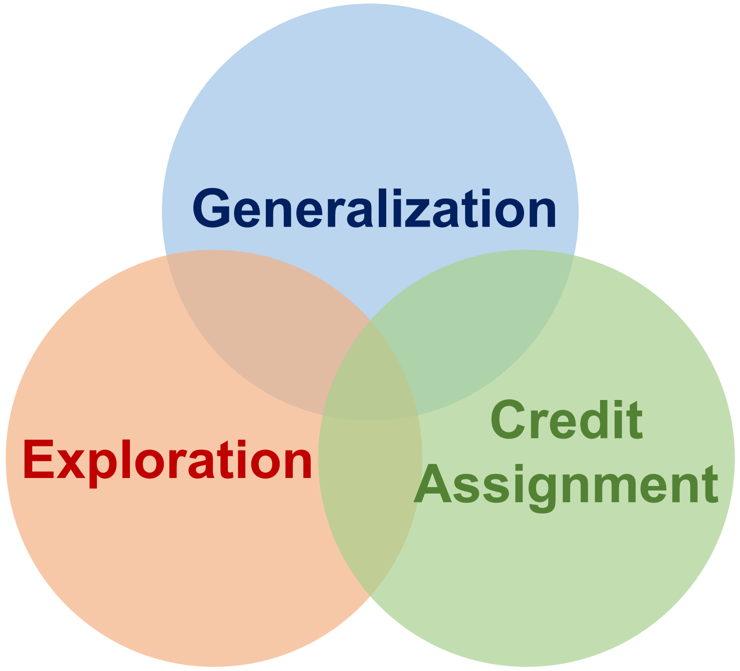
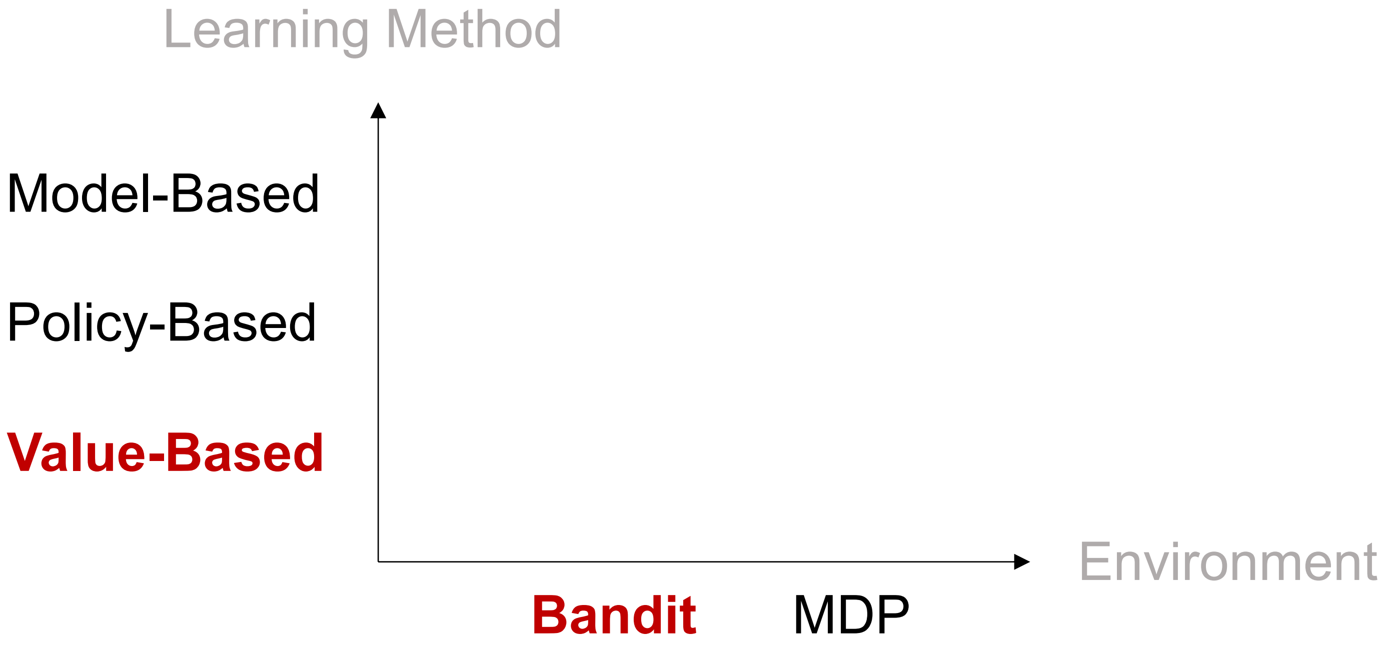
Value-Based Bandit Algorithms

Chen-Yu Wei

Roadmap



Generalization
Exploration
Credit
Assignment



Learning Method
Model-Based
Policy-Based
Value-Based
Environment
Bandit
MDP

Contextual Bandits and Non-Contextual Bandits

Context
(Input)

Contextual Bandit

a_1 ?
$\boldsymbol{a_2}$ $\mathbf{0.5}$
a_3 ?

Policy: $\pi(a|\text{context})$

Non-Contextual Bandit

a_1 ?
$\boldsymbol{a_2}$ $\mathbf{0.5}$
a_3 ?

Policy: $\pi(a)$

Multi-Armed Bandits

Non-Contextual Bandits with Discrete Actions

Multi-Armed Bandits

A slot machine

One-armed bandit

A row of slot machines

Multi-armed bandit

Multi-Armed Bandits

Given: arm set $\mathcal{A} = \{1, \ldots, A\}$

For time $t = 1, 2, \ldots, T$:

Learner chooses an arm $a_t \in \mathcal{A}$

Learner observes $r_t = R(a_t) + w_t$

Arm = Action

Assumption: $R(a)$ is the (hidden) ground-truth reward function

w_t is (zero-mean) noise

Goal: maximize the total reward $\sum_{t=1}^{T} R(a_t)$

Regret = $\max_a TR(a) - \sum_{t=1}^{T} R(a_t)$

Multi-Armed Bandits (MAB)

- Key challenge in MAB: **Exploration**
- The other challenges of RL are not presented in MAB:
  - Generalization (there is no input in MAB)
  - Credit assignments (there is no delayed feedback)

- We will discuss about two categories of exploration strategies
  - Based on mean estimation
  - Based on mean and uncertainty estimation

Multi-Armed Bandits

Based on mean estimation

The Exploration and Exploitation Trade-off in MAB

- To perform as well as the best policy (i.e., best arm), the learner has to pull the best arm most of the time

  ⇒ need to **exploit**

- To identify the best arm, the learner has to try every arm sufficiently many times

  ⇒ need to **explore**

A Simple Strategy: Explore-then-Commit

Explore-then-commit (Parameter: T_0)

In the first T_0 rounds, sample each arm T_0/A times. **(Explore)**

Compute the **empirical mean** $\hat{R}(a)$ for each arm a

In the remaining $T - T_0$ rounds, draw $\hat{a} = \mathrm{argmax}_a \hat{R}(a)$ **(Exploit)**

What is the *right* amount of exploration (T_0)?

Another Simple Strategy: ϵ-Greedy

Mixing exploration and exploitation in time

ϵ-Greedy (Parameter: ϵ)

Take action

$$a_t = \begin{cases} \text{uniform}(\mathcal{A}) & \text{with prob. } \epsilon & \textbf{(Explore)} \\ \text{argmax}_a \hat{R}_t(a) & \text{with prob. } 1-\epsilon & \textbf{(Exploit)} \end{cases}$$

where $\hat{R}_t(a) = \frac{\sum_{s=1}^{t-1} \mathbb{I}\{a_s=a\}\, r_s}{\sum_{s=1}^{t-1} \mathbb{I}\{a_s=a\}}$ is the empirical mean of arm a using samples up to time $t-1$.

What is the *right* amount of exploration (ϵ)?

Comparison

- ϵ-Greedy is more **robust to non-stationarity** than Explore-then-Commit
- ϵ-Greedy has a better performance in the early phase of the learning process

Quantifying the Estimation Error

In Explore-then-Commit, we obtain $N = T_0 / A$ i.i.d. samples of each arm.

