

Lecture 19: Markov Decision Processes: Prediction

10:30 AM, Apr 7, 2009

Contents

1 Overview	1
2 Definitions and An Example	2
2.1 Markov Reward Processes	2
3 State Values	3
3.1 Return	3
3.2 Bellman's Theorem	4
3.3 Bellman's Equations	5
4 Policy Evaluation	6
4.1 Example: Gambler's Ruin	6

1 Overview

Our aim in this series of lectures is to extend our suite of heuristic search and optimization algorithms to the case in which transitions to successor states are non-deterministic. We introduce Markov reward processes (MRPs) and Markov decision processes (MDPs) as modeling tools in the study of non-deterministic state-space search problems. These models provide frameworks for computing optimal behavior in uncertain worlds. For example, we might be interested in planning an optimal route to work, or maximizing the returns on an investment in the stock market. Solutions may involve linear programming or dynamic programming methods (*e.g.*, value iteration and policy iteration) in the case where the non-deterministic nature of the process is known with certainty and the state and action spaces are sufficiently small; otherwise, they may 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 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. the **(passive) prediction problem**, or **policy evaluation**: compute the state-value function V^π , given policy π
2. the **(active) control problem**: find an optimal policy π^* , by computing the optimal action-value function Q^*

This series of lectures (Lectures 19-22) are primarily based on Chapters 4, 5, and 6 of Sutton and Barto's book entitled *Reinforcement Learning*.

2 Definitions and An Example

A (discrete-time) **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 of the form $P[X_{t+1} = s_{t+1} \mid X_t = s_t, \dots, X_0 = s_0]$: *i.e.*, the probability that the state at future time $t + 1$ is s_{t+1} , given that the states at past times $t, \dots, 0$ were s_t, \dots, s_0 , respectively.

A **Markov process** (or **chain**) is a stochastic process that satisfies the following conditional independence conditions: for all t , for all s_0, \dots, s_t, s_{t+1} ,

$$P[X_{t+1} = s_{t+1} \mid X_t = s_t, \dots, X_0 = s_0] = P[X_{t+1} = s_{t+1} \mid X_t = s_t] \quad (1)$$

Equation 1 is the **Markov property**, sometimes called the memoryless property; it implies that the probability of transitioning to a future state s_{t+1} depends on the present state s_t , but is otherwise independent of the remote past, s_{t-1}, \dots, s_0 .

2.1 Markov Reward Processes

An agent operating in a Markovian 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 $P[s_{t+1} \mid s_t]$

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.*, $t \in T = \{0, 1, \dots\}$, and

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

The form of the probability transition function encodes the fact that it satisfies the Markov property.

Remark: Markov reward processes can have stochastic rewards as well as stochastic transitions. But our framework is sufficiently general, since such processes can be reduced to Markov reward processes with deterministic rewards simply by letting the deterministic rewards equal the expected values of the corresponding stochastic rewards.

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: $S = \{0, \dots, N, \text{END}\}$. The rewards are defined as $R(\text{END}) = R(i) = 0$, for $i = 0, \dots, N - 1$, but $R(N) = 1$. The transition probabilities are such that $P[i + 1 | i] = 1/3$ and $P[i - 1 | i] = 2/3$, for $i = 1, \dots, N - 1$; $P[\text{END} | i] = 1$, for $i = 0, N$; and $P[\text{END} | \text{END}] = 1$.

Given initial state $i \in 1, \dots, N - 1$, what is the probability that the gambler wins: *i.e.*, reaches state N ?

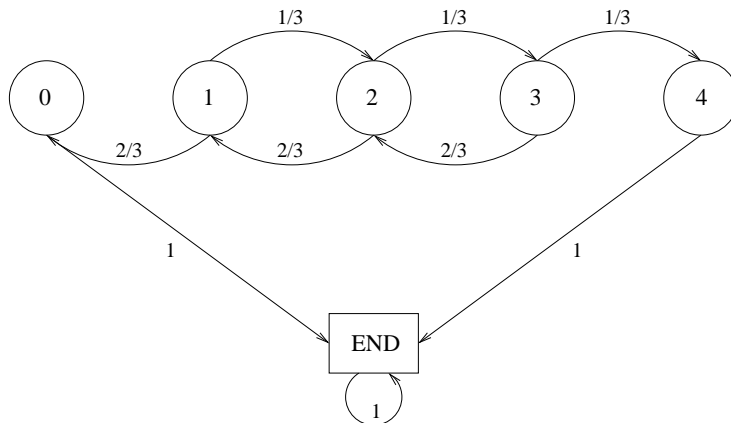


Figure 1: Gambler’s Ruin: $N = 4$. An **absorbing state** $s \in S$ is *s.t.* $P[s | s] = 1$. The END state is an absorbing state in Gambler’s Ruin.

3 State Values

The value $V(s_t)$ of state s_t is defined as the expected reward that is accrued from time t on; that is, the expected value of ρ_t^τ , where ρ_t^τ is the reward (or return) that is accrued along trajectory $\tau = (s_t, s_{t+1}, s_{t+2}, \dots)$:

$$V(s_t) = \sum_{\tau} P[\tau | s_t] \rho_t^\tau \tag{2}$$

3.1 Return

Given trajectory $\tau = (s_t, s_{t+1}, s_{t+2}, \dots)$, the return ρ_t^τ at time t is a function of the current reward r_t and the stream of future rewards r_{t+1}, r_{t+2}, \dots . 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.*,

$$\rho_t^\tau = r_t + r_{t+1} + r_{t+2} + \dots + r_T = \sum_{i=0}^{T-t} r_{t+i} \tag{3}$$

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 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 $0 \leq \gamma < 1$,

$$\rho_t^\tau = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots = \sum_{i=0}^{\infty} \gamma^i r_{t+i} \quad (4)$$

If rewards are assumed to be bounded, return as defined by Equation 4 is finite. (Why?) But even in the case of finite horizons or proper trajectories, ρ_t^τ is often computed with discounting, because of the following economic “law”: *a dollar today is worth more than a dollar tomorrow*.

