Prediction & control

Evaluate the future
- given a policy

Optimise the future
- Find the best policy

Lec-2 Markov Decision Process (MDPs)

- MDP formally describes an environment for RL, where the env. is fully observable.
  i.e. the current state fully characterizes the process
- Almost all RL problems can be formalized as MDPs
- For a Markov state s and successor state s', the state transition prob. is defined by

$$P_{ss'} = P(S_{t+1} = s' \mid S_t = s)$$

State transition matrix

$$P = \begin{array}{cc} & \text{to} \\ \text{from} & \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & & \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} \end{array}$$

Each row sums to 1.

- A Markov process is a memoryless random process, i.e. a sequence of random states $S_1, S_2, \ldots$ with Markov property.

A Markov process is a tuple $\langle \mathcal{S}, \mathcal{P} \rangle$ with

* $\mathcal{S}$ is a (finite) set of states
* $\mathcal{P}$ is state-transition prob. matrix

$$\mathcal{P}_{ss'} = P(S_{t+1} = s' \mid S_t = s)$$

- An episode is a finite sequence of Markov states when agent-env. interaction breaks naturally.

Markov Reward Process

- Markov chain with values
- MRP is a tupe $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

  * $\mathcal{S}$ set of states
  * $\mathcal{P}$ state transition matrix
  * $\mathcal{R}$ reward fun^n

  $$\mathcal{R}_s = \mathbb{E}[R_{t+1} \mid S_t = s]$$

  * γ is discount factor, $0 \leq \gamma \leq 1$

Return

→ Return G_t is total discounted return from time-step t

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

→ γ is present value of future rewards

γ trades off b/w immediate vs. long term rewards

* Using γ provides
  - Mathematical convenient to discount rewards
  - Avoid infinite returns in cyclic Markov processes
  - Uncertainty about the future is better represented

Value function of MRP

Gives the long-term value of state. State-value funⁿ of MRP

$$v(s) = \mathbb{E}\left[G_t \mid S_t = s\right]$$

$$= \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid S_t = s\right]$$

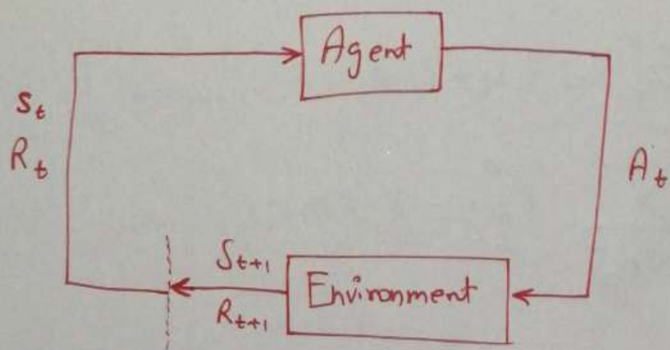
$$= \mathbb{E}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right]$$

$$= \mathbb{E}\left[R_{t+1} + \gamma\, \mathbb{E}\left[G_{t+1} \mid S_t = s\right] \mid S_t = s\right]$$

from law of iterated expectation

$$\boxed{v(s) = \mathbb{E}\left[R_{t+1} + \gamma\, v(S_{t+1}) \mid S_t = s\right]}$$

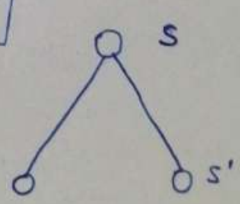
Bellman Equation for MRPs



Agent Environment Interaction in a MDP

$$v(s) = \mathbb{E}\left[R_{t+1} \mid S_t = s\right] + \gamma\, \mathbb{E}\left[v(S_{t+1}) \mid S_t = s\right]$$

$$\boxed{v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}\, v(s')}$$



Bellman eqⁿ for MRPs

$$v = R + \gamma P v$$

$$\Rightarrow \boxed{v = (I - \gamma P)^{-1} R}$$

Markov Decision Process

An MDP is a tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$

* $\mathcal{S}$ is a (finite) set of states
* $\mathcal{A}$ is a (finite) set of actions
* P is state transition prob matrix

$$P_{ss'}^{a} = P(S_{t+1} = s' \mid S_t = s, A_t = a)$$

* R is a reward function

$$R_s^a = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$$

* γ is a discount factor $\gamma \in [0,1]$

Let

$$r(s,a,s') = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s']$$

then $$R_s^a = \sum_{s' \in \mathcal{S}} P_{ss'}^{a} \, r(s,a,s')$$

xample : Recycling Robot (Ex. 3.3 - Sutton)

