

Lecture 17. Multi-armed bandits

COMP90051 Statistical Machine Learning

Guest Lecturer: Junhao Gan

(Slides made by Ben Rubinstein)



THE UNIVERSITY OF
MELBOURNE

This lecture

- **Bandit** setting vs Learning with experts
 - * Have to pick an expert (aka. arm)
 - * Observe rewards only for chosen arm
- Aka. **Sequential decision making under uncertainty**
 - * Simplest explore-vs-exploit setting
 - * Incredibly rich area with heaps of industrial applications
- **Basic algorithms**
 - * Greedy
 - * ϵ -Greedy
 - * Upper Confidence Bound (UCB)
- More: Contextual bandits, RL, ...

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.

Exploration vs. Exploitation



CCO

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
 - * Caching in databases
 - * Stochastic search in games (e.g. AlphaGo!)
 - * Adaptive A/B testing
 - *



CC0

Stochastic MAB setting

- Possible actions $\{1, \dots, k\}$ called “**arms**”
 - * Arm i has distribution P_i on bounded **rewards** with mean μ_i
 - In round $t = 1 \dots T$
 - * Play action $i_t \in \{1, \dots, k\}$ (*possibly randomly*)
 - * Receive reward $R_{i_t}(t) \sim P_{i_t}$
 - Goal: minimise cumulative **regret**
 - * $\mu^* T - \sum_{t=1}^T E[R_{i_t}(t)]$
 - ← 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

Greedy

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

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

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

- * **Exploit**, baby, exploit!

$$i_t \in \arg \max_{1 \leq i \leq k} Q_{t-1}(i)$$

- * Tie breaking randomly

- What do you expect this to do? Effect of initial Q s?

ε -Greedy

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

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

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

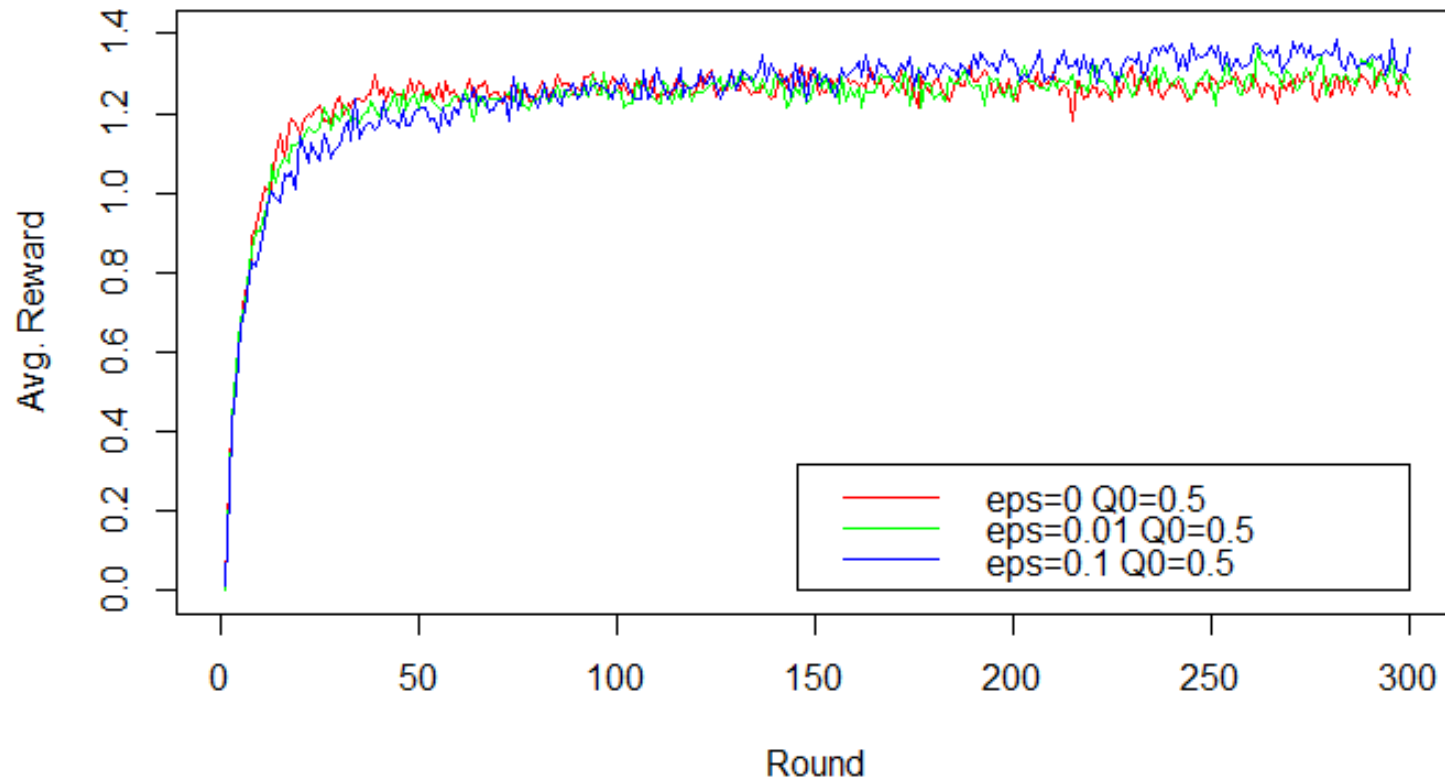
- * **Exploit**, baby exploit... probably; or possibly **explore**

