Reinforcement Learning (INF11010)

Lecture 2: Introduction to Markov Decision Processes

Pavlos Andreadis, January 19th 2018

Today’s Content

- *(*discrete-time*) finite* Markov Decision Process (MDPs)
  - State space; Action space; Transition function; Reward function.
  - Policy; Value function.

- Markov property/assumption

- MDPs with set policy → Markov chain

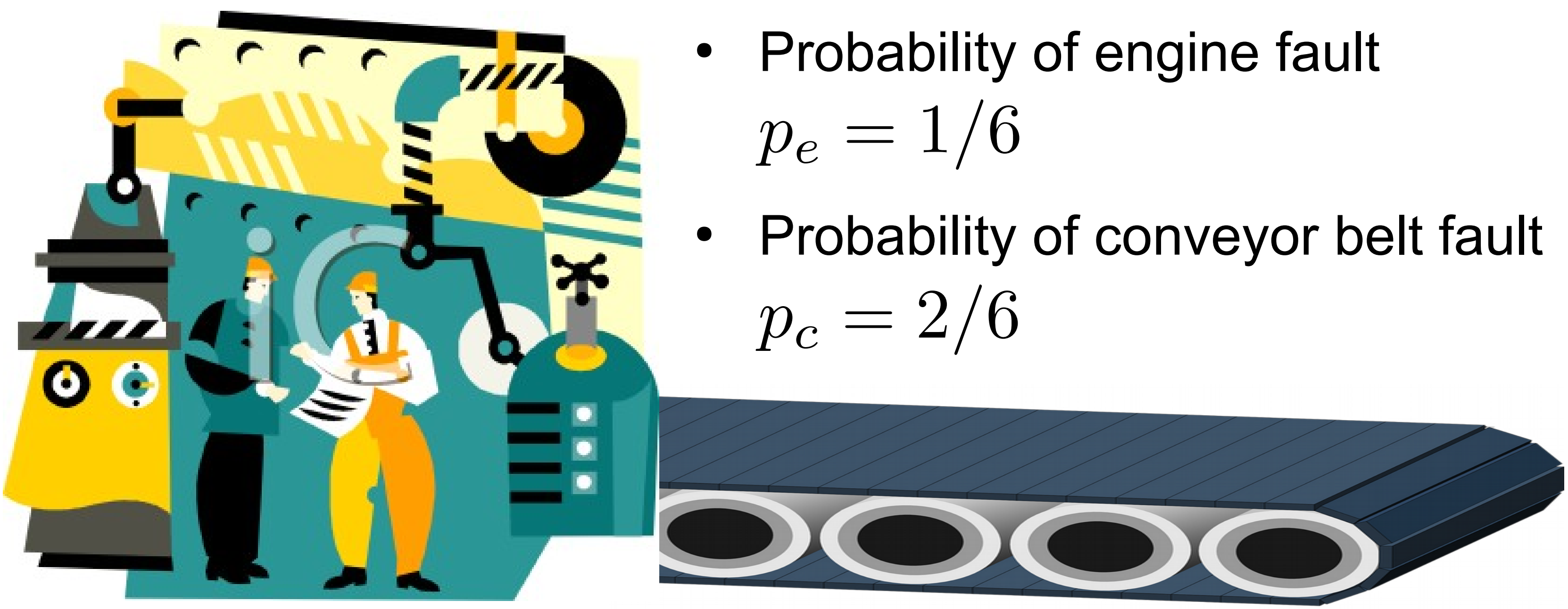
- The Reinforcement Learning problem:
  - Maximise the accumulation of rewards across time

- Modelling a problem as an MDP (example)

a Repair Scenario

- Output in 1000s of $:
  - Good: 5 - No conveyor belt: 3 - No production: 0

- Cost of repairs (regardless of condition) in 1000s of $: 10



- Probability of engine fault
  $p_e = 1/6$
- Probability of conveyor belt fault
  $p_c = 2/6$

