Lecture 13. Multi-armed bandits

COMP90051 Statistical Machine Learning

Semester 2, 2019
Lecturer: Ben Rubinstein

THE UNIVERSITY OF
MELBOURNE

This lecture

- Stochastic multi-armed bandits
  - *Where we learn to take actions; we receive only indirect supervision in the form of rewards; and we only observe rewards for actions taken – the simplest setting with an explore-exploit trade-off.*
  - Sequential decision making under uncertainty
  - (ε)-greedy
  - UCB algorithm

Exploration vs. Exploitation

CC0

Exploration vs. Exploitation

- "Multi-armed bandit" (MAB)
  * Simplest setting for balancing exploration, exploitation
  * Same family of ML tasks as reinforcement learning

- Numerous applications
  * Online advertising
  * Portfolio selection
  * Caching in databases
  * Stochastic search in games (e.g. AlphaGo!)
  * Adaptive A/B testing
  * ….

CC0

Stochastic MAB setting

- Possible actions $\{1, \ldots, k\}$ called "arms"
  - Arm i has distribution P_i on bounded rewards with mean μ_i

- In round $t = 1 \ldots T$
  - Play action $i_t \in \{1, \ldots, k\}$ *(possibly randomly)*
  - Receive reward $X_{i_t}(t) \sim P_{i_t}$

- Goal: minimise cumulative regret
  - $\mu^* T - \sum_{t=1}^{T} E\left[X_{i_t}(t)\right]$ Expected cumulative reward of bandit

    Best expected cumulative reward with hindsight

  where $\mu^* = \max_i \mu_i$
  - Intuition: Do as well as a rule that is simple but has knowledge of the future

5

Greedy

- At round t

  * Estimate value of each arm i as average reward observed

$$Q_{t-1}(i) = \begin{cases} \dfrac{\sum_{s=1}^{t-1} X_i(s) 1[i_s = i]}{\sum_{s=1}^{t-1} 1[i_s = i]}, & \text{if } \displaystyle\sum_{s=1}^{t-1} 1[i_s = i] > 0 \\ Q_0, & \text{otherwise} \end{cases}$$

  … some init constant $Q_0(i) = Q_0$ used until arm i has been pulled

  * Exploit, baby, exploit!
  $$i_t \in \arg\max_{1 \le i \le k} Q_{t-1}(i)$$

  * Tie breaking randomly

- What do you expect this to do? Effect of initial Qs?

ε-Greedy

- ## At round t

  * Estimate value of each arm i as average reward observed

  $$Q_{t-1}(i) = \begin{cases} \dfrac{\sum_{s=1}^{t-1} X_i(s) 1[i_s = i]}{\sum_{s=1}^{t-1} 1[i_s = i]}, & \text{if } \sum_{s=1}^{t-1} 1[i_s = i] > 0 \\ Q_0, & \text{otherwise} \end{cases}$$

