Multi-Armed Bandits

Credit: David Silver

Google DeepMind

Presenter: Tianlu Wang

Outline

1. Introduction
   - Exploration vs. Exploitation Dilemma
   - How to do Exploration
2. Multi-Armed Bandits
   - The Multi-Armed Bandit
   - Regret
   - Greedy and ϵ-greedy algorithms
   - Lower Bound
   - Upper Confidence Bound

Outline

Exploration vs. Exploitation Dilemma

- Online decision-making involves a fundamental choice:
  - Exploitation Make the best decision given current information
  - Exploration Gather more information
- The best long-term strategy may involve short-term sacrifices
- Gather enough information to make the best overall decisions
- Examples:
  - Online Banner Advertisements:
    Exploitation Show the most successful advert;
    Exploration Show a different advert
  - Game Playing:
    Exploitation Play the move you believe is best;
    Exploration Play an experimental move

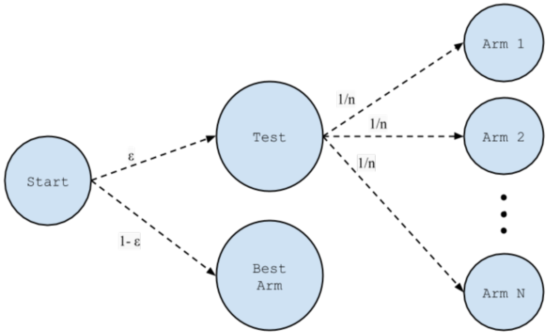
Outline

How to do Exploration

- Random exploration:
  - ϵ-greedy: Pull a random chosen arm a fraction ϵ of the time and the other $1 - \epsilon$ time, pull the arm which estimated to be the most profitable.(Devote a fraction ϵ of resources to testing)

How to do Exploration

- Random exploration:
  - ϵ-greedy: Pull a random chosen arm a fraction ϵ of the time and the other $1 - \epsilon$ time, pull the arm which estimated to be the most profitable.(Devote a fraction ϵ of resources to testing)



How to do Exploration

- Optimism in the face of uncertainty:
  - Estimate uncertainty on value
  - Prefer to explore states/actions with highest uncertainty

How to do Exploration

- Information state space(most correct but computationally difficult):
  - Consider agent's information as part of its state
  - Look ahead to see how information helps reward

Outline

1. Introduction
   - Exploration vs. Exploitation Dilemma
   - How to do Exploration
2. Multi-Armed Bandits
   - The Multi-Armed Bandit
   - Regret
   - Greedy and ϵ-greedy algorithms
   - Lower Bound
   - Upper Confidence Bound

The Multi-Armed Bandit

- A multi-armed bandit is a tuple $< \mathcal{A}, \mathcal{R} >$
- $\mathcal{A}$ is a known set of m actions(or "arms")
- $\mathcal{R}^a(r) = \mathbb{P}[r|a]$ is an unknown probability distribution over rewards
- At each step t the agent selects an action $a_t \in \mathcal{A}$
- The environment generates a reward $r_t \in \mathcal{R}^{a_t}$
- The goal is to maximise cumulative reward $\Sigma_{\tau=1}^{t} r_\tau$

Outline

Regret

- The action-value is the mean reward for action a:

$$Q(a) = \mathbb{E}[r|a]$$

- The optimal value V^* is

$$V^* = Q(a^*) = \max_{a \in \mathcal{A}} Q(a)$$

- The regret is the opportunity loss for one step:

$$l_t = \mathbb{E}[V^* - Q(a_t)]$$

- The total regret is the total opportunity loss

$$L_t = \mathbb{E}[\Sigma_{\tau=1}^{t} V^* - Q(a_\tau)]$$

- Maximise cumulative reward $\equiv$ minimise total regret

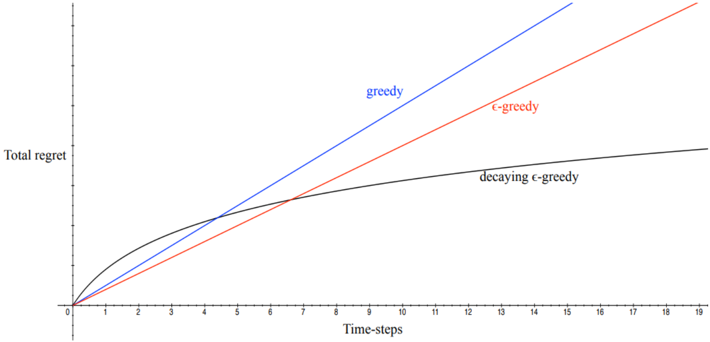
Counting Regret

- The count $N_t(a)$ is expected number of selections for action a
- The gap Δ_a is the difference in value between action a and optimal action a^*, $\Delta = V^* - Q(a)$
- Regret is a function of gaps and the counts:

$$\begin{aligned} L_t &= \mathbb{E}[\Sigma_{\tau=1}^{t} V^* - Q(a_\tau)] \\ &= \Sigma_{a \in \mathcal{A}} \mathbb{E}[N_t(a)](V^* - Q(a)) \\ &= \Sigma_{a \in \mathcal{A}} \mathbb{E}[N_t(a)]\Delta_a \end{aligned} \quad (1)$$

- A good algorithm ensures small counts for large gaps
- Problem: gaps are not known.

Linear or Sublinear regret



- If an algorithm forever explores it will have linear total regret
- If an algorithm never explores it will have linear total regret
- Is it possible to achieve sublinear total regret?

Outline

Greedy Algorithm

- We consider algorithms that estimate $\hat{Q}_t(a) \approx Q(a)$
- Estimate the value of each action by Monte-Carlo evaluation:

$$\hat{Q}_t(a) = \frac{1}{N_t(a)} \Sigma_{t=1}^{T} r_t \mathbf{1}(a_t = a)$$

- The greedy algorithm selects action with highest value:

$$a_t^* = argmax_{a \in \mathcal{A}} \hat{Q}_t(a)$$

- Greedy can lock onto a suboptimal action forever
- Greedy has linear total regret

ϵ-Greedy Algorithm

- The ϵ-greedy algorithm continues to explore forever
  - With probability $1 - \epsilon$ select $a = argmax_{a \in \mathcal{A}} \hat{Q}(a)$
  - With probability ϵ select a random action
- Constant ϵ ensures minimum regret:

$$l_t \geq \frac{\epsilon}{\mathcal{A}} \Sigma_{a \in \mathcal{A}} \Delta_a$$

- ϵ-Greedy has linear total regret

Optimistic Initialisation

- Initialise $Q(a)$ to high value
- Update action value by incremental Monte-Carlo evaluation:

$$\hat{Q}_t(a_t) = \hat{Q}_{t-1} + \frac{1}{N_t(a_t)}(r_t - \hat{Q}_{t-1})$$

- Encourages systematic exploration early on
- But can still lock onto suboptimal action
- greedy(ϵ-greedy) + optimistic initialisation has linear total regret

Decaying ϵ_t-Greedy Algorithm

- Pick a decay schedule for ϵ_1, ϵ_2, ...
- Consider the following schedule:

$$c > 0$$
$$d = min_{a|\Delta_a > 0} \Delta_a$$
$$\epsilon_t = min\{1, \frac{c|\mathcal{A}|}{d^2 t}\}$$

- Logarithmic asymptotic total regret
- Requires advance knowledge of gaps
- Goal: find an algorithm with sublinear regret for any multi-armed bandit (**without knowledge of** $\mathcal{R}$)

Outline

1. Introduction
   - Exploration vs. Exploitation Dilemma
   - How to do Exploration
2. Multi-Armed Bandits
   - The Multi-Armed Bandit
   - Regret
   - Greedy and ϵ-greedy algorithms
   - Lower Bound
   - Upper Confidence Bound

Lower Bound

- The performance of any algorithm is determined by **similarity** between optimal arm and other arms
- Hard problems have similar-looking arms with different means
- This is described formally by the gap Δ_a and the similarity in distributions $KL(\mathcal{R}^a || \mathcal{R}^{a*})$

