COMS W4701: Artificial Intelligence

Lecture 4c: Multi-Armed Bandits

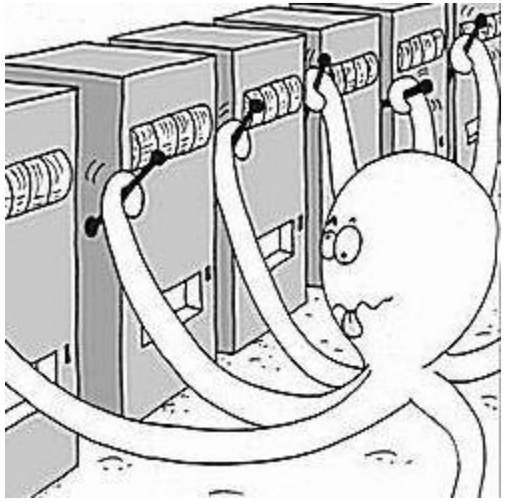
Tony Dear, Ph.D.
Department of Computer Science
School of Engineering and Applied Sciences

Today

- Multi-armed bandit problems
- Exploration vs exploitation tradeoff
- ε-greedy methods
- Upper confidence bound

Multi-Armed Bandits

- Suppose we have K slot machines with different reward distributions
- We can only learn about the machine by trying them (taking actions)
- We want to maximize the overall rewards received



- Tradeoff between **exploration** and **exploitation**
  - Gather more information or maximize best rewards so far?
  - How to determine when current knowledge is good enough?

- Applications: Resource allocation for maximizing productivity, clinical trials to explore different treatments, financial portfolio design, recommendation systems

Action Values

- Suppose action (slot machine) $a \in A$ has unknown mean reward value μ_a
- Define and update **action values** $Q_t(a)$ to estimate μ_a by trying different actions and recording the results

$$Q_t(a) = \frac{\text{sum of rewards from taking } a \text{ prior to } t}{\text{number of times taking } a \text{ prior to } t}$$

- We can initialize $Q_0(a)$ by trying each action once and recording reward
- As each $Q(a)$ better estimates μ_a, the optimal strategy would be to always pick action $\text{argmax}_a Q(a)$

Updating Action Values

- Suppose we take a and receive r, and we have N observations of a so far

$$Q_{t+1}(a) = \frac{1}{N}\left((N-1)Q_t(a) + r\right) = Q_t(a) + \frac{1}{N}\left(r - Q_t(a)\right)$$

- Update form: "new estimate" = "old estimate" + "step size" × "error"

- For **nonstationary** problems in which reward distributions change over time, we may want to give more weight to recent rewards:

$$Q_{t+1}(a) = Q_t(a) + \alpha\left(r - Q_t(a)\right)$$

Recency-Weighted Average

- For constant α, the action value update rule ends up weighting all rewards, with weights on past rewards *decaying exponentially*

$$Q_{t+1}(a) = Q_t(a) + \alpha\big(r_t - Q_t(a)\big) = \alpha r_t + (1-\alpha)Q_t(a)$$

