Reinforcement Learning **Spring Semester, 2018/9**

Lecture 3: March 13, 2019

Lecturer: Yishay Mansour *Scribe: ym*

In the previous lecture we considered multi-stage decision problems for *deterministic* systems. In many problems of interest, the system dynamics also involves *randomness*, which leads us to stochastic decision problems. In this lecture we introduce the basic model of Markov Decision Processes, which will be considered in the rest of the course.

3.1 Markov Chains

We start by considering a stochastic dynamics model which does not have any actions, so that we can concentrate on the dynamics.

A Markov chain $\{X_t,\ t = 0, 1, 2, \ldots\}$, with $X_t \in X$, is a discrete-time stochastic process, over a finite or countable state-space X, that satisfies the following Markov property:

$$\mathcal{P}(X_{t+1} = j | X_t = i, X_{t-1}, \ldots X_0) = \mathcal{P}(X_{t+1} = j | X_t = i).$$

We focus on time-homogeneous Markov chains, where

$$\mathcal{P}(X_{t+1} = j | X_t = i) = \mathcal{P}(X_1 = j | X_0 = i) \stackrel{\Delta}{=} p_{ij}.$$

The p_{ij}'s are the transition probabilities, which satisfy $p_{ij} \geq 0, \quad \sum_{j \in X} p_{ij} = 1 \ \ \forall i \in X$, namely, $\{p_{i,j} : j \in X\}$ is a distribution on the next state following state i. The matrix $P = (p_{ij})$ is the transition matrix.

Given the initial distribution p_0 of X_0, namely $\mathcal{P}(X_0 = i) = p_0(i)$, we obtain the finite-dimensional distributions:

$$\mathcal{P}(X_0 = i_0, \ldots, X_t = i_t) = p_0(i_0) p_{i_0 i_1} \cdot \ldots \cdot p_{i_{t-1} i_t}.$$

Define $p_{ij}^{(m)} = \mathcal{P}(X_m = j | X_0 = i)$, the m-step transition probabilities. It is easy to verify that $p_{ij}^{(m)} = [P^m]_{ij}$ (here P^m is the m-th power of the matrix P).

State classification:

- State j is *accessible* from i (denoted by $i \to j$) if $p_{ij}^{(m)} > 0$ for some $m \geq 1$.
- States i and j are *communicating* states (or communicate) if $i \to j$ and $j \to i$.
- A *communicating class* (or just class) is a maximal collection of states that communicate.
- The Markov chain is *irreducible* if all states belong to a single class (i.e., all states communicate with each other).
- State i has a *period* $d_i = GCD\{m \geq 1 : p_{ii}^{(m)} > 0\}$, where GCD is the greatest common divisor. A state is a-periodic if $d_i = 1$. Periodicity is a class property: all states in the same class have the same period. Specifically, if some state is *a-periodic*, then all states in the class are a-periodic.

Recurrence:

- State i is *recurrent* if $\mathcal{P}(X_t = i \text{ for some } t \geq 1 | X_0 = i) = 1$. Otherwise, state i is *transient*.
- State i is recurrent if and only if $\sum_{m=1}^{\infty} p_{ii}^{(m)} = \infty$. By Borel-Cantteli this implies (with probability 1) that recurrent states appear infinitely often and transient states appear only a finite number of times.
- Recurrence is a class property.
- If states i and j are in the same recurrent (communicating) class, then state j is (eventually) reached from state i with probability 1: $\mathcal{P}(X_t = j \text{ for some t} \geq 1 | X_0 = i) = 1$.
- Let T_i be the return time to state i (number of stages required for (X_t) to return to i). If i is a recurrent state, then $T_i < \infty$ w.p. 1.
- State i is positive recurrent if $E(T_i) < \infty$, and null recurrent if $E(T_i) = \infty$. If the state space is finite, all recurrent states are positive recurrent.

Invariant Distribution: The probability vector $\mu = (\mu_i)$ is an invariant distribution or stationary distribution for the Markov chain if $\pi P = \pi$, namely

$$\mu_j = \sum_i \mu_i p_{ij} \quad \forall j.$$

Clearly, if $X_t \sim \mu$ then $X_{t+1} \sim \mu$. If $X_0 \sim \mu$, then the Markov chain (X_t) is a stationary stochastic process.

