max Total Reward

min Ave Regret

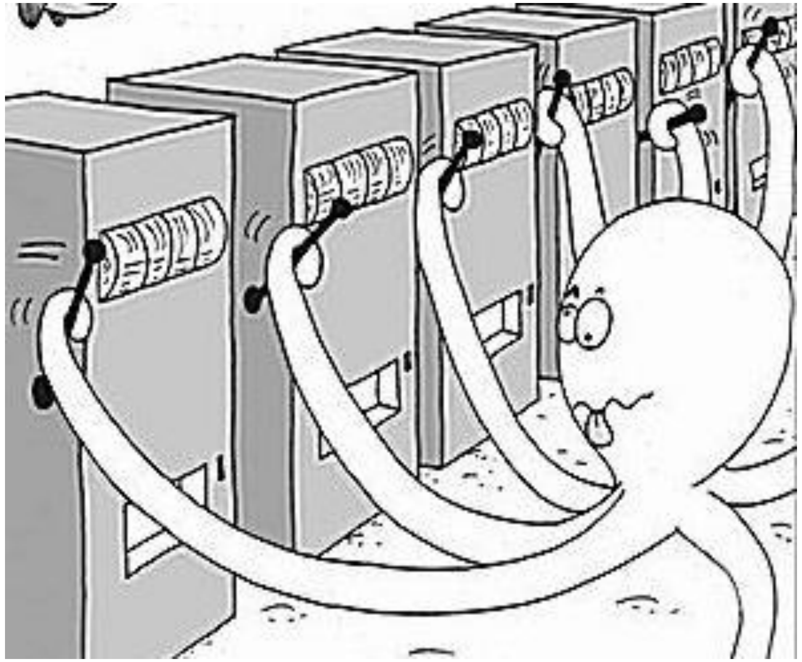
$A = \{1, \ldots, n\}$

$r_a^* \sim$ Distribution

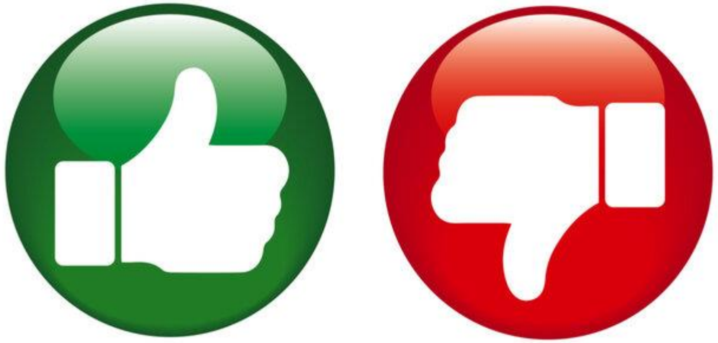
Multi-Armed Bandits

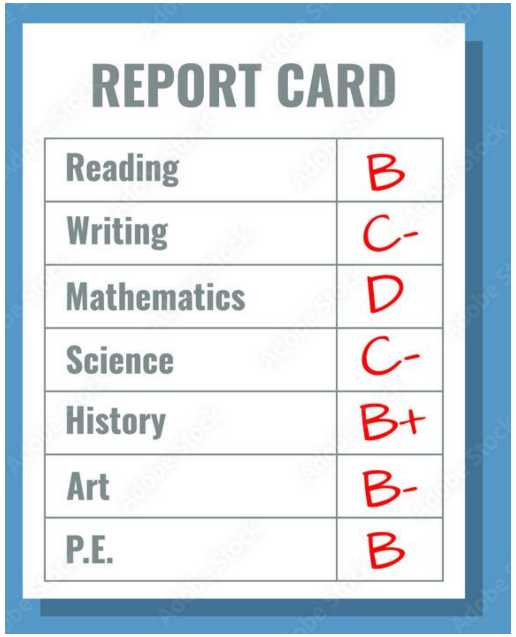
$\cancel{D = \{(s,a) \ldots (s', a')\}}$