Key Question:

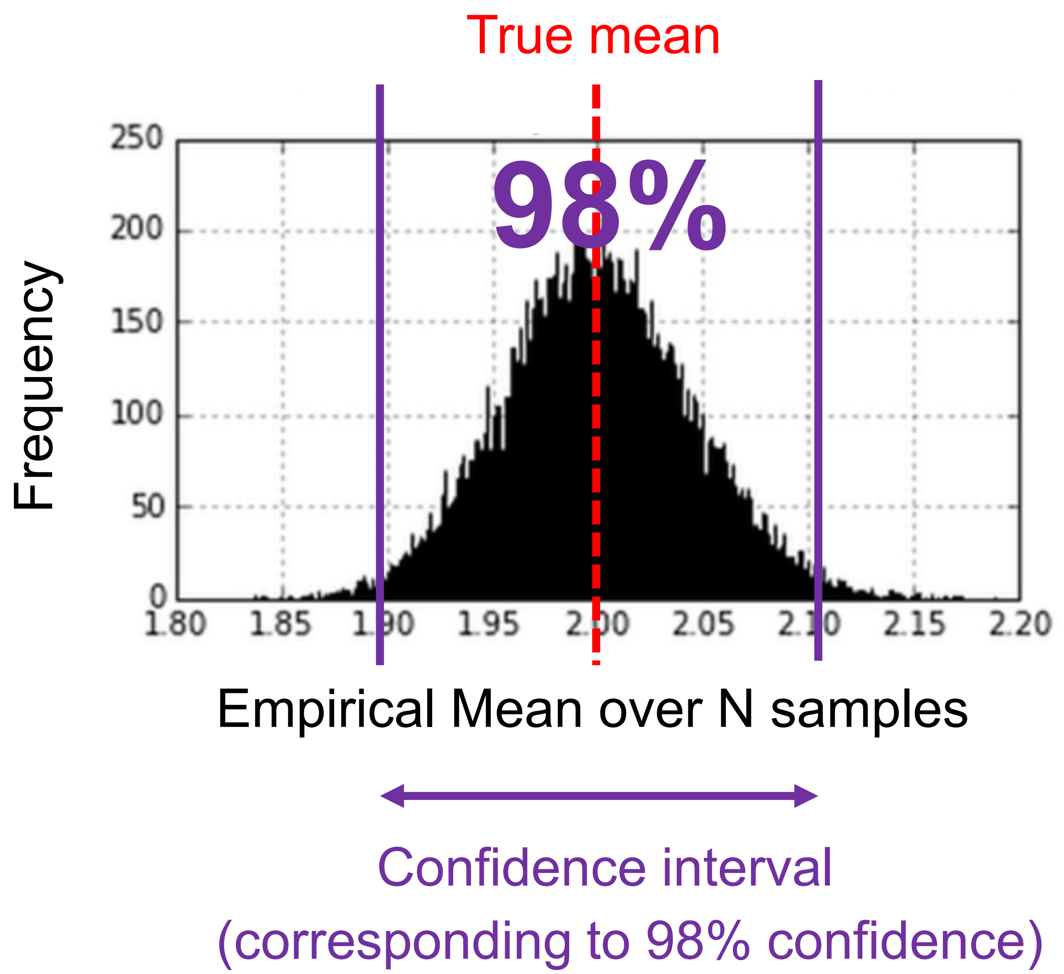
$$\left| \hat{R}(a) - R(a) \right| \leq ? \; f(N)$$

should decrease with N

Empirical mean of N independent samples

True mean

Quantifying the Estimation Error



Quantifying the Error: Concentration Inequality

Theorem. Hoeffding's Inequality

Let $X_1, \dots, X_N \in [-1,1]$ be independent random variables with mean μ.

Then with probability at least $\underset{1.98}{\underline{1-\delta}}$, $\boxed{\delta = 0.02}$

$$\left| \frac{1}{N} \sum_{i=1}^{N} X_i - \mu \right| \leq \sqrt{\frac{2\log(2/\delta)}{N}}\ .$$

Quantifying the Estimation Error

In Explore-then-Commit, we obtain $N = T_0/A$ independent samples of each arm.

With probability **0.99**,

$$\left|\,\hat{R}(a) - R(a)\,\right| \leq ?\ \ f(N) \approx \underline{c_{0.99}}\sqrt{\frac{1}{N}}$$

Empirical mean of N independent samples

True mean

Calculating the Regret for Explore-then-Commit (1/4)

In the first T_0 rounds, sample each arm T_0/A times. **(Explore)**

Compute the **empirical mean** $\hat{R}(a)$ for each arm a

In the remaining $T - T_0$ rounds, draw $\underline{\hat{a} = \mathrm{argmax}_a \hat{R}(a)}$ **(Exploit)**

$\boxed{a^* = \mathrm{argmax}_a R(a)}$

Regret $= TR(a^\star) - \sum_{t=1}^{T} R(a_t) \;\; = \sum_{t=1}^{T} (R(a^\star) - R(a_t))$

Assume $R(a) \in [0,1]$ for simplicity.

Calculating the Regret for Explore-then-Commit (2/4)

In the first T_0 rounds, sample each arm T_0/A times. **(Explore)**

Compute the **empirical mean** $\hat{R}(a)$ for each arm a

In the remaining $T - T_0$ rounds, draw $\hat{a} = \mathrm{argmax}_a \hat{R}(a)$ **(Exploit)**

$R(a) \in [0,1]$

Exploration Phase a_t is chosen evenly across arms

$$R(a^\star) - R(a_t) \leq 1$$

Calculating the Regret for Explore-then-Commit (3/4)

In the first T_0 rounds, sample each arm T_0/A times. **(Explore)**

Compute the **empirical mean** $\hat{R}(a)$ for each arm a

In the remaining $T - T_0$ rounds, draw $\hat{a} = \operatorname{argmax}_a \hat{R}(a)$ **(Exploit)**

Exploitation Phase $a_t = \operatorname{argmax}_a \hat{R}(a)$ ①

For all arm a, $\left|\hat{R}(a) - R(a)\right| \le c\sqrt{\dfrac{1}{\#\text{ samples of arm } a}} = c\sqrt{\dfrac{A}{T_0}}$ ②

$$R(a^\star) - R(a_t) = \underbrace{\hat{R}(a^*) - \hat{R}(a_t)}_{\substack{\leq 0 \\ (\text{by } ①)}} + \underbrace{\left[R(a^*) - \hat{R}(a^*)\right]}_{\substack{\leq c\sqrt{\frac{A}{T_0}} \\ (\text{by } ②)}} + \underbrace{\left[\hat{R}(a_t) - R(a_t)\right]}_{\substack{\leq c\sqrt{\frac{A}{T_0}} \\ (\text{by } ②)}} \leq 2c\sqrt{\frac{A}{T_0}}$$

Calculating the Regret for Explore-then-Commit (4/4)

In the first T_0 rounds, sample each arm T_0/A times. **(Explore)**

Compute the **empirical mean** $\hat{R}(a)$ for each arm a

In the remaining $T - T_0$ rounds, draw $\hat{a} = \text{argmax}_a \hat{R}(a)$ **(Exploit)**

$$\textbf{Regret} = \sum_{t=1}^{T} (R(a^\star) - R(a_t))$$

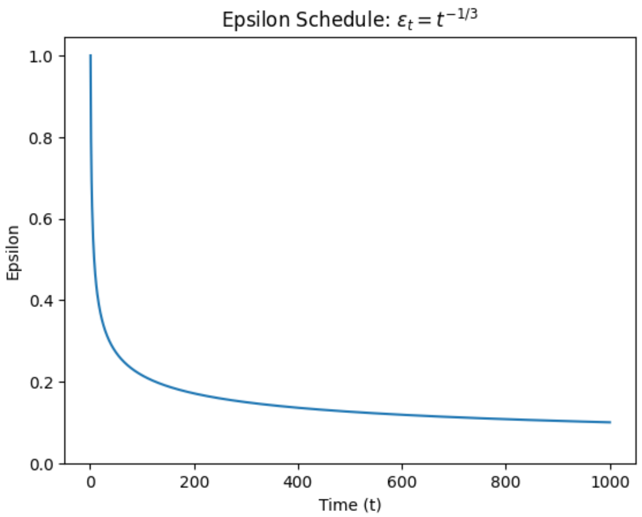
$$\leq \underbrace{T_0 \times 1}_{\substack{\text{Exploration Phase} \\ \text{(regret increases with } T_0\text{)}}} + \underbrace{(T - T_0) \times 2c\sqrt{\frac{A}{T_0}}}_{\substack{\text{Exploitation Phase} \\ \text{(regret decreases with } T_0\text{)}}}$$

How much to spend on exploration?

