CS 747, Autumn 2022: Lecture 2

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

Autumn 2022

Multi-armed Bandits

1. Evaluating algorithms: Regret
2. Achieving sub-linear regret
3. A lower bound on regret

Multi-armed Bandits

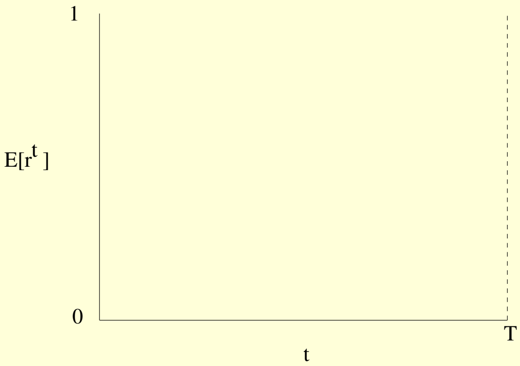
1. Evaluating algorithms: Regret

2. Achieving sub-linear regret

3. A lower bound on regret

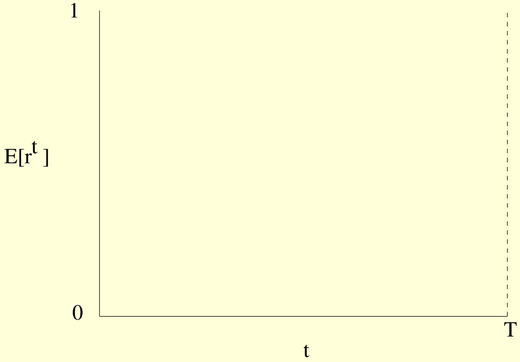
Visualising Performance

- Consider a plot of $\mathbb{E}[r^t]$ against t.



Visualising Performance

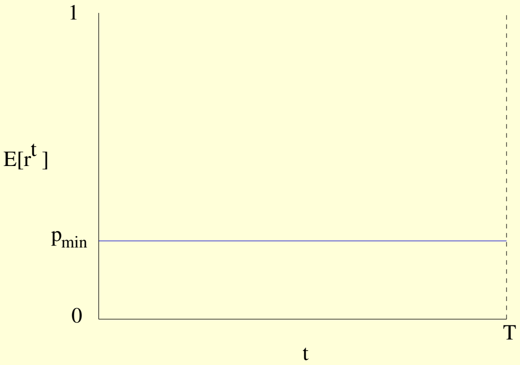
- Consider a plot of $\mathbb{E}[r^t]$ against t.
- What is the least expected reward that can be achieved?



Visualising Performance

- Consider a plot of $\mathbb{E}[r^t]$ against t.
- What is the least expected reward that can be achieved?

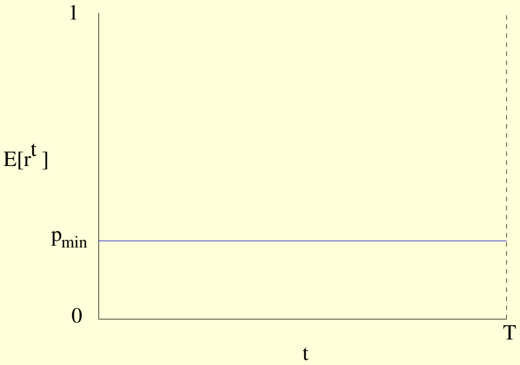
  $p_{\min} = \min_{a \in A} p_a$.



Visualising Performance

- Consider a plot of $\mathbb{E}[r^t]$ against t.
- What is the least expected reward that can be achieved?

  $p_{\min} = \min_{a \in A} p_a$.
- What is the highest expected reward that can be achieved?

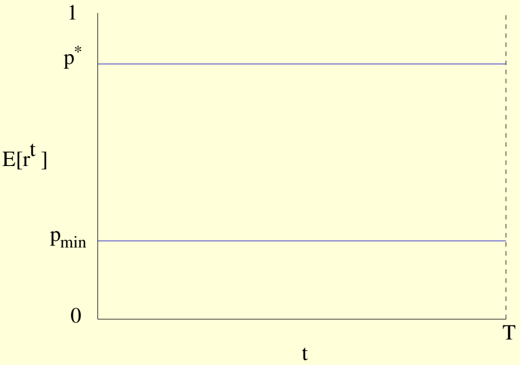


Visualising Performance

- Consider a plot of $\mathbb{E}[r^t]$ against t.
- What is the least expected reward that can be achieved?

  $p_{\min} = \min_{a \in A} p_a$.
- What is the highest expected reward that can be achieved?

  $p^\star = \max_{a \in A} p_a$.