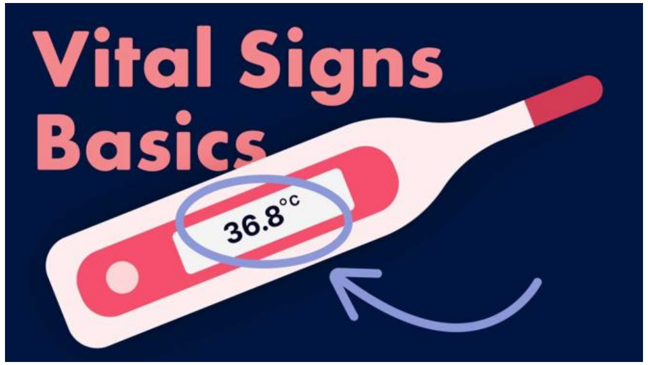
Daniel Brown



Evaluative feedback







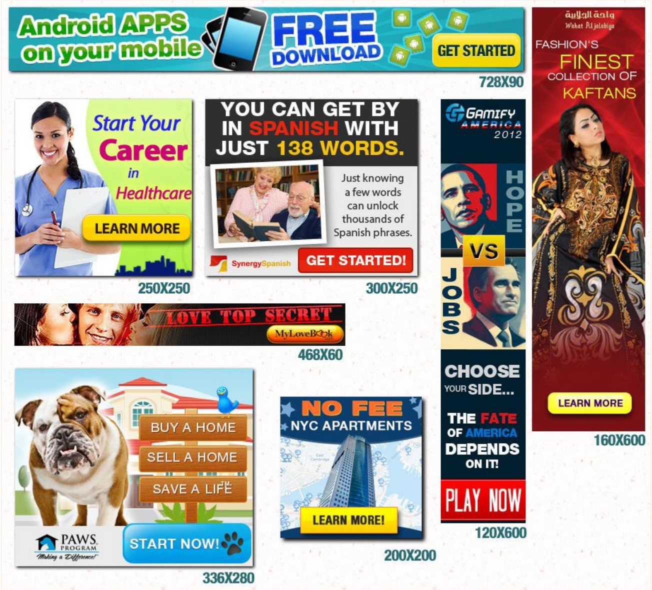
Applications

A? R?

$\pi \to A$

- Online Advertising and Recommendation
- Clinical Trials
- Robotics
- Dynamic Pricing
- Search Engine Optimization
- Education and Learning Platforms

MAB → Contextual Bandit



Problem formalism

Actions

- Arms $\mathcal{A} = \{a_1, \dots, a_k\}$
  - Each arm is associated with an unknown reward distribution
- Rewards $r_t(a_i)$ — $\mathbb{E}[r(a_i)] = \mu_{a_i}$
- Possible Goals
  - Maximize cumulative reward (Minimize regret)
  - Best arm identification
- Standard Assumptions
  - Independence: Rewards from each arm are independent
  - Stationarity: Reward distributions don't change over time

  $\hookrightarrow$ if violated $\varepsilon \triangleq 0.1$

How should we solve this problem?

$\pi \rightarrow a_3$

$+10$

a_1 a_2 a_3

$t=0$

-2

$t=1$ $+1$ $1/10$

$t=2$ $+1M$ $9/10$

$+10$

$+10$

$t=3$

$t=4$ $+1$ -10

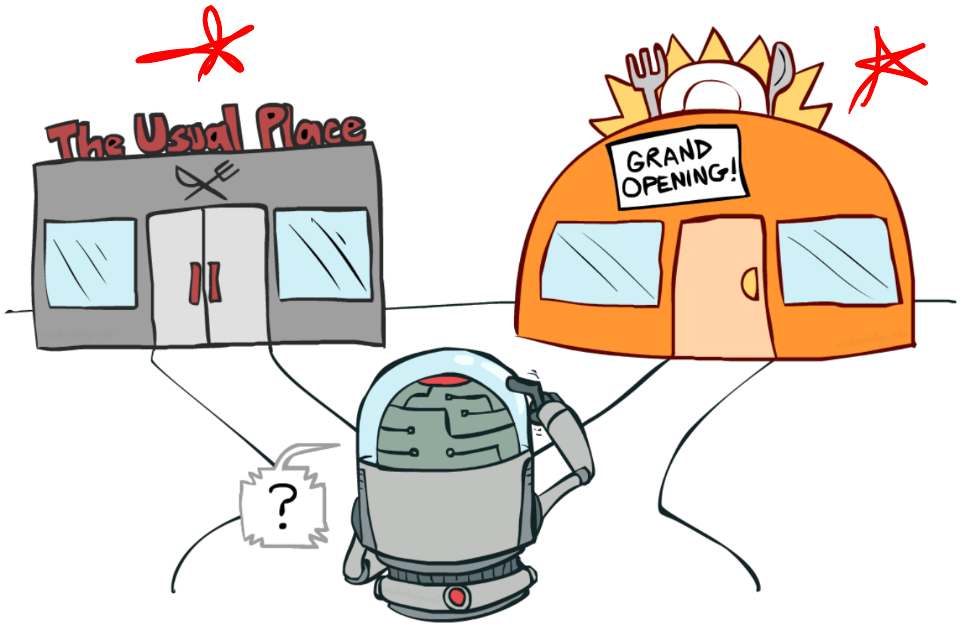
0 $+10$

Random

Big Regret

Greedy

Exploration



ϵ-Greedy

$\varepsilon \in (0,1)$ e.g. 0.1, 0.05, 0.01

* anneal/reduce/decay ε over time

loop:

rand $\in [0,1]$

random (if rand $< \varepsilon$:

explore action at random

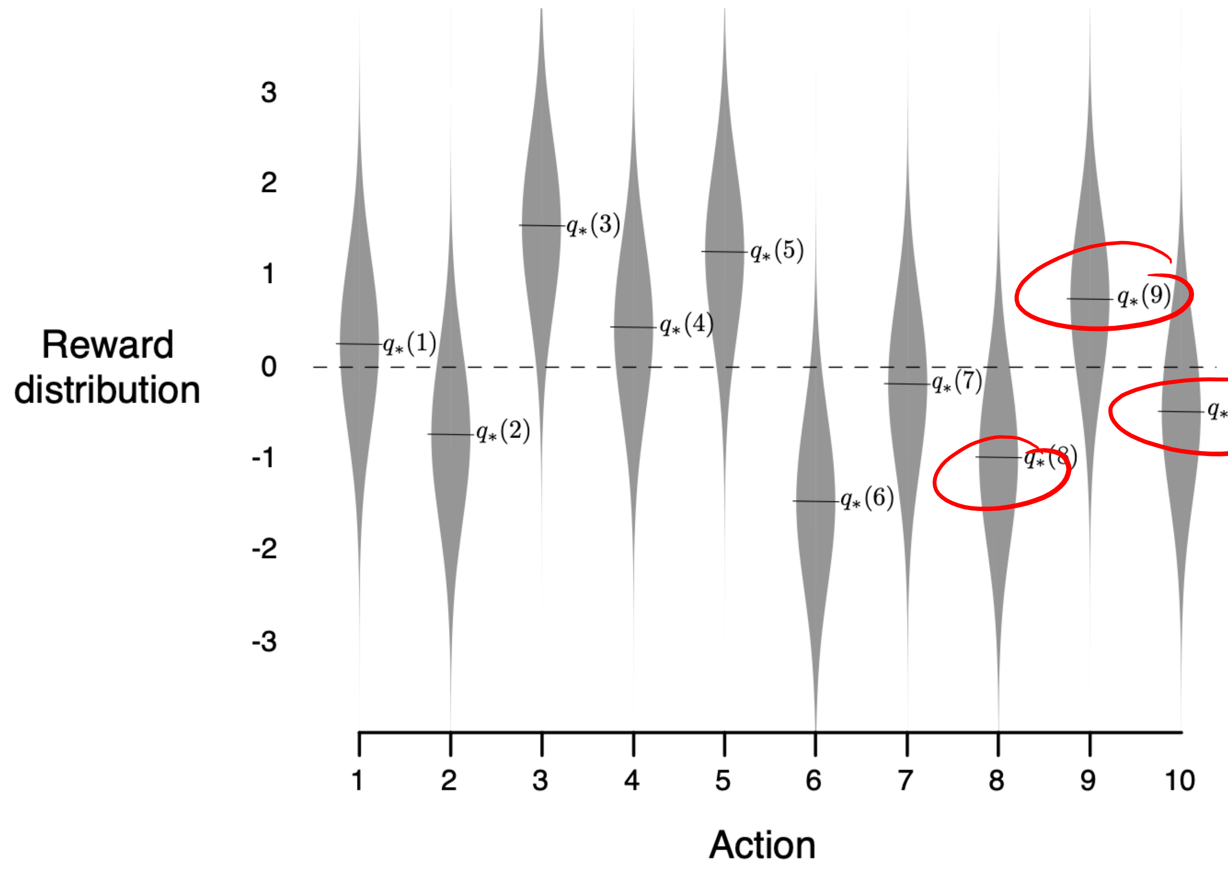
greedy (else:

act greedy

2.3 The 10-armed Testbed

To roughly assess the relative effectiveness of the greedy and ε-greedy action-value methods, we compared them numerically on a suite of test problems. This was a set of 2000 randomly generated k-armed bandit problems with $k = 10$. For each bandit problem, such as the one shown in Figure 2.1, the action values, $q_*(a)$, $a = 1, \ldots, 10$,

$q_*(1)$



$N(0,1)$

Figure 2.1: An example bandit problem from the 10-armed testbed. The true value $q_*(a)$ of each of the ten actions was selected according to a normal distribution with mean zero and unit variance, and then the actual rewards were selected according to a mean $q_*(a)$ unit variance normal distribution, as suggested by these gray distributions.