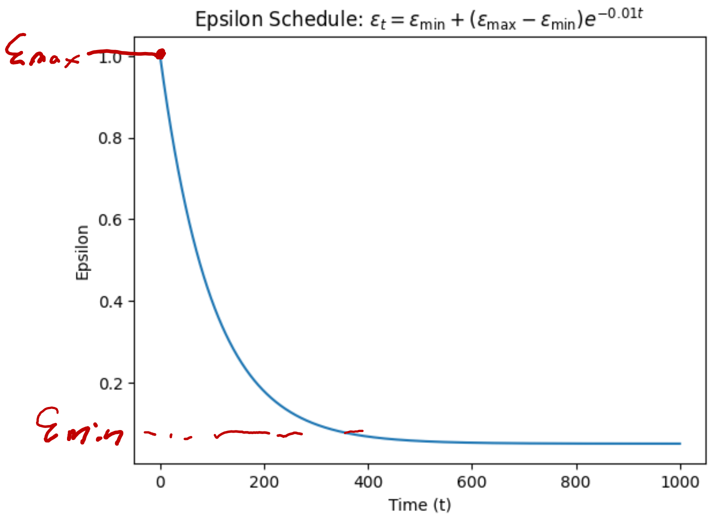
The T_0 that minimizes the regret satisfies roughly $\frac{T_0}{T} \approx \left(\frac{A}{T}\right)^{1/3}$ $\propto T^{-1/3}$

$\Rightarrow$ The percentage of exploration should decrease with time.

In ϵ-greedy, we usually decrease the exploration rate ϵ with time. For example:

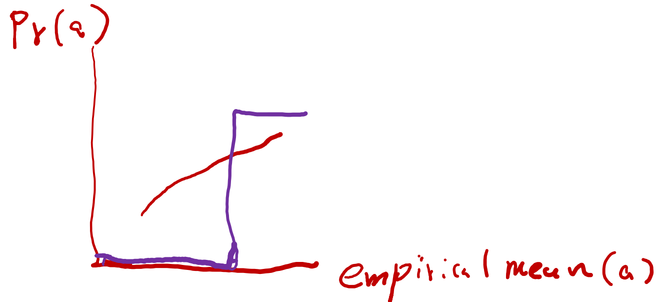


$$\epsilon_t \approx t^{-1/3}$$



$$\epsilon_t \approx \underset{\varepsilon_{min}}{0.05} + (\underset{\varepsilon_{max}}{1} - \underset{\varepsilon_{min}}{\underbrace{0.05}})e^{-0.01t}$$

Can We Do Better?



In explore-then-commit and ϵ-greedy, the probability to choose arms do not depend on the estimated mean (except for the empirically best arm).

… Maybe, the probability of choosing arms can be adaptive to the estimated mean?

Solution: Refine the amount of exploration for each arm **based on the current mean estimation**.

(Has to do this carefully to avoid **under-exploration**)

Refined Exploration

Boltzmann Exploration

(exploration) $\lambda = 0 \Rightarrow \pi_t$ is uniform

(exploitation) $\lambda = \infty \Rightarrow \pi_t$ concentrates to $\underline{\arg\max \hat{R}_t(a)}$

In each round, sample a_t according to

temperature

$$\pi_t(a) \propto \exp\left(\lambda_t \, \hat{R}_t(a)\right)$$

where $\hat{R}_t(a)$ is the empirical mean of arm a using samples up to time $t-1$.

λ_t controls the degree of exploration.

Should λ_t be increasing or decreasing over time?

Summary: MAB Based on Mean Estimation

For $t = 1, 2, \dots, T$,

Design a distribution $\pi_t(\cdot)$ based on the current mean estimation $\hat{R}_t(\cdot)$

EG $\pi_t(a) = (1-\epsilon_t)\mathbb{I}\left\{a = \underset{a'}{\operatorname{argmax}}\, \hat{R}_t(a')\right\} + \frac{\epsilon_t}{A}$

BE $\pi_t(a) \propto \exp\left(\lambda_t \hat{R}_t(a)\right)$

Sample an arm $a_t \sim \pi_t$ and receive the corresponding reward r_t.

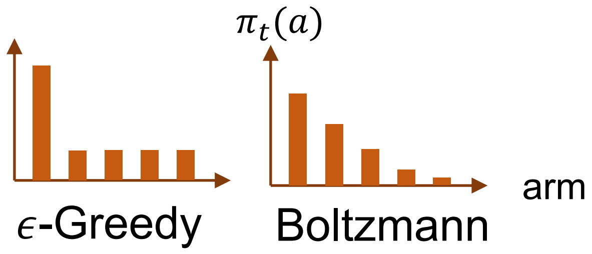
Refine the mean estimation $\hat{R}_{t+1}(\cdot)$ with the new sample (a_t, r_t).

Summary: MAB Based on Mean Estimation

(a_t, r_t)

Mean Estimation $\xrightarrow{\hat{R}_t(\cdot)}$ Decision Rule $\xrightarrow{\pi_t(\cdot)}$ Pick action $a_t \sim \pi_t$
Receive r_t

$$\hat{R}_t(a) = \frac{\sum_{s=1}^{t-1} \mathbb{I}\{a_s = a\} r_s}{\sum_{s=1}^{t-1} \mathbb{I}\{a_s = a\}}$$



$$\pi_t(a) = (1-\epsilon)\mathbb{I}\left\{a = \underset{a'}{\operatorname{argmax}}\ \hat{R}_t(a')\right\} + \frac{\epsilon}{A}$$

$$\pi_t(a) \propto \exp\left(\lambda \hat{R}_t(a)\right)$$

Summary: MAB Based on Mean Estimation

- Both methods are based on the same **mean estimation**
  - ϵ-Greedy, Boltzmann exploration

- The key difference is in the **decision rule**, i.e., the mapping from estimated means $\hat{R}_t$ to a distribution π_t.
  - The shape of the mapping makes differences

- There is a scalar hyperparameter that allows for a tradeoff between exploration and exploitation (ϵ_t in EG, λ_t in BE)

