A Quick Review on RL and MDP

LINKÖPINGS UNIVERSITET

Farnaz Adib Yaghmaie

Linkoping University, *Sweden*
farnaz.adib.yaghmaie@liu.se

April 6, 2021

Machine Learning

- Supervised Learning
- Unsupervised Learning
- **Reinforcement Learning**

Finding suitable actions to take in a given situation in order to maximize a reward [1]***.***

[1] Richard S Sutton & Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

How RL is different from other branches of ML?

- No supervisor; only a reward
- The action will effect subsequent data
- Dynamic data *vs.* Static data

An RL framework

- Reward
- Environment
- Agent

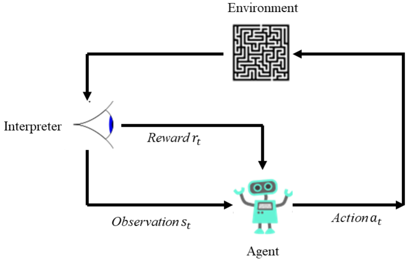
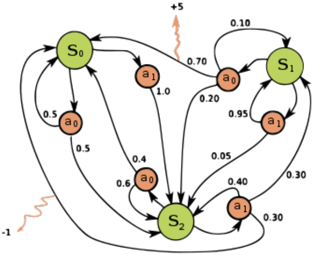


Photo Credit: @ https://en.wikipedia.org/wiki/Reinforcement_learning

A Markov Decision Process (MDP) is a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$

- $\mathcal{S}$: The set of states.
- $\mathcal{A}$: The set of actions.
- $\mathcal{P}$: The set of transition probability.
- $\mathcal{R}$: The set of immediate rewards associated with the state-action pairs.
- $0 \leq \gamma \leq 1$: Discount factor.



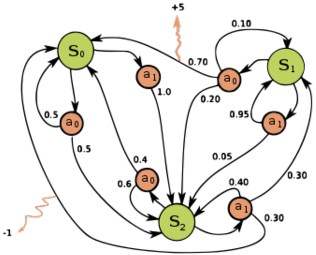
Modified version of @ https://en.wikipedia.org/wiki/Markov_decision_process

States: Describe internal status of MDP

Actions: Possible choices to make in each state of MDP

Transitions probability: $\mathcal{P}$ is the set of transition probability with n_a matrices each of dimension $n_s \times n_s$ where s, s' entry reads

$$[\mathcal{P}^a]_{ss'} = p[s_{t+1} = s' | s_t = s, \; a_t = a] \quad (1)$$



Reward:

$$r_t = r(s,\ a) \tag{2}$$

Total reward:

$$R(T) = \sum_{t=1}^{T} \gamma^t r_t \tag{3}$$

Average reward:

$$R(T) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r_t \tag{4}$$

Do you care about future as much as now (and past)?

$\gamma \in [0, 1]$:

- $\gamma \to 0$: We only care about the current reward not what we'll receive in future
- $\gamma \to 1$: We care about all rewards equally

RL goal

Generate actions to maximize the future rewards

- Policy: The agent's decision
- Value function: how good the agent does in a state

$$V(s) = \mathbf{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots | s_t = s\right]$$

- Model: The agent's interpretation of the environment

Not all components are necessary!

Policy Gradient	*Learning policy*
Dynamic Programming based	*Learning value function*
Model building	*Learning the model of environment*

Email your questions to

farnaz.adib.yaghmaie@liu.se