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

- * Tie breaking randomly

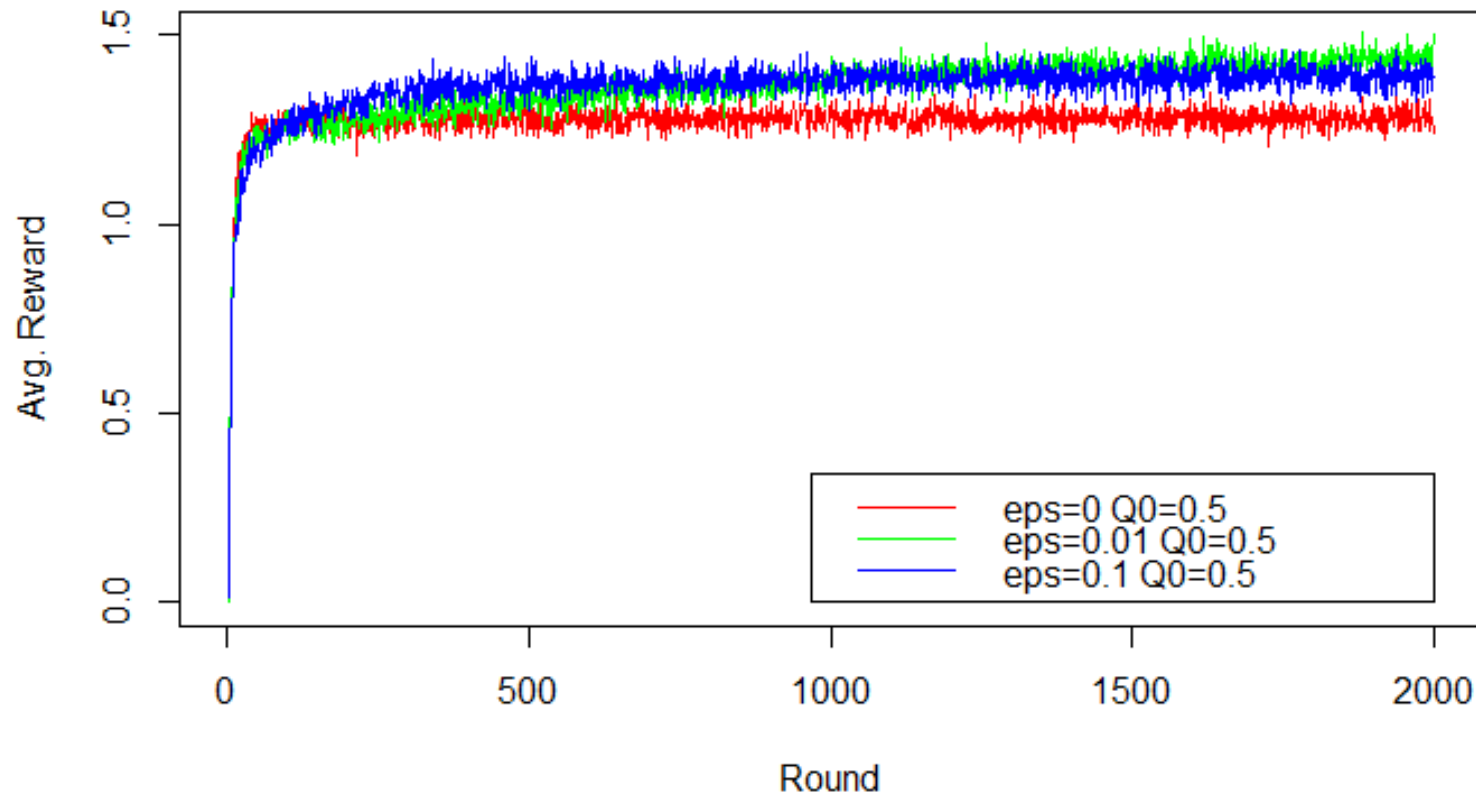
- Hyperparam. ε controls exploration vs. exploitation