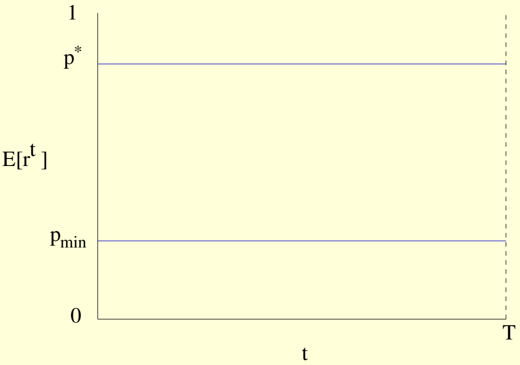


Visualising Performance

- Consider a plot of $\mathbb{E}[r^t]$ against t.
- What is the least expected reward that can be achieved?

  $p_{\min} = \min_{a \in A} p_a$.
- What is the highest expected reward that can be achieved?

  $p^\star = \max_{a \in A} p_a$.



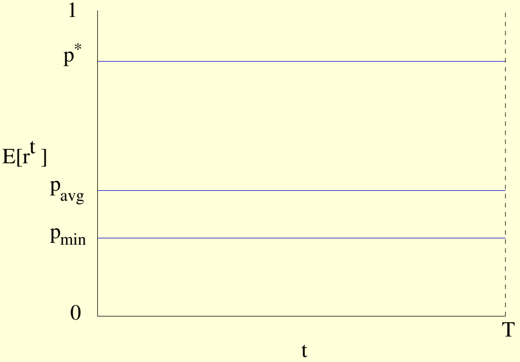
- If an algorithm pulls arms uniformly at random, what reward will it achieve?

Visualising Performance

- Consider a plot of $\mathbb{E}[r^t]$ against t.
- What is the least expected reward that can be achieved?

  $p_{\min} = \min_{a \in A} p_a$.
- What is the highest expected reward that can be achieved?

  $p^\star = \max_{a \in A} p_a$.



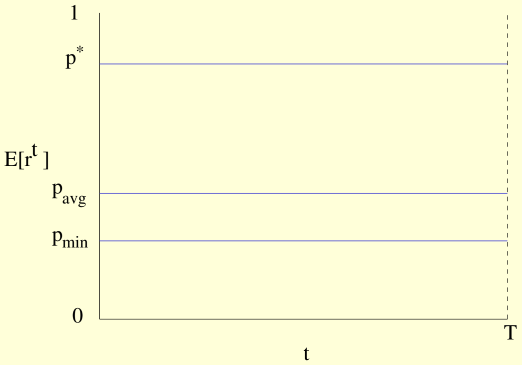
- If an algorithm pulls arms uniformly at random, what reward will it achieve?
  $p_{\text{avg}} = \frac{1}{n} \sum_{a \in A} p_a$.

Visualising Performance

- Consider a plot of $\mathbb{E}[r^t]$ against t.
- What is the least expected reward that can be achieved?

  $p_{\min} = \min_{a \in A} p_a$.
- What is the highest expected reward that can be achieved?

  $p^\star = \max_{a \in A} p_a$.



- If an algorithm pulls arms uniformly at random, what reward will it achieve?

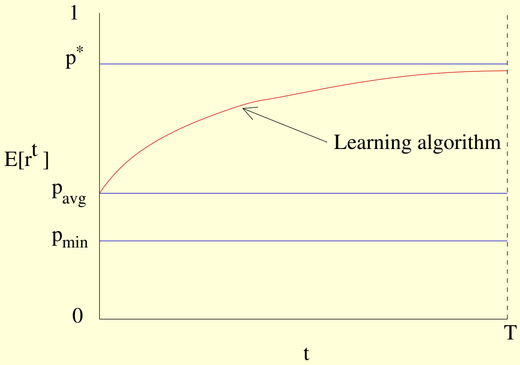
  $p_{\text{avg}} = \frac{1}{n} \sum_{a \in A} p_a$.
- How will the graph look for a reasonable learning algorithm?

Visualising Performance

- Consider a plot of $\mathbb{E}[r^t]$ against t.
- What is the least expected reward that can be achieved?

  $p_{\min} = \min_{a \in A} p_a$.
- What is the highest expected reward that can be achieved?

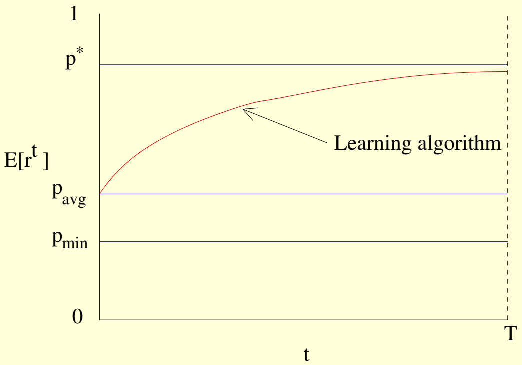
  $p^\star = \max_{a \in A} p_a$.



- If an algorithm pulls arms uniformly at random, what reward will it achieve? $p_{\text{avg}} = \frac{1}{n} \sum_{a \in A} p_a$.
- How will the graph look for a reasonable learning algorithm?