  … some init constant $Q_0(i) = Q_0$ used until arm i has been pulled
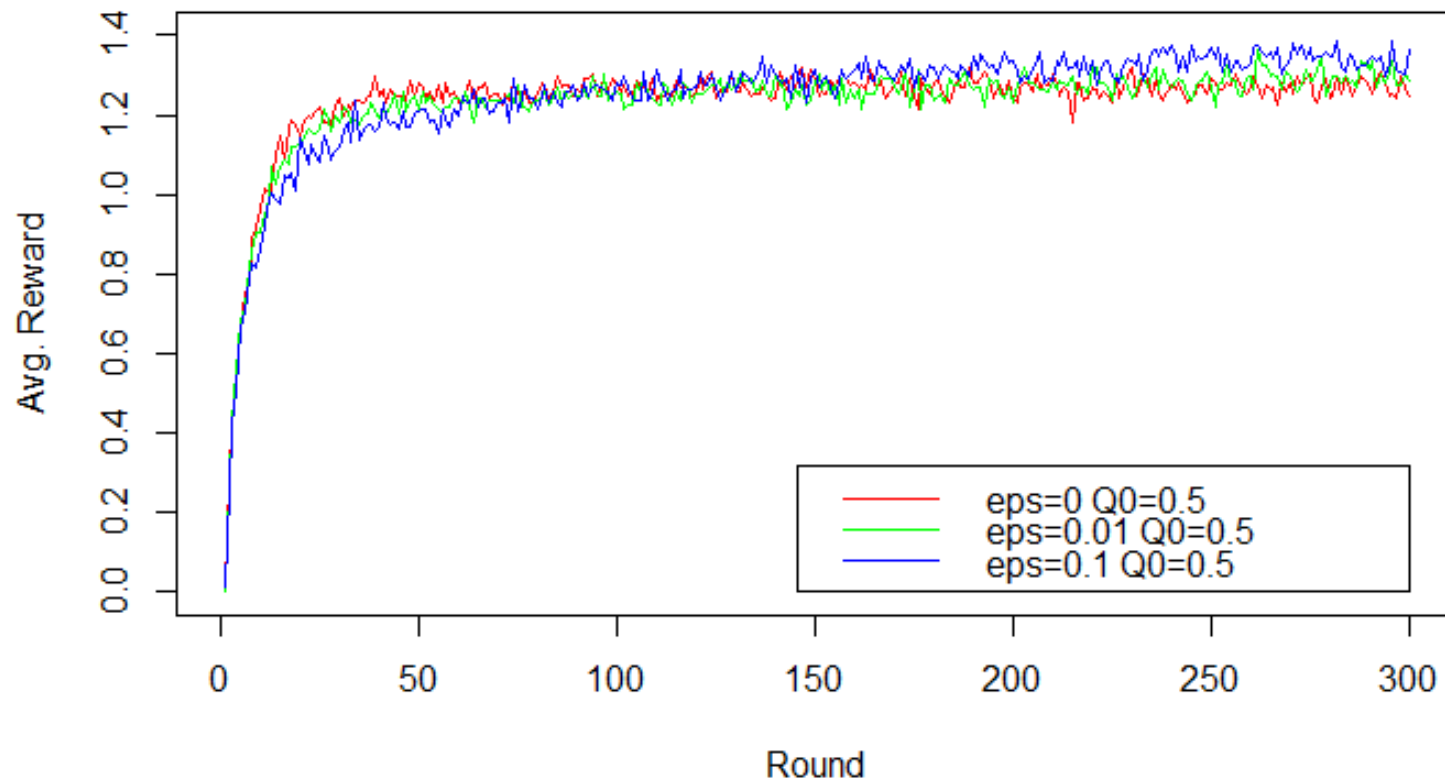
  * Exploit, baby exploit… probably; or possibly explore

  $$i_t \sim \begin{cases} \arg\max_{1 \leq i \leq k} Q_{t-1}(i) & w.p. \ 1 - \varepsilon \\ \text{Unif}(\{1, \dots, k\}) & w.p. \ \varepsilon \end{cases}$$

