50.021 – AI

Alex

Week 06: Markov Decision Processes

[The following notes are compiled from various sources such as textbooks, lecture materials, Web resources and are shared for academic purposes only, intended for use by students registered for a specific course. In the interest of brevity, every source is not cited. The compiler of these notes gratefully acknowledges all such sources.]

Plan for RL lectures:

most mathematical part of the whole AI lectures, search and CSPs afterwards will be much easier

- MDPs, value-iteration
- coding: value iteration on a toy grid problem
- Q-iteration
- Q-learning, Sarsa, eligibility traces
- coding: Q-learning, Sarsa on a toy grid problem
- deep Q-learning
- coding: take deep-Q learning code from pytorch, adapt it to some faster/simpler openAI gym problems
- policy gradient methods, reinforce, A2C
- coding: reinforce, A2c on some faster/simpler openAI gym problems
- see the instability and problems when learning from experience

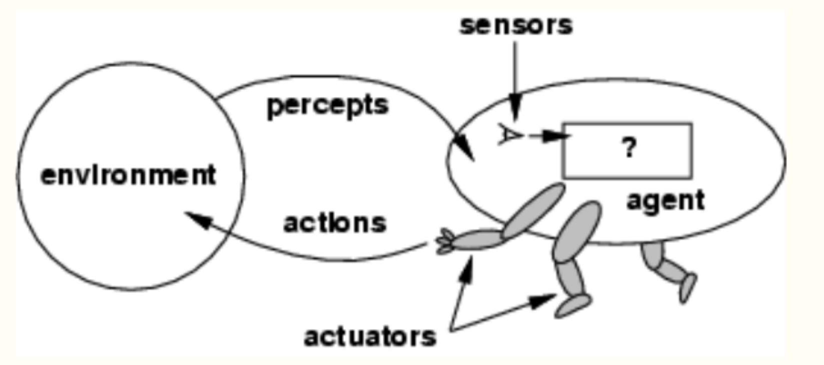
Key takeaways:

Be able to explain the main ideas behind:

- the concept of an agent, world states, world types/properties
- concepts: expected reward, value function, policy
- MDP
- Bellman equation for any policy
- Bellman optimality criterion
- Estimating $V^{\pi^*}(s)$ and $\pi^*(s)$ and proof of its convergence

1 The agent in a world

Concept of an agent interacting with its environment.



An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators
We assume the agent is in interaction with a domain.

- The world is in a state $s_t \in S$
- The agent senses observations $o_t \in O$
- The agent decides on an action $a_t \in A$
- The world transitions in a new state s_{t+1}
- specific to reinforcement learning: agent receives a reward $r_t \in \mathbb{R}^1$ at some time steps
- S – set of all possible states
- O,A analogously

The domain is the model for the setup of world (S,O,A).
What is a valid state description s_t? Intuitively:

- contains everything about the world, that is relevant in the following sense:
- no additional hidden variables that are needed to know in order to be able to model future observations $o_{t^+}, t^+ > t$ and future observations $o_{t^+}, t^+ > t$, under the assumptions that one knows all future actions until t^+: $\{a_{t:t^+}\}$ and the current state s_t

Formally: s_t is a valid state description iff any future observation $o_{t^+}, t^+ > t$ is conditionally statistically independent of every historical observation $o_{t^-}, t^- < t$ given s_t and all future actions until t^+: $\{a_{t:t^+}\}$

Recap conditional independence:

What is the meaning of a conditional probability like: $P(X_{set1}|X_{set3})$?

Conditioning on X_{set3} as in $P(X_{set1}|X_{set3})$ means 1. to have observed / to know the values for the variables in X_{set3} and 2. computing the probability for (values of) the variables in X_{set1} given a world for which we know what values the variables in X_{set3} have.