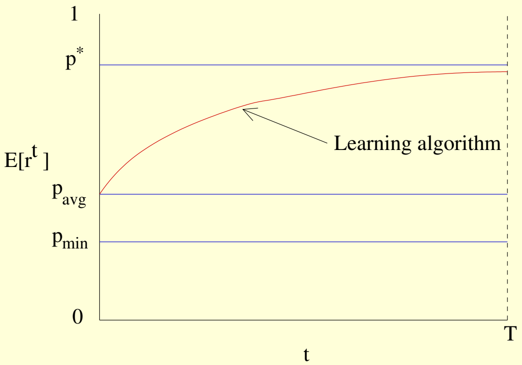
Regret

- The maximum achievable expected reward in T steps is $Tp^{\star}$.



Regret

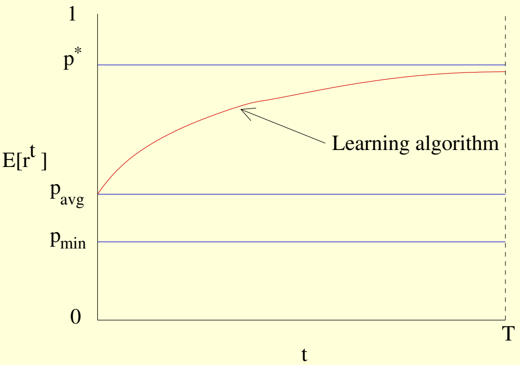
- The maximum achievable expected reward in T steps is $Tp^{\star}$.
- The actual expected reward for an algorithm is $\sum_{t=0}^{T-1} \mathbb{E}[r^t]$.



Regret

- The maximum achievable expected reward in T steps is $Tp^{\star}$.
- The actual expected reward for an algorithm is $\sum_{t=0}^{T-1} \mathbb{E}[r^t]$.
- The (expected cumulative) regret of the algorithm for horizon T is the difference

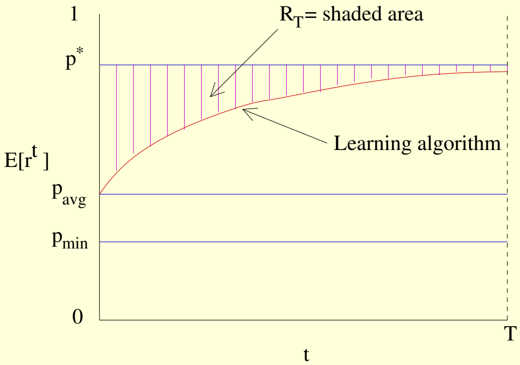
$$R_T = Tp^* - \sum_{t=0}^{T-1} \mathbb{E}[r^t].$$



Regret

- The maximum achievable expected reward in T steps is $Tp^\star$.
- The actual expected reward for an algorithm is $\sum_{t=0}^{T-1} \mathbb{E}[r^t]$.
- The (expected cumulative) regret of the algorithm for horizon T is the difference

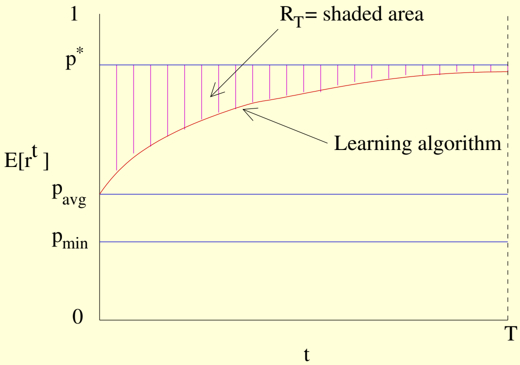
$$R_T = Tp^* - \sum_{t=0}^{T-1} \mathbb{E}[r^t].$$



Regret

- The maximum achievable expected reward in T steps is $Tp^\star$.
- The actual expected reward for an algorithm is $\sum_{t=0}^{T-1} \mathbb{E}[r^t]$.
- The (expected cumulative) regret of the algorithm for horizon T is the difference

$$R_T = Tp^* - \sum_{t=0}^{T-1} \mathbb{E}[r^t].$$

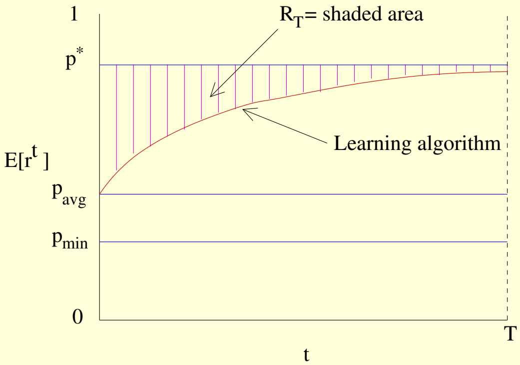


- We would like R_T to be small, in fact for $\lim_{T\to\infty} \frac{R_T}{T} = 0$.