  * Tie breaking randomly

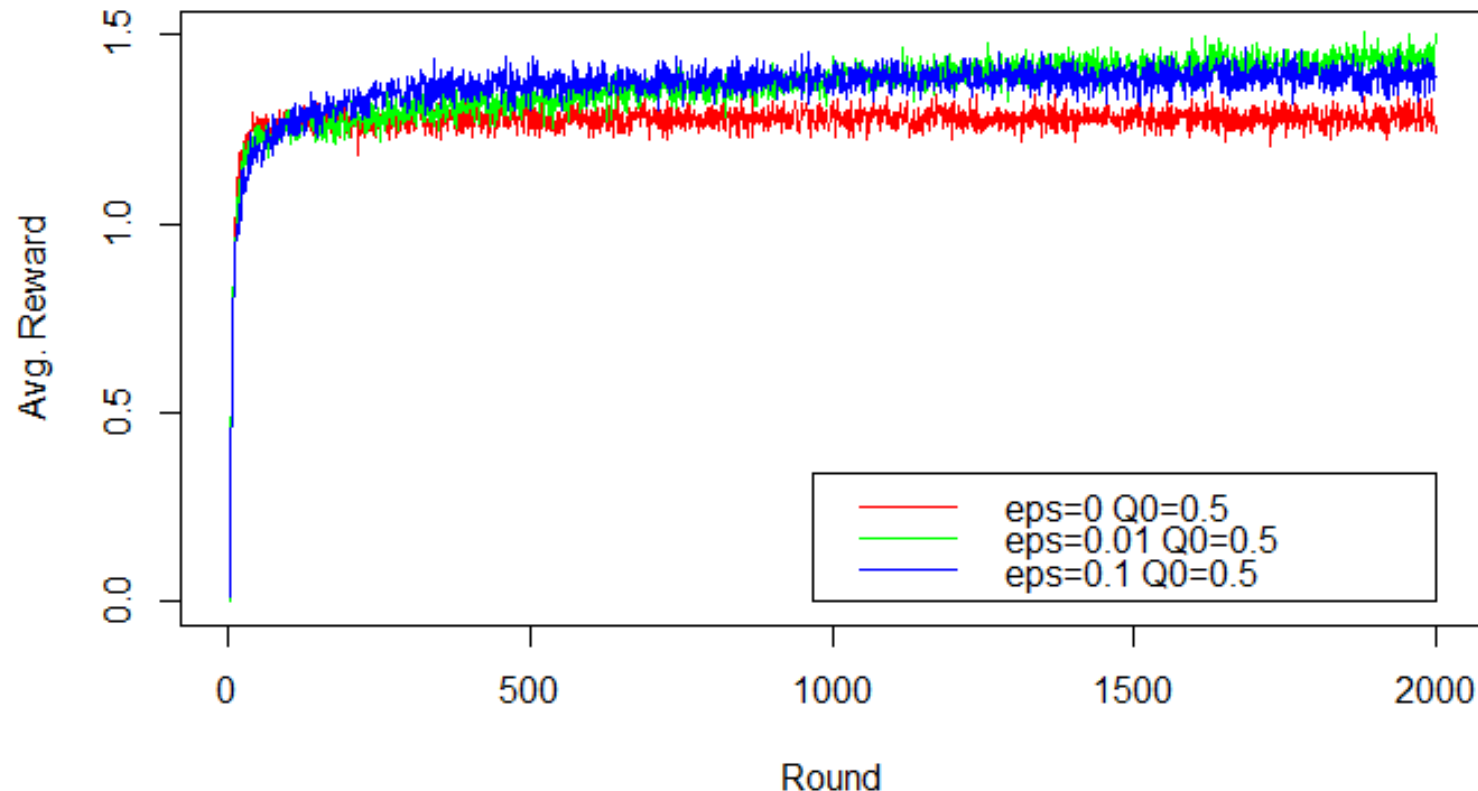- ## Hyperparam. ε controls exploration vs. exploitation

Kicking the tyres


- 10-armed bandit
- Rewards $P_i = Normal(\mu_i, 1)$ with $\mu_i \sim Normal(0,1)$
- Play game for 300 rounds
- Repeat 1,000 games, plot average per-round rewards

Kicking the tyres: More rounds

- Greedy increases fast, but levels off at low rewards
- ε-Greedy does better long-term by exploring
- 0.01-Greedy initially slow (little explore) but eventually superior to 0.1-Greedy (exploits after enough exploration)

