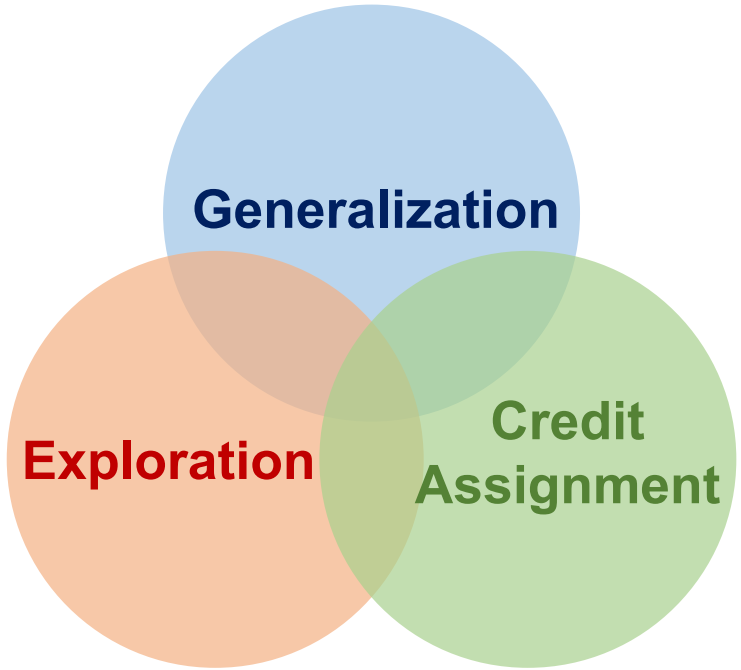
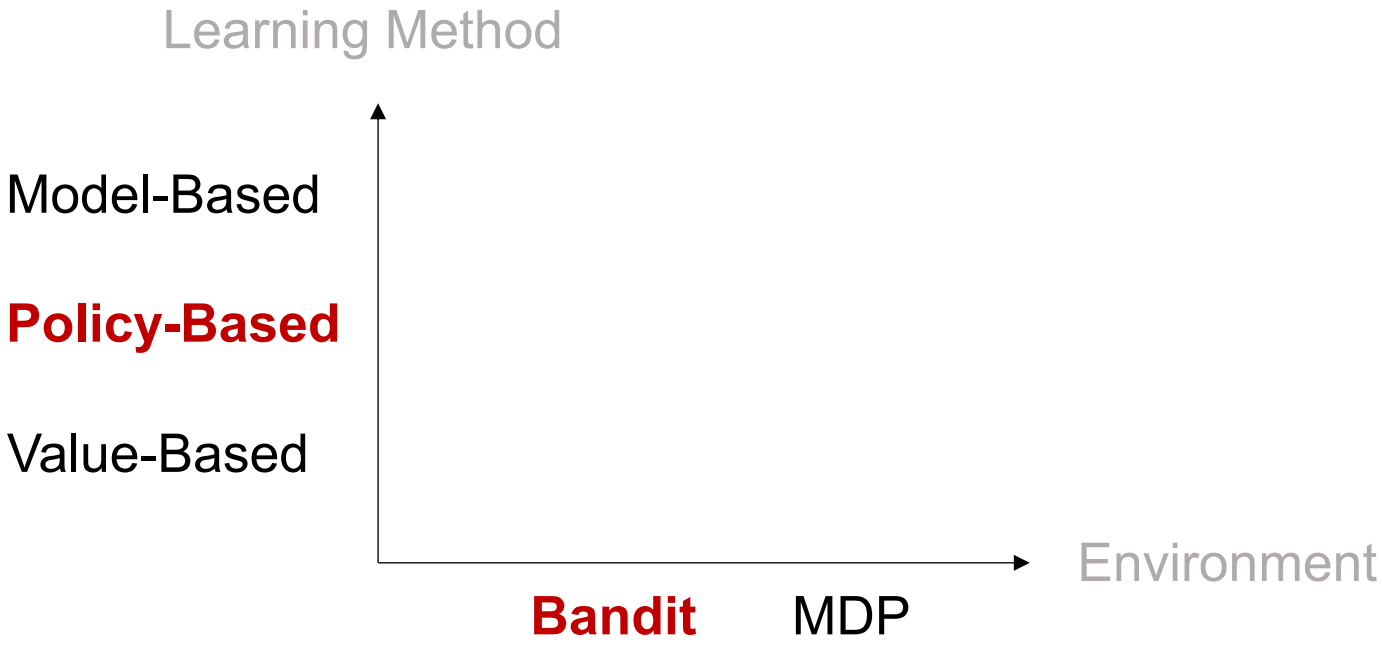
Bandits 2

