

# CMPT 419/983: Theoretical Foundations of Reinforcement Learning

## Lecture 1

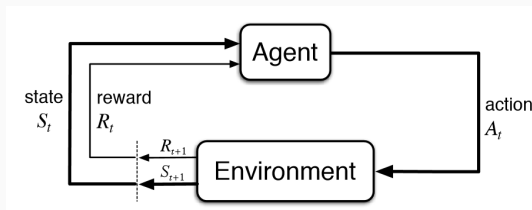
---

Sharan Vaswani

September 8, 2023

- Supervised machine learning involves learning from a fixed, static dataset.
- Once a dataset is collected, supervised learning does not typically reason about how the data was acquired nor does it involve further interactions with the world.
- Applications in computational advertising, robotics, clinical trials involve collecting data in an online fashion, and reasoning about the decisions used to gather it.
- **Sequential decision-making** under uncertainty focuses on problems that involve interacting with the world, collecting data and reasoning about it, all with incomplete information about the world.

# Introduction



- A typical problem in sequential decision-making involves an *agent* (e.g: marketer, robot, investor) sequentially interacting with the *environment* (e.g: online advertising platform, Mars terrain, stock market).
- An interaction involves the agent choosing an *action* and receiving feedback.
- For example, the feedback can be in the form of a *reward*) designed to measure the agent's performance in achieving its goal.
- One possible objective: Find a sequence of actions (referred to as a *policy*) that maximizes the *cumulative reward* across the sequence of interactions.

# Motivating Applications

- Games. E.g: Go and Atari by DeepMind.
- Conversational agents. Eg: ChatGPT by OpenAI.
- Chip design by Google AI
- Cooling the interior of large commercial buildings by DeepMind
- Recommendation system by Microsoft
- Healthcare and Clinical Trials.
- Autonomous Navigation of Stratospheric Balloons by Google AI.
- For more applications, refer to Glen Berseth's and Csaba Szepesvari's lists.

## Motivation

- Typical algorithms used in practice are often (a) brittle (their performance is sensitive to hyper-parameters) (b) inefficient (require a large number of interactions to learn to make good decisions) and (c) do not have theoretical guarantees on their performance and can fail on simple problems.
- Numerous fundamental theoretical questions remain unanswered and there is a large discrepancy between the theory and practice.

## Objective:

- Understand the foundational concepts in bandits and reinforcement learning (RL) from a theoretical perspective.
- Use this knowledge to inform the design of theoretically-principled, statistically and computationally efficient algorithms.

## Topics:

- **Bandits:** Multi-armed/Contextual Bandit framework, Algorithms for regret minimization
- **Markov Decision Processes:** Structural properties, (Approximate) Value/Policy Iteration, Linear Programming, Temporal Difference Learning, Policy Gradients
- **Online & Batch RL:** Q Learning, LSVI-UCB, Learning with access to a simulator

**What we won't cover:** Continuous state-action spaces, Constrained MDPs, Multi-objective RL

- **Instructor:** Sharan Vaswani. [sharan\_vaswani@sfu.ca]
- **Teaching Assistant:** Michael Lu. [michael\_lu\_3@sfu.ca]
- **Course Webpage:** [https://vaswanis.github.io/419\\_983-F23.html](https://vaswanis.github.io/419_983-F23.html)
- **Piazza:** <https://piazza.com/sfu.ca/fall2023/cmpt419983/home>
- **Prerequisites:** Probability, Linear Algebra, Calculus, Undergraduate Machine Learning

# Course Logistics – Grading

## Assignments $[4 \times 12\% = 48\%]$

- Assignments to be submitted online (via Coursys), typed up in Latex with accompanying code submitted as a zip file.
- Each assignment will be due in 3 weeks (at 11.59 pm PST).

