Basics of Reinforcement Learning & Markov Decision Process

Lecture 1.2

Alina Vereshchaka

University at Buffalo

avereshc@buffalo.edu

May 28, 2019

*Slides has been modified from David Silver’s RL course
Overview

1. Learning

2. Definition

3. Markov Decision Processes (MDP)
Why do we need to learn?

There are (at least) two distinct reasons to learn:

1. Find previously unknown solutions. E.g., a program that can play Go better than any human, ever

2. Find solutions online, for unforeseen circumstances. E.g., a robot that can navigate terrains that differ greatly from any expected terrain

Reinforcement learning seeks to provide algorithms for both cases. Note that the second point is not (just) about generalization — it is about learning efficiently online, during operation.
Why do we need to learn?

Science of learning to make decisions from interaction. This requires us to think about:

- ...time
- ...(long-term) consequences of actions
- ...actively gathering experience
- ...predicting the future
- ...dealing with uncertainty
Examples of decision problems

Examples:

- Fly a helicopter
- Manage an investment portfolio
- Control a power station
- Make a robot walk
- Play video or board games

These are all reinforcement learning problems (no matter which solution method you use)
Learning

Definition

Markov Decision Processes (MDP)
Core concepts of a reinforcement learning system are:

- Environment
- Reward signal
- Agent, containing:
  - Agent state
  - Policy
  - Value function (probably)
  - Model (optionally)
The **agent** is acting in an **environment**. How the environment reacts to certain actions is defined by a **model** which we may or may not know. The agent can stay in one of many **states** \((s \in S)\) of the environment, and choose to take one of many **actions** \((a \in A)\) to switch from one state to another. Which state the agent will arrive in is decided by the **transition probabilities** between states \(P(s'|s, a)\). Once an action is taken, the environment delivers a **reward** \((r \in R)\) as a feedback.
Table of Contents

1 Learning
2 Definition
3 Markov Decision Processes (MDP)
Markov Decision Processes (MDP)
Finite Markov Decision Processes (MDP)

At each step $t$ the agent:

- Receives state / observation $O_t$ and reward $R_t$
- Executes action $A_t$

The environment:

- Receives action $A_t$
- Emits state / observation $O_{t+1}$ and reward $R_{t+1}$
Finite Markov Decision Processes (MDP)

Markov property:

$$\mathbb{P}[S_{t+1}|S_t] = \mathbb{P}[S_{t+1}|S_1, S_2, \ldots, S_t]$$
Markov property:

\[ P[S_{t+1}|S_t] = P[S_{t+1}|S_1, S_2, \ldots, S_t] \]

“The future is independent of the past given the present”

Daily life trajectory:

\[ S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots, S_T \]
A Markov Chain is a tuple $\langle S, P \rangle$

- $S$ is a set of states
- $P$ is a state transition probability matrix

\[ P_{ss'} = \mathbb{P}[S_{t+1} = s'|S_t = s] \] (1)
Markov Chain - Student Example

- States: Class 1, Class 2, Class 3, Pub, Sleep, Facebook, Pass
- Transition Probabilities:
  - Class 1 to Class 2: 0.5
  - Class 1 to Pub: 0.4
  - Class 2 to Class 3: 0.8
  - Class 2 to Pub: 0.4
  - Class 3 to Pass: 0.6
  - Pub to Class 1: 0.2
  - Pub to Class 2: 0.4
  - Sleep to Class 1: 0.1
  - Sleep to Class 2: 0.9
  - Facebook to Class 1: 0.5
  - Facebook to Pub: 0.1
  - Pass to Sleep: 1.0
Sample episodes for Student Markov Chain starting from $S_1 = C1$.

Episodes:
- C1 C2 C3 Pass Sleep
- C1 FB FB C1 C2 Sleep
- C1 C2 C3 Pub C2 C3 Pass Sleep
- C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 C3 Pub C2 Sleep
Markov Chain - Student Example
Markov reward process is a Markov chain with values.