Chen-Yu Wei

Roadmap


Generalization
Exploration
Credit
Assignment

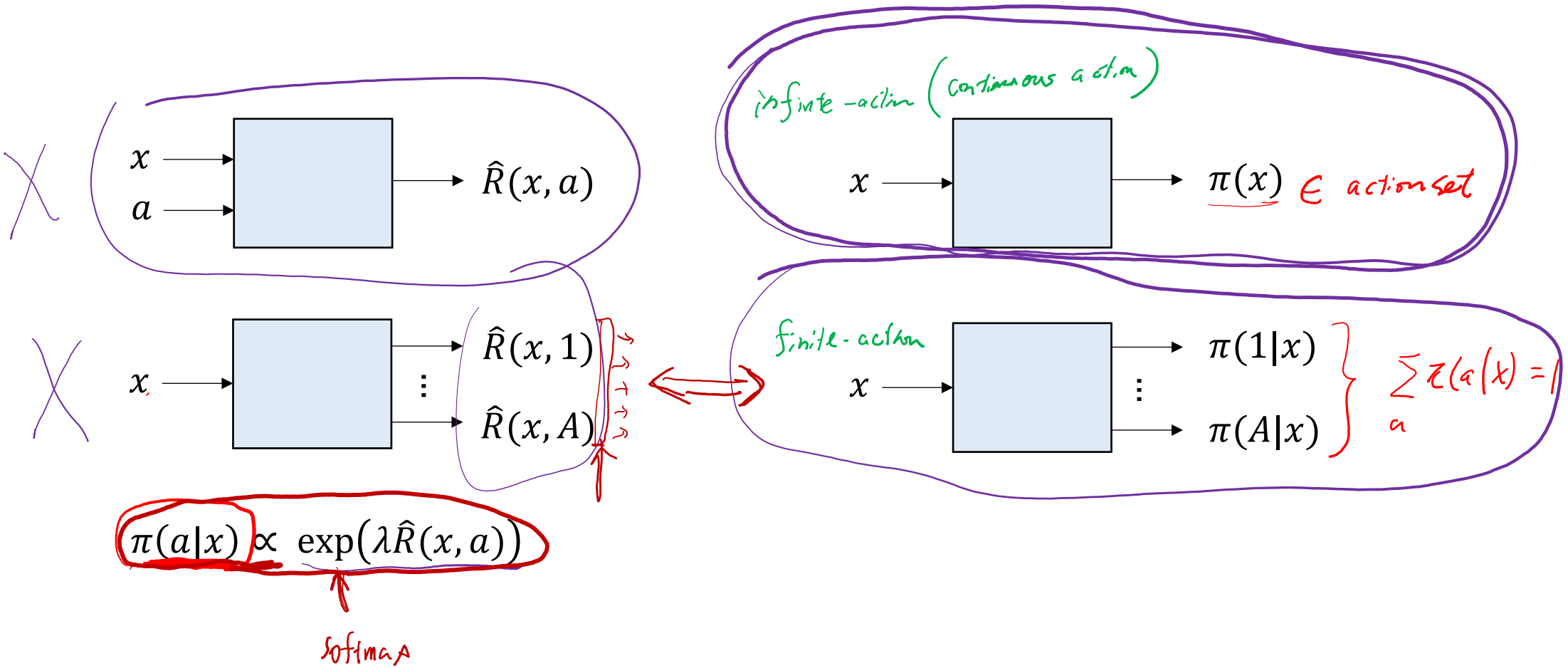

Learning Method
Model-Based
Policy-Based
Value-Based
Environment
Bandit
MDP

Policy-Based Bandits

- Key challenges: **Exploration** and **Generalization (if there are contexts)**
- Algorithms we will discuss:
  - KL-regularized policy updates (PPO)
  - Policy gradient (REINFORCE)

Policy-Based Bandits

x: context, a: action



Value-based approach

Policy-based approach

Policy-Based Bandits

Why policy-based bandit algorithms?

- Actually, in finite-action contextual bandit problems, value- and policy-based approaches are almost equivalent.
- But we have to use policy-based approaches to handle **continuous action space.**
- They are also different in MDPs. (later in the course)

The Full-Information MAB

Given: set of actions $\mathcal{A} = \{1, \dots, A\}$

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

The learner chooses an action a_t

Environment reveals the reward $r_t(a) = R(a) + w_t(a)$ **of all actions**

Policy-based algorithm: Maintain a distribution $\pi_t(a)$ and update it with feedback

Sample $a_t \sim \pi_t$

How should we update from π_t to π_{t+1} using $r_t(1), \dots, r_t(A)$?

$\bar{r}_t \sim r_t$

$\pi_{t+1} \leftarrow \pi_t + \bar{r}_t$

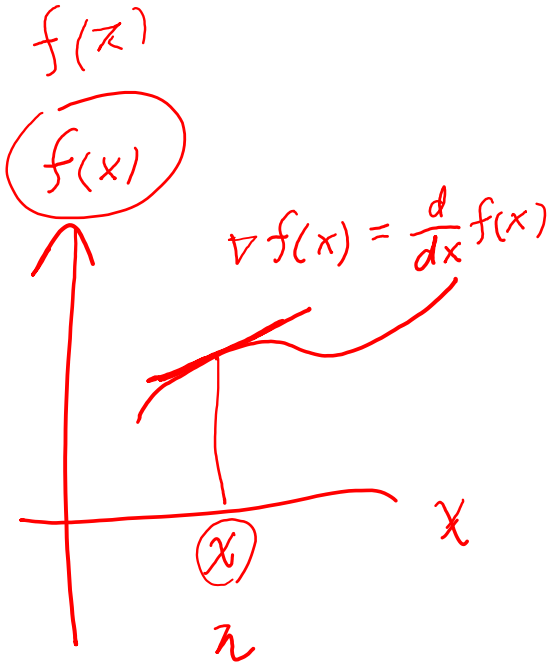
$\pi_{t+1}(a) \neq \pi_t(a) + \bar{r}_t(a)$

Algorithm for the Full-Information MAB

$$f(\pi) = \sum_{a=1}^{A} \pi(a) R(a)$$ ← We want to find a π that **maximizes** this value

But we don't know $R(a)$

But we get noisy samples of $R(a)$, i.e., $r_t(a)$



Gradient Ascent

$$f(\pi) = \sum_{a=1}^{A} \pi(a) R(a) \qquad \Rightarrow \quad \nabla_\pi f(\pi) = R$$

Gradient Ascent

For $t = 1, 2 \ldots$

$$\pi_{t+1} \leftarrow \pi_t + \eta R$$

$$\pi_{t+1} \leftarrow \Pi(\pi_{t+1})$$

Stochastic Gradient Ascent

For $t = 1, 2 \ldots$

$$\pi_{t+1} \leftarrow \pi_t + \eta r_t$$

$$\pi_{t+1} \leftarrow \Pi(\pi_{t+1})$$

Exponential Weight Update

For $t = 1, 2 \ldots$

$$\pi_{t+1}(a) \propto \pi_t(a)\, e^{\eta r_t(a)}$$

or $$\pi_{t+1}(a) = \frac{\pi_t(a)\, e^{\eta r_t(a)}}{\sum_{b \in \mathcal{A}} \pi_t(b)\, e^{\eta r_t(b)}}$$

