
Introduction

Herke van Hoof

Studying together safely (1)

- The 1.5 standard has been dropped in the UvA buildings;
- Wearing a mouth mask is compulsory when moving around in the building;
- It is preferable to wear a mouth mask during lectures; many students and lecturers feel more comfortable doing so;
- Follow the designated walking routes when moving around;
- Observe the rules regarding max. 75 students in a room;
- Observe the hygiene rules;
- Clean your study area beforehand and tidy up afterwards (put all rubbish in the waste bin);
- Do not stay on campus unnecessarily: go home immediately after a lecture if you do not have a follow-up lecture or other commitments;

Studying together safely (2)

- Take a self-test at least twice a week. The tests can be ordered free of charge at zelftestonderwijs.nl;
- If you have symptoms, stay at home and take a test;
- If any symptoms develop during your stay on campus, go home and take a test;
- We appreciate it if you report a positive (self-)test to your teacher. It is not necessary, though;
- The UvA encourages you to get vaccinated. You can make an appointment via www.coronatest.nl or prikkenzonderafspraak.rijksoverheid.nl. Or visit the GGD vaccination location on the Roeterseiland campus. You don't need to make an appointment! If you have any questions or doubts about this, we are open to help you with your choice.

About reinforcement learning

How to **sequentially interact**
with an environment to
maximise a long-term objective?

About reinforcement learning

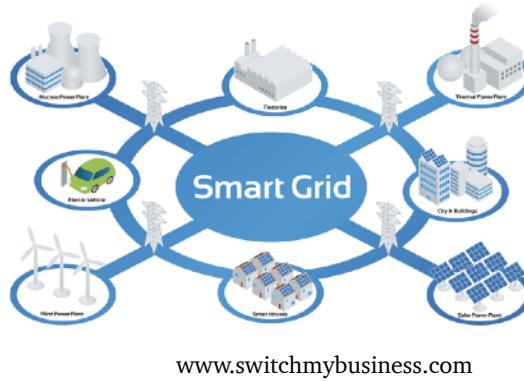
How to **sequentially interact**
with an environment to
maximise a long-term objective?

Why is this important?

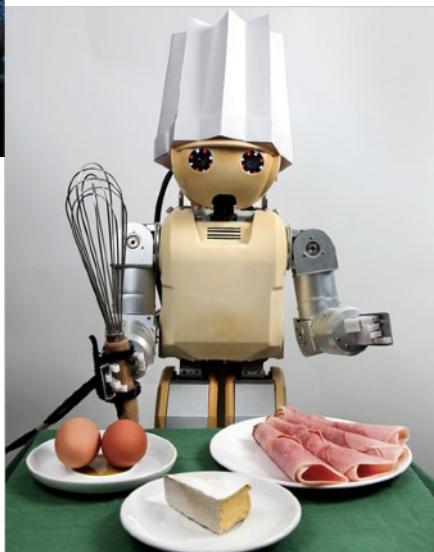
Reinforcement learning

Can you think of some scenarios where sequential interaction is important?

