Lecture 1 and 2: Value, Expected Value, and Optimal Decisions

Reinforcement Learning and Decision Making MLSALT7, Lent 2016

Matthew W. Hoffman, Zoubin Ghahramani, Carl Edward Rasmussen

Department of Engineering
University of Cambridge

http://mlg.eng.cam.ac.uk/teaching/mlsalt7/

Reinforcement Learning and Decision Making

Reinforcement learning is an area of machine learning concerned with how an agent can perform actions to receive rewards from an evironment.

Reinforcement Learning is closely related to models from various disparate fields such as behaviourist psychology, game theory and control theory.

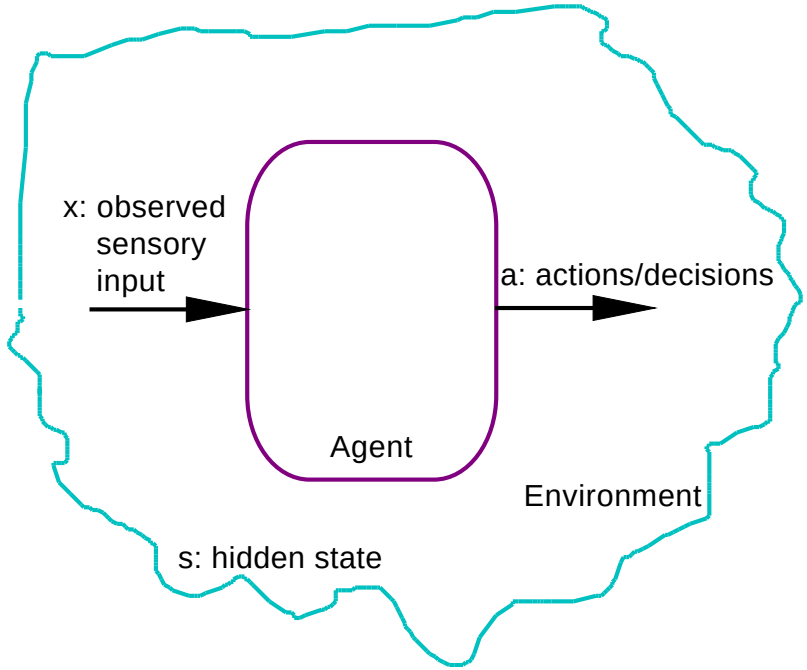
Although details vary, in each case a central question is what are the rational and/or optimal actions when faced with various types of information.

Intelligent Behaviour?

Imagine
a creature/agent (human/animal/machine)
which receives sensory inputs and
can take some actions in an environment:

Assume
that the creature also receives rewards
(or penalties/losses) from the environment.

The goal of the creature
is to maximise the rewards it receives
(or equivalently minimise the losses).



A theory for choosing actions that minimize losses is a theory for how to behave optimally...

Examples

- Autonomous helicopters
- Atari games

Bayesian Decision Theory

Bayesian decision theory deals with the problem of making optimal decisions—that is, decisions or actions that minimize an expected loss.

- Let's say we have a choice of taking one of k possible **actions** $a_1 \ldots a_k$.
- Assume that the world can be in one of m different **states** $s_1, \ldots, s_m$.
- If we take action a_i and the world is in state s_j we incur a **loss** ℓ_{ij}
- Given all the observed data $\mathcal{D}$ and prior background knowledge $\mathcal{B}$, our **beliefs** about the state of the world are summarized by $p(s|\mathcal{D}, \mathcal{B})$.
- *The optimal action is the one which is expected to minimize loss (or maximize utility)*:

$$a^* = \arg\min_{a_i} \sum_{j=1}^{m} \ell_{ij}\, p(s_j|\mathcal{D}, \mathcal{B})$$

Bayesian sequential decision theory	(statistics)
Optimal control theory	(engineering)
Reinforcement learning	(computer science / psychology)

Making decisions

We can often frame decisions using a table of our payouts. Let's say we are diagnosed with cancer and have to decide between treatment A or treatment B with the following payouts:

A	-10
B	-100

with these payouts the optimal action is easily seen to be treatment A. But what if there is uncertainty in the state:

	p_1 cancer	$1 - p_1$ no cancer
A	-10	-50
B	-100	0

$$V(A) = -10p_1 - 50(1 - p_1) = -50 + 40p_1$$
$$V(B) = -100p_1$$

What if treatment A has probability p_2 of eliminating the cancer, and otherwise we must only then consider taking actions A, B, or some new action C.

