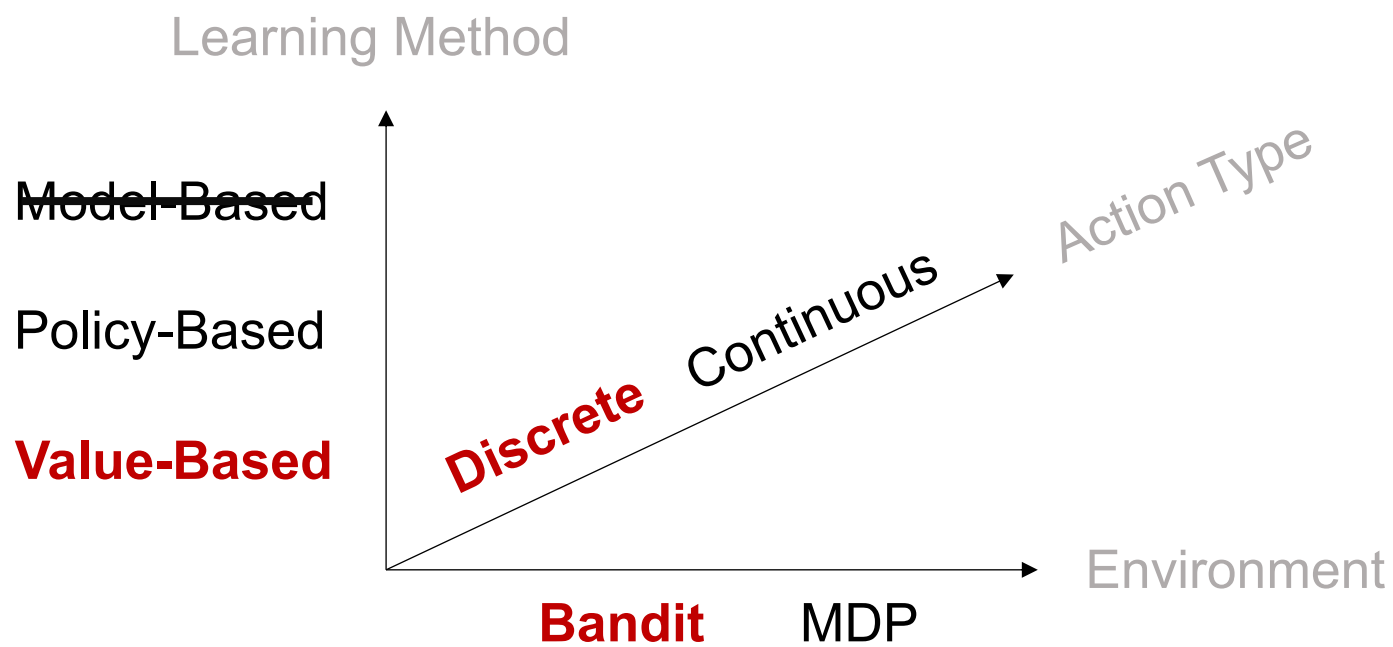
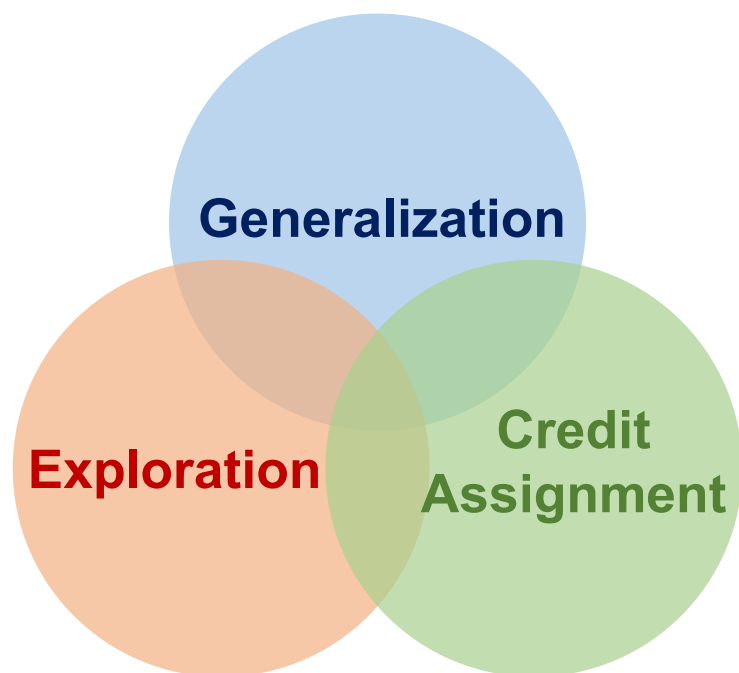


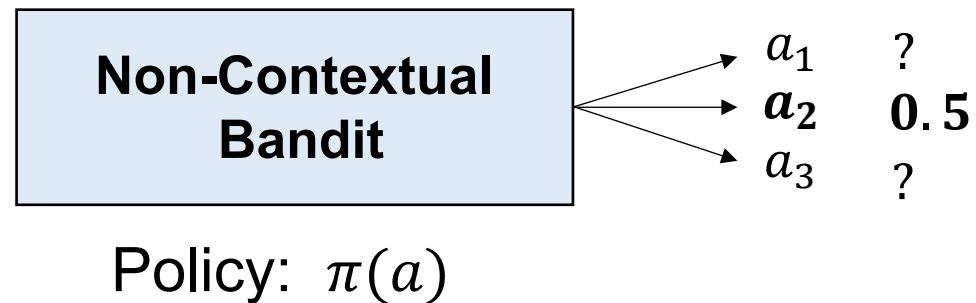
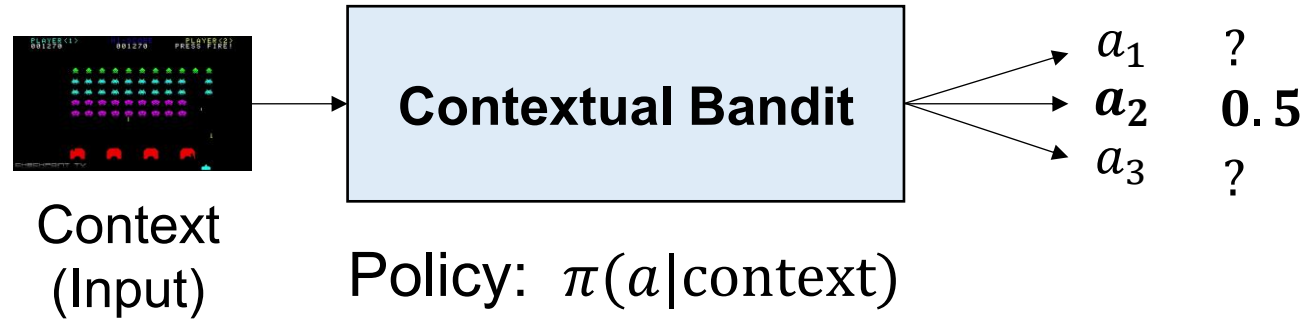
Bandits 1

Chen-Yu Wei

Roadmap



Contextual Bandits and Non-Contextual Bandits



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 $\mathcal{A} = \{1, \dots, A\}$

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

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 $\sum_{t=1}^T R(a_t)$ (or $\sum_{t=1}^T r_t$)

How to Evaluate an Algorithm's Performance?

$$\text{Regret} := \underbrace{\max_{\pi} \sum_{t=1}^T R(\pi)}_{\text{The total reward of the best policy}} - \sum_{t=1}^T R(a_t) = \overset{\substack{\uparrow \\ \text{In MAB}}}{\max_a TR(a)} - \sum_{t=1}^T R(a_t)$$

- “My algorithm obtains $0.3T$ total reward within T rounds” – Is my algorithm good?
- “My algorithm ensures $\text{Regret} \leq 5T^{3/4}$ ”
- $\text{Regret} = o(T) \Rightarrow$ the algorithm is as good as the optimal policy *asymptotically*
- Remark: the learner doesn't need to know or track the regret when running the algorithm. Regret is just an analytical tool to analyze the algorithm in hindsight.

