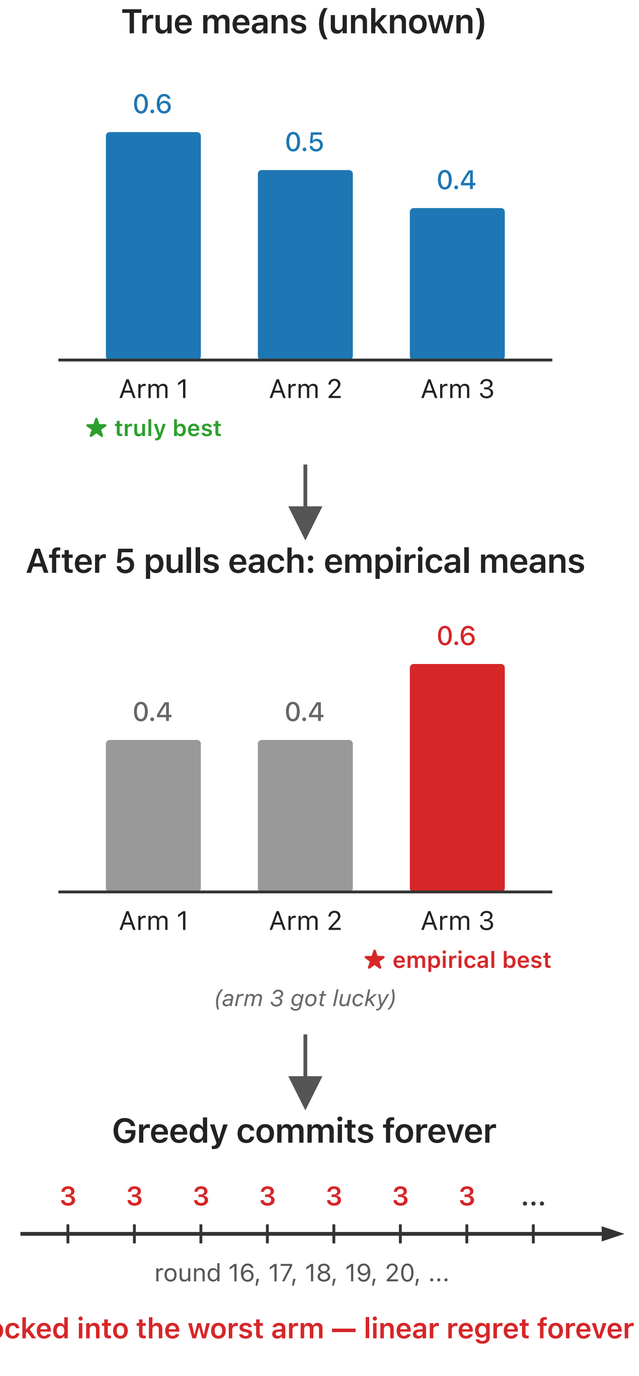
Two Naive Strategies

Pure Greedy

- Pull each arm a few times (n)
  - Calculate empirical mean of each arm $\hat{\mu}_i = (\Sigma_{t:A_t=i} X_t)/n$
  - Forever pull the empirically best arm: $\arg \max_{i \in \{1,2,\ldots,K\}} \hat{\mu}_i$

Pure Random

- Pull a uniformly random arm at every round
  - Equivalently, pull arms in round-robin fashion
- $\hat{\mu}_i \to \mu_i$, but regret grows linearly in T



