DDA4230 Reinforcement learning Discrete MDPs

Lecture 9

Lecturer: Guiliang Liu *Scribe: Baoxiang Wang*

1 Goal of this lecture

In this lecture we will introduce the formulation of discrete Markov decision processes and some properties induced by the Bellman equation and the Bellman optimality equation thereof.
Suggested reading: Chapter 3 of *Bandit algorithms*; Chapter 3 and 4 of *Reinforcement learning: An introduction*; Chapter 1 and 2 of *Reinforcement learning: Theory and algorithms.*

2 Recap: Discrete Markov decision processes

We consider the discrete-time Markov decision process (MDP) setting, denoted as the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho_0, \gamma)$.

- $\mathcal{S} = [n]$ the state space;
- $\mathcal{A} = [m]$ the action space. $\mathcal{A}$ can depend on the state s for $s \in \mathcal{S}$;
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ the environment transition probability function;
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$ the reward function;
- $\rho_0 \in \Delta(\mathcal{S})$ the initial state distribution;
- $\gamma \in [0, 1]$ the unnormalized discount factor.

Note that $\Delta(\mathcal{X})$ denotes the set of all distributions over set $\mathcal{X}$.

A stationary MDP follows for $t = 0, 1, \dots$ as below, starting with $s_0 \sim \rho_0$.

- The agent observes the current state s_t;
- The agent chooses an action $a_t \sim \pi(a_t \mid s_t)$;
- The agent receives the reward $r_t \sim \mathcal{R}(s_t, a_t)$;
- The environment transitions to a subsequent state according to the Markovian dynamics $s_{t+1} \sim \mathcal{T}(s_t, a_t)$.

This process generates the sequence $s_0, a_0, r_0, s_1, \dots$ indefinitely. The sequence up to time t is defined as the trajectory indexed by t, as $\tau_t = (s_0, a_0, r_0, s_1, \dots, r_t)$.

The goal is to optimize the expected return

$$\mathbb{E}_{s_t,a_t,r_t,t\geq 0}\left[R_0\right] = \mathbb{E}_{s_t,a_t,r_t,t\geq 0}\Big[\sum_{t=0}^{\infty} \gamma^t r_t\Big]$$

over the agent's policy π.

3 Discrete Markov chains

A Markov chain, also known as a homogeneous Markov chain, refers to a infinite process $x_1, \ldots, x_T, \ldots$ where

$$\mathbb{P}(x_{t+1} \mid x_t, \ldots, x_1) = \mathbb{P}(x_{t+1} \mid x_t) = \mathbb{P}_{\mathcal{M}}(x' \mid x)$$

holds almost surely for some probability measure $\mathbb{P}_{\mathcal{M}}$. A discrete Markov chain restricts the state space $\mathcal{S}$ to be countable and in this lecture notes we assume the state space $[n]$ to be finite. In a Markov chain, $\mathbb{P}_{\mathcal{M}}(x_{t+1} \mid x_t)$ is called the probability kernel and is required to be time-invariant. When the context is clear we write $\mathbb{P}_{\mathcal{M}} = \mathbb{P}$.

As $x, x' \in [n]$, it is convenient to represent $\mathbb{P}$ with n^2 many values of probabilities. Define the transition probability matrix P (A Markov chain could be induced by an MDP with a fixed policy. So in MDPs this P corresponds to the notation P^π.) where the element $P_{ii'}$ on the i-th and i'-th column equals $\mathbb{P}(x' = i' \mid x = i)$. Similarly, denote the state value function $V(s)$ as $V \in \mathbb{R}^n$ and the reward function $\mathcal{R}(s)$ as $r \in \mathbb{R}^n$. The occupancy vector ρ_t, where the i-th element of ρ_t denotes $\mathbb{P}(s_t = i \mid s_0 \sim \rho_0)$, is then $P^t \rho_0$.