Two sets of random variables X_{set1} and X_{set2} are **conditionally independent** given a third set X_{set3} if for all values of the variables in X_{set_k} one of the three conditions hold:

$$P(X_{set1}, X_{set2}|X_{set3}) = P(X_{set1}|X_{set3})P(X_{set2}|X_{set3})$$
$$P(X_{set1}|X_{set2}, X_{set3}) = P(X_{set1}|X_{set3})$$
$$P(X_{set2}|X_{set1}, X_{set3}) = P(X_{set2}|X_{set3})$$

The third has a nice interpretation:

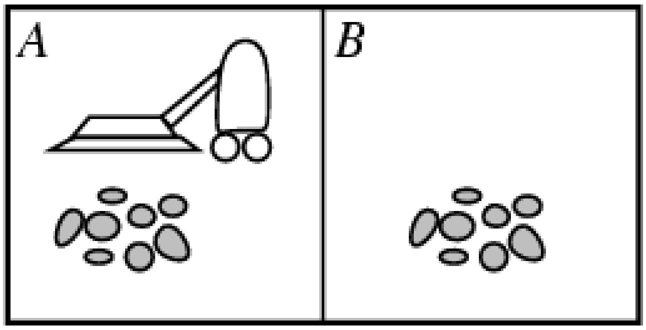
X_{set1} and X_{set2} are conditionally independent given X_{set3} means: **having observed the value of X_{set1} does not change the probability of** $P(X_{set2}|X_{set3})$ – because the probability after plugging in the observations for X_{set1}: $P(X_{set2}|X_{set1}, X_{set3})$ is just the same :).

In real worlds, the exact state is not representable. All models of domains will make approximating assumptions (e.g. independencies)

a principled way to describe agents and the world in interaction – PEAS:

- performance measure
- environment
- actuators
- sensors

World state: one state of this world.



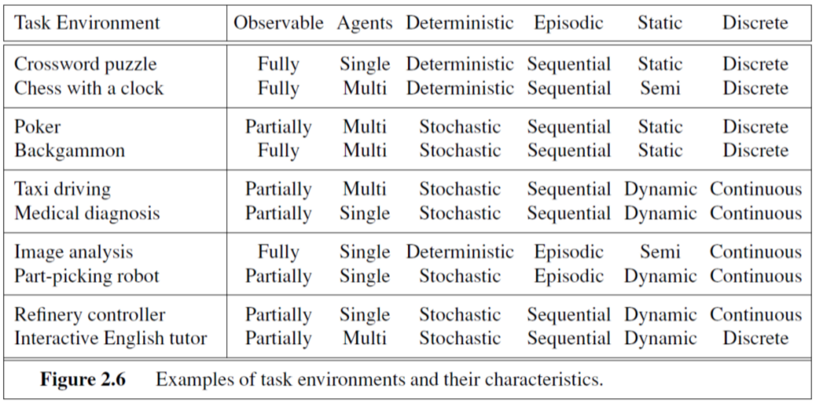
from Russell & Norvig

- world state contains what information?
- performance measure of agent is what ?

- set of all possible locations, whether they are clean or dirty
- how locations are connected to each other (for being able to model the effect of movement actions)!
- position of the agent!

Above is very general. There are many flavors of this:

- Fully observable vs. partially observable world
- Single agent vs. multiagent
- Deterministic vs. stochastic world (state transitions, effect of actions, observations)
- Structure of the state space: Discrete, continuous, hybrid, factored as a product of some spaces
- Discrete vs. continuous time
- (static vs dynamic, episodic vs sequential)



Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Figure 2.6 Examples of task environments and their characteristics.

from Russell & Norvig

2 Possible directions – overview

only stationary MDP, RL covered in this course

- MDP
  - know probability how the world state s_t changes under an action a into a new state s_{t+1}: $P_t(s_{t+1}|s_t, a)$
  - know the probability/function for rewards $R_t(s, a, s)$
  - agent senses the world state s_t
  - most general agent maps $s_t, (a_0, \dots, a_{t-1}) \mapsto a_t$ to an action
  - stationary MDP: $P_t = P$, $R_t = R$ are the same for all time steps
- POMDP: partially observable MDP

- agent senses observations y_t.
  - most general agent maps $(y_0, \dots, y_{t-1}, \mathbf{y_t}), (a_0, \dots, a_{t-1}) \mapsto a_t$ to an action
  - options: agent may keep an internal state n_t, then do two-step model to generate a new action: $(n_{t-1}, y_t) \mapsto n_t$, $n_t \mapsto a_t$
  - options: agent may deduce a probability distribution $b_t(\cdot)$ over world states $b_t(s_t)$, then perform $(b_{t-1}, y_t) \mapsto b_t$, $b_t \mapsto a_t$

