CS 747, Autumn 2020: Week 1, Lecture 1

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

Autumn 2020

Multi-armed Bandits

1. The exploration-exploitation dilemma
2. Definitions: Bandit, Algorithm
3. ϵ-greedy algorithms
4. Evaluating algorithms: Regret

Multi-armed Bandits

1. The exploration-exploitation dilemma
2. Definitions: Bandit, Algorithm
3. ϵ-greedy algorithms
4. Evaluating algorithms: Regret

A Game

Coin 1

$\mathbb{P}\{\text{heads}\} = p_1$

Coin 2

$\mathbb{P}\{\text{heads}\} = p_2$

Coin 3

$\mathbb{P}\{\text{heads}\} = p_3$

- p_1, p_2, and p_3 are unknown.
- You are given a total of 20 tosses.
- Maximise the total number of heads!

A Game

Coin 1

$\mathbb{P}\{\text{heads}\} = p_1$

Coin 2

$\mathbb{P}\{\text{heads}\} = p_2$

Coin 3

$\mathbb{P}\{\text{heads}\} = p_3$

- p_1, p_2, and p_3 are unknown.
- You are given a total of 20 tosses.
- Maximise the total number of heads!

Let's play!

A Game

Coin 1

$\mathbb{P}\{\text{heads}\} = p_1$

Coin 2

$\mathbb{P}\{\text{heads}\} = p_2$

Coin 3

$\mathbb{P}\{\text{heads}\} = p_3$

- p_1, p_2, and p_3 are unknown.
- You are given a total of 20 tosses.
- Maximise the total number of heads!

Let's play!

- Now we know: $p_1 = 0.6, p_2 = 0.3, p_3 = 0.8$.

A Game

Coin 1

$\mathbb{P}\{\text{heads}\} = p_1$

Coin 2

$\mathbb{P}\{\text{heads}\} = p_2$

Coin 3

$\mathbb{P}\{\text{heads}\} = p_3$

- p_1, p_2, and p_3 are unknown.
- You are given a total of 20 tosses.
- Maximise the total number of heads!

Let's play!

- Now we know: $p_1 = 0.6, p_2 = 0.3, p_3 = 0.8$.
- If you knew these biases beforehand, how would you have played?

A Game

Coin 1

$\mathbb{P}\{\text{heads}\} = p_1$

Coin 2

$\mathbb{P}\{\text{heads}\} = p_2$

Coin 3

$\mathbb{P}\{\text{heads}\} = p_3$

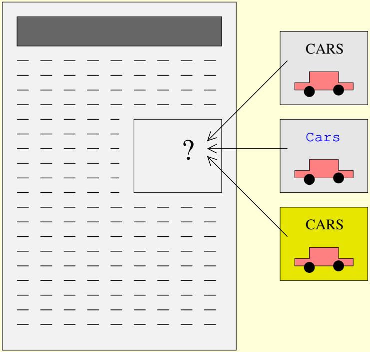
- p_1, p_2, and p_3 are unknown.
- You are given a total of 20 tosses.
- Maximise the total number of heads!

Let's play!

- Now we know: $p_1 = 0.6, p_2 = 0.3, p_3 = 0.8$.
- If you knew these biases beforehand, how would you have played?
- By so doing, how many heads would you have got in 20 tosses?

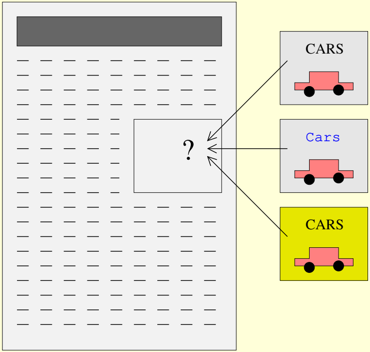
To Explore or to Exploit?

- On-line advertising: Template optimisation



To Explore or to Exploit?

