Markov Decision Process

Slides from

1. 이웅원 외, 파이썬과 케라스로 배우는 강화학습, 주교재
2. 이웅원, 가깝고도 먼 DeepRL, PPT
3. David Silver, Reinforcement Learning, PPT

References

1. Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction, MIT Press
2. 유튜브, 전민영, 노승은, 강화학습의 기초 이론, 팡요랩

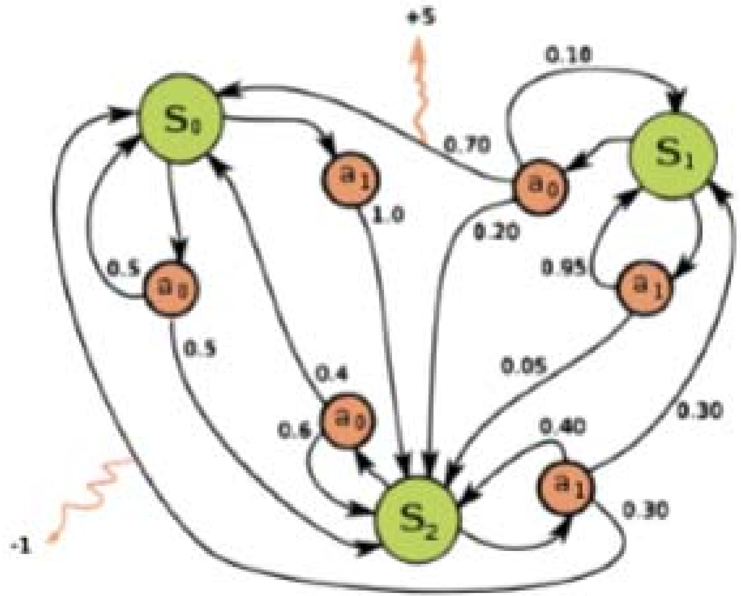
Contents

1. 강화학습이 풀고자 하는 문제 : Sequential Decision Problem
2. 문제에 대한 수학적 정의 : MDP & Bellman Equation
3. MDP를 계산으로 푸는 방법 : Dynamic Programming
4. MDP를 학습으로 푸는 방법 : Reinforcement Learning
5. 상태공간이 크고 차원이 높을 때 쓰는 방법 : Function Approximation
6. 바둑과 같은 복잡하고 어려운 문제를 푸는 방법 : Deep Reinforcement Learning

Markov Decision Process

- 환경을 컴퓨터가 이해할 수 있도록 재정의
- 시간에 따라 변하는 "상태"가 있으며 상태 공간 안에서 움직이는 "에이전트"가 있다
  - ➢ 에이전트는 행동을 선택할 수 있다 -> 확률적
  - ➢ 에이전트의 행동에 따라 다음 상태와 보상이 결정된다 -> 확률적

=> 확률적 모델링 : Markov Decision Process



https://en.wikipedia.org/wiki/Markov_decision_process

Introduction to MDPs

- *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 formalised as MDPs, e.g.
    - Optimal control primarily deals with continuous MDPs
    - Partially observable problems can be converted into MDPs
    - Bandits are MDPs with one state

David Silver, Reinforcement Learning, PPT

Markov Decision Process

1. Sequential Decision Problem을 수학적으로 정의
2. MDP(Markov Decision Process)의 목표는 reward를 최대화
3. Markov Process -> MRP(Markov Reward Process) -> MDP(Markov Decision Process)

Markov?

- 1800년대의 러시아 수학자

- $\boldsymbol{P}[S_{t+1}|S_t] = \boldsymbol{P}[S_{t+1}|S_1,S_2...,S_t]$
  -> 미래는 현재로부터 정해지며 과거는 영향을 주지 못한다.

Markov Property

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

Definition

A state S_t is *Markov* if and only if

$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, ..., S_t]$$

- The state captures all relevant information from the history
- Once the state is known, the history may be thrown away
- i.e. The state is a sufficient statistic of the future

David Silver, Reinforcement Learning, PPT

State Transition Matrix

For a Markov state s and successor state s', the *state transition probability* is defined by

$$\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$$

State transition matrix $\mathcal{P}$ defines transition probabilities from all states s to all successor states s',

$$\mathcal{P} \quad = \textit{from} \quad \begin{array}{c} \textit{to} \\ \begin{bmatrix} \mathcal{P}_{11} & \ldots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \ldots & \mathcal{P}_{nn} \end{bmatrix} \end{array}$$

where each row of the matrix sums to 1.