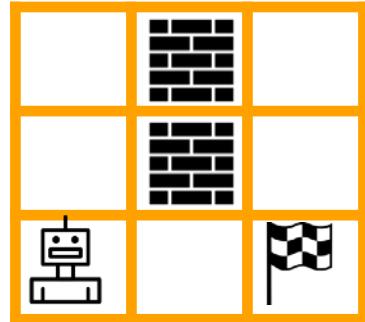
About reinforcement learning



robots.nu



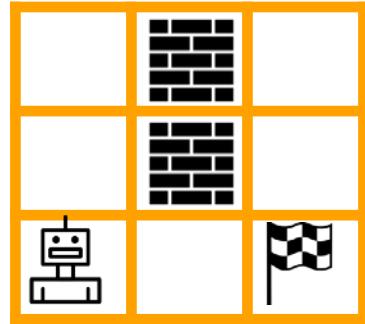
About reinforcement learning



state

'chosen' by environment

About reinforcement learning

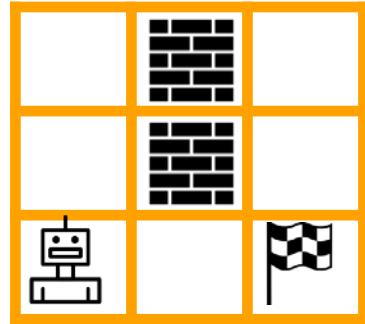


state

action

'chosen' by environment chosen by agent

About reinforcement learning



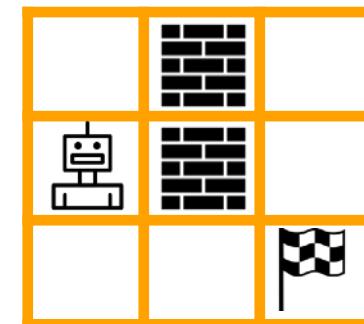
state
'chosen' by environment



action
chosen by agent



reward
'chosen' by environment



next state

About reinforcement learning



About reinforcement learning

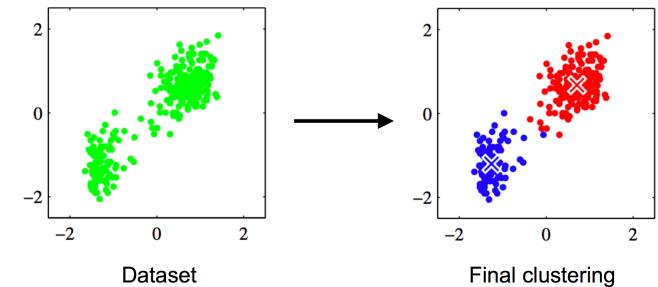
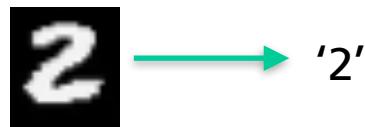
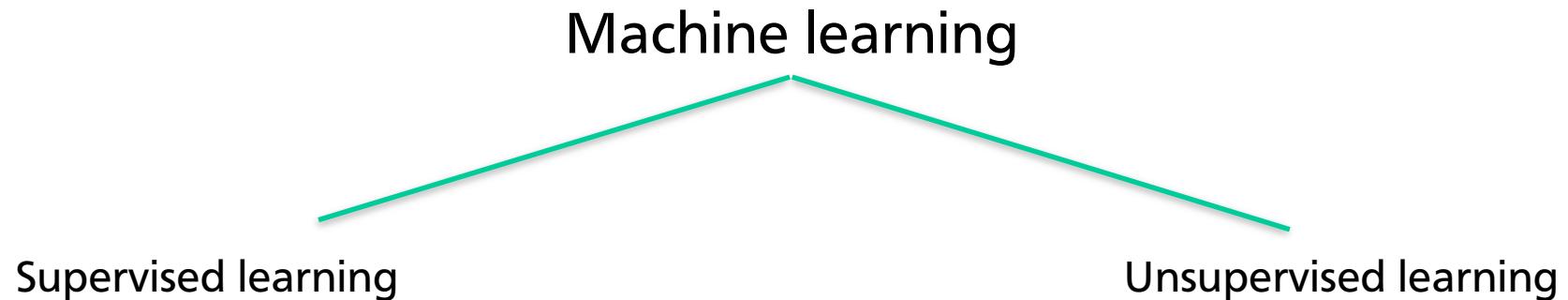
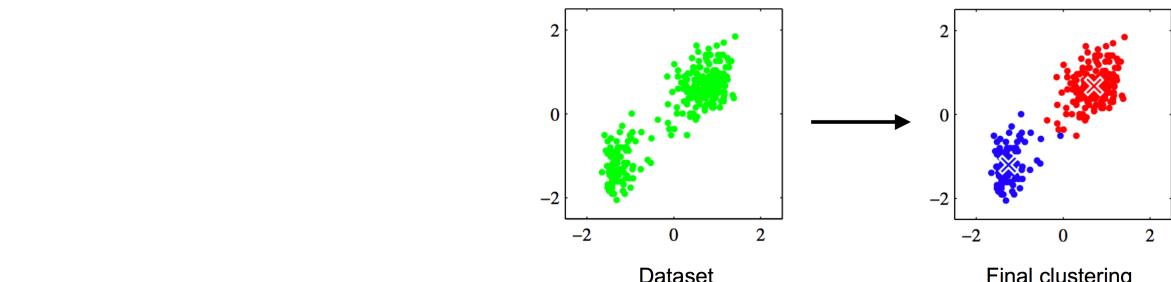
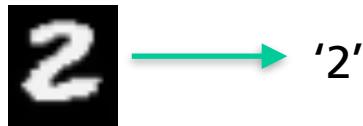
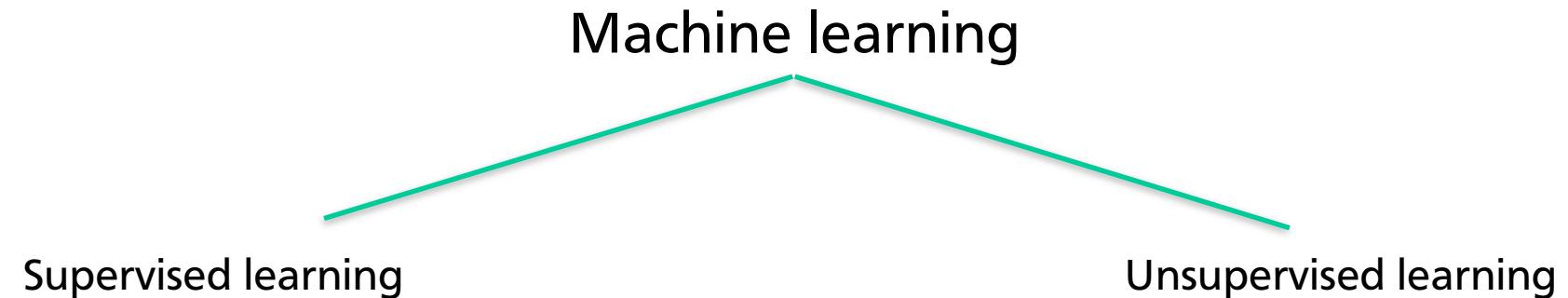