Theorem 3.1 (**Recurrence of finite Markov chains**). *Let (X_t) be an irreducible, a-periodic Markov chain over a finite state space X. Then the following properties hold:*

1. *All states are **positive recurrent***
2. *There exists a **unique stationary distribution** μ^**
3. ***Convergence** to the stationary distribution: $\lim_{t\to\infty} p_{ij}^{(t)} = \mu_j \quad (\forall j)$*
4. ***Ergodicity**: For any finite f: $\lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t-1} f(X_s) = \sum_i \mu_i f(i) \triangleq \pi \cdot f$.*

For countable Markov chains, there are other possibilities.

Theorem 3.2 (**Countable Markov chains**). *Let (X_t) be an irreducible and a-periodic Markov chain over a countable state space X. Then:*

1. *Either (i) all states are positive recurrent, or (ii) all states are null recurrent, or (iii) all states are transient.*
2. *If (i) holds, then properties (2)-(4) of the previous Theorem hold as well.*
3. *Conversely, if there exists a stationary distribution μ then properties (1)-(4) are satisfied.*

Reversible Markov chains: Suppose there exists a probability vector $\mu = (\mu_i)$ so that

$$\mu_i p_{ij} = \mu_j p_{ji}, \qquad i, j \in X. \tag{3.1}$$

It is then easy to verify by direct summation that μ is an invariant distribution for the Markov chain defined by (p_{ij}). This follows since $\sum_i \mu_i p_{i,j} = \sum_i p_{i,j} \mu_j = \mu_j$. The equations (3.1) are called the *detailed balance equations*. A Markov chain that satisfies these equations is called reversible.

Example 3.1 (**Discrete-time queue**). *Consider a discrete-time queue, with queue length $X_t \in \mathbb{N}_0 = \{0, 1, 2, \dots\}$. At time instant t, A_t new jobs arrive, and then up to S_t jobs can be served, so that*

$$X_{t+1} = (X_t + A_t - S_t)^+.$$

Suppose that (S_t) is a sequence of i.i.d. RVs, and similarly (A_t) is a sequence of i.i.d. RVs, with (S_t), (A_t) and X_0 mutually independent. It may then be seen that $(X_t,\ t \geq 0)$ is a Markov chain. Suppose further that each S_t is a Bernoulli RV with parameter q, namely $P(S_t = 1) = q$, $P(S_t = 0) = 1 - q$. Similarly, let A_t be a Bernoulli RV with parameter p. Then

$$p_{ij} = \begin{cases} p(1-q) & : \ j = i+1 \\ (1-p)(1-q) + pq & : \ j = i, \ \ i > 0 \\ (1-p)q & : \ j = i-1, \ \ i > 0 \\ (1-p) + pq & : \ j = i = 0 \\ 0 & : \ \text{otherwise} \end{cases}$$

Denote $\lambda = p(1-q)$, $\eta = (1-p)q$, and $\rho = \lambda/\eta$. The detailed balance equations for this case are:

$$\mu_i \lambda = \mu_{i+1} \eta, \qquad i \geq 0$$

These equations have a solution with $\sum_i \mu_i = 1$ if and only if $\rho < 1$. The solution is $\mu_i = \mu_0 \rho^i$, with $\pi_0 = 1 - \rho$. This is therefore the stationary distribution of this queue.

3.2 Controlled Markov Chains

A Markov Decision Process consists of two main parts:

1. A controlled dynamic system, with stochastic evolution.
2. A performance objective to be optimized.

In this section we describe the first part, which is modeled as a controlled Markov chain.

Consider a controlled dynamic system, defined over:

- A discrete time axis $\mathbb{T} = \{0, 1, \dots, T-1\}$ (finite horizon), or $\mathbb{T} = \{0, 1, 2, \dots\}$

(infinite horizon). To simplify the discussion we refer below to the infinite horizon case, which can always be "truncated" at T if needed.

- A finite state space S, where $S_t \subset S$ is the set of possible states at time t .
- A finite action set A, where $A_t(s) \subset A$ is the set of possible actions at time t and state $s \in S_t$.