David Silver, Reinforcement Learning, PPT

Markov Process

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

Definition

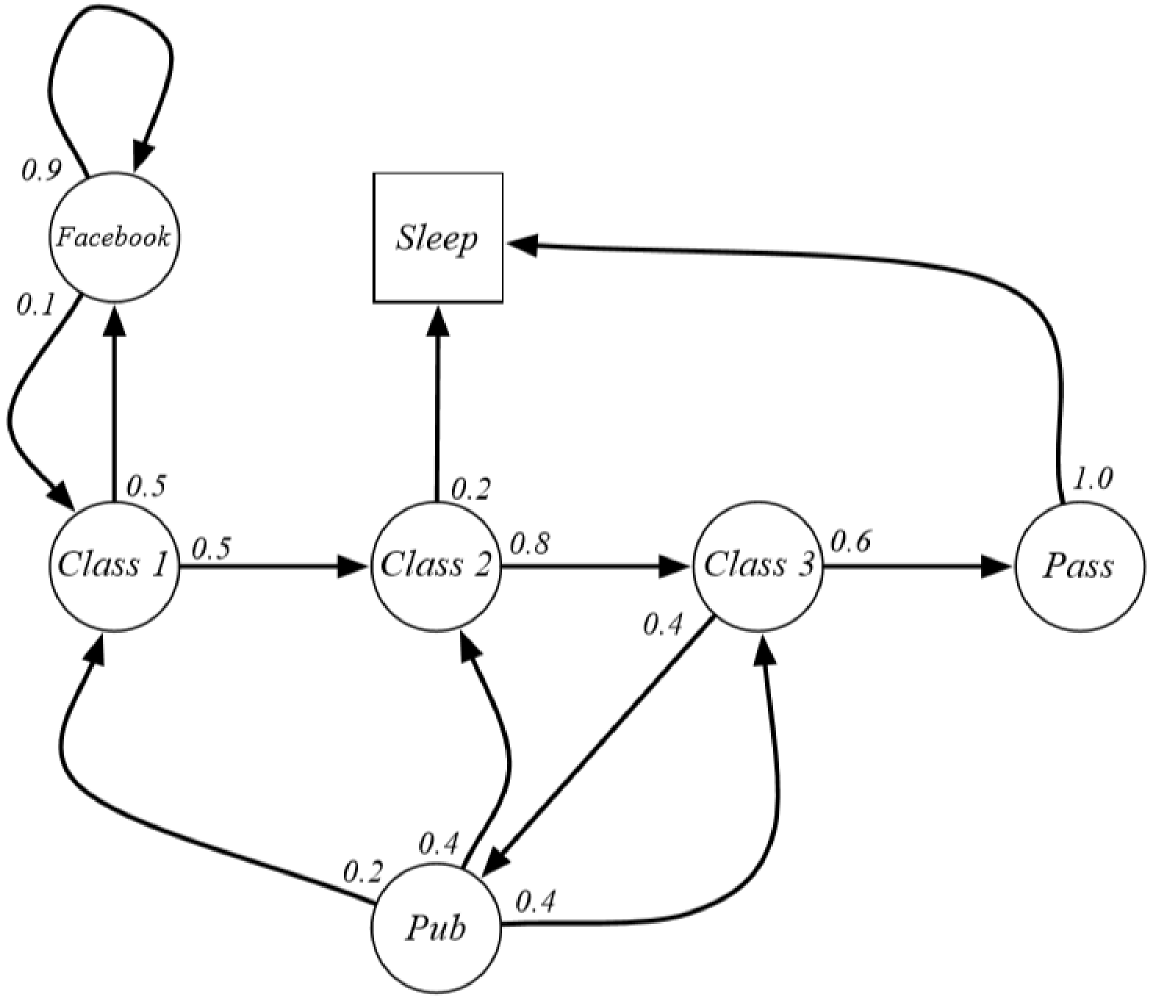
A *Markov Process* (or *Markov Chain*) is a tuple $\langle \mathcal{S}, \mathcal{P} \rangle$

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$

Markov Process

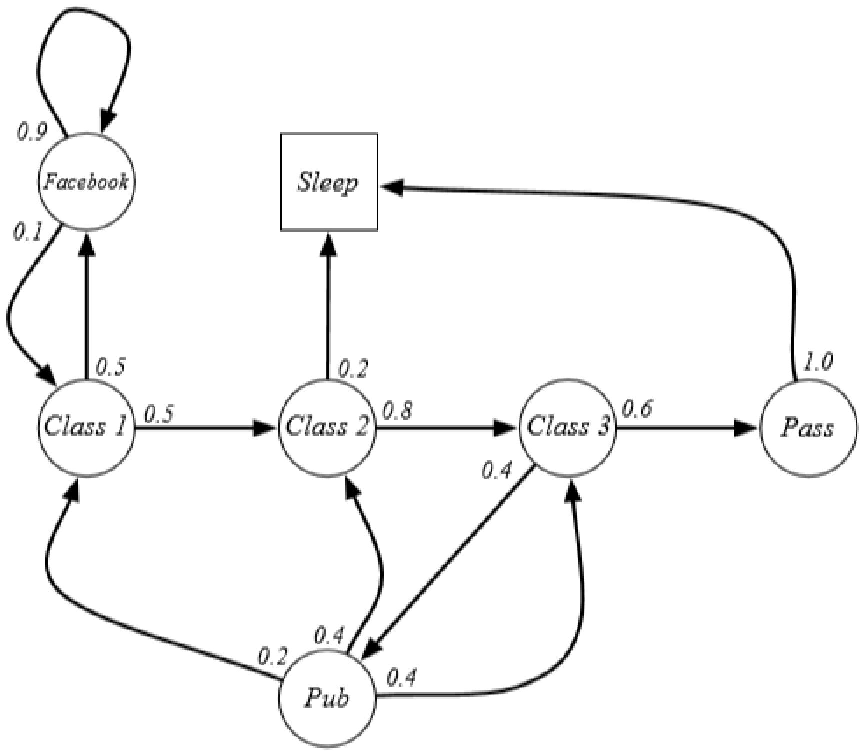
- MP = $\{S,\ P_{ss'}\}$ 로 정의
- MP의 구성요소
  - ➢S : 상태(state) 집합
  - ➢$P_{ss'} = \boldsymbol{P}[S_{t+1} = s' | S_t = s]$
    : 상태변환확률
    (state transition probability)