Regret

- The maximum achievable expected reward in T steps is $Tp^\star$.
- The actual expected reward for an algorithm is $\sum_{t=0}^{T-1} \mathbb{E}[r^t]$.
- The (expected cumulative) regret of the algorithm for horizon T is the difference

$$R_T = Tp^* - \sum_{t=0}^{T-1} \mathbb{E}[r^t].$$



- We would like R_T to be small, in fact for $\lim_{T\to\infty} \frac{R_T}{T} = 0$. Does this happen for ϵG1, ϵG2, ϵG3?

Multi-armed Bandits

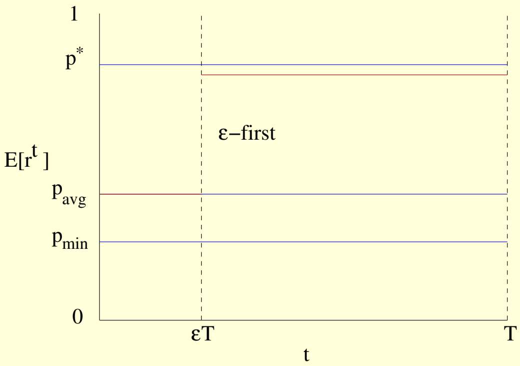
1. Evaluating algorithms: Regret

2. Achieving sub-linear regret

3. A lower bound on regret

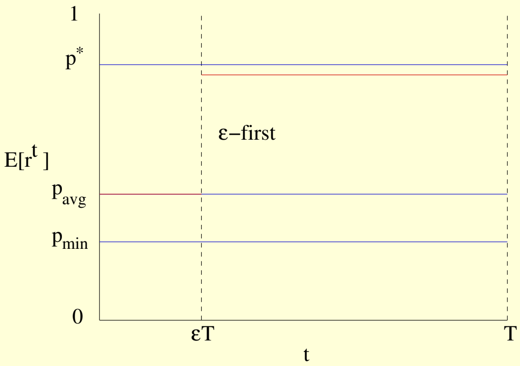
Review of ϵG1, ϵG2

- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.



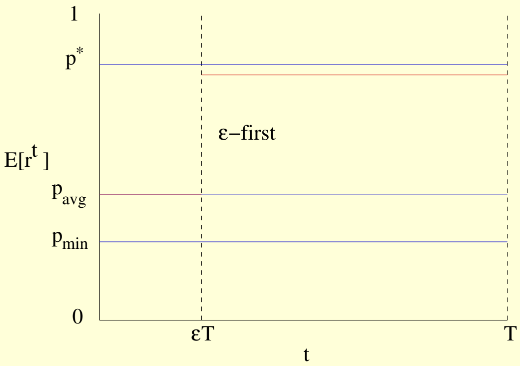
Review of ϵG1, ϵG2

- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.
- What would happen if we ran for horizon $2T$ instead of T?



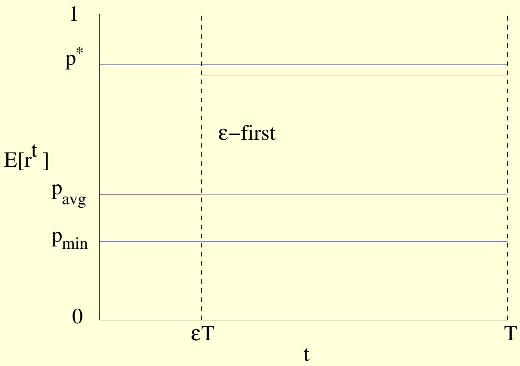
Review of ϵG1, ϵG2

- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.
- What would happen if we ran for horizon $2T$ instead of T? Exploratory phase would last $2\epsilon T$ steps!



Review of ϵG1, ϵG2

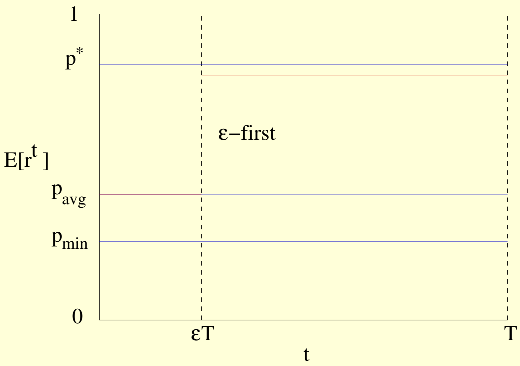
- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.
- What would happen if we ran for horizon $2T$ instead of T? Exploratory phase would last $2\epsilon T$ steps!