Kicking the tyres



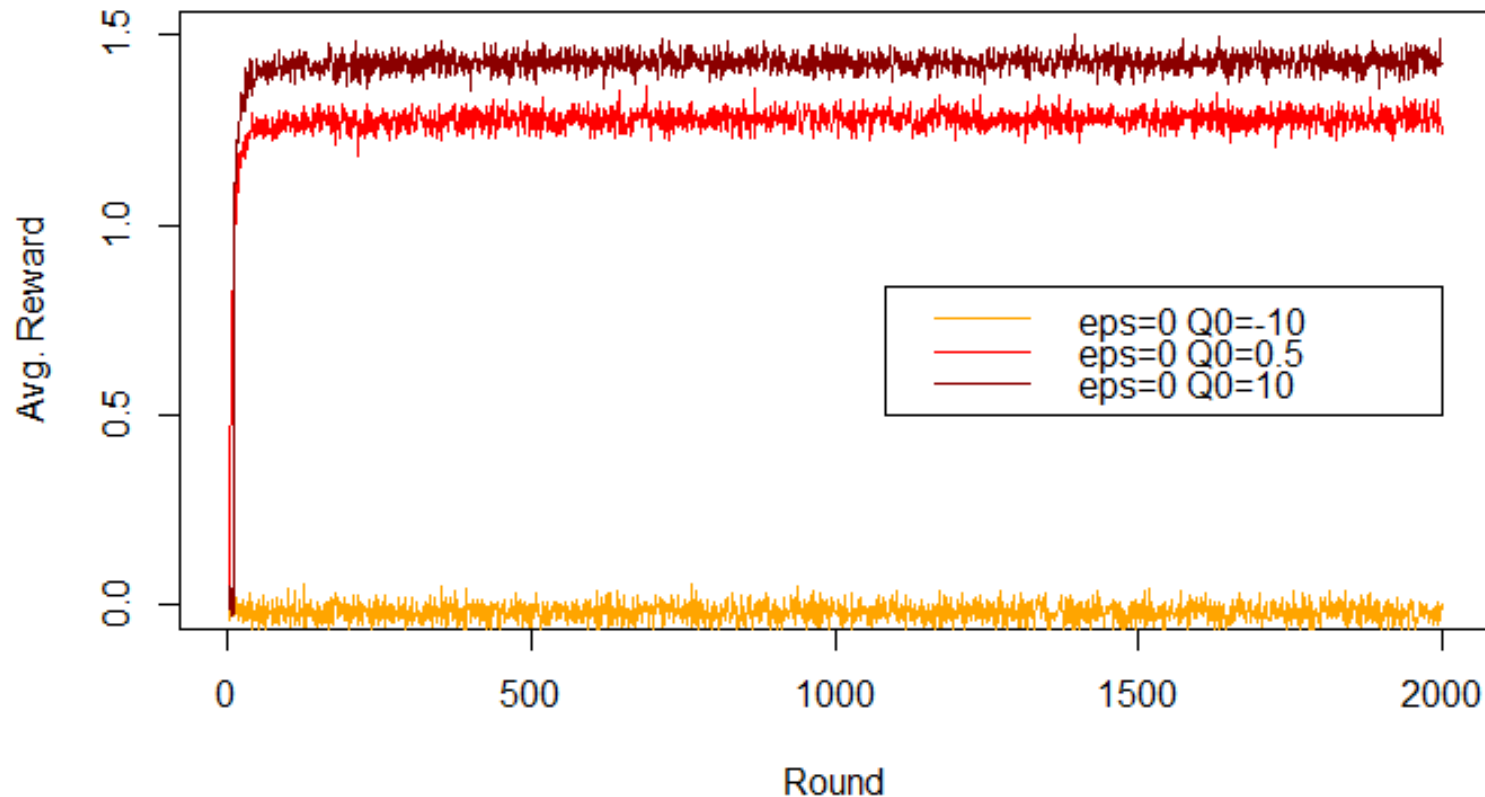
- 10-armed bandit
- Rewards $P_i = \text{Normal}(\mu_i, 1)$ with $\mu_i \sim \text{Normal}(0,1)$
- Play game for 300 rounds
- Repeat 1,000 games, plot average per-round rewards