- Ex) Student Markov Chain



David Silver, Reinforcement Learning, PPT

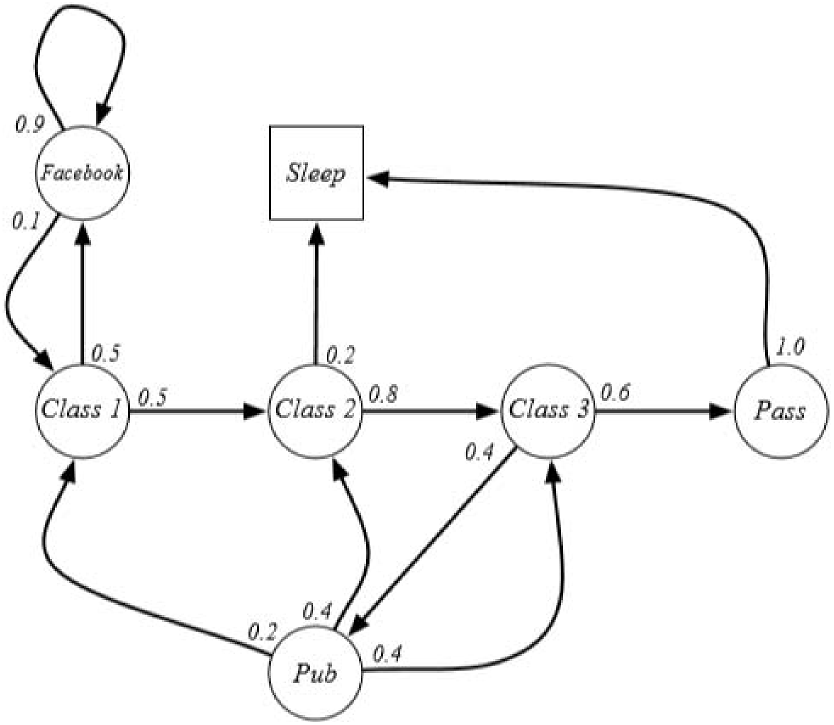
Example: Student Markov Chain Transition Matrix



$$
\mathcal{P} = \begin{array}{c} \\ C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep \end{array}
\begin{array}{c}
\begin{array}{ccccccc} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{array} \\
\left[\begin{array}{ccccccc}
 & 0.5 & & & & 0.5 & \\
 & & 0.8 & & & & 0.2 \\
 & & & 0.6 & 0.4 & & \\
 & & & & & & 1.0 \\
0.2 & 0.4 & 0.4 & & & & \\
0.1 & & & & & 0.9 & \\
 & & & & & & 1
\end{array}\right]
\end{array}
$$

David Silver, Reinforcement Learning, PPT

Example: Student Markov Chain Episodes



Sample episodes for Student Markov Chain starting from $S_1 = C1$

$$S_1, S_2, ..., S_T$$

- 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

David Silver, Reinforcement Learning, PPT

Markov Reward Process

A Markov reward process is a Markov chain with values.

Definition

A *Markov Reward Process* is a tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

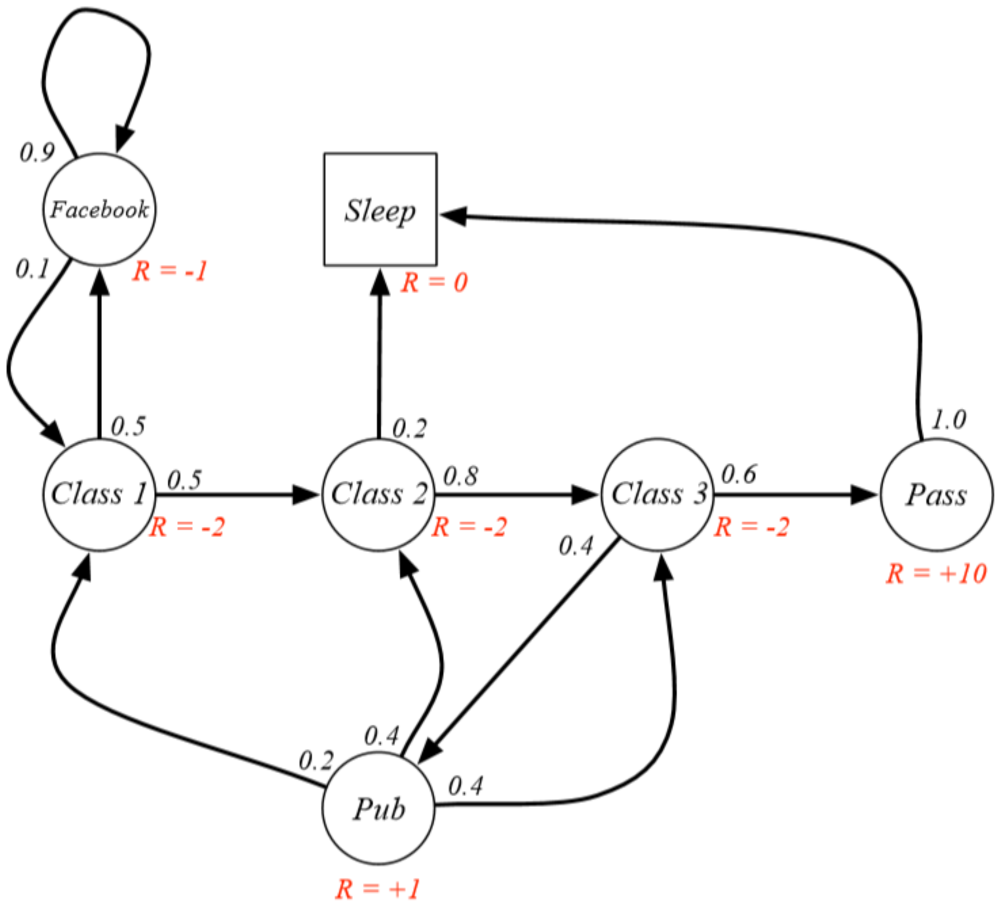
- $\mathcal{S}$ is a finite set of states
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}_s = \mathbb{E}\left[R_{t+1} \mid S_t = s\right]$
- γ is a discount factor, $\gamma \in [0, 1]$

