

Multi-armed Bandits

Sargur N. Srihari

srihari@cedar.buffalo.edu

Topics in Multi-armed Bandits

1. The Bandit Problem
2. Bernoulli Bandits
3. Greedy Solutions
4. UCB Algorithm
5. Comparison of Algorithms

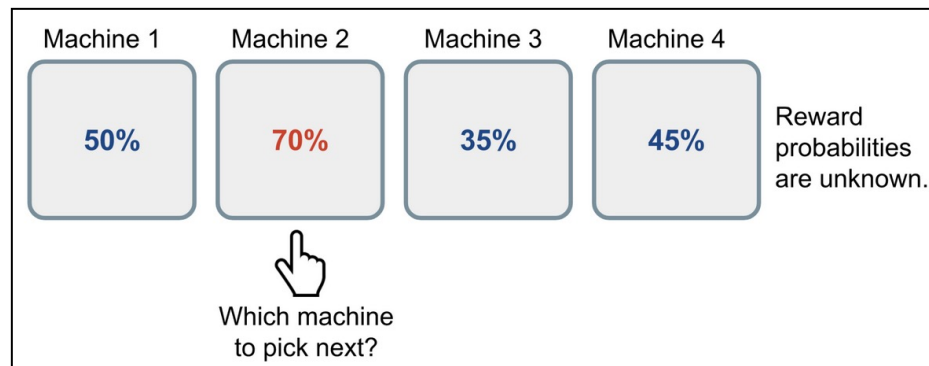
Multi-armed Bandits

- Repeated choice among k actions
 - Reward from an action-dependent distribution
- k slot machines
 - Each action is a play on one of the levers
 - Rewards for hitting one of the jackpots
 - Through action selections maximize winnings by concentrating actions on the best levers



