Markov Decision Processes

Our aim in this series of lectures is to extend our suite of heuristic search and optimization algorithms to the case where transitions to successor states are stochastic rather than deterministic: i.e., to problems characterized as sequential decision making under uncertainty. For example, we might be interested in planning an optimal route to work, or optimizing our blackjack strategy, or maximizing the returns on a portfolio of investments in the stock market. All three of these examples involve stochasticity, as traffic, cards dealt, and stock prices are all unpredictable; but stochastic models of traffic and stock prices can be built from data, while probabilities over cards dealt can be deduced via combinatorics.

In these lectures, we introduce **Markov reward processes (MRPs)** and **Markov decision processes (MDPs)** as modeling tools in the study of sequential decision making under uncertainty. These models provide frameworks for predicting long-term rewards and computing optimal behavior in uncertain worlds. Solutions to MRPs and MDPs may involve linear programming or dynamic programming methods (e.g., value iteration and policy iteration) when the stochastic model is known and the state and decision spaces are sufficiently small; otherwise, they can be solved using Monte Carlo simulations or reinforcement learning (e.g., TD-learning, Q-learning, and SARSA).

Our discussion of Markov processes is divided into two parts: the first part is concerned with computing state values V in Markov reward (or decision) processes, and the second with computing action values Q in Markov decision processes. This division coincides with two related problems, namely:

1. **(passive) prediction**, or **policy evaluation**: compute the state-value function V^π, given policy π
2. **(active) control**: find an optimal policy π^*, by computing the optimal action-value function Q^*

1 Definitions and An Example

A **stochastic process** is a sequence of random variables $\{X_t\}_{t=0}^\infty$. A stochastic process $\{X_t\}_{t=0}^\infty$ induces a probability transition function $P[X_{t+1} = s_{t+1} \mid X_t = s_t, \ldots, X_0 = s_0]$: i.e., the probability that the state at future time $t+1$ is s_{t+1}, given that the states at past times $t, \ldots, 0$ were $s_t, \ldots, s_0$, respectively.

A **Markov process** is a stochastic process *s.t.* for all t, for all $s_0, \ldots, s_t, s_{t+1}$,

$$P[X_{t+1} = s_{t+1} \mid X_t = s_t, \ldots, X_0 = s_0] = P[X_{t+1} = s_{t+1} \mid X_t = s_t] \tag{1}$$

Equation 1 is the **Markov property**, sometimes called the memoryless property. It states that the future is independent of the past, given the present. Mathematically, probability transitions to future states, such as s_{t+1}, are conditionally independent of the past, $s_{t-1}, \ldots, s_0$, given the present state s_t.

A **time-homogeneous** Markov process is one where the probability of transitioning from one state to another in n steps is always the same, regardless of absolute time: for all times t, t' and states s, s',

$$P[X_{t+n} = s' \mid X_t = s] = P[X_{t'+n} = s' \mid X_{t'} = s] \tag{2}$$

A time-homogeneous Markov process over a discrete state space is called a **Markov chain**. Markov chains can be represented as row- (or column-)stochastic probability transition matrices.

1.1 Markov Reward Processes

An agent operating in a stochastic environment transitions from state to state, in general obtaining rewards along the way, as follows: at time t,

1. state is s_t
2. receive reward r_t
3. transition to state s_{t+1} with probability $P[s_{t+1} \mid s_t, \ldots, s_0]$

We model this agent's interactions as a (discrete-time) **Markov reward process**, a tuple $\langle S, R, P \rangle$, where time is discrete: i.e., $t \in T = \{0, 1, \ldots\}$, and

- S is a finite set of states
- $R : S \to \mathbb{R}$ is a reward function
- $P : S \to \Delta(S)$ is a probability transition function (or matrix)
  $\Delta(S)$ is the set of probability distributions over S

Implicit in this definition is the assumption that the probability transition function P is a Markov chain.

N.B. Markov reward processes can have stochastic rewards as well as stochastic transitions. Our framework is nonetheless sufficiently general, because MRPs with stochastic rewards can be reduced to MRPs processes with deterministic rewards by setting the deterministic rewards equal to the expected stochastic rewards.