$$= \alpha r_t + (1-\alpha)\big(\alpha r_{t-1} + (1-\alpha)Q_{t-1}(a)\big)$$

$$= \alpha r_t + (1-\alpha)\alpha r_{t-1} + (1-\alpha)^2 Q_{t-1}(a)$$

$$= \cdots = \alpha \sum_{i=1}^{t} (1-\alpha)^{t-i} r_i$$

ε-greedy Action Selection

- Action selection should balance exploitation (maximizing Q) and exploration

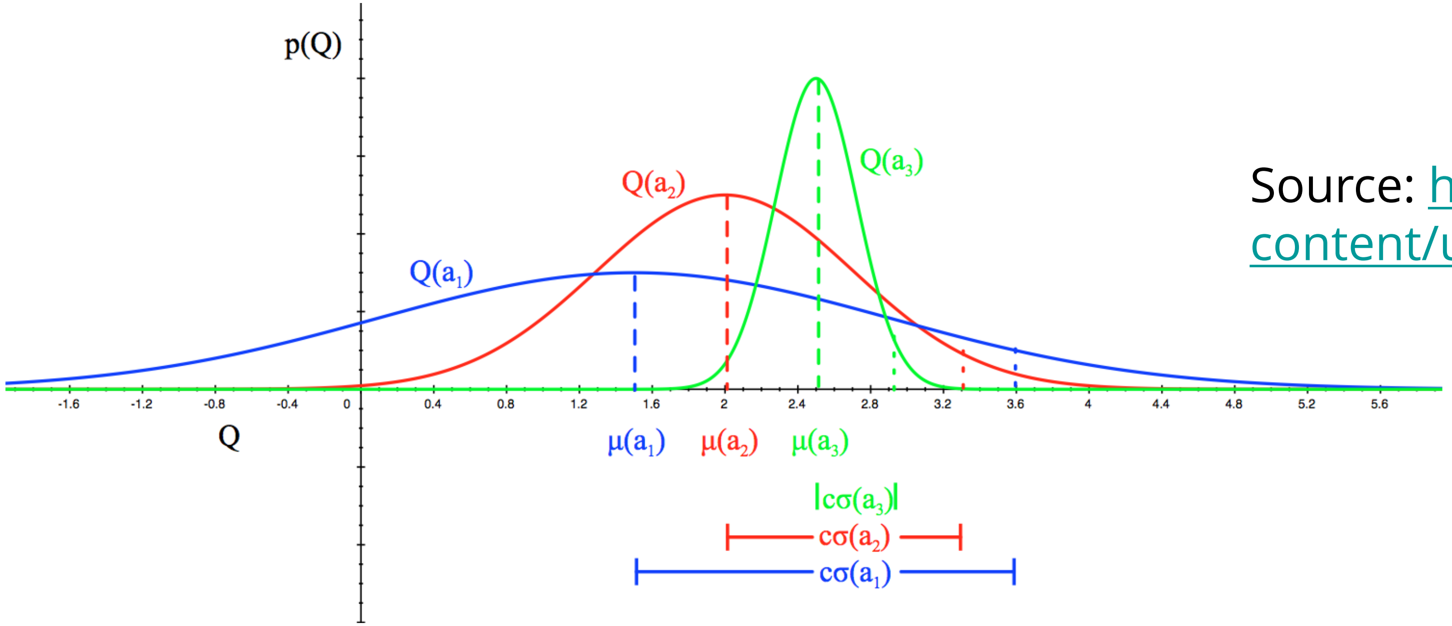
- ***ε*-greedy**: *Exploit* and select $\operatorname{argmax}_a\big(Q(a)\big)$ *most* of the time, but with small probability ε, pick a random action to *explore* instead (may also include greedy action)

- For constant ε, every action will be sampled infinitely often
- In the limit, estimates $Q_t(a)$ will converge to μ_a (though limit may be very large!)

- ***ε*-first**: Set $\varepsilon = 1$ for a fixed number of trials, then set $\varepsilon = 0$ afterward
- ***ε*-decreasing**: Set ε to high initial value (e.g., 1) and decrease it over time

Estimate Uncertainty

- ε methods only estimate value means, but not *uncertainty* (variance)
- Instead of exploring randomly, we can measure the uncertainty $U(a)$ of each action value estimate to perform "targeted" exploration



Source: https://www.davidsilver.uk/wp-content/uploads/2020/03/XX.pdf

- Exploitation-exploration tradeoff: Pick action that maximizes $Q(a) + U(a)$

Upper Confidence Bound

- **UCB1 algorithm** defines $U_t(a)$ as follows:

$$U_t(a) = c\sqrt{\frac{\ln t}{N_t(a)}}$$

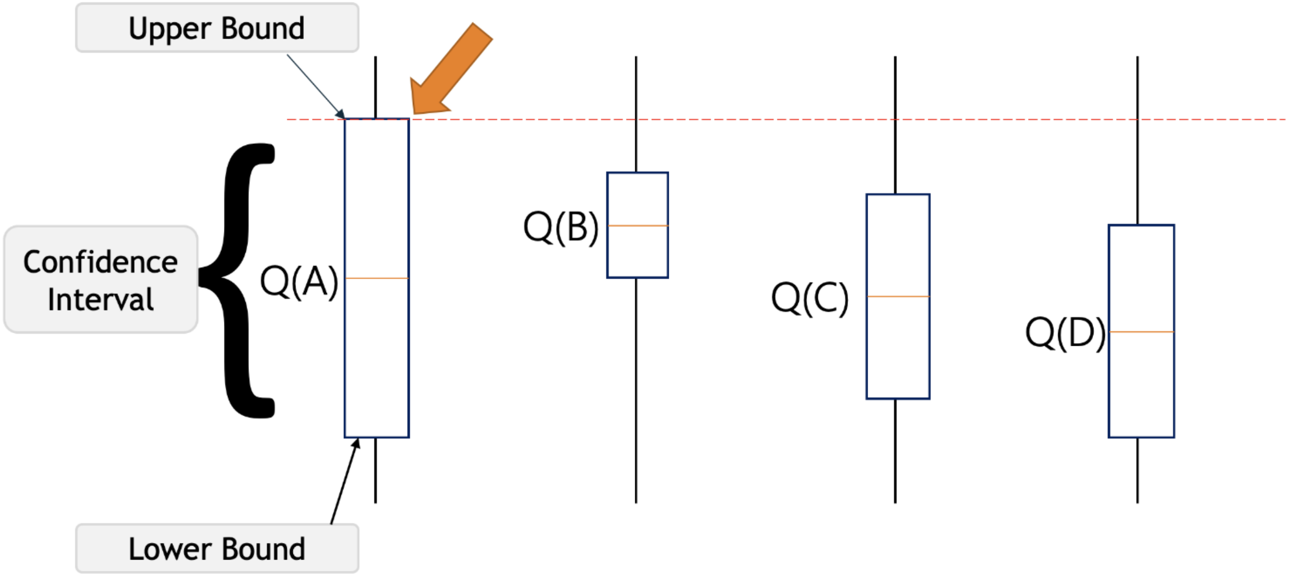
- At each step, pick action $\text{argmax}_a\big(Q(a) + U(a)\big)$
- $c \geq 0$: Tunable hyperparameter controlling exploration
- $N_t(a)$: Number of times action a taken prior to time t

- $1/\sqrt{N(a)}$ is proportional to standard deviation of $Q(a)$
- Initially large; decreases as a is repeatedly tried and we become confident
- $\ln t$ increases (slowly) over time; all actions tried infinitely often as $t \to \infty$

Optimism Under Uncertainty

- Maximizing $Q + U$ means that we are *optimistic under uncertainty*
- Higher uncertainty gives an action value a larger "bonus" for selection
- For UCB1, Hoeffding's inequality shows that the probability of the "error" being greater than $U(a)$ shrinks over time

$$\Pr[\mu_a - Q_t(a) > U_t(a)] \leq t^{-2c^2}$$



https://www.geeksforgeeks.org/upper-confidence-bound-algorithm-in-reinforcement-learning/

General Bandit Algorithm Outline

Algorithm 1: General Bandit Algorithm Procedure

Initialize, for $i = 1$ to k:

$Q_0(a_i) \leftarrow 0$

$N_0(a_i) \leftarrow 0$

for $t = 1, 2, \ldots, \infty$ **do**

$A_t \leftarrow \textsc{Choose-Action}\big(Q_{t-1}(a_1), Q_{t-1}(a_2), \ldots, Q_{t-1}(a_k)\big)$

$R_t \leftarrow \textsc{Pull-Arm}\big(A_t\big)$

$Q_t(A_t), N_t(A_t) \leftarrow \textsc{Update}\big(N_{t-1}(A_t), Q_{t-1}(A_t), R_t\big)$

end

Adapted from *Reinforcement Learning: An Introduction, 2nd ed. (Richard Sutton & Andrew Barto, 2020)*

Summary

- MAB problems model decision making in stochastic environments
- Fundamental tradeoff of exploration vs exploitation

- We can keep track of rewards and observations so far
- We can weight this info alongside uncertainty to determine our actions

- ε-greedy methods explore randomly with fixed or varying probability
- UCB1 is optimistic under uncertainty, choosing actions using a weighted balance between exploitation and exploration