## Final Project [50%]

- Aim is to give you a taste of research in RL Theory.
- Projects to be done in groups of 3-4. Will maintain a list of possible topics. Can choose from the list or propose your own topic. (more details will be on Piazza)
- Project Proposal [10%] – Discussion (before 20 October) + Report (due 20 October)
- Project Milestone [5%] – Update (before 20 November)
- Project Presentation [10%] – (*tentatively* 1, 4 December)
- Project Report [25%] (15 December)

## Participation [2%] In class (during lectures, project presentations), on Piazza

# Stochastic Multi-armed Bandits



# Motivating Application: Clinical Trials

- Do not have complete information about the effectiveness or side-effects of the drugs.
- **Aim:** Maximize the number of patients healed.
- Each drug choice is mapped to an *arm* and the drug's effectiveness is mapped to the arm's *mean reward*.
- Administering a drug is an *action* that is equivalent to *pulling* the corresponding arm.
- Each time an arm is pulled, we get a *noisy* reward that models a patient's reaction to the drug.
- The trial goes on for  $T$  rounds.
- Other motivating applications: Recommendation systems, computational advertising.



# Problem Formulation

**Input:**  $K$  arms (possible actions) and their corresponding unknown reward distributions  $\{\nu_a\}_{a=1}^K$ . Define  $\mu_a := \mathbb{E}_{r \sim \nu_a}[r]$  as the expected reward obtained by choosing action  $a$ .

---

**Algorithm** Generic Bandit Framework ( $K$  arms,  $T$  rounds)

---

- 1: **for**  $t = 1 \rightarrow T$  **do**
  - 2:   **SELECT:** Use a bandit algorithm to decide which arm(s) to pull.
  - 3:   **OBSERVE:** Pull the selected arm  $a_t \in [K]$  and observe reward  $R_t \sim \nu_{a_t}$ .
  - 4:   **UPDATE:** Update the estimated reward for arm  $a_t$ .
  - 5: **end for**
- 

**Bandit Feedback:** Can only observe the noisy reward  $R_t$  from the pulled arm  $a_t$ .

**Objective:** Maximize  $\mathbb{E}[\sum_{t=1}^T R_t]$  where the expectation is over both the randomness of the algorithm (if any) and the distribution of rewards.

Bandit problems are a special case of RL, and capture a lot of the underlying intricacy.

# Problem Formulation

- Define  $a^* := \arg \max_{a \in [K]} \mu_a$  as the *best or optimal arm* in hindsight, and  $\mu^* := \max_a \mu_a$ .
- Maximizing cumulative rewards  $\implies$  Select  $a^*$  as much as possible  $\implies$  Minimize the *cumulative regret*.
- **Cumulative Regret:**  $\text{Regret}(T) := \sum_{t=1}^T [\mu^* - \mathbb{E}[R_t]] = T\mu^* - \sum_{t=1}^T \mathbb{E}[R_t]$ .
- Since the optimal arm is unknown, the algorithm needs to *explore* to narrow down on the best arm. If we can identify the best arm, the algorithm should *exploit* and always choose it.
- Need to find a *policy* that trades off exploration and exploitation to minimize  $\text{Regret}(T)$ .
- Ideally, want  $\text{Regret}(T) = o(T)$  i.e. the regret grows sub-linearly with  $T$ , meaning that  $\lim_{T \rightarrow \infty} \frac{\text{Regret}(T)}{T} = 0$ .

# Regret Decomposition

**Claim:** If  $\Delta_a := \mu^* - \mu_a$  and  $N_a(T)$  is the number of times arm  $a$  was chosen *until* round  $T$ , then,

$$\text{Regret}(T) = \sum_{a=1}^K \Delta_a \mathbb{E}[N_a(T)].$$

*Proof:*

$$\text{Regret}(T) = \mu^* T - \sum_{t=1}^T \mathbb{E}[R_t] = \mu^* T - \sum_{t=1}^T \mathbb{E}[\mu_{a_t}] = \sum_{t=1}^T \mathbb{E}[\mu^* - \mu_{a_t}]$$

(Taking the expectation w.r.t to the reward distribution)

$$= \sum_{a=1}^K [\mu^* - \mu_a] \mathbb{E} \left[ \sum_{t=1}^T \mathcal{I} \{a_t = a\} \right] = \sum_{a=1}^K [\mu^* - \mu_a] \mathbb{E}[N_a(T)]$$

$$\implies \text{Regret}(T) = \sum_{a=1}^K \Delta_a \mathbb{E}[N_a(T)].$$

- Hence, to minimize the regret, an algorithm should (i) not pull arms with  $\Delta_a > 0$  too often (**exploit**) which requires (ii) estimating the values of  $\Delta_a$  to sufficient accuracy (**explore**).