Figure: C.M. Bishop, Pattern recognition & Machine learning

About reinforcement learning



Reinforcement learning
?

Figure: C.M. Bishop, Pattern recognition & Machine learning

Learning to do vs learning to predict

Supervised learning



(Deng et al., 2009)

Learn to predict

Dataset with human response

Data doesn't depend on what
is learned

Data are iid

Learning to do vs learning to predict

Supervised learning



(Deng et al., 2009)

Learn to predict

Dataset with human response

Data doesn't depend on what
is learned

Data are iid

Reinforcement learning



robotshop.com



Learn to do

Improve over human response

Data depends on current
strategy

Data are sequential

About the course

Before we dive into any details, let's look at what the course is going to be like!

Hybrid course

Lectures:

- On-campus for max 75 students.
Sign-up via Canvas obligatory (see Canvas announcement).
- Live stream and on-demand recording.
First time to live stream, hope all works well...

Tutorial sessions

- Fully on campus

Exam

- Unclear. Hopefully in-person.

All subject to rule changes...

Tutorials & Exercises

Practice is essential for learning!

- *ungraded exercises* to practice the material
- Some questions from last years' exams, shows level & type of questions
- Roughly, 1st hour tutorial: work on ungraded exercises together
- Every week, written homework & programming assignment(s)
- Roughly, 2nd hour tutorial: work on assignments, can ask questions

How to do well in the course

Watch the lecture and review as needed before tutorial session

Actively participate in tutorial classes. You can benefit from interaction with TAs and other students

Don't skip the ungraded exercises!

Reading material offers additional insights and different perspectives, suggested to read alongside lectures.

Lecture

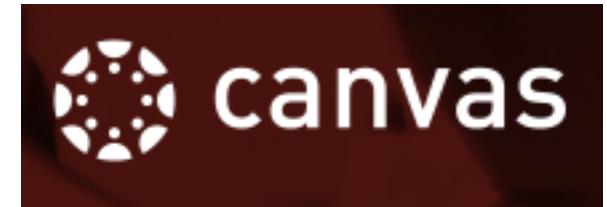
Review
(as needed)

Tutorial

Finalize exercises
& homework

About the course

Recordings, slides, and assignments will be posted on Canvas!



Important announcements will be sent from Canvas as well - make sure they reach you!