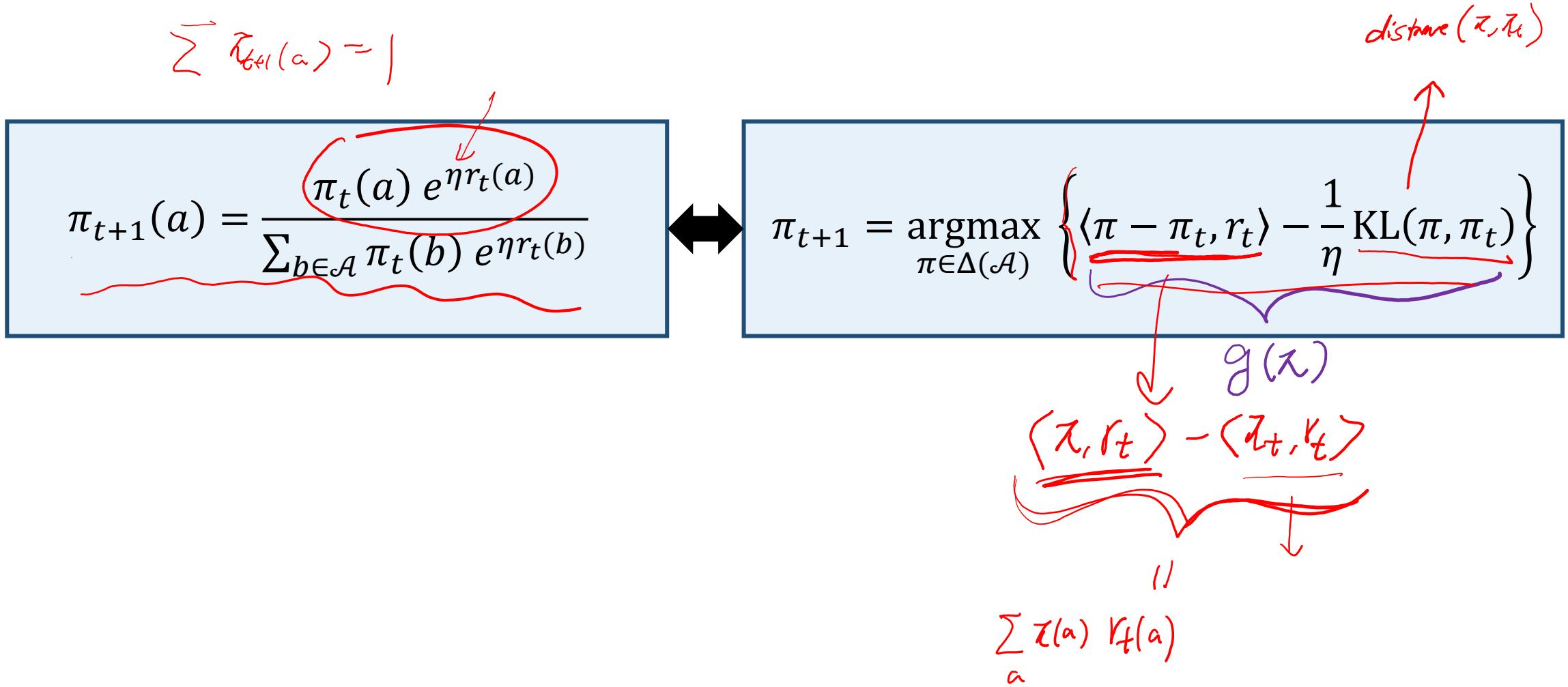
Better for bandit problems (because we never get $\pi_t(a) = 0$)

Exponential Weight Update = KL-Regularized Policy Updates

$$\pi_{t+1}(a) = \frac{\pi_t(a)\, e^{\eta r_t(a)}}{\sum_{b \in \mathcal{A}} \pi_t(b)\, e^{\eta r_t(b)}}$$

$$\Leftrightarrow$$

$$\pi_{t+1} = \underset{\pi \in \Delta(\mathcal{A})}{\operatorname{argmax}} \left\{ \langle \pi - \pi_t, r_t \rangle - \frac{1}{\eta} \mathrm{KL}(\pi, \pi_t) \right\}$$



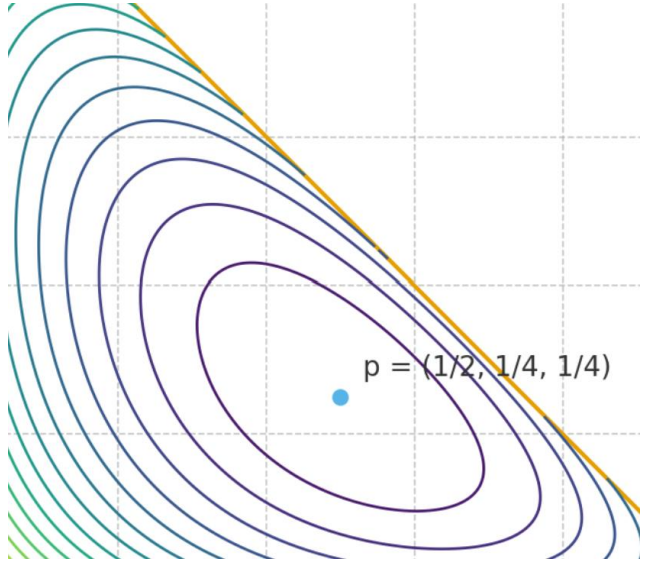
KL Divergence – A Distance Measure for Distributions

$$\mathrm{KL}(\pi, \pi') = \sum_a \pi(a) \log \frac{\pi(a)}{\pi'(a)}$$

$\mathrm{KL}(\pi, \pi') \geq 0$

$\mathrm{KL}(\pi, \pi') = 0$ if and only if $\pi = \pi'$

$\mathrm{KL}(\pi, \pi') \neq \mathrm{KL}(\pi', \pi)$

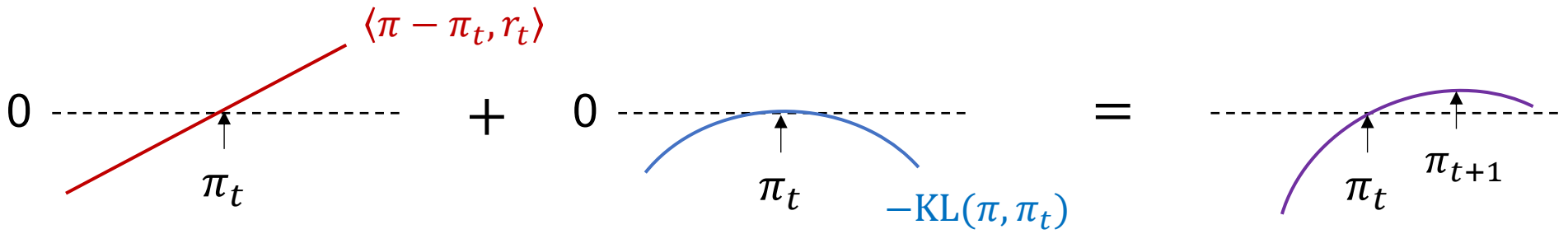


Regularized Policy Updates

$$\pi_{t+1} = \operatorname*{argmax}_{\pi \in \Delta(\mathcal{A})} \left\{ \langle \pi - \pi_t, r_t \rangle - \frac{1}{\eta} \mathrm{KL}(\pi, \pi_t) \right\}$$

$$= \operatorname*{argmax}_{\pi \in \Delta(\mathcal{A})} \left\{ \underbrace{\sum_a \big(\pi(a) - \pi_t(a)\big) r_t(a)} - \frac{1}{\eta} \mathrm{KL}(\pi, \pi_t) \right\}$$

