STATS 701 – Theory of Reinforcement Learning
Markov Decision Processes

Ambuj Tewari

Associate Professor, Department of Statistics, University of Michigan
tewaria@umich.edu
https://ambujtewari.github.io/stats701-winter2021/

Slide Credits: Prof. M. Vidyasagar @ IIT Hyderabad, India

Winter 2021

Outline

Outline

Markov Processes with Action Variables

In a Markov process, the state $\{X_t\}_{t\geq 0}$ evolves on its own, with a state transition matrix A.

In a Markov Decision Process (MDP), there is an additional "action" variable U_t taking values in a finite set $\mathcal{U}$ with $|\mathcal{U}| = m$.

For each $u_k \in \mathcal{U}$, there is a corresponding state transition matrix A^{u_k}. So if action u_k is applied at time t, then

$$\Pr\{X_{t+1} = x_j | X_t = x_i, U_t = u_k\} = a_{ij}^{u_k}.$$

There is also a "reward" function $R : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$.

Note: R can (i) be a random function of X_t and U_t, (ii) be paid at the "next" time $t+1$, or both.

The Key Idea: Policy

A key idea in MDPs is policy: A map $\pi : \mathcal{X} \to \mathcal{U}$ that assigns X_t to U_t.

- Deterministic policy: A map from $\mathcal{X}$ to $\mathcal{U}$.
- The set Π_d of deterministic policies has cardinality $|\mathcal{U}|^{|\mathcal{X}|}$.
- A policy $\pi \in \Pi_d$ can be represented by an $m \times n$ matrix where each column i has a single entry of 1 in row $\pi(x_i)$ and the rest are zero.
- Probabilistic policy: A map from $\mathcal{X}$ to the set of probabilities on $\mathcal{U}$.
- The set Π_p of probabilistic policies is uncountable.
- A policy $\pi \in \Pi_p$ can be represented by a nonnegative $m \times n$ matrix where the entries of each column add up to one.

Impact of a Policy

Key Observation: Whether a policy $\pi \in \Pi_d$ or Π_p, the resulting process is a Markov reward process.

If $\pi \in \Pi_d$, then the resulting state transition matrix is

$$(A^{\pi})_{ij} = a_{ij}^{\pi(x_i)},$$

and reward function is

$$R_{\pi}(x_i) = R(x_i, \pi(x_i)).$$

Example

Suppose $|\mathcal{X}| = 4$, $|\mathcal{U}| = 2$, and the policy $\pi \in \Pi_d$ is represented by

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

Then the state transition matrix A^π consists of the first row of A^2, the second row of A^1, the third row of A^1 and the fourth row of A^2.

If we represent R as an $n \times m$ matrix where $R_{ij} = R(x_i, u_k)$, then under π we get

$$R_\pi = [\ R_{12} \quad R_{21} \quad R_{31} \quad R_{42}\ ].$$

Probabilistic Policies

Suppose $\pi \in \Pi_p$, and suppose that, for each index i, we have

$$\pi(x_i) = [\ (\pi(x_i))_{u_1} \quad \cdots \quad (\pi(x_i))_{u_m}\ ].$$

Then the i-th row of the state transition matrix A^π is given by

$$(A^\pi)^i = \sum_{k=1}^{n} (A^{u_k})^i \pi(x_i)_{u_k},$$

where A^i denotes the i-th row of the matrix A. That is

$$\Pr\{X_{t+1} = x_j | X_t = x_i, \pi\} = \sum_{u_k \in \mathcal{U}} (\pi(x_i))_{u_k} a_{ij}^{u_k}.$$

Example

Suppose as before that $|\mathcal{X}| = 4$, $|\mathcal{U}| = 2$, and the policy $\pi \in \Pi_p$ is represented by

$$\begin{bmatrix} 0.3 & 0.6 & 0.5 & 0.8 \\ 0.7 & 0.4 & 0.5 & 0.2 \end{bmatrix}.$$

