Lecture Notes:

Bandits

Course: Reinforcement Learning,
Bachelor AI, Leiden University

Written by: Thomas Moerland

1 Definition

A bandit is defined by the tuple

$$\langle \mathcal{A}, p(r|a) \rangle,$$

where

- $\mathcal{A}$ is a set of discrete actions ('arms').
- $p(r|a)$ is a conditional probability distribution, mapping each action to a distribution over the possible rewards (either discrete or continuous).

Policy $\pi(a)$ is a probability distribution over the discrete action space.

- Explicit policy: directly stores the probabilities in $\pi(a)$.

  Example:

$\pi(a=1)$	$\pi(a=2)$	$\pi(a=3)$	$\pi(a=4)$
0.2	0.7	0.0	0.1

- Implicit policy: stores other quantities, and computes $\pi(a)$ from these upon action selection.

  Example:

$Q(a=1)$	$Q(a=2)$	$Q(a=3)$	$Q(a=4)$
1.2	0.3	-2.4	3.5

  and

  $$\pi = f(Q(a))$$

2 Objective

At each timestep t, we sample an action $a_t \in \mathcal{A}$, and receive a reward $r_t \sim p(r|a_t)$.

Algorithm 1: Bandit algorithm pseudocode.

Input: Maximum number of timesteps T, often also an exploration parameter.

Initialization: Initialize policy $\pi(a)$

for $t = 1...T$ **do**

    $a_t \sim \pi(a)$ `/* Sample from policy */`

    $r_t \sim p(r|a_t)$ `/* Observe reward */`

    Update π based on (a_t, r_t)

end

Values Define the action value $Q(a)$ as the expected pay-off of an arm:

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

The best possible average pay-off in the problem is

$$V^\star = \max_a Q(a)$$

Our goal is to find the policy that maximizes the cumulative sum of reward J that we obtain over some horizon T:

$$J_T(\pi) = \mathbb{E}_{a_t \sim \pi(a), r_t \sim p(r|a_t)}[\sum_{t=1}^{T} r_t]$$

$$\pi^\star = \arg\max_\pi J_T(\pi)$$

3 Bandit algorithm choices

For each bandit algorithm, we typically need to decide on three aspects:

- The initial estimates of $\hat{Q}(a)$.
- The policy, i.e., the way to select actions, which should balance exploration and exploitation.
- The update, i.e., the way we update our estimates of $\hat{Q}(a)$ based on the observed reward after trying a particular action.

3.1 Initialization of mean

- Realistic

$$Q(a) = 0 \quad \forall \quad a \in \mathcal{A}$$

- Optimistic

$$Q(a) = \psi \quad \forall \quad a \in \mathcal{A}$$

for some initial value $\psi \in \mathbb{R}$ (a hyperparameter that should be tuned per problem).

3.2 Policy

- Greedy policy (with optimistic intialization):

$$\pi_{\text{greedy}}(a) = f(Q) = \begin{cases} 1, & \text{if } a = \arg\max_{b \in \mathcal{A}} Q(b) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

which we also write as

$$\pi_{\text{greedy}} = \arg\max_{b \in \mathcal{A}} Q(b)$$

(which returns an action instead of a probability of an action).

- ϵ-greedy policy:

$$\pi_{\epsilon-\text{greedy}}(a) = f(Q, \epsilon) = \begin{cases} 1-\epsilon, & \text{if } a = \arg\max_{b \in \mathcal{A}} Q(b) \\ \frac{\epsilon}{|\mathcal{A}|-1}, & \text{otherwise} \end{cases} \tag{2}$$

  where $\epsilon \in [0, 1]$ scales the amount of exploration.

- Softmax/Boltzmann policy:

$$\pi_{\text{softmax}}(a) = f(Q, \tau) = \frac{\exp Q(a)/\tau}{\sum_{b \in \mathcal{A}} \exp Q(b)/\tau} \tag{3}$$

  where $\tau \in \mathbb{R}^+$ is a *temperature* parameter that scales the amount of exploration.

- Upper confidence bound (UCB) policy:

$$\pi_{\text{UCB}}(a) = f(Q, n, c) = \begin{cases} 1, & \text{if } a = \arg\max_b \left[Q(b) + c \cdot \sqrt{\frac{\ln t}{n(b)}} \right] \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

  where $c \in \mathbb{R}^+$ scales the amount of exploration, t denotes the timestep, and $n(a)$ denotes the number of previous visits to action a.