The Improvement of π over π_t on r_t



Multi-Armed Bandits

Multi-Armed Bandits

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

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

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

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

Recall: Exponential Weight Updates

$$\pi_{t+1} = \underset{\pi \in \Delta(\mathcal{A})}{\operatorname{argmax}} \left\{ \langle \pi - \pi_t, r_t \rangle - \frac{1}{\eta} \mathrm{KL}(\pi, \pi_t) \right\} \quad \Leftrightarrow \quad \pi_{t+1}(a) = \frac{\pi_t(a)\, e^{\eta r_t(a)}}{\sum_{b \in \mathcal{A}} \pi_t(b)\, e^{\eta r_t(b)}}$$

Exponential Weight Updates for Bandits?

$$\pi_{t+1} = \underset{\pi \in \Delta(\mathcal{A})}{\operatorname{argmax}} \left\{ \langle \pi - \pi_t, r_t \rangle - \frac{1}{\eta} \mathrm{KL}(\pi, \pi_t) \right\} \quad \Leftrightarrow \quad \pi_{t+1}(a) = \frac{\pi_t(a)\, e^{\eta r_t(a)}}{\sum_{b \in \mathcal{A}} \pi_t(b)\, e^{\eta r_t(b)}}$$

No longer observable

Only update the arm that we choose?

Exponential Weight Updates for Bandits?

$$\pi_{t+1} = \underset{\pi \in \Delta(\mathcal{A})}{\operatorname{argmax}} \left\{ \langle \pi - \pi_t, \hat{r}_t \rangle - \frac{1}{\eta} \mathrm{KL}(\pi, \pi_t) \right\} \quad \Leftrightarrow \quad \pi_{t+1}(a) = \frac{\pi_t(a)\, e^{\eta \hat{r}_t(a)}}{\sum_{b \in \mathcal{A}} \pi_t(b)\, e^{\eta \hat{r}_t(b)}}$$

- $\hat{r}_t(a)$ is an **"estimator"** for $r_t(a)$
- But we can only observe the reward of one arm
- And let's set the restriction that we can only construct $\hat{r}_t$ from ~~$r_t(a_t)$~~ real $(r_t(1), r_t(2), \ldots, r_t(A))$

What's the problem of setting $\hat{r}_t = (0, 0, \ldots, r_t(a_t), \ldots, 0)$?

$$\mathbb{E}[\hat{r}_t] = (\pi_t(1) r_t(1),\ \pi_t(2) r_t(2),\ \ldots,\ \pi_t(A) r_t(A))$$

Unbiased Reward / Gradient Estimator

Weight a sample by **the inverse of the probability we observe it**

$$\hat{r}_t(a) = \frac{r_t(a)}{\pi_t(a)} \mathbb{I}\{a_t = a\} = \begin{cases} \frac{r_t(a)}{\pi_t(a)} & \text{if } a_t = a \\ 0 & \text{otherwise} \end{cases}$$

(Fixed a)

$$\begin{aligned} \mathbb{E}[\hat{r}_t(a)] &= \Pr\{a_t = a\} \frac{r_t(a)}{\pi_t(a)} + \Pr\{a_t \neq a\} 0 \\ &= \pi_t(a) \frac{r_t(a)}{\pi_t(a)} \\ &= r_t(a) \end{aligned}$$

$$\left(0, 0, \cdots, \frac{r_t(a_t)}{\pi_t(a_t)}, \cdots, 0\right)$$

Importance Weighting

$$\frac{1}{\pi_t(a)}$$

Directly Applying Exponential Weights

$\pi_1(a) = 1/A$ for all a

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

Sample $a_t \sim \pi_t$, and observe $r_t(a_t)$

Define for all a:

$$\hat{r}_t(a) = \frac{r_t(a)}{\pi_t(a)} \mathbb{I}\{a_t = a\}$$

Assume $r_t(a) \geq 0$

$\Rightarrow$ For arms we choose : $\hat{r}_t(a) \geq 0$

we don't choose $\hat{r}_t(a) = 0$

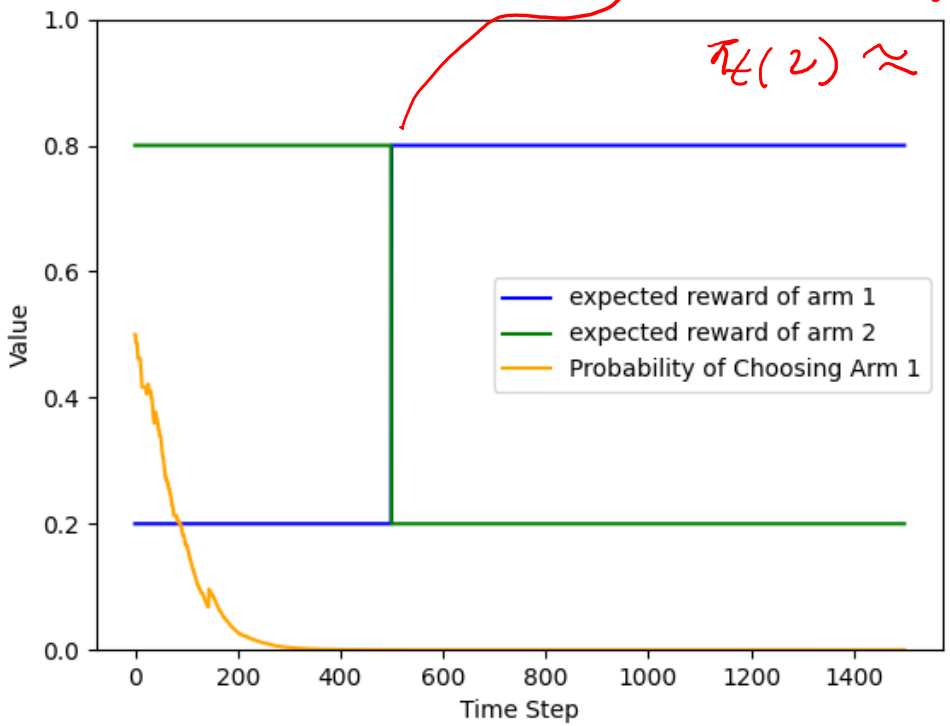
$\Rightarrow$ increase the prob of the arm that we just chose

