Lecture 17: Markov Decision Processes (I)

EMAT31530/April 2018/Raul Santos-Rodriguez

The road so far

- Machine learning

  Binary classification:

  $x \to \boxed{?} \to y \in \{-1, +1\}$, single action

- Search
- Bayesian networks
- MDP/RL
- Game theory

The road so far

- Machine learning

  Binary classification:

  $x \rightarrow \boxed{?} \rightarrow y \in \{-1, +1\}$, single action

- Search

  Search problem:

  $x \rightarrow \boxed{?} \rightarrow$ action sequence $(a_1, a_2, a_3, a_4, \dots)$

- Bayesian networks
- MDP/RL
- Game theory

The road so far

- Machine learning

  Binary classification:

  $x \rightarrow \boxed{?} \rightarrow y \in \{-1, +1\}$, single action

- Search

  Search problem:

  $x \rightarrow \boxed{?} \rightarrow$ action sequence $(a_1, a_2, a_3, a_4, \ldots)$

- Bayesian networks

  Bayesian Networks:

  Model uncertainty

- MDP/RL

- Game theory

The road so far

- Machine learning

  Binary classification:

  $x \rightarrow \boxed{?} \rightarrow y \in \{-1, +1\}$, single action

- Search

  Search problem:

  $x \rightarrow \boxed{?} \rightarrow$ action sequence $(a_1, a_2, a_3, a_4, \ldots)$

- Bayesian networks

  Bayesian Networks:

  Model uncertainty

- MDP/RL

  Decision making in an uncertain world

  Search: state s and action $a \xrightarrow{\textit{deterministic}}$ state s'
  MDPs: state s and action $a \xrightarrow{\textit{randomness}}$?

- Game theory

The road so far

- Machine learning — Binary classification:

  $x \to \boxed{?} \to y \in \{-1, +1\}$, single action

- Search — Search problem:

  $x \to \boxed{?} \to$ action sequence $(a_1, a_2, a_3, a_4, \ldots)$

- Bayesian networks — Bayesian Networks:

  Model uncertainty

- MDP/RL — Decision making in an uncertain world

  Search: state s and action $a \xrightarrow{\text{deterministic}}$ state s'
  MDPs: state s and action $a \xrightarrow{\text{randomness}}$?

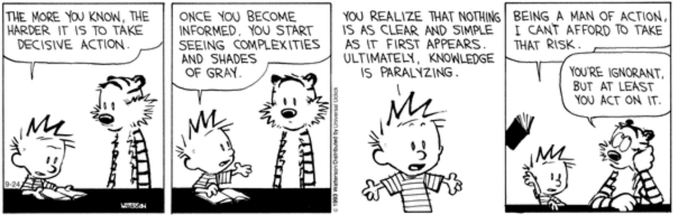
- Game theory

In the real world, the deterministic assumption is often unrealistic and there is randomness: taking an action might lead to any one of many possible states. We will study the tools to tackle this more challenging setting. We will fortunately still be able to reuse many of the intuitions about search problems, in particular the notion of a state.

Have a look at ...

... Russell and Norvig (Ch. 17.1)

... OpenAI: `https://gym.openai.com/`



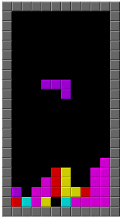
Outline

This lecture discusses complex decision making. The objective is to present the foundations of Markov Decision Processes:

- Sequential decision problems
- Rewards, Utiliy and Policies

Many important problems are MDPs ...

- Cleaning robot
- Autonomous aircraft navigation
- Games
- Network switching and routing
- Travel route planning
- Models of animals



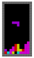
2018-04-14

Motivation

Motivation

Many important problems are MDPs ...

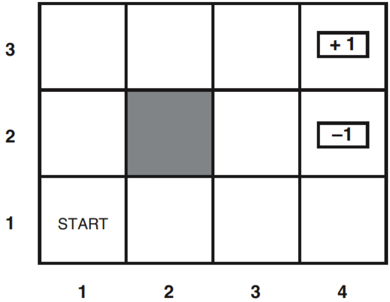
- Cleaning robot
- Autonomous aircraft navigation
- Games
- Network switching and routing
- Travel route planning
- Models of animals



Randomness shows up in many places. They could be caused by limitations of the sensors and actuators of the robot (which we can control to some extent). Or they could be caused by market forces or nature, which we have no control over.

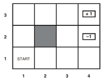
Motivation

Example: Deterministic Grid World



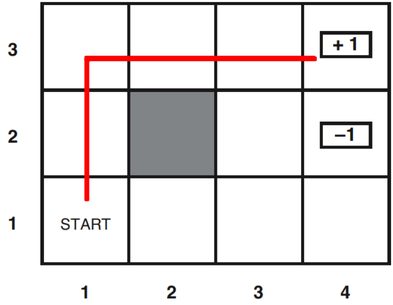
2018-04-14

Example: Deterministic Grid World



Suppose that a robot is situated in the 4x3 environment shown in the figure. Beginning in the start state, it must choose an action at each time step. The interaction with the environment terminates when the robot reaches one of the goal states, marked $+1$ or -1. Just as for search problems, the actions available to the robot in each state are given by Actions(s), sometimes abbreviated to A(s); in the 4x3 environment, the actions in every state are Up, Down, Left, and Right. We assume that the environment is fully observable, so that the robot always knows where it is.