$S \sim N(a,b)$ $r \sim N(0,1)$ $S = a + br$

Sutton/Barto figure

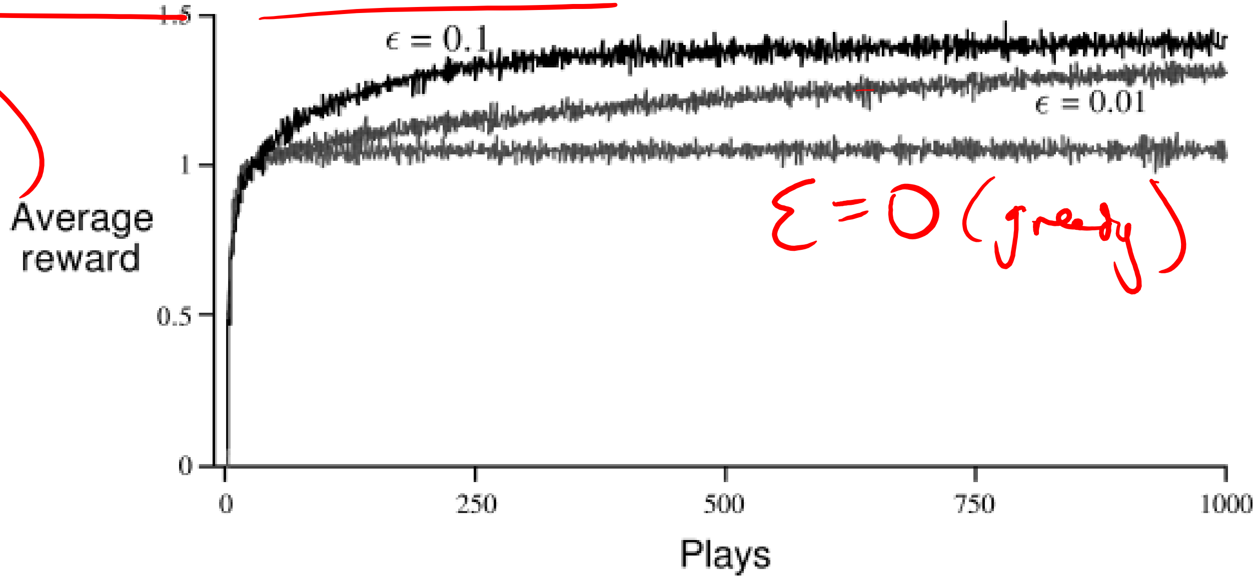
random.randn( ) $\sim N(0,1)$

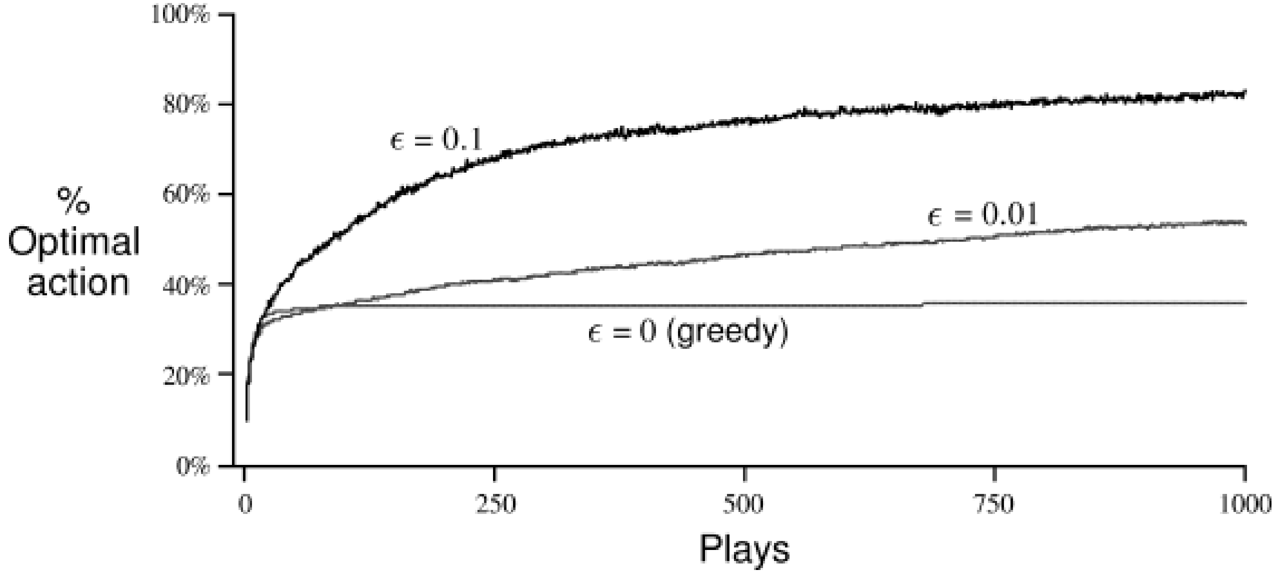
- 10 arms
- Each arm has stochastic reward

$$r \sim N(Q^*(a), 1)$$

- Averaged over 2000 bandit problems where each problem starts with $Q^*(a) \sim N(0,1)$ for all a

$N(0.1,1)$ $N(-0.2,1)$ $N(1.5,1)$





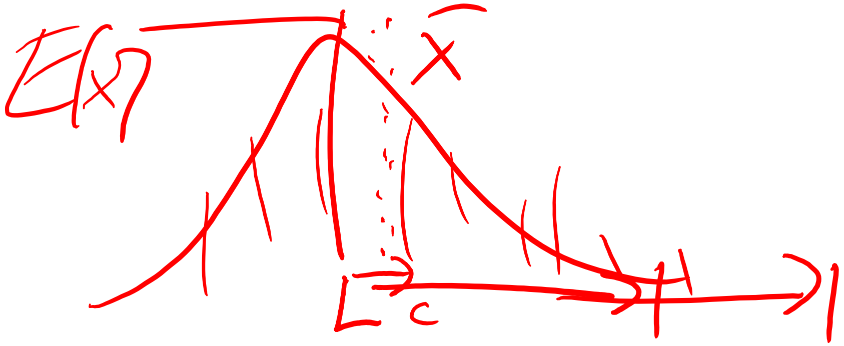
Problems?