Update policy:

$$\pi_{t+1}(a) = \frac{\pi_t(a) \exp(\eta \hat{r}_t(a))}{\sum_{a' \in \mathcal{A}} \pi_t(a') \exp(\eta \hat{r}_t(a'))}$$

Simple Experiment

- $A = 2,\ T = 1500,\ \eta = 1/\sqrt{T}$
- For $t \leq 500,\ r_t = [\text{Bernoulli}(0.2), \text{Bernoulli}(0.8)]$
- For $500 < t \leq 1500,\ r_t = [\text{Bernoulli}(0.8), \text{Bernoulli}(0.2)]$
- code



$\pi_t(1)$ very small

$\pi_t(2) \approx 1$

$$\hat{r}_t(1) = \frac{\gamma_t(1)}{\pi_t(1)}$$

Solution 1: Adding Extra Exploration

- **Idea:** use at least ϵ probability to explore uniformly

- Instead of sampling a_t according to π_t, use

$$\pi'_t(a) = (1-\epsilon)\pi_t(a) + \frac{\epsilon}{A}$$

$\Rightarrow$ w.p. ϵ sample $a_t \sim$ uniform; w.p. $1-\epsilon$ sample $a_t \sim \pi_t$

Then the unbiased reward estimator becomes

$$\hat{r}_t(a) = \frac{r_t(a)}{\pi'_t(a)}\mathbb{I}\{a_t = a\} = \frac{r_t(a)}{(1-\epsilon)\pi_t(a) + \frac{\epsilon}{A}}\mathbb{I}\{a_t = a\}$$

Applying Solution 1

$\pi_1(a) = 1/A$ for all a

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

Sample a_t from $\pi'_t = (1-\epsilon)\pi_t + \epsilon \,\text{uniform}(\mathcal{A})$, and observe $r_t(a_t)$

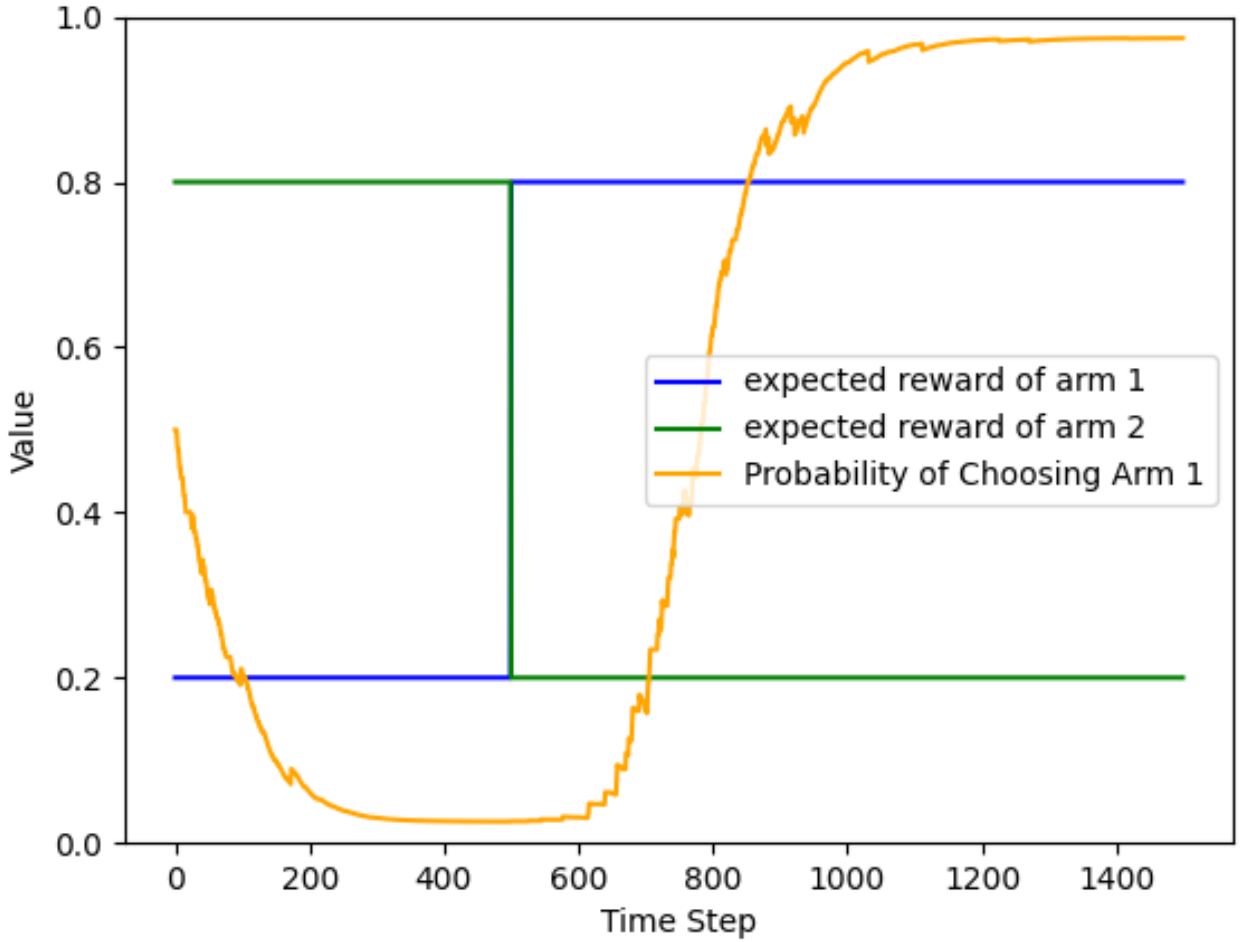
Define for all a:

$$\hat{r}_t(a) = \frac{r_t(a)}{\pi'_t(a)} \mathbb{I}\{a_t = a\}$$

Update policy:

$$\pi_{t+1}(a) = \frac{\pi_t(a) \exp(\eta \hat{r}_t(a))}{\sum_{a' \in \mathcal{A}} \pi_t(a') \exp(\eta \hat{r}_t(a'))}$$

Solution 1: Adding Extra Exploration



Solution 2: Reward Estimator with a Baseline

- Still sample a_t from π_t, but construct the reward estimator as

$$\hat{r}_t(a) = \frac{r_t(a) - b}{\pi_t(a)} \mathbb{I}\{a_t = a\} + b$$

- Why this resolves the issue?