When $\mathcal{R}(s)$ is deterministic, r is a deterministic vector. The Bellman equation then writes $V = r + \gamma P V$. Since P is a Markov matrix, $I - \gamma P$ is invertible when $\gamma < 1$ and the value function can be solved by

$$V = (I - \gamma P)^{-1} r\,.$$

Reducible states For two states $i, i' \in [n]$, if there exists a T such that $\mathbb{P}(i' \in \{s_1, \ldots, s_T\} \mid s_0 = i) > 0$, we say that i' is accessible from i. If i is accessible from i' and i' is accessible from i, we say that i and i' communicate with each other. If for any $i, i' \in [n]$, i and i' communicate with each other, the Markov chain is irreducible.

For Markov chain that is reducible, it is intuitive to partition the chain into irreducible components (likewise, to consider each connected components in a graph). It is therefore sensible to assume that the Markov chain to be irreducible.

Periodicity For $i \in [n]$, the period of state i is the largest integer d satisfying $\mathbb{P}(s_t \neq i \mid s_0 = i,\ t \neq 0 \mod d)$, or infinity if such a largest integer does not exist. When $d = 1$, state i is aperiodic, and otherwise, state i is periodic with period d.

In a irreducible Markov chain, all states have the same period. A irreducible Markov chain is aperiodic is all states are aperiodic. Periodicity plays an important role in the limiting distribution of a Markov chain. Mathematically, a chain is aperiodic if and only if P^t contains only positive elements for some positive integer t.

Ergodicity A Markov chain that is irreducible and aperiodic must be ergodic. We commonly assume a chain to be ergodic without loss of generality.

For the rest of the course, unless otherwise specified, we assume the Markov chains to be ergodic. In MDPs however, in general, there exist policies such that the chain induced by the policies are not ergodic.

4 Policy evaluation

Recall that in reinforcement learning, our goal is to optimize over the policy space to maximize the value function. This describes reinforcement learning as an optimization problem, where we have no explicit expression of the objective function. One critical information that the majority of the algorithms need is at least the query access to this objective function, known as the optimization oracle. In the context of reinforcement learning, this corresponds to computing the value function given a fixed policy. Note that the computation can be approximate or probabilistic in general.

When the policy is fixed, the MDP reduces into a Markov chain as described in the last section. Therefore, the policy evaluation problem is to find V or equivalently Q given P, r, and γ. When at least one of P and r is known, the problem is *policy evaluation with a known model*. When both P and r are unknown we can make an effort to estimate a P' such that P and P' are close in some measure of discrepancy (or r', respectively). If we do so our method is categorized into *model-based policy evaluation*. If otherwise and we only utilize the access to the environment transition, the method is categorized as *model-free policy evaluation*.

4.1 Policy evaluation with a known model

Under discrete state and action spaces, when both P and r are known the solution $V = (I - \gamma P)^{-1} r$ is immediate when $\gamma < 1$ as we pointed out in the last section. One argument to show that $(I - \gamma P)$ is indeed invertible is that the largest element of $(I - \gamma P)x$ for an arbitrary non-negative vector x is at least $(1 - \gamma)\|x\|_\infty$, which is strictly positive unless $x = 0$. The eigenvalue of $(I - \gamma P)$ must therefore be nonzero, indicating that its rank must be full (n).

Alternatively, one can resort to an iterative solution that maintains a value function estimate that converges to the true value function, for which we will cover in the next lecture.

4.2 Model-based policy evaluation

A model-based approach does not know the model, but can maintain an estimation of it and use the estimation when calculating the value function. It is then very straightforward to consider replacing P with $\hat{P}$ for some estimation $\hat{P}$, where the most simple way to obtain $\hat{P}$ is to use the empirical distribution of the state transitions collected from the trajectories.