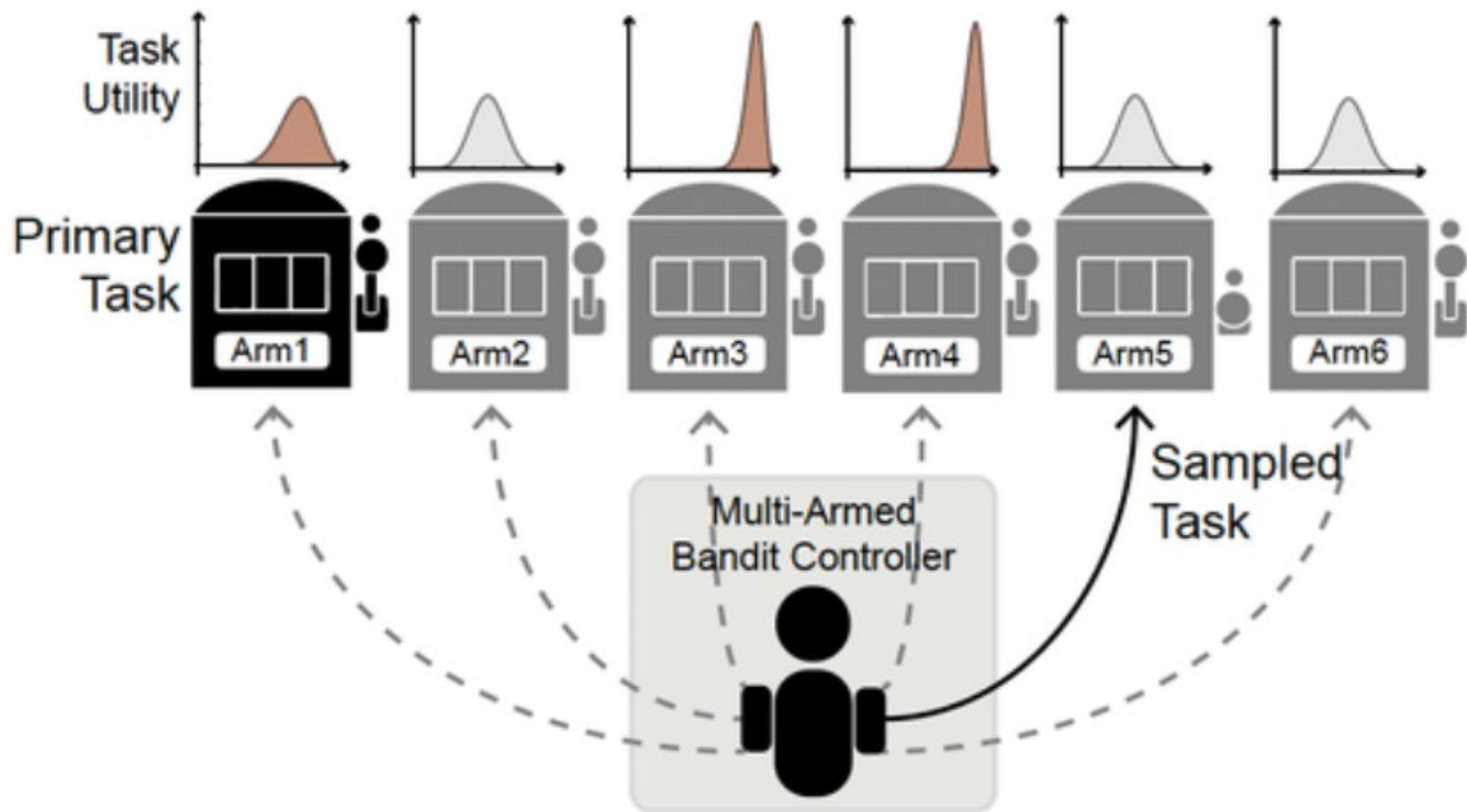
Multi-armed bandit Bernoulli reward

- Each machine provides a random reward
 - Machine-specific distribution unknown a-priori
- Binary case
 - Bernoulli distributions
 - Reward of 1 with probability p , otherwise 0



- Maximize expected total reward
 - e.g., over 1000 action selections, or time steps

Multi-armed bandit Gaussian reward



Medical domain: multi-armed bandit

- A doctor choosing between experimental treatments for a series of seriously ill patients
 - Action is a treatment
 - Reward is survival or death of the patient
- To evaluate k possible treatments for a disease
 - Incoming patients are partitioned into k groups
 - Reward: 1 if the treatment is successful else 0
 - After a while the majority of the patients can be put to the best found treatment

Definitions for k -armed bandit

- Each of k actions has an expected reward, i.e., value of that action
- Action at time step t is A_t , and the corresponding reward is R_t
- Value of action a , $q^*(a)$, is the expected reward given that a is selected:

$$q^*(a) = \mathbb{E}[R_t | A_t = a]$$

Value of Actions

- If action values are known, then trivial solution:
 - Select action with highest value
- Action values unknown, but there are estimates
 - Estimated value of action a at time t denoted $Q_t(a)$
- We would like $Q_t(a)$ to be close to $q^*(a)$

Exploitation

- If estimates of action values are maintained, then at any time step there is at least one action whose estimated value is greatest
- These are greedy actions
- When one of these actions is selected, we are *exploiting* current knowledge of values of actions

Exploration

- Selecting a nongreedy action is *exploring*
 - Enables improving estimate of nongreedy action value
- Exploitation maximizes expected reward on one step, but exploration may produce greater total reward in the long run

Action-Value Methods

- Value of action a is mean value of reward

$$Q_t(a) \stackrel{\cdot}{=} \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_i=a}},$$

