Reinforcement Learning

Christos Dimitrakakis

December 12, 2023

The multi-armed bandit (MAB) problem

- At time t:
- Select action $a_t \in A$
- Obtain reward $r_t \in \mathbb{R}$

Basic objective

Maximise total reward

$$U = \sum_{t=1}^{T} r_t,$$

where T is the horizon. It may be unknown, or random.

Regret

We can instead minimise total regret

$$L = \sum_{t=1}^{T} [r_t^* - r_t],$$

where r^* is the reward an oracle that knew the "best" arm would have obtained.
No let's make this more precise.

The stochastic MAB

For each arm $i \in A$:

- $r_t \mid a_t = i \sim \mu_i$ is the reward distribution
- $\rho_i \triangleq E_\mu[r_t \mid a_t = i]$ the expected reward
- $\rho^* \triangleq \max_i \rho_i$.

Policy

The policy $\pi \in \mathit{\Pi}$ is a adaptive: $\pi(a_t \mid a_{t-1}, r_{t-1}, \ldots, a_1, r_1)$

Objective

Maximise expected total reward

$$\mathbb{E}^\pi_\mu[U] = \mathbb{E}^\pi_\mu\left[\sum_{t=1}^T r_t\right]$$

The total expected regret is

$$\mathbb{E}^\pi_\mu[L] = \mathbb{E}^\pi_\mu\left[\sum_{t=1}^T \rho^* - \rho_t\right]$$

The horizon and regret

Discounted T

- $U = \sum_{t=1}^{T} \gamma^{t-1} r_t$
- Same as non-discounted with stopping probability $(1 - \gamma)$.

Arbitrary T

To compare algorithms, we use the notion of regret growth

- Linear regret: $L_T = O(T)$. i.e. insufficient learning
- Sub-linear regret, e.g. $L_T = O(\sqrt{T})$ or $O(\ln T)$.

Algorithms

ϵ-greedy

- $\hat{\rho}_{i,t}$ is the average reward of arm i at time t.
- w.p. ϵ, $a_t \sim \mathrm{Unif}(A)$
- otherwise, $a_t = \arg\max_i \hat{\rho}_{i,t}$,

UCB

- Play all arms once, and for $t > |A|$:
- $a_t = \arg\max_i \hat{\rho}_{i,t} + \sqrt{2\ln(t)/n_{i,t}}$.
- $n_{i,t}$ is the number of times arm i has been pulled until time t.

Thompson (posterior) sampling

Input: a prior β_1 over $\mathcal{M}$.

- At time t:
- Sample from the posterior $\mu^{(t)} \mid a_1, r_t, \ldots, a_{t-1}, r_{t-1} \sim \beta_t(\mu)$
- Choose best action for sample: $a_t = \arg\max_i \mathbb{E}_{\mu^{(t)}}[r_t \mid a_t = i]$.
- Observe r_t.
- Calculate new posterior $\beta_{t+1}(\mu) = \beta_t(\mu \mid a_t, r_t)$.

Other bandit problems

Adversarial bandits

- Rewards are arbitrary.
- Compare with best arm in hindsight.

Continuous bandits

- Actions $a_t \in \mathbb{R}^d$
- Example: Lipschitz bandits where $|\rho(a) - \rho(a')| \leq \|a - a'\|$.

Contextual bandits (in particular linear)

- Contexts $x_t \in \mathbb{R}^d$
- Unknown parameters $\theta_a \in \mathbb{R}^d$
- For the linear case $\rho(x, a) = x^\top \theta_a$.

The Markov decision process

Bandit problems are not dynamic. We can generalise reinforcement learning to dynamical systems through the MDP formalism:

- Action space A.
- State space S.
- Transition kernel $s_{t+1} = j \mid s_t = s, a_t = a \sim P_\mu(j \mid s, a)$.
- Reward $r_t = \rho(s_t, a_t)$ (can also be random).
- Utility

$$U_t = \sum_{k=t}^{T} r_t.$$

Value functions

The state value function

For any given MDP μ and policy π we define

$$V^{\pi}_{\mu,t}(s) \triangleq \mathbb{E}^{\pi}_{\mu,t}\left[U_t \mid s_t = s\right]$$

