lecture 1

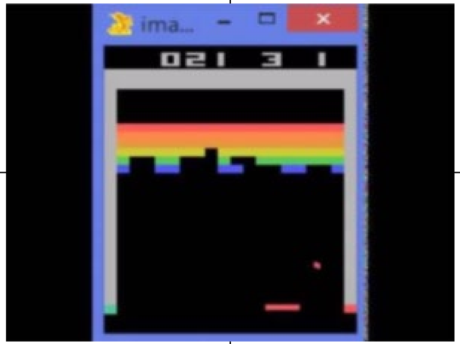
lecture 1

optimization

- Out of relative decisions → yield the decision with the best outcome

delayed consequences

- No immediate outcome feedback
- Induces credit assignment problem



exploration

- “Agent as a scientist”
- Reward predictable only for what the system has experienced (= outcomes based on previous decisions)
- Vs. exploitation

generalization

- Too much representations without generalization → require too much computing power
- Use a higher level representation of given task

Markov assumption

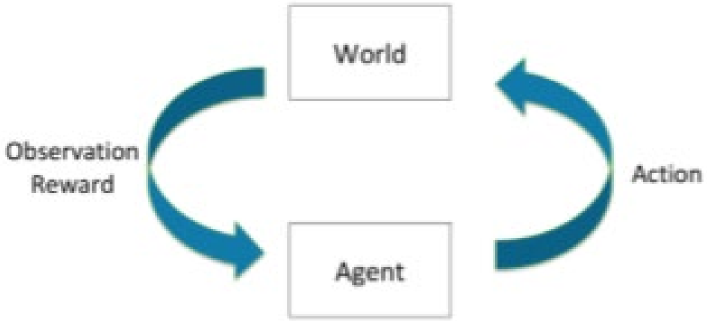
$$p(s_{t+1}|s_t, a_t) = p(s_{t+1}|h_t, a_t)$$

- Only require current state
- For predicting the future
- Independent of the past
- Although may use aggregate statistics (may be record of history : previous state, actions, rewards)

Think of other Markov systems:
- Knowledge of current blood pressure to determine medication control

lecture 1

sequential decision making



- Actions chosen in order to maximize total expected future reward

MDP

- Markov decision process
- Refer to Markov assumption

POMDP

- Partially observable MDP
- Many unknown factors of the world that can determine the observation & reward