ε-Greedy Algorithm

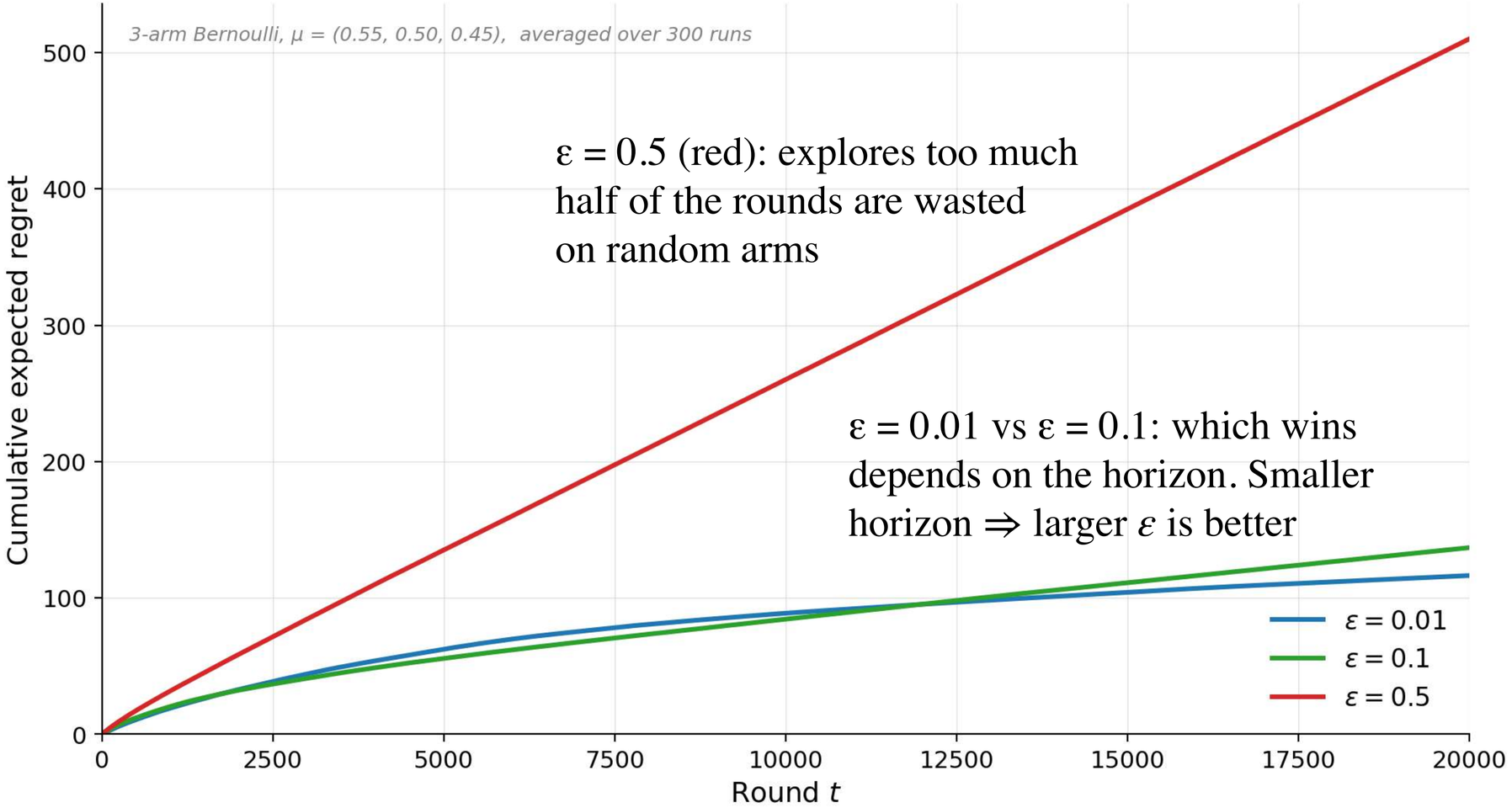
A better solution

- With probability ε: pull a uniformly random arm (explore)
- With probability $1 - \varepsilon$: pull the empirically best arm (exploit)
- $\varepsilon \in (0,1)$ is the exploration rate — a knob you tune

The simplest algorithm that does both exploration and exploitation by combining pure random and pure greedy strategies

ε-Greedy Algorithm

Behaviour and Limitations



ε = 0.5 (red): explores too much
half of the rounds are wasted
on random arms

ε = 0.01 vs ε = 0.1: which wins
depends on the horizon. Smaller
horizon ⇒ larger ε is better

All three curves
keep growing —
any fixed ε gives
linear regret

No fixed ε is universally best

For fixed ε, even after we know the best arm, we still explore forever. This leads to linear regret

What if we choose a fixed exploration time, estimate mean rewards, and then commit to best arm?

That is the Explore-Then-Commit algorithm

Where Does Regret Come From?

A useful decomposition

Recall that regret is defined as $R(T) = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} X_t\right]$

- Let A_t denote the action taken in round t
  - Then $\mathbb{E}[X_t \mid A_t] = \mu_{A_t}$

Where Does Regret Come From?