State transition probabilities:

- Suppose that at time t we are in state $s_t = s$, and choose an action $a_t = a$. The next state $s_{t+1} = s'$ is then determined randomly according to a probability distribution $p_t(\cdot|s, a)$ on S_{t+1}. That is,

$$\mathcal{P}(s_{t+1} = s'|s_t = s, a_t = a) = p_t(s'|s, a), \qquad s' \in S_{t+1}$$

- The probability $p_t(s'|s, a)$ is the *transition probability* from state s to state s' for a given action a. We naturally require that $p_t(s'|s, a) \geq 0$, and $\sum_{s' \in S_{t+1}} p_t(s'|s, a) = 1$ for all $s \in S_t, a \in A_t(s)$.
- Implicit in this definition is the controlled-Markov property:

$$\mathcal{P}(s_{t+1} = s'|s_t, a_t) = \mathcal{P}(s_{t+1} = s'|s_t, a_t, \ldots, s_{0,}a_0)$$

- The set of probability distributions

$$P = \{p_t(\cdot|s, a) \quad : \quad s \in S_t, a \in A_t(s), t \in \mathbf{T}\}$$

is called the *transition law* or *transition kernel* of the controlled Markov process.

Stationary Models: The controlled Markov chain is called stationary or time-invariant if the transition probabilities do not depend on the time t. That is:

$$\forall t, \quad p_t(s'|s, a) \equiv p(s'|s, a), \quad S_t \equiv S, \quad A_t(s) \equiv A(s).$$

Graphical Notation: The state transition probabilities of a Markov chain are often illustrated via a state transition diagram, such as in Figure 3.1.

A graphical description of a controlled Markov chain is a bit more complicated because of the additional action variable. We obtain the diagram (drawn for state

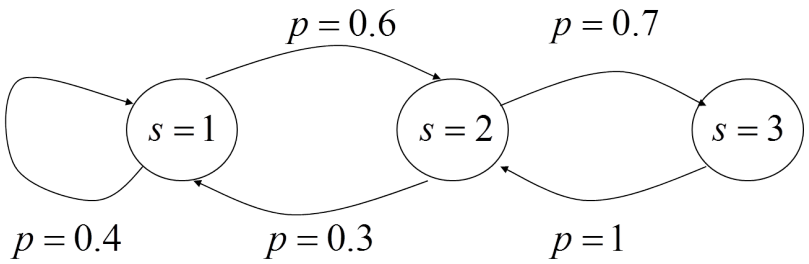

Figure 3.1: Markov chain

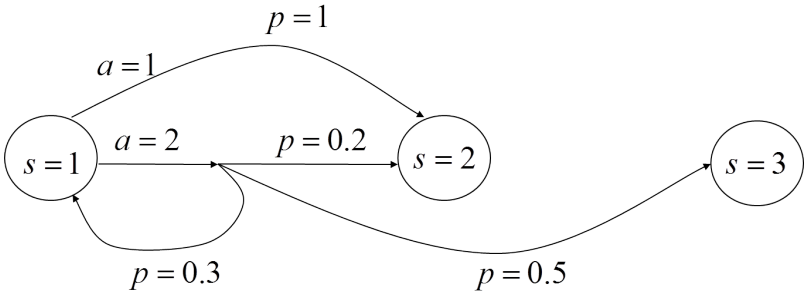

Figure 3.2: Controlled Markov chain

$s = 1$ only, and for a given time t) in Figure 3.2, reflecting the following transition probabilities:

$$p(s' = 2|s = 1, a = 1) = 1$$
$$p(s'|s = 1, a = 2) = \begin{cases} 0.3 & : \quad s' = 1 \\ 0.2 & : \quad s' = 2 \\ 0.5 & : \quad s' = 3 \end{cases}$$

State-equation notation: The stochastic state dynamics can be equivalently defined in terms of a state equation of the form

$$s_{t+1} = f_t(s_t, a_t, w_t),$$

where w_t is a random variable. If $(w_t)_{t\geq 0}$ is a sequence of independent RVs, and further each w_t is independent of the "past" $(s_{t-1}, a_{t-1}, \ldots s_0)$, then $(s_t, a_t)_{t\geq 0}$ is a controlled Markov process. For example, the state transition law of the last example can be written in this way, using $w_t \in \{4, 5, 6\}$, with $p_w(4) = 0.3$, $p_w(5) =$

0.2, $p_w(6) = 0.5$ and, for $s_t = 1$:

$$\begin{aligned} f_t(1, 1, w_t) &= 2 \\ f_t(1, 2, w_t) &= w_t - 3 \end{aligned} .$$

The state equation notation is especially useful for problems with continuous state space, but also for some models with discrete states.

Control Policies

- A general or **history-dependent** control policy $\pi = (\pi_t)_{t \in \mathbb{T}}$ is a mapping from each possible history $h_t = (s_0, a_0, \ldots, s_{t-1}, a_{t-1}, s_t)$, $t \in \mathbb{T}$, to an action $a_t = \pi_t(h_t) \in A_t$. We denote the set of general policies by Π_H.

