22. Introduction

Online learning in reinforcement learning refers to the idea that a learner is placed in an (initially) *unknown* MDP. By interacting with the MDP, the learner collects data about the unknown transition and reward function. The learner's goal is to collect as much reward as possible, or output a near-optimal policy. The difference to planning is that the learner does *not* have access to the true MDP. Unlike in batch RL, the learner gets to decide what actions to play. Importantly, this means the learner's action affect the data that is available to the learner (sometimes refered to as "closed loop").

The fact that the learner needs to create its own data leads to an important decision: Should the learner sacrifice reward to collect more data that will improve decision making in the future? Or should it act according to what seems currently best? Clearly, too much exploration will be costly if the learner chooses actions with low reward too often. On the other hand, playing actions that appear optimal with limited data comes at the risk of missing out on even better rewards. In the literature, this is commonly known as *exploration-exploitation dilemma*.

The exploration-exploitation dilemma is not specific to the MDP setting. It already arises in the simpler (multi-armed) bandit setting (i.e. an MDP with only one state and stochastic reward).

In the following, we focus on finite-horizon episodic (undiscounted) MDPs $M = (\mathcal{S}, \mathcal{A}, P, r, \mu)$. The learner interacts with the MDP for K episodes of length $H > 0$. At the beginning of each episode $k = 1 \ldots, K$, an initial state is sampled from the initial distribution $S_0^k \sim \mu$. The data collected during the k^{th} episode is

$$S_0^{(k)}, A_0^{(k)}, R_1^{(k)}, S_1^{(k)} A_1^{(k)}, R_2^{(k)}, S_2^{(k)}, \ldots S_{H-1}^{(k)}, A_{H-1}^{(k)}, R_H^{(k)}, S_H^{(k)}$$

where A_h^k is the action chosen by the learner at step h, $S_{h+1}^{(k)} \sim P_{A_h^{(k)}}(S_h^{(k)})$ is the next state and $R_{h+1}^{(k)} \sim r_{A_h^{(k)}}(S_h^{(k)})$ is the (possibly stochastic) reward.

This model contains some important settings as a special case. Most notably,

- $H = 1$ recovers the contextual bandit setting, where the "context" $S_0^{(k)}$ is sampled from the distribution μ

- $H = 1$ and $S = 1$ is the finite multi-armed bandit setting.

Sample complexity and regret: How good is the learner?

The goal of the learner is to collect as much reward as possible. We denote $V_k = \sum_{h=0}^{H-1} r_{A_h^{(k)}}(S_h^{(k)})$ as the reward collected by the learner in episode k. The total reward is $\sum_{k=1}^{K} V_k$. For the analysis it will be useful to introduce a normalization: Instead of directly arguing about the total reward, we compare the learner to the value $v_0^*(S_0^{(k)})$ of the best policy in the MDP. This leads to the notation of *regret* defined as follows:

$$R_K = \sum_{k=1}^{K} \left(v_0^*(S_0^{(k)}) - V_k\right)$$

A learner has sublinear expected regret if $\mathbb{E}[R_K/K] \to 0$ as $K \to \infty$. Sublinear regret means that the average reward of the learner approaches the optimal value $v_0^*(\mu)$ as the number of episodes increases. Certainly that is a desirable property!

Before we go on to construct learners with small regret, we briefly note that there are also other objectives. The most common alternative is PAC - which stands for *probably approximately correct*. A learner is said to be (ϵ, δ)-PAC if upon termination in episode K, it outputs a policy such that $v_0^*(s_0^{(k)}) - \mathbb{E}[V_K] \le \epsilon$ with probability at least $1 - \delta$. We have discussed PAC bounds already in the context of planning.

The difference to bounding regret is that in the first $K - 1$ episodes, the learner does not 'pay' for choosing suboptimal actions. This is sometimes called a *pure exploration problem*. Note that a learner that achieves sublinear regret can be converted into a PAC learner (discussed in the notes). However, this may lead to a suboptimal (large) K in the PAC framework.

ϵ-greedy

There exist many ideas on how to design algorithms with small regret. We first note that a "greedy" agent can easily fail: Following the best actions according to some empirical estimate can easily get you trapped in a supoptimal policy (think of some examples where this can happen!).