A useful decomposition

Recall that regret is defined as $R(T) = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} X_t\right]$

- Let A_t denote the action taken in round t
  - Then $\mathbb{E}[X_t \mid A_t] = \mu_{A_t}$
- However, A_t is also random!
  - It usually depends on past rewards, which are random
  - So $\mathbb{E}\left[X_t\right] = \mathbb{E}\left[\mu_{A_t}\right] \Rightarrow R(T) = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} X_t\right] = \mathbb{E}\left[\sum_{t=1}^{T} (\mu^* - \mu_{A_t})\right]$

Where Does Regret Come From?

A useful decomposition

Can we simplify the expression $R(T) = \mathbb{E}\left[\sum_{t=1}^{T}(\mu^* - \mu_{A_t})\right]$ further?

- Define pull count of arm i: $T_i(T) \Rightarrow \sum_{i=1}^{K} T_i(T) = T$

Where Does Regret Come From?

A useful decomposition

Can we simplify the expression $R(T) = \mathbb{E}\left[\sum_{t=1}^{T}(\mu^* - \mu_{A_t})\right]$ further?

- Define pull count of arm i: $T_i(T) \Rightarrow \sum_{i=1}^{K} T_i(T) = T$
- If $T_i(T)$ are known for all i,

$$\sum_{t=1}^{T}(\mu^* - \mu_{A_t}) = \sum_{i=1}^{K}\Delta_i T_i(T)$$

Where Does Regret Come From?

A useful decomposition

Can we simplify the expression $R(T) = \mathbb{E}\left[\sum_{t=1}^{T}(\mu^* - \mu_{A_t})\right]$ further?

- Define pull count of arm i: $T_i(T) \Rightarrow \sum_{i=1}^{K} T_i(T) = T$
- If $T_i(T)$ are known for all i,

$$\sum_{t=1}^{T}(\mu^* - \mu_{A_t}) = \sum_{i=1}^{K}\Delta_i T_i(T)$$

- Taking expectation: $R(T) = \sum_{i=1}^{K}\Delta_i \mathbb{E}[T_i(T)]$

To minimise regret, we must minimise how often we pull arms with large gaps

Explore-Then-Commit

Basic Idea

ε-greedy mixes exploration and exploitation at every round.
What if we separate them in time?

Phase 1 (Explore)

Pull each arm m times,
calculate empirical averages $\hat{\mu}_1, \ldots, \hat{\mu}_K$

→

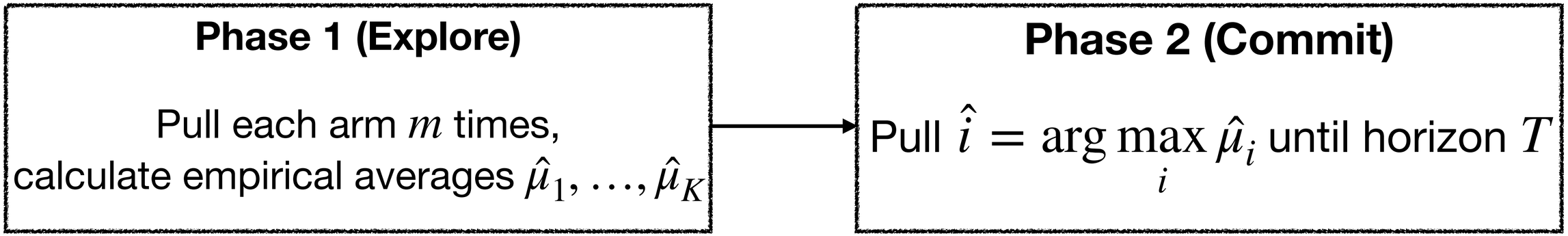
Phase 2 (Commit)

Pull $\hat{i} = \arg\max_i \hat{\mu}_i$ until horizon T

Explore-Then-Commit

Basic Idea

ε-greedy mixes exploration and exploitation at every round.
What if we separate them in time?