**Definition**

A *Markov Reward Process* is a tuple \( \langle S, P, R, \gamma \rangle \)

- \( S \) is a set of states
- \( P \) is a state transition probability matrix

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

- \( R \) is a reward function, \( R_s = \mathbb{E}[R_{t+1} | S_t = s] \)
- \( \gamma \) is a discount factor, \( \gamma \in [0, 1) \)
Markov Reward Process - Student Example
The discounting factor $\gamma \in [0, 1)$ penalize the rewards in the future.
Reward at time $k$ worth only $\gamma^{k-1}$

**Motivation:**

- The future rewards may have higher uncertainty (stock market)
- The future rewards do not provide immediate benefits (As human beings, we might prefer to have fun today rather than 5 years later ;)
- Discounting provides mathematical convenience (we don’t need to track future steps infinitely to compute return)
- It is sometimes possible to use undiscounted Markov reward processes (i.e. $\gamma = 1$) e.g. if all sequences terminate.
Definition

The *return* $G_t$ is the total discounted reward from time-step $t$.

\[
G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}
\]  

- $\gamma$ is a discount factor ($\gamma \in [0, 1)$)
- $R$ is the reward
- The value of receiving reward $R$ after $k + 1$ time-steps is $\gamma^k R$
Sample returns for Student MRP:
Starting from $S_1 = C1$ with $\gamma = 0.5$

$$G_1 = R_2 + \gamma R_3 + \cdots + \gamma^{T-2} R_T$$

<table>
<thead>
<tr>
<th>C1 C2 C3 Pass Sleep</th>
<th>$v_1 = -2 - 2 \times \frac{1}{2} - 2 \times \frac{1}{4} + 10 \times \frac{1}{8}$</th>
<th>$= -2.25$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1 FB FB C1 C2 Sleep</td>
<td>$v_1 = -2 - 1 \times \frac{1}{2} - 1 \times \frac{1}{4} - 2 \times \frac{1}{8} - 2 \times \frac{1}{16}$</td>
<td>$= -3.125$</td>
</tr>
<tr>
<td>C1 C2 C3 Pub C2 C3 Pass Sleep</td>
<td>$v_1 = -2 - 2 \times \frac{1}{2} - 2 \times \frac{1}{4} + 1 \times \frac{1}{8} - 2 \times \frac{1}{16} \cdots$</td>
<td>$= -3.41$</td>
</tr>
<tr>
<td>C1 FB FB C1 C2 C3 Pub C1 …</td>
<td>$v_1 = -2 - 1 \times \frac{1}{2} - 1 \times \frac{1}{4} - 2 \times \frac{1}{8} - 2 \times \frac{1}{16} \cdots$</td>
<td>$= -3.20$</td>
</tr>
<tr>
<td>FB FB FB C1 C2 C3 Pub C2 Sleep</td>
<td>$v_1 = -2 - 1 \times \frac{1}{2} - 1 \times \frac{1}{4} - 2 \times \frac{1}{8} - 2 \times \frac{1}{16} \cdots$</td>
<td>$= -3.20$</td>
</tr>
</tbody>
</table>
Markov Decision Process (MDP)

Definition

A Markov Decision Process (MDP) is a tuple \( \langle S, A, P, R, \gamma \rangle \)

- \( S \) is a set of states
- \( A \) is a set of actions
- \( P \) is a state transition probability matrix
  \[
  P_{ss'} = \mathbb{P}[S_{t+1} = s'|S_t = s, A_t = a]
  \]  
  \( (4) \)
- \( R \) is a reward function, \( R_s = \mathbb{E}[R_{t+1}|S_t = s, A_t = a] \)
- \( \gamma \) is a discount factor, \( \gamma \in [0, 1) \)
Markov Decision Process - Student Example

![Diagram of Markov Decision Process]

- **Facebook**: $R = -1$
- **Quit**: $R = 0$
- **Sleep**: $R = 0$
- **Study**: $R = -2$
- **Pub**: $R = +1$
- **Study**: $R = +10$
Summary So Far

- Mains reasons to learn (not just for agents) are to find previously unknown solutions and to find solutions in unforeseen circumstances
- Core parts of a reinforcement learning are: Environment, Reward, Agent
- Markov property: The future is independent of the past given the present
- The discounting factor $\gamma \in [0, 1)$ penalize the rewards in the future
- Markov Decision Process (MDP) defined as a tuple $\langle S, A, P, R, \gamma \rangle$