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**)

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

Mini Summary

- Multi-armed bandit setting
 - * Simplest instance of an explore-exploit problem
 - * Greedy approaches cover exploitation fine
 - * Greedy approaches overly simplistic with exploration (if have any!)
- Compared to: learning with experts
 - * Superficial changes: Experts \rightarrow Arms; Losses \rightarrow Rewards
 - * Choose one arm (like probabilistic experts algorithm)
 - * Big difference: Only observe rewards on chosen arm

Next: A better way: optimism under uncertainty principle

Upper-Confidence Bound (UCB)

*Optimism in the face of uncertainty;
A smarter way to balance exploration-exploitation.*

(Upper) confidence interval for Q estimates

- **Theorem: Hoeffding's inequality**

- * Let R_1, \dots, R_n be i.i.d. random variables in $[0,1]$ mean μ , denote by \overline{R}_n their sample mean
- * For any $\varepsilon \in (0,1)$ with probability at least $1 - \varepsilon$

$$\mu \leq \overline{R}_n + \sqrt{\frac{\log(1/\varepsilon)}{2n}}$$

- 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 $\overline{R}_n = Q_{t-1}(i)$
 - * Critical level $\varepsilon = 1/t$ (Lai & Robbins '85), take $\varepsilon = 1/t^4$

Upper Confidence Bound (UCB) algorithm

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

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

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

$$N_{t-1}(i) = \sum_{s=1}^{t-1} 1[i_s = i] \quad \hat{\mu}_{t-1}(i) = \frac{\sum_{s=1}^{t-1} R_i(s) 1[i_s = i]}{\sum_{s=1}^{t-1} 1[i_s = i]}$$