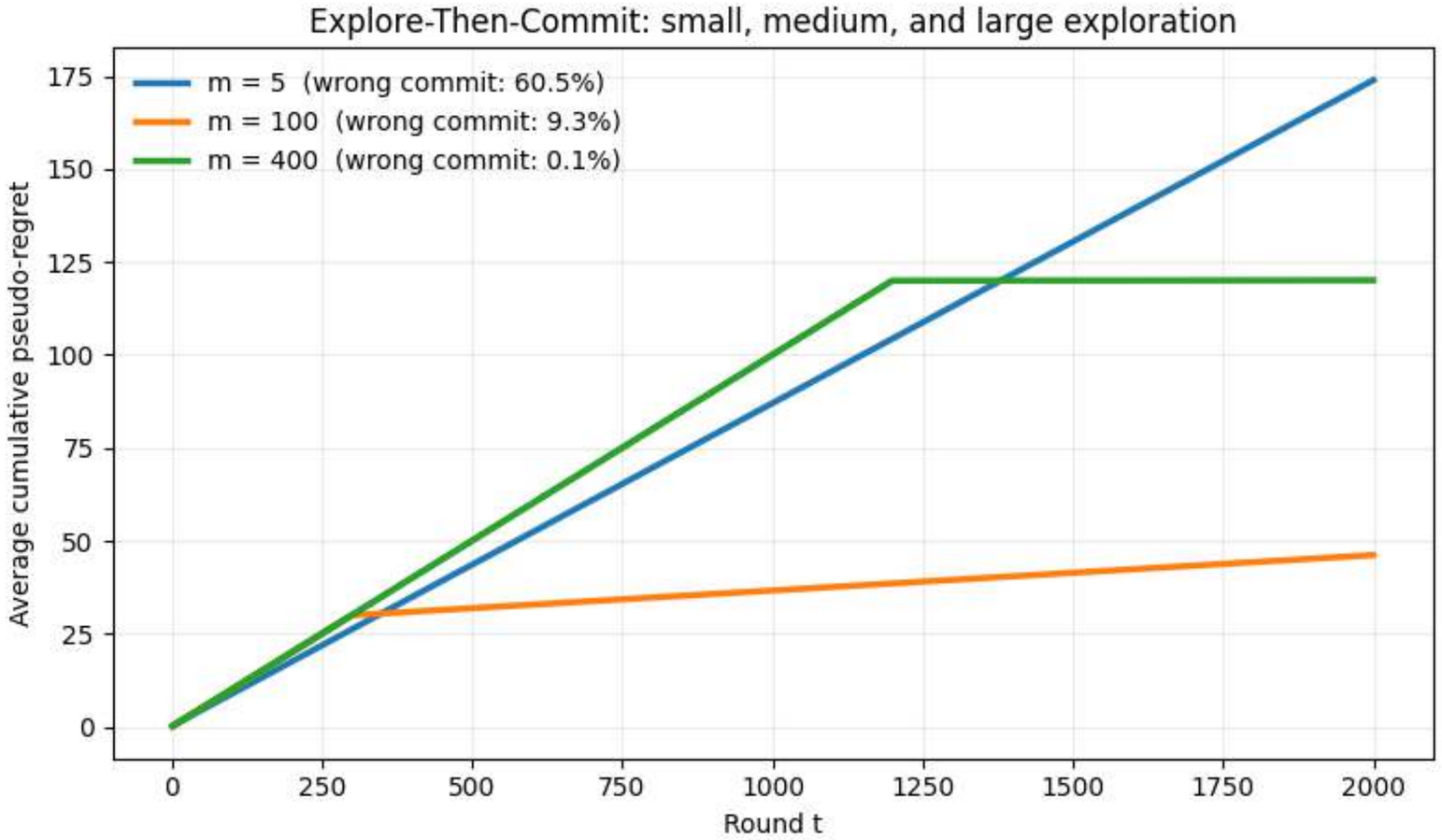


Key parameter: m (exploration budget per arm). Total exploration length: Km

Like Sequential Elimination (SE), it uses samples to identify the best arm.
Unlike SE, stopping time is fixed in advance

Explore-Then-Commit

Visualisation



- When m is small: many runs commit to wrong arm
  $\Rightarrow$ large, linearly growing regret

- When m is large: long exploration phase causes regret to blow up

Medium m balances both costs

Regret comes from two factors: initial exploration (phase 1) + potentially wrong-commit (phase 2)