- A **Markov** control policy π is allowed to depend on the current state and time only: $a_t = \pi_t(s_t)$. We denote the set of Markov policies by Π_M.

- For stationary models, we may define **stationary** control policies that depend on the current state alone. A stationary policy is defined by a single mapping $\pi : S \to A$, so that $a_t = \pi(s_t)$ for all $t \in \mathbb{T}$. We denote the set of stationary policies by Π_S.

- Evidently, $\Pi_H \supset \Pi_M \supset \Pi_S$.

Randomized Control policies

- The control policies defined above specify deterministically the action to be taken at each stage. In some cases we want to allow for a random choice of action.

- A general randomized control policy assigns to each possible history h_t a probability distribution $\pi_t(\cdot|h_t)$ over the action set A_t. That is, $\mathcal{P}\{a_t = a|h_t\} = \pi_t(a|h_t)$. We denote the set of history-dependent stochastic policies by Π_{HS}.

- Similarly, we can define the set Π_{MS} of Markov stochastic control policies, where $\pi_t(\cdot|h_t)$ is replaced by $\pi_t(\cdot|s_t)$, and the set Π_{SS} of stationary stochastic control policies, where $\pi_t(\cdot|s_t)$ is replaced by $\pi(\cdot|s_t)$.

- Note that the set Π_{HS} includes all other policy sets as special cases.

The Induced Stochastic Process Let $p_0 = \{p_0(s), s \in S_0\}$ be a probability distribution for the initial state s_0. A control policy $\pi \in \Pi_{HS}$, together with the transition law $P = \{p_t(s'|s,a)\}$ and the initial state distribution $p_0 = (p_0(s),\ s \in S_0)$, induce a probability distribution over any finite state-action sequence $h_T = (s_0, a_0, \ldots, s_{T-1}, a_{T-1}, s_T)$, given by

$$P(h_T) = p_0(s_0) \prod_{t=0}^{T-1} p_t(s_{t+1}|s_t, a_t)\pi_t(a_t|h_t),$$

where $h_t = (s_0, a_0, \ldots, s_{T-1}, a_{T-1}, s_t)$. To see this, observe the recursive relation:

$$\begin{aligned} cP(h_{t+1}) &= P(h_t, a_t, s_{t+1}) = P(s_{t+1}|h_t, a_t)P(a_t|h_t)P(h_t) \\ &= p_t(s_{t+1}|s_t, a_t)\pi_t(a_t|h_t)P(h_t). \end{aligned}$$

In the last step we used the conditional Markov property of the controlled chain: $P(s_{t+1}|h_t, a_t) = p_t(s_{t+1}|s_t, a_t)$, and the definition of the control policy π. The required formula follows by recursion.

Therefore, the state-action sequence $h_\infty = (s_k, a_k)_{k\geq 0}$ can now be considered a stochastic process. We denote the probability law of this stochastic process by $P^{\pi,p_0}(\cdot)$. The corresponding expectation operator is denoted by $E^{\pi,p_0}(\cdot)$. When the initial state s_0 is deterministic (i.e., $p_0(s)$ is concentrated on a single state s), we may simply write $P^{\pi,s}(\cdot)$ or $P^{\pi}(\cdot|s_0 = s)$.

Under a Markov control policy, the state sequence $(s_t)_{t\geq 0}$ becomes a *Markov chain*, with transition probabilities

$$P(s_{t+1} = s'|s_t = s) = \sum\nolimits_{a\in A_t} p_t(s'|s,a)\pi_t(a|s).$$

This follows since:

$$P(s_{t+1} = s'|s_t = s) = \sum\nolimits_{a\in A_t} P(s_{t+1} = s', a|s_t = s) =$$

$$\sum\nolimits_{a\in A_t} P(s_{t+1} = s'|s_t = s)P(a|s_t = s) = \sum\nolimits_{a\in A_t} p_t(s'|s,a)\pi_t(a|s)$$

If the controlled Markov chain is stationary (time-invariant) and the control policy is stationary, then the induced Markov chain is stationary as well.