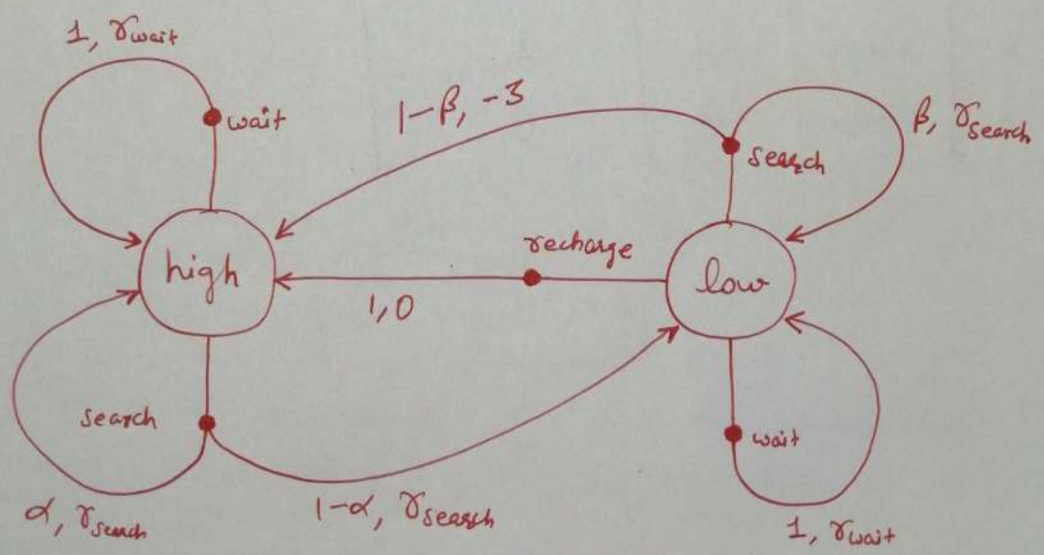
- A robot with the task to collect empty cans
- State is the battery level $\mathcal{S} = \{high, low\}$
- Actions are to either search, wait or head back to recharge

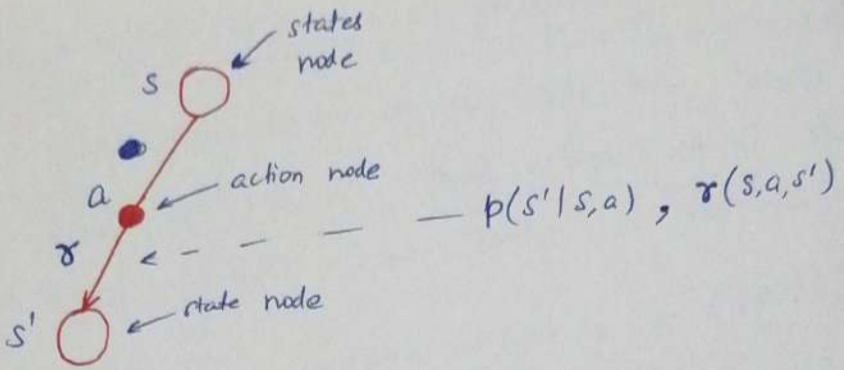
  $\mathcal{A}(high) = \{search, wait\}$

  $\mathcal{A}(low) = \{search, wait, recharge\}$
- when state is high & action is search, robot stays in high with prob. α & goes to low battery with $1-\alpha$
- similarly from low, if search action is taken, then

  $P(low) = \beta$, $P(deplete \rightarrow high) = 1-\beta$
- If battery depletes reward is -3 & robot is recharged to high battery
- otherwise rewards are r_{search} & r_{wait}

Transition graph for this MDP





System dynamics in tabuler format

s	a	s'	$P_{ss'}^{a} = P(s' \mid s,a)$	$r(s,a,s')$
h	search	h	α	r_s
h	search	low	$1-\alpha$	r_s
h	wait	high	1	r_w
h	wait	low	0	—
l	search	high	$1-\beta$	-3
l	search	low	β	r_s
l	wait	high	0	—
l	wait	low	1	r_w
l	recharge	high	1	0
l	recharge	low	0	—