David Silver, Reinforcement Learning, PPT

Markov Reward Process

- MRP = $\{S, P_{ss'}, R_{s}, \gamma\}$ 로 정의되는 tuple
- MRP의 구성요소
  - S : 상태(state) 집합
  - $P_{ss'}$: 상태변환확률(state transition probability)

    $$P_{ss'} = \boldsymbol{P}[S_{t+1} = s' | S_t = s]$$
  - $R_s = \boldsymbol{E}[R_{t+1} | S_t = s]$
    : 보상(reward)
  - γ : 할인율(discount factor) ,
    $\gamma \in [0,1]$



David Silver, Reinforcement Learning, PPT

Markov Decision Process

A Markov decision process (MDP) is a Markov reward process with decisions. It is an *environment* in which all states are Markov.

Definition

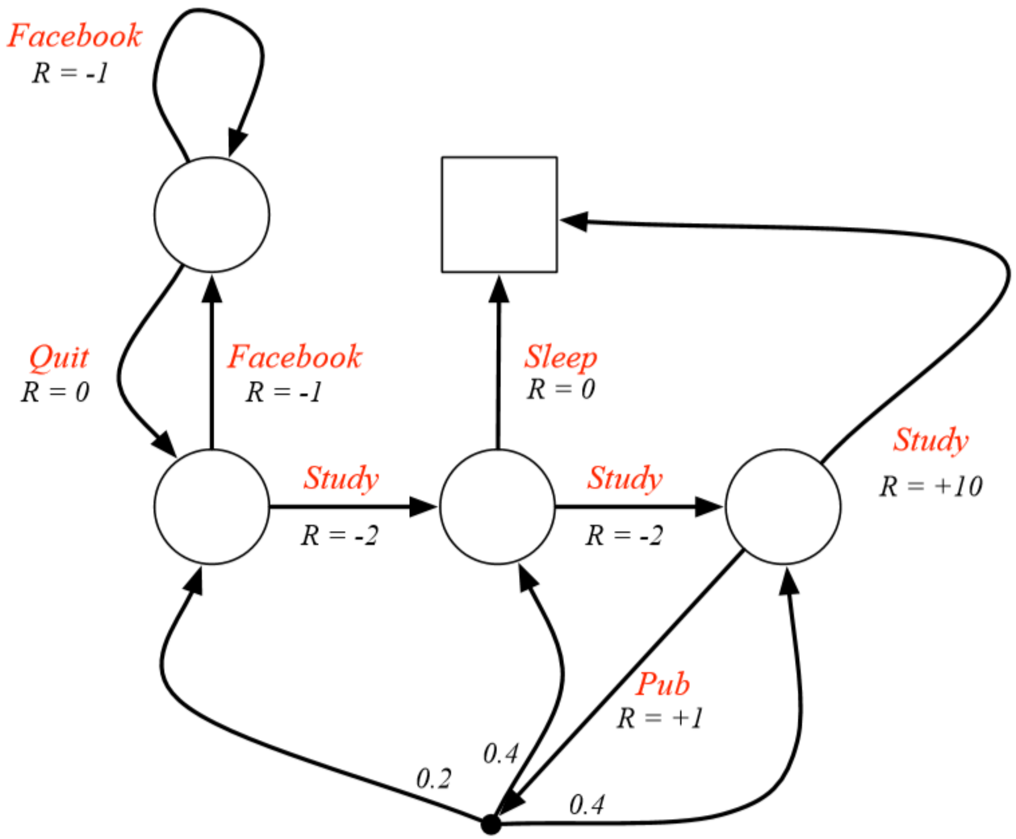
A *Markov Decision Process* is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}_{ss'}^{a} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s, A_t = a\right]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right]$
- γ is a discount factor $\gamma \in [0, 1]$.