Remark 3.1. *For most non-learning optimization problems, Markov policies suffice to achieve the optimum.*

Remark 3.2. *Implicit in these definitions of control policies is the assumption that the current state s_t can be fully observed before the action a_t is chosen . If this is not the case we need to consider the problem of a Partially Observed MDP (POMDP), which is more involved.*

3.3 Performance Criteria

3.3.1 Finite Horizon Problems

Consider the finite-horizon problem, with a fixed time horizon T. As in the deterministic case, we are given a running reward function $r_t = \{r_t(s,a) : s \in S_t, a \in A_t\}$ for $0 \leq t \leq T-1$, and a terminal reward function $r_T = \{r_T(s) : s \in S_T\}$. The obtained rewards are then $E[R_t] = r_t(s_t, a_t)$ at times $t \leq T-1$, and $E[R_T] = r_T(s_T)$ at the last stage. Our general goal is to maximize the cumulative return:

$$E[\sum_{t=0}^{T} R_t] = \sum_{t=0}^{T-1} r_t(s_t, a_t) + r_T(s_T).$$

However, since the system is stochastic, the cumulative return will generally be random, and we need to specify in which sense to maximize it. A natural first option is to consider the expected value of the return. That is, define:

$$J_T^\pi(s) = E^\pi(\sum_{t=0}^{T} R_t | s_0 = s) \equiv E^{\pi,s}(\sum_{t=0}^{T} R_t)$$

Here π is the control policy as defined above, and s denotes the initial state. Hence, $J_T^\pi(s)$ is the expected cumulative return under the control policy π. Our goal is to find an optimal control policy that maximized $J_T^\pi(s)$.

Remarks:

1. Dependence on the next state: In some problems, the obtained reward may depend on the next state as well: $R_t = \tilde{r}_t(s_t, a_t, s_{t+1})$. For control purposes, when we only consider the expected value of the reward, we can reduce this reward function to the usual one by defining

$$r_t(s,a) \stackrel{\Delta}{=} E(R_t | s_t = s, a_t = a) \equiv \sum\nolimits_{s' \in S} p(s'|s,a)\tilde{r}_t(s,a,s')$$

2. Random rewards: The reward R_t may also be random, namely a random variable whose distribution depends on (s_t, a_t). This can also be reduced to our standard model for planning purposes by looking at the expected value of R_t, namely

$$r_t(s,a) = E(R_t | s_t = s, a_t = a).$$

3. Risk-sensitive criteria: The expected cumulative return is by far the most common goal for planning. However, it is not the only one possible. For example, one may consider the following risk-sensitive return function:

$$J^{\pi}_{T,\lambda}(s) = \frac{1}{\lambda} \log E^{\pi,s}(\exp(\lambda \sum_{t=0}^{T} R_t)).$$

For $\lambda > 0$, the exponent gives higher weight to high rewards, and the opposite for $\lambda < 0$.

3.3.2 Infinite Horizon Problems

We next consider planning problems that extend to an unlimited time horizon, $t = 0, 1, 2, \ldots$. Such planning problems arise when the system in question is expected to operate for a long time, or a large number of steps, possibly with no specific "closing" time. Infinite horizon problems are most often defined for stationary problems. In that case, they enjoy the important advantage that optimal policies can be found among the class of stationary policies. We will restrict attention here to stationary models. As before, we have the running reward function $r(s, a)$, which extends to all $t \geq 0$. The expected reward obtained at stage t is $E[R_t] = r(s_t, a_t)$.

Discounted return: The most common performance criterion for infinite horizon problems is the expected discounted return:

$$J^{\pi}_{\alpha}(s) = E^{\pi}(\sum_{t=0}^{\infty} \alpha^t r(s_t, a_t) | s_0 = s) \equiv E^{\pi,s}(\sum_{t=0}^{\infty} \alpha^t r(s_t, a_t))$$

Here $0 < \alpha < 1$ is the discount factor. Mathematically, the discount factor ensures convergence of the sum (whenever the reward sequence is bounded). This make the problem "well behaved", and relatively easy to analyze.

Average return: Here we are interested to maximize the long-term average return. The most common definition of the long-term average return is

$$J^{\pi}_{av}(s) = \liminf_{T \to \infty} E^{\pi,s}(\frac{1}{T} \sum_{t=0}^{T-1} r(s_t, a_t))$$

The theory of average-return planning problems is more involved, and relies to a larger extent on the theory of Markov chains.