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) asymptotically, 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 $\hat{a} = \operatorname{argmax}_a \hat{R}(a)$ **(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

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

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-Exploit
- ϵ -Greedy has a better performance in the early phase of the learning process

Quantifying the Estimation Error

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

Key Question:

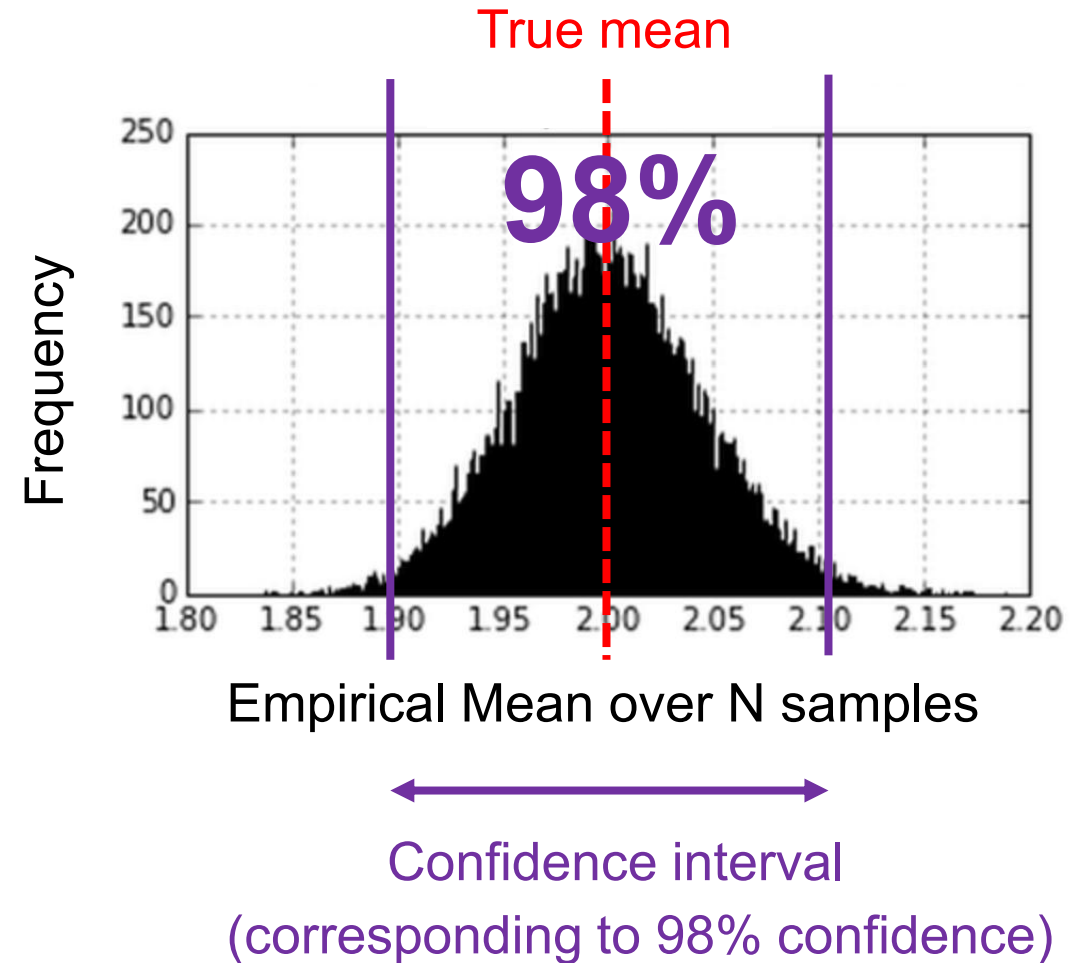
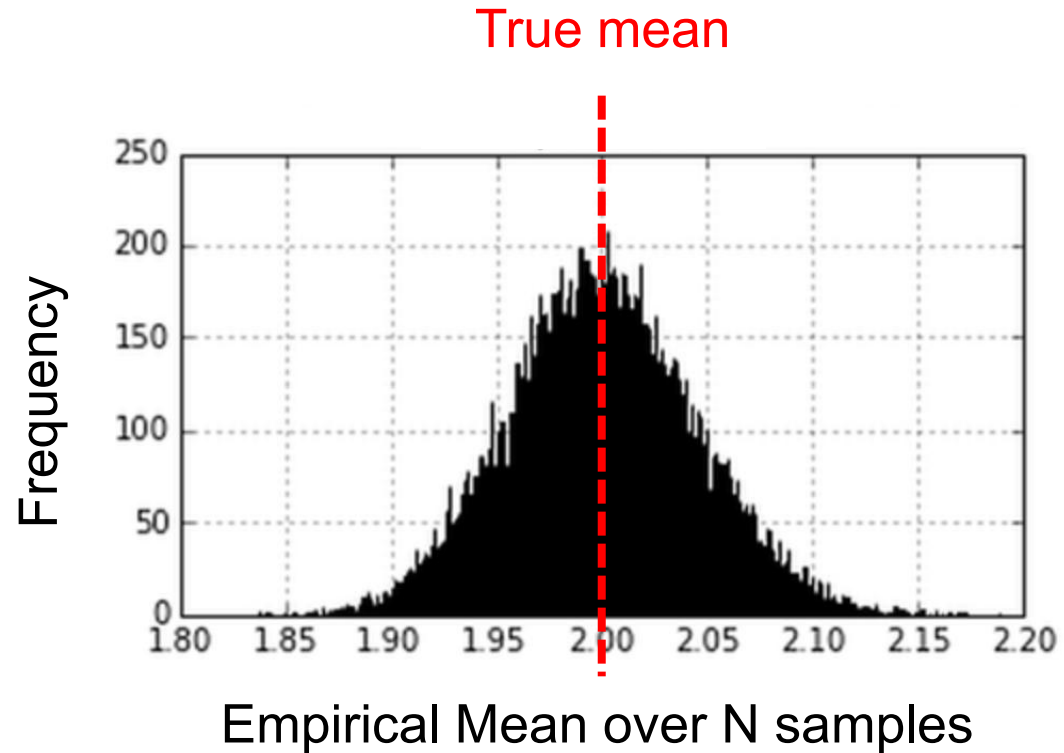
$$\left| \hat{R}(a) - R(a) \right| \leq ? \quad f(N)$$

some decreasing function of N

Empirical mean
of N i.i.d. samples

True mean

Quantifying the Estimation Error



Quantifying the Estimation Error

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

Key Question:

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

some decreasing function of N

Empirical mean
of N i.i.d. samples

True mean

Quantifying the Estimation Error

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

Key Question:

With probability at least $1 - \delta$,

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

some decreasing function of N

Empirical mean
of N i.i.d. samples

True mean

Quantifying the Error: Concentration Inequality

Theorem. Hoeffding's Inequality

Let X_1, \dots, X_N be independent σ -**sub-Gaussian** random variables with mean μ .
Then with probability at least $1 - \delta$,

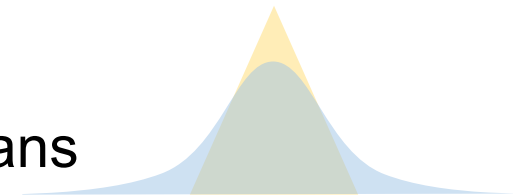
$$\left| \frac{1}{N} \sum_{i=1}^N X_i - \mu \right| \leq \sigma \sqrt{\frac{2 \log(2/\delta)}{N}} .$$

A random variable is called σ -sub-Gaussian if $\mathbb{E}[e^{\lambda(X - \mathbb{E}[X])}] \leq e^{\lambda^2 \sigma^2 / 2} \quad \forall \lambda \in \mathbb{R}$.

