Reinforcement Learning
(advances)
Outline

Previous class (basics)

- Markov Decision Process (MDP): basic model
- Planning: how to find optimal control given model
- Reinforcement learning: how to learn the model
- RL with function approximation: how to design scalable RL

Today (advances)

- Exploration vs. exploitation
- Advanced topics: meta RL, hierarchical RL, world model, ...
Outline

Exploration vs. exploitation

- Multi-Armed Bandit (MAB) problem
- Formulation of regret minimization
- UCB (Upper Confidence Bound) algorithm

Advances in efficient exploration

- Typical exploration in RL
- Structured exploration: meta RL, hierarchical RL
- Generative model: world model
Exploration and Exploitation

RL agent can actively or **adaptively collect** samples hence the trade-off between exploration and exploitation is a fundamental problem in RL:

- **Exploration**: try under-sampled options (policies/actions) as they can be better than the current best
- **Exploitation**: stick with the current best only if a certain confidence is guaranteed from sufficient evidence
Multi-Armed Bandit (MAB) Problem

Suppose slot machines have not necessarily the same but initially unknown payoffs. Which one should we play at each turn in order to maximize rewards?

source: Microsoft Research
A Mathematical Model: Stochastic MAB

\(K\)-armed bandit problem

- Random variable \(X_{i,n} \in [0,1]\) denotes arm \(i\)'s \(n\)-th reward (the bounded support is assumed w.o.l.g.) *with out loss of generality

- Assume that for \(n \geq 1\), \(X_{i,n}\) is \textit{independently} identically drawn from a distribution with expectation \(\mu_i\)

- Also, for given \(i\), \(X_{i,n}\)'s are assumed to be \textit{independent} to each other

source: Microsoft Research
Examples

The independence assumption implies that actions make no change in system state, i.e., MAB = an RL problem with single state

Real-world Applications

- Clinical trials to find cure for cancer
- Rate adaptation in wireless network
- Online ad placement or recommender system
Clinical Trials

To cure cancer:

- **Arms** = medicines
- **Reward** = reduction of cancer size
- c.f., best medicine may depend on cancer size, and there can be some correlation between successive treatments (MDP: an RL model with multiple states?)
Rate Adaptation in Wireless Network

To maximize throughput over noisy wireless channel:

- Arms = transmission rate
- Reward = bytes of successful transmission
- c.f., interference with other AP’s may need to be considered (multi-agent RL?)
Recommender System

To maximize number of hits in online ad placement:

- Arms = ads/contents
- Reward = user click
- c.f., personalization may improve hit-rate (contextual MAB?)
Regret Minimization

- $K$-armed bandit problem is to maximize cumulated reward

- Equivalently, we want to minimize the expected regret after $n$ plays, which is defined as

$$\text{Regret}(n) := \mu^* n - \sum_{i=1}^{K} \mu_i \mathbb{E}[T_i(n)] = \sum_{i=1}^{K} (\mu^* - \mu_i) \mathbb{E}[T_i(n)]$$

where $\mu^* := \max_{i \in [K]} \mu_i$ is the best expected reward, and $T_i(n)$ is the number of times playing arm $i \in [K]$. 
Simple Heuristics

For regret minimization,

- **Greedy strategy**: select the arm with the highest average after sampling $L$ times for each arm at the beginning
  - A little knowledge is dangerous
  - We may need some continuous exploration

- $\epsilon$-greedy: select an arm at random with probability $\epsilon \in [0, 1]$ and otherwise do a greedy selection
  - The value of $\epsilon$ controls the trade-off between exploration and exploitation
Regret Analysis of $\epsilon$-greedy

When $\epsilon$ is constant,

- After sufficient time $t \gg 1$, it eventually identifies best arm $i^*$
- However, it may keep exploring other arms with constant portion of time, i.e., linear regret $\simeq \sum_{t=1}^{n} \epsilon = O(n)$

When $\epsilon_t \propto 1/t$,

- After sufficient time $t \gg 1$, it also identifies best arm $i^*$ eventually, i.e., logarithmic regret

\[ \text{Regret} \simeq \sum_{t=1}^{n} \epsilon_t = \sum_{t=1}^{n} \frac{1}{t} = O(\log n) \]

Inefficient exploration! All the arms are explored equally! An efficient one may explore little clearly bad arm $i$ s.t. $\mu_i \ll \mu^*$. 
Upper Confidence Bound (UCB)

UCB1 selects arm with the highest potential which is quantified by UCB:

- **Initialization**: play each arm once.
- **Loop**: play arm $i$ that maximizes $\text{UCB} \ x_i + \sqrt{\frac{2\log n}{T_i(n)}}$, where
  - $n$: the overall number of plays done so far
  - $\bar{x}_i$: the empirical mean of arm $i$
  - $T_i(n)$: the number of times arm $i$ has been played so far

Observation: as $n$ increases, the term $\sqrt{\frac{2\log n}{T_i(n)}}$ increases.  
⇒ This ensures that all arms are tried infinitely often
Optimism in the Face of Uncertainty

The potential, UCB \( \bar{x}_i + \sqrt{\frac{2 \log n}{T_i(n)}} \), is higher for less explored arms.
Outline