Then the state transition matrix A^π is given by

$$A^\pi = \begin{bmatrix} 0.3(A^1)^1 + 0.7(A^2)^1 \\ 0.6(A^1)^2 + 0.4(A^2)^2 \\ 0.5(A^1)^3 + 0.5(A^2)^3 \\ 0.8(A^1)^4 + 0.2(A^2)^4 \end{bmatrix},$$

where $(A)^i$ denotes the i-th row of the matrix A.

$$R_\pi = \begin{bmatrix} 0.3R_{11} + 0.7R_{12} & 0.6R_{21} + 0.4R_{22} & 0.5R_{31} + 0.5R_{32} & 0.8R_{41} + 0.2R_{42} \end{bmatrix}.$$

Key Question 1

Policy evaluation: For a given policy π, define $V_\pi(x_i)$ to be the "value" associated with the policy π and initial state x_i, that is, the expected discounted future reward with $X_0 = x_i$. How can $V_\pi(x_i)$ be computed for each $x_i \in \mathcal{X}$?

Key Question 2

Optimal Value Determination: For a specified initial state x_i, define

$$V^*(x_i) := \max_{\pi \in \Pi_d} V_\pi(x_i),$$

to be the **optimal value** over all policies. How can $V^*(x_i)$ be computed?

Note that in the above equation, the optimum is taken over all *deterministic* policies. In principle one could also seek the optimum value over all *probabilistic* policies; but we do not study this situation in these notes.

Key Question 3

Optimal Policy Determination: Define the optimal policy map $\mathcal{X} \to \Pi_d$ via

$$\pi^*(x_i) := \arg\max_{\pi \in \Pi_d} V_\pi(x_i).$$

How can the optimal policy map π^* be determined? Note that, while the set Π_d may be enormous, it is a finite set; hence the maximum indicated above definitely exists (though it may not be unique).

Outline

Outline

Value of a Policy by Iteration

Each policy $\pi \in \Pi_d$ results in a Markov process with state transition matrix A^π and reward function R_π.

Define the vector $\mathbf{v}_\pi$ by

$$\mathbf{v}_\pi = [\ V_\pi(x_1) \quad \dots \quad V_\pi(x_n)\ ],$$

and the reward vector $\mathbf{r}_\pi$ by

$$\mathbf{r}_\pi = [\ R_\pi(x_1) \quad \dots \quad R_\pi(x_n)\ ].$$

Then $\mathbf{v}_\pi$ satisfies the familiar relation

$$\mathbf{v}_\pi = \mathbf{r}_\pi + \gamma A^\pi \mathbf{v}_\pi.$$

This equation can be solved by iteration, as before.

Outline

Acton-Value Function

Definition

The action-value function $Q : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is defined by

$$Q_\pi(x_i, u_k) := E_\pi\left[\sum_{t=0}^{\infty} \gamma^t R_\pi(X_t) | X_0 = x_i, U_0 = u_k\right].$$

Note: In the definition of $Q_\pi(x_i, u_k)$, at the first instant $t = 0$ we choose the action u_k as we wish, not necessarily as $u_k = \pi(x_i)$.

But for $t \geq 1$, we choose $U_t = \pi(X_t)$.

Q can be viewed as a real vector of dimension $|\mathcal{X}| \cdot |\mathcal{U}|$.

Recursive Relationshop of Action-Value Function

Theorem

The function Q satisfies the recursive relationship

$$Q_\pi(x_i, u_k) = R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} Q_\pi(x_j, \pi(x_j)).$$

Proof in the notes.