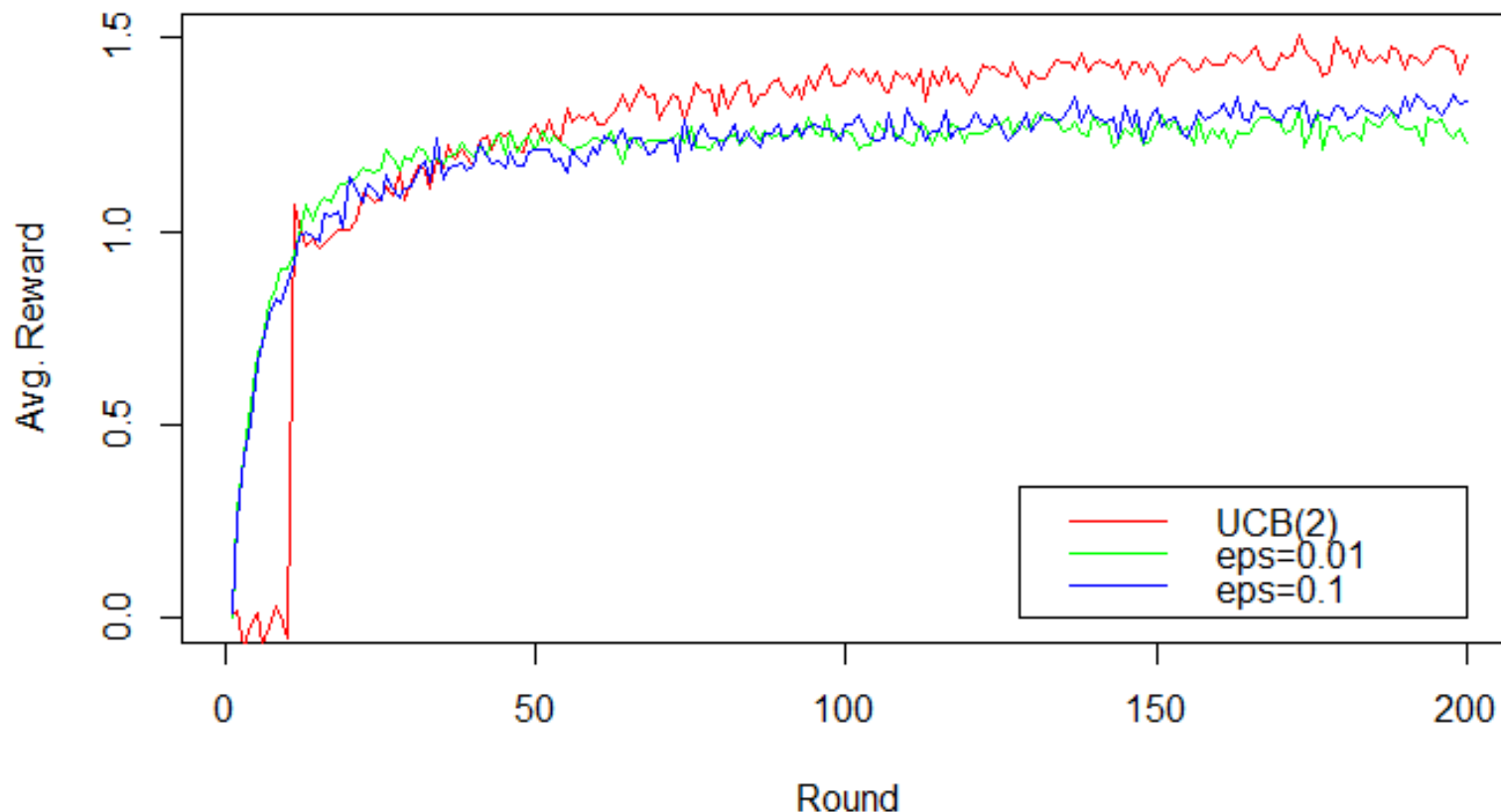
- * **“Optimism in the face of uncertainty”**