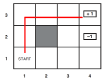
Example: Deterministic Grid World



2018-04-14

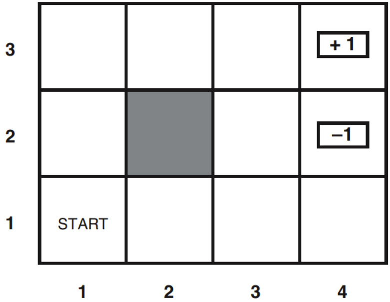
Example: Deterministic Grid World

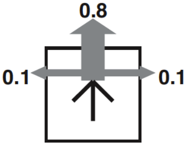
Example: Deterministic Grid World



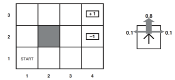
If the environment were deterministic, a solution would be easy: [Up, Up, Right, Right, Right].

Example





Example



Unfortunately, the environment won't always go along with this solution, because the actions are unreliable. In the example, each action achieves the intended effect with probability 0.8, but the rest of the time, the action moves the robot at right angles to the intended direction. Furthermore, if the robot bumps into a wall, it stays in the same square. For example, from the start square (1,1), the action Up moves the robot to (1,2) with probability 0.8, but with probability 0.1, it moves right to (2,1), and with probability 0.1, it moves left, bumps into the wall, and stays in (1,1). In such an environment, the sequence [Up, Up, Right , Right , Right] goes up around the barrier and reaches the goal state at (4,3) with probability $0.8^5 = 0.32768$. There is also a small chance of accidentally reaching the goal by going the other way around with probability $0.1^4 \cdot 0.8$, for a total of 0.32776.

Definitions

[Transition model] describes the outcome of each *action* in each *state*.

$P(s'|s,a) \rightarrow$ probability of reaching state s' if action a is done in state s.

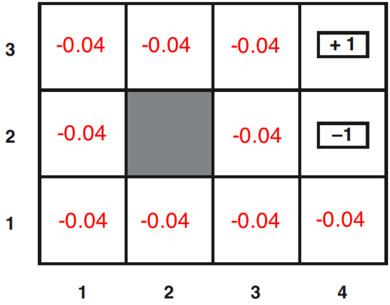
[Markov assumption] The probability of reaching s' from s depends only on s and not on the history of earlier states.

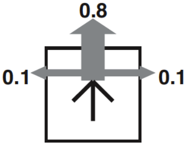
Transition model can be represented as a Dynamic Bayesian Network.

[Rewards] In each state s, we receive a reward $R(s)$ (positive or negative but bounded).

[Utility or Value function] For now, the utility U_h (or V_h) is the sum of the rewards received. The utility function will depend on a sequence of states rather than on a single state.

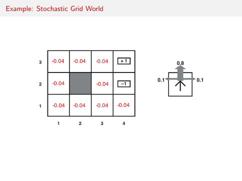
Example: Stochastic Grid World





2018-04-14

Example: Stochastic Grid World



For our particular example, the reward is -0.04 in all states except the terminal states (which have rewards +1 and -1). For example, if the robot reaches the +1 state after 10 steps, its total utility will be 0.6. The negative reward of -0.04 gives the robot an incentive to reach (4,3) quickly.

MDP formulation

MDP requires a structure to keep track of the decision sequences:

MDP

- s: state
- s_{start}: starting state
- Actions(s): possible actions
- $P(s'|s, a)$ (or $T(s, a, s')$): probability of s' if take action a in state s
- Reward(s), R(s), r(s): reward for the state s
- Goal(s): whether at the end of the process
- $U_h([s_1, s_2, \ldots])$: utility or value of a sequence of states

Policies

A solution should describe what the robot does in every state: this is called a policy, π.

- $\pi(s)$ for an individual state describes which action should be taken in s.

Each time a given policy is executed starting from the initial state, the stochastic nature of the environment may lead to a different environment history.

Optimal policy is one that yields the highest *expected utility*, denoted by π^*

2018-04-14

└─Policies

Policies

A solution should describe what the robot does in every state: this is called a policy, π.