Some Experiments

$T = 10000$ rounds

$A = 2$ arms

Reward mean $R = [0.5, 0.5 - \Delta]$

Bernoulli distribution

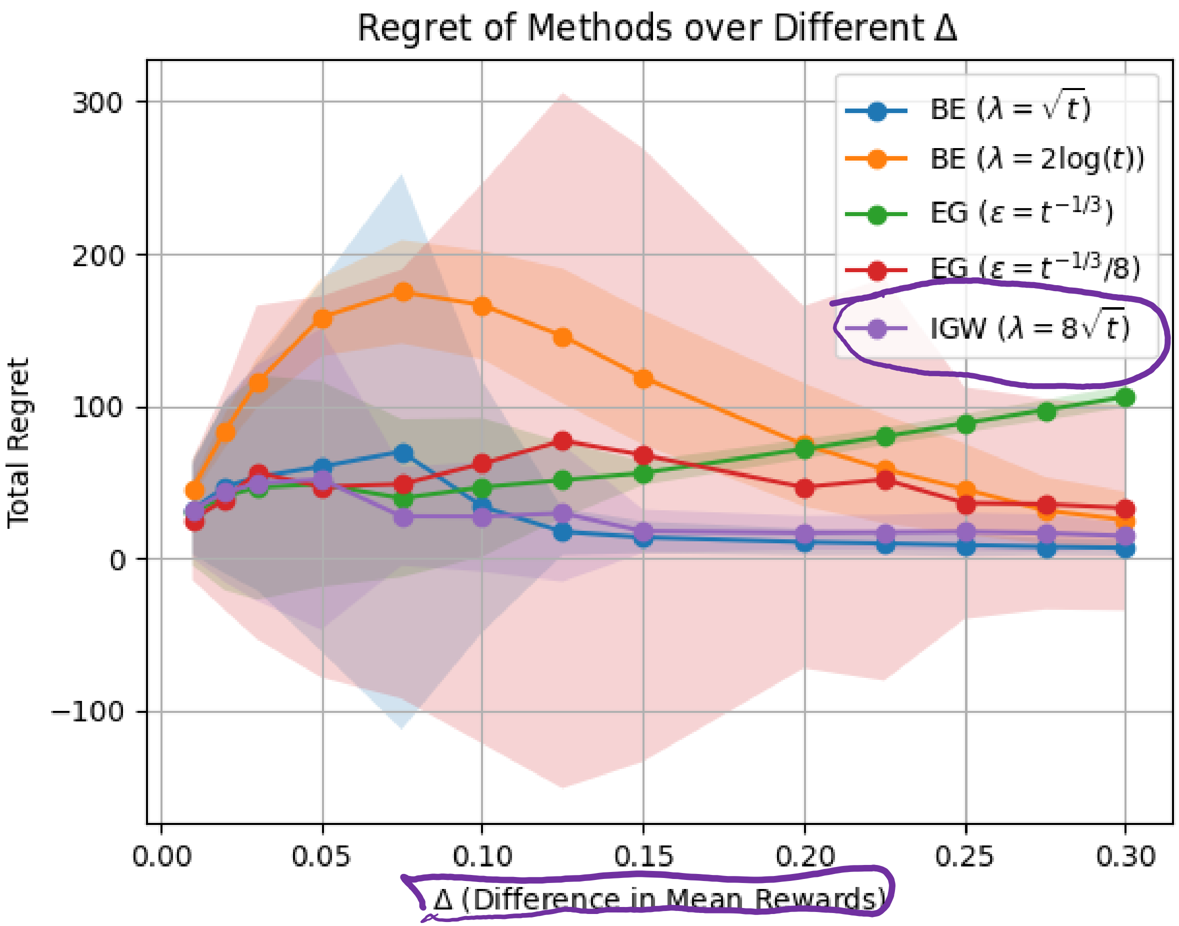
Time-dependent parameters

30 random seeds

code

Observations:

- Most algorithms have its worst regret at some intermediate Δ value
- Smaller exploration leads to larger variation in performance

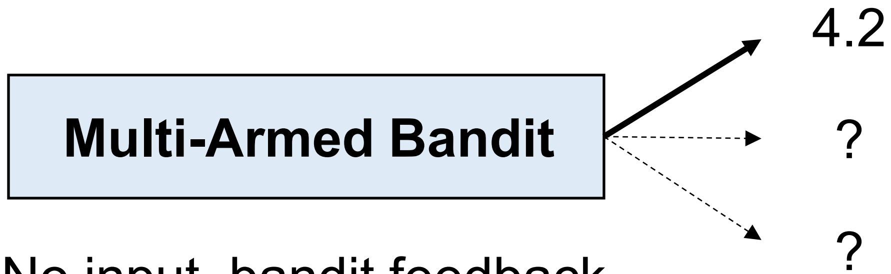


Small Δ is easy: don't need to distinguish the two arms
Large Δ is also easy: easy to distinguish the two arms

Contextual Bandits

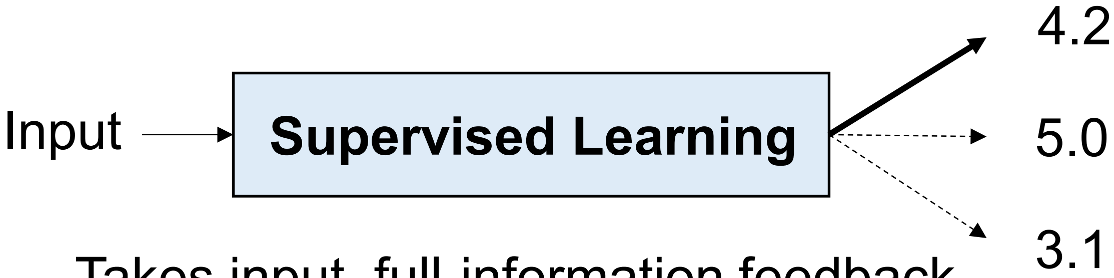
Based on reward function estimation

Contextual Bandits Generalizes MAB and SL



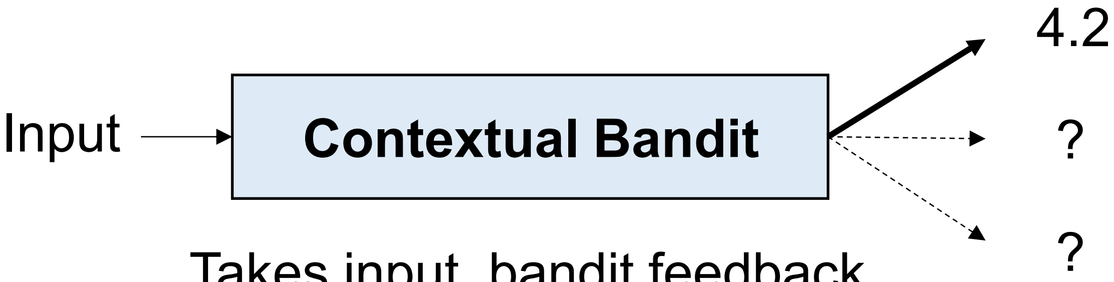
No input, bandit feedback

- Generalization
- **Exploration**
- Credit assignment



Takes input, full-information feedback

- **Generalization**
- Exploration
- Credit assignment



Takes input, bandit feedback

- **Generalization**
- **Exploration**
- Credit assignment

Multi-Armed Bandits vs. Contextual Bandits

Multi-Armed Bandit

all-user recommendation

Context → **Contextual Bandit**

personalized recommendation

E.g. the user’s history, location, social network activity, …

Contextual Bandits

For time $t = 1, 2, \dots, T$:

Environment generates a context $x_t \in \mathcal{X}$

Learner chooses an action $a_t \in \mathcal{A}$

Learner observes $r_t = R(x_t, a_t) + w_t$

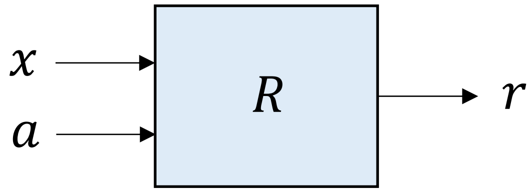
Discussion

- Contextual bandits is a minimal simultaneous generalization of supervised learning (SL) and multi-armed bandits (MAB)
- SL is extensively discussed in machine learning courses
- We just learned some simple MAB algorithms
  - Two strategies based on mean estimation

- **Question:** Can you design a contextual bandits algorithm based on the techniques for SL and MAB?

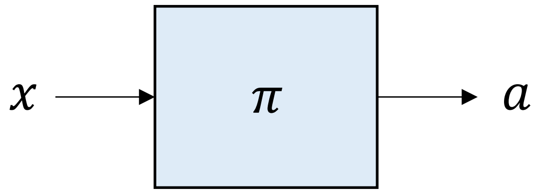
Two ways to leverage SL techniques in CB

x: context, a: action, r: reward



Learn a mapping from
(context, action) to reward

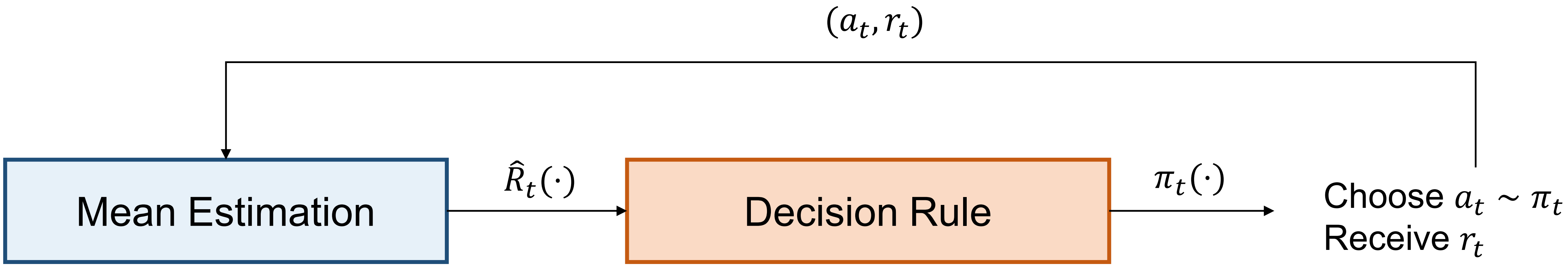
Value-based approach
(discussed next)