$$R_T = Tp^\star - \sum_{t=0}^{T-1} \mathbb{E}[r^t]$$

Review of ϵG1, ϵG2

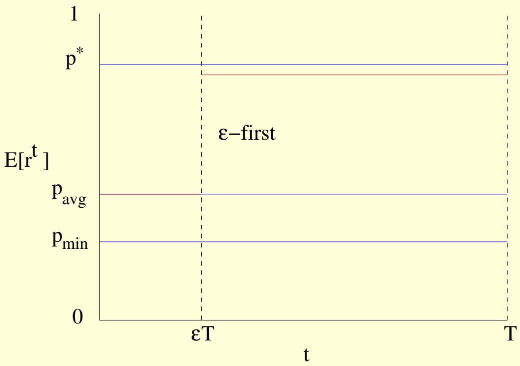
- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.
- What would happen if we ran for horizon $2T$ instead of T? Exploratory phase would last $2\epsilon T$ steps!



$$R_T = Tp^\star - \sum_{t=0}^{T-1} \mathbb{E}[r^t] = Tp^\star - \sum_{t=0}^{\epsilon T-1} \mathbb{E}[r^t] - \sum_{t=\epsilon T}^{T-1} \mathbb{E}[r^t]$$

Review of ϵG1, ϵG2

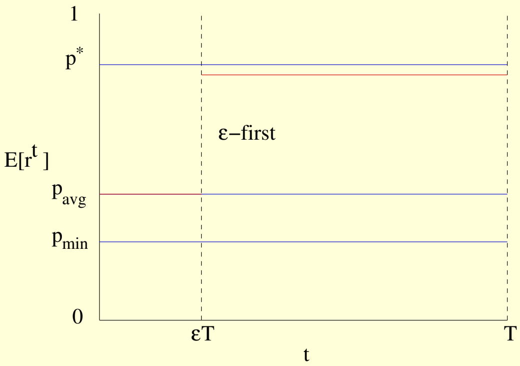
- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.
- What would happen if we ran for horizon $2T$ instead of T? Exploratory phase would last $2\epsilon T$ steps!



$$R_T = Tp^\star - \sum_{t=0}^{T-1} \mathbb{E}[r^t] = Tp^\star - \sum_{t=0}^{\epsilon T-1} \mathbb{E}[r^t] - \sum_{t=\epsilon T}^{T-1} \mathbb{E}[r^t] = Tp^\star - \epsilon T p_{\text{avg}} - \sum_{t=\epsilon T}^{T-1} \mathbb{E}[r^t]$$

Review of ϵG1, ϵG2

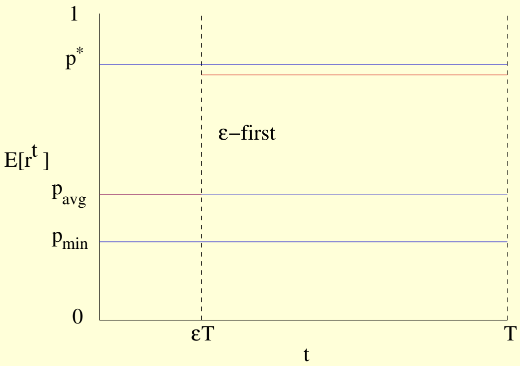
- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.
- What would happen if we ran for horizon $2T$ instead of T? Exploratory phase would last $2\epsilon T$ steps!



$$R_T = Tp^\star - \sum_{t=0}^{T-1} \mathbb{E}[r^t] = Tp^\star - \sum_{t=0}^{\epsilon T-1} \mathbb{E}[r^t] - \sum_{t=\epsilon T}^{T-1} \mathbb{E}[r^t] = Tp^\star - \epsilon T p_{\text{avg}} - \sum_{t=\epsilon T}^{T-1} \mathbb{E}[r^t]$$
$$\geq Tp^\star - \epsilon T p_{\text{avg}} - (T - \epsilon T)p^\star$$

Review of ϵG1, ϵG2

- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.
- What would happen if we ran for horizon $2T$ instead of T? Exploratory phase would last $2\epsilon T$ steps!

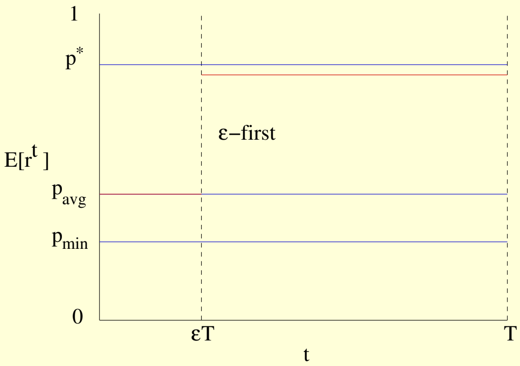


$$R_T = Tp^\star - \sum_{t=0}^{T-1} \mathbb{E}[r^t] = Tp^\star - \sum_{t=0}^{\epsilon T-1} \mathbb{E}[r^t] - \sum_{t=\epsilon T}^{T-1} \mathbb{E}[r^t] = Tp^\star - \epsilon T p_{\text{avg}} - \sum_{t=\epsilon T}^{T-1} \mathbb{E}[r^t]$$

$$\geq Tp^\star - \epsilon T p_{\text{avg}} - (T - \epsilon T)p^\star = \epsilon(p^\star - p_{\text{avg}})T$$