In this course we will consider problems wherein our actions have an effect on the state of the world and we must solve a sequential decision process.

Markov decision processes

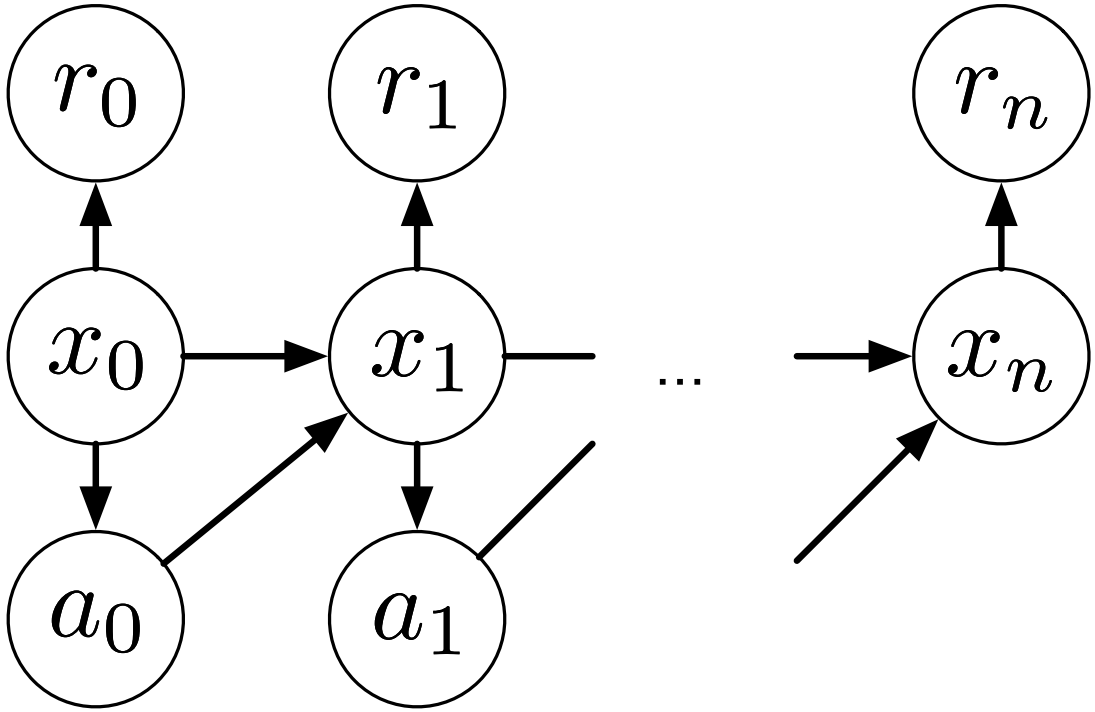
An MDP is a tuple $(\mathcal{X}, \mathcal{A}, p, r)$ where

- $\mathcal{X}$ is a set of states;
- $\mathcal{A}$ is a set of actions;
- $p(z|x, a)$ is a transition distribution defining the probability of moving to state $z \in \mathcal{X}$ from state $x \in \mathcal{X}$ on action $a \in \mathcal{A}$;
- and $r(x)$ is a reward function.