$0 \le \pi_t(a) \le 1$

$$\mathbb{E}\left(\hat{r}_t(a)\right) = \pi_t(a)\left[\frac{r_t(a) - b}{\pi_t(a)} + b\right] + \left(1 - \pi_t(a)\right) \cdot b = r_t(a) - b + b\pi_t(a) + b\left(1 \cdot \pi_t(a)\right)$$

$$= r_t(a)$$

Applying Solution 2

$\pi_1(a) = 1/A$ for all a

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

Sample a_t from π_t, and observe $r_t(a_t)$

Define for all a:

$$\hat{r}_t(a) = \frac{r_t(a) - b}{\pi_t(a)} \mathbb{I}\{a_t = a\} + b \quad \text{or equivalently} \quad \hat{r}_t(a) = \frac{r_t(a) - b}{\pi_t(a)} \mathbb{I}\{a_t = a\}$$

baseline

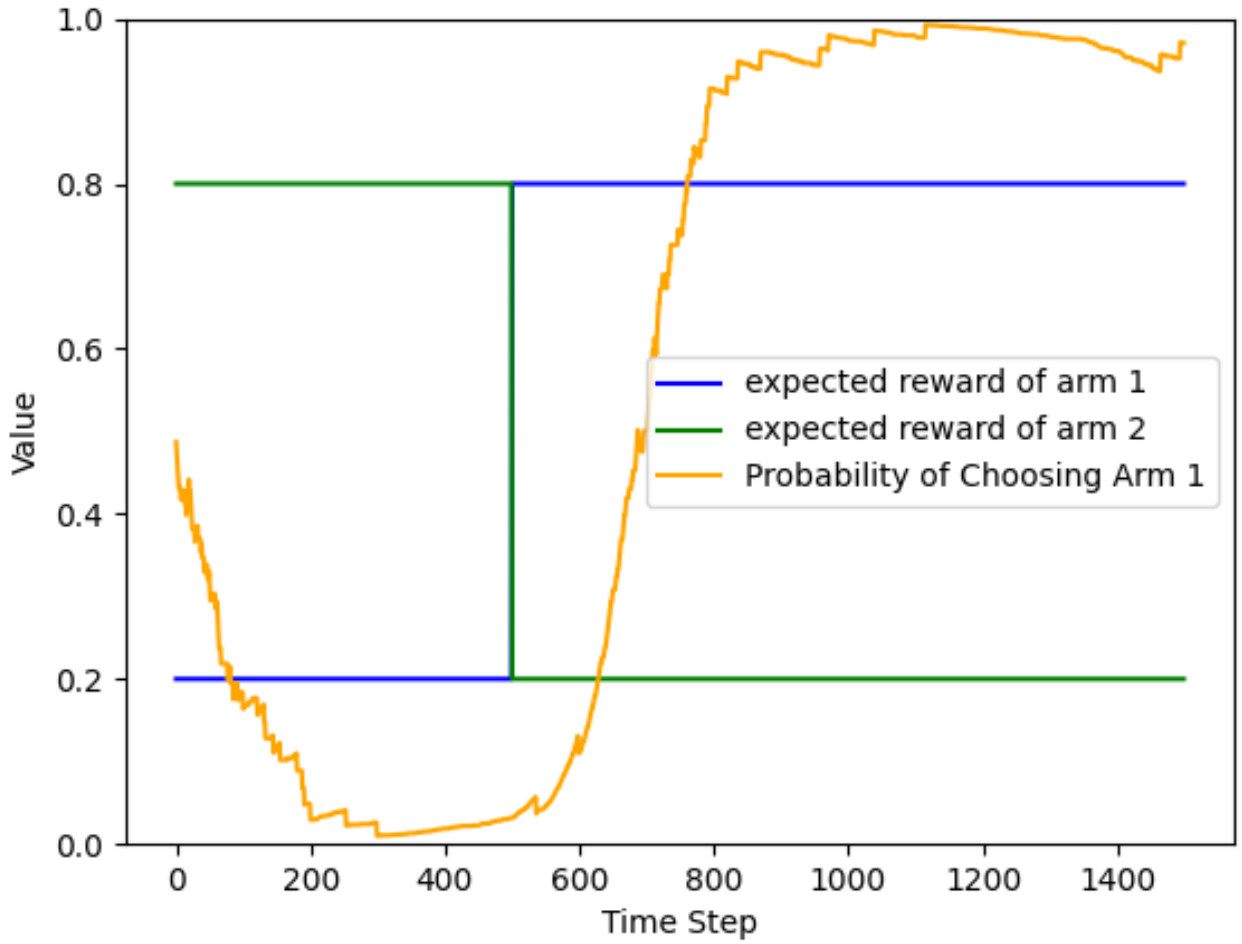
Update policy:

$$\pi_{t+1}(a) = \frac{\pi_t(a) \exp(\eta \hat{r}_t(a))}{\sum_{a' \in \mathcal{A}} \pi_t(a') \exp(\eta \hat{r}_t(a'))}$$

$\exp(\eta b)$

$\exp(\eta b)$

Solution 2: Reward Estimator with a Baseline



This is the EXP3 Algorithm

“**Exp**onential weight algorithm for **Exp**loration and **Exp**loitation”

- Exponential weights + either of the two solutions

The Role of Baseline

$$\hat{r}_t(a) = \frac{r_t(a) - b_t}{\pi_t(a)} \mathbb{I}\{a_t = a\}$$

$$\pi_{t+1}(a) = \frac{\pi_t(a) \exp(\eta \hat{r}_t(a))}{\sum_{a' \in \mathcal{A}} \pi_t(a') \exp(\eta \hat{r}_t(a'))} \quad \text{or} \quad \pi_{t+1} = \underset{\pi \in \Delta(\mathcal{A})}{\operatorname{argmax}} \left\{ \langle \pi, \hat{r}_t \rangle - \frac{1}{\eta} \mathrm{KL}(\pi, \pi_t) \right\}$$

Larger b_t: More exploratory (tends to decrease the probability of the action just chosen)

– needed to detect changes in the environment.