9

Optimistic initialisation improves Greedy
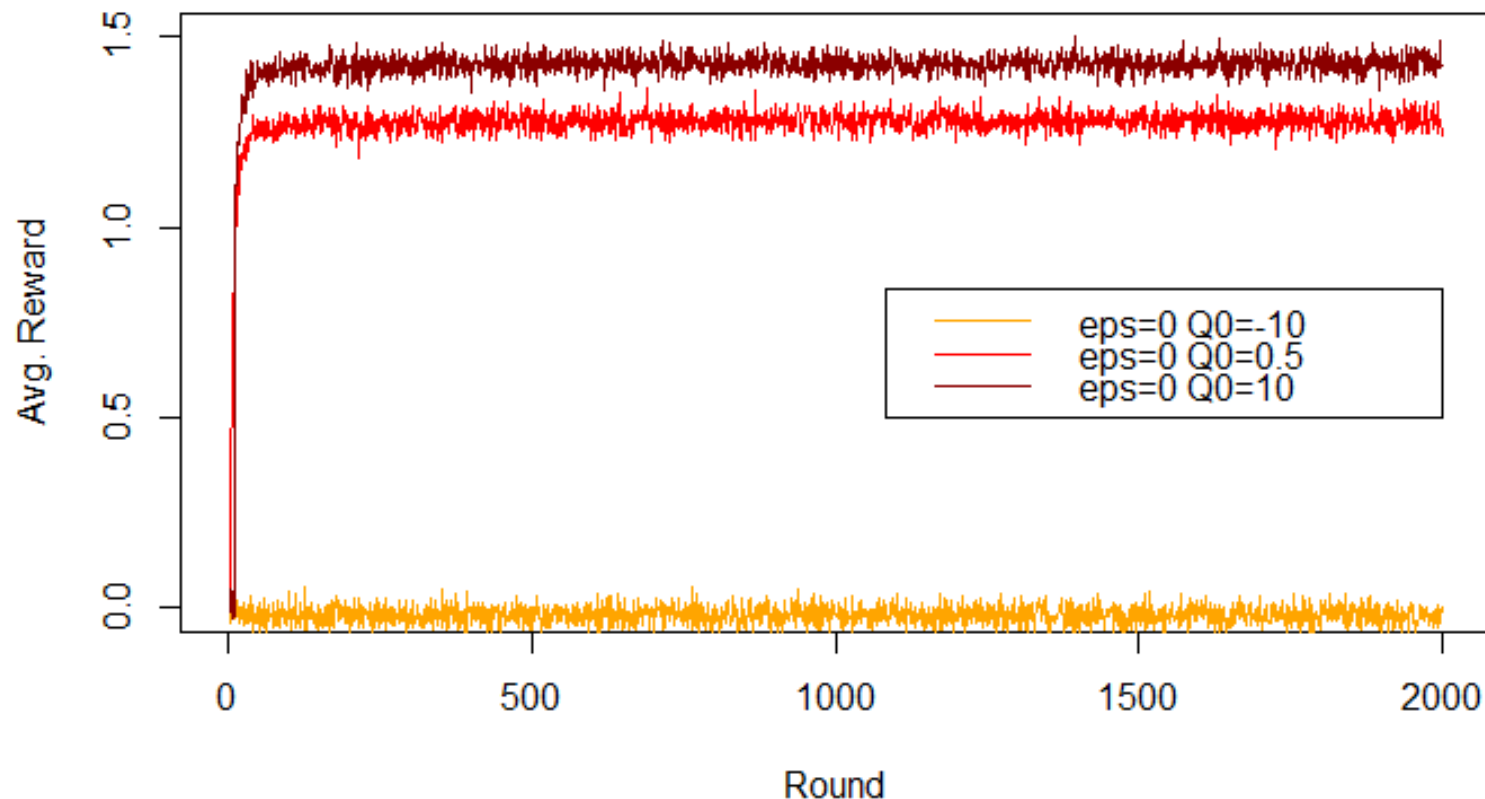
- **Pessimism**: Init Q's below observable rewards → Only try one arm
- **Optimism**: Init Q's above observable rewards → Explore arms once
- Middle-ground init Q → Explore arms at most once

But pure greedy never explores an arm more than once

Limitations of ε-Greedy

- While we can improve on basic Greedy with optimistic initialisation and decreasing ε…

- Exploration and exploitation are too "distinct"
  * Exploration actions completely blind to promising arms
  * Initialisation trick only helps with "cold start"