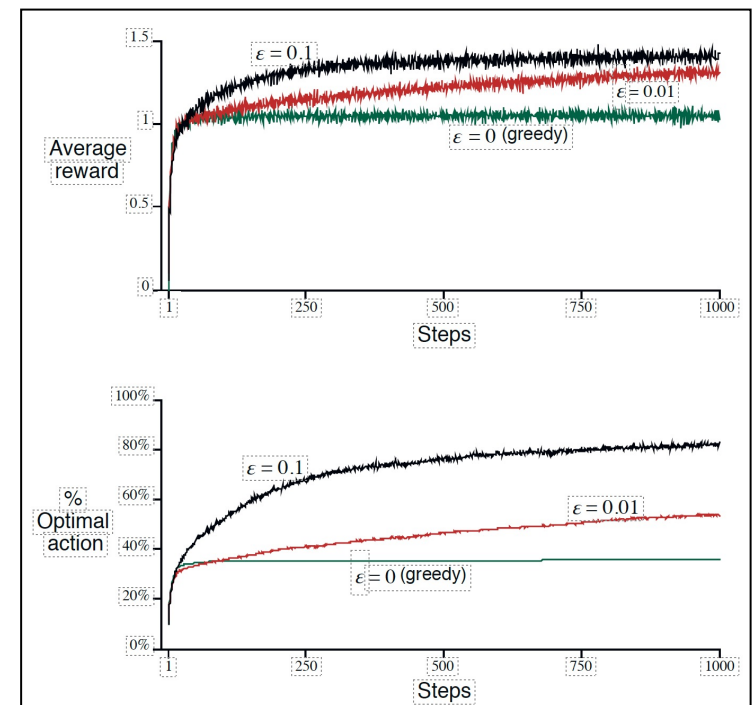
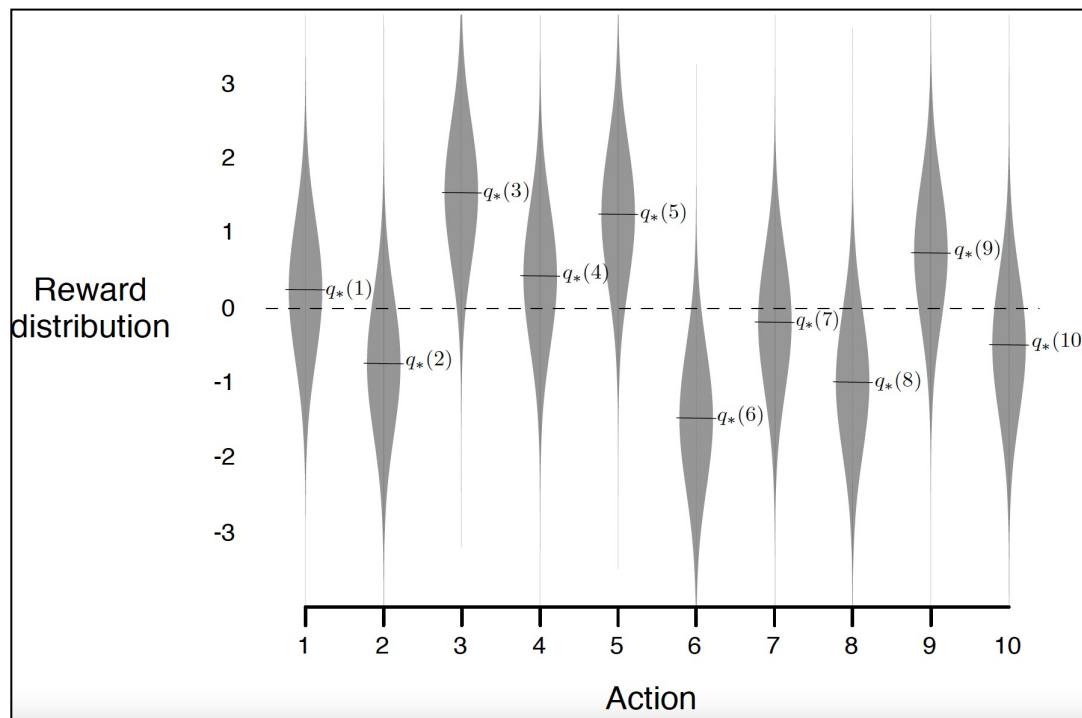
- Greedy selection method is:

$$A_t \stackrel{\cdot}{=} \underset{a}{\operatorname{argmax}} Q_t(a),$$

- ϵ -Greedy selection method is:
 - Greedy most of the time
 - With probability ϵ select another action
 - With equal probability
 - Since all actions sampled infinitely
 - $Q_t(a)$ converges to $q^*(a)$