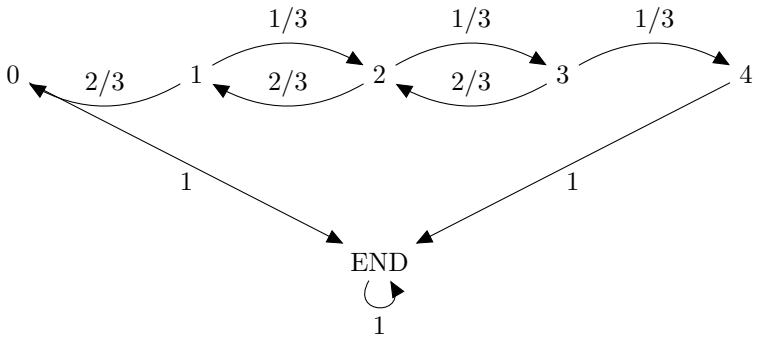

Figure 1: Gambler's Ruin: $N = 4$. An *absorbing state* $s \in \mathcal{S}$ is *s.t.* $P[s \mid s] = 1$. The END state is an absorbing state in Gambler's Ruin.

Example *Gambler's Ruin* is an example of a Markov reward process. A gambler gambles until he either wins a set amount of money, say $\$N$, or loses all his money. At state s_t, his wealth increases by \$1 with probability 1/3, and it decreases by \$1 with probability 2/3.

The set of states is defined by the worth of the gambler: $S = \{0, \ldots, N, \text{END}\}$. The transition probabilities are s.t. $P[i+1|i] = 1/3$ and $P[i-1|i] = 2/3$, for $i = 1, \ldots, N-1$, $P[\text{END}|i] = 0$, for $i = 0, N$, and $P[\text{END} \mid \text{END}] = 1$. What is the probability that the gambler wins, i.e., reaches state N?

2 State Values

Being a stochastic process, there are many trajectories that can be realized in a Markov reward process, with various probabilities and rewards. The expected value over the rewards of all trajectories originating at a state s is called the **state-value function** $V : \mathcal{S} \to \mathbb{R}$. Given a Markov reward process, such as the Gambler's ruin, we are interested in computing V.

We write S_t to represent the random variable[1] denoting the state at time t: e.g., $S_t = s$, for some $s \in \mathcal{S}$. We then write $\tau_t = (S_t, S_{t+1}, \ldots)$ to denote the random trajectory originating at state S_t at time t, and $G_t^\tau = \sum_{i=0}^{\infty} R_{t+i}^\tau$ to denote the return accrued along trajectory τ: i.e., the sum of all the rewards from time t on, where $R_{t+i}^\tau = R(S_{t+i})$, for all $i \in \mathbb{N}$. The expected value of G_t^τ is then $V(s)$, where the expectation is taken over all trajectories τ that initiate at s: i.e., $V(s) = \mathbb{E}_\tau \left[G_t^\tau \mid S_t = s \right]$.

In this section, we present a proof sketch of Bellman's seminal theorem, known as **Bellman's equation**, namely the state value $V(s_t)$ can be decomposed recursively into the sum of the immediate reward obtained at time t and the discounted sum of the expected future rewards obtained thereafter (i.e., $V(S_{t+1})$).

2.1 But First a Word on Return

Given trajectory $\tau = (S_t, S_{t+1}, S_{t+2}, \ldots)$, the return G_t^τ is a function of the current reward R_t and the stream of future rewards $R_{t+1}, R_{t+2}, \ldots$. In the case of a finite horizon, say of length $T < \infty$, return can be computed simply as the sum of current and future rewards: i.e.,

$$G_t^\tau = R_t + R_{t+1} + R_{t+2} + \ldots + R_T = \sum_{i=0}^{T-t} R_{t+i} \tag{3}$$

In the case of infinite horizons, however, the sum of future rewards is potentially infinite. If all trajectories are **proper** (a trajectory is called a **proper** trajectory iff it eventually transitions to a zero-reward, absorbing state with positive probability), return can be computed simply as the sum of current and future rewards, as in Equation 3. Otherwise, return is computed as the sum of current rewards and the **discounted** sum of future rewards: i.e., assuming discount factor $0 \leq \gamma < 1$,

$$G_t^\tau = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots = \sum_{i=0}^{\infty} \gamma^i R_{t+i} \tag{4}$$

If rewards are assumed to be bounded, return as defined by Equation 4 is finite. (**Exercise**) But even in the case of finite horizons or proper trajectories, G_t^τ is often computed with discounting, because of the following economic intuition.

The motivation for discounting future rewards can be simply stated: *a dollar today is worth more than a dollar tomorrow.* For example, given an interest rate of $x\%$ per annum, d dollars today are worth $(1 + x)d$ dollars 365 days from now. Thus, d dollars 365 days from now are worth only $d/(1 + x)$ dollars today. The discount factor γ is inversely related to the interest rate: $\gamma = 1/(1 + x)$.

Intuitively, γ determines the relative worth of immediate vs. future rewards. As $\gamma \to 0$, immediate rewards are deemed more and more relevant; agents that attempt to maximize return in these circumstances are called **myopic**. As $\gamma \to 1$, future rewards are weighted more and more heavily; agents that aim to maximize discounted rewards based on high values of γ exhibit **foresight**.