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

...tie breaking randomly

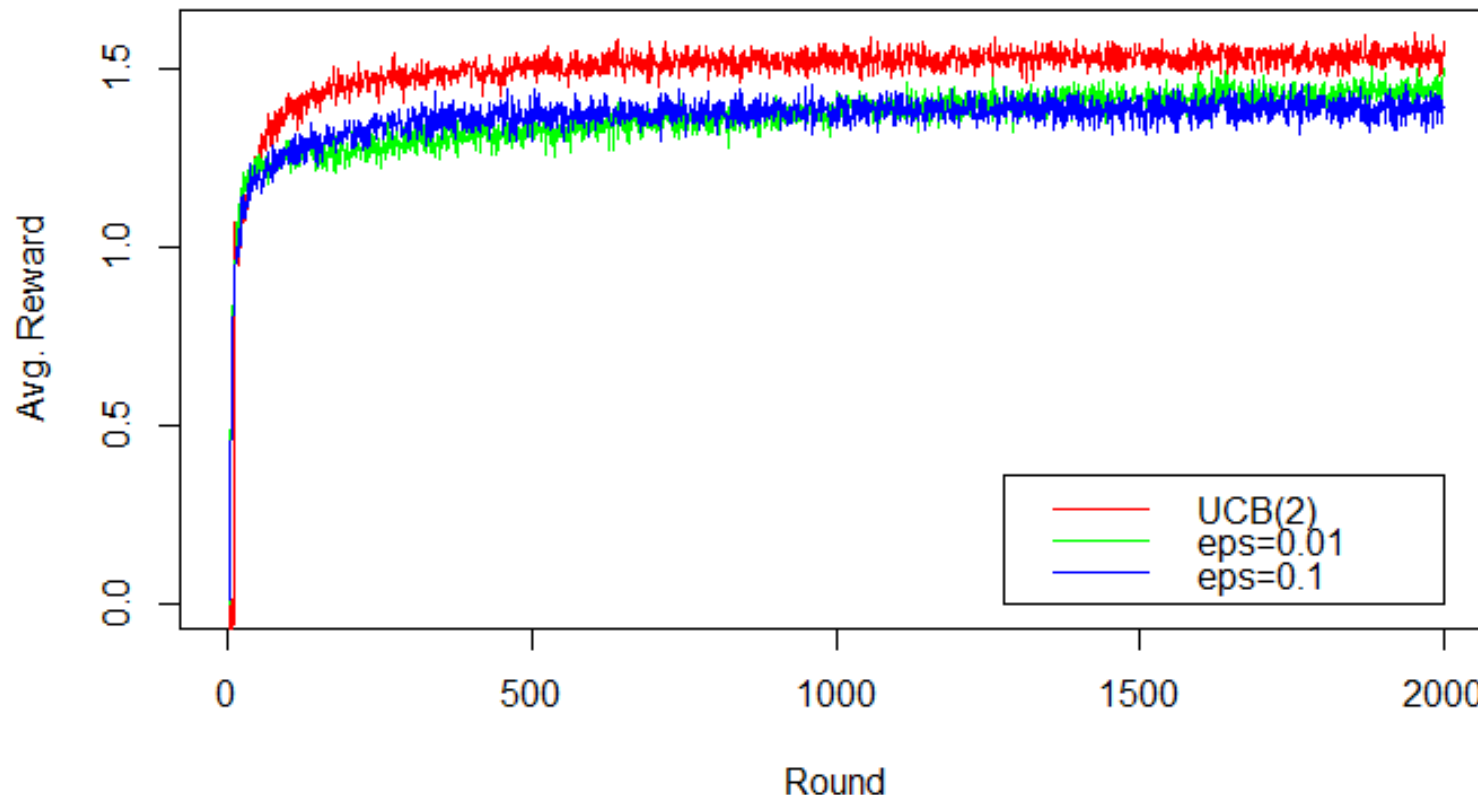
- Addresses several limitations of ε -greedy
- Can “pause” in a bad arm for a while, but eventually find best

Kicking the tyres: How does UCB compare?