- Exploitation is blind to confidence of estimates

- These limitations are serious in practice

11

(Upper) confidence interval for Q estimates

- **Theorem**: Hoeffding's inequality
  * Let $X_1, \ldots, X_n$ be i.i.d. random variables in [0,1] mean μ, denote by $\overline{X_n}$ their sample mean
  * For any $\varepsilon \in (0,1)$ with probability at least $1 - \varepsilon$

$$\mu \leq \overline{X_n} + \sqrt{\frac{\log(1/\varepsilon)}{2n}}$$
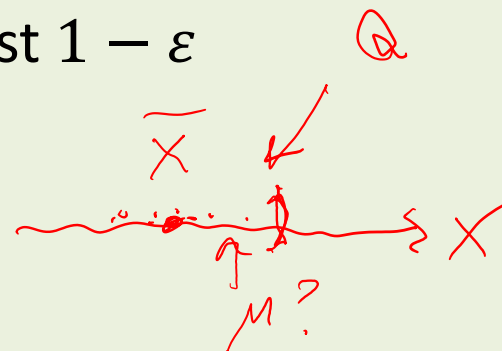
- Application to $Q_{t-1}(i)$ estimate – also i.i.d. mean$!!$
  * Take $n = N_{t-1}(i) = \sum_{s=1}^{t-1} 1[i_s = i]$ number of i plays
  * Then $\overline{X_n} = Q_{t-1}(i)$