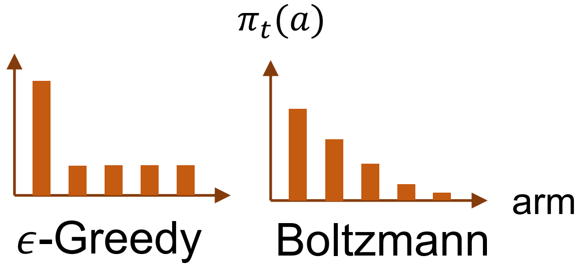
Learn a mapping from
context to action (or action distribution)

Policy-based approach
(slightly later in the course)

Recall: MAB Based on Mean Estimation



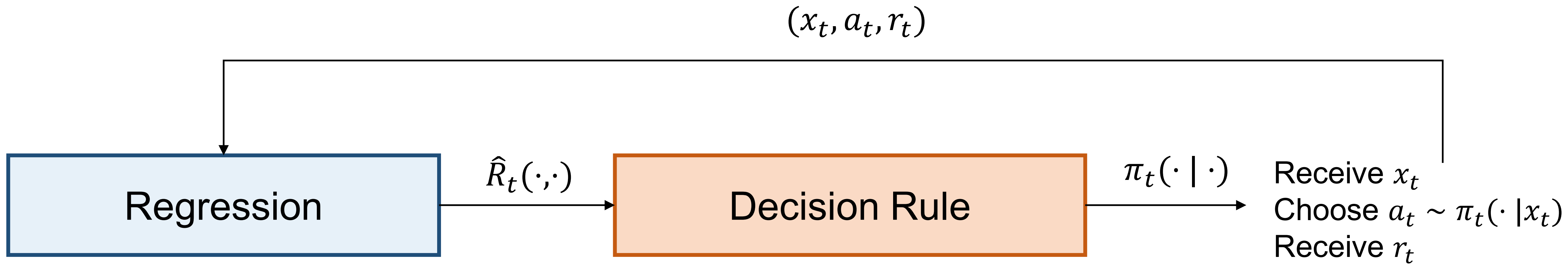
$$\hat{R}_t(a) = \frac{\sum_{s=1}^{t-1} \mathbb{I}\{a_s = a\} r_s}{\sum_{s=1}^{t-1} \mathbb{I}\{a_s = a\}}$$

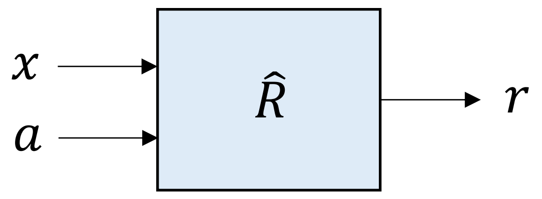


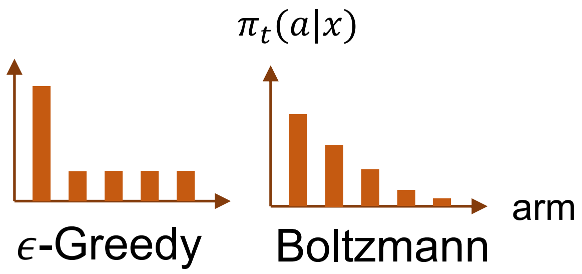
$$\pi_t(a) = (1 - \epsilon_t)\mathbb{I}\left\{a = \underset{a'}{\operatorname{argmax}} \hat{R}_t(a')\right\} + \frac{\epsilon_t}{A}$$

$$\pi_t(a) \propto \exp\left(\lambda_t \hat{R}_t(a)\right)$$

CB Based on Reward Function Estimation (Regression)





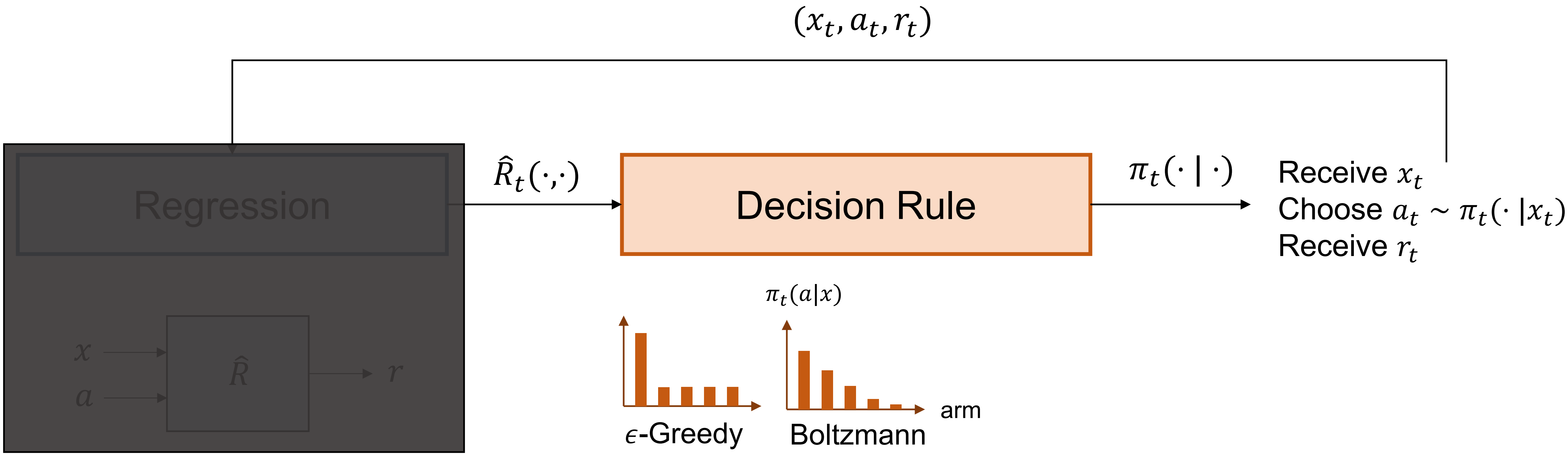


Train a $\hat{R}$ such that $r_i \approx \hat{R}(x_i, a_i)$

$$\pi_t(a|x) = (1 - \epsilon_t)\mathbb{I}\left\{a = \underset{a'}{\operatorname{argmax}}\, \hat{R}_t(x, a')\right\} + \frac{\epsilon_t}{A}$$

$$\pi_t(a|x) \propto \exp\left(\lambda_t \hat{R}_t(x, a)\right)$$

CB Based on Reward Function Estimation (Regression)