- Decentralized POMDP: partially observable MDP with multi-agents
- RL:
  - in RL we dont know $P(s_{t+1}|s_t, a)$, $R(s, a, s)$.
  - learn from observations (s_t, a_t, r_t, s_{t+1}) – what happens when you apply in s_t action a_t
  - several ways: learning P, R, learning Q, V (see next lectures), direct learning of a policy $\pi(a|s)$
- continuous time I: Control theory
  x is system state, y observable, u control input. Differential equations are just another way to describe state transitions under a control input u

  linear:

  $$\frac{dx}{dt} = Ax + Bu$$
  $$\frac{dy}{dt} = Cx + Du$$

  A, B, C, D matrices

  non-linear:

  $$\frac{dx}{dt} = f(x, u)$$
  $$\frac{dy}{dt} = g(x, u)$$

  f, g functions

  an agent could be for example a simple linear feed-back regulator $u = Ky$ (memory-free, state-free), stimulus-response approach. Reward $\Leftrightarrow$ negative loss from some objective function, e.g. a potential function on values of x.
- continuous time II: stochastic control – the differential equations become stochastic differential equations with some noise

  $$dx = f(x, u)dt + d\eta^{(x)}$$
  $$dy = g(x, u)dt + d\eta^{(y)}$$

  $d\eta$ could be Wiener processes `https://en.wikipedia.org/wiki/Wiener_process`, that is a time series, continuous, but with randomized (+ independent) increments $d\eta_{u+t} - d\eta_t$ governed by a normal distribution $d\eta_{u+t} - d\eta_t \sim N(0, \sigma^2 = u)$

3 MDP

A very simple model. Most ideas in RL are based on it.

- have world states, actions, rewards
- $P(s'|a,s)$, $R(s,a,s')$ – transition probability and reward function (can be a probability e.g. $R(s,a,s') = N(\hat{R}(s,a,s'),\sigma^2)$, where $\hat{R}(s,a,s')$ is a function)

MDP

an MDP is a 4-tuple: S, A, P, R

- S – state space, the set of all world states,
- A – action space, the set of all action
- P – the transition probabilities. $P(s'|a,s)$ is the probability to go to world state s' conditioned on the facts that the world was in state s, and the agent did action a
- R – reward function, potentially depends on s, a, s', the old state, the action and the new state

The consequence of an MDP as model for an agent-world interaction: We know the transition probabilities in the world and the reward function. Difference to true RL later on:

- We do not need to learn anything from actions taken and rewards received.
- we do not need to infer anything about the world state from observations. We know the world state we are in precisely. (no need for experience, we know everything).

In MDPs the world state can be known / perfectly sensed by the agent (compare to partially observable environments) and all dynamics are known.

3.1 policy, goals and rewards

policy

Policy $\pi(a|s)$ – what action to choose given that one is in state s. Can be a probability over all allowed actions.

What is the performance measure of an agent? See PEAS-P.

Expected (future) reward

for finite time horizons one can use a plain sum

$$\bar{r}(s) = E_{(a_0\sim\pi(a|s_0=s),a_1\sim\pi(a|s_1=s_1),\ldots,a_t\sim\pi(a|s_t=s_t),\ldots)}[\sum_{t=0}^{T_e} r_t|s_0 = s]$$

for unbounded time horizons and in general with a discount factor $\gamma \in (0,1]$

$$\bar{r}(s) = E_{(\dots,a_t\sim\pi(a|s_t=s_t),\dots)}[\sum_{t=0}^{\infty}\gamma^t r_t|s_0 = s]$$

the expectation runs over drawing an action for every time step.

the Value function

The value $V^\pi(s)$ of a policy π is the gamma-discounted expected future reward when starting in state s and continuing according to the policy:

$$V^\pi(s) = E_{(a_0\sim\pi(a|s_0=s),a_1\sim\pi(a|s_1=s_1),\dots,a_t\sim\pi(a|s_t=s_t),\dots)}[\sum_{t=0}^{T_e}\gamma^t r_t|s_0 = s]$$

A policy π^* is optimal if

$$\forall s \; V^{\pi^*}(s) = \sup_{\pi} V^\pi(s)$$

Note:

- in MDPs with a finite state and action space exists at least one optimal deterministic policy (the space of all policies is final then :))
- the action drawing $a_t \sim \pi(a|s_t = s_t)$ can be probabilistic
- the state transition ... $s_{t+1} \sim P(s'|s = s_t, a = a_t)$
- the reward as well a probability, reward can depend on s_t, a_t, s_{t+1}

The impact of the discount factor: $\gamma = 0$ greedy on the current time step. γ larger – higher weight to achieving future rewards. If the rewards are bounded, then for any $\gamma \in (0,1)$ the expected reward must be finite, but not for $\gamma = 1$.

the goal in MDPs

goal: find a policy π^* which maximizes V^π, the expected reward, averaged over all uncertainties in state transitions, actions and rewards.