This law encapsulates the following equation, which states that the future value (FV) of money equals the present value (PV) scaled by the interest rate (r):

$$\text{FV} = \text{PV}(1 + r)$$

For example, $\$d$ today would be worth $\$(1 + r)d$ 1 year from now. Similarly, $\$d$ that are scheduled to be accrued 1 year from now are worth only $\$/(1 + r)$ today.

The **discount factor** γ is inversely related to the interest rate: $\gamma = 1/(1 + r)$. Intuitively, γ determines the relative worth of immediate vs. future rewards. As $\gamma \rightarrow 0$, immediate rewards are deemed more and more relevant; agents that attempt to maximize return in these circumstances are called **myopic**. As $\gamma \rightarrow 1$, future rewards are weighted more and more heavily; agents that aim to maximize return in these circumstances exhibit **foresight**.

3.2 Bellman’s Theorem

We can now state Bellman’s seminal theorem for Markov reward processes (*i.e.*, for state values).

Theorem: The state value $V(s_t)$ at state s_t —which is defined as the expected reward that is accrued from time t on—can be equivalently expressed as the sum of the immediate reward r_t and the discounted expected value of state s_{t+1} : *i.e.*, for $0 \leq \gamma < 1$,

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

Proof: (Sketch) In what follows, $\tau' = (s_{t+1}, s_{t+2}, \dots)$ and $\tau'' = (s_{t+2}, \dots)$.

$$\begin{aligned} V(s_t) &= \sum_{\tau} P[\tau | s_t] \rho_t^\tau \\ &= r_t + \gamma \sum_{\tau'} P[\tau' | s_t] \rho_{t+1}^{\tau'} \\ &= r_t + \gamma \sum_{s_{t+1} \in S} P[s_{t+1} | s_t] \left(r_{t+1} + \gamma \sum_{\tau''} P[\tau'' | s_{t+1}, s_t] \rho_{t+2}^{\tau''} \right) \\ &= r_t + \gamma \sum_{s_{t+1} \in S} P[s_{t+1} | s_t] \left(r_{t+1} + \gamma \sum_{\tau''} P[\tau'' | s_{t+1}] \rho_{t+2}^{\tau''} \right) \\ &= r_t + \gamma \sum_{s_{t+1} \in S} P[s_{t+1} | s_t] \left(\sum_{\tau'} P[\tau' | s_{t+1}] \rho_{t+1}^{\tau'} \right) \\ &= r_t + \gamma \sum_{s_{t+1} \in S} P[s_{t+1} | s_t] V(s_{t+1}) \\ &= r_t + \gamma \mathbb{E}[V(s_{t+1})] \end{aligned}$$

The fourth line follows from the Markov property. The seventh line is simply an abbreviation.

3.3 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 $s \in S$,

$$V(s) = R(s) + \gamma \sum_{s'} P[s' | s] V(s') \quad (6)$$

To find a solution to this system of equations, we rely on Banach's fixed point theorem, also called the contraction mapping theorem. Given a metric space¹ (X, d) , a mapping $f : X \rightarrow X$ 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 $x, y \in X$.

The L_∞ , or **max**, norm is defined as follows on \mathbb{R}^n : for all $x, y \in \mathbb{R}^n$, $\|x - y\| = \max_{1 \leq i \leq n} |x_i - y_i|$. In the special case where $n = 1$, the max norm reduces to absolute value.

Example: The function $f(x) = \frac{1}{2}x$ is a contraction mapping on (\mathbb{R}, L_∞) , since $|f(x) - f(y)| = \left| \frac{1}{2}x - \frac{1}{2}y \right| = \frac{1}{2}|x - y|$.

Banach's Theorem: Given a complete² metric space (X, d) as well as a contraction mapping $f : X \rightarrow X$, (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 $f : \mathbb{R}^S \rightarrow \mathbb{R}^S$ as follows:

$$(f(x))(s) = R(s) + \gamma \sum_{s'} P[s' | s] x(s') \quad (7)$$

Theorem: The mapping f in Equation 7 is a contraction on (\mathbb{R}^S, L_∞) .

Proof: For all $x, y \in X$, and for arbitrary state $s \in S$,

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

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.

¹A metric space (X, d) is a set X together with a distance function $d : X \times X \rightarrow \mathbb{R}$ 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 $x, y, z \in X$.

²An example of a complete metric space is \mathbb{R} .

4 Policy Evaluation

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

$$V(s) \leftarrow R(s) + \gamma \sum_{s'} P[s' | s] V(s') \quad (8)$$

The Gauss-Seidel 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.

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 $s \in S$
(a)	$V(s) = R(s) + \gamma \sum_{s'} P[s' s] V'(s')$
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 $s \in S$
(a)	$V(s) = R(s) + \gamma \sum_{s'} P[s' s] V(s')$
return V	

Table 2: Gauss-Seidel.

4.1 Example: Gambler's Ruin

Assuming $\gamma = 1$, the extent of the gambler's ruin is indicated by the state values in the table below. These values are 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 *à 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

Since rewards are 0 everywhere unless the gambler is *not* ruined, in which case rewards are 1, the final state values can be interpreted as the probability that the gambler is *not* ruined. At all states $1, \dots, N - 1$, the gambler is more likely to be ruined than not.