Review of ϵG1, ϵG2

- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.
- What would happen if we ran for horizon $2T$ instead of T? Exploratory phase would last $2\epsilon T$ steps!



$$R_T = Tp^\star - \sum_{t=0}^{T-1} \mathbb{E}[r^t] = Tp^\star - \sum_{t=0}^{\epsilon T-1} \mathbb{E}[r^t] - \sum_{t=\epsilon T}^{T-1} \mathbb{E}[r^t] = Tp^\star - \epsilon T p_{\text{avg}} - \sum_{t=\epsilon T}^{T-1} \mathbb{E}[r^t]$$

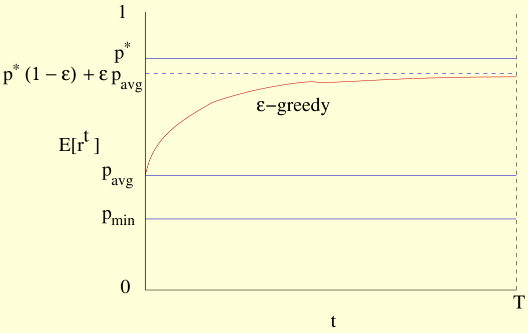
$$\geq Tp^\star - \epsilon T p_{\text{avg}} - (T - \epsilon T)p^\star = \epsilon(p^\star - p_{\text{avg}})T = \Omega(T).$$

Review of ϵG3

- ϵ-greedy: On each step explore (uniformly) w.p. ϵ, exploit w.p. $1 - \epsilon$.

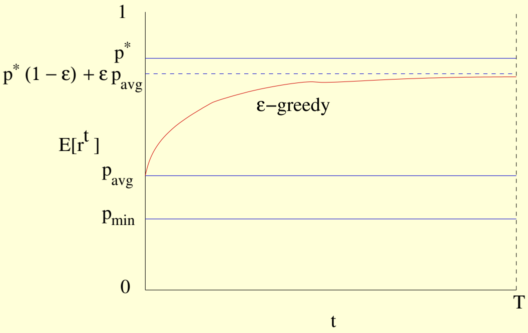
Review of ϵG3

- ϵ-greedy: On each step explore (uniformly) w.p. ϵ, exploit w.p. $1 - \epsilon$.



Review of ϵG3

- ϵ-greedy: On each step explore (uniformly) w.p. ϵ, exploit w.p. $1 - \epsilon$.
- $\mathbb{E}[r^t]$ can never exceed $p^\star(1 - \epsilon) + \epsilon p_{avg}$!

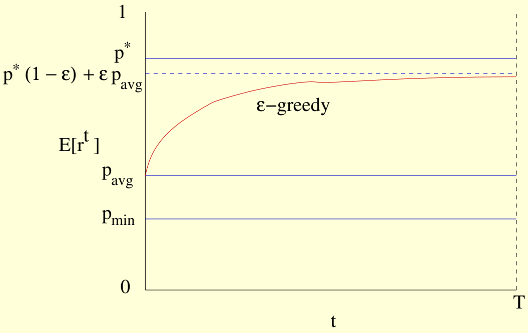


Review of ϵG3

- ϵ-greedy: On each step explore (uniformly) w.p. ϵ, exploit w.p. $1 - \epsilon$.

- $\mathbb{E}[r^t]$ can never exceed $p^\star(1 - \epsilon) + \epsilon p_{\text{avg}}$!

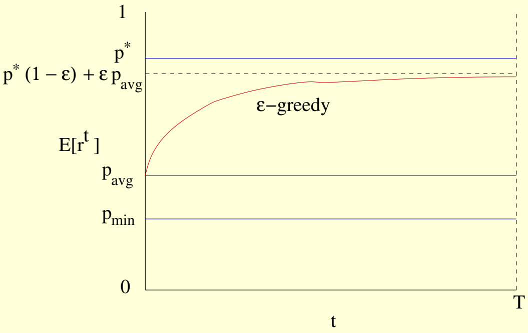
$$R_T = Tp^\star - \sum_{t=0}^{T-1} \mathbb{E}[r^t]$$



Review of ϵG3

- ϵ-greedy: On each step explore (uniformly) w.p. ϵ, exploit w.p. $1 - \epsilon$.

- $\mathbb{E}[r^t]$ can never exceed $p^\star(1 - \epsilon) + \epsilon p_{\text{avg}}$!