Also, you'll find the syllabus with:

- Reading suggestions per week
- Deadlines
- Organisational information

About the course

| Week | Topic | |
|------|---------------------------------|----------------------------------|
| 1 | Recap RL basics | |
| 2 | Tabular methods | HW 1 due fri 13:00 |
| 3 | Learning with approximation | HW 2 due fri 13:00 |
| 4 | Direct policy learning | HW 3 due fri 13:00 |
| 5 | Advanced direct policy learning | HW 4 due fri 13:00 |
| 6 | Planning, guest lecture | |
| 7 | Partial observability, recap | peer feedback assignment due! |
| 8 | Exam! | |

About the course

| Week | Topic | |
|------|---------------------------------|-----------------|
| 1 | Recap RL basics | RL basics |
| 2 | Tabular methods | |
| 3 | Learning with approximation | |
| 4 | Direct policy learning | |
| 5 | Advanced direct policy learning | Advanced topics |
| 6 | Planning, guest lecture | |
| 7 | Partial observability, recap | |
| 8 | Exam! | |

Prior knowledge

Working knowledge of machine learning: least-squares, neural networks, gradient descents

The basics: calculus, linear algebra, probability, statistics, programming (Python)

Workload

Total: 168 hours (6 credits x 28 hours): 21 hrs/week

Lectures: 28 hours

Tutorial session: 28 hours

Reading (lecture and exam prep): 48 hrs (avg. 6 hrs/week)

Assignments: 71 hours (avg 9 hrs/week + class time)

Assessment

5 coding assignments. 2% each

- Suggested in pairs (individual possible)
- Hand-in online as instructed. Use feedback from codegra.de!
- No extensions will be given

4 exercise sets. 4% each

- Suggested in pairs (individual possible)
- Hand-in online on Canvas.
- No extensions will be given

Reproducibility lab. 14%

- Groups of 3-4
- Peer feedback session counts toward grade (Oct. 18th)
- Starts in 5th week. Instructions will follow

Exam (exam format tbd depending on regulations). 60%

What can you expect of us?

We are happy to receive any feedback

- Just get in touch with me via e-mail, Canvas, Piazza, or talk to me in the break!

In previous years, the workload was experienced to be on the high side

- We've tried to somewhat reduce the workload of the exercises
- Expected workload: ca. 21 hrs/wk
- If you find yourself doing significantly more, please let us know (most helpful to us is if you can indicate where you ended up spending more time)
- I'll do a survey during the course on this topic as well

What can you expect of the course?

RL is a complex: Even basics require quite a bit of nuance

Basics essential for understanding new papers

We'll touch on state of the art, but focus will be on solid basis

I'll illustrate underlying principles through algorithms

Thus, we'll cover quite many algorithms. Not all are practical, but they help understand fundamentals.

Important part: understand the principles that distinguish them.

Study materials

Main resource:

Reinforcement learning:

An Introduction (RL:AI)

R. S. Sutton & A.G. Barto

We use 2nd edition, it is also available for free online
(link on canvas)

(Most topics are also described in the old book)



Study materials

Our discussion of policy based methods will be based on the survey by Deisenroth, Neumann & Peters (link on Canvas)

A Survey on Policy Search for Robotics

By Marc Peter Deisenroth,
Gerhard Neumann and Jan Peters

We will also look at additional papers (links will be distributed on Canvas)

The additional papers help understand the lecture content, but details not covered in the lecture will not be on the exam.

Study materials

Slides will be available as soon as possible after the lecture

Slides are not intended to be self-contained study material, but to support the lecture. For self-study the recommended reading from the book is more appropriate

Exam will cover all material (lectures, reading material, assignments)

Questions about course organization?