A simple remedy is to add a small amount of "forced" exploration: With (small) probability ϵ, we choose an action uniformly at random. Thereby we eventually collect samples from all actions to improve our estimates. With probabilty $(1 - \epsilon)$ we follow the "greedy" choice, that is the action that appears best under our current estimates. This gives raise to the name ϵ-greedy.

It is often possible to show that ϵ-greedy converges. By carefully choosing the exploration probability ϵ, we may show that in finite MDPs, the regret is at most $R_K \leq \mathcal{O}(K^{2/3})$. As we will discuss later, there are multiple algorithms that achieve a regret of only $\mathcal{O}(K^{1/2})$. Thus, ϵ-greedy is not the best algorithm to minimize regret.

Not unexpectedly, this type of exploration can be quite sub-optimal. It is easy to construct examples, where ϵ-greedy takes exponential time (in the number of states) to reach an optimal policy. Can you find an example (Hint: construct the MDP such that each time the agent explores a suboptimal action, the agent is reset to the starting state)?

On the upside, ϵ-greedy is very simple and can easily used in more complex scenarios. In fact, it is a popular choice when using neural network function approximations, where theoretically grounded exploration schemes are much harder to obtain.

Optimism Principle

A popular technique to construct regret minimizing algorithms is based on *optimism in the face of uncertainty*. To formally define the idea, let $\mathcal{M}$ be the set of possible environments (e.g. finite MDPs). We make the realizability assumption that the true environment $M^* \in \mathcal{M}$ is in this set. After obtaining data in rounds $1, \dots, k-1$, the learner uses the observations to compute a set of plausible models $\mathcal{M}_k \subset \mathcal{M}$. The plausible model set is such that it contains the true model with high probabilty. Although this is not always required, it is useful to think of a decreasing sequence of sets $\mathcal{M} \supset \mathcal{M}_1 \supset \mathcal{M}_2 \supset \cdots \supset \mathcal{M}_k$. This simply means that as more data arrives, the learner is able to exclude models that are statistically unlikely to produce the observation data.

The optimism principle is to act according to the policy that achieves the highest reward among all plausible models, i.e.

$$\pi_k = \arg\max_{\pi} \max_{M \in \mathcal{M}_k} v_M^\pi \tag{1}$$

At this point it not be clear why this leads to an efficient learning algorithm (with small regret). The idea is that the learner systematically obtains data about the environment. For example, if data contradicts the optimistic model $\tilde{M}_k = \arg\max_{M \in \mathcal{M}} \max_\pi v_M^\pi$, then $\tilde{M}_k \notin \mathcal{M}_{k+1}$ is excluded from the set of plausible models in the future. Consequently, the learner chooses a different policy in the next round.

On the other hand, the learner ensures that $M^* \in \mathcal{M}_k$ with high probability. In this case, it is often possible to show that the gap $v_{\tilde{M}_k}^{\pi_k} - v_{M^*}^* \geq 0$ is small (more specifically, behaves like a statistical estimation error of order $\mathcal{O}(t^{-1/2})$ with a leading constant that depends on the "size" of $\mathcal{M}$).

One should also ask if the optimization problem (1) can be solved efficiently. This is far from always the case. Often one needs to rely on heuristics to implement the optimistic policy, or use other exploration techniques such as Thompson sampling (see below).

How much regret the learner has of course depends on the concrete setting at hand. In the next lecture we will see how we can make use of optimism to design (and analyize) an online learning algorithm for finite MDPs. The literature has produced a large amount of papers with algorithms that use the optimism principle in many settings. This however does not mean that optimism is a universal tool. More recent literature has also pointed out limitations of the optimsm principle, and in lieu proposed other design ideas.

Notes

Other Exploration Techniques

Some other notable exploration strategies are:

- Phased-Elimination and Experimental Design
- Thompson Sampling
- Information-Directed Sampling (IDS) and Estimation-To-Decisions (E2D)

References

The paper showing the details behind how to convert between Regret and PAC bounds.

- Dann, C., Lattimore, T., & Brunskill, E. (2017). Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning. Advances in Neural Information Processing Systems, 30. [link]