  * Critical level $\varepsilon = 1/t$ (Lai & Robbins '85), take $\varepsilon = 1/t^4$

12

Upper Confidence Bound (UCB) algorithm

explore

boost

- At round t
  - * Estimate value of each arm i as average reward observed

$$Q_{t-1}(i) = \begin{cases} \hat{\mu}_{t-1}(i) + \sqrt{\dfrac{2\log(t)}{N_{t-1}(i)}}, & \text{if } \sum_{s=1}^{t-1} 1[i_s = i] > 0 \\ Q_0, & \text{otherwise} \end{cases}$$

…some constant $Q_0(i) = Q_0$ used until arm i has been pulled; where:

$$N_{t-1}(i) = \sum_{s=1}^{t-1} 1[i_s = i] \qquad \hat{\mu}_{t-1}(i) = \frac{\sum_{s=1}^{t-1} X_i(s) 1[i_s = i]}{\sum_{s=1}^{t-1} 1[i_s = i]}$$
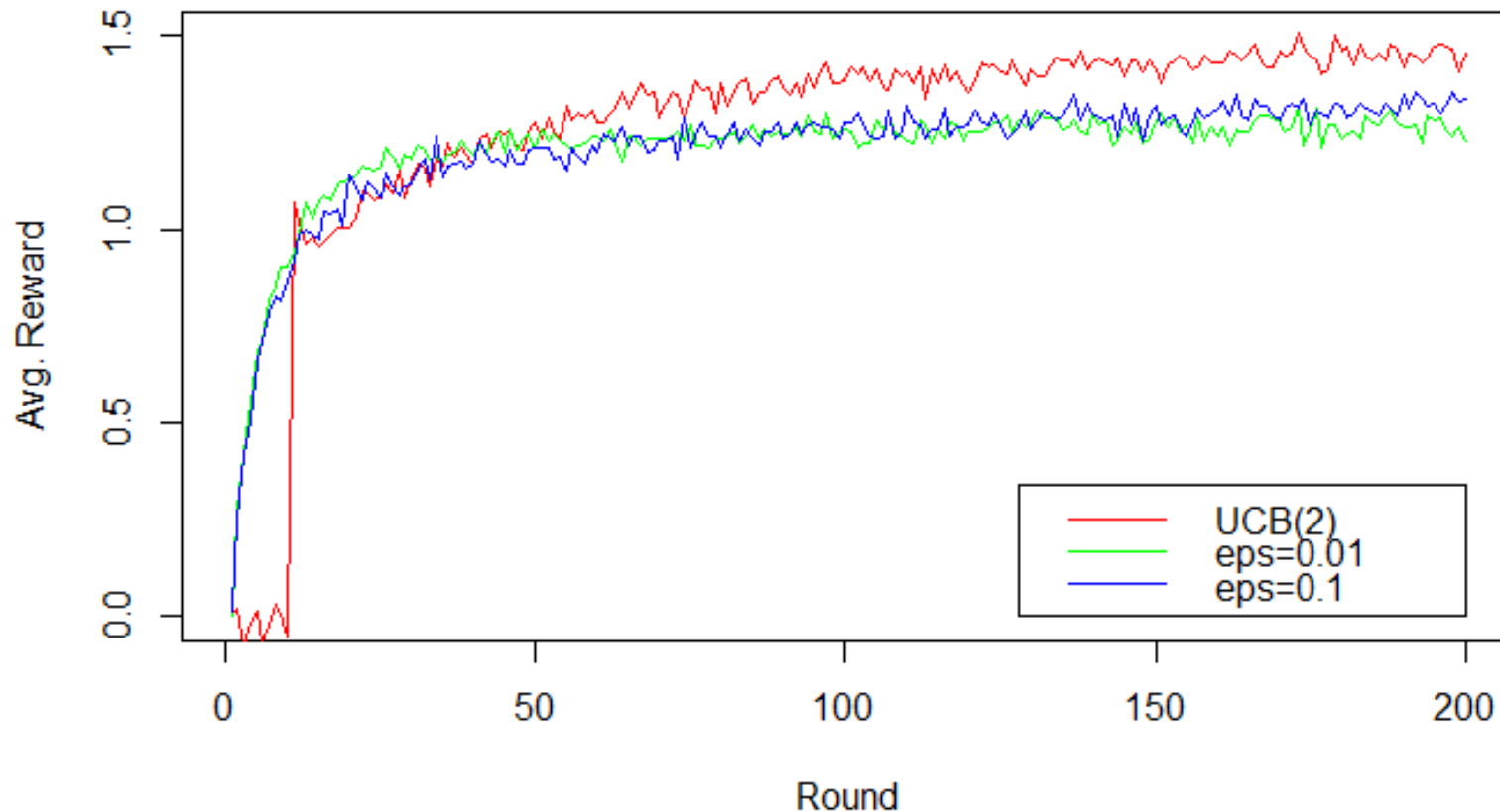
  - * "Optimism in the face of uncertainty"