Lemma 1 *Assume that $0 \le r \le 1$. Let $\varepsilon \in (0, \frac{1}{1-\gamma})$. There is an absolute constant c such that once one have collected at least*

$$N \ge \frac{\gamma}{(1-\gamma)^4} \frac{n^2 m \log(cnm/\delta)}{\varepsilon^2}$$

samples for each $(s, a) \in \mathcal{S} \times \mathcal{A}$ pair, then we could estimate $\hat{P}$ and $\hat{Q}^\pi$ such that with probability at least $1 - \delta$,

$$\|P(\cdot \mid s, a) - \hat{P}(\cdot \mid s, a)\|_1 \le (1-\gamma)^2 \varepsilon$$

for every (s, a) pair, and

$$\|Q^\pi - \hat{Q}^\pi\|_\infty \le \varepsilon$$

for every policy π.

The number N of samples needed could be reduced to

$$N \ge \frac{c}{(1-\gamma)^3} \frac{nm \log(cnm/\delta)}{\varepsilon^2} \,,$$

if we only desire the accurate estimate on the optimal policy and value, i.e., $\|Q^* - \hat{Q}^*\|_\infty \le \varepsilon$ and $\|Q^* - Q^{\hat{\pi}^*}\|_\infty \le \varepsilon$. This improvement is minimax optimal for estimating the optimal policy and value in this way.

The natural question remaining is that if we are able to obtain N samples for each (s, a) pair so as to fulfill the condition of the lemma. The answer is, unfortunately, no, in general.

5 The Bellman optimality equation

We have already discussed that the Bellman equation leads to the exact solution $V = (I - \gamma P)^{-1} r$ of the value function, where P corresponds to the transition matrix given a fixed policy. We now discuss the Bellman optimality equation of an optimal policy. Without loss of generality, let the reward function be deterministic. We also assume that the reward is bounded by $[0, 1]$ in discrete MDPs unless otherwise stated. Then the reward function $\mathcal{R}(s, a)$ is written by a matrix $r \in \mathbb{R}^{n \times m}$, where the element at the i-th row and the j-th column denotes $\mathcal{R}(i, j)$. Let P_j be the transition matrix for the policy that choose action j at every state. Recall that in discrete MDPs a value function V is optimal if and only if the Bellman optimality equation is satisfied. In fact, the "if" relation is immediate, and the "only if" relation is shown in Page 64 of *Reinforcement learning: An introduction.* A more formal argument could be found in Theorem 1.8 of *Reinforcement learning: Theory and algorithms.*

The Bellman optimality equation states that the optimal value function V equals $r + \gamma P^* V$, for some P^* optimized over the policies. In the discrete setting, this translates to that V is greater than or equal to $r + \gamma P V$ for any feasible P. Since the "greater than or equal to $\ge$" operator is element-wise, it is equivalent to that V is greater than or equal to $r + \gamma P V$ for every $P \in \{P_1, \dots, P_m\}$.

By exhausting the action set under the max operator and numbering the actions from 1 to m, the Bellman optimality equation is formulated into the below linear program:

$$\begin{aligned} \underset{V}{\text{minimize}} \quad & \mathbf{e}^T V \\ \text{subject to} \quad & (I - \gamma P_j) V - r_j \ge 0 \,, \quad j = 1, \dots, m \,, \end{aligned} \tag{1}$$

where $\mathbf{e}$ is the all-one vector and $\mathbf{e}^T V$ is a dummy objective. Linear programming is in P and can be solved in $\text{poly}(n, m)$. We consider a problem solved if we can cast it to a linear program. Though, this requires P_i to be known.

The dual of the linear program (1) is

$$\begin{aligned} \underset{\lambda_1,\ldots,\lambda_m}{\text{maximize}} \quad & \sum_j \lambda_j^T r_j \\ \text{subject to} \quad & \sum_j (I - \gamma P_j^T)\lambda_j = \mathbf{e}\,, \\ & \lambda_j \geq 0, \quad j = 1, \ldots, m\,. \end{aligned}$$

The dual formulation optimizes $\lambda_1, \ldots, \lambda_m$, which could be regarded as the policy.