The state-action value function

$$Q^{\pi}_{\mu,t}(s, a) \triangleq \mathbb{E}^{\pi}_{\mu,t}\left[U_t \mid s_t = s, a_t = a\right]$$

The optimal value functions

For an optimal policy π^*

$$V^{*}_{\mu,t}(s) \triangleq V^{\pi^*}_{\mu,t}(s) \geq V^{\pi}_{\mu,t}(s), \qquad Q^{*}_{\mu,t}(s, a) \triangleq Q^{\pi^*}_{\mu,t}(s, a) \geq V^{\pi}_{\mu,t}(s, a)$$

The Bellman equations

State value function

$$
\begin{aligned}
V^{\pi}_{\mu,t}(s) &\triangleq \mathbb{E}^{\pi}_{\mu,t}[U_t \mid s_t = s] \\
&= \mathbb{E}^{\pi}_{\mu,t}[r_t + U_{t+1} \mid s_t = s] \\
&= \mathbb{E}^{\pi}_{\mu}[r_t \mid s_t = s] + \mathbb{E}^{\pi}_{\mu}[U_{t+1} \mid s_t = s] \\
&= \mathbb{E}^{\pi}_{\mu}[r_t \mid s_t = s] + \sum_{j \in S} \mathbb{E}^{\pi}_{\mu}[U_{t+1} \mid s_{t+1} = j] \, \mathbb{P}^{\pi}_{\mu}(s_{t+1} = j \mid s_t = s) \\
&= \mathbb{E}^{\pi}_{\mu}[r_t \mid s_t = s] + \sum_{j \in S} V^{\pi}_{\mu,t+1}(j) \, \mathbb{P}^{\pi}_{\mu}(s_{t+1} = j \mid s_t = s) \\
&= \mathbb{E}^{\pi}_{\mu}[r_t \mid s_t = s] + \sum_{j \in S} V^{\pi}_{\mu,t+1}(j) \sum_{a \in A} P_{\mu}(j \mid s, a)\pi(a_t \mid s_t).
\end{aligned}
$$

State-action value function

$$
Q^{\pi}_{\mu,t}(s) = [\rho(s, a) + \sum_{j \in S} V^{\pi}_{\mu,t+1}(j) P_{\mu}(j \mid s, a)]
$$

Optimal policies

Bellman optimality condition

The value function of the optimal policy satisfies this:

$$V^*_{\mu,t}(s) = \max_a [\rho(s, a) + \sum_{j \in S} V^*_{\mu,t+1}(j) P_\mu(j \mid s, a)$$

Dynamic programming

To find V^*, Q^*, first initialise $V^*_{\mu,T}(s) = \max_a \rho(s, a)$. Then for $t = T - 1, T - 2, \ldots, 1$:

$$Q^*_{\mu,t}(s, a) = \rho(s, a) + \sum_{j \in S} V^*_{\mu,t+1}(j) P_\mu(j \mid s, a).$$

$$V^*_{\mu,t}(s) = \max_a Q^*_{\mu,t}(s, a).$$

The optimal policy

The optimal policy is deterministic with:

$$a_t = \arg\max_a Q^*(s_t, a)$$

The Reinforcement Learning Problem

- Observe x_t
- Take action a_t
- Obtain reward r_t

Requirement for learning

- The model is not known
- Our policies must be adaptive

Reinforcement learning settings

Fully observable, discrete Markov problems

- $x_t = s_t$, a Markovian state, S, A finite.
- Optimal policies are Markov
- Can be solved efficiently with classical RL algorithms

Continuous Markov problems

- Requires function approximation
- Even when the model is known, hard to compute

Partially observable problems

- Sufficient statistics are not finite

Further resources

- The Sutton/Barto RL intro book
  http://incompleteideas.net/book/the-book-2nd.html
- The Lattimore/Szepesvari bandit book
  https://tor-lattimore.com/downloads/book/book.pdf
- The Dimitrakakis/Ortner RL book
- Reinforcement Learning Course at Neuchatel
  https://mcs.unibnf.ch/courses/
  reinforcement-learning-and-decision-making-under-uncertainty
- OpenAI Gym https://github.com/openai/gym/