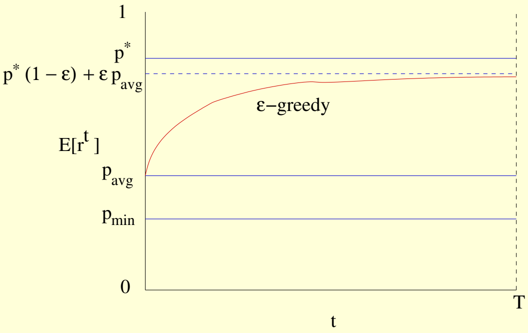


$$
\begin{aligned}
R_T &= Tp^\star - \sum_{t=0}^{T-1} \mathbb{E}[r^t] \\
&\geq Tp^\star - \sum_{t=0}^{T-1} ((\epsilon)p_{\text{avg}} + (1 - \epsilon)p^\star)
\end{aligned}
$$

Review of ϵG3

- ϵ-greedy: On each step explore (uniformly) w.p. ϵ, exploit w.p. $1 - \epsilon$.

- $\mathbb{E}[r^t]$ can never exceed $p^\star(1 - \epsilon) + \epsilon p_{\text{avg}}$!

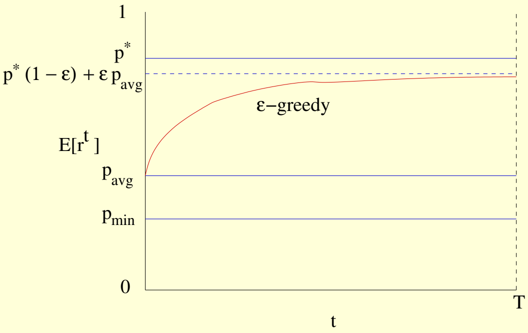


$$R_T = Tp^\star - \sum_{t=0}^{T-1} \mathbb{E}[r^t]$$

$$\geq Tp^\star - \sum_{t=0}^{T-1} ((\epsilon)p_{\text{avg}} + (1 - \epsilon)p^\star) = \epsilon(p^\star - p_{\text{avg}})T$$

Review of ϵG3

- ϵ-greedy: On each step explore (uniformly) w.p. ϵ, exploit w.p. $1 - \epsilon$.
- $\mathbb{E}[r^t]$ can never exceed $p^\star(1 - \epsilon) + \epsilon p_{\text{avg}}$!



$$R_T = Tp^\star - \sum_{t=0}^{T-1} \mathbb{E}[r^t]$$

$$\geq Tp^\star - \sum_{t=0}^{T-1} ((\epsilon)p_{\text{avg}} + (1 - \epsilon)p^\star) = \epsilon(p^\star - p_{\text{avg}})T = \Omega(T).$$

How to achieve Sub-linear Regret?

- Two conditions must be met: C1 and C2.

How to achieve Sub-linear Regret?

- Two conditions must be met: C1 and C2.

C1. **Infinite exploration.** In the limit ($T \to \infty$), each arm must almost surely be pulled an infinite number of times.

How to achieve Sub-linear Regret?

- Two conditions must be met: C1 and C2.

C1. **Infinite exploration.** In the limit ($T \to \infty$), each arm must almost surely be pulled an infinite number of times.

- On the contrary, suppose we pull some arm a only a finite U times.
- We cannot be 100% sure based on the pulls of a that it is non-optimal.
- Even an optimal arm a will have the lowest possible empirical mean (0) with positive probability $(1 - p^\star)^U$.
- Pulling only arms other than a will give linear regret if no other optimal arms.

How to achieve Sub-linear Regret?

C2. **Greed in the Limit**. Let *exploit*(T) denote the number of pulls that are greedy w.r.t. the empirical mean up to horizon T. For sub-linear regret, we need

$$\lim_{T \to \infty} \frac{\mathbb{E}[exploit(T)]}{T} = 1.$$

How to achieve Sub-linear Regret?

C2. **Greed in the Limit**. Let *exploit*(T) denote the number of pulls that are greedy w.r.t. the empirical mean up to horizon T. For sub-linear regret, we need

$$\lim_{T \to \infty} \frac{\mathbb{E}[exploit(T)]}{T} = 1.$$

- Let $\bar{\mathcal{I}}$ be the set of all bandit instances with reward means strictly less than 1.
- **Result.** An algorithm L achieves sub-linear regret on all instances $I \in \bar{\mathcal{I}}$ if and only if it satisfies C1 and C2 on all $I \in \bar{\mathcal{I}}$.

How to achieve Sub-linear Regret?

C2. **Greed in the Limit**. Let *exploit*(T) denote the number of pulls that are greedy w.r.t. the empirical mean up to horizon T. For sub-linear regret, we need

$$\lim_{T \to \infty} \frac{\mathbb{E}[exploit(T)]}{T} = 1.$$

- Let $\bar{\mathcal{I}}$ be the set of all bandit instances with reward means strictly less than 1.
- **Result.** An algorithm L achieves sub-linear regret on all instances $I \in \bar{\mathcal{I}}$ if and only if it satisfies C1 and C2 on all $I \in \bar{\mathcal{I}}$.