ε?

Boltzmann (Softmax) Exploration

$$\Pr(a) = \frac{e^{\beta \hat{Q}(a)}}{\sum_{a' \in A} e^{\beta \hat{Q}(a')}}$$

Chernoff-Hoeffding Inequality



-rE

- Let X be a random variable in the range [0,1] and $x_1, x_2, \dots, x_n$ be n independent and identically distributed samples of X.

$\hat{Q}(a)$

- Let $\bar{X} = \frac{1}{n} \sum_i x_i$ (the empirical average)
- Then we have $P(\bar{X} \geq \mathbb{E}[X] + c) \leq e^{-2nc^2}$

Some fun math!

$E[X] \xrightarrow{c} \bar{X}$

99%

95% $\quad \delta = 0.05$

- $P(\bar{X} \geq \mathbb{E}[X] + c) \leq e^{-2nc^2}$
- Typically, we want to pick some kind of high confidence $1 - \delta$ such that we are very confident about our sample mean being close to the true expectation.
- If we want

0.01

$$P(\bar{X} \geq \mathbb{E}[X] + c) \leq \delta$$

What is c in terms of δ?

$\delta = e^{-2nc^2}$ $\quad \sqrt{\frac{\log \delta}{-2n}} = c$

More math

$$\overset{c}{\longleftrightarrow}$$
$\mathbb{E}[X]$ $\quad$ $\bar{X}$

- We can pick δ to be whatever we want, so let's pick $\frac{1}{t^4}$
- If we select $\delta = t^{-4}$

$t \to \infty$

$\delta \to 0$

What is c?

$$c = \sqrt{\frac{\log \delta}{-2n}} = \sqrt{\frac{\log t^{-4}}{-2n}} = \sqrt{\frac{-4 \log t}{-2n}}$$

$$= \sqrt{\frac{2 \log t}{n}}$$

UCB1 (UCB = Upper Confidence Bound)

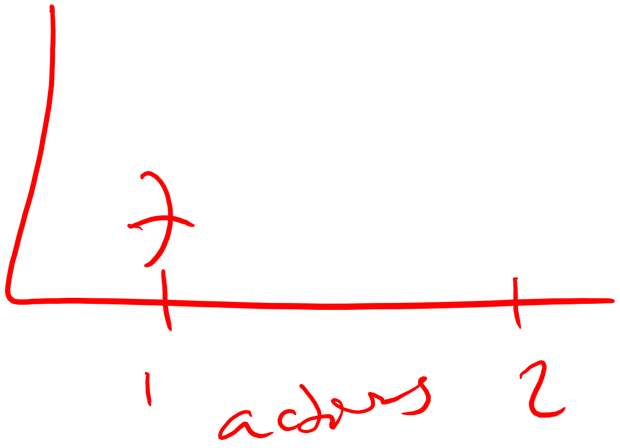
Key Idea: Optimism in the face of uncertainty

- Play each action once to get initial averages of arm values
- Keep track of counts of pulls for each arm n_i
- At each step t, select $\arg\max_{i} \bar{X}_i + c(i,t)$
  - Where $c(i,t) = \sqrt{\frac{2 \cdot log(t)}{n_i}}$

t = total # actions

$t \to \infty$

r



Regret

- Define μ^* as the maximum expected payoff over all k arms
- Regret(T) = $T\mu^* - \sum_{t=1}^{T} r_t$