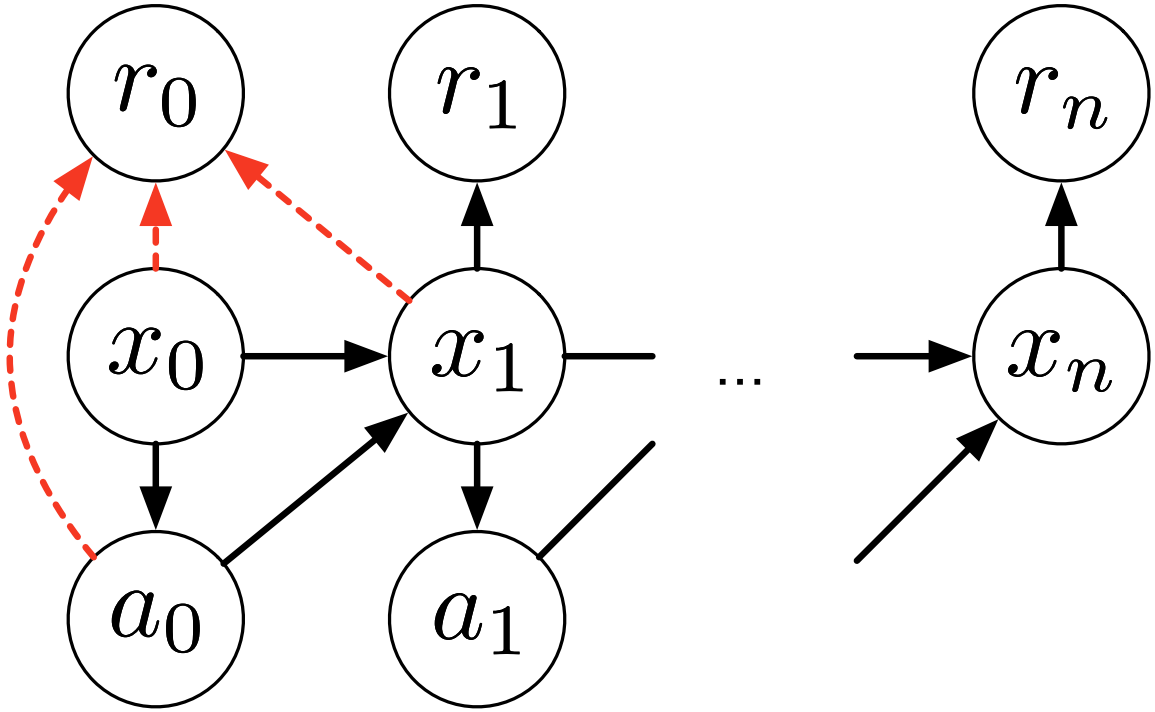
Let $\pi(a|x)$ be a *policy* defining the probability of taking action a from state x.

In this lecture we will assume π is given and try to compute its value. In later lectures we will consider modifying or optimizing π.

A graphical model for MDPs



A graphical model for MDPs



Alternatively the reward can be written as $r(x_n, a_n, x_{n+1})$ or some subset of these variables. These can all be made equivalent WLOG by modifying the state-space.

Markov reward processes

By integrating out the actions we can write

$$p^{\pi}(z|x) = \int_{a \in \mathcal{A}} p(z|x, a)\, \pi(a|x)\, da$$

which defines a *Markov process* over the space $\mathcal{X}$.

This transition model combined with the reward function defines a *Markov reward process* (MRP) which transitions between states with this probability and spits out rewards.

For a few slides we will ignore the dependency on π and just consider $p(z|x)$

Using the Markov process

Let $\mu(x)$ be some distribution over $\mathcal{X}$ often called an *initial state distribution*.

The formulation as a Markov process allows us to compute the distribution over n steps by chaining integrals:

$$
\begin{aligned}
p_1(x_1) &= \int p(x_1|x_0)\,\mu(x_0)\,dx_0 \\
p_n(x_n) &= \int p(x_n|x_{n-1})\Big(\int \cdots \Big(\int p(x_1|x_0)\,\mu(x_0)\,dx_0\Big)\cdots\Big)dx_{n-1} \\
&= \int p(x_n|x_{n-1})\,p_{n-1}(x_{n-1})\,dx_{n-1} \quad \text{where } p_0(x_0) = \mu(x_0).
\end{aligned}
$$

Discrete state-spaces

For discrete state-spaces the previous integrals simplify. Let $\mathbf{P} = [p_{ij}]$ be a matrix such that p_{ij} is the probability of moving to state i from state j and $\boldsymbol{\mu} = [\mu_i]$ be some initial-state vector.

The previous integral can then be written as

$$[\mathbf{p}_1]_i = \sum_j p_{ij} \mu_j$$

which can be summarized as

$$\mathbf{p}_t = \mathbf{P}\mathbf{p}_{t-1}.$$

What does $\mathbf{P}^\pi$ look like given arrays $[P_{iaj}]$ and $[\pi_{aj}]$?

Value of an MDP

Now that we have a handle on the distribution of a Markov process over time, we can use it to compute the expected value of the process

Reintroducing π let $p_t^\pi(x_t)$ be the t-step distribution given some policy π.

$$J(\pi) = \sum_{t \leqslant T} \mathbb{E}[r(x_t)] = \sum_{t \leqslant T} \int p_t^\pi(x_t)\, r(x_t)\, dt$$

Note, though, that this depends on having an initial distribution $p_0(x)$.