3.3.3 Stochastic Shortest-Path Problems

In an important class of planning problems, the time horizon is not set beforehand, but rather the problem continues until a certain event occurs. This event can be defined as reaching some goal state. Let $S_G \subset S$ define the set of *goal states*. Define

$$\tau = \inf\{t \geq 0 : s_t \in S_G\}$$

as the first time in which a goal state is reached. The total expected return for this problem is defined as:

$$J^{\pi}_{ssp}(s) = E^{\pi,s}(\sum_{t=0}^{\tau-1} r(s_t, a_t) + r_G(s_\tau))$$

Here $r_G(s),\ s \in S_G$ specified the reward at goal states.

This class of problems provides a natural extension of the standard shortest-path problem to stochastic settings. Some conditions on the system dynamics and reward function must be imposed for the problem to be well posed (e.g., that a goal state may be reached with probability one). Such problems are known as stochastic shortest path problems, or also episodic planning problems.

3.4 *Sufficiency of Markov Policies

In all the performance criteria defined above, the criterion is composed of sums of terms of the form $E(r_t(s_t, a_t))$. It follows that if two control policies induce the same marginal probability distributions $p_t(s_t, a_t)$ over the state-action pairs (s_t, a_t) for all $t \geq 0$, they will have the same performance.

Using this observation, the next claim implies that it is enough to consider the set of (stochastic) Markov policies in the above planning problems.

Proposition 3.1. *Let $\pi \in \Pi_{HS}$ be a general (history-dependent, stochastic) control policy. Let*

$$p^{\pi,s_0}_t(s, a) = P^{\pi,s_0}(s_t = s, a_t = a), \qquad (s, a) \in S_t \times A_t$$

Denote the marginal distributions induced by (s_t, a_t) on the state-action pairs (s_t, a_t), for all $t \geq 0$. Then there exists a stochastic Markov policy $\tilde{\pi} \in \Pi_{MS}$ that induces the same marginal probabilities (for all initial states s_0).

In the previous lecture we showed for Deterministic Decision Process for the finite horizon that there is an optimal deterministic policy. The proof that every stochastic

history dependent strategy has an equivalent stochastic Markovian policy showed how to generate the same state-action distribution, and applies to other setting as well. The proof that every stochastic Markovian policy has a deterministic Markovian policy depended on the finite horizon, but it is easy to extend it to other settings as well.

3.5 Finite-Horizon Dynamic Programming

Recall that we consider the expected total reward criterion, which we denote as

$$J^{\pi}(s_0) = E^{\pi,s_0}\left(\sum_{t=0}^{T-1} r_t(s_t, a_t) + r_T(s_T)\right)$$

Here π is the control policy used, and s_0 is a given initial state. We wish to maximize $J^{\pi}(s_0)$ over all control policies, and find an optimal policy π^* that achieves the maximal reward $J^*(s_0)$ for all initial states s_0. Thus,

$$J_T^*(s_0) \stackrel{\Delta}{=} J_T^{\pi*}(s_0) = \max_{\pi \in \Pi_{HS}} J_T^{\pi}(s_0)$$

3.5.1 The Principle of Optimality

The celebrated principle of optimality (stated by Bellman) applies to a large class of multi-stage optimization problems, and is at the heart of Dynamic Programming. As a general principle, it states that:

The tail of an optimal policy is optimal for the "tail" problem.

This principle is not an actual claim, but rather a guiding principle that can be applied in different ways to each problem. For example, considering our finite-horizon problem, let $\pi^* = (\pi_0, \ldots, \pi_{T-1})$ denote an optimal Markov policy. Take any state $s_t = s'$ which has a positive probability to be reached under π^*, namely $P^{\pi^*,s_0}(s_t = s') > 0$. Then the tail policy $\pi_{t:T}^* = (\pi_t, \ldots, \pi_{T-1})$ is optimal for the "tail" criterion $J_{t:T}^{\pi}(s') = E^{\pi}\left(\sum_{k=t}^{T} R_k | s_t = s'\right)$.

3.5.2 Dynamic Programming for Policy Evaluation

As a "warmup", let us evaluate the reward of a given policy. Let $\pi = (\pi_0, \ldots, \pi_{T-1})$ be a given Markov policy. Define the following reward-to-go function, or value function:

$$V_k^{\pi}(s) = E^{\pi}\left(\sum_{t=k}^{T} R_t | s_k = s\right)$$

Observe that $V_0^\pi(s_0) = J^\pi(s_0)$.