State & Action spaces

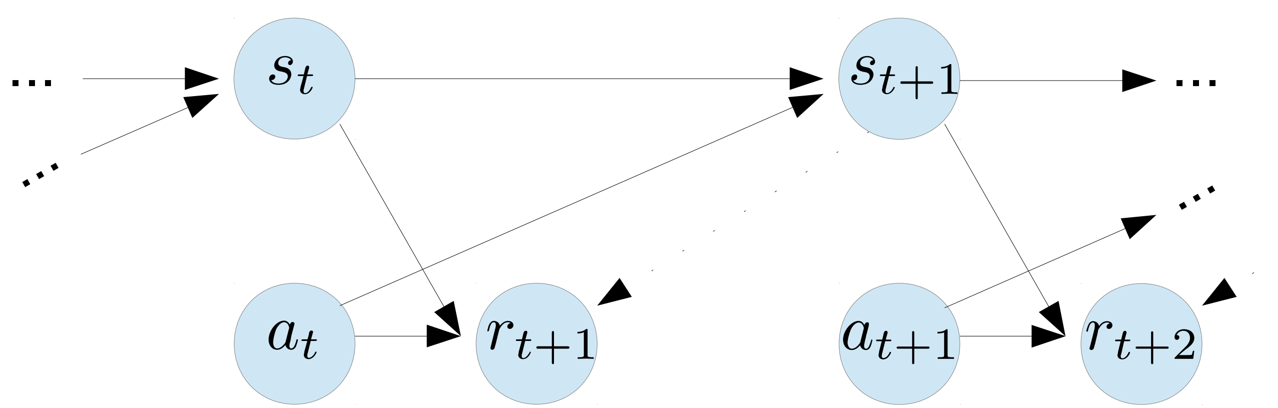
$$S = \{s_0,\ s_1,\ s_2\}$$

- s_0 No problems
- s_1 Conveyor belt fault
- s_2 Engine fault

$$\mathrm{A} = \{a_0\,, a_1\}$$

- a_0 wait
- a_1 repair

- the MDP model as a *Dynamic* Bayesian Network (i.e. a *dynamic* probabilistic directed acyclic graph):



- Markov property!

Reward & Transition Functions

- The Reward function: $R : S, A, \boxed{S} \rightarrow \mathbb{R}$ $R^a_{s,s'}$

	a_0	a_1
s_0	5	-5
s_1	3	-7
s_2	0	-10

- The Transition function: $P : S, A, S \rightarrow [0, 1]$ $P^a_{s,s'}$

wait	s_0	s_1	s_2
s_0	$\tilde{p}_e \cdot \tilde{p}_c$	$\tilde{p}_e \cdot p_c$	p_e
s_1	0	$\tilde{p}_e$	p_e
s_2	0	0	1

repair	s_0	s_1	s_2
s_0	1	0	0
s_1	1	0	0
s_2	1	0	0

Markov Property

- Environment response, Generally:

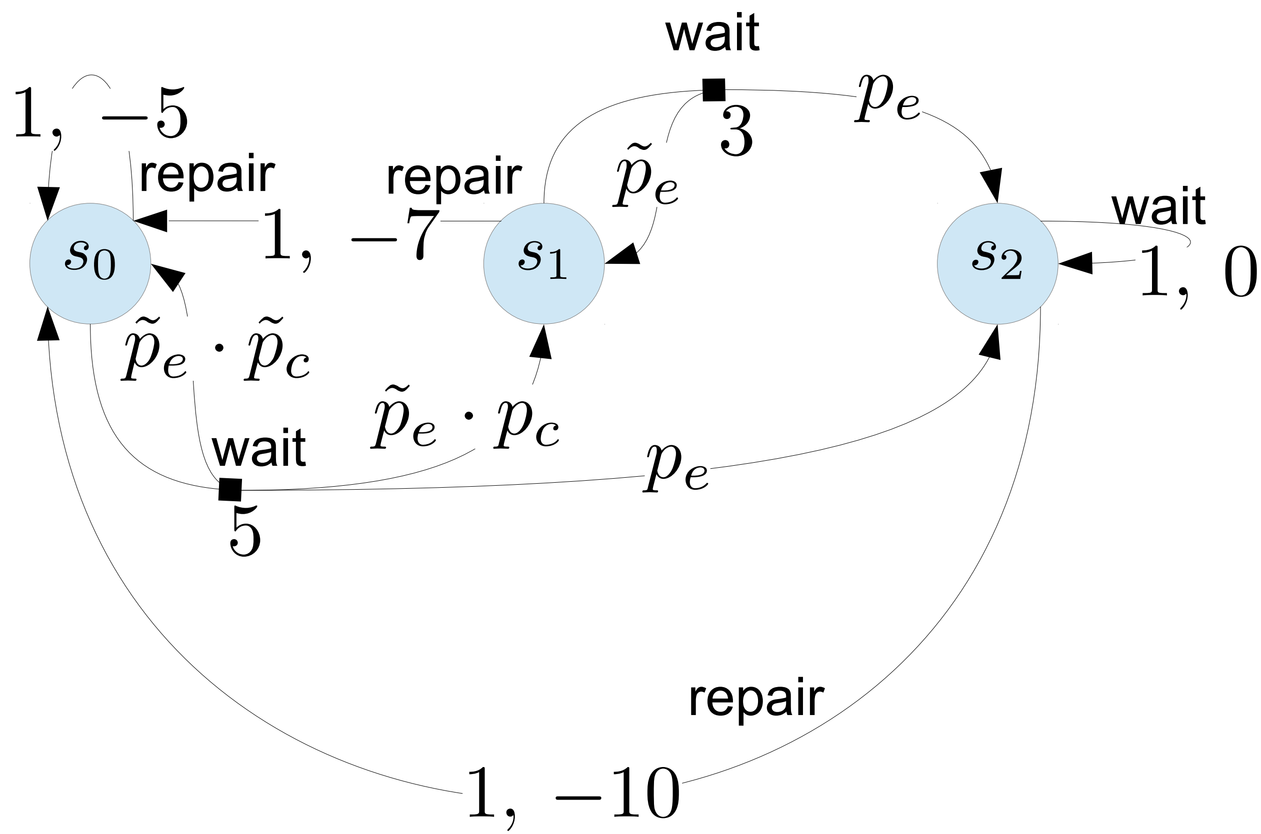
$$Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t, r_t, s_{t-1}, a_{t-1}, ..., r_1, s_0, a_0\}$$

- … with the Markov property:

$$Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t\}$$

Transition Graph

- the *Transition Graph* for our MDP model for the Repair Scenario:



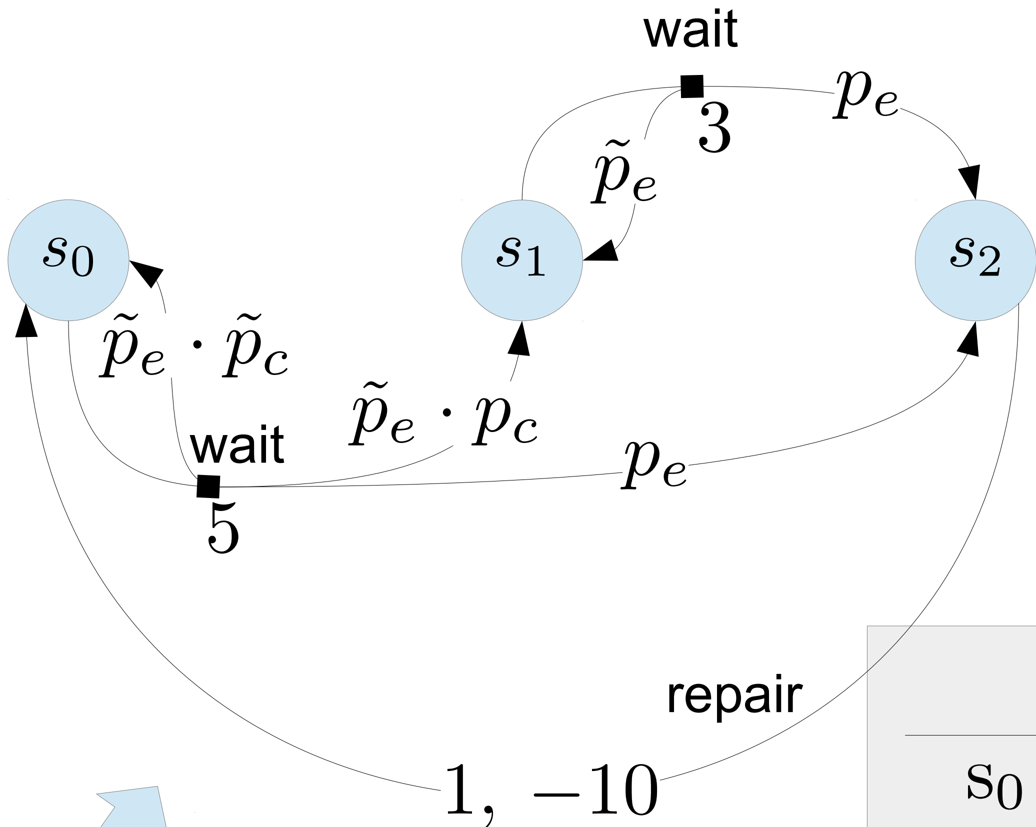
Policy

- A policy π is a mapping from each state $s \in S$ and action $a \in A$ to a probability $\pi(s, a)$

    - For example:

	a_0	a_1
s_0	0.6	0.4
s_1	1	0
s_2	0.3	0.7

a Deterministic Policy



- A Markov Chain

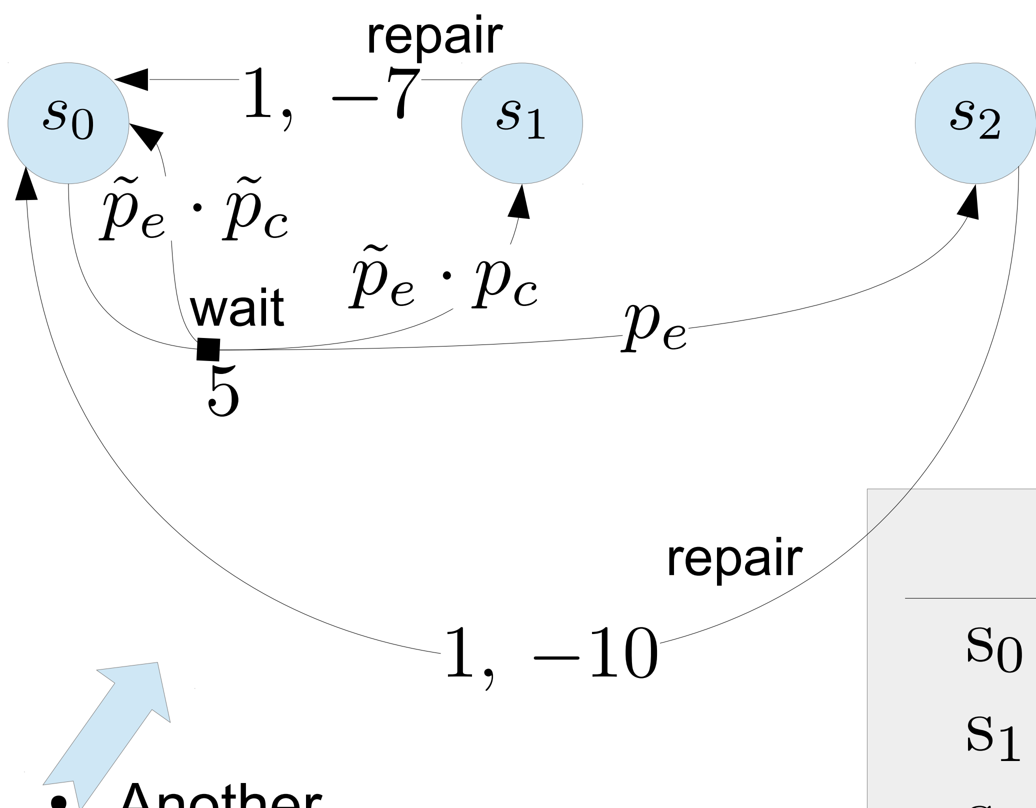
- "Wait till it breaks" policy:

s_0	wait
s_1	wait
s_2	repair

- Stochastic/Transition matrix:

	s_0	s_1	s_2
s_0	$\tilde{p}_e \cdot \tilde{p}_c$	$\tilde{p}_e \cdot p_c$	p_e
s_1	0	$\tilde{p}_e$	p_e
s_2	1	0	0

another Deterministic Policy



- Another Markov Chain

- "Repair" policy:

s_0	wait
s_1	repair
s_2	repair

- Stochastic/Transition matrix:

	s_0	s_1	s_2
s_0	$\tilde{p}_e \cdot \tilde{p}_c$	$\tilde{p}_e \cdot p_c$	p_e
s_1	1	0	0
s_2	1	0	0

Returns (finite time)

- Return at time t = the reward accumulated starting from the next time step:

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + ... + r_T$$

- T = a final time step
- *Episodic* tasks, i.e. there is a final time step
- Each episode ends in a *terminal* (absorbing) state

- Assuming we are at time t our goal is to maximise the *expected* return at t

Returns (infinite time)

- *Discounted* Return at time t

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

- γ = *discount rate* $0 \leq \gamma \leq 1$ (prevents a sum to infinity / weights reward across time)
- *Continuing* tasks, i.e. there is *no* final time step
- A single neverending episode

- Assuming we are at time t our goal is to maximise the *expected discounted* return at t

Returns (unified notation)

- *Discounted* Return at time t

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1}$$

- *Continuing* tasks by setting $T = \infty$
- In which case we can't have both $T = \infty$ and $\gamma = 1$

OR

- Define *absorbing* states as transitioning to themselves with a reward of 0

Value Function

- We can define the *value* of a state s *under policy* π using the *state-value function*:

$$V^{\pi}(s) = E_{\pi}\{R_t|s_t = s\} = E_{\pi}\Big\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s\Big\}$$

- … or the *action-value* (or Q-) *function*:

$$\begin{aligned} Q^{\pi}(s,a) &= E_{\pi}\{R_t|s_t = s, a_t = a\} \\ &= E_{\pi}\Big\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s, a_t = t\Big\} \end{aligned}$$

Bellman Equation

$$
\begin{aligned}
V^\pi(s) &= E_\pi\{R_t | s_t = s\} \\
&= E_\pi\Big\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\Big\} \\
&= E_\pi\Big\{r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_t = s\Big\} \\
&= \sum_a \pi(s,a) \sum_{s'} P^a_{s,s'} \Big[R^a_{s,s'} + \gamma\, E_\pi\Big\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_{t+1} = s'\Big\}\Big] \\
&= \sum_a \pi(s,a) \sum_{s'} P^a_{s,s'} [R^a_{s,s'} + \gamma\, V_\pi(s')]
\end{aligned}
$$

Optimal Value Function

$$V^*(s) = \max_\pi V_\pi(s), \text{ for all } s \in S$$

$$Q^*(s,a) = \max_\pi Q_\pi(s,a), \text{ for all } s \in S \text{ and } a \in A$$

Markov Decision Processes

- A finite Markov Decision Process (MDP) is a tuple (S, A, P, R, γ) where:

- S is a finite set of states
- A is a finite set of actions
- P is a state transition probability function
- R is a reward function
- γ is a discount factor

Reading +

- Chapter 3 of Sutton and Barto (1^{st} Edition)
  http://incompleteideas.net/book/ebook/the-book.html

- Please join Piazza for announcements and support:
  https://piazza.com/ed.ac.uk/spring2018/infr11010

Optional:

- *Excercise*: pick a policy for the Repair Scenario, and write a procedure in Matlab that evaluates the Expected Return from s_0. (feel free to use Piazza to ask for tips)

a Repair Scenario