Theorem (Lai and Robbins)

Asymptotic total regret is at least logarithmic in number of steps

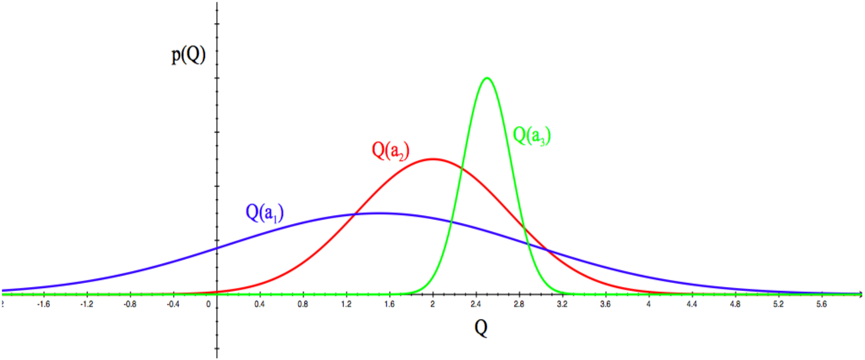
$$\lim_{t\to\infty} L_t \geq \log t \Sigma_{a|\Delta_a>0} \frac{\Delta_a}{KL(\mathcal{R}^a || \mathcal{R}^{a*})}$$

Outline

1. Introduction
   - Exploration vs. Exploitation Dilemma
   - How to do Exploration

2. Multi-Armed Bandits
   - The Multi-Armed Bandit
   - Regret
   - Greedy and ϵ-greedy algorithms
   - Lower Bound
   - Upper Confidence Bound

Upper Confidence Bound



- Which action should we pick?
- The more uncertain we are about an action-value
- The more important it is to explore that action
- It could turn out to be the best action

Upper Confidence Bound

- Estimate an upper confidence $\hat{U}_t(a)$ for each action value
- Such that $Q(a) \leq \hat{Q}_t(a) + \hat{U}_t(a)$ with high probability
- This depends on the number of times $N(a)$ has been selected
  - Small $N_t(a) \Rightarrow large \hat{U}_t(a)$(estimated value is uncertain)
  - Large $N_t(a) \Rightarrow small \hat{U}_t(a)$(estimated value is accurate)
- Select action maximising Upper Confidence Bound (UCB)

$$a_t = argmax_{a \in \mathcal{A}} \hat{Q}_t(a) + \hat{U}_t(a)$$

Upper Confidence Bound

Theorem (Hoeffding's Inequality)

Let X_1, ..., X_t be i.i.d. random variables in [0,1], and let $\bar{X}_t = \frac{1}{\tau}\Sigma_{\tau=1}^{t} X_\tau$ be the sample mean. Then

$$\mathbb{P}[\mathbb{E}[X] > \bar{X}_t + u] \leq e^{-2tu^2}$$

- We will apply Hoeffdings Inequality to rewards of the bandit
- conditioned on selecting action a

$$\mathbb{P}[Q(a) > \hat{Q}_t(a) + U_t(a)] \leq e^{-2N_t(a)U_t(a)^2}$$

Calculating Upper Confidence Bounds

- Pick a probability p that true value exceeds UCB
- Now solve for $U_t(a)$:

$$e^{-2N_t(a)U_t(a)^2} = p$$

$$U_t(a) = \sqrt{\frac{-\log p}{2N_t(a)}}$$

- Reduce p as we observe more rewards, e.g. $p = t^{-4}$
- Ensures we select optimal action as $t \to \infty$

UCB1

- This leads to the UCB1 algorithm

$$a_t = argmax_{a \in \mathcal{A}} Q(a) + \sqrt{\frac{2 \log t}{N_t(a)}}$$

Theorem

The UCB algorithm achieves logarithmic asymptotic total regret

$$\lim_{t \to \infty} L_t \leq 8 \log t \Sigma_{a|\Delta_a > 0} \Delta_a$$