We usually set b_t to be close to the recent performance level of the learner itself

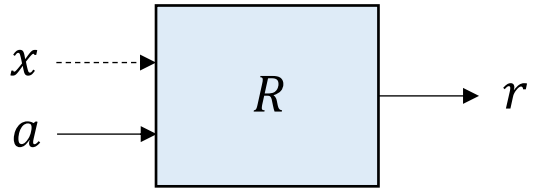
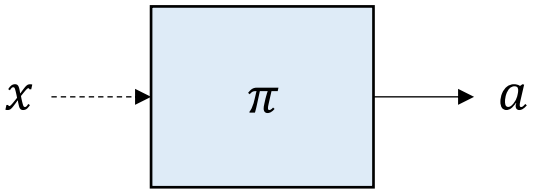
- When finding an action better than the learner itself, increase its probability
- Otherwise, decrease its probability

Summary

- Exponential weight update elements:
  - Incremental update (2 equivalent forms)
  - Importance weighting because we only observe the reward of the action we choose (otherwise the reward is **biased**)
  - **Baseline or extra uniform exploration** to encourage exploration

Review: Exploration Strategies for Bandits

x: context, a: action, r: reward

		MAB	CB
Value-based	 (context, action) to reward	Mean estimation + EG, BE Uncertainty as bonus	Regression + EG, BE
Policy-based	 context to action distribution	KL-regularized update with reward estimators (EXP3) + baseline, uniform exploration	**Next**

Contextual Bandits

Contextual Bandits

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

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

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

Learner observes $r_t(x_t, a_t)$

KL-Regularized Policy Updates

$$\pi_{t+1} = \underset{\pi \in \Delta(\mathcal{A})}{\operatorname{argmax}} \left\{ \sum_a \pi(a) \hat{r}_t(a) - \frac{1}{\eta} \mathrm{KL}(\pi, \pi_t) \right\}$$

$$\hat{r}_t(a) = \frac{r_t(a) - b_t}{\pi_t(a)} \mathbb{I}\{a_t = a\}$$

In practice, set b_t as a **running average** of $r_t(a_t)$ to track the learner's own performance.

The larger b_t is, the more exploration.

$$\theta_{t+1} = \underset{\theta}{\operatorname{argmax}} \left\{ \sum_a \pi_\theta(a|x_t)\, \hat{r}_t(x_t, a) - \frac{1}{\eta} \mathrm{KL}\big(\pi_\theta(\cdot\,|x_t), \pi_{\theta_t}(\cdot\,|x_t)\big) \right\}$$

$$\hat{r}_t(x_t, a) = \frac{r_t(x_t, a) - b_t(x_t)}{\pi_{\theta_t}(a|x_t)} \mathbb{I}\{a_t = a\}$$

KL-Regularized Policy Updates

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

Receive context x_t

Take action $a_t \sim \pi_{\theta_t}(\cdot|x_t)$ and receive reward $r_t(x_t, a_t)$

Create reward estimator $\hat{r}_t(x_t, a) = \frac{r_t(x_t,a) - b_t(x_t)}{\pi_{\theta_t}(a|x_t)} \mathbb{I}\{a_t = a\}$

Update

$$\theta_{t+1} = \operatorname*{argmax}_{\theta} \left\{ \sum_a \pi_\theta(a|x_t)\, \hat{r}_t(x_t, a) - \frac{1}{\eta} \mathrm{KL}\big(\pi_\theta(\cdot\,|x_t), \pi_{\theta_t}(\cdot\,|x_t)\big) \right\}$$

KL-Regularized Policy Updates with Batches (PPO)

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

For $i = 1, \ldots, N$:

Receive context x_i

Take action $a_i \sim \pi_{\theta_t}(\cdot|x_i)$ and receive reward $r_i(x_i, a_i)$

Create reward estimator $\hat{r}_i(x_i, a) = \frac{r_i(x_i,a) - b_t(x_i)}{\pi_{\theta_t}(a|x_i)} \mathbb{I}\{a_i = a\}$

For $j = 1, \ldots, M$:

Solve argmax

For minibatch $\mathcal{B} \subset \{1, 2, \ldots, N\}$ of size B:

$$\theta \leftarrow \theta + \nabla_\theta \frac{1}{B} \sum_{i \in \mathcal{B}} \left(\sum_a \pi_\theta(a|x_i)\, \hat{r}_i(x_i, a) - \frac{1}{\eta} \mathrm{KL}\big(\pi_\theta(\cdot\,|x_i), \pi_{\theta_t}(\cdot\,|x_i)\big) \right)$$

$$= \theta + \nabla_\theta \frac{1}{B} \sum_{i \in \mathcal{B}} \left(\frac{\pi_\theta(a_i|x_i)}{\pi_{\theta_t}(a_i|x_i)} (r_i(x_i, a_i) - b_t(x_i)) - \frac{1}{\eta} \mathrm{KL}\big(\pi_\theta(\cdot\,|x_i), \pi_{\theta_t}(\cdot\,|x_i)\big) \right)$$

$\theta_{t+1} \leftarrow \theta$

KL-Regularized Policy Updates with Batches (PPO)

$$\theta \leftarrow \theta + \nabla_\theta \frac{1}{B} \sum_{i \in \mathcal{B}} \left(\frac{\pi_\theta(a_i|x_i)}{\pi_{\theta_t}(a_i|x_i)} (r_i(x_i, a_i) - b_t(x_i)) - \underbrace{\frac{1}{\eta} \sum_a \pi_\theta(a|x_i) \log \frac{\pi_\theta(a|x_i)}{\pi_{\theta_t}(a|x_i)}}_{\mathrm{KL}\left(\pi_\theta(\cdot|x_i), \pi_{\theta_t}(\cdot|x_i)\right)} \right)$$

Estimating KL by Samples

http://joschu.net/blog/kl-approx.html

Sample $a_i \sim \pi_{\theta_t}(\cdot\,|x_i)$ and define $kl_i(\theta, \theta_t) = \dfrac{\pi_\theta(a_i|x_i)}{\pi_{\theta_t}(a_i|x_i)} - 1 - \log \dfrac{\pi_\theta(a_i|x_i)}{\pi_{\theta_t}(a_i|x_i)}$