David Silver, Reinforcement Learning, PPT

Markov Decision Process

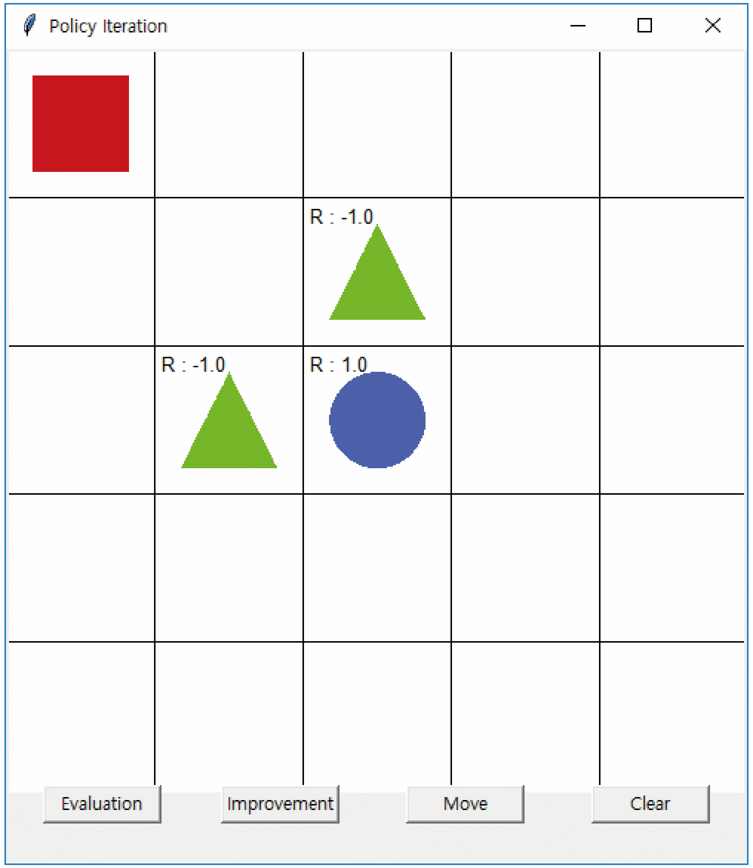
- MDP = $\{S, A, P_{ss'}^{a}, R_{s}^{a}, \gamma\}$ 로 정의되는 tuple
- MDP의 구성요소
  - ➢ S : 상태(state) 집합
  - ➢ A : 행동(action) 집합
  - ➢ $P_{ss'}^{a} = \boldsymbol{P}[S_{t+1} = s' | S_t = s, A_t = a]$: 상태변환확률(state transition probability)
  - ➢ $R_{s}^{a} = \boldsymbol{E}[R_{t+1} | S_t = s, A_t = a]$: 보상(reward)
  - ➢ γ : 할인율(discount factor), $\gamma \in [0,1]$



David Silver, Reinforcement Learning, PPT

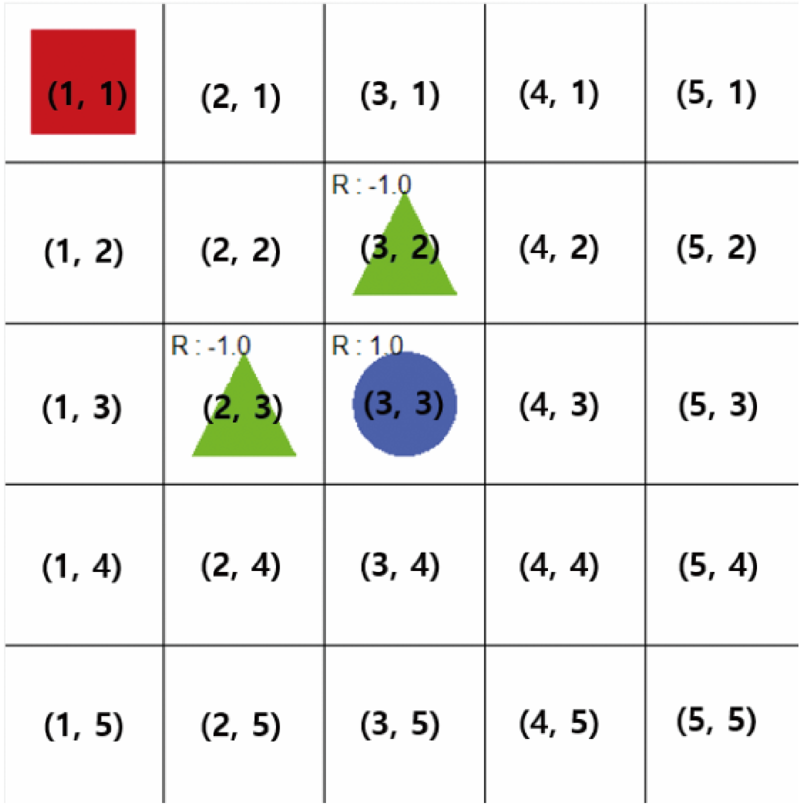
Grid World 예제

- 격자를 기반으로 한 예제 : 5 X 5 = 25개의 격자를 가짐
- 고전 강화학습의 가장 기본적인 예제 -> 에이전트가 학습하는 과정을 눈으로 보기 쉬움
- 목표 : 세모를 피해서 파란색 동그라미로 가기



MDP 1: 상태 (State)