[1] As usual, we use capital letters to denote random variables, and lower case letters to denote their realized values.

2.2 Bellman's Theorem

We now derive Bellman's theorem for Markov reward processes: the state value $V(s_t)$ at state s_t is the sum of the immediate reward obtained at time t and the discounted sum of the expected rewards obtained thereafter: i.e., for $0 \leq \gamma < 1$,

$$V(s_t) = r_t + \gamma \mathbb{E}_{s_{t+1}}[V(s_{t+1})] \tag{5}$$

The crux of the proof of this theorem is roughly as follows: for all states $s \in \mathcal{S}$,

$$\begin{aligned}
V(s) &= \mathbb{E}_\tau \left[G_t^\tau \mid S_t = s \right] \\
&= \int_\tau P[\tau \mid S_t = s] G_t^\tau \\
&= r_t + \gamma \int_\tau P[\tau \mid S_t = s] G_{t+1}^\tau \\
&= r_t + \gamma \sum_{s' \in \mathcal{S}} P[S_{t+1} = s' \mid S_t = s] \left(\int_{\tau'} P[\tau' \mid S_{t+1} = s', S_t = s] G_{t+1}^{\tau'} \right) \\
&= r_t + \gamma \sum_{s' \in \mathcal{S}} P[S_{t+1} = s' \mid S_t = s] \left(\int_{\tau'} P[\tau' \mid S_{t+1} = s'] G_{t+1}^{\tau'} \right) \\
&= r_t + \gamma \sum_{s' \in \mathcal{S}} P[S_{t+1} = s' \mid S_t = s] \mathbb{E}_{\tau'} \left[G_{t+1}^{\tau'} \mid S_{t+1} = s' \right] \\
&= r_t + \gamma \sum_{s' \in \mathcal{S}} P[S_{t+1} = s' \mid S_t = s] V(s') \\
&= r_t + \gamma \mathbb{E}_{s'}[V(s')]
\end{aligned}$$

The first equality follows from the definition of V; the second, from the meaning of an expectation; the third, from the fact that r_t is not a random quantity; the fourth, from the definition of joint probability; the fifth, from the Markov property; the sixth, from the meaning of an expectation (again); the seventh, from the definition of V (again); and the last line is simply an abbreviation.

2.3 Bellman's Equations

Bellman's theorem gives rise to the following system of $|S|$ equations with $|S|$ unknowns, known as Bellman's equations: for all states $s \in \mathcal{S}$,

$$V(s) = r(s) + \gamma \sum_{s'} P[s' \mid s] V(s') \tag{6}$$

Assuming a finite state state, with n states, we can rewrite this equation in matrix notation as follows: for value and reward vectors $v, r \in \mathbb{R}^n$ and $n \times n$ transition probability matrix P,

$$v = r + \gamma P v \tag{7}$$

$$v - \gamma P v = r \tag{8}$$

$$Iv - \gamma P v = r \tag{9}$$

$$(I - \gamma P) v = r \tag{10}$$

$$v = (I - \gamma P)^{-1} r \tag{11}$$

Therefore, Bellman's equation can be solved by matrix inversion. But just like in linear regression, where we tend to prefer least-squares solutions to computing the closed-form solution directly, an iterative method

is the norm for solving Bellman's equations. These iterative methods are preferable when the system of equations is large, making matrix inversion expensive, and because iterative methods are more robust to numerical approximation errors.

To solve Bellman's equations, we rely on Banach's fixed point theorem, also called the contraction mapping theorem. Given a metric space[2] (X, d), a mapping $f : X \to X$ is called a *contraction* iff there exists some $0 \leq k < 1$ *s.t.* $d(f(x), f(y)) \leq kd(x, y)$, for all $x, y \in X$.

Theorem [Banach] Given a complete[3] metric space (X, d) and a contraction mapping $f : X \to X$, (i) there exists a unique $x^* \in X$ *s.t.* $f(x^*) = x^*$; and (ii) for arbitrary $x^0 \in X$, the sequence $\{x^n\}$ defined by $x^{n+1} = f(x^n) = f^{n+1}(x^0)$ converges to x^*.

Define the mapping $f : \mathbb{R}^S \to \mathbb{R}^S$ as follows:

$$(f(x))(s) = r(s) + \gamma \sum_{s'} P[s' \mid s]x(s') \tag{12}$$

Theorem The mapping f is a contraction on $(\mathbb{R}^S, L_\infty)$.