About us

Instructor: Herke van Hoof

- Assistant professor in the Amsterdam machine learning lab (AMLab)
- Background: machine learning for robots
- Research focus:
 - Policy-based methods
 - Learning for robots and dynamical systems
 - Reinforcement learning with small data sets

About us



Jin Huang

PhD student at ILPS

Research focus: RL for recommendation



Niklas Höpner

PhD student at AMLAB

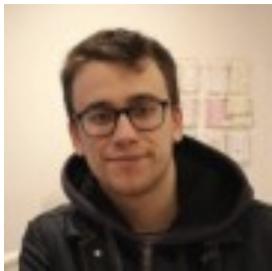
Research focus: Symbolic knowledge & RL



David Kuric

PhD student at AMLAB

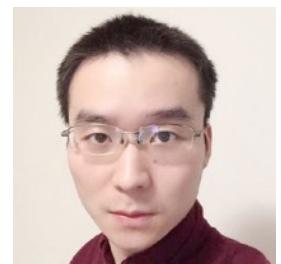
Research Focus: Meta reinforcement learning



Matthew Macfarlane

PhD student at AMLAB

Research focus: RL for combinatorial optimisation



Qi Wang

PhD student at AMLAB

Research focus: Model-based RL

Groups

Group assignments should avoid conflicts as much as possible

In principle, we'll only **switch** people between groups

- Can post a request in Canvas discussion forum
- Send Herke an e-mail with the names of the two people (with e-mail addresses and student IDs) that want to switch.

Contacting us

Questions about lecture content:

- Piazza
- During lecture or break

Questions regarding exercises & practicals, (inc. grading) should be send to the **TA of your group**

Feedback on lecture content

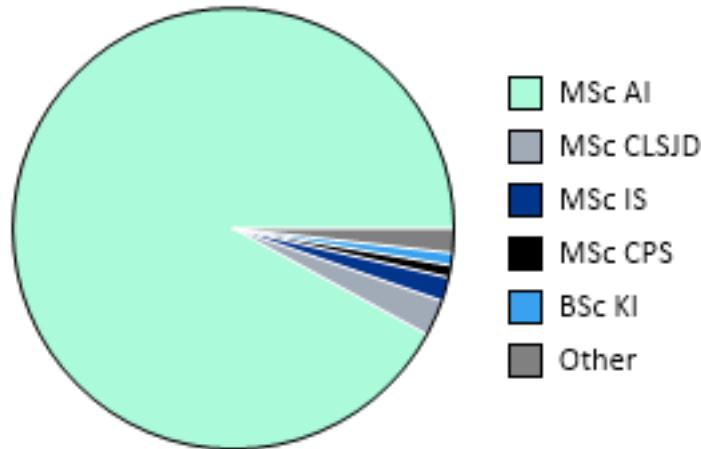
- Canvas message or email h.c.vanhoof@uva.nl

Confidential matters or when an issue cannot be resolved by TA

- Canvas message or email h.c.vanhoof@uva.nl

We try to answer all questions soon, but it might take us a day or two to reply

About you



Which BSc?
Where BSc?
Prior RL knowledge? (MAS?)



If you encounter any problems during programme related event or a course and you want to file a complaint or submit separate feedback, please do not hesitate to contact the programme committee at:
ocai-science@uva.nl

For more information, please see [this](#) page.

Course evaluation last year

Mostly, last year's evaluation was positive
(though only 30% filled it out)

Some attention points:

- Second half of the course is less structured than the first
Partly, this is hard to avoid as it dives into more advanced topics, but I'll try to give it more structure
 - Workload is on the high side
*We revised exercises to reduce workload and spread it better.
We'd love to hear which parts are taking up most time, if problem persists*
 - Tutorial sessions could be structured better
We'll try a clear model of 50% working together, 50% working by yourself. Let us know whether it works for you
-