Lemma 2 *There exists an optimal dual solution λ_j^*, $j = 1, \ldots, m$, an optimal deterministic policy $\pi^*(\cdot)$, and the corresponding transition matrix P^*, such that*

$$\sum_j \lambda_j^* = (I - \gamma P^{*T})^{-1}\mathbf{e},$$

and the i-th entry of λ_j^* equals to the i-th entry of $\sum_j \lambda_j^*$ if $\pi^*(i) = j$, and zero otherwise.

Proof: Denote the superscript (i) as the i-th element for a vector and as the i-th row for a matrix. Specify ξ_j^* to be any dual optimal solution and construct the policy $\pi^*(i) = \arg\max_j \xi_j^{*(i)}$ where arg max breaks ties arbitrarily. Then, let

$$\lambda^* = (I - \gamma P^{*T})^{-1}\mathbf{e}\,,$$

where P^* is the transition matrix of $\pi^*(\cdot)$. The inversion exists since all the eigenvalues of the Markov matrix P^* are smaller than one. Define λ_j^*, $j = 1, \ldots, m$, such that $\lambda_j^{*(i)} = \lambda^{*(i)}$ whenever $\pi^*(i) = j$ and zero otherwise. We have for λ_j^* that

$$\sum_i \sum_j \lambda_j^{(i)} (I - \gamma P_j)^{(i)} = \mathbf{e}\,,$$

which rewrites the dual feasibility by summing over i. We also have $\lambda_i^{*(i)} = 0$ whenever $\xi_j^{*(i)} = 0$ for any j and i, and together with the slackness

$$\xi_j^{*T}((I - \gamma P_j)V - r_j) = 0\,,$$

we have $\lambda_j^{*T}((I - \gamma P_j)v - r_j) = 0$. The optimality of λ_j^*, $j = 1, \ldots, m$, follows. □

Lemma 3 *The ℓ^1-norm $\|\sum_j \lambda_j^*\|_1$ of the dual optimum is exactly $n/(1-\gamma)$.*

Proof: By definition we have $\|\sum_j \lambda_j^*\|_1 = \|\lambda^*\|_1$ and $(I - \gamma P^{*T})\lambda^* = \mathbf{e}$. Since P^* is a Markov matrix, we have $\|P^{*T}\lambda^*\|_1 = \|\lambda^*\|_1$. Taking ℓ^1-norm and we have $\|\lambda^*\|_1 - \gamma\|\lambda^*\|_1 = \|\mathbf{e}\|_1$. The statement follows. □

Lemma 4 *The stochastic policy $\pi(j \mid i) = \lambda_j'^{(i)} / \sum_{j'} \lambda_{j'}'^{(i)}$ achieves a value V' such that $\mathbf{e}^T V' = \sum_j \lambda_j'^T r_j$.*

Proof: With Lemma 2 showing the existence, specify $\lambda'' = (I - \gamma P''^T)^{-1}\mathbf{e}$ and λ''_j to be the optimal solution of the dual problem, where P'' is the corresponding transition matrix. The Bellman optimality equation indicates that $((I - \gamma P_j)V' - r_j)^{(i)} = 0$ whenever $\lambda_j''^{(i)} > 0$. It is equivalent to $(I - \gamma P'')V' - \tilde{r} = 0$ where $\tilde{r}^{(i)} = r_{\pi(i)}^{(i)}$, $i = 1, \ldots, n$. Hence,

$$\mathbf{e}^T V' = \mathbf{e}^T (I - \gamma P'')^{-1} \tilde{r} = \tilde{r}^T (I - \gamma P'')^{-1} \mathbf{e} = \tilde{r}^T \lambda'' = \sum_j \lambda_j'^T r_j \,,$$

where the last equality follows the definition of λ''. □

Armed with these lemmas, we find that the dual of the Bellman optimality equation serves as a formulation of policy optimization. As we investigate more methods in MDPs without a known model, we will see similar interconnections of value optimization and policy optimization

Acknowledgement

This lecture notes partially use material from *Reinforcement learning: An introduction* and *Reinforcement learning: Theory and algorithms.*