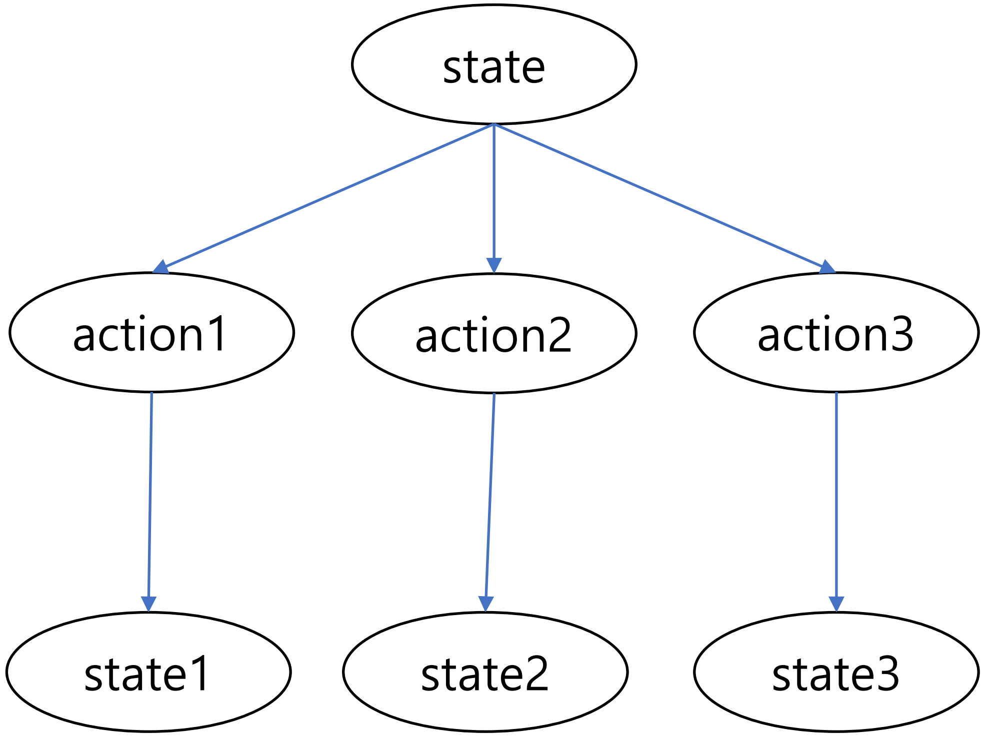
Bandit

- Actions have no influence on next observation & reward

lecture 1

deterministic policy

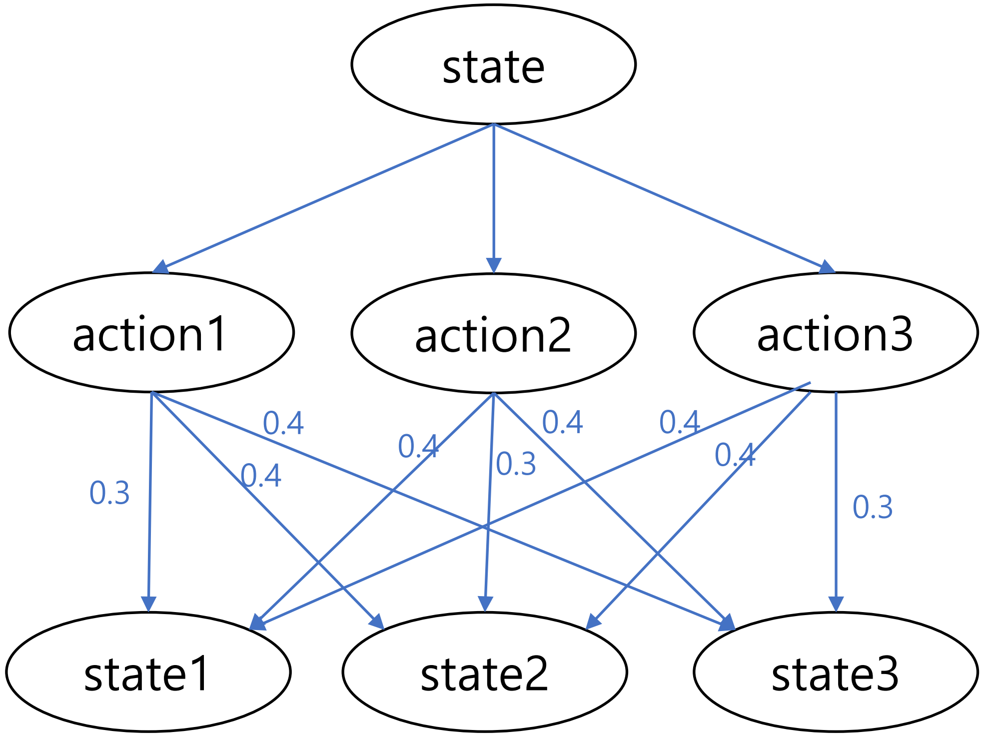
$$\pi(s) = a$$



- 100% certainty
- Definite next state

stochastic policy

$$\pi(a|s) = Pr(a_t = a|s_t = s)$$



- Many possible outcomes with relative probabilities
- Cannot be sure of next state

lecture 1

Value function

$$V^{\pi}(s_t = s) = \mathbb{E}_{\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots | s_t = s]$$

- “expected discounted sum of future rewards under a particular policy”
- Discount factor gamma weighs immediate vs future rewards

lecture 2

Simple definitions

- Model : “Mathematical models of dynamics and reward”
  = expected rewards from particular action and current state
- Policy : “Function mapping agent’s states to actions”
- Model : “future rewards from being in a state and/or action when following a particular policy”
  = expected discount sum

lecture 2

Markov chain (S, P)

- Memoryless random process(not totally) with no rewards and no actions

$$P = \begin{pmatrix} P(s_1|s_1) & P(s_2|s_1) & \cdots & P(s_N|s_1) \\ P(s_1|s_2) & P(s_2|s_2) & \cdots & P(s_N|s_2) \\ \vdots & \vdots & \ddots & \vdots \\ P(s_1|s_N) & P(s_2|s_N) & \cdots & P(s_N|s_N) \end{pmatrix}$$

Markov chain + rewards

MRP (S, P, R, gamma)