Models of decision processes

k-armed bandits

- a simple model for decision making

Markov decision processes

- a model for sequential decision processes

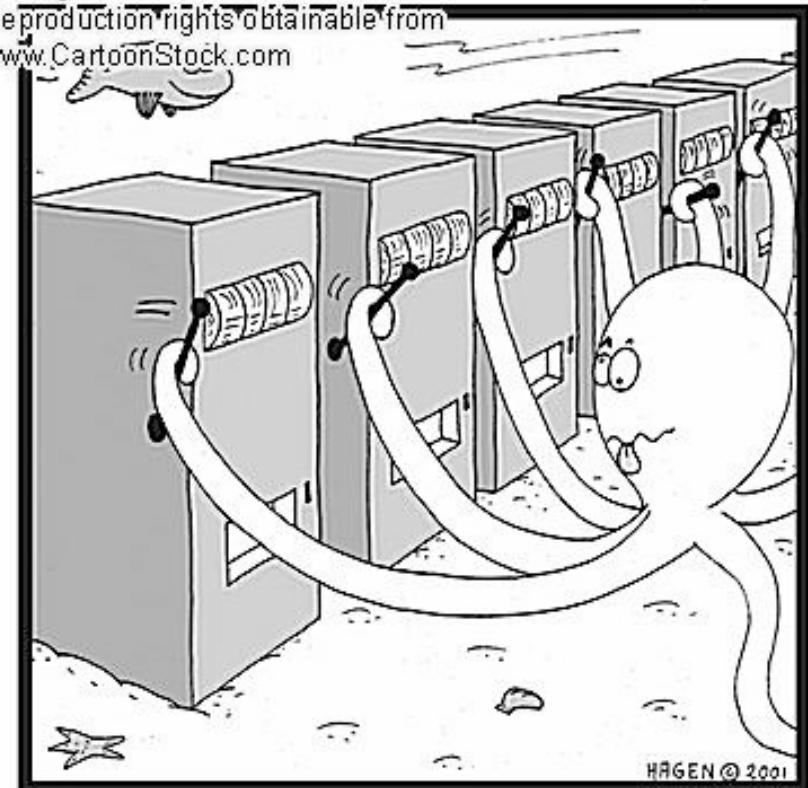
k-armed bandits

There are k slot machines to choose from

Each machine has an unknown distribution of payoffs

Goal: maximize cumulative payoff over some period

© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com



Thanks to Shimon Whiteson

k-armed bandits

Formally:

- There are k actions available at any time step
- After action a_t , receive reward r_t from unknown distribution $p(r_t | a_t)$
- Two types of objectives:

Finite horizon,
maximise total reward over T actions

$$\sum_{t=1}^T r_t$$

Infinite horizon,
maximise total discounted rewards
with discount factor $\gamma \in [0, 1)$

$$\sum_{t=1}^{\infty} \gamma^t r_t$$

k-armed bandits

Formally:

- There are k actions available at any time step
- After action a_t , receive reward r_t from unknown distribution $p(r_t | a_t)$
- Two types of objectives:

Finite horizon,
maximise total reward over T actions

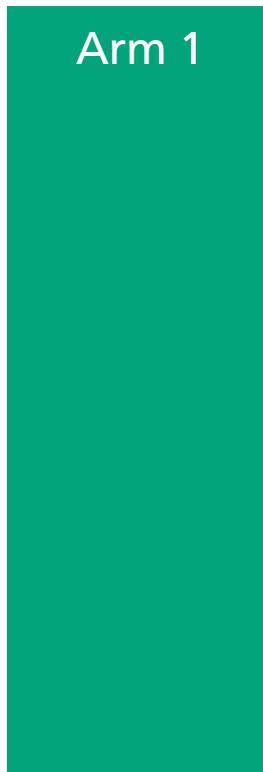
$$\sum_{t=1}^T r_t$$

Infinite horizon,
maximise total discounted rewards
with discount factor $\gamma \in [0, 1)$