# Naive Strategy

---

**Algorithm** Naive Strategy

---

- 1: **for**  $t = 1 \rightarrow K$  **do**
  - 2:   Select arm  $a_t = t$  and observe reward  $R_t$
  - 3: **end for**
  - 4: Calculate empirical mean reward for arm  $a \in [K]$  as  $\hat{\mu}_a(K) := \frac{\sum_{t=1}^K R_t \mathbb{I}\{a_t=a\}}{N_a(K)}$
  - 5: **for**  $t = K + 1 \rightarrow T$  **do**
  - 6:   Pull arm  $\hat{a} := \arg \max_{a \in [K]} \hat{\mu}_a(K)$  (*choose lower-indexed arm if there is a tie*).
  - 7: **end for**
- 

**Q:** Will this naive strategy result in sublinear regret? **Ans:** No! Consider  $K = 2$ ,  $\mu_1 = \frac{1}{2}$ ,  $\mu_2 = \frac{1}{4}$  and  $\nu_a$  to be Bernoulli distributions, i.e.  $R_t \in \{0, 1\}$ .

$$\mathbb{E}[N_2(T)] = 1 + (T - 2) \Pr[\hat{\mu}_2 > \hat{\mu}_1] + (0) \Pr[\hat{\mu}_2 \leq \hat{\mu}_1] = 1 + (T - 2) \frac{1}{8} = \frac{T}{8} + \frac{3}{4}$$

$$\text{Regret}(\text{Naive}, T) = \left(\frac{1}{2} - \frac{1}{4}\right) \left(\frac{T}{8} + \frac{3}{4}\right) = \frac{3}{16} + \frac{T}{32}.$$

Hence, the Naive strategy will suffer  $O(T)$  regret and  $\lim_{T \rightarrow \infty} \frac{\text{Regret}(T)}{T} = \frac{1}{32} > 0$ .

# Explore-Then-Commit (ETC)

---

**Algorithm** Explore-Then-Commit

---

```
1: Input:  $m \in \{1, \dots, \lfloor \frac{T}{K} \rfloor\}$ .  
2: for  $t = 1 \rightarrow mK$  do  
3:   Select arm  $a_t = t \bmod K + 1$  and observe reward  $R_t$    (Explore)  
4: end for  
5: Calculate empirical mean reward for arm  $a \in [K]$  as  $\hat{\mu}_a(mK) := \frac{\sum_{t=1}^{mK} R_t \mathbb{I}\{a_t=a\}}{N_a(mK)}$   
6: for  $t = mK + 1 \rightarrow T$  do  
7:   Pull arm  $\hat{a} := \arg \max_{a \in [K]} \hat{\mu}_a(mK)$    (Commit)  
8: end for
```

---

**Q:** Will ETC result in sublinear regret?

Yes! under suitable distributional assumptions on the rewards.

In particular, if  $r \sim \nu_a$ , we will assume that  $r - \mu_a$  are sub-Gaussian random variables, then we will prove that ETC results in sub-linear regret. For this, we need to first recap some concentration (tail) inequalities from undergraduate probability.