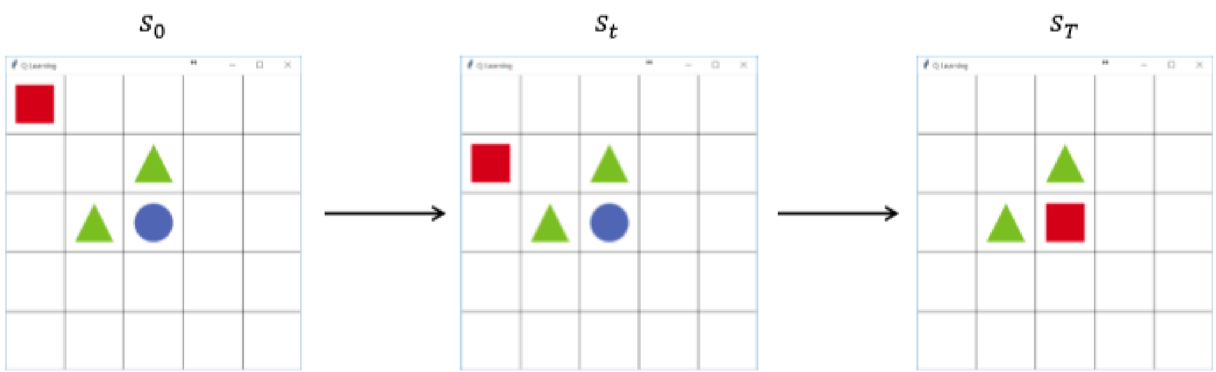
- 상태 : 자신의 상황에 대한 관찰
- 상태 집합 : 에이전트가 관찰 가능한 상태의 집합
- 그리드월드의 상태 집합 : $\boldsymbol{S}$ = { (1,1) , (2,1) , (1,2) , ⋯ , (5,5) }



MDP 1: 상태

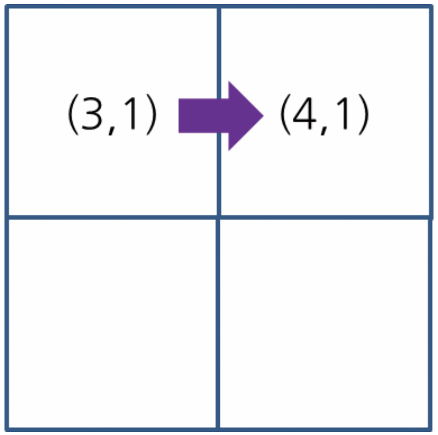
- 에이전트는 시간에 따라 환경을 탐험 -> 상태도 시간에 따라 변한다 -> 확률변수(random variable)
- 예) 시간 t일 때 상태 : $S_t = s$ or $S_t = (1,3)$
  - 확률변수(random variable)은 대문자, 특정 상태는 소문자
- Episode : 처음 상태부터 마지막 상태까지의 sequence

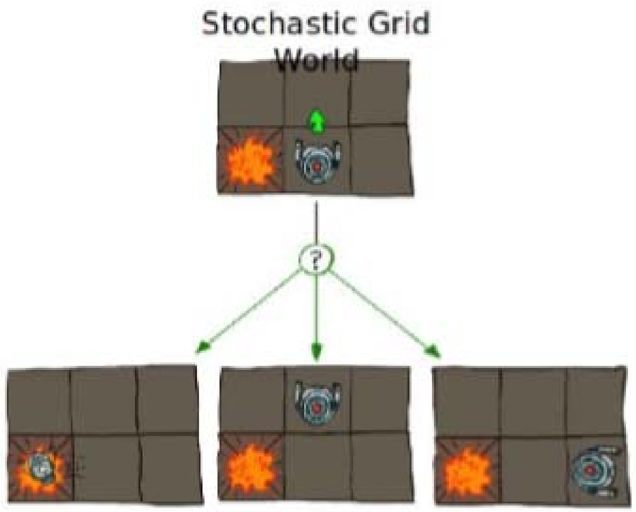
$$\tau = s_0, s_1, s_2, \cdots, s_{T-1}, s_T$$



MDP 2: 행동

- 에이전트가 할 수 있는 행동의 집합 : $\boldsymbol{A}$ = {위,아래,좌,우}
- 시간 t에 취한 행동 $A_t = a$ -> 확률변수
- 만약 A_t = 우 라면 항상 (3, 1)에서 (4, 1)로 갈까?
  - 상태 변환 확률에 따라 다르다





MDP 3: 상태변환확률

- 상태변환확률 : 상태 s에서 행동 a를 했을 때 상태 s'으로 갈 확률

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

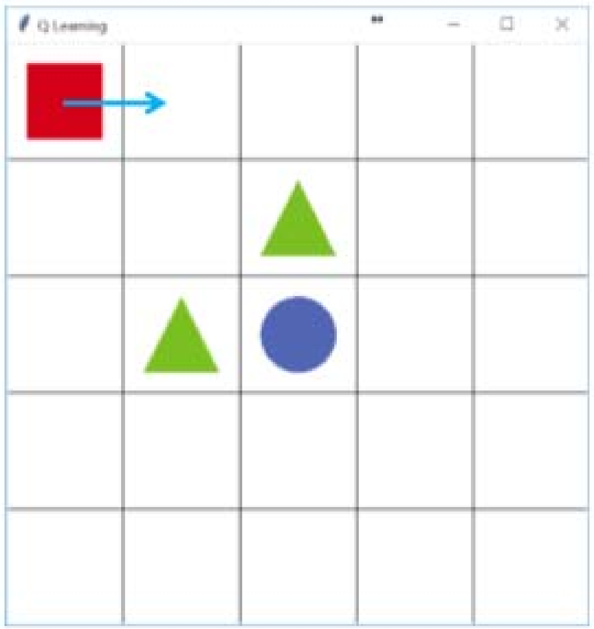
  -> model of environment
- 상태변환확률을 안다면 : model-based