Policies

A Policy π is a distribution over actions given states

$$\boxed{\pi(a|s) = P(A_t = a \mid S_t = s)}$$

A Policy fully defines the behaviour of an agent.
Policies are time independent (stationary).

$$A_t \sim \pi(\cdot|S_t), \ \forall t > 0$$

- Given an MDP $M = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ and policy π

→ the state sequence $S_1, S_2, \ldots$ is a Markov process $\langle \mathcal{S}, \mathcal{P}^\pi \rangle$

⇒ the state & reward seq. $S_1 R_2 S_2 R_3 \ldots$ is an MRP $\langle \mathcal{S}, \mathcal{P}^\pi, \mathcal{R}^\pi, \gamma \rangle$

where

$$\mathcal{P}^\pi_{s,s'} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}^a_{ss'}$$

$$\mathcal{R}^\pi_s = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}^a_s$$

Value functions of an MDP

The state-value funⁿ $V_\pi(s)$ of an MDP is the expected return starting from state s, and then following policy π

$$\boxed{V_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]}$$

The action-value function $q_\pi(s,a)$ is the expected return starting from s, taking action a, and then following policy π

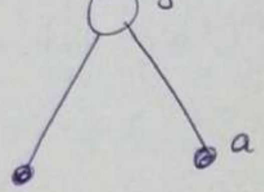
$$\boxed{q_\pi(s,a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]}$$

Bellman expectation equation

$$v_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma\, v_\pi(S_{t+1}) \mid S_t = s\right] \quad \text{--- ①}$$

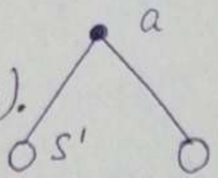
$$\& \quad q_\pi(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma\, q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right] \quad \text{--- ②}$$

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s)\, q_\pi(s,a) \quad \text{--- ③} \quad \Leftarrow$$



backup diagram for v_π

$$\& \quad q_\pi(s,a) = \sum_{s' \in \mathcal{S}} P^a_{ss'} \left(r(s,a,s') + \gamma\, v_\pi(s')\right). \quad \text{--- ④}$$



backup diagram for q_π

from ④ & ③

$$\boxed{v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} P^a_{ss'} \left(r(s,a,s') + \gamma\, v_\pi(s')\right)}$$

or

$$\boxed{v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(R^a_s + \gamma \sum_{s' \in \mathcal{S}} P^a_{ss'}\, v_\pi(s')\right)}$$

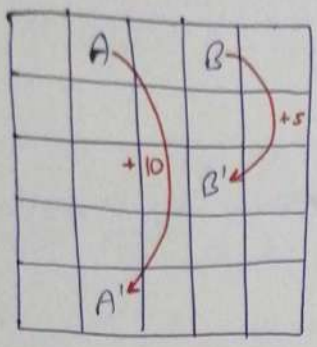
&

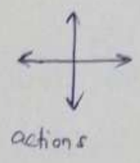
$$\boxed{q_\pi(s,a) = \sum_{s' \in \mathcal{S}} P^a_{ss'} \left(r(s,a,s') + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s')\, q_\pi(s',a')\right)}$$

or

These are called Bellman (expectation) equations for v_π and q_π.

Example : Gridworld (Sutton - Ex. 3.5)





	8.8			
1.5	$\boxed{3.0}$	2.3		
	0.7			

Reward dynamics
all other ~~states~~ actions result in 0 rewards.

state-value fun^n for equiprobable random policy with $\gamma = 0.9$

Exercise 3.14 Lets verify that Bellman eq^n holds for highlighted state.

$$V_\pi(s) = 0.25\left[0.9 \times 1.5 + 0.9 \times 8.8 + 0.9 \times 0.7 + 0.9 \times 2.3\right]$$

$$= 0.25 \times 0.9 \times 13.3$$

$$= 2.9925 \approx 3$$

Excercise 3.15 Lets prove that adding a constant c to all rewards ~~state values~~ doesn't affect their relative values under any policy.

Let original rewards $= r_t$

& original state-value fun^n $= V_\pi(s)$

Then new state-value fun^n of state s

$$\bar{V}_\pi(s) = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \dots \mid S_t = s\right]$$

$$\bar{V}_\pi(s) = \mathbb{E}\left[(r_{t+1} + c) + \gamma(r_{t+2} + c) + \cdots \mid S_t = s\right]$$

$$= \mathbb{E}\left[r_{t+1} + \gamma r_{t+2} + \cdots \mid S_t = s\right]$$

$$+ \mathbb{E}\left[c + \gamma c + \gamma^2 c + \cdots \mid S_t = s\right]$$

$$\boxed{\bar{V}_\pi(s) = V_\pi(s) + \frac{c}{1-\gamma}}$$

$\Rightarrow$ state-values of all states are incremented by constant $v_c = c/(1-\gamma)$.