Use any supervised learning technique to find a function

$$\hat{R}(x,a) \approx r$$

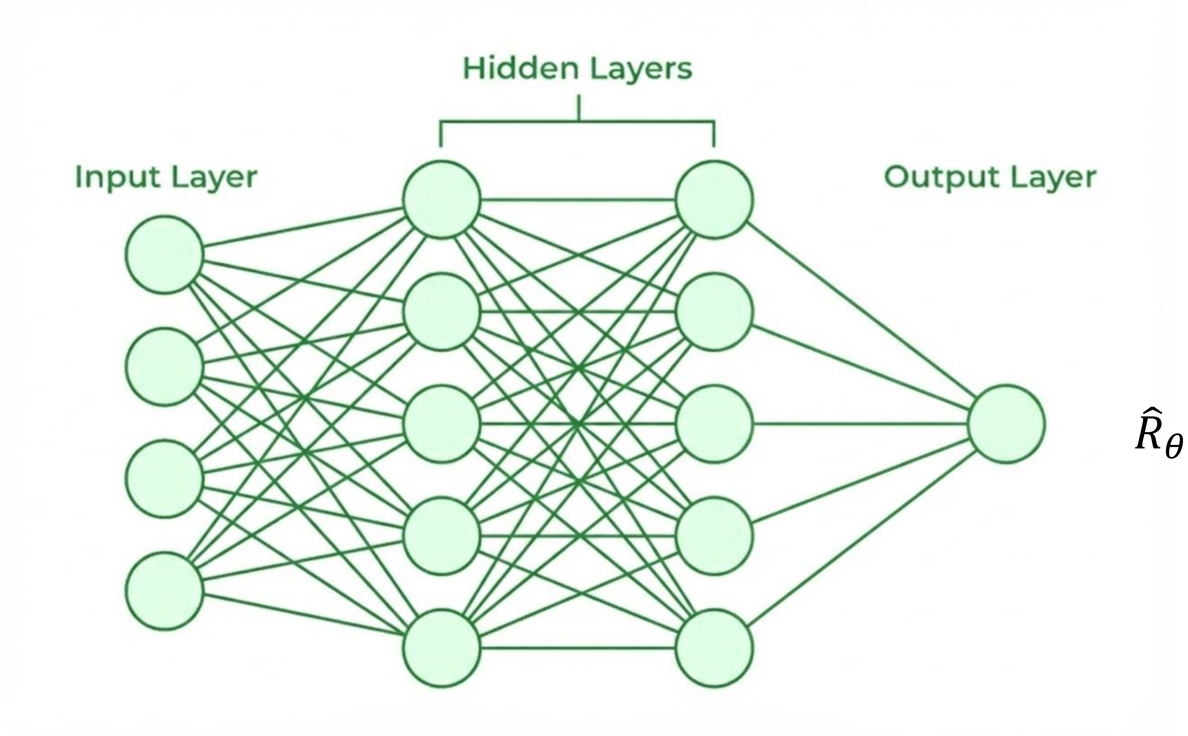
$$\pi_t(a|x) = (1-\epsilon_t)\mathbb{I}\left\{a = \operatorname*{argmax}_{a'} \hat{R}_t(x,a')\right\} + \frac{\epsilon_t}{A}$$

$$\pi_t(a|x) \propto \exp\left(\lambda_t \hat{R}_t(x,a)\right)$$

The Regression Procedure

Context x_t ⟶

Action a_t ⟶

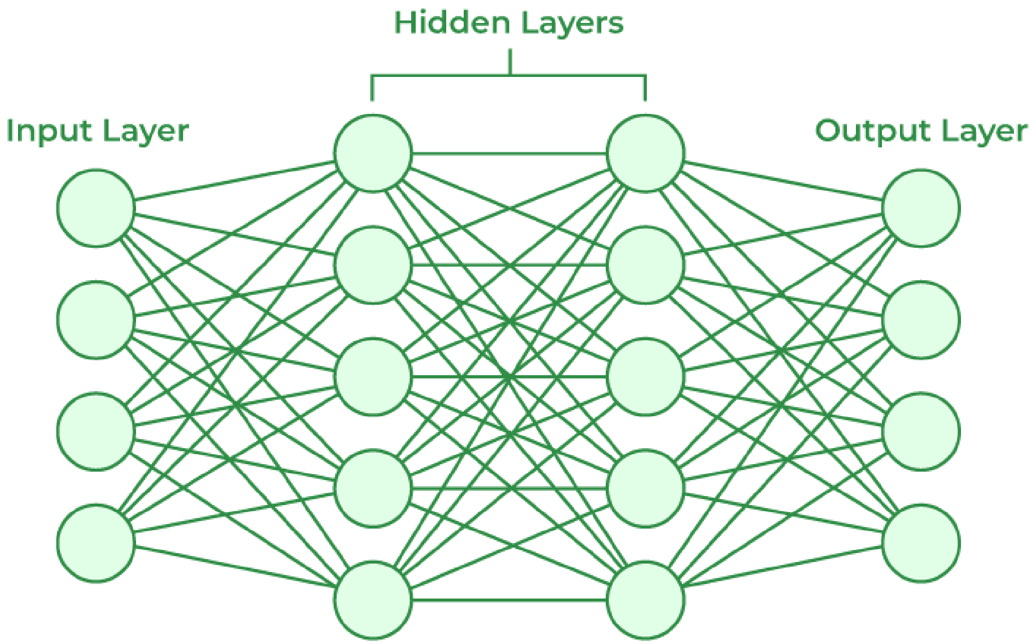


$\hat{R}_\theta(x_t, a_t)$

Training loss: $L(\theta) = \left(\hat{R}_\theta(x_t, a_t) - r_t\right)^2$

The Regression Procedure

Context x_t $\longrightarrow$



$\hat{R}_\theta(x_t, 1)$

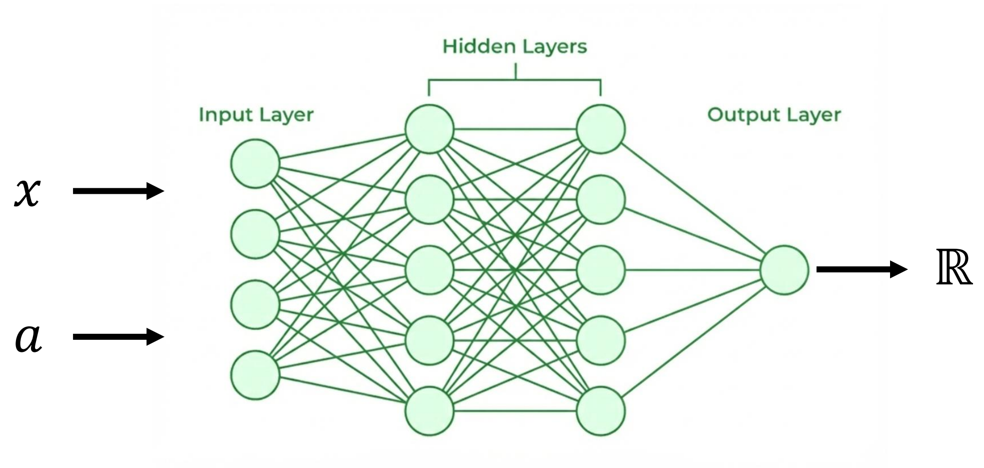
$\hat{R}_\theta(x_t, 2)$

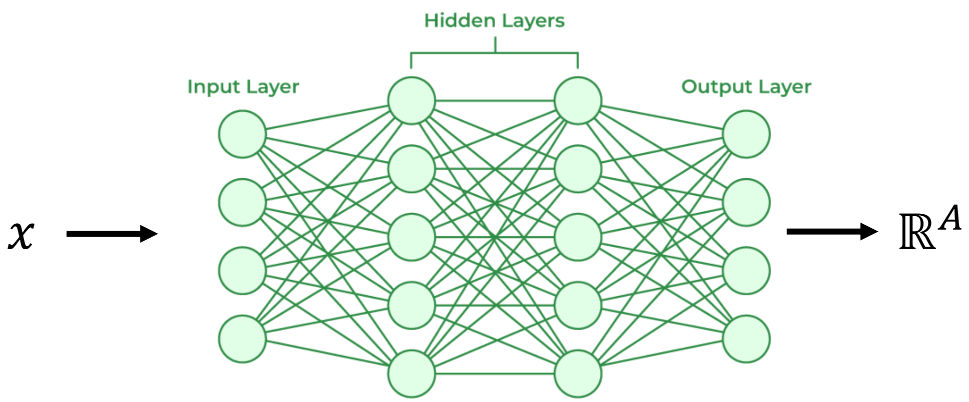
$\vdots$

$\hat{R}_\theta(x_t, A)$

Training loss: $L(\theta) = \left(\hat{R}_\theta(x_t, a_t) - r_t\right)^2$

The Regression Procedure: Comparison





Suitable when
1) actions are continuous,
2) available action set changes with contexts

CB Based on Reward Function Estimation

Instantiate a regression procedure $\hat{R}_1$

For $t = 1, 2, \dots, T$,

Receive context x_t

Design a distribution $\pi_t(\cdot|x_t)$ based on the estimated reward $\hat{R}_t(x_t,\cdot)$

EG $\pi_t(a|x_t) = (1-\epsilon_t)\mathbb{I}\left\{a = \underset{a'}{\operatorname{argmax}}\, \hat{R}_t(x_t, a')\right\} + \frac{\epsilon_t}{A}$

BE $\pi_t(a|x_t) \propto \exp\left(\lambda_t \hat{R}_t(x_t, a)\right)$

Sample an action $a_t \sim \pi_t(\cdot\,|x_t)$ and receive the corresponding reward r_t.

Refine the reward estimator $\hat{R}_{t+1}(\cdot,\cdot)$ with the new sample (x_t, a_t, r_t).