  Importantly, when $n(a) = 0$, we evaluate the expression between the brackets as ∞, which ensure that we always prefer an untried action over an action that has already been tried.

  This can be easier written as

$$\pi_{\text{UCB}} = \arg\max_a \left[Q(a) + c \cdot \sqrt{\frac{\ln t}{n(a)}} \right]$$

  (which returns an action instead of a probability of an action).

3.3 Update of a mean estimate

Given a sequence of observations $r_1, r_2, ...r_n$ for a particular arm a, we often want to estimate the mean

$$Q_n = \frac{r_1 + r_2 + ... + r_n}{n} = \frac{1}{n} \sum_{i=1}^{n} r_i$$

- Incremental mean update

$$\begin{aligned}
Q_n &= \frac{1}{n}\sum_{i=1}^{n} r_i \\
&= \frac{1}{n}\Big[r_n + \sum_{i=1}^{n-1} r_i\Big] \\
&= \frac{1}{n}\Big[r_n + (n-1)\frac{1}{(n-1)}\sum_{i=1}^{n-1} r_i\Big] \\
&= \frac{1}{n}\Big[r_n + (n-1)Q_{n-1}\Big] \\
&= \frac{1}{n}\Big[r_n + n \cdot Q_{n-1} - Q_{n-1}\Big] \\
Q_n &= Q_{n-1} + \frac{1}{n}\Big[r_n - Q_{n-1}\Big]
\end{aligned} \tag{5}$$

- Learning mean update

$$Q_n = Q_{n-1} + \alpha\Big[r_n - Q_{n-1}\Big] \tag{6}$$

for learning rate $\alpha \in (0, 1)$. This update is preferable for non-stationary problems, since it will weight more recent observations more heavily.

4 Full Algorithms

Algorithm 2: ϵ-greedy bandit algorithm.

Input: Exploration parameter $\epsilon \in [0, 1]$, maximum number of timesteps T.

Initialization: Initialize $Q(a) = 0$, $n(a) = 0$ $\forall a \in \mathcal{A}$

for $t = 1...T$ **do**

$$a_t = \begin{cases} \arg\max_{a \in \mathcal{A}} Q(a) & \text{with } p = 1 - \epsilon \\ \text{random non-greedy action,} & \text{with } p = \epsilon \end{cases}$$ `/*` ϵ`-greedy action */`

$r_t \sim p(r|a_t)$ `/* Sample reward */`

$n(a_t) \leftarrow n(a_t) + 1$ `/* Update count */`

$Q(a_t) \leftarrow Q(a_t) + \frac{1}{n(a_t)}\Big[r_t - Q(a_t)\Big]$ `/* Incr. update mean */`

end

Algorithm 3: Optimistic initialization with greedy action selection bandit algorithm.

Input: Initial value $\psi \in \mathbb{R}$, learning rate $\eta \in \mathbb{R}^+$, maximum number of timesteps T.

Initialization: Initialize $Q(a) = \psi$ $\forall a \in \mathcal{A}$ `/* Optimistic init. */`

for $t = 1...T$ **do**

$a_t = \arg\max_{a \in \mathcal{A}} Q(a)$ `/* Sample greedy action */`

$r_t \sim p(r|a_t)$ `/* Sample reward */`

$Q(a_t) \leftarrow Q(a_t) + \eta \cdot \Big[r_t - Q(a_t)\Big]$ `/* Learning update mean */`

end

Algorithm 4: UCB bandit algorithm.

Input: Exploration parameter $c \in \mathbb{R}^+$, maximum number of timesteps T.

Initialization: Initialize $Q(a) = 0$, $n(a) = 0$ $\forall a \in \mathcal{A}$

for $t = 1...T$ **do**

$a_t = \arg\max_a \left[Q(a) + c \cdot \sqrt{\frac{\ln t}{n(a)}} \right]$ `/* UCB action */`

$r_t \sim p(r|a_t)$ `/* Sample reward */`

$n(a_t) \leftarrow n(a_t) + 1$ `/* Update count */`

$Q(a_t) \leftarrow Q(a_t) + \frac{1}{n(a_t)} \left[r_t - Q(a_t) \right]$ `/* Incr. update mean */`

end