- Markov reward process
- No actions
- Value function = expected return

$$\begin{pmatrix} V(s_1) \\ \vdots \\ V(s_N) \end{pmatrix} = \begin{pmatrix} R(s_1) \\ \vdots \\ R(s_N) \end{pmatrix} + \gamma \begin{pmatrix} P(s_1|s_1) & \cdots & P(s_N|s_1) \\ P(s_1|s_2) & \cdots & P(s_N|s_2) \\ \vdots & \ddots & \vdots \\ P(s_1|s_N) & \cdots & P(s_N|s_N) \end{pmatrix} \begin{pmatrix} V(s_1) \\ \vdots \\ V(s_N) \end{pmatrix}$$

$$V = R + \gamma P V$$

Markov chain + rewards + actions

MDP (S, P, R, gamma, A)

- Markov decision process

$$P(s_{t+1} = s' | s_t = s, a_t = a)$$

$$R(s_t = s, a_t = a) = \mathbb{E}[r_t | s_t = s, a_t = a]$$

MRP + actions

- Simulation
- Analytic
- Iterative

MDP + policy

$(S, R^{\pi}, P^{\pi}, \gamma)$.

Within MDP...

There exists a unique optimal value function

=

Optimal policy in infinite horizon problem is deterministic

Policy Iteration

Q-function

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi}(s')$$

Policy improvement

$$\pi_{i+1}(s) = \arg\max_{a} Q^{\pi_i}(s,a) \; \forall s \in S$$

Note:

- Different to gradient based approaches – no problem with local minimum vs. global minimum / maximum

Policy Iteration

Policy improvement

$$\pi_{i+1}(s) = \arg\max_a Q^{\pi_i}(s, a) \ \forall s \in S$$

- Pi_i+1 only for the first action, and then follow pi_i
- Instead follow pi_i+1 onwards and it still monotonically improves

Policy Iteration

Monotonic improvement in policy value

$$
\begin{aligned}
V^{\pi_i}(s) &\le \max_a Q^{\pi_i}(s,a) \\
&= \max_a R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi_i}(s') \\
&= R(s,\pi_{i+1}(s)) + \gamma \sum_{s' \in S} P(s'|s,\pi_{i+1}(s)) V^{\pi_i}(s') \text{ //by the definition of } \pi_{i+1} \\
&\le R(s,\pi_{i+1}(s)) + \gamma \sum_{s' \in S} P(s'|s,\pi_{i+1}(s)) \left(\max_{a'} Q^{\pi_i}(s',a')\right) \\
&= R(s,\pi_{i+1}(s)) + \gamma \sum_{s' \in S} P(s'|s,\pi_{i+1}(s)) \\
&\quad \left(R(s',\pi_{i+1}(s')) + \gamma \sum_{s'' \in S} P(s''|s',\pi_{i+1}(s')) V^{\pi_i}(s'')\right) \\
&\ \ \vdots \\
&= V^{\pi_{i+1}}(s)
\end{aligned}
$$

Q. How many iterations should be required?

Or

Q. How many iterations with improvement can there be?

lecture 2

Value iteration

" Idea : maintain optimal value of starting in a state s if have a finite number of steps k left in the episode."   = assuming finite horizon?

" value iteration update is equal to policy evaluation update "

" value iteration update is equal to Bellman optimality equation into an update rule "

" value iteration combines one sweep of policy evaluation and one sweep of policy improvement "

Sutton, 82-83

lecture 2

Direction of understanding

Direction of application

Value iteration

Bellman backup operator

$$BV(s) = \max_a \left(R(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V(s') \right)$$

BV yields a new value function

Value iteration

$$V_{k+1} = BV_k$$

$$\pi_{k+1}(s) = \arg\max_a \left(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_k(s') \right)$$

Until V stops changing vs. no significant difference

$$V^{\pi} = B^{\pi} B^{\pi} \cdots B^{\pi} V$$

Policy iteration vs. Value iteration

" value iteration update is equal to policy evaluation update "

- Generally same thing
- But policy iteration is focused on updating the policy (given that value monotonically improves)
- And value iteration is focused on improving the value (utilizing a method that is same as updating and using the improved policy)

앞으로...

- 날짜 시간: 금 1400 ZOOM
- 어떻게 공부 할 것인가: 강의 2개 일단 다음부턴 무조건 3개씩
- 광훈 : 백준 100개 마스터 7월 31일까지
- 힘들면 편하게 얘기하기
- frozen lake 7월 31일까지
- 목표... 완강 assignment 따라하기