$$i_t \sim \arg\max_{1 \le i \le k} Q_{t-1}(i)$$

  …tie breaking randomly

- Addresses several limitations of ε-greedy
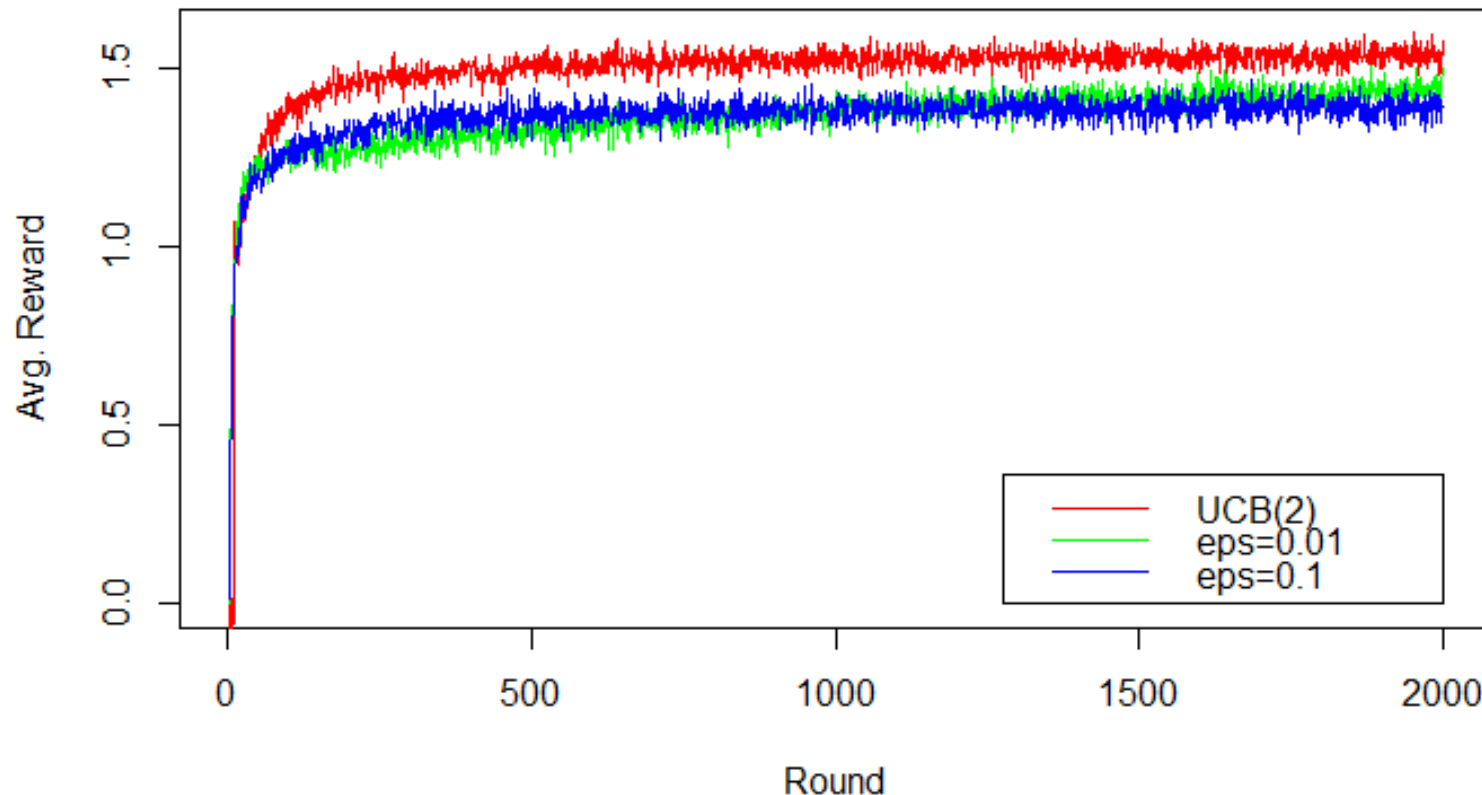- Can "pause" in a bad arm for a while, but eventually find best