CB Based with Neural Networks and Batches

Instantiate a reward network $\hat{R}_\theta$

For $t = 1, 2, \dots, T$,

For $i = 1, 2, \dots, n$:

Receive context $x_{t,i}$

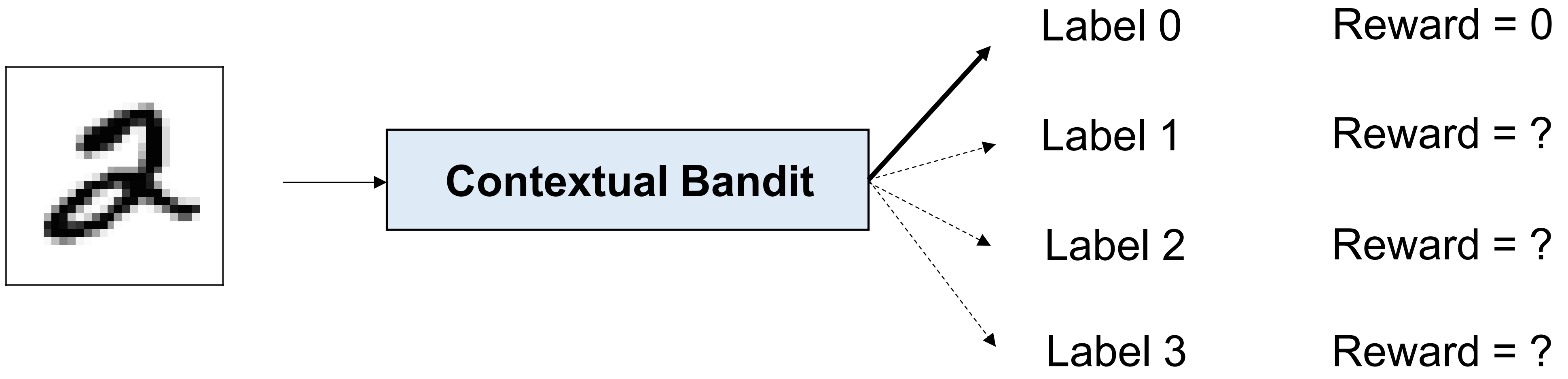
Design a distribution $\pi_t(a|x_{t,i})$ based on the estimated reward $\hat{R}_\theta(x_{t,i}, a)$

Sample an action $a_{t,i} \sim \pi_t(\cdot\,|x_{t,i})$ and receive reward $r_{t,i}$.

For $j = 1,2, \dots, m$:

$$\theta \leftarrow \theta - \alpha \, \nabla_\theta \frac{1}{n} \sum_{i=1}^{n} \left(\hat{R}_\theta\left(x_{t,i}, a_{t,i}\right) - r_{t,i} \right)^2$$

Homework 1



Contextual Bandit
Label 0
Reward = 0
Label 1
Reward = ?
Label 2
Reward = ?
Label 3
Reward = ?

Summary

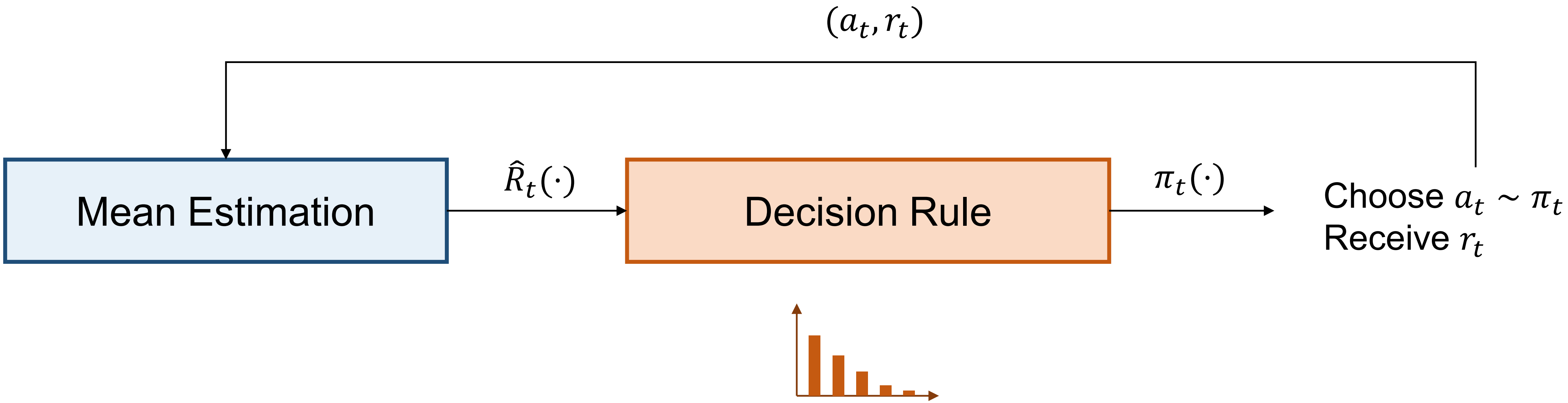
- Contextual bandits (CB) simultaneously generalizes supervised learning (SL) and multi-armed bandits (MAB). It captures the challenges of **generalization** and **exploration** in online RL.

- Any MAB algorithm based on “**mean estimation**” can be converted to a CB algorithm with “**reward function estimation**” by leveraging a regression.
    - This gives a general framework for value-based CB

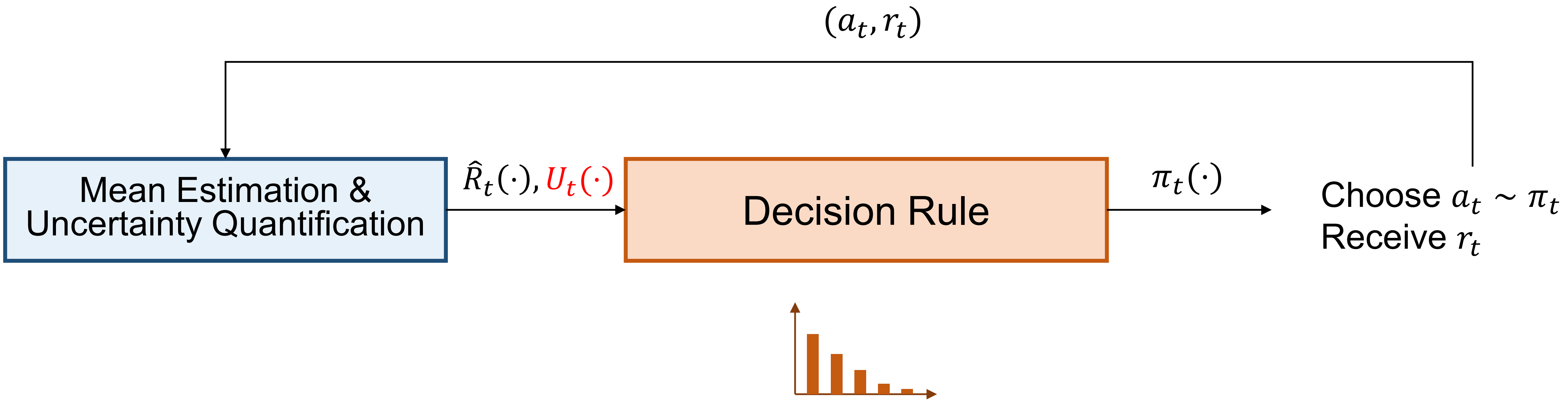
Multi-Armed Bandits

Based on mean estimation and uncertainty quantification

Recall: MAB Based on Mean Estimation



MAB Based on Mean Estimation and Uncertainty Quantification



$U_t(a)$: quantifies the uncertainty of $\hat{R}_t(a)$

$$\left|\hat{R}_t(a) - R(a)\right| \le c\sqrt{\frac{1}{N_t(a)}} \triangleq U_t(a)$$

Useful Idea: "Optimism in the Face of Uncertainty"

In words:

Act according to the **best plausible world**.