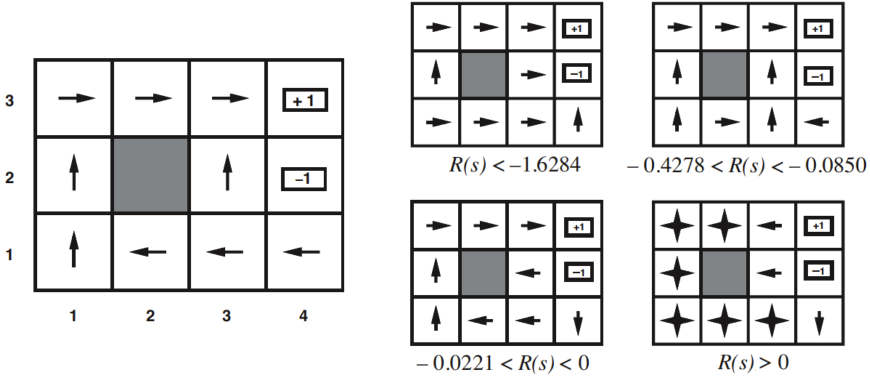
- $\pi(s)$ for an individual state describes which action should be taken in s.

Each time a given policy is executed starting from the initial state, the stochastic nature of the environment may lead to a different environment history.

Optimal policy is one that yields the highest *expected utility*, denoted by π^*

In an MDP, we want an optimal policy (a policy that maximizes the expected sum of rewards). In contrast, in a deterministic setting, we want an optimal plan, or sequence of actions, from start to a goal.

Policies: Example

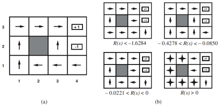


(a) (b)

2018-04-14

Policies: Example

Policies: Example



(a) An optimal policy for the stochastic environment with $R(s) = -0.04$ in the nonterminal states. (b) Optimal policies for four different ranges of $R(s)$.

Utilities over time

Finite horizon or infinite horizon?

Finite horizon

There is a fixed time N after which nothing matters:

- $\forall k \quad U_h([s_0, s_1, ..., s_{N+k}]) = U_h([s_0, s_1, \ldots, s_N])$
- Leads to non-stationary optimal policies (N matters)

Infinite horizon

Stationary optimal policies (time at state doesn't matter):

- Does **not** mean that all state sequences are infinite; it just means that there is no fixed deadline.
- If two state sequences $[s_0, s_1, s_2, \ldots]$ and $[s'_0, s'_1, s'_2, \ldots]$ begin with the same state (i.e., $s_0 = s'_0$), then the two sequences should be preference-ordered the same way as the sequences $[s_1, s_2, \ldots]$ and $[s'_1, s'_2, \ldots]$.

Additive rewards

Additive rewards

The utility of a state sequence is

$$U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \ldots.$$

2018-04-14

Additive rewards

Additive rewards

The utility of a state sequence is

$$U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \ldots.$$

The 4x3 world uses additive rewards. Notice that additivity was used implicitly in our use of path cost functions in heuristic search algorithms.

Additive rewards

A lecturer gets paid, say, 20K per year.
How much, in total, will the lecturer earn in his/her life?

$$20 + 20 + 20 + 20 + 20 + \ldots = \infty$$

What's wrong with this argument?

Discounted rewards

A reward (payment) in the future is not worth quite as much as a reward now.

- Because of chance of obliteration
- Because of inflation

Discounted rewards

The utility of a state sequence is

$$U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots,$$

where the discount factor γ is a number between 0 and 1.

Discount factor makes more distant future rewards less significant!

Discounted rewards

A reward (payment) in the future is not worth quite as much as a reward now.

- Because of chance of obliteration
- Because of inflation

Discounted rewards

The utility of a state sequence is

$$U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots,$$

where the discount factor γ is a number between 0 and 1.

Discount factor makes more distant future rewards less significant!

The discount factor describes the preference for current rewards over future rewards. When γ is close to 0, rewards in the distant future are viewed as insignificant. When γ is 1, discounted rewards are exactly equivalent to additive rewards, so additive rewards are a special case of discounted rewards. People in economics and probabilistic decision making do this all the time. Discounting appears to be a good model of both animal and human preferences over time. A discount factor of γ is equivalent to an interest rate of $(1/\gamma) - 1$.

Infinite horizon rewards

Choosing infinite horizon rewards creates a problem: some sequences will be infinite with infinite reward, **how do we compare them?**

[Solution 1] With discounted rewards, the utility of an infinite sequence is finite. In fact, if $\gamma < 1$ and rewards are bounded by $\pm R_{max}$, we have

$$U_h([s_0, s_1, s_2, \ldots]) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \leq \sum_{t=0}^{\infty} \gamma^t R_{max} = \frac{R_{max}}{(1-\gamma)}$$

[Solution 2] Under proper policies, i.e. if we will eventually visit terminal state, additive rewards are finite.

[Solution 3] Compare average reward per time step.

Problem

For each round $r = 1, 2, \ldots$

- You choose stay or quit.
- If quit, you get £10 and we end the game.
- If stay, you get £4 and then I roll a 6-sided dice.
  - If the dice results in 1 or 2, we end the game.
  - Otherwise, continue to the next round.

Question

What is the expected utility if we follow the policy stay? and if we follow the policy quit?

Next lecture

How do we deal with MDP in practice?

- Value iteration
- Policy iteration