Fact 1. $\mathcal{N}(\mu, \sigma^2)$ is σ -sub-Gaussian.


Fact 2. A random variable $\in [a, b]$ is $(b - a)$ -sub-Gaussian.

Intuition: tail probability $\Pr\{|X - \mathbb{E}[X]| \geq z\}$ bounded by that of Gaussians




Quantifying the Estimation Error

With probability at least $1 - \delta$, $\left| \hat{R}(a) - R(a) \right| = O \left(\sqrt{\frac{\log(1/\delta)}{N}} \right)$



Omit constants

With high probability, $\left| \hat{R}(a) - R(a) \right| = \tilde{O} \left(\sqrt{\frac{1}{N}} \right)$



Omit constants and $\log(1/\delta)$ factors

Explore-then-Commit Regret Analysis

In the first T_0 rounds, sample each arm T_0/A times.

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

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

Regret Bound of Explore-then-Commit and ϵ -Greedy

Theorem. Regret Bound of Explore-then-Commit

Assume that $R(a) \in [-1,1]$ and w_t is 1-sub-Gaussian.

Then Explore-then-Exploit ensures with high probability,

$$\text{Regret} \lesssim T_0 + T \sqrt{\frac{A}{T_0}} \approx A^{1/3} T^{2/3} \text{ (choosing } T_0 = A^{1/3} T^{2/3} \text{)}$$

Theorem. Regret Bound of ϵ -Greedy

Assume that $R(a) \in [-1,1]$ and w_t is 1-sub-Gaussian.

Then ϵ -Greedy ensures with high probability,

$$\text{Regret} \lesssim \epsilon T + \sqrt{\frac{AT}{\epsilon}} \approx A^{1/3} T^{2/3} \text{ (choosing } \epsilon = \left(\frac{A}{T}\right)^{1/3} \text{)}$$

In practice, we prefer time-varying exploration $\epsilon_t \approx \left(\frac{A}{t}\right)^{1/3}$

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 (Parameter: λ)

In each round, sample a_t according to

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

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

Inverse Gap Weighting (Parameter: λ)

γ_t is a **normalization factor**
that makes $\sum_a \pi_t(a) = 1$

$$\pi_t(a) = \frac{1}{\gamma_t - \lambda \hat{R}_t(a)} = \frac{1}{\gamma'_t + \lambda \text{Gap}_t(a)}$$

where $\text{Gap}_t(a) = \max_b \hat{R}_t(b) - \hat{R}_t(a)$

Refined Exploration

- Boltzmann Exploration

- A quite commonly used exploration strategy (like ϵ -greedy)
- However, its theoretical guarantee is less clear and probably less desirable.

Cesa-Bianchi, Gentile, Lugosi, Neu. Boltzmann Exploration Done Right, 2017.

Bian and Jun. Maillard Sampling: Boltzmann Exploration Done Optimally. 2021.

- Inverse Gap Weighting

- Less well-known
- Allows to achieve a near-optimal regret bound \sqrt{AT} , improving the $A^{1/3}T^{2/3}$ by ϵ -greedy

Foster and Rakhlin. Beyond UCB: Optimal and Efficient Contextual Bandits with Regression Oracles. 2020.

Guarantee of Inverse Gap Weighting

Inverse Gap Weighting ensures with high probability,

$$\text{Regret} \lesssim \frac{A}{\lambda} + \lambda \log T \approx \sqrt{AT \log T} \text{ (choosing } \lambda = \sqrt{\frac{T}{A \log T}})$$

D. Foster and A. Rakhlin. Beyond UCB: Optimal and Efficient Contextual Bandits with Regression Oracles. 2020.

See supplementary materials for a formal proof.

$$\text{Time-varying version: } \lambda_t \approx \sqrt{\frac{t}{A}}$$

Summary: MAB Based on Mean Estimation

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

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

$$\textbf{EG} \quad \pi_t(a) = (1 - \epsilon) \mathbb{I}\{a = \operatorname{argmax} \hat{R}_t(\cdot)\} + \frac{\epsilon}{A} \quad A^{1/3} T^{2/3}$$

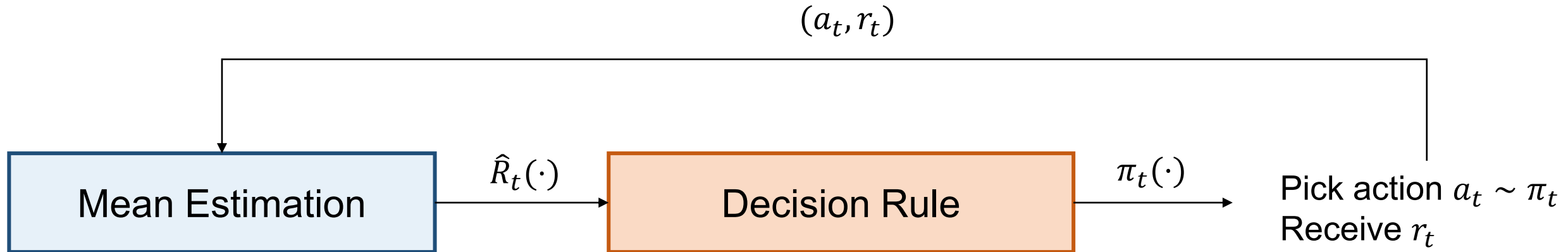
$$\textbf{BE} \quad \pi_t(a) \propto \exp(\lambda \hat{R}_t(a)) \quad \text{XXX}$$

$$\textbf{IGW} \quad \pi_t(a) = \frac{1}{\gamma_t - \lambda \hat{R}_t(a)} \quad \sqrt{AT}$$

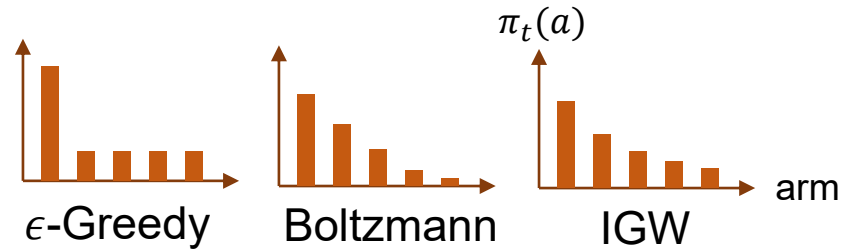
Sample an arm $a_t \sim \pi_t$ and receive the corresponding reward r_t .

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

Summary: MAB Based on Mean Estimation



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



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

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

$$\pi_t(a) = \frac{1}{\gamma_t - \lambda \hat{R}_t(a)}$$

Summary: MAB Based on Mean Estimation

- All 3 methods are based on the same **mean estimation**
 - ϵ -Greedy, Boltzmann exploration, Inverse gap weighting
- 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 in EG, λ in BE or IGW)