- Epsilon-Greedy Regret
  - $O(T)$

fixed ε

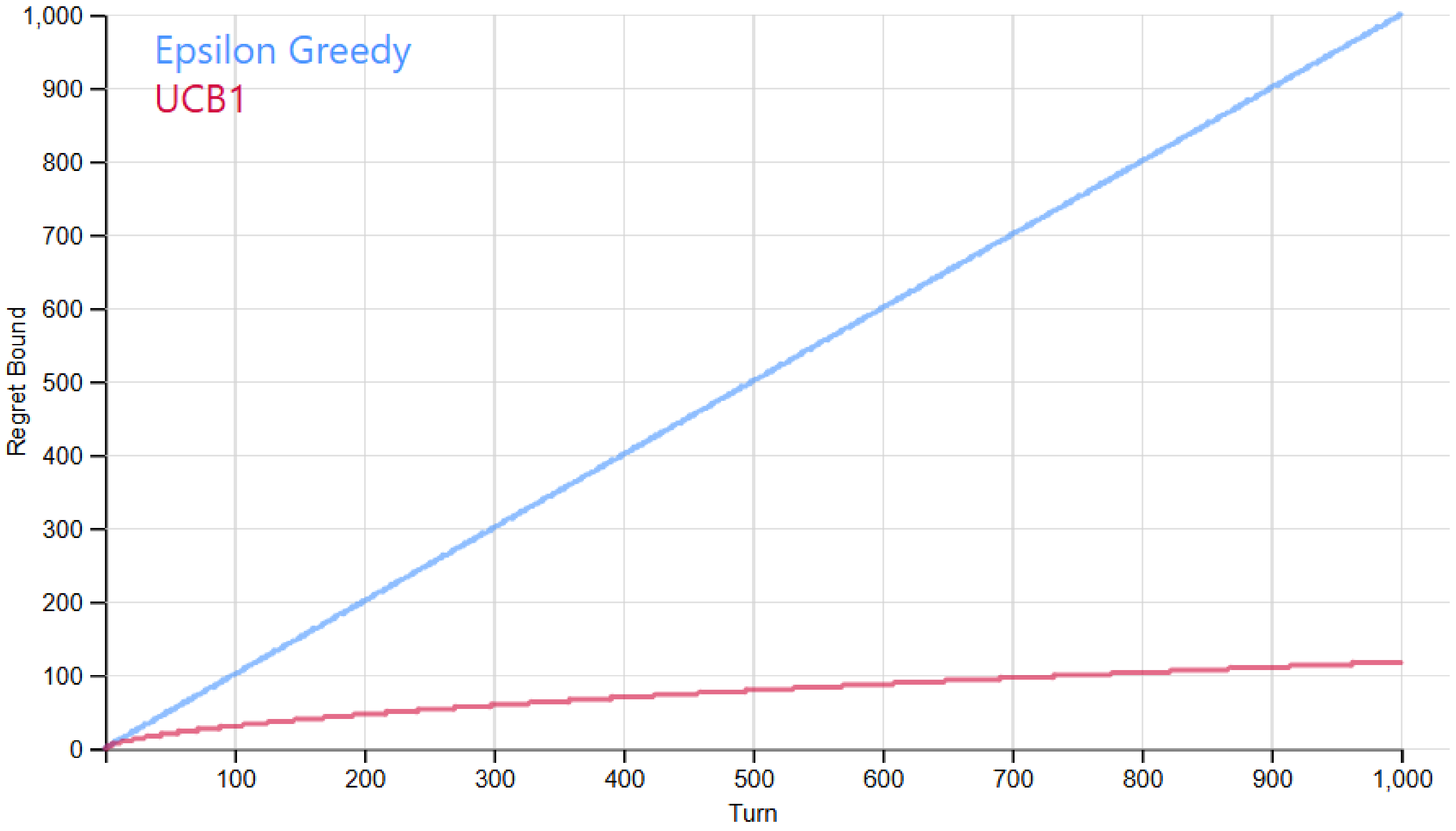
K = # arms

$K \ll T$

- UCB1 Regret
  - $O(\sqrt{kTlog(T)})$

- A **No-Regret** algorithm is such that Regret(T)/T → 0 as $T \to \infty$
  - Average regret goes to zero