## Digression – Concentration inequalities

**Concentration inequalities** bound the probability that the r.v. takes a value much different from its mean.

*Example:* Consider a r.v.  $X$  that can take on only non-negative values and  $\mathbb{E}[X] = 99.99$ . Show that  $\Pr[X \geq 300] \leq \frac{1}{3}$ .

$$\begin{aligned} \text{Proof: } \mathbb{E}[X] &= \sum_{x \in \text{Range}(X)} x \Pr[X = x] = \sum_{x|x \geq 300} x \Pr[X = x] + \sum_{x|0 \leq x < 300} x \Pr[X = x] \\ &\geq \sum_{x|x \geq 300} (300) \Pr[X = x] + \sum_{x|0 \leq x < 300} x \Pr[X = x] \\ &= (300) \Pr[X \geq 300] + \sum_{x|0 \leq x < 300} x \Pr[X = x] \end{aligned}$$

If  $\Pr[X \geq 300] > \frac{1}{3}$ , then,  $\mathbb{E}[X] > (300) \frac{1}{3} + \sum_{x|0 \leq x < 300} x \Pr[X = x] > 100$  (since the second term is always non-negative). Hence, if  $\Pr[X \geq 300] > \frac{1}{3}$ ,  $\mathbb{E}[X] > 100$  which is a contradiction since  $\mathbb{E}[X] = 99.99$ .

## Digression – Markov's Theorem

Markov's theorem formalizes the intuition on the previous slide, and can be stated as follows.

**Markov's Theorem:** If  $X$  is a non-negative random variable, then for all  $x > 0$ ,

$$\Pr[X \geq x] \leq \frac{\mathbb{E}[X]}{x}.$$

*Proof:* Define  $\mathcal{I}\{X \geq x\}$  to be the indicator r.v. for the event  $[X \geq x]$ . Then for all values of  $X$ ,  $x\mathcal{I}\{X \geq x\} \leq X$ .

$$\begin{aligned}\mathbb{E}[x\mathcal{I}\{X \geq x\}] &\leq \mathbb{E}[X] \implies x\mathbb{E}[\mathcal{I}\{X \geq x\}] \leq \mathbb{E}[X] \implies x\Pr[X \geq x] \leq \mathbb{E}[X] \\ &\implies \Pr[X \geq x] \leq \frac{\mathbb{E}[X]}{x}. \quad \square\end{aligned}$$

Since the above theorem holds for all  $x > 0$ , we can set  $x = c\mathbb{E}[X]$  for  $c \geq 1$ . In this case,  $\Pr[X \geq c\mathbb{E}[X]] \leq \frac{1}{c}$ . Hence, the probability that  $X$  is “far” from the mean in terms of the multiplicative factor  $c$  is upper-bounded by  $\frac{1}{c}$ .



## Digression – Sub-Gaussian random variables

If a centered r.v.  $X$  (meaning that  $\mathbb{E}[X] = 0$ ) is  $\sigma$  sub-Gaussian, then for all  $\lambda \in \mathbb{R}$ ,

$$\mathbb{E}[\exp(\lambda X)] \leq \exp\left(\frac{\lambda^2 \sigma^2}{2}\right).$$

*Example 1:* If  $X \sim N(0, 1)$ , then its moment generating function  $\mathbb{E}[\exp(\lambda X)] = \exp\left(\frac{\lambda^2 \sigma^2}{2}\right)$ , meaning that Gaussian r.v. are sub-Gaussian.

*Example 2:* If  $X \in [a, b]$  and  $\mathbb{E}[X] = 0$ , then  $X$  is  $(b - a)$  sub-Gaussian.

**Properties:** If  $X$  is centered and  $\sigma$  sub-Gaussian, then,

- (a)  $\mathbb{E}[X] = 0$ ,  $\text{Var}[X] \leq \sigma^2$
- (b) For a constant  $c \in \mathbb{R}$ ,  $cX$  is  $|c| \sigma$  sub-Gaussian.
- (c) If  $\{X_i\}_{i=1}^n$  are independent and  $\sigma_i$  sub-Gaussian respectively, then,  $\sum_{i=1}^n X_i$  is  $\sqrt{\sum_{i=1}^n \sigma_i^2}$  sub-Gaussian.

Need to prove some of these properties in Assignment 1!