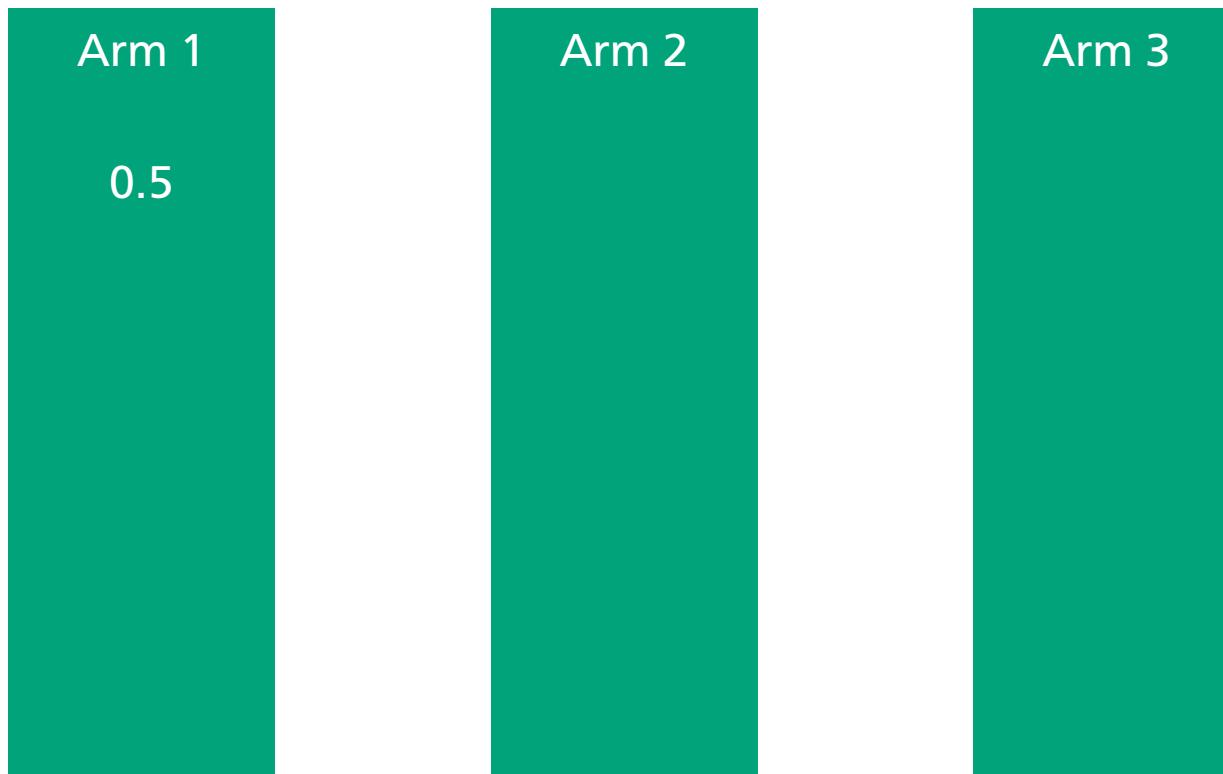
$$\sum_{t=1}^{\infty} \gamma^t r_t$$

What happens if $\gamma=1$ with an infinite horizon?