Explore-Then-Commit

Choosing the right m

- Consider a two-armed bandit with $|\mu_1 - \mu_2| = \Delta$

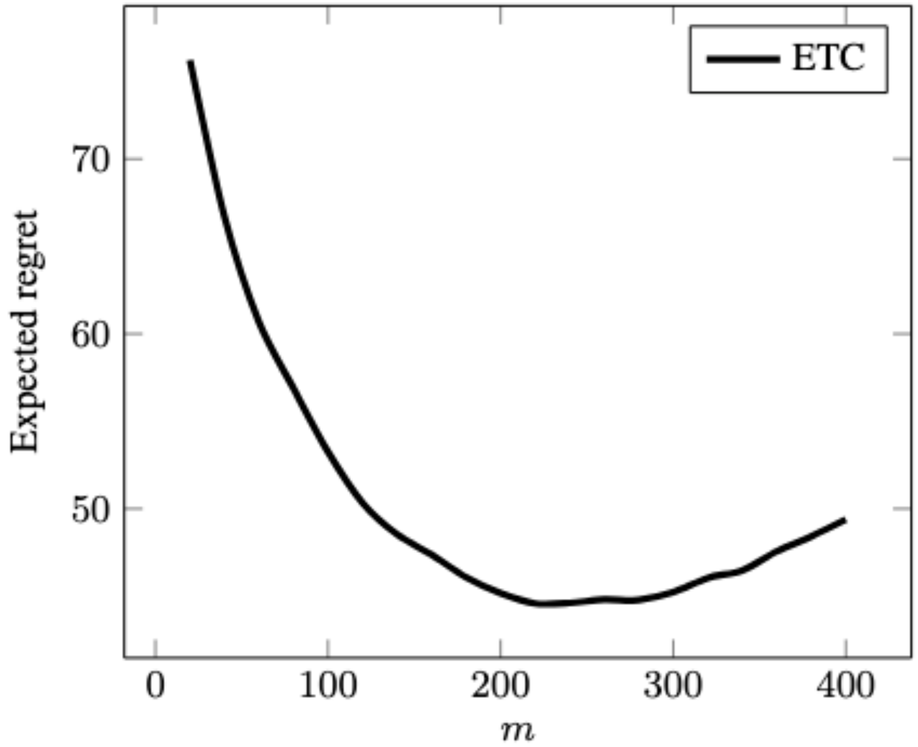


Figure 6.2 from Lattimore & Szepesvari

Figure shows regret as a function of m for a two-armed Gaussian bandit, $\Delta = 0.1$

Explore-Then-Commit

Choosing the right m

- Consider a two-armed bandit with $|\mu_1 - \mu_2| = \Delta$
- Recall: to distinguish two arms of gap Δ with probability $\geq 1 - \delta$, we need
  $$m \sim \log(1/\delta)/\Delta^2 \text{ samples}$$
- We choose $\delta \propto 1/T \Rightarrow m \sim \dfrac{\log T}{\Delta^2}$
  - Ensures expected regret from phase 2 is small