## Digression – Concentration inequalities for sub-Gaussian r.v's

**Claim:** If  $X$  is centered and  $\sigma$  sub-Gaussian, then for any  $\epsilon \geq 0$ ,  $\Pr[X \geq \epsilon] \leq \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$ .

*Proof:* For some constant  $c > 0$  to be tuned later,

$$\begin{aligned}\Pr[X \geq \epsilon] &= \Pr[cX \geq c\epsilon] = \Pr[\exp(cX) \geq \exp(c\epsilon)] \\ &\leq \mathbb{E}[\exp(cX)] \exp(-c\epsilon) && \text{(Markov's inequality)} \\ &\leq \exp\left(\frac{c^2\sigma^2}{2} - c\epsilon\right) && \text{(Def. of sub-Gaussian r.v's)} \\ &= \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right) \quad \square && \text{(Setting } c = \epsilon/\sigma^2\text{)}\end{aligned}$$

Similarly,  $\Pr[X \leq -\epsilon] \leq \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$ . By the union bound,  $\Pr[|X| \geq \epsilon] \leq 2 \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$ .

Setting  $\delta = 2 \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right) \implies \epsilon = \sqrt{2\sigma^2 \log(2/\delta)}$ . Hence, w.p.  $1 - \delta$ ,  $X$  will take on values in the range  $\left[-\sqrt{2\sigma^2 \log(2/\delta)}, +\sqrt{2\sigma^2 \log(2/\delta)}\right]$ .

## Digression – Concentration inequalities for sub-Gaussian r.v's

**Claim:** Consider  $n$  i.i.d r.v's  $X_i$  such that  $\mathbb{E}[X_i] = \mu$ . If  $X_i - \mu$  are  $\sigma$  sub-Gaussian and  $\hat{\mu} := \frac{1}{n} \sum_{i=1}^n X_i$  is the empirical mean, then,  $\Pr[|\hat{\mu} - \mu| \geq \epsilon] \leq \exp\left(-\frac{n\epsilon^2}{2\sigma^2}\right)$ .

*Proof:* Using property (c) of  $\sigma$  sub-Gaussian r.v's,  $\sum_{i=1}^n [X_i - \mu]$  is  $\sqrt{n\sigma^2}$  sub-Gaussian. Using property (b) of  $\sigma$  sub-Gaussian r.v's,  $\frac{\sum_{i=1}^n [X_i - \mu]}{n}$  is  $\frac{\sigma}{\sqrt{n}}$  sub-Gaussian.

$\implies \hat{\mu} - \mu$  is  $\frac{\sigma}{\sqrt{n}}$  sub-Gaussian. Using the concentration result from the previous slide,  $\Pr[|\hat{\mu} - \mu| \geq \epsilon] \leq 2 \exp\left(-\frac{n\epsilon^2}{2\sigma^2}\right)$ . □

Hence, as we collect more data, the empirical mean concentrates around the true mean at an exponential rate.

# Back to Explore-Then-Commit (ETC)

---

**Algorithm** Explore-Then-Commit

---

- 1: **Input:**  $m \in \{1, \dots, \lfloor \frac{T}{K} \rfloor\}$ .
  - 2: **for**  $t = 1 \rightarrow mK$  **do**
  - 3:   Select arm  $a_t = (t \bmod K) + 1$  and observe reward  $R_t$    **(Explore)**
  - 4: **end for**
  - 5: Calculate empirical mean reward for arm  $a \in [K]$  as  $\hat{\mu}_a(mK) := \frac{\sum_{t=1}^{mK} R_t \mathbb{I}\{a_t=a\}}{N_a(mK)}$
  - 6: **for**  $t = mK + 1 \rightarrow T$  **do**
  - 7:   Pull arm  $\hat{a} := \arg \max_{a \in [K]} \hat{\mu}_a(mK)$    **(Commit)**
  - 8: **end for**
- 

**Distributional Assumption:** The *noise*  $\eta_t := R_t - \mu_{a_t}$  is 1 sub-Gaussian.  $\implies$  after pulling each arm  $m$  times in the **exploration** phase, for all  $a \in [K]$ ,  $|\hat{\mu}_a - \mu_a|$  is  $\frac{\sigma}{\sqrt{m}}$  sub-Gaussian and hence,  $\Pr[|\hat{\mu}_a - \mu_a| \geq \epsilon] \leq 2 \exp\left(-\frac{m\epsilon^2}{2\sigma^2}\right)$ .