Greedy vs ϵ -Greedy: 10-armed bandit

- 2000 randomly generated 10-arm bandits
 - Action values $q^*(a)$, $a = 1, \dots, 10$ selected from $N(0,1)$
 - Gaussian rewards selected from $N(q^*(A_t), 1)$
 - Violin plots: mean $q^*(n)$, same std. dev.



Efficient Implementation

- R_i : probabilistic reward at i^{th} selection of action
- Q_n : average of action after $n-1$ selections

$$Q_n \triangleq \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$$

– Incremental formula for updating averages

$$\begin{aligned} Q_{n+1} &= \frac{1}{n} \sum_{i=1}^n R_i \\ &= \frac{1}{n} \left(R_n + \sum_{i=1}^{n-1} R_i \right) \\ &= \frac{1}{n} \left(R_n + (n-1) \frac{1}{n-1} \sum_{i=1}^{n-1} R_i \right) \\ &= \frac{1}{n} (R_n + (n-1)Q_n) \\ &= \frac{1}{n} (R_n + nQ_n - Q_n) \\ &= Q_n + \frac{1}{n} [R_n - Q_n], \end{aligned}$$

A simple bandit algorithm

Initialize, for $a = 1$ to k :

$$Q(a) \leftarrow 0$$

$$N(a) \leftarrow 0$$

Loop forever:

$$A \leftarrow \begin{cases} \arg \max_a Q(a) & \text{with probability } 1 - \varepsilon \quad (\text{breaking ties randomly}) \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$$R \leftarrow \text{bandit}(A)$$

$$N(A) \leftarrow N(A) + 1$$

bandit(a) is a function that takes an action and returns a reward

$$Q(A) \leftarrow Q(A) + \frac{1}{N(A)} [R - Q(A)]$$

• General formula

New Estimate \leftarrow *Old Estimate* + *Stepsize* (*Target* – *New Estimate*)

Target is a desirable direction to move, may be noisy, here it is the n^{th} reward

Non-stationary Problem

- Stationary: reward probabilities same over time
- Non-stationary: more weight to recent than past rewards
 - Step-size parameter $\alpha \in (0, 1]$ is constant
 - Q_{n+1} : weighted average of past rewards and initial estimate Q_1

$$\begin{aligned}
 Q_{n+1} &= Q_n + \alpha [R_n - Q_n] \\
 &= \alpha R_n + (1 - \alpha) Q_n \\
 &= \alpha R_n + (1 - \alpha) [\alpha R_{n-1} + (1 - \alpha) Q_{n-1}] \\
 &= \alpha R_n + (1 - \alpha) \alpha R_{n-1} + (1 - \alpha)^2 Q_{n-1} \\
 &= \alpha R_n + (1 - \alpha) \alpha R_{n-1} + (1 - \alpha)^2 \alpha R_{n-2} + \\
 &\quad \dots + (1 - \alpha)^{n-1} \alpha R_1 + (1 - \alpha)^n Q_1 \\
 &= (1 - \alpha)^n Q_1 + \sum_{i=1}^n \alpha (1 - \alpha)^{n-i} R_i.
 \end{aligned}$$

- For convergence $\sum_{n=1}^{\infty} \alpha_n(a) = \infty$ and $\sum_{n=1}^{\infty} \alpha_n^2(a) < \infty$.