Some Experiments

$T = 10000$ rounds

$A = 2$ arms

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

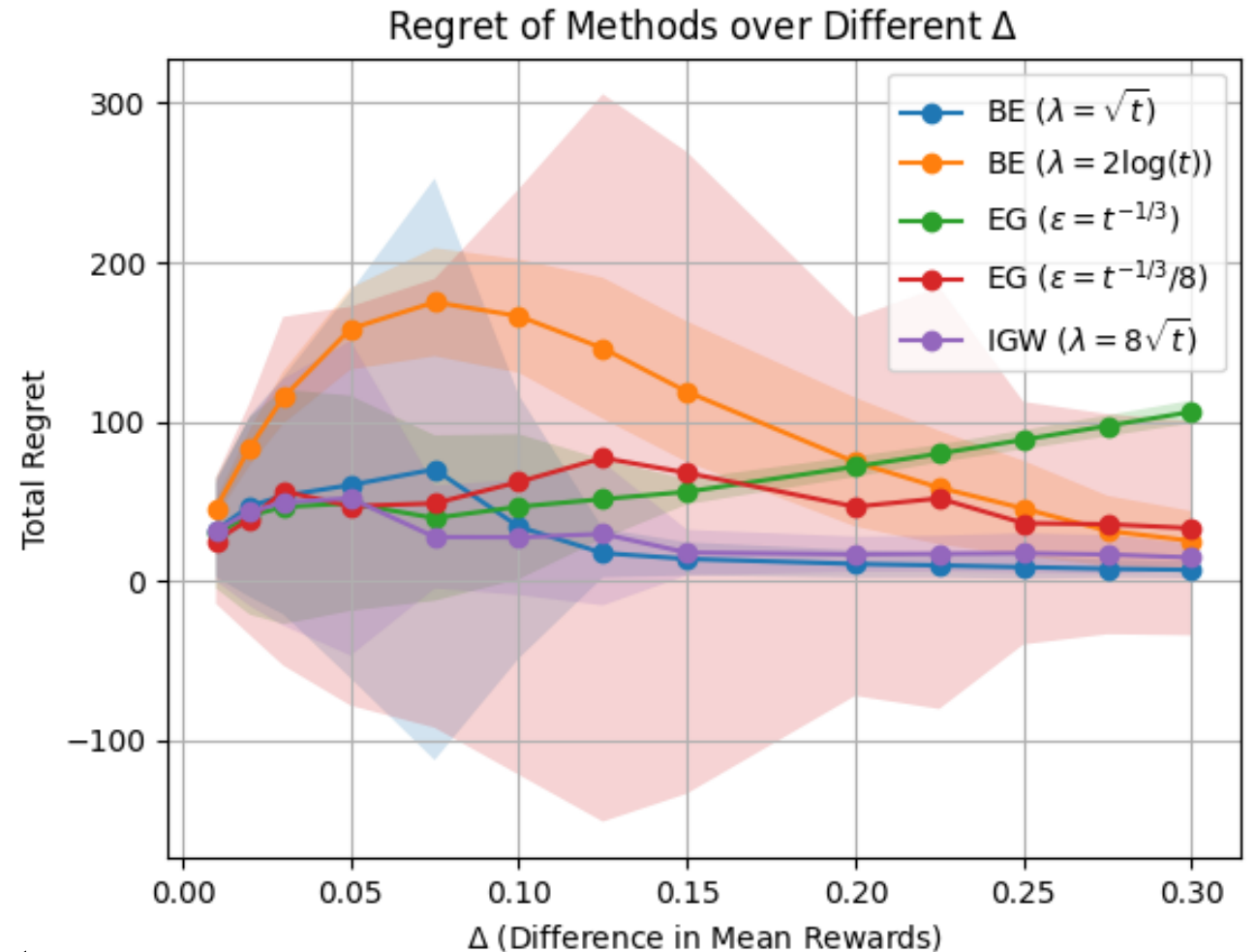
Bernoulli distribution

Time-dependent parameters

30 random seeds

Observations:

- Bound from theory could be loose
-- theory captures **worst-case** guarantee
- Most algorithms have its worst regret at some intermediate Δ value

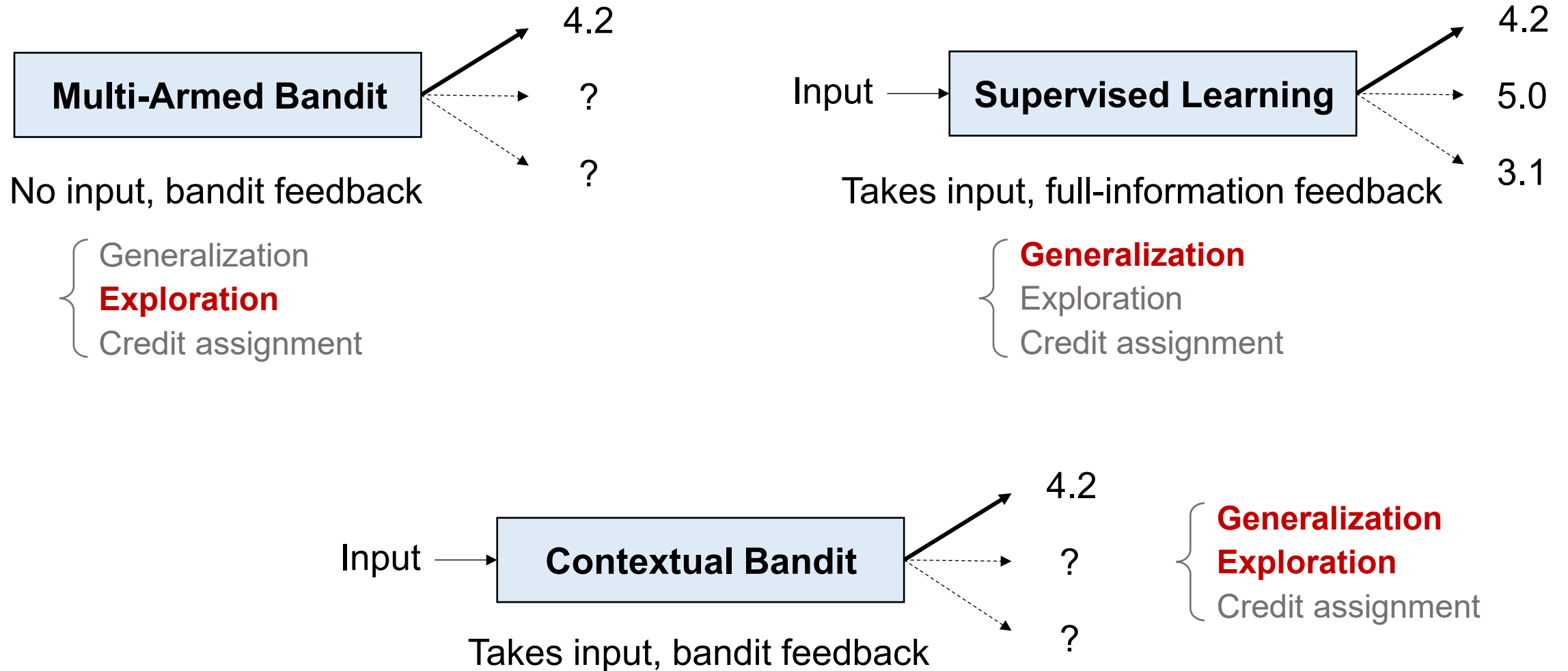


Small Δ is easy: don't need to distinguish the two arms
Large Δ is also easy: easy to distinguish the two arms