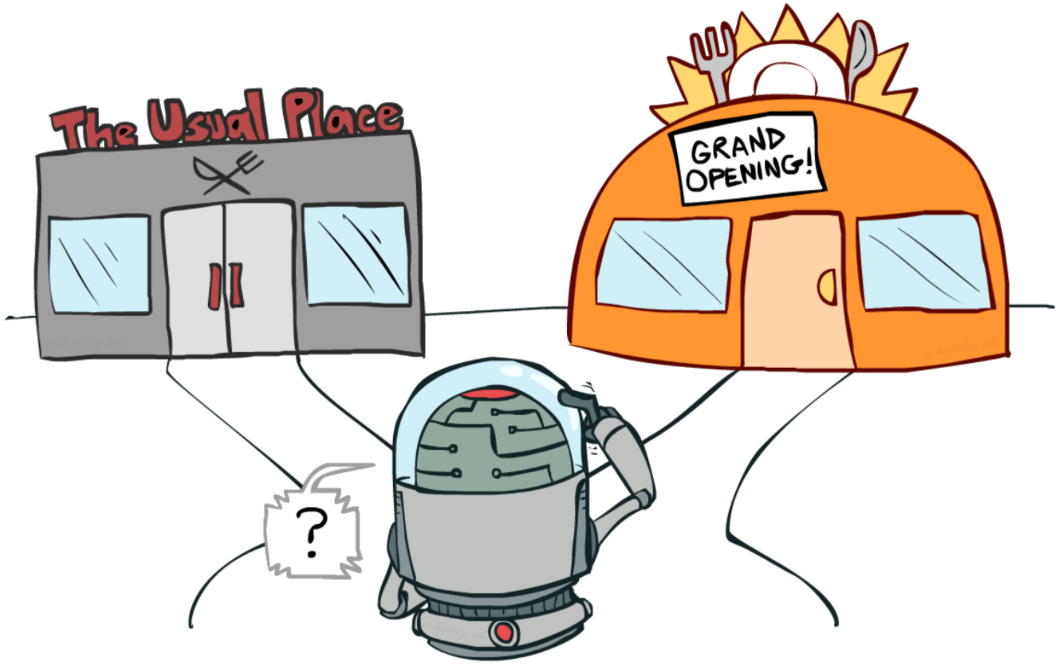


Image source: UC Berkeley CS188

Another Idea: “Optimism in the Face of Uncertainty”

In words:

Act according to the **best plausible world**.

At time t, suppose that arm a has been drawn for $N_t(a)$ times, with empirical mean $\hat{R}_t(a)$.

What can we say about the true mean $R(a)$?

$$\left| R(a) - \hat{R}_t(a) \right| \leq c\sqrt{\frac{1}{N_t(a)}} \quad \text{w.p.} \geq 0.99$$

What’s the most optimistic mean estimation for arm a?

$$\hat{R}_t(a) + c\sqrt{\frac{1}{N_t(a)}}$$

Upper Confidence Bound (UCB)

UCB (Parameter: c)

In round t, draw

Exploration Bonus
= Amount of Uncertainty

$$a_t = \mathrm{argmax}_a \quad \hat{R}_t(a) + c\sqrt{\frac{2\log t}{N_t(a)}}$$

where $\hat{R}_t(a)$ is the empirical mean of arm a using samples up to time $t-1$.
$N_t(a)$ is the number of samples of arm a up to time $t-1$.

cf. Mean-estimation-based algorithms samples $a_t \sim \pi_t(\cdot) =$ an increasing function of $\hat{R}_t(\cdot)$

In those algorithms, Hoeffding's inequality is used in the **regret analysis**, but not in the **algorithm**.

Visualizing UCB

True mean: [0.2, 0.4, 0.6, 0.7]   animation   code

Summary: Algorithms We Learned So Far

	Approach
Explore-then-Exploit ϵ-Greedy Boltzmann Exploration Inverse Gap Weighting	Mean estimation + decision rule
Upper Confidence Bound	Mean estimation + uncertainty quantification + decision rule

Summary

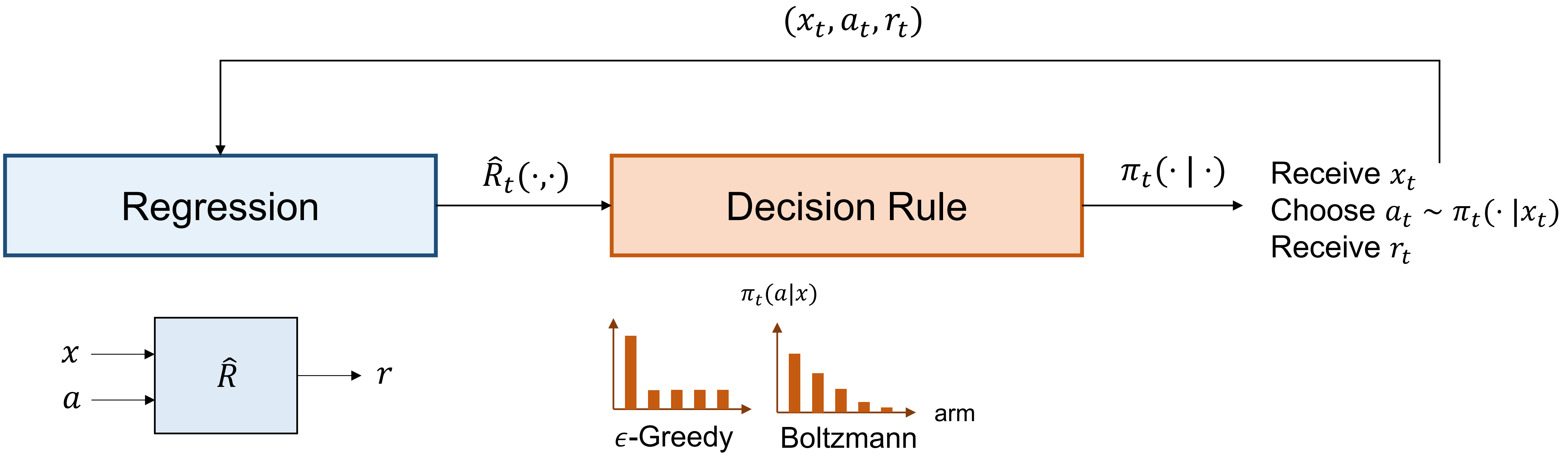
Summary

Value-based bandit algorithms

- Multi-armed bandits (non-contextual bandits)
  - Based on mean estimation
  - Based on mean estimation and uncertainty quantification
- Contextual bandits
  - Based on reward function estimation

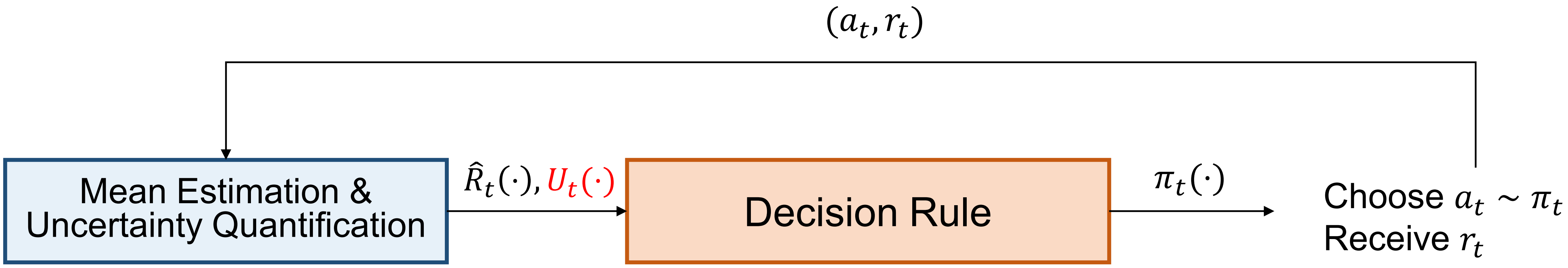
CB Based on Reward Function Estimation

(Special Case: MAB Based on Mean Estimation)



Train a $\hat{R}$ such that $r_i \approx \hat{R}(x_i, a_i)$

MAB Based on Mean and Uncertainty Estimation



UCB: $\underset{a}{\operatorname{argmax}}\ \hat{R}(a) + U_t(a)$

TS: $\underset{a}{\operatorname{argmax}}\ \hat{R}(a) + U_t(a) n_t(a)$

noise

Uncertainty quantification for CB is less trivial – discussed in the future (special topics).