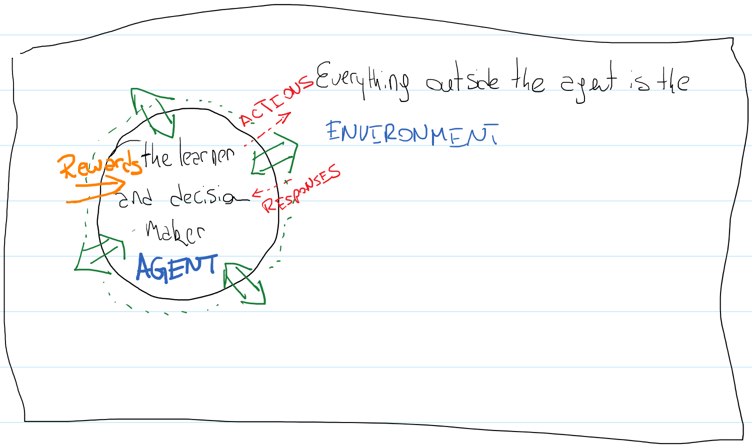
The Agent-Environment Interface

Thursday, September 7, 2017 12:56 AM

The reinforcement learning problem is meant to be a straightforward framing of the problem of learning from interaction to achieve a goal.

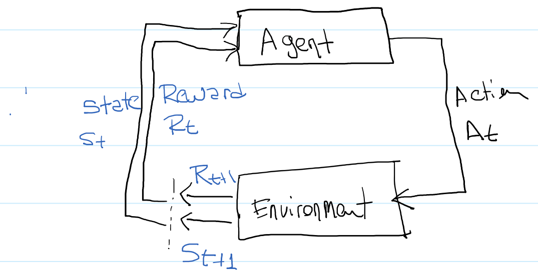


Agent and environment continually interact, the Agent selecting actions and the environment responding with new situations.
The Environment also gives rise to rewards, a special numerical value that the Agent tries to maximize over time

A complete specification of an Environment, including how rewards are determined, defines a Task.

The Agent-Environment Interface (cont.)

Wednesday, September 6, 2017 6:51 PM



t = discrete time steps, $t = 0, 1, 2, 3, \ldots$

S = set of possible states

$A(S_t) \Rightarrow$ is the set of actions available at state S_t.

$R_{t+1} \Rightarrow$ reward received for taking an action

At each time step, the agent implements a mapping from states to probabilities of selecting each possible action. This mapping is called agent's policy and is denoted π_t

$\pi_t(a|s)$: probability that $A_t = a$ if $S_t = s$

Introduction to MDPs

Thursday, September 7, 2017 11:42 AM

Markov decision processes formally describe an environment for Reinforcement Learning

* Where the environment is fully observable
  - → i.e. the current state completely characterises the process
* Almost all RL problems can be formalized as MDPs. e.g.
  - → Partially observable problems can be converted to MDPs.
  - → Bandits are MDPs with one state

Markov Property

Thursday, September 7, 2017 11:50 AM

"The future is independent of the past given the present"

Definition

State S_t is Markov if and only if:

$$\mathbb{P}[S_{t+1} | S_t] = \mathbb{P}[S_{t+1} | S_1, S_2, S_3, \dots, S_t]$$

* The state captures all relevant information from the history.
* Once the state is known, the history may be thrown away.
  → State is a sufficient statistic of the future.

State Transition Matrix

Thursday, September 7, 2017 12:13 PM

For a Markov state s and succesor state s', the state transition probability is defined by:

$$P_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s]$$ ∴ Read transition from s to s'

State transition matrix P determines the transition probabilities from all states s to all sucess states s',

$$P = \begin{array}{cc} & \text{to} \\ \text{from} & \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & & \\ P_{11} & & P_{nn} \end{bmatrix} \end{array}$$ ∴ Where each row of the matrix sums 1.

Markov Process

Thursday, September 7, 2017 12:26 PM

A Markov process is a memoryless random process, i.e. sequence of random states $S_1, S_2, S_3, \dots$ with Markov property.