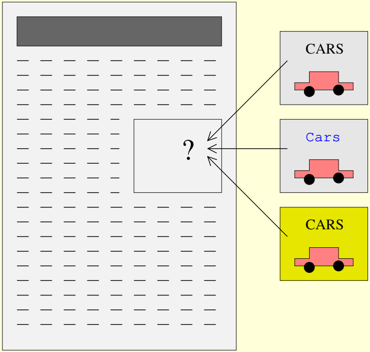
- On-line advertising: Template optimisation



- Clinical trials

To Explore or to Exploit?

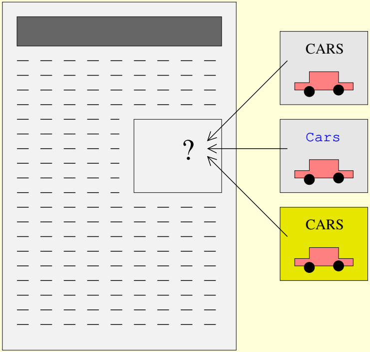
- On-line advertising: Template optimisation



- Clinical trials
- Packet routing in communication networks

To Explore or to Exploit?

- On-line advertising: Template optimisation

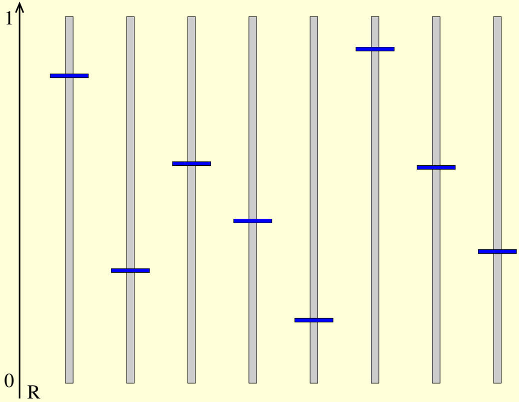


- Clinical trials
- Packet routing in communication networks
- Game playing and reinforcement learning

Multi-armed Bandits

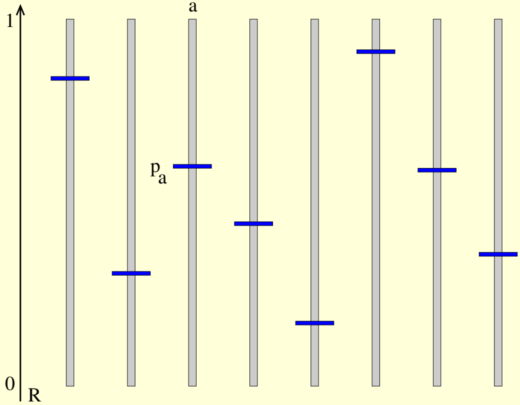
1. The exploration-exploitation dilemma
2. Definitions: Bandit, Algorithm
3. ϵ-greedy algorithms
4. Evaluating algorithms: Regret

Stochastic Multi-armed Bandits



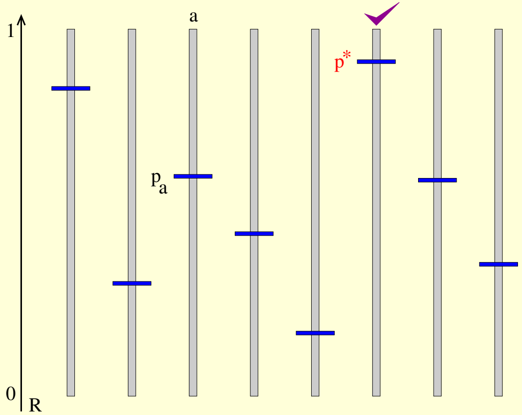
- *n* arms, each associated with a Bernoulli distribution (rewards are 0 or 1).

Stochastic Multi-armed Bandits



- n arms, each associated with a Bernoulli distribution (rewards are 0 or 1).
- Let A be the set of arms. Arm $a \in A$ has mean reward p_a.

Stochastic Multi-armed Bandits



- *n* arms, each associated with a Bernoulli distribution (rewards are 0 or 1).
- Let A be the set of arms. Arm $a \in A$ has mean reward p_a.
- Highest mean is $p^\star$.

