CS540 Introduction to Artificial Intelligence

Lecture 14

Young Wu

Based on lecture slides by Jerry Zhu, Yingyu Liang, and Charles Dyer

July 21, 2023

Reinforcement Learning

Motivation

- An agent interacts with an environment and receives a reward based on the state of the environment and its action.
- The goal of reinforcement learning is to maximize the cumulative reward by learning the optimal actions in every state.

Applications of Reinforcement Learning

Motivation

- Robot control. (*e.g.* Google AI Robotics)
- Autonomous driving (*e.g.* Tesla Electric Cars)
- Large language model (*e.g.* ChatGPT)
- Game playing (*e.g.* AlphaGO)

Simplest Problem

Motivation

- A simple reinforcement problem with only one state is called the multi-armed bandit problem.

Multi-Armed Bandit Definition

Definition

- There is a set of actions $\mathcal{A} = \{1, 2, ..., K\}$.
- The reward from action k is given by $r : \mathcal{A} \to \mathbb{R}$, where $r(k) \sim D(\mu_k), \mu_k$ is the mean reward from action $k \in \mathcal{A}$.
- Time horizon T.
- The agent's objective is,

$$\max_{a_t \in \mathcal{A}} \sum_{t=1}^{T} r_t(a_t).$$

Regret minimization vs Best Action Identification

Definition

- Another way to write down the agent's objective is,

$$\min_{a_t \in \mathcal{A}} \left\{ \max_{k \in \mathcal{A}} \mu_k - \frac{1}{T} \sum_{t=1}^{T} r_t \left(a_t \right) \right\}.$$

- For theoretical analysis, sometimes μ_{a_t} is used in place of $r_t \left(a_t \right)$.
- An algorithm is no-regret if the as $T \to \infty$, the regret approaches 0 with probability 1.

Applications of Multi-Armed Bandit

Motivation

- Multi-armed bandits.
- Clinical trials.
- Stock selection.

Exploration vs Exploitation

Definition

- Epsilon-first strategy: $\varepsilon \cdot T$ rounds of pure exploration and use the empirically best action in the remaining $(1 - \varepsilon) \cdot T$ rounds.
- Epsilon-greedy strategy: the empirically best action is always used with probability $1 - \varepsilon$, and use a random action with probability ε, where ε could be decreasing over time.

Empirically Best Action

Definition

- Best action at round T can be the action with the highest average:

$$\underset{k \in \mathcal{A}}{\operatorname{argmax}}\, \hat{\mu}_k,$$

where $\hat{\mu}_k := \frac{1}{n_k} \sum_{t=1}^{T} r_t(a_t)\, \mathbb{1}_{\{a_t = k\}}$ and $n_k := \sum_{t=1}^{T} \mathbb{1}_{\{a_t = k\}}$.

- Best action at round T can also be the action with the highest upper confidence bound:

$$\underset{k \in \mathcal{A}}{\operatorname{argmax}}\, \hat{\mu}_k + c\sqrt{\frac{2\log(T)}{n_k}}.$$

Upper Confidence Bound

Definition

- The algorithm with $\varepsilon = 0$ and the best action chosen according to the upper confidence bound is called the UCB1 Algorithm.
- UCB1 uses the principle of optimism under uncertainty and $\hat{\mu}_k + c\sqrt{\frac{2\log(T)}{n_k}}$ is an optimistic guess of the μ_k.
- The expression $\sqrt{\frac{2\log(T)}{n_k}}$ computes the confidence width based on Hoeffding's inequality.

UCB1 Algorithm

Algorithm

- Input: K arms, T periods, constant parameter c.
- Output: a list of actions $\{a_t\}_{t=1}^{T}$
- For $t = 1, 2, ..., K$, pull each arm once, say $a_t = k$, and initialize $\hat{\mu}_{a_t} = r_t$.
- For $t > K$, pull the arm

  $a_t = \text{argmax}_{k \in \mathcal{A}} \hat{\mu}_k + c\sqrt{\left(2\frac{\log(T)}{n_k}\right)}$, and update

  $n'_{a_t} = n_{a_t} + 1$ and $\hat{\mu}'_{a_t} = \frac{1}{n_{a_t}+1}(\hat{\mu}_{a_t} n_{a_t} + r_{a_t})$.

Adversarial Bandit and EXP3

Discussion

- If the environment is adversarial, for example, the rewards are chosen by another agent, then a deterministic algorithm would fail. An example of a stochastic algorithm is the EXP3 algorithm: Exponential weight algorithm for Exploration and Exploitation.
- It keeps track of a weight vector and pull arms randomly according the weights. The weights are updated based on the rewards.

EXP3 Algorithm

Discussion

- Input: K arms, T periods, constant γ.
- Output: a list of actions $\{a_t\}_{t=1}^{T}$
- Initialize $w_k = 1$ for $k = 1, 2, ..., K$.
- In period t, randomly select action $a_t = k$ with probability $p_k = (1-\gamma)\frac{w_k}{\sum_{k'=1}^{K} w_{k'}} + \frac{\gamma}{K}$, and update $w'_{a_t} = w_{a_t} e^{\frac{\gamma r_{a_t}}{p_{a_t} K}}$.