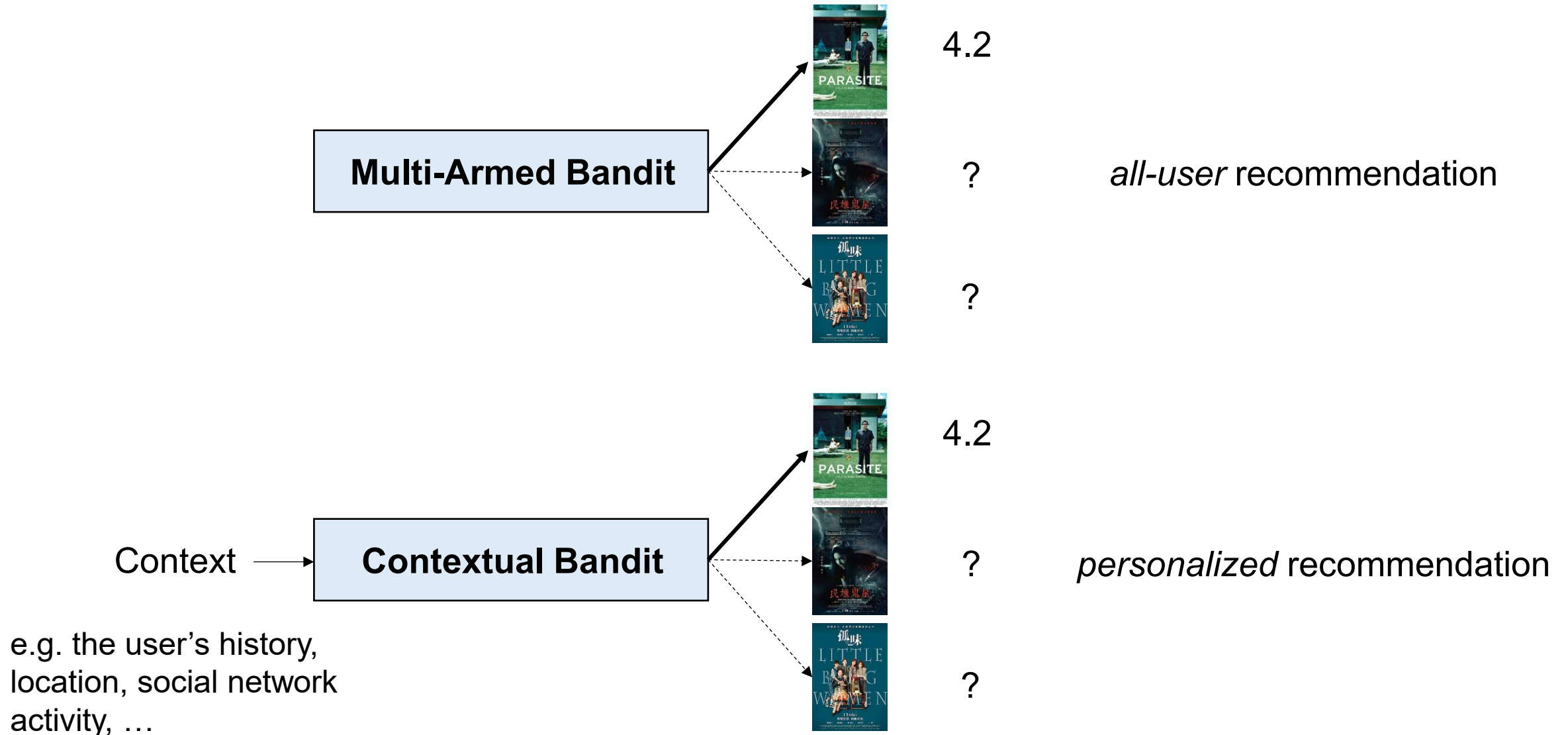
Contextual Bandits

Based on reward function estimation

Contextual Bandits Generalizes MAB and SL



Multi-Armed Bandits vs. Contextual Bandits



Contextual Bandits

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

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

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

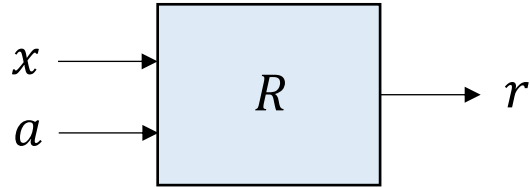
Learner observes $r_t = R(x_t, a_t) + w_t$

Discussion

- Contextual bandits is a minimal simultaneous generalization of supervised learning (SL) and multi-armed bandits (MAB)
- We learned a lot about SL in machine learning courses
- We just learned some simple MAB algorithms
 - 3 strategies based on mean estimation
- **Question:** Can you design a contextual bandits algorithm based on the techniques you know for SL and MAB?

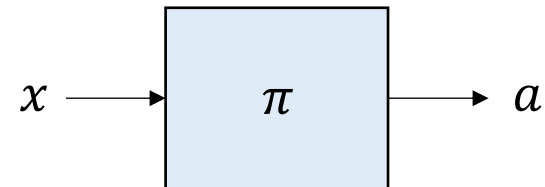
Two ways to leverage SL techniques in CB

x : context, a : action, r : reward



Learn a mapping from
(context, action) to reward

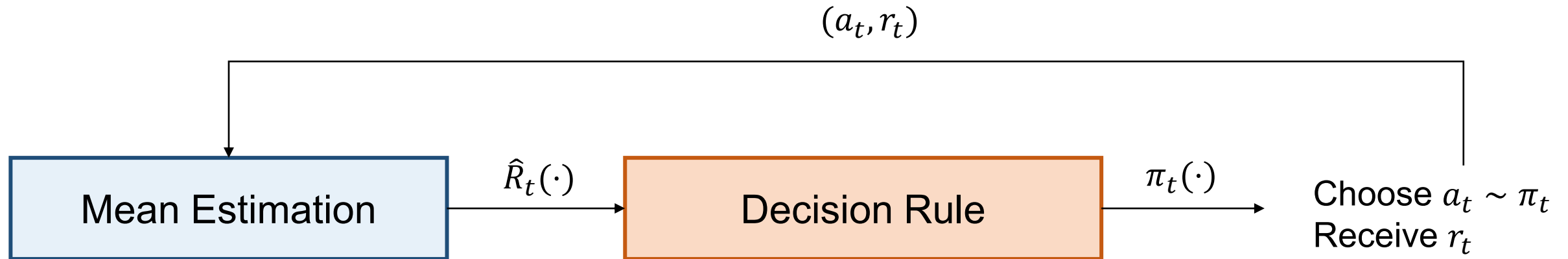
CB with **regression oracle**
Value-based approach
(discussed next)



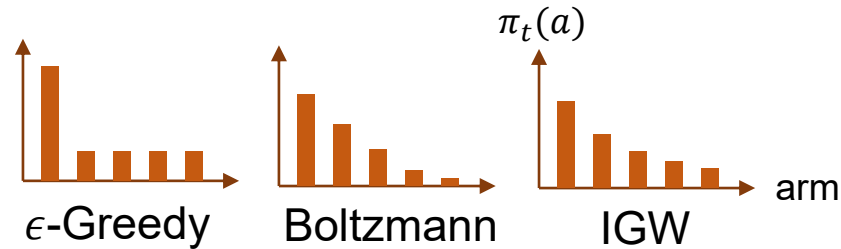
Learn a mapping from
context to action (or action distribution)

CB with **classification oracle**
Policy-based approach
(slightly later in the course)

Recall: MAB Based on Mean Estimation



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

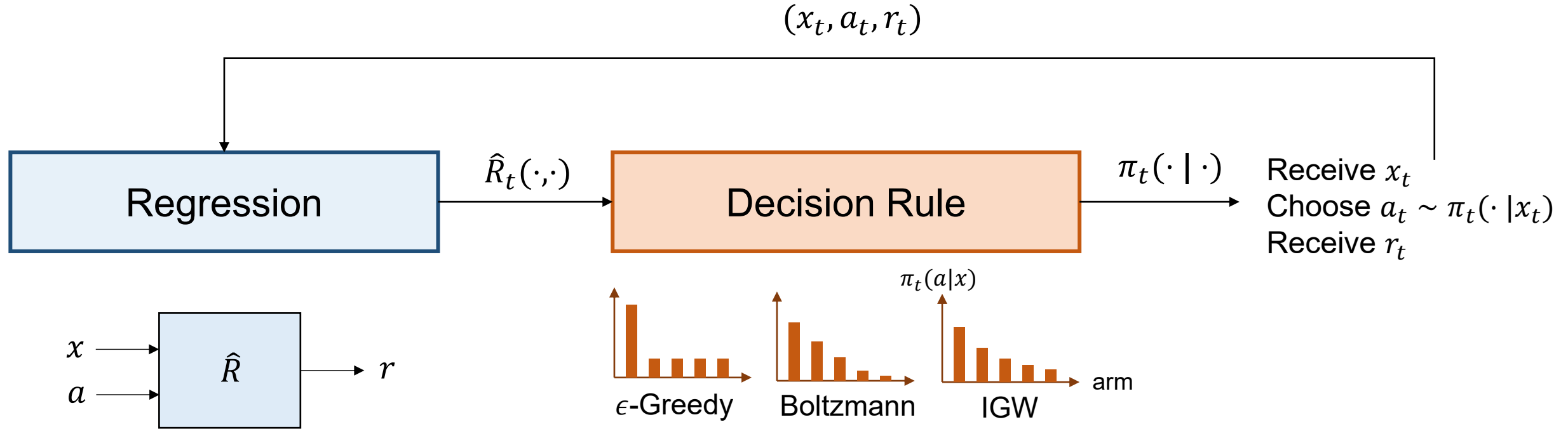


$$\pi_t(a) = (1 - \epsilon) \mathbb{I}\{a = \operatorname{argmax} \hat{R}_t(\cdot)\} + \frac{\epsilon}{A}$$