Definition

A Markov process (or Markov Chain) is a tuple (S, P) → state space

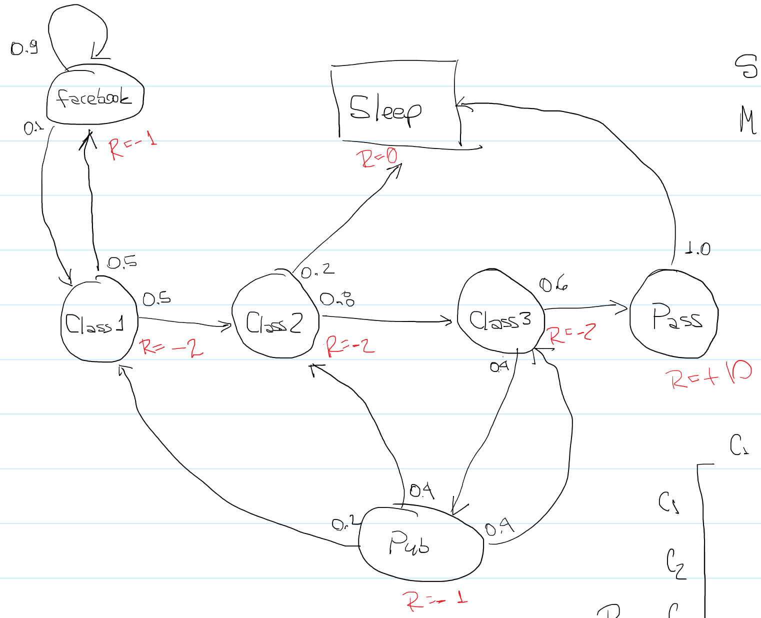
→ S is a (finite) set of states

→ $\underline{P}$ is a state transition probability matrix

$$P_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s]$$

Student Markov Chain

Thursday, September 7, 2017   12:32 PM



Sample episodes for Student Markov starting from $S_1 = C_1$

$S_1, S_2, S_3, \dots S_T$

* C1 C2 C3 PASS SLEEP
* C1 FB FB C1 C2 SLEEP

$$
P = \begin{array}{c|cccccc}
 & C_1 & C_2 & C_3 & \cdots & FB & Sleep \\
\hline
C_1 & & 0.5 & & & 0.5 & \\
C_2 & & & 0.8 & & & 0.2 \\
C_3 & & & & & & \\
\vdots & & & & & & \\
FB & & & & & & \\
Sleep & & & & & &
\end{array}
$$

Markov Reward Process

Thursday, September 7, 2017 12:59 PM

A Markov reward process is a Markov chain with values:

Definition

A Markov Reward Process is a tuple $\langle S, P, R, \gamma \rangle$ → Gamma Discount factor.

→ S is a finite set of states

→ P is the state transition probability matrix

→ $P_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s]$

→ R is a reward function, $R_s = \mathbb{E}[R_{t+1} \mid S_t = s]$ → immediate Reward → How much reward do I get from that moment

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

Return

Thursday, September 7, 2017 1:12 PM

The definition of return is the total discounted reward from time-step t.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma R_{t+3} = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

G_t = Return

∴ The discount $\gamma \in [0,1]$ is the present value of future rewards.

→ the value of receiving reward R after $k+1$ time-steps is $\gamma^k R$

→ This values immediate reward above delayed reward.

- γ close to φ leads to "myopic" evaluation
- γ close to 1 leads to "far-sighted" evaluations

Why Discount?

Thursday, September 7, 2017 1:22 PM

Most Markov reward and decision processes are discounted. Why?

- There is more uncertainty about the future
- Accounts for inperfect models $\Longleftrightarrow$ How much you trust your model?
- Mathematically convinient to discount rewards.
- Avoids infinite return in cyclic Markov processes.
- If the reward is financial, inmediate rewards may earn more interest than delayed rewards.
- Animal or human behaviour shows preference for inmediate rewards.
- It is sometimes possible to use undiscounted Markov reward processes (i.e. $\gamma=1$) e.g. if all ~~Processes~~ sequences terminate.

Bellman Equation For MRPs

Thursday, September 7, 2017 3:42 PM

The value function can be decomposed into two parts:

* immediate reward
* discounted value of successor state $\gamma v(S_{t+1})$

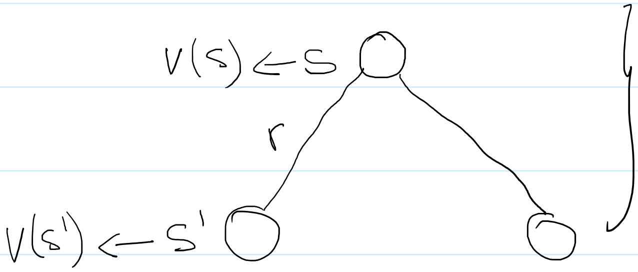
$$
\begin{aligned}
v(s) &= \mathbb{E}[G_t \mid S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s]
\end{aligned}
$$

immediate reward + discounted Reward of the next State

Bellman Equation for MRPs (2)

Thursday, September 7, 2017 3:53 PM

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



Backup diagrams

$$V(s) = R_s + \gamma \sum_{s' \in \mathcal{S}} P_{ss'} V(s')$$

Bellman Equation in Matrix Form

Thursday, September 7, 2017 4:11 PM

The Bellman Equation can be expressed concisely using matrices,

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

where v is a column vector with one entry per state:

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} R_1 \\ \vdots \\ R_n \end{bmatrix} + \gamma \begin{bmatrix} P_{11} & \dots & P_{1n} \\ \vdots & & \\ P_{1n} & \dots & P_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \\ v(n) \end{bmatrix}$$

Markov Decision Process

Thursday, September 7, 2017 4:27 PM

A Markov Decision Process (MDP) is a Markov reward process with decisions. It is an Environment in which all states are Markov.

Definition:

A Markov decision process is a tuple $\langle S, A, P, R, \gamma \rangle$

* S is a finite set of states
* A is a finite set of Actions
* P is a state transition probability Matrix

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

* R is the reward function
* γ is the discount factor

44:05

Policies (1)

Thursday, September 7, 2017 8:49 PM

Definition

A policy π is a distribution over actions given states,

$$\pi(a|s) = \mathbb{P}[A_t = a \mid S_t = s]$$

* A policy fully defines the behaviour of an agent.
* MDP policies fully depend on the current state (Not the history)
  → Policies are stationary (time-independent) $A_t \sim \pi(\cdot|T)$, $\forall t > 0$

Value Function

Thursday, September 7, 2017 9:04 PM

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

$$V_\pi(s) = \mathbb{E}_\pi\left[G_T \mid S_T = s\right]$$