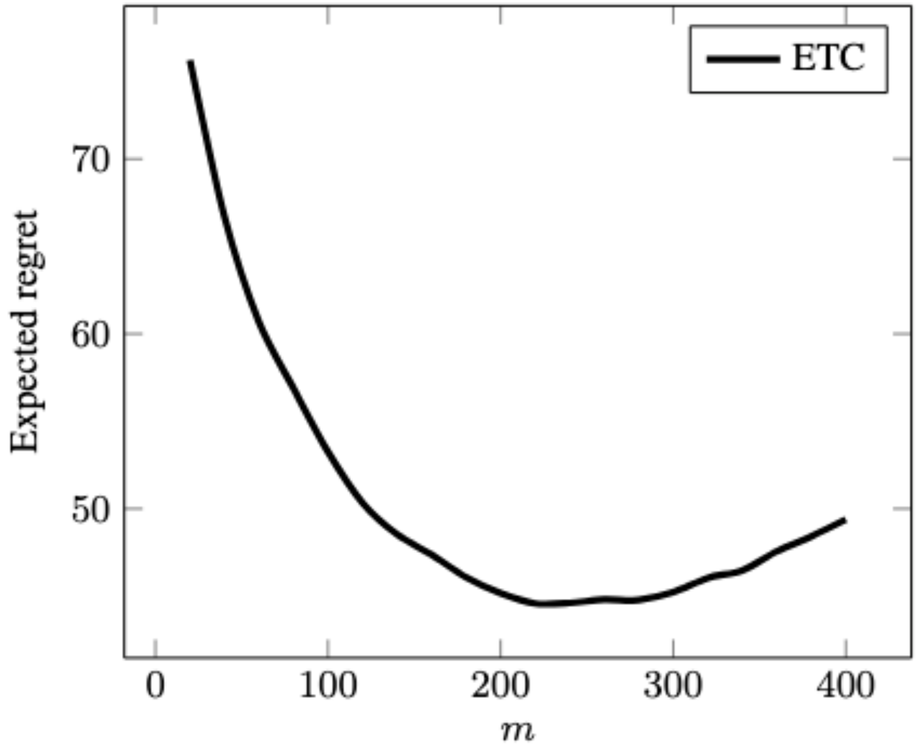


Figure 6.2 from Lattimore & Szepesvari

Figure shows regret as a function of m for a two-armed Gaussian bandit, $\Delta = 0.1$

Explore-Then-Commit

Choosing the right m

- Consider a two-armed bandit with $|\mu_1 - \mu_2| = \Delta$
- Recall: to distinguish two arms of gap Δ with probability $\geq 1 - \delta$, we need
  $$m \sim \log(1/\delta)/\Delta^2 \text{ samples}$$
- We choose $\delta \propto 1/T \Rightarrow m \sim \dfrac{\log T}{\Delta^2}$
  - Ensures expected regret from phase 2 is small
- Exploration regret $= m\Delta \sim \dfrac{\log T}{\Delta}$
  $$\Rightarrow R(T) \sim \log T/\Delta$$

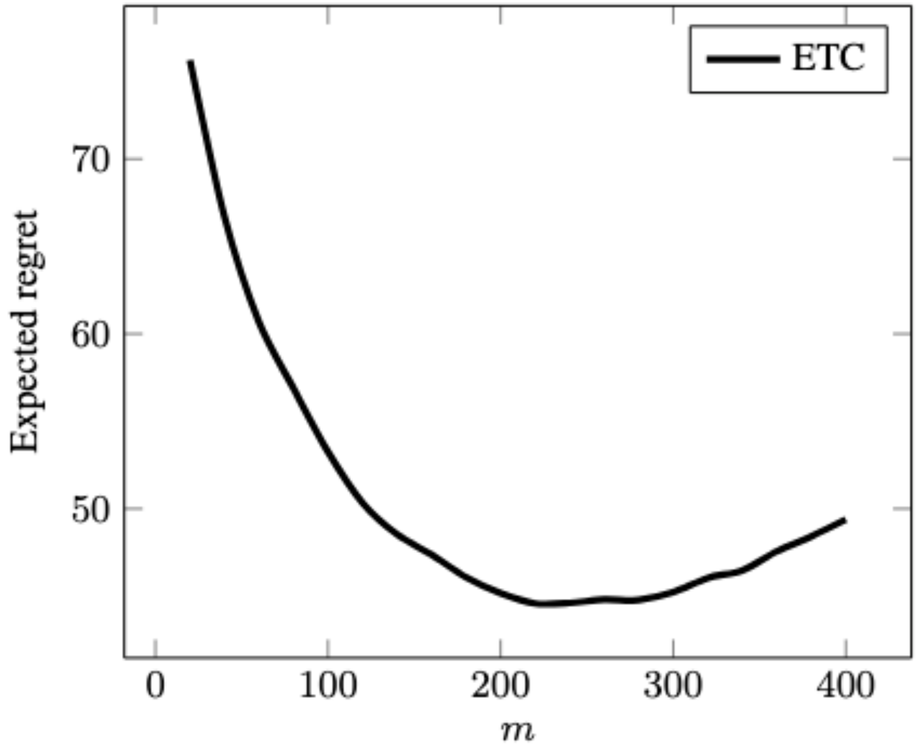


Figure 6.2 from Lattimore & Szepesvari

Figure shows regret as a function of m for a two-armed Gaussian bandit, $\Delta = 0.1$

Explore-Then-Commit

Limitations

If Δ is known:

- Choose $m \sim \log T / \Delta^2$, suffer regret $R(T) \sim \log T / \Delta$

If Δ is unknown:

- Can choose m conservatively to ensure $R(T) \sim T^{2/3}$
- But it does not adapt to easy problems where best arm is clear

Can we avoid choosing m at all?

Need to slowly reduce the extent of exploration over time, using confidence intervals

The Upper Confidence Bound (UCB) Algorithm smoothly trades off exploration and exploitation, giving regret $\log T/\Delta$ without knowing Δ