One-armed Bandits

[1]

1. https://pxhere.com/en/photo/942387.

Algorithm

- Here is what an algorithm does—

<u>For $t = 0, 1, 2, \ldots, T - 1$:</u>

- Given the history $h^t = (a^0, r^0, a^1, r^1, a^2, r^2, \ldots, a^{t-1}, r^{t-1})$,
- Pick an arm a^t to sample (or “pull”), and
- Obtain a reward r^t drawn from the distribution corresponding to arm a^t.

Algorithm

- Here is what an algorithm does—

For $t = 0, 1, 2, \ldots, T-1$:

- Given the history $h^t = (a^0, r^0, a^1, r^1, a^2, r^2, \ldots, a^{t-1}, r^{t-1})$,
- Pick an arm a^t to sample (or "pull"), and
- Obtain a reward r^t drawn from the distribution corresponding to arm a^t.

- T is the total sampling budget, or the horizon.

Algorithm

- Here is what an algorithm does—

For $t = 0, 1, 2, \ldots, T-1$:

- Given the history $h^t = (a^0, r^0, a^1, r^1, a^2, r^2, \ldots, a^{t-1}, r^{t-1})$,
- Pick an arm a^t to sample (or "pull"), and
- Obtain a reward r^t drawn from the distribution corresponding to arm a^t.

- T is the total sampling budget, or the horizon.

- Formally: a deterministic algorithm is a mapping

  from the set of all histories

  to the set of all arms.

Algorithm

- Here is what an algorithm does—

For $t = 0, 1, 2, \ldots, T - 1$:

- Given the history $h^t = (a^0, r^0, a^1, r^1, a^2, r^2, \ldots, a^{t-1}, r^{t-1})$,
- Pick an arm a^t to sample (or "pull"), and
- Obtain a reward r^t drawn from the distribution corresponding to arm a^t.

- T is the total sampling budget, or the horizon.
- Formally: a deterministic algorithm is a mapping

  from the set of all histories

  to the set of all arms.
- Formally: a randomised algorithm is a mapping

  from the set of all histories

  to the set of all probability distributions over arms.

Algorithm

- Here is what an algorithm does—

For $t = 0, 1, 2, \ldots, T-1$:

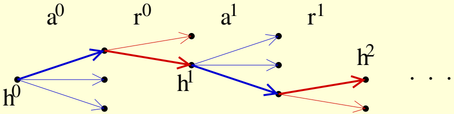
- Given the history $h^t = (a^0, r^0, a^1, r^1, a^2, r^2, \ldots, a^{t-1}, r^{t-1})$,
- Pick an arm a^t to sample (or "pull"), and
- Obtain a reward r^t drawn from the distribution corresponding to arm a^t.

- T is the total sampling budget, or the horizon.

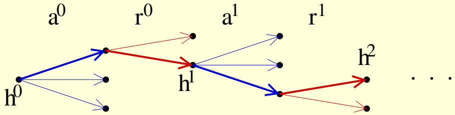
- Formally: a deterministic algorithm is a mapping
  from the set of all histories
  to the set of all arms.
- Formally: a randomised algorithm is a mapping
  from the set of all histories
  to the set of all probability distributions over arms.

- **Note**: The algorithm picks the arm to pull; the bandit instance returns the reward.

Illustration

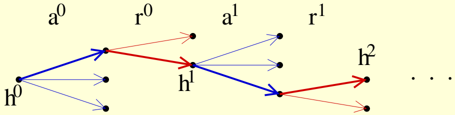


Illustration



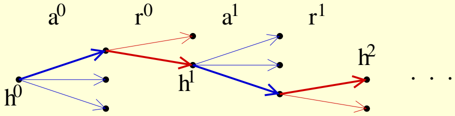
- Consider $h^T = (a^0, r^0, a^1, r^1, \ldots, a^{T-1}, r^{T-1})$.