3.2 Specifying a stationary MDP

$P(s'|s,a)$, $R(s,a)$ (possibly a random variable with a density for each s,a which integrates over all real numbers to 1), $P(s_0)$ - initial state distribution, $\pi(a|s_t)$ - policy, the probability of an action given a state.

3.3 Properties of the Value function

$$\begin{aligned} V^\pi(s) &= E[r_0 + \gamma r_1 + \gamma^2 r_2 + \dots|s_0 = s,\pi] \\ &= E[r_0|s_0 = s,\pi] + \gamma E[r_1 + \gamma^1 r_2 + +\gamma^2 r_3 + \dots|s_0 = s,\pi] \end{aligned}$$

If Rewards and policies are deterministic $R = R(s,a,s')$, $a = \pi(s)$, then

$$V^\pi(s) = \sum_{s'} P(s'|s,\pi(s))R(s,\pi(s),s') + \gamma \sum_{s'} P(s'|s,\pi(s)) \underbrace{E[r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots |s_1 = s', \pi]}_{=V^\pi(s') \text{ !!}}$$

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

If Rewards and policies are not deterministic, then we simply have to compute the expectation of any function $T(a)$, which depends on actions, over all actions weighted by the probability $\pi(a|s)$: $T(a=\pi(s)) \rightsquigarrow \sum_a \pi(a|s)T(a)$.

This applies for $T(a) = R(s,a,s')$ and for $T(a) = P(s'|a,s)$

In that case compare the deterministic with the probabilistic version:

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

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

The Bellman equation

If rewards and policy are deterministic, that is when the policy π returns one state $a = \pi(s)$ as a function of the current state, then

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

In the general case, replacing any terms depending on an action $T(a=\pi(s)) \rightsquigarrow \sum_a \pi(a|s)T(a)$ by the expectation under probability of an action (, we have:

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

This applies for $T(a) = R(s,a,s')$ and for $T(a) = P(s'|a,s))$

3.4 The Bellman optimality criterion

Consider the case of a deterministic policy $a = \pi(s)$.
We have for any deterministic policy:

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

Remember: A policy π^* is optimal if

$$\forall s \; V^{\pi^*}(s) = \sup_\pi V^\pi(s)$$

Consider any policy π_1. It will choose in s the action $\bar{a} = \pi_1(s)$. What happens if there exists an action a^* such that we use a policy π_2 from π_1:

$$\pi_2(s') = \begin{cases} \pi_1(s') & \text{if } s' \neq s \\ a^* & \text{if } s' = s \end{cases}$$

and it holds

$$\begin{aligned} V^{\pi_1}(s) &= \sum_{s'} P(s'|s, a = \pi_1(s)) R(s, a = \pi_1(s), s') + \gamma \sum_{s'} P(s'|s, a = \pi_1(s)) V^{\pi_1}(s') \\ &< \sum_{s'} P(s'|s, a = a^*) R(s, a = a^*, s') + \gamma \sum_{s'} P(s'|s, a = a^*) V^{\pi_2}(s') \ \ ? \end{aligned}$$

In that case we would have in s: $V^{\pi_2}(s) > V^{\pi_1}(s)$.
As a consequence, we can improve the value of any policy $V^{\pi_1}(s)$ in s if

$$V^{\pi_1}(s) < \max_a \sum_{s'} P(s'|s, a) R(s, a, s') + \gamma \sum_{s'} P(s'|s, a) V^{\pi_2}(s')$$

Therefore: if π_1 is the optimal policy, then it must hold in s that

$$V^{\pi_1}(s) = \max_a \sum_{s'} P(s'|s, a) R(s, a, s') + \gamma \sum_{s'} P(s'|s, a) V^{\pi_1}(s')$$

The Bellman optimality criterion

In an MDP the optimal Value and the optimal policy satisfy the following equations:

$$\begin{aligned} V^{\pi^*}(s) &= \max_a & \sum_{s'} P(s'|s, a) R(s, a, s') + \gamma \sum_{s'} P(s'|s, a) V^{\pi^*}(s') \\ \pi^*(s) &= \text{argmax}_a & \sum_{s'} P(s'|s, a) R(s, a, s') + \gamma \sum_{s'} P(s'|s, a) V^{\pi^*}(s') \end{aligned}$$