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

$$\pi_t(a) = \frac{1}{\gamma_t - \lambda \hat{R}_t(a)}$$

CB Based on Reward Function Estimation (Regression)



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

$$\pi_t(a|x) = (1 - \epsilon)\mathbb{I}\{a = \operatorname{argmax} \hat{R}_t(x, \cdot)\} + \frac{\epsilon}{A}$$

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

$$\pi_t(a|x) = \frac{1}{\gamma_t - \lambda \hat{R}_t(x, a)}$$

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 $\hat{R}_t(x_t, \cdot)$

$$\mathbf{EG} \quad \pi_t(a|x_t) = (1 - \epsilon)\mathbb{I}\{a = \operatorname{argmax} \hat{R}_t(x_t, \cdot)\} + \frac{\epsilon}{A}$$

$$\mathbf{BE} \quad \pi_t(a|x_t) \propto \exp(\lambda \hat{R}_t(x_t, a))$$

$$\mathbf{IGW} \quad \pi_t(a|x_t) = \frac{1}{\gamma_t - \lambda \hat{R}_t(x_t, a)}$$

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) .

Regret in 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 $r_t = R(x_t, a_t) + w_t$

$$\begin{aligned} \text{Regret} &= \sum_{t=1}^T R(x_t, \pi^*(x_t)) - \sum_{t=1}^T R(x_t, a_t) & \text{Benchmark policy: } \pi^*(x) &= \operatorname{argmax}_{a \in \mathcal{A}} R(x, a) \\ &= \sum_{t=1}^T \max_{a \in \mathcal{A}} R(x_t, a) - \sum_{t=1}^T R(x_t, a_t) \end{aligned}$$

Regret in Contextual Bandits

Regret Bound of ϵ -Greedy

ϵ -Greedy ensures

$$\text{Regret} \lesssim \epsilon T + \sqrt{\frac{AT \cdot \text{Err}}{\epsilon}}$$

Regression error

$$\text{Err} = \sum_{t=1}^T \left(\hat{R}_t(x_t, a_t) - R(x_t, a_t) \right)^2$$

Regret Bound of Inverse Gap Weighting

IGW ensures

$$\text{Regret} \lesssim \frac{AT}{\lambda} + \lambda \cdot \text{Err}$$

Will be proven in HW1

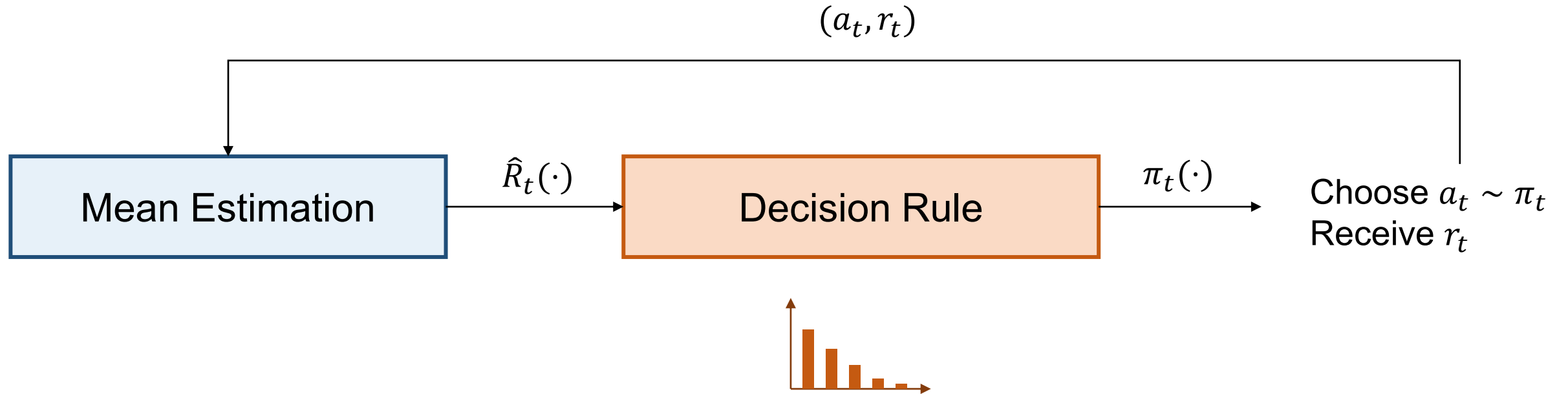
Summary

- 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 lifted as a CB algorithm with “**reward function estimation**” by leveraging a regression oracle.
 - This gives a general framework for value-based CB

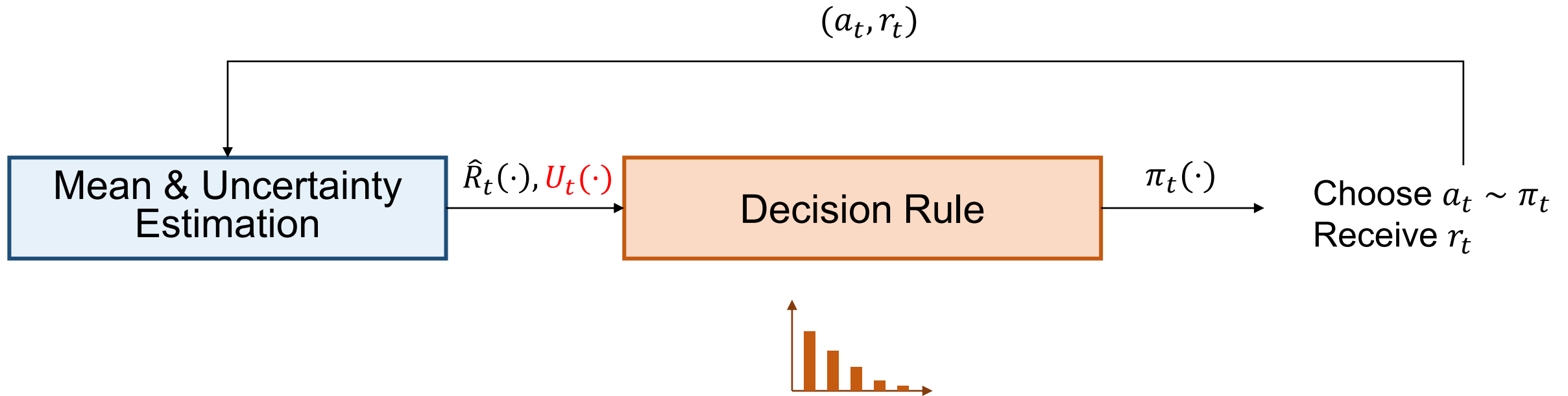
Multi-Armed Bandits

Based on mean and uncertainty estimation

Recall: MAB Based on Mean Estimation



MAB Based on Mean and Uncertainty Estimation



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

$$|\hat{R}_t(a) - R(a)| \leq \sqrt{\frac{2\log(2/\delta)}{N_t(a)}} \triangleq U_t(a)$$

This inequality is used in the **math analysis** of ϵ -Greedy and IGW, but not in their **algorithm**.