Relationship Between Action-Value and Value Functions

Theorem

The functions V_π and Q_π are related via

$$V_\pi(x_i) = Q_\pi(x_i, \pi(x_i)).$$

In view of this theorem, the recursive equation for Q_π, namely

$$Q_\pi(x_i, u_k) = R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} Q_\pi(x_j, \pi(x_j)).$$

can be rewritten as

$$Q_\pi(x_i, u_k) = R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} V_\pi(x_j).$$

Optimal Value Function and Optimal Policy

Let $V^*(x_i)$ denote the maximum value of the discounted future reward, over all policies $\pi \in \Pi_d$:

$$V^*(x_i) := \max_{\pi \in \Pi_d} V_\pi(x_i).$$

Note that, though the set Π_d may be huge, it is nevertheless a finite set. Therefore the maximum above exists. Also define

$$\pi^* = \arg\max_{\pi \in \Pi_d} V_\pi(x_i).$$

Thus π^* is any policy such that $V_{\pi^*} = V^*$.

Bellman Optimality Equation

Theorem

Define $V^*(x_i)$ as above. Then $V^*(x_i)$ satisfies the reursive relationship

$$V^*(x_i) = \max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \gamma \sum_{j \in [n]} a_{ij}^{u_k} V^*(x_j) \right].$$

This is known as the Bellman optimality equation. Note that it does not help us to find the function V^*; it just a characterization of V^*.

Rationalization of the Bellman Optimality Equation

Suppose that we have somehow determined the maximum possible value $V^*(1, x_j)$ at time $t = 1$ for each state $x_j \in \mathcal{X}$. Now at time $t = 0$, suppose the state $X_0 = x_i$. Then each action $u_k \in \mathcal{U}$ leads to the value

$$R(x_i, u_k) + \gamma \sum_{j \in [n]} a_{ij}^{u_k} V^*(1, x_j).$$

So the maximum value at time $t = 0$, state $X_0 = x_i$ is given by

$$V^*(0, x_i) = \max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \gamma \sum_{j \in [n]} a_{ij}^{u_k} V^*(1, x_j) \right].$$

However, since both the MDP and policy are time-invariant, we must have that $V(1, x_i) = V(0, x_i)$ for each $x_i \in \mathcal{X}$.

Recursive Relationship for Optimal Action-Value Function

Theorem

Define

$$Q^*(x_i, u_k) = R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} V^*(x_j).$$

Then $Q^*(\cdot,\cdot)$ satisfies the relationship

$$Q^*(x_i, u_k) = R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} \max_{w_l \in \mathcal{U}} Q^*(x_j, w_l).$$

Note that the maximum w.r.t. $u_k \in \mathcal{U}$ is in a different place compared to the Bellman equation. This is crucial.

Advantage of Optimal Action-Value Function

Theorem

Once $Q^*(\cdot, \cdot)$ is determined, we have that

$$V^*(x_i) = \max_{u_k \in \mathcal{U}} Q^*(x_i, u_k),$$

$$\pi^*(x_i) = \arg\max_{u_k \in \mathcal{U}} Q^*(x_i, u_k),$$

Moreover, it is easier to "learn" $Q^*(\cdot, \cdot)$ than to "learn" $V^*(\cdot)$, as we shall see.

Outline

1. Markov Decision Processes: Problem Formulation
2. Markov Decision Processes: Solution Methodologies
   - Value of a Policy by Iteration
   - Action-Value Function and the Bellman Optimality Equation
   - Value and Policy Iterations
   - Linear Programming Formulation

Value Update Map

Define the optimal value vector $\mathbf{v}^*$ as

$$\mathbf{v}^* = [\ V^*(x_1) \quad \cdots \quad V^*(x_n)\ ].$$

Next, define a "value update" map $T : \mathbb{R}^n \to \mathbb{R}^n$, as follows:

$$(T\mathbf{v})_i := \max_{u \in \mathcal{U}} \left[R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u v_j \right].$$

One can think of $\mathbf{v}$ as the current guess for the vector $\mathbf{v}^*$, and of $T\mathbf{v}$ as an updated guess. The next theorem shows that this intuition is valid.

Value Iteration

Theorem

The map T is both monotone and a contraction. As a result, for all $\mathbf{v}_0 \in \mathbb{R}^n$, the sequence of iterations $\{T^k \mathbf{v}_0\}$ approaches $\mathbf{v}^*$ as $k \to \infty$.