Exploration vs. exploitation

- Multi-Armed Bandit (MAB) problem
- Formulation of regret minimization
- UCB (Upper Confidence Bound) algorithm

Advances in efficient exploration

- Typical exploration in RL
- Structured exploration: meta RL, hierarchical RL
- Generative model: world model
Convergence of Q-learning

Q-learning evaluates and improves simultaneously current policy $\pi$: when $(S_n, A_n, R_n, S'_n) = (s, a, r, s')$,

$$Q_{n+1}(s, a) \leftarrow (1 - \alpha_n(s, a)) Q_n(s, a) + \alpha_n(s, a) \left( r + \gamma \max_{a'} Q_n(s', a') \right) - Q_n(s, a)$$

Theorem ([Wat:QL])

For MDP with bounded reward, i.e., $\|R_n\| < \infty$, $Q_n \to Q^*$ w.p. 1 as $n \to \infty$ if

$$\sum_{n=0}^{\infty} \alpha_n(s, a) = \infty \quad \text{and} \quad \sum_{n=0}^{\infty} (\alpha_n(s, a))^2 < \infty \quad \forall (s, a) \in S \times A,$$

where the first part ensures that every pair of state and action is visited infinitely often

Typical Exploration in RL

**\(\varepsilon\)-greedy exploration**
- Select action maximizing \(Q(s, a)\) at state \(s\) w.p. \(1 - \varepsilon\)
- Select action **uniformly** at random w.p. \(\varepsilon\)

**Softmax exploration**
- Select random action \(a\) from **Boltzmann distribution**:

\[
\text{softmax}(a; s) = \frac{\exp(\beta Q(s, a))}{\sum_{a'} \exp(\beta Q(s, a'))}
\]

where \(\beta\) controls the degree of noise

If an MDP is communicating, i.e., for any \(s, s' \in S\), there always exists a sequence of action connecting them with positive probability, both exploration schemes guarantee infinitely often visits at each state and action
Algorithm 1 Soft Actor-Critic

Initialize parameter vectors $\psi, \bar{\psi}, \theta, \phi$.

for each iteration do
  for each environment step do
    $a_t \sim \pi_\phi(a_t|s_t)$
    $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$
    $D \leftarrow D \cup \{(s_t, a_t, r(s_t, a_t), s_{t+1})\}$
  end for
  for each gradient step do
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau \psi + (1 - \tau) \bar{\psi}$
  end for
end for

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim D} \left[ D_{KL} \left( \pi_\phi(\cdot | s_t) \parallel \frac{\exp(\mathcal{Q}_\theta(s_t, \cdot))}{Z_\theta(s_t)} \right) \right].$$

Soft Actor Critic [Haa:SAC]

Meta Reinforcement Learning (1)

We cannot train machine for every single task, whereas there are some structure underlying a set of tasks

- e.g., for the set of tasks, the state dynamics are the same, but each task is described by different parameters $g = (3, 2, 1), (1, 3, 2), ...$
Hindsight Experience Replay

Experience augmentation by imagining reward for different tasks

Algorithm 1 Hindsight Experience Replay (HER)

Given:
- an off-policy RL algorithm $A$,
- a strategy $S$ for sampling goals for replay,
- a reward function $r : S \times A \times G \rightarrow \mathbb{R}$.

Initialize $A$
Initialize replay buffer $R$

for episode = 1, $M$ do
    Sample a goal $g$ and an initial state $s_0$.
    for $t = 0, T-1$ do
        Sample an action $a_t$ using the behavioral policy from $A$:
        $a_t \leftarrow \pi_b(s_t||g)$
        Execute the action $a_t$ and observe a new state $s_{t+1}$
    end for
    for $t = 0, T-1$ do
        $r_t := r(s_t, a_t, g)$
        Store the transition $(s_t||g, a_t, r_t, s_{t+1}||g)$ in $R$
        Sample a set of additional goals for replay $G := S$(current episode)
        for $g' \in G$ do
            $r' := r(s_t, a_t, g')$
            Store the transition $(s_t||g', a_t, r', s_{t+1}||g')$ in $R$
        end for
    end for
    for $t = 1, N$ do
        Sample a minibatch $B$ from the replay buffer $R$
        Perform one step of optimization using $A$ and minibatch $B$
    end for
end for
Hindsight Experience Replay

Experience augmentation by imagining reward for different tasks

Meta Reinforcement Learning (2)

We cannot train machine for every single task, whereas there are some structure underlying a set of tasks

▶ e.g., a policy to accomplish task can be decomposed into several sub-policies in common

Frans, Kevin, et al. ”Meta learning shared hierarchies.” ICRL 2018
Hierarchical Reinforcement Learning

We do not know which hierarchy should we use. Let machine learn the hierarchy

https://youtu.be/zkJmH4NIzPs
Model-based RL

Summary

Exploration vs. exploitation
- Multi-Armed Bandit (MAB) problem
- Formulation of regret minimization
- UCB (Upper Confidence Bound) algorithm

Advances in efficient exploration
- Typical exploration in RL
- Structured exploration: meta RL, hierarchical RL