  -> Dynamic Programming
- 상태변환확률을 모른다면 : model-free

  -> Reinforcement Learning

MDP 3: 상태변환확률

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



상태 (1, 1)에서 행동 “우”를 했을 경우

1. 상태 (2, 1)에 갈 확률은 0.8
2. 상태 (1, 2)에 갈 확률은 0.2

MDP 4: 보상

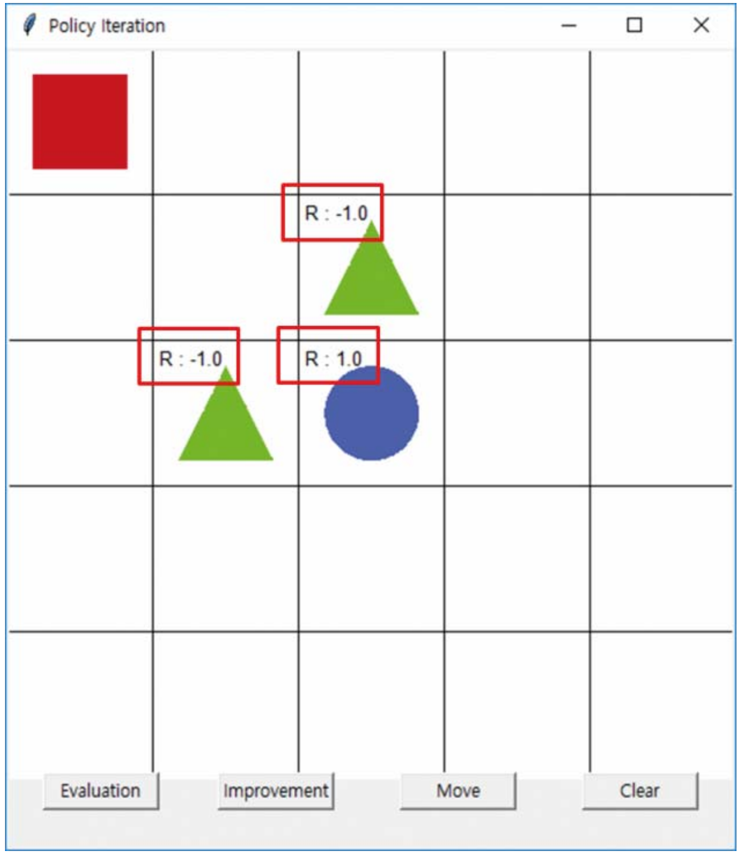
- 에이전트가 한 행동에 대한 환경의 피드백 : 보상(+ or -)
  -> 에이전트가 학습할 수 있는 유일한 정보
- 시간이 t이고 상태 $S_t = s$ 에서 $A_t = a$를 선택했을 때 받는 보상
  -> $R_{t+1} \mid S_t = s, A_t = a$
- 보상은 현재 시간 t가 아닌 $t + 1$에 환경으로부터 받는다
- 같은 상태 s에서 같은 행동 a를 했더라도 그때 그때마다 보상이 다를 수 있음 -> 확률변수
  -> 기댓값(expectation)으로 표현 -> 스칼라 값

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

- Model of environment = $\{P_{ss'}^a, R_s^a\}$

MDP 4: 보상

- 보상은 에이전트의 목표에 대한 정보를 담고 있어야 함
- 그리드월드의 보상 : 초록색 세모 (-1), 파란색 동그라미 (+1)
  - 초록색 세모를 피해 파란색 동그라미로 가라!



MDP 5: 할인율(Discount Factor)

- 할인율 : 미래에 받은 보상을 현재의 시점에서 고려할 때 할인하는 비율
- 만약 복권에 당첨되었다면 당첨금 1억원을 당장 받을지 10년 뒤에 받을지?
  - 가까운 보상이 미래의 보상보다 더 가치가 있다 -> 할인
- 보상에서 시간의 개념을 포함하는 방법
- aka 감가율, 감쇠율, 감쇄인자