Intuitively, the **exploration** phase estimates the gap  $\Delta_a$  for each arm upto a certain error. After this initial estimation, the algorithm **commits** to the *best empirical arm*.

## Explore-Then-Commit – Regret Analysis

**Claim:** For any  $m \in \{1, \dots, \lfloor T/K \rfloor\}$ ,

$$\text{Regret}(\text{ETC}, T) \leq m \sum_{a=1}^K \Delta_a + (T - mK) \sum_{a=1}^K \Delta_a \exp\left(-\frac{m \Delta_a^2}{4}\right)$$

*Proof:* Without loss of generality, assume that arm 1 is the best arm. Using the regret decomposition, we know that  $\text{Regret}(\text{ETC}, T) = \sum_a \Delta_a \mathbb{E}[N_a(T)]$ . For each arm  $a \neq 1$ ,  $\mathbb{E}[N_a(T)] = m + (T - mK) \Pr[\text{algorithm commits to arm } a]$ .

$$\Pr[\text{algorithm commits to arm } a] = \Pr[\hat{\mu}_a > \max_{j \neq a} \hat{\mu}_j] \leq \Pr[\hat{\mu}_a > \hat{\mu}_1]$$

(Since  $\{\hat{\mu}_a > \max_{j \neq a} \hat{\mu}_j\}$  is a subset of  $\{\hat{\mu}_a > \hat{\mu}_1\}$ )

$$= \Pr[\hat{\mu}_a - \mu_a > \hat{\mu}_1 - \mu_1 + [\mu_1 - \mu_a]] = \Pr[\underbrace{[\hat{\mu}_a - \mu_a]}_{X_a} - \underbrace{[\hat{\mu}_1 - \mu_1]}_{X_1} \geq \Delta_a]$$

$$= \Pr[X_a - X_1 \geq \Delta_a]$$

## Explore-Then-Commit – Regret Analysis

Recall that  $\text{Regret}(\text{ETC}, T) = \sum_a \Delta_a [m + (T - mK) \Pr[\text{algorithm commits to arm } a]]$  and  $\Pr[\text{algorithm commits to arm } a] \leq \Pr[X_a - X_1 \geq \Delta_a]$  where  $X_a = \hat{\mu}_a - \mu_a$ . Because of our assumption, both  $X_a$  and  $X_1$  are  $\frac{1}{\sqrt{m}}$  sub-Gaussian. Using property (c) of sub-Gaussian r.v's,  $X_a - X_1$  is  $\frac{\sqrt{2}}{\sqrt{m}}$  sub-Gaussian. Using the concentration result for sub-Gaussian r.v's,

$$\Pr[X_a - X_1 \geq \Delta_a] \leq \exp\left(-\frac{m \Delta_a^2}{4}\right)$$

Putting everything together,

$$\begin{aligned} \text{Regret}(\text{ETC}, T) &\leq \sum_a \Delta_a \left[ m + (T - mK) \exp\left(-\frac{m \Delta_a^2}{4}\right) \right] \\ \Rightarrow \text{Regret}(\text{ETC}, T) &\leq m \sum_{a=1}^K \Delta_a + (T - mK) \sum_{a=1}^K \Delta_a \exp\left(-\frac{m \Delta_a^2}{4}\right) \quad \square \end{aligned}$$

## Explore-Then-Commit – Regret Analysis

Recall that  $\text{Regret}(\text{ETC}, T) \leq m \sum_{a=1}^K \Delta_a + (T - mK) \sum_{a=1}^K \Delta_a \exp\left(-\frac{m\Delta_a^2}{4}\right)$ .