Illustration

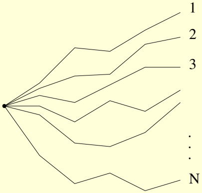


- Consider $h^T = (a^0, r^0, a^1, r^1, \ldots, a^{T-1}, r^{T-1})$.
  Observe that $\mathbb{P}\{h^T\} = \prod_{t=0}^{T-1} \mathbb{P}\{a^t|h^t\}\mathbb{P}\{r^t|a^t\}$, where
  $\mathbb{P}\{a^t|h^t\}$ is decided by the algorithm, and
  $\mathbb{P}\{r^t|a^t\}$ comes from the bandit instance.

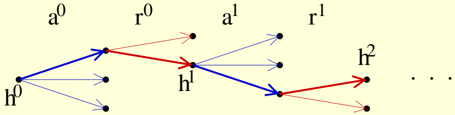
Illustration



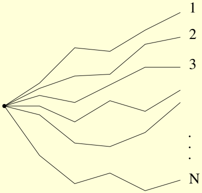
- Consider $h^T = (a^0, r^0, a^1, r^1, \ldots, a^{T-1}, r^{T-1})$.
  Observe that $\mathbb{P}\{h^T\} = \prod_{t=0}^{T-1} \mathbb{P}\{a^t | h^t\}\mathbb{P}\{r^t | a^t\}$, where
  $\mathbb{P}\{a^t | h^t\}$ is decided by the algorithm, and
  $\mathbb{P}\{r^t | a^t\}$ comes from the bandit instance.
- An algorithm, bandit instance pair can generate many possible T-length histories.



Illustration



- Consider $h^T = (a^0, r^0, a^1, r^1, \ldots, a^{T-1}, r^{T-1})$.
  Observe that $\mathbb{P}\{h^T\} = \prod_{t=0}^{T-1} \mathbb{P}\{a^t|h^t\}\mathbb{P}\{r^t|a^t\}$, where
  $\mathbb{P}\{a^t|h^t\}$ is decided by the algorithm, and
  $\mathbb{P}\{r^t|a^t\}$ comes from the bandit instance.
- An algorithm, bandit instance pair can generate many possible T-length histories.



- How many histories possible if the algorithm is deterministic and rewards 0–1?

Multi-armed Bandits

1. The exploration-exploitation dilemma
2. Definitions: Bandit, Algorithm
3. ϵ-greedy algorithms
4. Evaluating algorithms: Regret

ϵ-greedy Strategies

- Parameter $\epsilon \in [0, 1]$ controls the amount of exploration.

ϵ-greedy Strategies

- Parameter $\epsilon \in [0, 1]$ controls the amount of exploration.
- ϵG1
  - If $t \leq \epsilon T$, sample an arm uniformly at random.
  - At $t = \lfloor \epsilon T \rfloor$, identify a^{best}, an arm with the highest empirical mean.
  - If $t > \epsilon T$, sample a^{best}.

ϵ-greedy Strategies

- Parameter $\epsilon \in [0, 1]$ controls the amount of exploration.
- ϵG1
  - If $t \leq \epsilon T$, sample an arm uniformly at random.
  - At $t = \lfloor \epsilon T \rfloor$, identify a^{best}, an arm with the highest empirical mean.
  - If $t > \epsilon T$, sample a^{best}.
- ϵG2
  - If $t \leq \epsilon T$, sample an arm uniformly at random.
  - If $t > \epsilon T$, sample an arm with the highest empirical mean.

ϵ-greedy Strategies

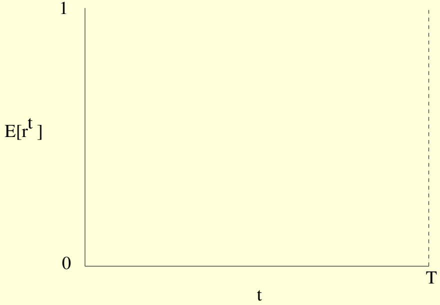
- Parameter $\epsilon \in [0, 1]$ controls the amount of exploration.
- ϵG1
  - If $t \leq \epsilon T$, sample an arm uniformly at random.
  - At $t = \lfloor \epsilon T \rfloor$, identify a^{best}, an arm with the highest empirical mean.
  - If $t > \epsilon T$, sample a^{best}.
- ϵG2
  - If $t \leq \epsilon T$, sample an arm uniformly at random.
  - If $t > \epsilon T$, sample an arm with the highest empirical mean.
- ϵG3
  - With probability ϵ, sample an arm uniformly at random; with probability $1 - \epsilon$, sample an arm with the highest empirical mean.