Useful Idea: “Optimism in the Face of Uncertainty”

In words:

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

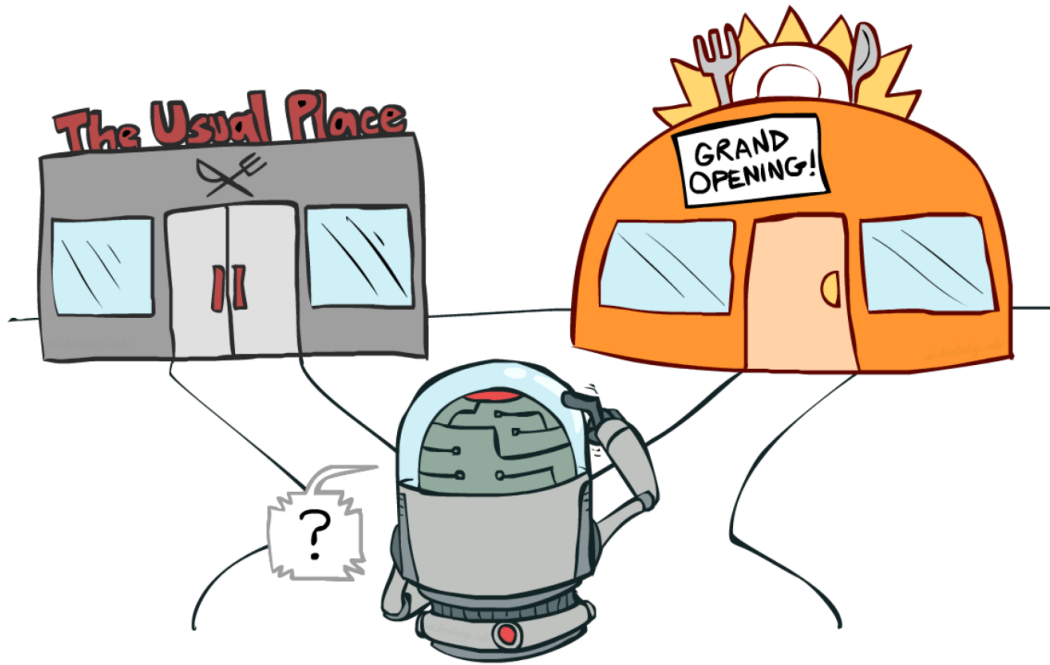


Image source: UC Berkeley CS188

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 $R(a)$?

$$|R(a) - \hat{R}_t(a)| \leq \sqrt{\frac{2 \log(2/\delta)}{N_t(a)}} \quad \text{w.p.} \geq 1 - \delta$$

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

$$\hat{R}_t(a) + \sqrt{\frac{2 \log(2/\delta)}{N_t(a)}}$$

Upper Confidence Bound (UCB)

UCB (Parameter: δ) Usually decreases over time as $\delta_t \sim 1/t$ (drives continual exploration)

In round t , draw

Exploration Bonus

$$a_t = \operatorname{argmax}_a \hat{R}_t(a) + \sqrt{\frac{2 \log(2/\delta)}{N_t(a)}}$$

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 $t - 1$.

Regret Bound of UCB

Theorem. Regret Bound of UCB

UCB ensures with high probability,

$$\text{Regret} \lesssim \sqrt{AT} .$$

Visualizing UCB

True mean: [0.2, 0.4, 0.6, 0.7] [animation](#)

Summary: Algorithms We Learned So Far

	Regret Bound	Approach
Explore-then-Exploit ϵ -Greedy Boltzmann Exploration Inverse Gap Weighting	$A^{1/3} T^{2/3}$ $A^{1/3} T^{2/3}$ X \sqrt{AT}	Mean estimation + decision rule
Upper Confidence Bound Thompson Sampling	\sqrt{AT}	Mean and uncertain estimation + decision rule

Thompson Sampling

William Thompson. **On the likelihood that one unknown probability exceeds another in view of the evidence of two samples**, 1933.

Thompson Sampling (Parameter: c)

In round t , draw

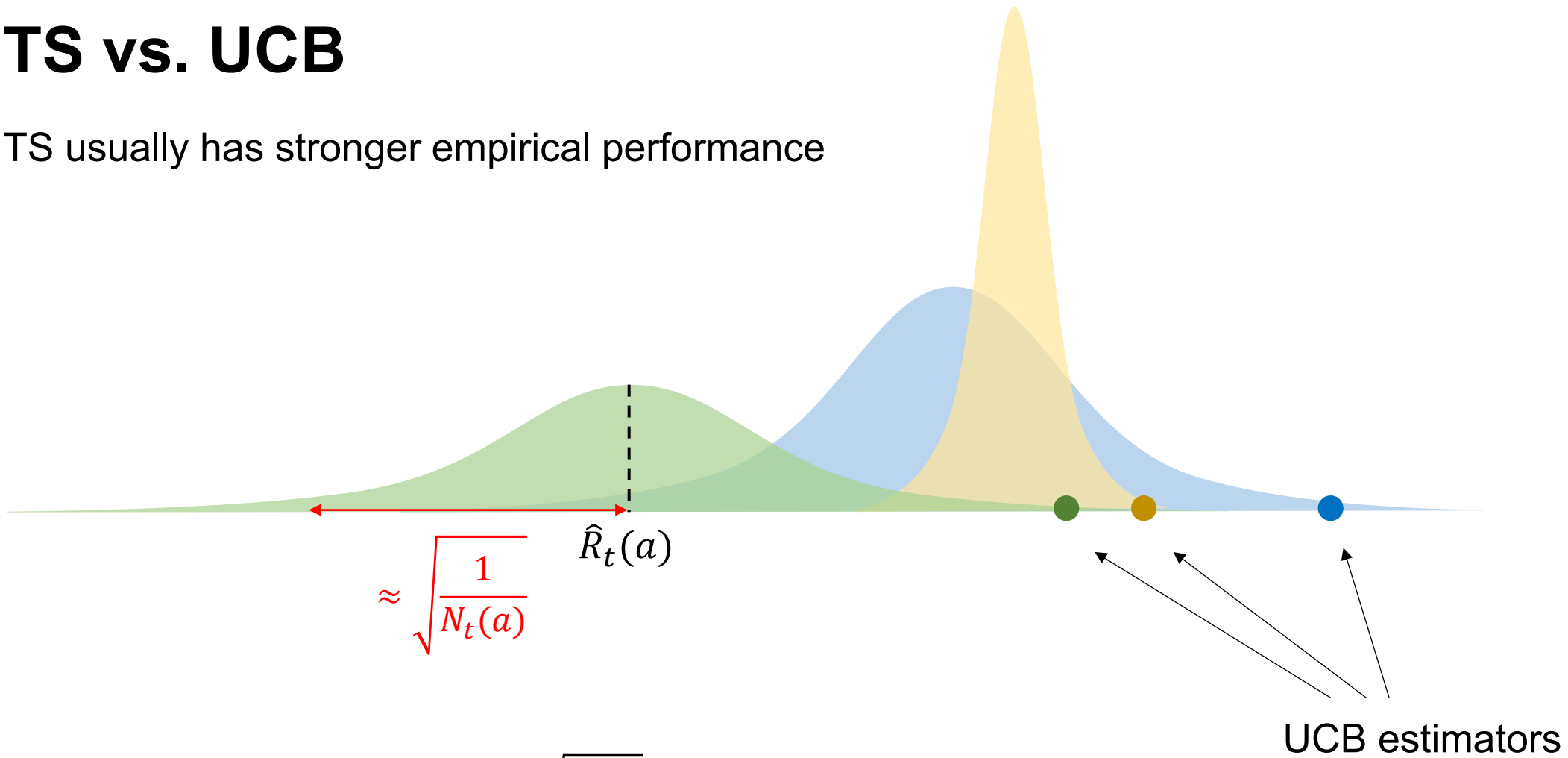
$$a_t = \operatorname{argmax}_a \hat{R}_t(a) + \sqrt{\frac{c}{N_t(a)}} n_t(a) \quad \text{with } n_t(a) \sim \mathcal{N}(0,1)$$

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 $t - 1$.