In order to gain some intuition about how to set  $m$ , consider  $K = 2$  with  $\Delta := \mu_1 - \mu_2$ .

$$\text{Regret}(\text{ETC}, T) \leq m\Delta + (T - 2m)\Delta \exp\left(-\frac{m\Delta^2}{4}\right) < m\Delta + T\Delta \exp\left(-\frac{m\Delta^2}{4}\right)$$

Optimizing the RHS w.r.t  $m$ , we get  $m = \frac{4}{\Delta^2} \log\left(\frac{\Delta^2 T}{4}\right)$ . Since  $m$  is an integer  $\geq 1$ , we should set  $m = \max\left\{1, \lceil \frac{4}{\Delta^2} \log\left(\frac{\Delta^2 T}{4}\right) \rceil\right\}$ . Plugging this value back,

$$\implies \text{Regret}(\text{ETC}, T) \leq \Delta + \frac{4}{\Delta} \left[1 + \log_+\left(\frac{\Delta^2 T}{4}\right)\right] \quad (\log_+(x) := \max\{0, \log(x)\})$$

Hence, ETC with  $m = O(1/\Delta^2)$  achieves  $O\left(\frac{\log(T)}{\Delta}\right)$  *instance or gap-dependent* regret.

**Q:** What is the problem with this bound? **Ans:** As  $\Delta \rightarrow 0$ ,  $\text{Regret}(\text{ETC}, T) \rightarrow \infty$  implying that for some instances, the bound on the gap-dependent regret can be very large.

## Explore-Then-Commit – Regret Analysis

To overcome the previous problem, one can bound the *worst-case problem-independent regret*.

**Claim:** For  $\Delta \leq 1$ , ETC results in an  $O(1 + \sqrt{T})$  worst-case bound on the regret.

*Proof:* In the worst-case, we pull the sub-optimal arm in every round. Hence, the regret for any algorithm is upper-bounded by  $T\Delta$ . Putting this together with the bound on the previous slide,

$$\text{Regret}(\text{ETC}, T) \leq \min \left\{ T\Delta, \Delta + \frac{4}{\Delta} \left[ 1 + \log_+ \left( \frac{\Delta^2 T}{4} \right) \right] \right\}$$

If  $\Delta < \frac{1}{\sqrt{T}}$ ,  $\text{Regret}(\text{ETC}, T) \leq \sqrt{T}$ . On the other hand, if  $\Delta \geq \frac{1}{\sqrt{T}}$ ,

$$\text{Regret}(\text{ETC}, T) \leq \Delta + 4\sqrt{T} + \left[ \frac{4}{\Delta} \log_+ \left( \frac{\Delta^2 T}{4} \right) \right] \leq \Delta + 4\sqrt{T} + 4 \max_{z>0} \frac{\log_+(Tz^2/4)}{z}$$

$$\text{Regret}(\text{ETC}, T) \leq \Delta + 4\sqrt{T} + \frac{4\sqrt{T}}{e} \leq 1 + \sqrt{T} \left( 4 + \frac{4}{e} \right) \quad (\text{Since } \Delta \leq 1)$$

- In general, for  $K$  arms, it can be shown that ETC results in  $O(\sqrt{KT})$  worst-case regret.



## Explore-Then-Commit – Regret Analysis

We have seen that ETC with  $m = O(1/\Delta^2)$  achieves an  $O(\Delta + \sqrt{T})$  regret for any instance.

**Q:** What is the problem with the ETC algorithm? **Ans:** Requires knowledge of  $\Delta$  to set  $m$ !

**Claim:** For  $\Delta \leq 1$ , there exists  $C > 0$  s.t. ETC with  $m = T^{2/3}$  results in  $(1 + C) T^{2/3}$  regret.

*Proof:* Need to prove this in Assignment 1!

*Hint:* Starting from the expression,  $\text{Regret}(\text{ETC}, T) \leq m\Delta + T\Delta \exp\left(-\frac{m\Delta^2}{4}\right)$ , upper-bound the second term independent of  $\Delta$  and then choose  $m$ .