Upper Confidence Bound Action

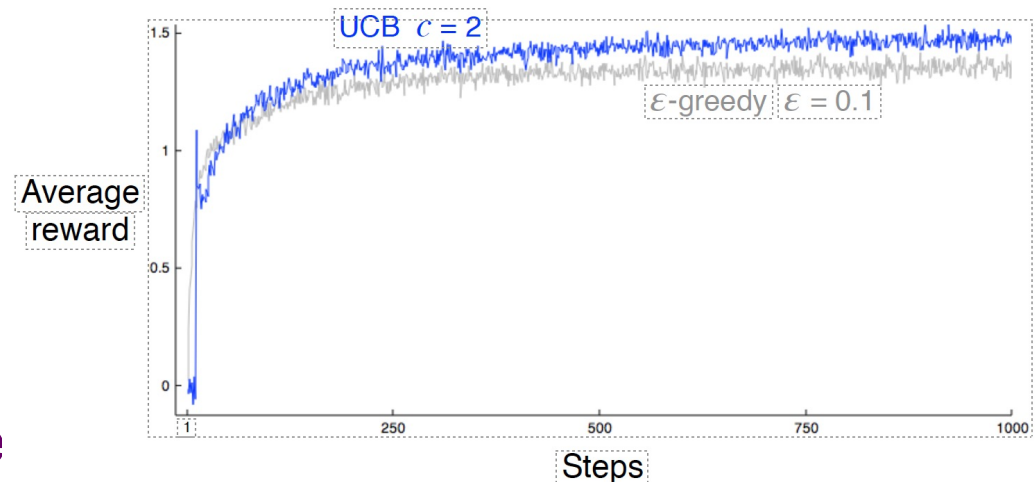
- Select non-greedy actions for being optimal

$$A_t \stackrel{\cdot}{=} \arg \max_a \left[Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$$

- $Q_t(a)$: Value of action a at t
- $N_t(a)$: no. of times a selected, prior to t
- c controls level of exploration

– Takes into account

1. Exploitation: Closene being maximal, $Q_t(a)$
2. Exploration: Uncertainties in estimates



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

Gradient Bandit Algorithms

- Numerical preference for action a denoted $H_t(a)$
- Probability of taking action a at time t

$$\Pr\{A_t = a\} \doteq \frac{e^{H_t(a)}}{\sum_{b=1}^k e^{H_t(b)}} \doteq \pi_t(a)$$

- Stochastic Gradient Ascent Algorithm

$$\begin{aligned} H_{t+1}(A_t) &\doteq H_t(A_t) + \alpha(R_t - \bar{R}_t)(1 - \pi_t(A_t)), & \text{and} \\ H_{t+1}(a) &\doteq H_t(a) - \alpha(R_t - \bar{R}_t)\pi_t(a), & \text{for all } a \neq A_t, \end{aligned}$$

- In gradient ascent, preference $H_t(a)$ is incremented proportional to its effect on performance:

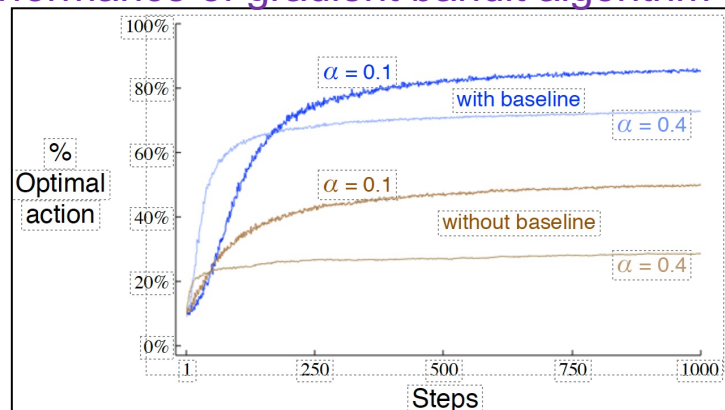
$$H_{t+1}(a) \doteq H_t(a) + \alpha \frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)}$$

Ave. performance of gradient bandit algorithm

- Where performance measure

- Is expected reward

$$\mathbb{E}[R_t] = \sum \pi_t(x) q_*(x),$$



Comparison of Bandit Algorithms

- 10-arm bandit performance
- As a function of parameters
- Upper Confidence Bound method performs best