Infinite horizon

In order to extend the value computations to infinite horizons we need some way to average over an infinite number of rewards:

$$J(\pi) := \lim_{T \to \infty} \frac{1}{T} \sum_{t \leqslant T} \mathbb{E}[r(x_t)]$$

or

$$J(\pi) := \sum_{t \leqslant \infty} \mathbb{E}[\gamma^t \, r(x_t)]$$

The second can also be thought of as a Geometric probability of "the world ending".

Policy search

There is a reason why we wrote our value function as $J(\pi)$—we can use this iteratively to define a sequence of policies π_i such that $J(\pi_i) \geqslant J(\pi_{i-1})$. This is known as policy search.

If the policy π_θ is indirectly defined by some differentiable parameters θ then

$$\theta_i = \theta_{i-1} + \alpha_i \nabla_\theta J(\theta_{i-1})$$

can be used—this uses the policy gradient.

Value functions

Rather than directly parameterizing the policy and computing its value we can write the value of the policy if we start in a specific state:

$$
\begin{aligned}
V^{\pi}(x) &= \mathbb{E}\Big[\sum_{t \geqslant 0} \gamma^t r(x_t) \Big| x_0 = x\Big] \\
&= r(x) + \mathbb{E}\Big[\sum_{t \geqslant 1} \gamma^t r(x_t) \Big| x_0 = x\Big] \\
&= r(x) + \gamma \mathbb{E}\big[V^{\pi}(x_1) \big| x_0 = x\big]
\end{aligned}
$$

Discrete value functions

Let's use again a discrete transition model $\mathbf{P}^\pi$ for which both the reward and value function are vectors $\mathbf{r}, \boldsymbol{v}^\pi \in \mathbb{R}^m$. The previous equation can be written as

$$\boldsymbol{v}^\pi = \mathrm{r} + \gamma \mathbf{P}^\pi \boldsymbol{v}^\pi$$
$$(\mathbf{I} - \gamma \mathbf{P}^\pi)\boldsymbol{v}^\pi = \mathrm{r}$$

or an iterative procedure can be used to find

$$\boldsymbol{v}^{(\mathrm{i})} = \mathrm{r} + \gamma \mathbf{P}^\pi \boldsymbol{v}^{(\mathrm{i}-1)}$$

Model-based versus model-free Reinforcement Learning

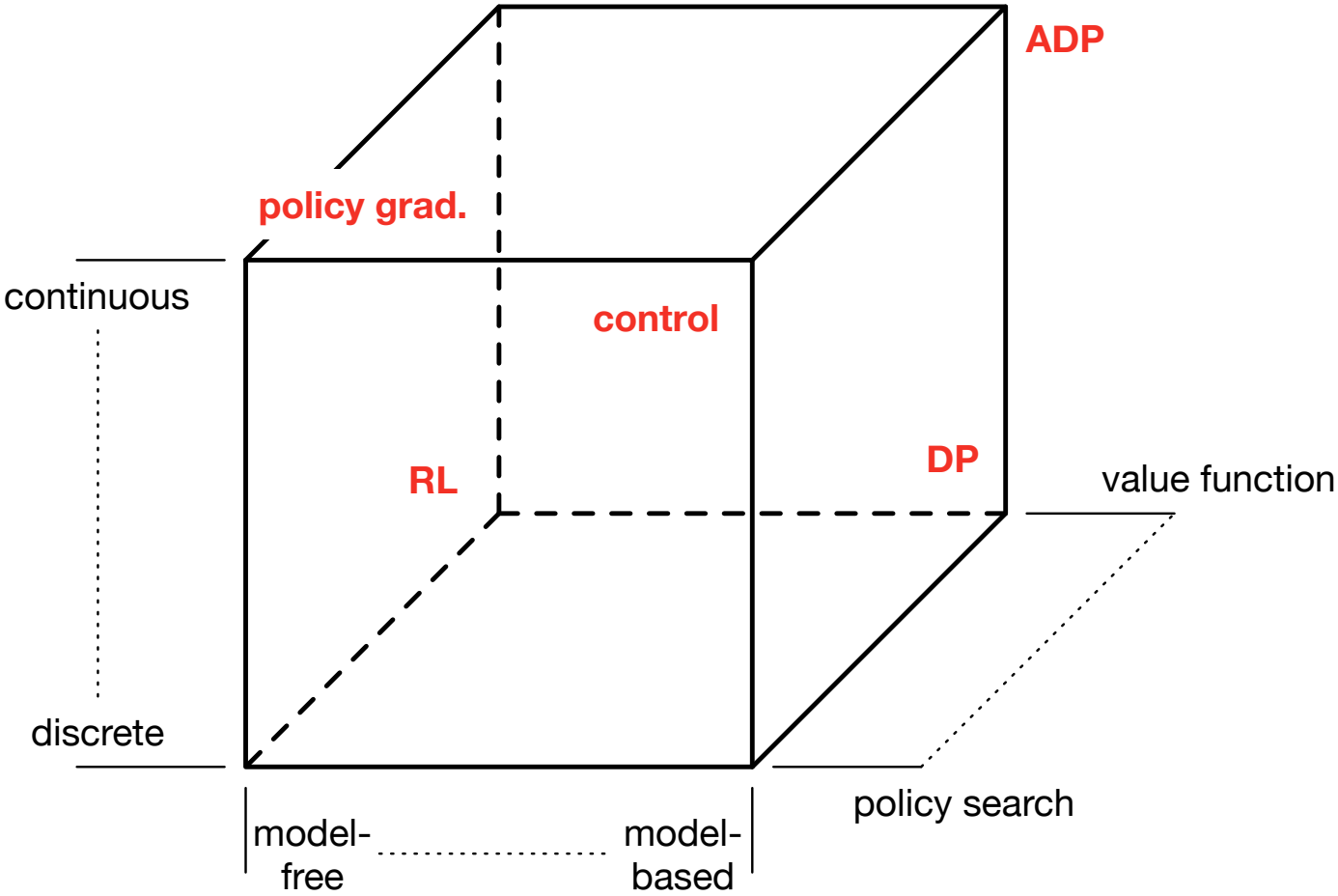
The previous slides have assumed the ability to integrate over the distribution $p_t^\pi(x_t)$ which itself requires knowledge of $\mu(x_0)$ and $p(z|x, a)$ to construct.