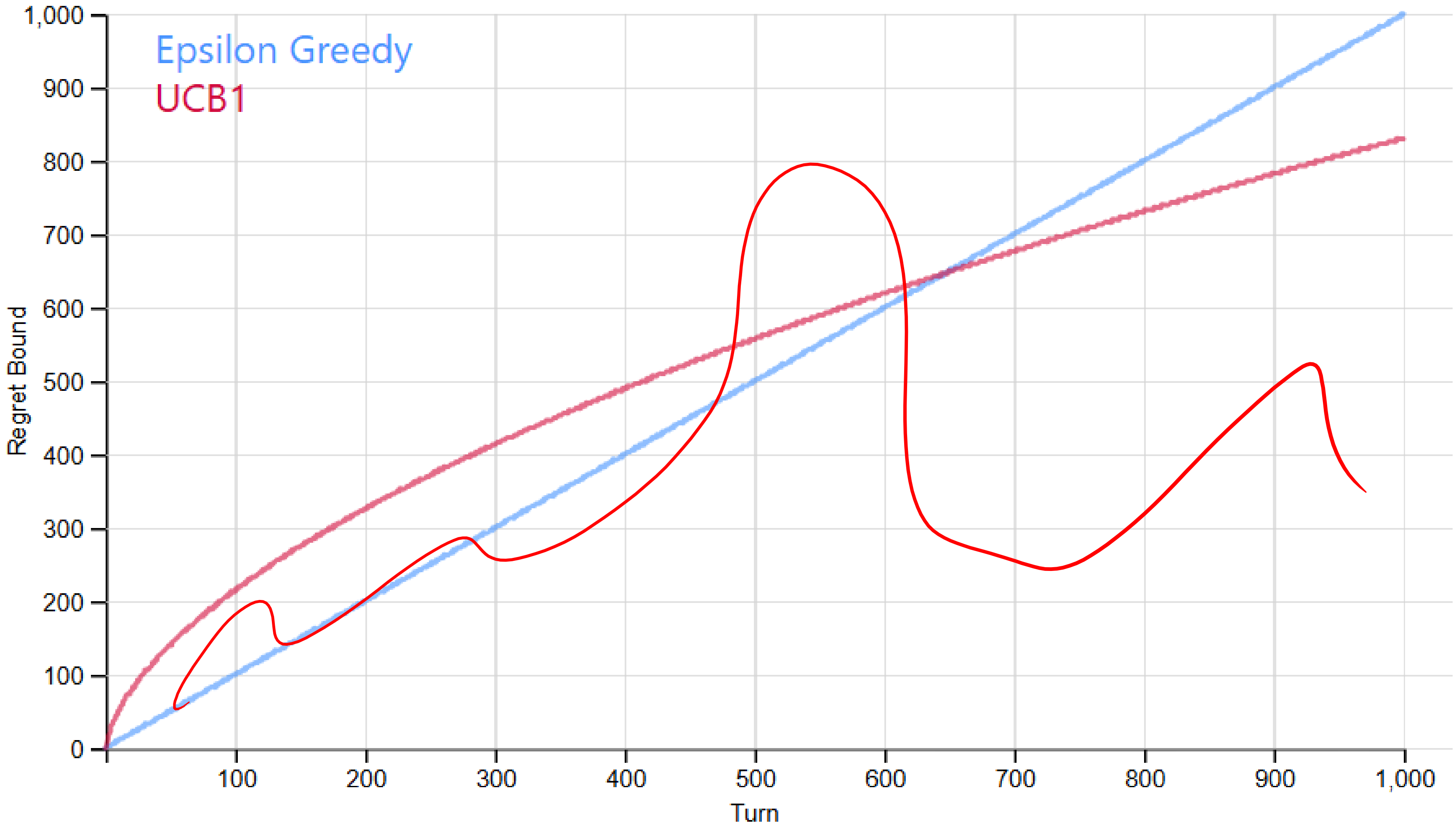
Regret Bound vs. Turn



k (number of arms): 2 T (number of steps): 1000

https://cse442-17f.github.io/LinUCB/

Regret Bound vs. Turn



k (number of arms): 100 T (number of steps): 1000

https://cse442-17f.github.io/LinUCB/

Notes

- The version we derived is for rewards in range [0,1]
- What happens if rewards are in range [a,b]?
  - Just scale the upper confidence value

$$\mathrm{c}(i,t) = (b-a)\sqrt{\frac{2 \cdot log(t)}{n_i}}$$

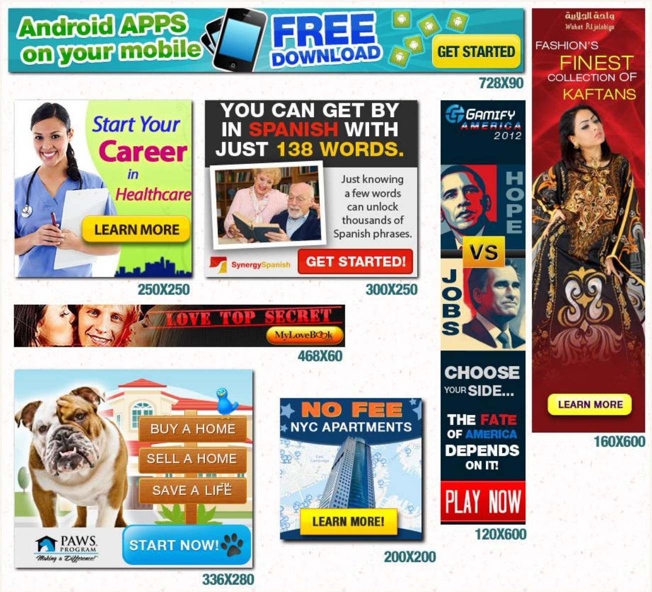
In practice the scaling term is often just treated as a hyperparameter that controls exploration vs. exploitation.

$$\mathrm{c}(i,t) = \alpha\sqrt{\frac{log(t)}{n_i}}$$

What problems do we have with vanilla MAB?

Bad/Strong Assumptions

- stationarity
- no state/context/obs



MIT OCW

edX

FutureLearn

ALISON

UDACITY

coursera

KHAN ACADEMY

UoPeople

udemy

iversity

Contextual Bandits

MAB

no context

$r_t(a_t)$

- **High-level definition**
  At each round t:
  - Observe **context** x_t
  - Choose action $a_t \in \{1, \ldots, K\}$
  - Observe reward $r_t(a_t, x_t)$

Each arm has a context-dependent reward function:

$$\mathbb{E}[r \mid x, a] = f_a(x)$$

LinUCB

$x_0, a_0, r_0(a_0, x_0)$

$x_1, a_1, r_1(a_1, x_1)$

$x_2, a_0, r_2(a_0, x_2)$

$x \to y$

$x \to r$

- **Assumption**
  - For each arm a:
  - Reward is linear in features
  - Separate parameter vector per arm

$$\mathbb{E}[r \mid x, a] = x^\top \theta_a$$

$D_{a_1} = \{ (x, r) \dots \}$

$\theta_{a_1}^\top x \to r_{a_1}$

- How should we choose arms (actions) given a context vector x?
  - We can maintain one linear regression model per arm
  - Plus uncertainty (e.g. upper confidence bound)