Value Iteration Proof

Monotonicity is easy to prove, hence focus on contraction

$$
\begin{aligned}
& |(T\mathbf{w})_i - (T\mathbf{v})_i| \\
&= |\max_u [R(x_i, u) + \gamma \sum_j a_{ij}^u w_j] - \max_u [R(x_i, u) + \gamma \sum_j a_{ij}^u v_j]| \\
&\leq \max_u |\gamma \sum_j a_{ij}^u w_j - \gamma \sum_j a_{ij}^u v_j| \\
&= \gamma \max_u |(A^u w)_i - (A^u v)_i| \leq \gamma \max_u \|A^u \mathbf{w} - A^u \mathbf{v}\|_\infty \\
&\leq \gamma \max_u \|A^u\|_{\infty \to \infty} \|\mathbf{w} - \mathbf{v}\|_\infty \\
&\leq \gamma \|\mathbf{w} - \mathbf{v}\|_\infty
\end{aligned}
$$

Action-Value Iteration

Define $\mathbf{q} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{U}|}$ as the vector $[Q(x_i, u_k), x_i \in \mathcal{X}, u_k \in \mathcal{U}]$. Define $F : \mathbb{R}^{|\mathcal{X}| \times |\mathcal{U}|} \to \mathbb{R}^{|\mathcal{X}| \times |\mathcal{U}|}$ by

$$(F\mathbf{q})(x_i, u_k) := R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} \max_{w_l \in \mathcal{U}} Q(x_j, w_l).$$

Theorem

The map F is monotone and is a contraction. Therefore for all $\mathbf{q}_0 \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{U}|}$, the sequence of iterations $\{T^k(\mathbf{q}_0)\}$ converges to $\mathbf{q}^*$.

Determining Optimal Policy from Optimal Value Function

Determining the optimal policy π^* from the optimal value function is easy.

Theorem

Suppose the optimal value vector $\mathbf{v}^*$ *is known, and define, for each* $x_i \in \mathcal{X}$, *the policy* $\pi^* : \mathcal{X} \to \mathcal{U}$ *via*

$$\pi^*(x_i) = \arg\max_{u_k \in \mathcal{U}} \left[R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^{u_k} V^*(x_j) \right].$$

But this requires knowledge of the optimal value function $\mathbf{v}^*$.

Is there another way? Yes, to update policy along with value iteration.

Policy Iteration

Set iteration counter $k = 0$ and choose some initial policy π_0. Then at iteration k,

- Compute the value vector $\mathbf{v}_{\pi_k}$ such that $\mathbf{v}_{\pi_k} = T_{\pi_k}\mathbf{v}_{\pi_k}$. Note that computing $\mathbf{v}_{\pi_k}$ by value iteration would require infinitely many applications of the map T_{π_k} to some arbitrary initial vector.
- Use this value vector $\mathbf{v}_{\pi_k}$ to compute an updated policy π_{k+1}, via

$$\pi_{k+1}(x_i) = \arg\max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \gamma \sum_{j \in [n]} a_{ij}^{u_k} (\mathbf{v}_{\pi_k})_j \right].$$

Policy Iteration (Cont'd)

Note that the above equation implies that

$$T_{\pi_{k+1}}\mathbf{v}_{\pi_k} = T\mathbf{v}_{\pi_k} \geq T_{\pi_k}\mathbf{v}_{\pi_k} = \mathbf{v}_{\pi_k}$$

where the last eq. is because v_{π_k} is the value function of π_k

Since the map $T_{\pi_{k+1}}$ is monotone, we can keep applying it to get

$$\forall \ell \geq 1,\, T^{\ell}_{\pi_{k+1}}\mathbf{v}_{\pi_k} \geq \mathbf{v}_{\pi_k}$$