13

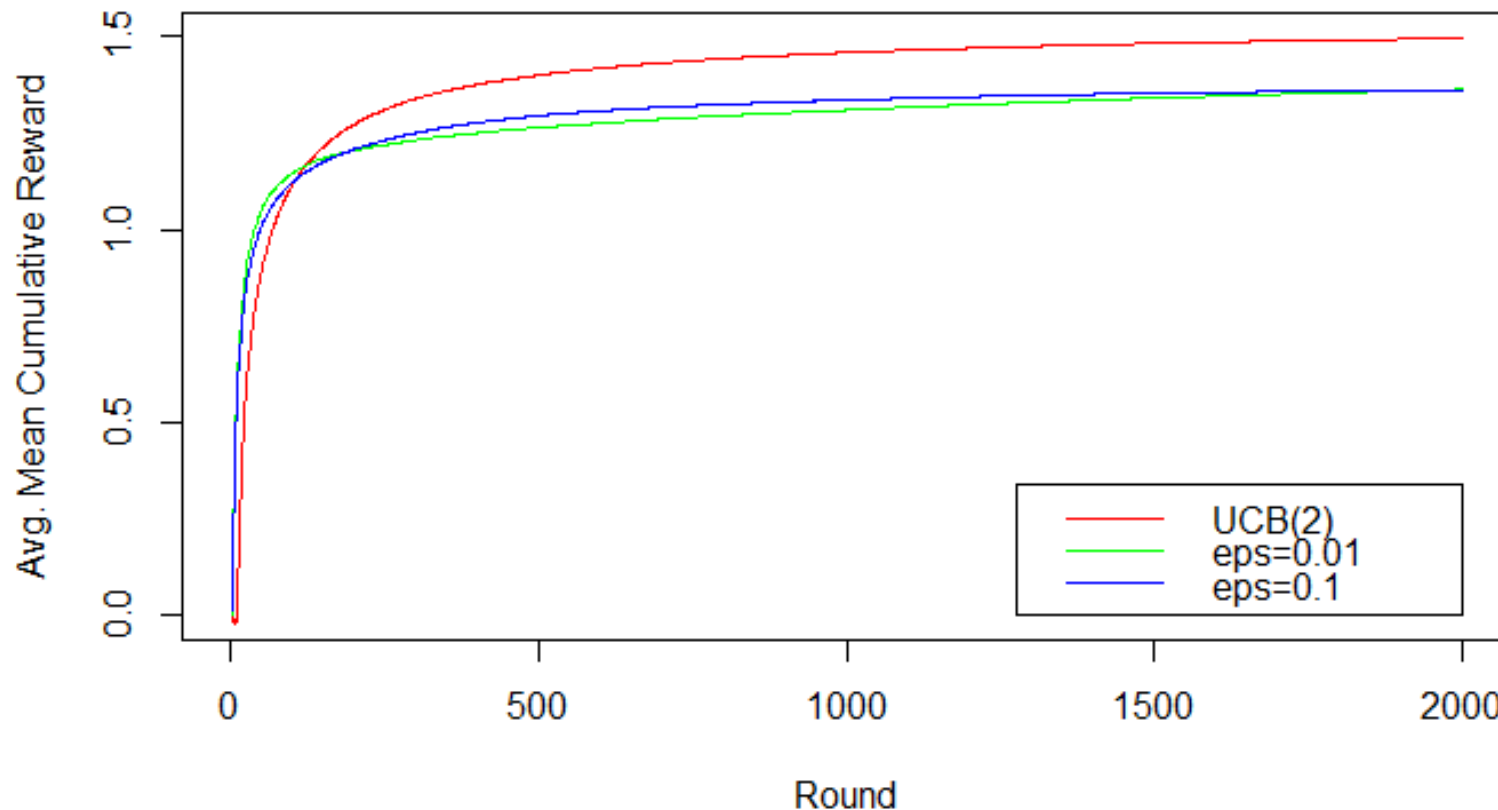# Kicking the tyres: How does UCB compare?

- UCB quickly overtakes the ε-Greedy approaches

Kicking the tyres: How does UCB compare?

- UCB quickly overtakes the ε-Greedy approaches
- Continues to outpace on per round rewards for some time

Kicking the tyres: How does UCB compare?

- UCB quickly overtakes the ε-Greedy approaches
- Continues to outpace on per round rewards for some time
- More striking when viewed as mean cumulative rewards

Notes on UCB

- Theoretical regret bounds, optimal up to multiplicative constant

  * Grows like $O(\log t)$ i.e. averaged regret goes to zero!

- Tunable $\rho > 0$ exploration hyperparam. replaces "2"

$$Q_{t-1}(i) = \begin{cases} \hat{\mu}_{t-1}(i) + \sqrt{\dfrac{\rho \log(t)}{N_{t-1}(i)}}, & \text{if } \sum_{s=1}^{t-1} 1[i_s = i] > 0 \\ Q_0, & \text{otherwise} \end{cases}$$

  * Captures different ε rates & bounded rewards outside [0,1]

- Many variations e.g. different confidence bounds

- Basis for Monte Carlo Tree Search used in AlphaGo!

MABs vs. Reinforcement Learning

- Contextual bandits introduce state
    * But don't model actions as causing state transitions
    * New state arrives "somehow"

- RL has rounds of states, actions, rewards too

- But (state,action) determines the next state
    * E.g. playing Go, moving a robot, planning logistics
    * Covered in COMP90054 as involves elements of planning

- Both RL and Contextual bandits learns value functions w regression, but RL "rolls out" predicted rewards into the future due to state transitions

This lecture

- Stochastic multi-armed bandits

  * Sequential decision making under uncertainty

  * Simplest explore-vs-exploit setting

  * (ε)-greedy, UCB

- Many applications, variations:

  * Adversarial MAB: rewards not stochastic, but *anything*

  * Contextual bandits: act based on context feature vector

  * Reinforcement learning: more general setting