There are other/better ways to design the noise for specific (e.g., Bernoulli) reward

TS vs. UCB

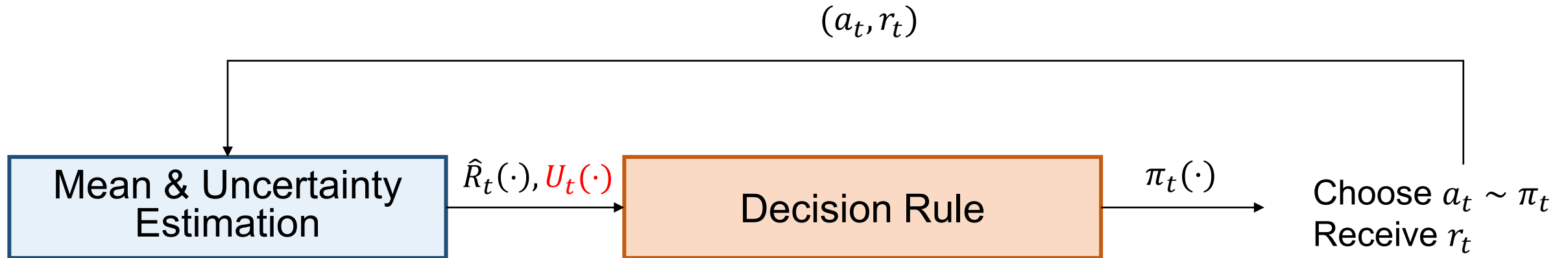
TS usually has stronger empirical performance



$$\text{UCB: } a_t \approx \operatorname{argmax}_a \hat{R}_t(a) + c \sqrt{\frac{1}{N_t(a)}}$$

$$\text{Thompson Sampling: } a_t \approx \operatorname{argmax}_a \hat{R}_t(a) + c \sqrt{\frac{1}{N_t(a)}} n_t(a) \quad \text{with } n_t(a) \sim \mathcal{N}(0,1)$$

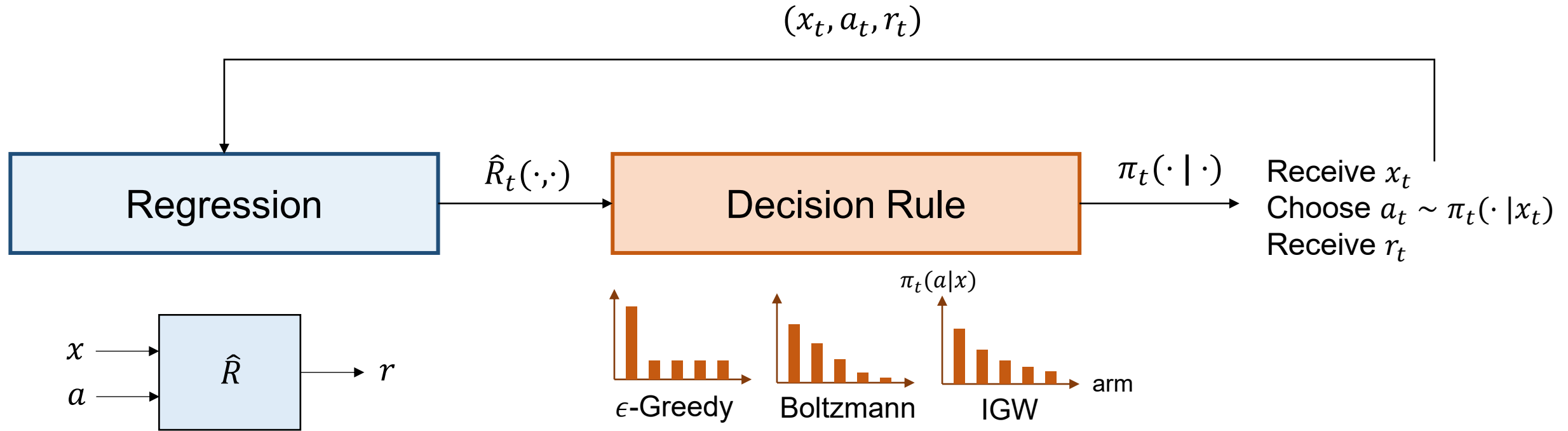
Extension to Contextual Bandits?



- Linear UCB, Linear TS (using linear regression)
- In UCB or TS or Linear UCB or Linear TS, we really did not perform “**uncertainty estimation**” – the uncertainty measure is directly derived from Hoeffding’s bound
- When general function approximation is used, it’s no longer easy to “derive” uncertainty measure, so it really needs to be “estimated”.
- Let’s talk about this more in the future.

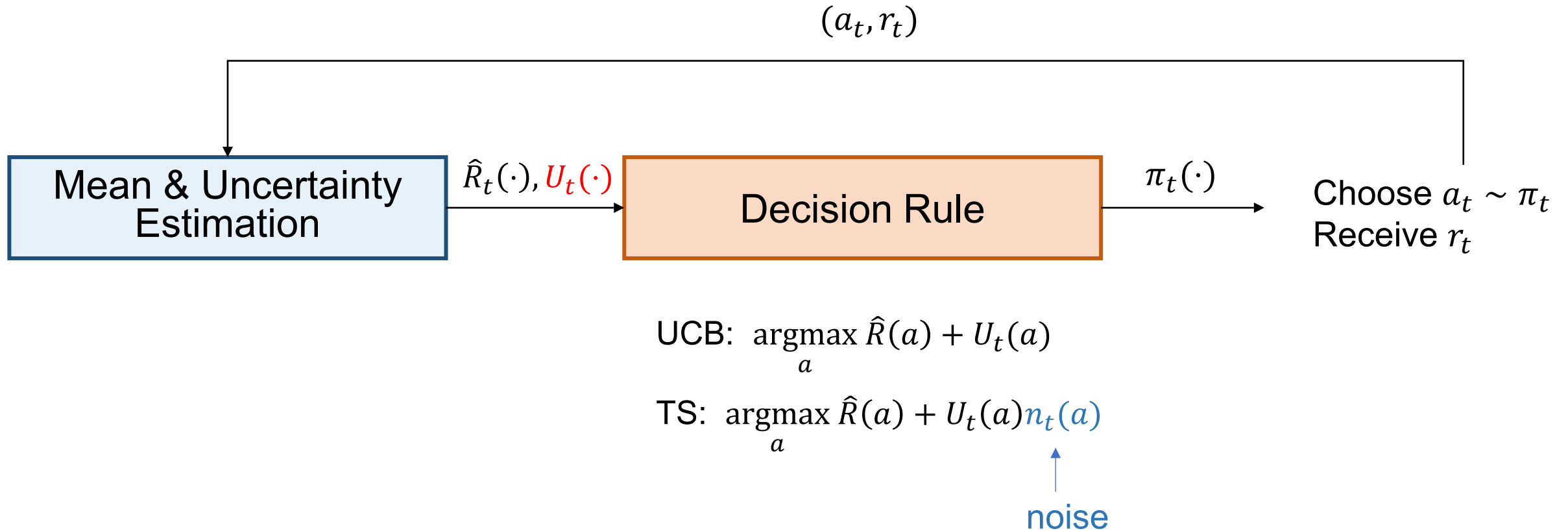
Summary

CB Based on Reward Function Estimation



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

MAB Based on Mean and Uncertainty Estimation



Uncertainty estimation for CB presents additional challenges – discussed in the future.