Multi-armed Bandits

1. The exploration-exploitation dilemma
2. Definitions: Bandit, Algorithm
3. ϵ-greedy algorithms
4. Evaluating algorithms: Regret

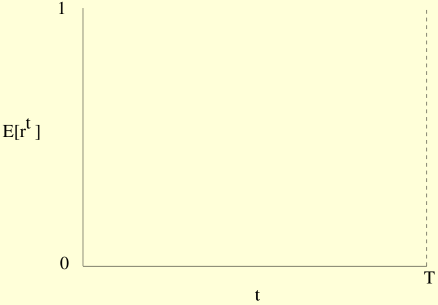
Visualising Performance

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



Visualising Performance

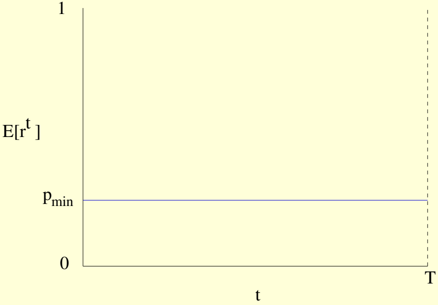
- Consider a plot of $\mathbb{E}[r^t]$ against t.
- What is the least expected reward that can be achieved?



Visualising Performance

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

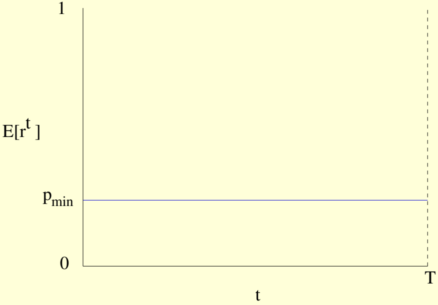
  $p_{\min} = \min_{a \in A} p_a$.



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?

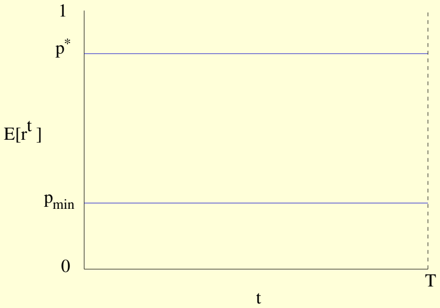


Visualising Performance

- 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?

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

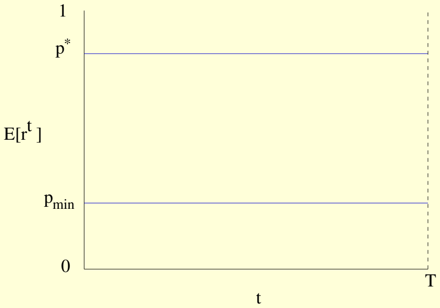


Visualising Performance

- 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?



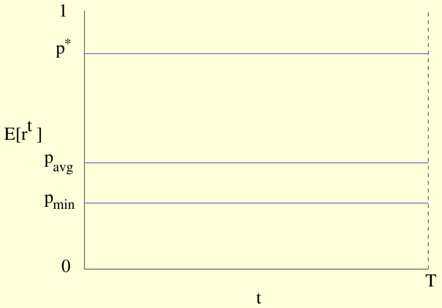
Visualising Performance

- 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?

  $p_{\text{avg}} = \frac{1}{n} \sum_{a \in A} p_a$.



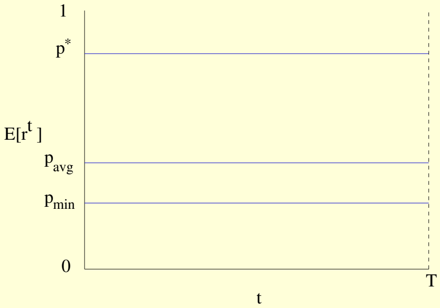
Visualising Performance

- 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?

