7.2. Reinforcement Learning

7.2.1. Markov Decision Processes (MDP)

Recapitulation MDP:

$$(S, A, \{P_{sa}\}, \gamma, R)$$

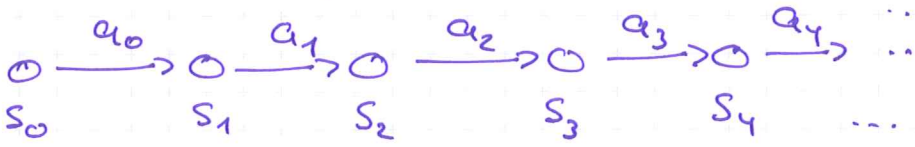
- S: states
- A: actions
- $\{P_{sa}\}$: transition prob.
- γ: discountfactor
- R: reward function

Payoff: $R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots$

Objective:

$$\max_{\text{actions} \in A} E[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots]$$

Dynamics of the process:



Note that $s_0, s_1, s_2, \dots$ are random variables.

Policy is a function

$$\pi : S \to A : s \mapsto \pi(s)$$

In state $s \in S$ take action $a = \pi(s)$

Value function for a policy π

$$V^\pi(s) = E\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots \mid s_0 = s, \pi \right]$$

Expected total payoff upon starting in s, applying policy π.

For any policy π the Bellman equations hold:

$$V^\pi(s) = R(s) + \gamma \sum_{s' \in S} P_{s,\pi(s)}(s')\, V^\pi(s'), \quad s \in S \qquad (BE)$$

RHS immediate reward $R(s)$ + expected sum of future discounted rewards.

Solving for V^π for fixed policy π means to solve $|S|$ linear equations with $|S|$ variables provided $|S| < \infty$, $S = \{1, \dots, m\}$.

Write $\underline{R} = (R(1), \dots, R(m))^T \in \mathbb{R}^m$

$$\Psi^\pi = \begin{pmatrix} P_{1,\pi(1)} \\ \vdots \\ P_{m,\pi(m)} \end{pmatrix} \in \mathbb{R}^{m \times m}$$

$$\underline{V}^\pi = (V^\pi(1), \dots, V^\pi(m))^T \in \mathbb{R}^m \text{ (variables)}$$

Then (BE) in matrix form

$$\underline{V}^{\pi} = \underline{R} + \gamma \Psi^{\pi} \underline{V}^{\pi}$$

$$\Leftrightarrow \quad \underline{V}^{\pi} = (\underline{I}_m - \gamma \Psi^{\pi})^{-1} \underline{R}$$

provided the inverse exists.

Optimal value function:

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

$$= \max_{\pi} E\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots \mid s_0 = s, \pi\right]$$

Max. expected payoff over all policies π starting in $s \in S$.

It holds, analogous to (BE)

$$V^*(s) = R(s) + \max_{a \in A} \sum_{s' \in S} P_{s,a}(s') V^*(s') \qquad (*)$$

Define a partial ordering over policies.

$$\pi \geq \pi' \text{ if } V^{\pi}(s) \geq V^{\pi'}(s) \text{ for all } s \in S.$$

Theorem 7.1

a) There exists an optimal policy π^* with

$$\pi^* \geq \pi \text{ for all policies } \pi.$$

b) All optimal policies achieve the same value

$$V^{\pi^*}(s) = V^*(s).$$

⊥

7-2-2. Computing the optimum policy

Algorithm policy iteration (PI) to find the opt. π^*:

(PI) 1. Initialize π randomly

2. Repeat until convergence {

(a) $\underline{V} := \underline{V}^{\pi}$

(b) For each $s \in S$, ~~let~~

$$\pi(s) = \arg\max_{a \in A} \sum_{s' \in S} P_{sa}(s') V(s')$$

}

(a) For fixed policy π, $\underline{V}^{\pi}$ can be computed from (BE), see 7.2.1.

(b) Note that $V(s')$ is w.r.t. to actual policy π. Update of π is greedy w.r.t. $\underline{V}$.

After a finite number of iterations $\underline{V}$ converges to $\underline{V}^*$ and π to π^*.

7.2.3 Model Learning for MDP

In practice trans. prob. $P_{s,a}$ and sometimes the reward fct. R are unknown.

Solution: estimate $P_{s,a}(s')$, $s' \in S$, and $R(s')$ from data.

Carry out trials / experiments / simulations:

$$s_0^{(\ell)} \xrightarrow{a_0^{(\ell)}} s_1^{(\ell)} \xrightarrow{a_1^{(\ell)}} s_2^{(\ell)} \xrightarrow{a_2^{(\ell)}} \dots, \quad \ell = 1, 2, \dots$$

ℓ counts the no. of trials.

Estimate

$$P_{s,a}(s') = \frac{\#\text{ times took action } a \text{ in state } s \text{ and got to } s'}{\#\text{ times took action } a \text{ in state } s}$$

In case $\frac{0}{0}$ set $P_{s,a}(s') = \frac{1}{|S|}$ (uniform distribution)

Similarly $R(s)$ can be estimated from observed data.

Possible algorithm for learning a MDP with unknown $P_{s,a}$

1. Initialize π randomly
2. Repeat {
   (a) Execute π in the MDP for some no. of trials
   (b) Upgrade estimates $P_{s,a}$ (and potentially R)
   (c) Update π by algorithm (PI)
   }