$\theta_{a_2}^\top x \to r_{a_2}$

$\vdots$

$\theta_{a_N}^\top x \to r_{a_N}$

Aside: Linear Regression

$x \in \mathbb{R}^d$

$$(x_0, y_0)$$
$$(x_1, y_1)$$
$$\vdots$$
$$(x_N, y_N)$$

$$\vec{y} = \begin{bmatrix} y_0 \\ \vdots \\ y_N \end{bmatrix}$$

- Model assumption

$$y = x^\top \theta + \varepsilon$$

- Closed form solution

$$\hat{\theta} = (X^\top X)^{-1} X^\top \vec{y}$$

$$X = \begin{bmatrix} - x_1 - \\ - x_2 - \\ \vdots \\ - x_N - \end{bmatrix}$$

$$X^\top \vec{y} = \underbrace{X^\top}_{d \times N} \underbrace{X}_{N \times d} \hat{\Theta}$$

$$(X^\top X)^{-1} X^\top \vec{y} = \hat{\Theta}$$

$$y = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$N > d$

Ridge Regression (adds L2 regularization)

$$\hat{\theta} = \arg\min_{\theta} \left(\|X\theta - y\|^2 + \lambda \|\theta\|^2 \right)$$

$$X = \begin{bmatrix} x \\ \vdots \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

- Also has a closed form solution:

$$\hat{\theta}_a = (X_a^\top X_a + \lambda I)^{-1} X_a^\top y_a$$

- How would you solve this in an online way? E.g., you only get one sample (x_t, y_t) at each timestep and you want to iteratively update θ

We want something like this but that can be updated as we get new data

$$\hat{\theta} = \underbrace{(X^\top X + \lambda I)}_{A}{}^{-1} \underbrace{X^\top y}_{b}$$

We can write $\hat{\theta} = A^{-1} b$

At each time t you observe (x_t, y_t)

What is A_t and b_t such that we can estimate θ given the data so far?

$x_t,\ y_t,\ t$

$X =$

$y =$

$$A_t = \lambda I + \sum_{s=1}^{t} x_s x_s^\top + x_{t+1} x_{t+1}^\top$$

$$b_t = \sum_{s=1}^{t} y_s x_s + y_{t+1} x_{t+1}$$

$$\hat{\theta} = A_t^{-1} b_t$$

Quiz Answer: D

LinUCB

θ_a

- Optimism in the face of uncertainty in ***function space.***

$$\boxed{\text{LinUCB} = \text{online ridge regression} + \text{optimism via confidence ellipsoids}}$$

LinUCB Algorithm

Assumption

For each arm $a \in \{1, \dots, K\}$:

$$\mathbb{E}[r \mid x, a] = x^\top \theta_a$$

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

Observe context

$$x_t \in \mathbb{R}^d$$

For each arm a:

$$\hat{\theta}_a = A_a^{-1} b_a$$

$$\mathrm{UCB}_a(x_t) = x_t^\top \hat{\theta}_a + \alpha \sqrt{x_t^\top A_a^{-1} x_t}$$

Select arm

$$a_t = \arg\max_a \ \mathrm{UCB}_a(x_t)$$

Initialization

λI 0

For each arm a:

$$A_a \leftarrow \lambda I_d \qquad b_a \leftarrow 0_d$$

Observe reward

$$r_t \in \mathbb{R}$$

Update (chosen arm only)

$$A_{a_t} \leftarrow A_{a_t} + x_t x_t^\top$$

$$b_{a_t} \leftarrow b_{a_t} + r_t x_t$$

LinUCB

- LinUCB puts confidence ellipsoids on models!

- Works really well in practice!
- Very data efficient.
- Has been used heavily in production systems (ads, recommendations, etc)

Modern, Deep Learning Approaches

(x, a, r)

- Neural Contextual Bandits
  - Replace linear model with neural network $\quad r \approx f_\theta(x, a)$
  - Train online from feedback
  - Figuring out good exploration/uncertainty is challenging
- Uncertainty via approximations
  - Ensembles
  - NeuralUCB (linear UCB on last-layer features)
    - Key idea: Use deep learning for representation, classical bandits for exploration.

$\theta_a^T x$

$x, a_1, \ldots, a_N$

$f_{\theta_2}(x, a)$

$f_{\theta_3}(x, a)$

$\arg\max \ \bar{f}_\theta(x, a) + \text{var}$

Transitioning to Reinforcement Learning

- At what point does a contextual bandit become full reinforcement learning?

$$R_t = \sum_{t=0}^{T} r_t$$

$$a \rightarrow x \xrightarrow{a} x$$