Lemma 3.1 (**Value Iteration**). *$V_k^\pi(s)$ may be computed by the backward recursion:*

$$V_k^\pi(s) = \left\{ r_k(s,a) + \sum\nolimits_{s' \in S_{k+1}} p_k(s'|s,a)\, V_{k+1}^\pi(s') \right\}_{a=\pi_k(s)}, \quad s \in S_k$$

for $k = T-1, \ldots, 0$, starting with $V_T^\pi(s) = r_T(s)$.

Proof. Observe that:

$$\begin{aligned} V_k^\pi(s) &= E^\pi \left(R_k + \sum\nolimits_{t=k+1}^{T} R_t |\ s_k = s, a_k = \pi_k(s) \right) \\ &= E^\pi \left(E^\pi \left(R_k + \sum\nolimits_{t=k+1}^{T} R_t |\ s_k = s, a_k = \pi_k(s), s_{k+1} \right) | s_k = s, a_k = \pi_k(s) \right) \\ &= E^\pi \left(r_k(s_k, a_k) + V_{k+1}^\pi(s_{k+1}) | s_k = s, a_k = \pi_k(s) \right) \\ &= r_k(s, \pi_k(s)) + \sum\nolimits_{s' \in S_{k+1}} p_k(s'|s, \pi_k(s))\, V_{k+1}^\pi(s') \end{aligned}$$

□

Remarks:

- Note that $\sum_{s' \in S_{k+1}} p_k(s'|s,a)\, V_{k+1}^\pi(s') = E^\pi(V_{k+1}^\pi(s_{k+1})|s_k = s, a_k = a)$.
- For the more general reward function $\tilde{r}_t(s,a,s')$, the recursion takes the form

$$V_k^\pi(s) = \left\{ \sum\nolimits_{s' \in S_{k+1}} p_k(s'|s,a)[\tilde{r}_k(s,a,s') + V_{k+1}^\pi(s')] \right\}_{a=\pi_k(s)}$$

A similar observation applies to all Dynamic Programming equations below.

3.5.3 Dynamic Programming for Policy Optimization

We next define the optimal value function at each time $k \geq 0$:

$$V_k(s) = \max_{\pi^k} E^{\pi^k} \left(\sum\nolimits_{t=k}^{T} R_t | s_k = s \right), \quad s \in S_k$$

The maximum is taken over "tail" policies $\pi^k = (\pi_k, \ldots, \pi_{T-1})$ that start from time k. Note that π^k is allowed to be a general policy, i.e., history-dependent and stochastic. Obviously, $V_0(s_0) = J^*(s_0)$.

Theorem 3.3 (**Finite-horizon Dynamic Programming**). *The following holds:*

1. *Backward recursion: Set $V_T(s) = r_T(s)$ for $s \in S_T$.*
   For $k = T-1, \ldots, 0$, $V_k(s)$ may be computed using the following recursion:

$$V_k(s) = \max_{a \in A_k} \left\{ r_k(s,a) + \sum\nolimits_{s' \in S_{k+1}} p_k(s'|s,a)\, V_{k+1}(s') \right\}, \quad s \in S_k.$$

2. *Optimal policy: Any Markov policy π^* that satisfies, for $t = 0, \ldots, T-1$,*

$$\pi_t^*(s) \in \arg\max_{a \in A_t} \left\{ r_t(s,a) + \sum\nolimits_{s' \in S_{t+1}} p_t(s'|s,a)\, V_{t+1}(s') \right\}, \quad s \in S_t,$$

   is an optimal control policy. Furthermore, π^* maximizes $J^\pi(s_0)$ simultaneously for every initial state $s_0 \in S_0$.

Note that Theorem 3.3 specifies an optimal control policy which is a deterministic Markov policy.

Proof. **Part (i):**

We use induction to show that the stated backward recursion indeed yields the optimal value function. The idea is simple, but some care is needed with the notation since we consider general policies, and not just Markov policies. The equality $V_T(s) = r_T(s)$ follows directly from the definition of V_T.

We proceed by backward induction. Suppose that $V_{k+1}(s)$ is the optimal value function for time $k+1$. We need to show that $V_k(s) = W_k(s)$, where

$$W_k(s) \triangleq \max_{a \in A_k} \left\{ r_k(s,a) + \sum\nolimits_{s' \in S_{k+1}} p_k(s'|s,a)\, V_{k+1}(s') \right\}.$$

We will first establish that $V_k(s) \geq W_k(s)$, and then that $V_k(s) \leq W_k(s)$.

(a) We first show that $V_k(s) \geq W_k(s)$. For that purpose, it is enough to find a policy π^k so that $V_k^{\pi^k}(s) = W_k(s)$.