In short: "GLIE" $\iff$ sub-linear regret.

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.

  Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls. Thereafter exploit.

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.
  Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls. Thereafter exploit.

  C1 satisfied since each arm gets at least $\Theta(\frac{1}{n}\sqrt{T})$ pulls with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq T - \sqrt{T}$.

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.
  Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls. Thereafter exploit.

  C1 satisfied since each arm gets at least $\Theta(\frac{1}{n}\sqrt{T})$ pulls with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq T - \sqrt{T}$.

- ϵ_t-greedy with $\epsilon_t = \frac{1}{t+1}$.
  On the t-th step, explore w.p. ϵ_t, exploit w.p. $1 - \epsilon_t$.

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.
  Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls. Thereafter exploit.

  C1 satisfied since each arm gets at least $\Theta(\frac{1}{n}\sqrt{T})$ pulls with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq T - \sqrt{T}$.

- ϵ_t-greedy with $\epsilon_t = \frac{1}{t+1}$.
  On the t-th step, explore w.p. ϵ_t, exploit w.p. $1 - \epsilon_t$.

  C1 satisfied: each arm assured $\sum_{t=0}^{T-1} \frac{1}{n(t+1)} = \Theta(\frac{\log T}{n})$ pulls with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq T - \Theta(\log T)$.

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.
  Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls. Thereafter exploit.

  C1 satisfied since each arm gets at least $\Theta(\frac{1}{n}\sqrt{T})$ pulls with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq T - \sqrt{T}$.

- ϵ_t-greedy with $\epsilon_t = \frac{1}{t+1}$.
  On the t-th step, explore w.p. ϵ_t, exploit w.p. $1 - \epsilon_t$.

  C1 satisfied: each arm assured $\sum_{t=0}^{T-1} \frac{1}{n(t+1)} = \Theta(\frac{\log T}{n})$ pulls with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq T - \Theta(\log T)$.
  What happened when we took $\epsilon_t = \epsilon$?

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.
  Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls. Thereafter exploit.

  C1 satisfied since each arm gets at least $\Theta(\frac{1}{n}\sqrt{T})$ pulls with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq T - \sqrt{T}$.

- ϵ_t-greedy with $\epsilon_t = \frac{1}{t+1}$.
  On the t-th step, explore w.p. ϵ_t, exploit w.p. $1 - \epsilon_t$.

  C1 satisfied: each arm assured $\sum_{t=0}^{T-1} \frac{1}{n(t+1)} = \Theta(\frac{\log T}{n})$ pulls with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq T - \Theta(\log T)$.
  What happened when we took $\epsilon_t = \epsilon$? What will happen by taking $\epsilon_t = \frac{1}{(t+1)^2}$?

Multi-armed Bandits

1. Evaluating algorithms: Regret
2. Achieving sub-linear regret
3. A lower bound on regret

A Lower Bound on Regret

- What is the least regret possible?

A Lower Bound on Regret

- What is the least regret possible?
- An algorithm that always pulls arm 3 gets zero regret on some instances. . .

A Lower Bound on Regret

- What is the least regret possible?
- An algorithm that always pulls arm 3 gets zero regret on some instances. . . but linear regret on other instances!

A Lower Bound on Regret

- What is the least regret possible?
- An algorithm that always pulls arm 3 gets zero regret on some instances. . . but linear regret on other instances!
- We desire “low” regret on all instances. What is the best we can do?

A Lower Bound on Regret

Paraphrasing Lai and Robbins (1985; see Theorem 2).

Let L be an algorithm such that for every bandit instance $I \in \bar{\mathcal{I}}$ and for every $\alpha > 0$, as $T \to \infty$:

$$R_T(L, I) = o(T^\alpha).$$

A Lower Bound on Regret

Paraphrasing Lai and Robbins (1985; see Theorem 2).

Let L be an algorithm such that for every bandit instance $I \in \bar{\mathcal{I}}$ and for every $\alpha > 0$, as $T \to \infty$:

$$R_T(L, I) = o(T^{\alpha}).$$

Then, for every bandit instance $I \in \bar{\mathcal{I}}$, as $T \to \infty$:

$$\frac{R_T(L, I)}{\ln(T)} \geq \sum_{a: p_a(I) \neq p^\star(I)} \frac{p^\star(I) - p_a(I)}{KL(p_a(I), p^\star(I))},$$

where for $x, y \in [0, 1)$, $KL(x, y) \stackrel{\text{def}}{=} x \ln \frac{x}{y} + (1 - x) \ln \frac{1-x}{1-y}$.

Multi-armed Bandits

1. Evaluating algorithms: Regret
2. Achieving sub-linear regret
3. A lower bound on regret

Multi-armed Bandits

1. Evaluating algorithms: Regret

2. Achieving sub-linear regret

3. A lower bound on regret

Next class: Optimal algorithms!