Proof Let the metric d be the L_∞, or max, norm: i.e., $||x - y|| = \max_i |x_i - y_i|$. For all $x, y \in X$, and for arbitrary state $s \in \mathcal{S}$,

$$\begin{aligned}
& |(f(x))(s) - (f(y))(s)| \\
= & \left| r(s) + \gamma \sum_{s'} P[s' \mid s]x(s') - \left(r(s) + \gamma \sum_{s'} P[s' \mid s]y(s') \right) \right| \\
= & \gamma \sum_{s'} P[s' \mid s] \, |x(s') - y(s')| \\
\leq & \gamma \sum_{s'} P[s' \mid s] \max_{s''} |x(s'') - y(s'')| \\
= & \gamma \sum_{s'} P[s' \mid s] ||x - y|| \\
= & \gamma ||x - y||
\end{aligned}$$

It follows that $|(f(x))(s) - (f(y))(s)| \leq \gamma ||x - y||$, for all states s. Therefore, $||f(x) - f(y)|| = \max_s |(f(x))(s) - (f(y))(s)| \leq \gamma ||x - y||$.

Corollary Bellman's system of equations (Equation 6) indeed has a fixed point solution, and the iterative application of f converges to this solution.

3 Policy Evaluation

Policy evaluation is a dynamic programming method that computes state values via iterative updates based on Bellman's equations:

$$V(s) \leftarrow r(s) + \gamma \sum_{s'} P[s' \mid s]V(s') \tag{13}$$

Gauss-Seidel's version of this algorithm incorporates in-place updating: i.e., updating with V, as shown in Table 2, rather than V', as shown in Table 1.

[2] A metric space (X, d) is a set X together with a distance function $d : X \times X \to \mathbb{R}$ that satisfies: (i) $d(x, x) = 0$, for all $x \in X$; (ii) $d(x, y) = d(y, x)$ for all $x, y \in X$; and (iii) the triangle inequality—$d(x, z) \leq d(x, y) + d(y, z)$ for all $x, y, z \in X$.

[3] An example of a complete metric space is $\mathbb{R}$.

POLICY_EVALUATION(MRP, γ, ϵ)	
Inputs	discount factor γ
	convergence test ϵ
Output	state-value function V
Initialize	$V = 0$ and $V' = \infty$

`while` $\max_s |V(s) - V'(s)| > \epsilon$ `do`

1. $V' = V$
2. for all $s \in \mathcal{S}$
   (a) $V(s) = r(s) + \gamma \sum_{s'} P[s' \mid s] V'(s')$

`return` V

Table 1: Policy Evaluation.

GAUSS_SEIDEL(MRP, γ, ϵ)	
Inputs	discount factor γ
	convergence test ϵ
Output	state-value function V
Initialize	$V = 0$ and $V' = \infty$

`while` $\max_s |V(s) - V'(s)| > \epsilon$ `do`

1. $V' = V$
2. for all $s \in \mathcal{S}$
   (a) $V(s) = r(s) + \gamma \sum_{s'} P[s' \mid s] V(s')$

`return` V

Table 2: Gauss-Seidel.

3.1 Example: Gambler's Ruin

In the Gambler's Ruin problem, we are interested in computing the probability that the gambler is ruined (or not). To compute this probability, we model the problem as a Markov reward process with rewards 0 everywhere except at state N, where the reward is 1. In this way, the value of a state s, which represents the expected value of all returns on trajectories emanating from s, is the total probability of all trajectories leading to a win times a reward of 1, plus the total probability of all trajectories leading to a lose times a reward of 0, which equals the total probability of all trajectories leading to a 1.] The value function of this MRP therefore represents the probability that the gambler is *not* ruined.

Assuming $\gamma = 1$, these values can be computed via policy evaluation as follows:

V	0	1	2	3	4	END
0	0	0	0	0	0	0
1	0	0	0	0	1	0
2	0	0	0	$\frac{1}{3}$	1	0
3	0	0	$\frac{1}{9}$	$\frac{1}{3}$	1	0
4	0	$\frac{1}{27}$	$\frac{1}{9}$	$\frac{11}{27}$	1	0
5	0	$\frac{1}{27}$	$\frac{13}{81}$	$\frac{11}{27}$	1	0

With in-place computation *á la* Gauss-Seidel, working backwards from END to state 4 down to state 0, the computation proceeds as follows:

V	0	1	2	3	4	END
0	0	0	0	0	0	0
1	0	$\frac{1}{27}$	$\frac{1}{9}$	$\frac{1}{3}$	1	0
2	0	$\frac{13}{243}$	$\frac{13}{81}$	$\frac{11}{27}$	1	0
3	0	$\frac{133}{2187}$	$\frac{133}{729}$	$\frac{107}{243}$	1	0
100	0	0.0667	0.2	0.4667	1	0

At all states $1, \ldots, N-1$, the gambler is more likely to be ruined than not.