Fix $s \in S_k$, and define π^k as follows: Choose $a_k = \bar{a}$, where

$$\bar{a} \in \arg\max_{a \in A_k} \left\{ r_k(s,a) + \sum\nolimits_{s' \in S_{k+1}} p_k(s'|s,a)\, V_{k+1}(s') \right\},$$

and then, after observing $s_{k+1} = s'$, proceed with the optimal tail policy $\pi^{k+1}(s')$ that obtains $V_{k+1}^{\pi^{k+1}(s')}(s') = V_{k+1}(s')$. Proceeding similarly to Subsection 3.5.2 above

(value iteration for a fixed policy), we obtain:

$$V_k^{\pi^k}(s) = r_k(s,\bar{a}) + \sum\nolimits_{s'\in S_{k+1}} p(s'|s,\bar{a})\, V_{k+1}^{\pi^{k+1}(s')}(s') \tag{3.2}$$

$$= r_k(s,\bar{a}) + \sum\nolimits_{s'\in S_{k+1}} p(s'|s,\bar{a})\, V_{k+1}(s') = W_k(s), \tag{3.3}$$

as was required.

(b) To establish $V_k(s) \le W_k(s)$, it is enough to show that $V_k^{\pi^k}(s) \le W_k(s)$ for any (general, randomized) "tail" policy π^k.

Fix $s \in S_k$. Consider then some tail policy $\pi^k = (\pi_k, \dots \pi_{T-1})$. Note that this means that $a_t \sim \pi_t(a|h_{k:t})$, where $h_{k:t} = (s_k, a_k, s_{k+1}, a_{k+1}, \dots, s_t)$. For each state-action pair $s \in S_k$ and $a \in A_k$, let $(\pi^k|s,a)$ denote the tail policy π^{k+1} from time $k+1$ onwards which is obtained from π^k given that $s_k = s,\ a_k = a$. As before, by value iteration for a fixed policy,

$$V_k^{\pi^k}(s) = \sum\nolimits_{a\in A_k} \pi_k(a|s) \left\{ r_k(s,a) + \sum\nolimits_{s'\in S_{k+1}} p_k(s'|s,a)\, V_{k+1}^{(\pi^k|s,a)}(s') \right\}.$$

But since V_{k+1} is optimal,

$$V_k^{\pi^k}(s) \le \sum\nolimits_{a\in A_k} \pi_k(a|s) \left\{ r_k(s,a) + \sum\nolimits_{s'\in S_{k+1}} p_k(s'|s,a)\, V_{k+1}(s') \right\}$$

$$\le \max_{a\in A_k} \left\{ r_k(s,a) + \sum\nolimits_{s'\in S_{k+1}} p_k(s'|s,a)\, V_{k+1}(s') \right\} = W_k(s),$$

which is the required inequality in (b).

Part (ii) (Outline - **exercise**):

Let π^* be the (Markov) policy defined in part 2 of Theorem 3.3. Using value iteration for this policy, prove by backward induction that $V_k^{\pi^*} = V_k$. □

To summarize:

- The optimal value function can be computed by backward recursion. This recursive equation is known as the *dynamic programming equation*, *optimality equation*, or *Bellman's Equation*.
- Computation of the value function in this way is known as the *finite-horizon value iteration* algorithm.
- The value function is computed for all states at each stage.

- An optimal policy is easily derived from the optimal value.
- The optimization in each stage is performed in the action space. The total number of minimization operations needed is $T \times |S|$ - each over $|A|$ choices. This replaces "brute force" optimization in policy space, with tremendous computational savings as the number of Markov policies is $|A|^{T \times |S|}$.

3.5.4 The Q function

Let

$$Q_k(s,a) \stackrel{\Delta}{=} r_k(s,a) + \sum\nolimits_{s' \in S_k} p_k(s'|s,a)\, V_k(s').$$

This is known as the optimal state-action value function, or simply as the *Q-function*. $Q_k(s,a)$ is the expected return from stage k onward, if we choose $a_k = a$ and then proceed optimally.

Theorem 3.3 can now be succinctly expressed as

$$V_k(s) = \max_{a \in A_k} Q_k(s,a),$$

and

$$\pi_k^*(s) \in \arg\max_{a \in A_k} Q_k(s,a).$$

The Q function provides the basis for the Q-learning algorithm, which is one of the basic Reinforcement Learning algorithms.