3.5 Estimating $V^{\pi^*}(s)$ and $\pi^*(s)$

Ok, we know what condition they should satisfy, ...
By fixpoint iteration! Start with $V_0(s) = 0$. Iterate: Compute $V_{k+1}(s)$ from $V_k(s)$

$$\begin{aligned} V^{\pi^*}(s) &= \max_a \sum_{s'} P(s'|s, a) R(s, a, s') + \gamma \sum_{s'} P(s'|s, a) V^{\pi^*}(s') \\ \rightsquigarrow V_{k+1}(s) &= \max_a \sum_{s'} P(s'|s, a) R(s, a, s') + \gamma \sum_{s'} P(s'|s, a) V_k(s') \end{aligned}$$

until an iteration k such that: $\max_s |V_{k+1}(s) - V_k(s)| < \delta$

Value iteration

The algorithm above. Computes the Value under the optimal policy π^*.

Why does that converge ?? Here comes a proof:

Lets define a Bellman operator:

$$BV(s) = \max_a R(s,a) + \gamma \sum_{s'} P(s'|s,a)V(s')$$

Then:

$$|BV^1(s) - BV^2(s)| = |\max_a \left(R(s,a) + \gamma \sum_{s'} P(s'|s,a)V^1(s') \right) - \max_{a'} \left(R(s,a') + \gamma \sum_{s'} P(s'|s,a')V^2(s') \right)|$$

Note:

$$|\max_a f(a) - \max_{a'} g(a')| \leq \max_a |f(a) - g(a)|$$

so:

$$\begin{aligned}
|BV^1(s) - BV^2(s)| &= |\max_a \left(R(s,a) + \gamma \sum_{s'} P(s'|s,a)V^1(s') \right) - \max_{a'} \left(R(s,a') + \gamma \sum_{s'} P(s'|s,a')V^2(s') \right)| \\
&\leq \max_a \left| \left(R(s,a) + \gamma \sum_{s'} P(s'|s,a)V^1(s') \right) - \left(R(s,a) + \gamma \sum_{s'} P(s'|s,a)V^2(s') \right) \right| \\
&= \max_a \left| \gamma \sum_{s'} P(s'|s,a)V^1(s') - \gamma \sum_{s'} P(s'|s,a)V^2(s') \right| \\
&= \max_a \gamma \left| \sum_{s'} P(s'|s,a)V^1(s') - \sum_{s'} P(s'|s,a)V^2(s') \right| \\
&\leq \max_a \gamma \sum_{s'} P(s'|s,a) \left| V^1(s') - V^2(s') \right| \\
&= \gamma \sum_{s'} P(s'|s,a^*) \left| V^1(s') - V^2(s') \right| \text{ for some } a^* \\
&\leq \gamma \sum_{s'} P(s'|s,a^*) \max_s \left| V^1(s) - V^2(s) \right| \text{ for some } a^* \\
&= \gamma \max_s \left| V^1(s) - V^2(s) \right|
\end{aligned}$$

This holds for every s, so also for the max over all s, therefore:

$$\max_s |BV^1(s) - BV^2(s)| \leq \gamma \max_s \left| V^1(s') - V^2(s') \right|$$

Now plug in: $V^2 = V^\pi$. This is a fixed point, means by Bellman equation $BV^\pi(s) = V^\pi(s)$. Now plug in: $V^1 = V_k$. By definition $BV_k = V_{k+1}$. Therefore:

$$\begin{aligned}
&\max_s |V_{k+1}(s) - V^\pi(s)| = \max_s |BV_k(s) - BV^\pi(s)| \leq \gamma \max_s |V_k(s) - V^\pi(s)| \\
\Rightarrow &\max_s |V_{k+r}(s) - V^\pi(s)| \leq \gamma^r \max_s |V_k(s) - V^\pi(s)|
\end{aligned}$$

Therefore: if $\gamma < 1$, then $\forall\ s : |V_{k+r}(s) - V^\pi(s)| \overset{r \to \infty}{\to} 0$
and therefore $V_u(s) \overset{u \to \infty}{\to} V^\pi(s)$.