Note: LHS converges to $\mathbf{v}_{\pi_{k+1}}$ as $\ell \to \infty$

Convergence of Policy Iteration

Suppose policy iteration doesn't improve the policy, i.e. $\pi_{k+1} = \pi_k$ and the inequality $T\mathbf{v}_{\pi_k} \geq \mathbf{v}_{\pi_k}$ is equality. Then we have

$$T\mathbf{v}_{\pi_k} = \mathbf{v}_{\pi_k}$$

So $\mathbf{v}_{\pi_k}$ must be the optimal value function and the corresponding greedy policy $\pi_{k+1} = \pi_k$ must be the optimal policy.

Theorem

We have that

$$\mathbf{v}_{\pi_{k+1}} \geq \mathbf{v}_{\pi_k},$$

where the dominance is componentwise. Consequently, there exists a finite integer k_0 such that $\mathbf{v}_{\pi_k} = \mathbf{v}^*$ for all $k \geq k_0$.

Outline

1. Markov Decision Processes: Problem Formulation

2. Markov Decision Processes: Solution Methodologies
   - Value of a Policy by Iteration
   - Action-Value Function and the Bellman Optimality Equation
   - Value and Policy Iterations
   - Linear Programming Formulation

LP: Primal

Let $\mathbf{d}$ be any distribution over states

$$\min_{\mathbf{v}} \mathbf{d}^\top \mathbf{v}$$
$$\text{s.t. } \mathbf{v} \geq T\mathbf{v}$$

It is clear that $\mathbf{v}^\star$ is feasible and therefore the minimum is at most $\mathbf{d}^\top \mathbf{v}^\star$

Claim: the minimum above is equal to $\mathbf{d}^\top \mathbf{v}^\star$
Why? $\mathbf{v} \geq T\mathbf{v}$ implies $\mathbf{v} \geq T^\ell \mathbf{v} \Rightarrow \mathbf{v} \geq \mathbf{v}^\star \Rightarrow \mathbf{d}^\top \mathbf{v} \geq \mathbf{d}^\top \mathbf{v}^\star$

LP: equivalent form

$$\min_{\mathbf{V}} \sum_{x_i \in \mathcal{X}} d(x_i) V(x_i)$$

$$\text{s.t. } \forall x_i \in \mathcal{X}, \quad V(x_i) \geq \max_{u_k \in \mathcal{U}} \; R(x_i, u_k) + \gamma \sum_{x_j \in \mathcal{X}} a_{ij}^{u_k} V(x_j)$$

LP: equivalent form

$$\min_{\mathbf{V}} \sum_{x_i \in \mathcal{X}} d(x_i) V(x_i)$$

$$\text{s.t. } \forall x_i \in \mathcal{X}, u_k \in \mathcal{U}, \quad V(x_i) \geq R(x_i, u_k) + \gamma \sum_{x_j \in \mathcal{X}} a_{ij}^{u_k} V(x_j)$$

This LP has n unconstrained variables and mn inequality constraints
Dual LP will have mn non-negative variables and n equality constraints

LP: Dual

$$\max_{\boldsymbol{\mu}} \sum_{x_i \in \mathcal{X}, u_k \in \mathcal{U}} \mu(x_i, u_k) R(x_i, u_k)$$

$$\text{s.t. } \forall x_i \in \mathcal{X}, \sum_{u_k \in \mathcal{U}} \mu(x_i, u_k) = d(x_i) + \gamma \sum_{x_j \in \mathcal{X}} \sum_{u_k \in \mathcal{U}} A_{ij}^{u_k} \mu(x_j, u_k)$$

Interpretation of $\boldsymbol{\mu}$: discounted state-action visitation frequencies

$$\mu(x_j, u_k) = \sum_{t=0}^{\infty} \gamma^t P(X_t = x_j, U_t = u_k)$$

Solution directly encodes optimal policy: $\pi^\star(x_j) = \arg\max_{u_k} \boldsymbol{\mu}^\star(x_j, u_k)$