MDP 5: 할인율

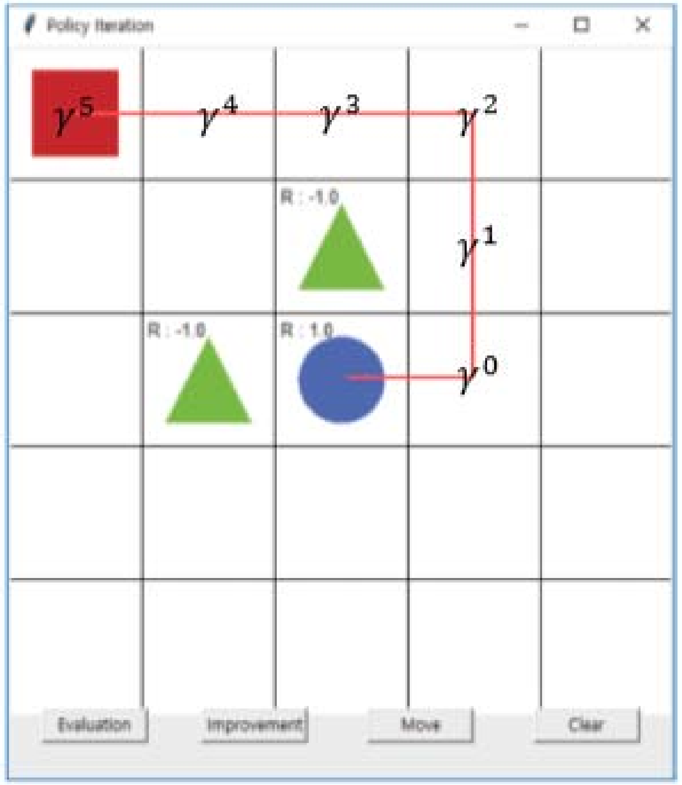
- 할인율은 0에서 1 사이의 값

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

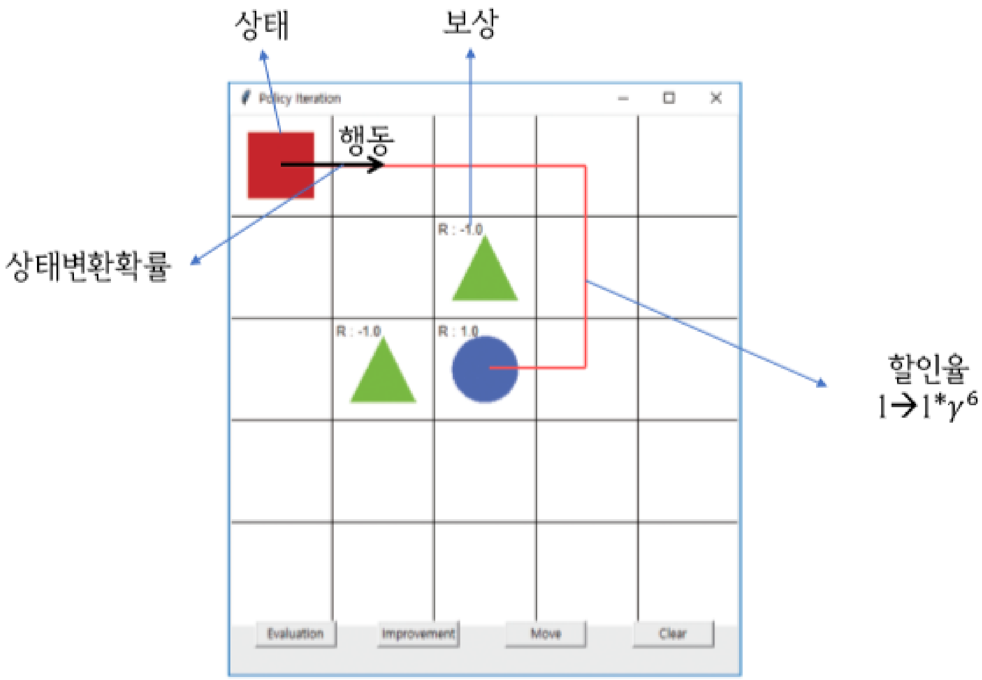
- 현재의 시간 t로부터 k만큼 지난 후 받은 보상의 현재 가치

$$\gamma^{k-1} R_{t+k}$$

- 할인율을 통해 보상을 얻는 최적의 경로를 찾을 수 있다



MDP 정리



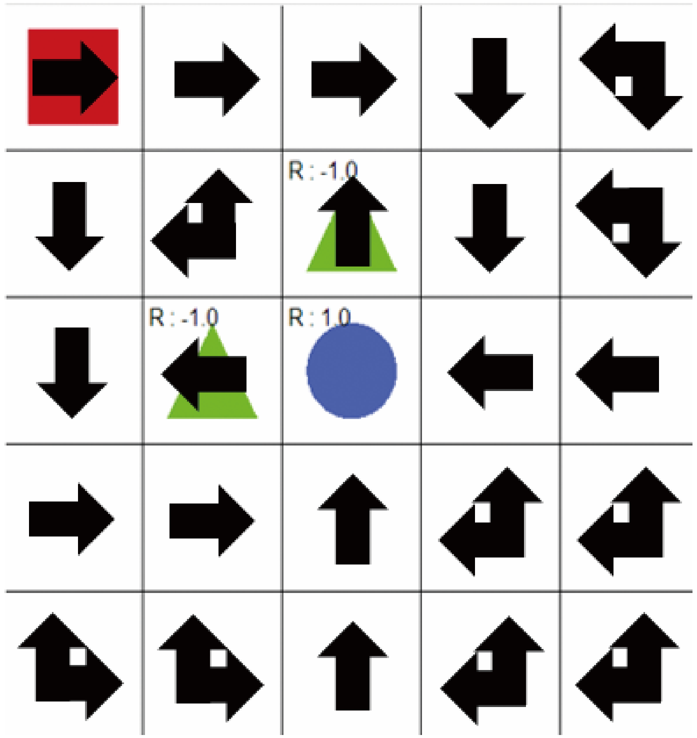
그리드월드 문제에서의 MDP

정책

1. 에이전트는 각 상태마다 행동을 선택
2. 각 상태에서 어떻게 행동할지에 대한 정보 : 정책(Policy)
   - 상태 s에서 행동 a를 선택할 확률

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

3. Stochastic Policy vs Deterministic Policy
4. 최적 정책 : 에이전트가 강화학습을 통해 학습해야 할 목표
   - 각 상태에서 하나의 행동만을 선택
   - Stochastic Policy -> Deterministic Policy



Summary

1. 강화학습이 풀고자 하는 문제
    - Sequential Decision Problem
2. Sequential Decision Problem의 수학적 정의
    - MP, MRP, MDP
3. MDP의 구성요소
    - 상태, 행동, 상태변환확률, 보상, 할인율
4. 각 상태에서 에이전트가 행동을 선택할 확률
    - 정책