However what if we do not have access to these distributions? Typically reinforcement learning assumes we can only sample from these.

This distinction is often also known as model-based versus model-free RL.

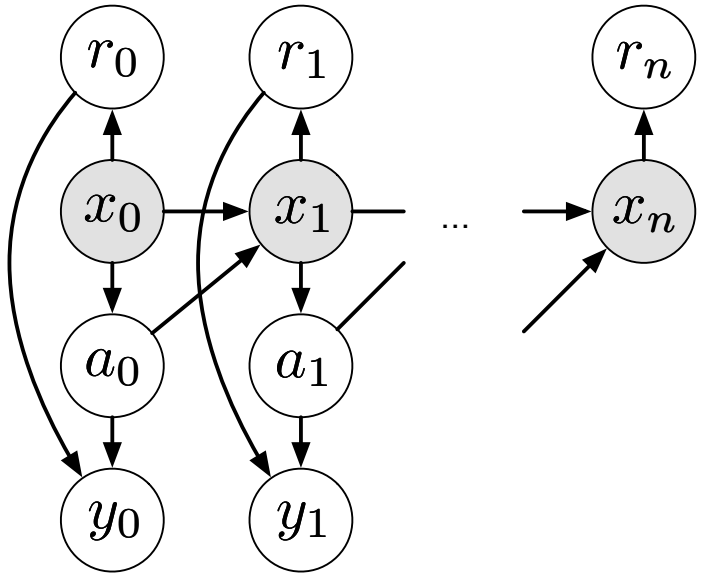
In the simplest case we can compute the value of a policy as

$$J(\pi) = \frac{1}{N} \sum_{i \leqslant N} \sum_{t \leqslant T} r(x_t^{(i)}) \quad \text{for } x_{0:T}^{(i)} \sim p^\pi(x_{0:T})$$


ADP
policy grad.
continuous
control
RL
DP
value function
discrete
policy search
model-
free
model-
based

Partially-observable MDPs (POMDPs)

The agent does not observe the full state of the environment. What is the optimal policy?



- If the agent has the correct model of the world, it turns out that the optimal policy is a (piece-wise linear) function of the **belief state**, $P(x_t|a_1, \ldots, a_{t-1}, r_1, \ldots, r_t, y_1, \ldots, y_t)$. Unfortunately, the belief state can grow exponentially complex.
- Equivalently, we can view the optimal policy as being a function of the entire sequence of past actions and observations (this is the usual way the policy in influence diagrams is represented). Unfortunately, the set of possible such sequences grows exponentially.

Efficient methods for approximately solving POMDPs is an active research area.