Then $\mathbb{E}[kl_i(\theta, \theta_t)] \approx \mathrm{KL}\left(\pi_{\theta_t}(\cdot\,|x_i), \pi_\theta(\cdot\,|x_i)\right)$

Just need one sample of a_i

PPO with KL Estimator

HW2 task

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

$$kl_i(\theta, \theta_t) = \frac{\pi_\theta(a_i|x_i)}{\pi_{\theta_t}(a_i|x_i)} - 1 - \log\frac{\pi_\theta(a_i|x_i)}{\pi_{\theta_t}(a_i|x_i)}$$

For $i = 1, \dots, N$:

Receive context x_i

Take action $a_i \sim \pi_{\theta_t}(\cdot|x_i)$ and receive reward $r_i(x_i, a_i)$

Create reward estimator $\hat{r}_i(x_i, a) = \frac{r_i(x_i,a) - b_t(x_i)}{\pi_{\theta_t}(a|x_i)} \mathbb{I}\{a_i = a\}$

For $j = 1, \dots, M$:

For minibatch $\mathcal{B} \subset \{1, 2, \dots, N\}$ of size B:

$$\theta \leftarrow \theta + \nabla_\theta \frac{1}{B} \sum_{i \in \mathcal{B}} \left(\frac{\pi_\theta(a_i|x_i)}{\pi_{\theta_t}(a_i|x_i)} (r_i(x_i, a_i) - b_t(x_i)) - \frac{1}{\eta} kl_i(\theta, \theta_t) \right)$$

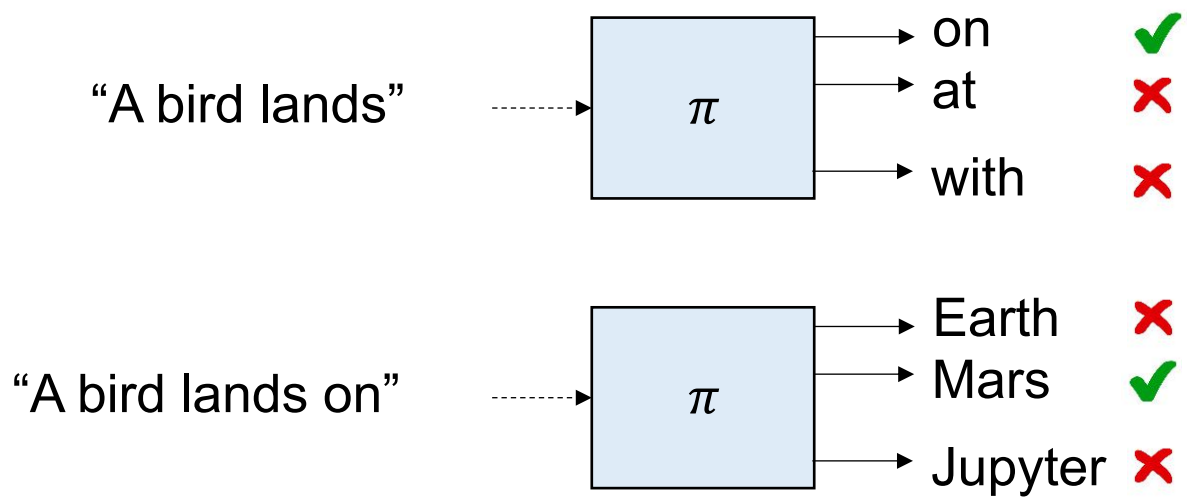
$\theta_{t+1} \leftarrow \theta$

Applications in Training LLM with RL

LLM Training

Phase 1: training with supervised learning (next-token prediction)

Given a **human generated** sentence: “A bird lands on Mars”



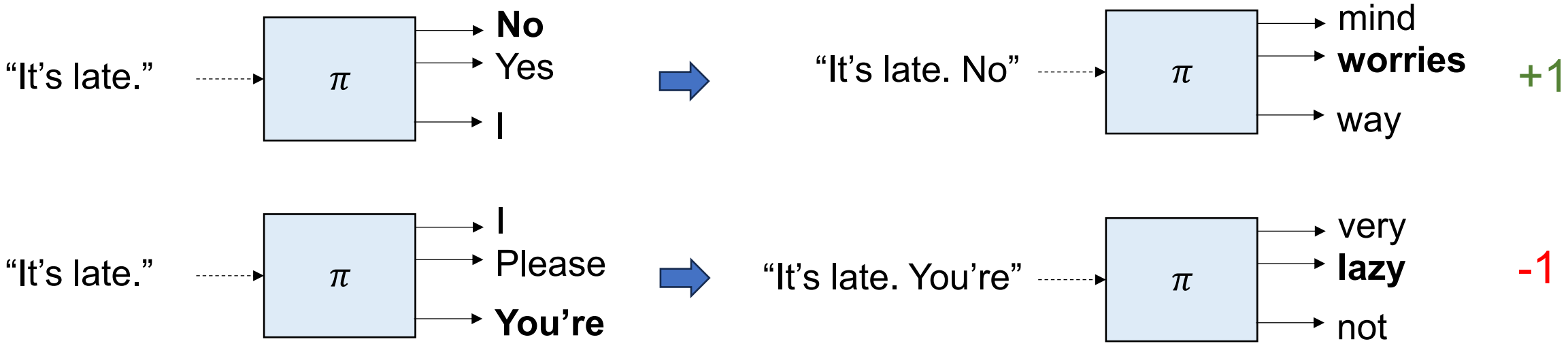
This gives a language model π_{SL} after training

LLM Training

Phase 2: training with reinforcement learning

Let the **machine (π_{SL}) generate** sentences:

Scores given by human



LLM Training

Phase 2: training with reinforcement learning

(x, a, r) tuples:

("It's late", "No", +1) ("It's late. No", "worries", +1)

("It's late", "You're", -1) ("It's late. You're", "lazy", -1)

Maximize $$\sum_i \left(\frac{\pi_\theta(a_i|x_i)}{\pi_{\theta_{\mathrm{SL}}}(a_i|x_i)} (r_i - b(x_i)) - \frac{1}{\eta} kl_i(\theta, \theta_{\mathrm{SL}}) \right)$$