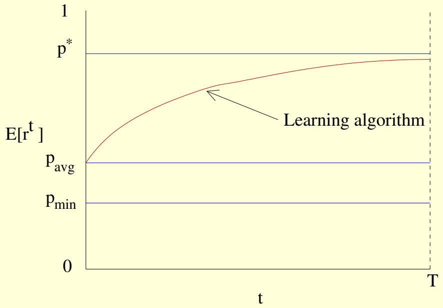
Visualising Performance

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

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

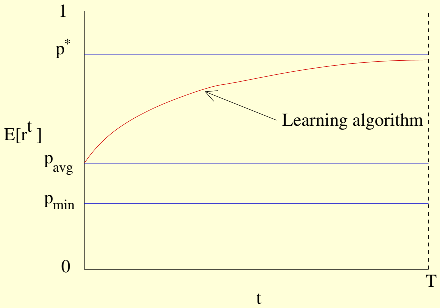
  $p^\star = \max_{a \in A} p_a$.
- If an algorithm pulls arms uniformly at random, what reward will it achieve?

  $p_{\text{avg}} = \frac{1}{n} \sum_{a \in A} p_a$.



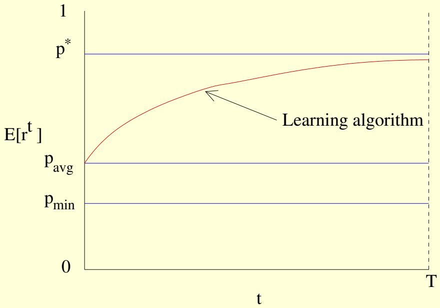
- How will the graph look for a reasonable learning algorithm?

Regret



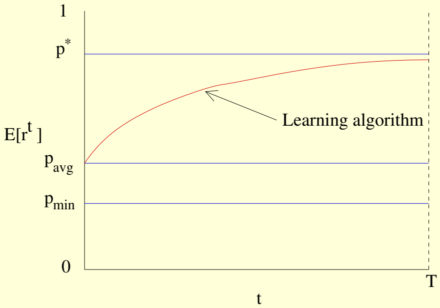
- The maximum achievable expected reward in T steps is $Tp^{\star}$.

Regret



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

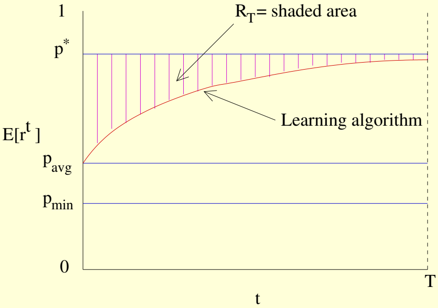
Regret



- 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

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

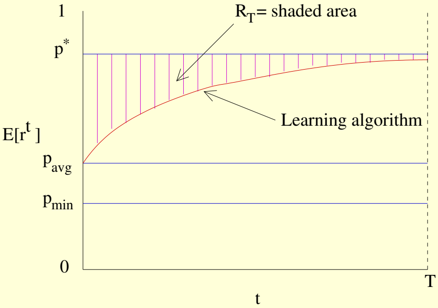
Regret



- 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

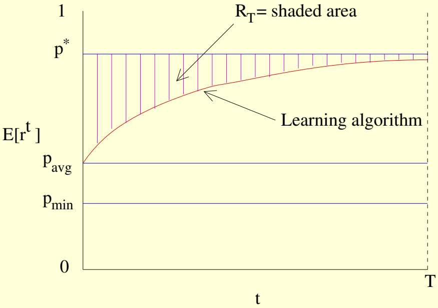


- The maximum achievable expected reward in T steps is $Tp^\star$.
- The actual expected reward for an algorithm is $\sum_{t=0}^{T-1} \mathbb{E}[r^t]$.
- The (expected cumulative) regret of the algorithm for horizon T is the difference

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

- 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?