Optimal Value function

The optimal state-value funⁿ is the maximum value-function over all policies:

$$\boxed{V_*(s) = \max_\pi V_\pi(s)}$$

similarly

$$\boxed{q_*(s,a) = \max_\pi q_\pi(s,a)}$$

- V_* & q_* specify the best possible performance in the MDP, not the best policy.
- An MDP is "solved" when we know the optimal value functions

optimal policy

Define a partial ordering over policies

$\pi \geq \pi'$ if $V_\pi(s) \geq V_{\pi'}(s)$, $\forall s$

Theorem: For any MDP, there exists an optimal policy π^* that is better than or equal to all other policies. i.e.

$$\pi_* \geq \pi \ , \ \forall \pi$$

All optimal policies achieve the same optimal value funn & same optimal action-value funn

$$V_{\pi_*}(s) = V_*(s)$$

$$q_{\pi_*}(s) = q_*(s)$$

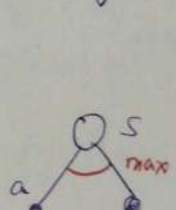
An optimal policy can be found by maximizing over $q_*(s,a)$.

$$\pi_*(a|s) = \begin{cases} 1 , & \text{if } a = \underset{a \in A}{\operatorname{argmax}}\ q_*(s,a) \\ 0 , & o/w \end{cases}$$

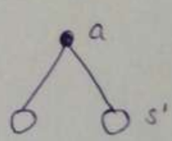
There is always a deterministic optimal policy for any MDP.

Bellman optimality equations

$$V_*(s) = \max_a q_*(s,a)$$



$$q_*(s,a) = \sum_{s' \in S} P_{ss'}^a \left(r(s,a,s') + \gamma V_*(s') \right)$$



Now,

$$\boxed{V_*(s) = \max_a \sum_{s' \in S} P_{ss'}^a \left(r(s,a,s') + \gamma V_*(s') \right)}$$

$$\boxed{V_*(s) = \max_a R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_*(s')}$$

$$\boxed{q_*(s,a) = \sum_{s' \in S} P_{ss'}^a \left(r(s,a,s') + \gamma \max_{a'} q_*(s',a') \right)}$$

$$\boxed{q_*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \max_{a'} q_*(s',a')}$$

- Bellman optimality eqn is non-linear and it doesn't have any closed form solution in general.
- There are iterative methods to solve it
  - Value iteration
  - Policy iteration
  - Q-learning
  - Sarsa

_xample 3.9 (Sutton): Bellman optimality eq^n for the recycling robot

$$V_*(h) = \max \begin{cases} p(h|h,s)\left[r(h,s,h)+\gamma V^*(h)\right] + p(l|h,s) \\ \left[r(h,s,l) + \gamma V^*(l)\right] & \text{(search)} \\ p(h|h,w)\left[r(h,w,h) + \gamma V^*(h)\right] + p(l|h,w) \\ \left[r(h,w,l) + \gamma V^*(l)\right] & \text{(wait)} \end{cases}$$

$$= \max \begin{cases} \alpha\left(r_s + \gamma V^*(s)\right) + (1-\alpha)\left(r_s + \gamma V^*(l)\right) \\ 1\left(r_w + \gamma V^*(h)\right) \end{cases}$$

$$= \max \begin{cases} r_s + \gamma\left(\alpha V^*(s) + (1-\alpha) V^*(l)\right) \\ r_w + \gamma V^*(h) \end{cases}$$

similarly

$$V^*(l) = \max \begin{cases} \beta r_s - 3(1-\beta) + \gamma\left[(1-\beta) V_*(h) + \beta V_*(l)\right] \\ r_w + \gamma V_*(l) \\ \gamma V_*(h) \end{cases}$$

for any choice of r_s, r_w, α, β & γ with $0 \le \alpha, \beta \le 1$, $0 \le \gamma \le 1$, there is exactly one pair of numbers $V_*(h)$ & $V_*(l)$ that simultaneously satisfy these two nonlinear eqns.