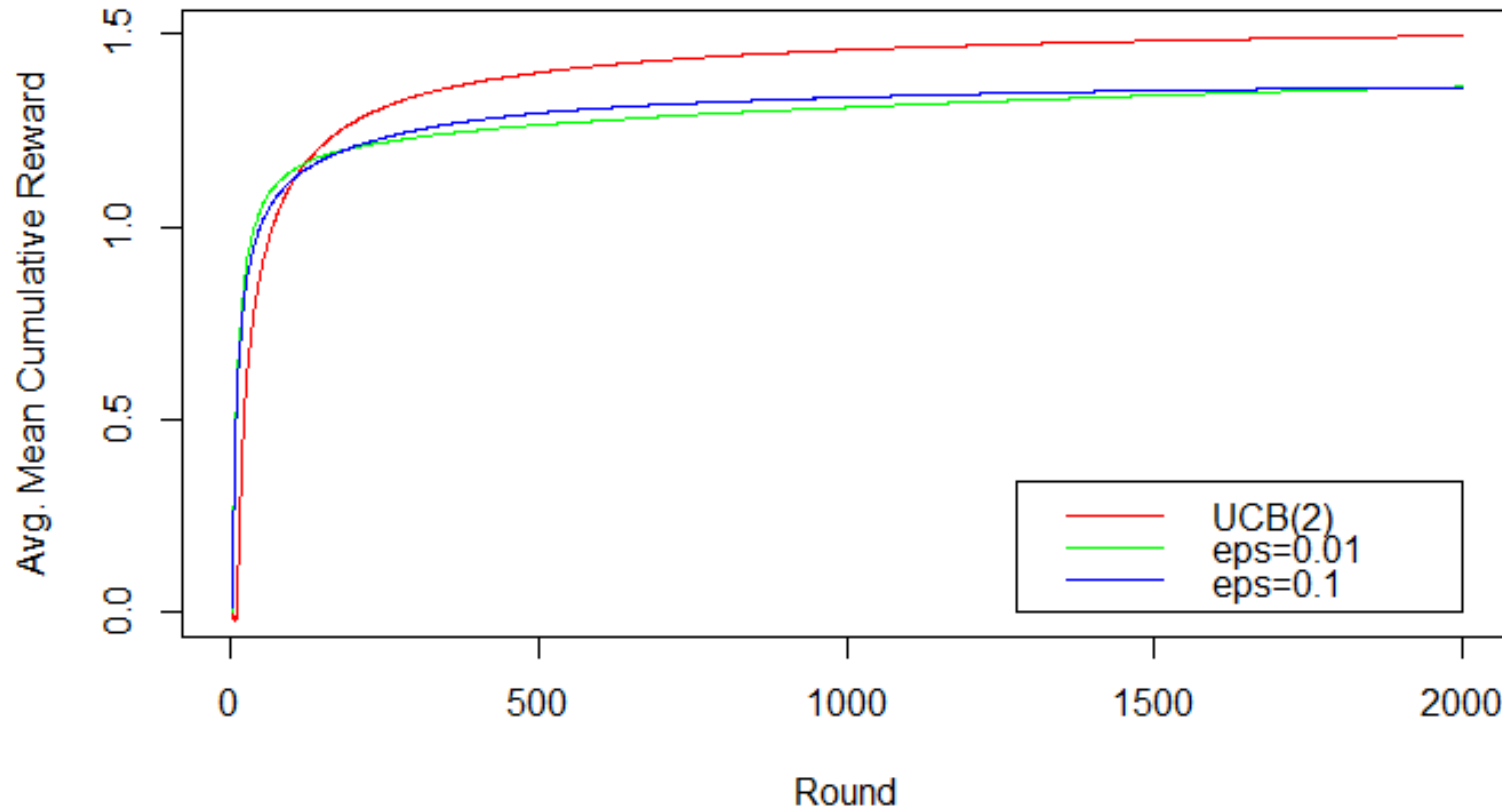
- UCB quickly overtakes the ϵ -Greedy approaches

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”

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

- * 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!

Mini Summary

- Addressing limitations of greedy
 - * Exploration blind to how good an arm is
 - * Exploration/exploitation blind to confidence of arm estimates
- Upper-confidence bound (UCB) algorithm
 - * Exploration boost: Upper-confidence bound (like PAC bound!)
 - * Example of: Optimism in the face of uncertain principle
 - * Achieves practical performance and good regret bounds

Next: Wrap-up with a few related directions

Beyond basic bandits

Adding state with contextual bandits;

State transitions/dynamics with reinforcement learning.

But wait, there's more!! Contextual bandits

- Adds concept of “state” of the world
 - * Arms' rewards now depend on state
 - * E.g. best ad depends on user and webpage
- Each round, observe arbitrary context (feature) vector representing state $\mathbf{X}_i(t)$ per arm
 - * Profile of web page visitor (state)
 - * Web page content (state)
 - * Features of a potential ad (arm)
- Reward estimation
 - * Was unconditional: $E[R_i(t)]$
 - * Now conditional: $E[R_i(t)|\mathbf{X}_i(t)]$
- A regression problem!!!

Still choose arm with maximizing UCB.

But UCB is not on a mean, but a regression prediction given context vector.

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
- Thus, RL still learns value functions w regression, but has to "roll out" predicted rewards into the future

Mini Summary

- Lecture: Stochastic multi-armed bandits
 - * Sequential decision making under uncertainty
 - * Simplest explore-vs-exploit setting
 - * (ϵ) -greedy, UCB, LinUCB
- Related directions:
 - * Contextual bandits: adding state; regression estimates rewards
 - * Reinforcement learning: introducing state transitions

Next lecture: It's Bayesian time!