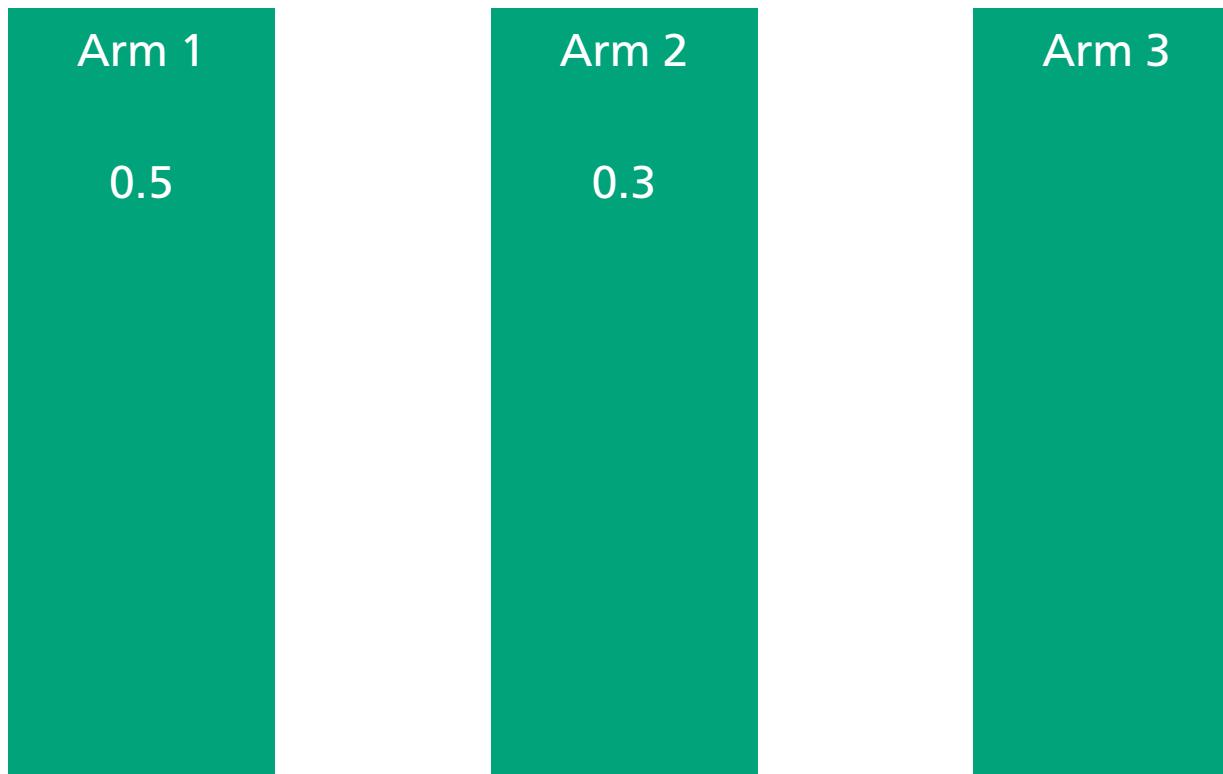
Example



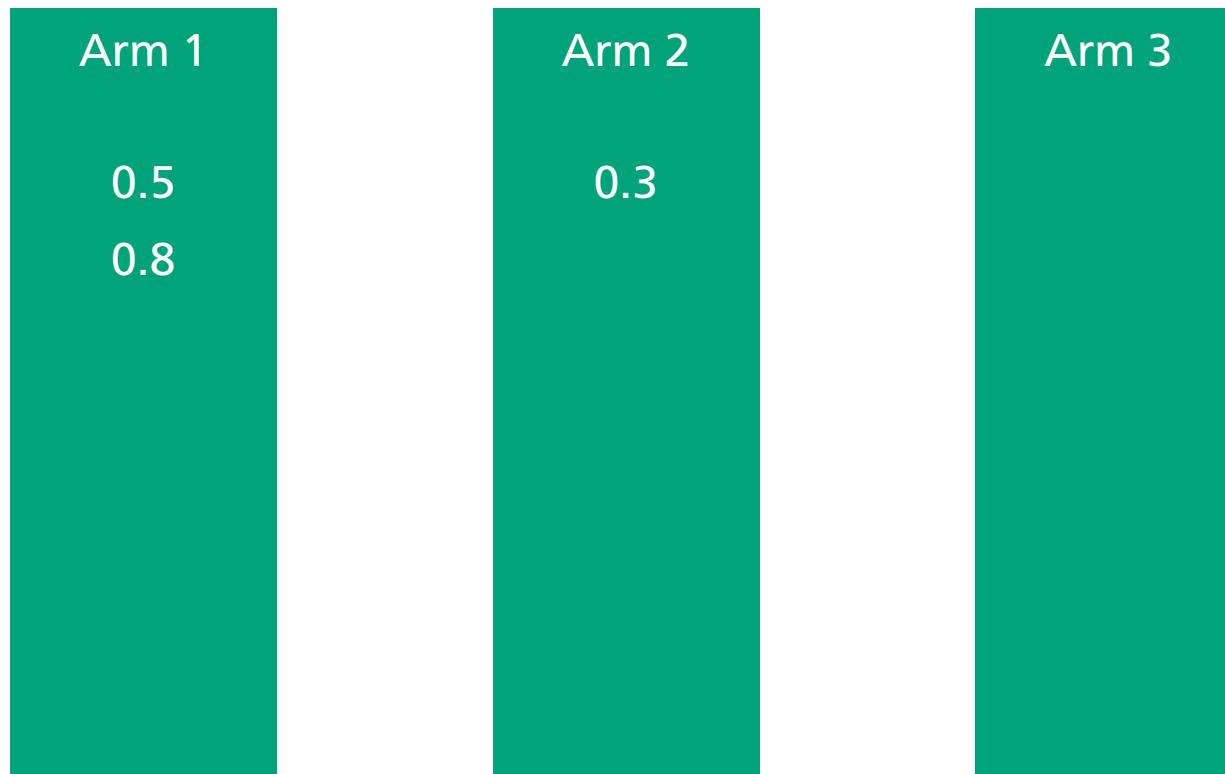
Example



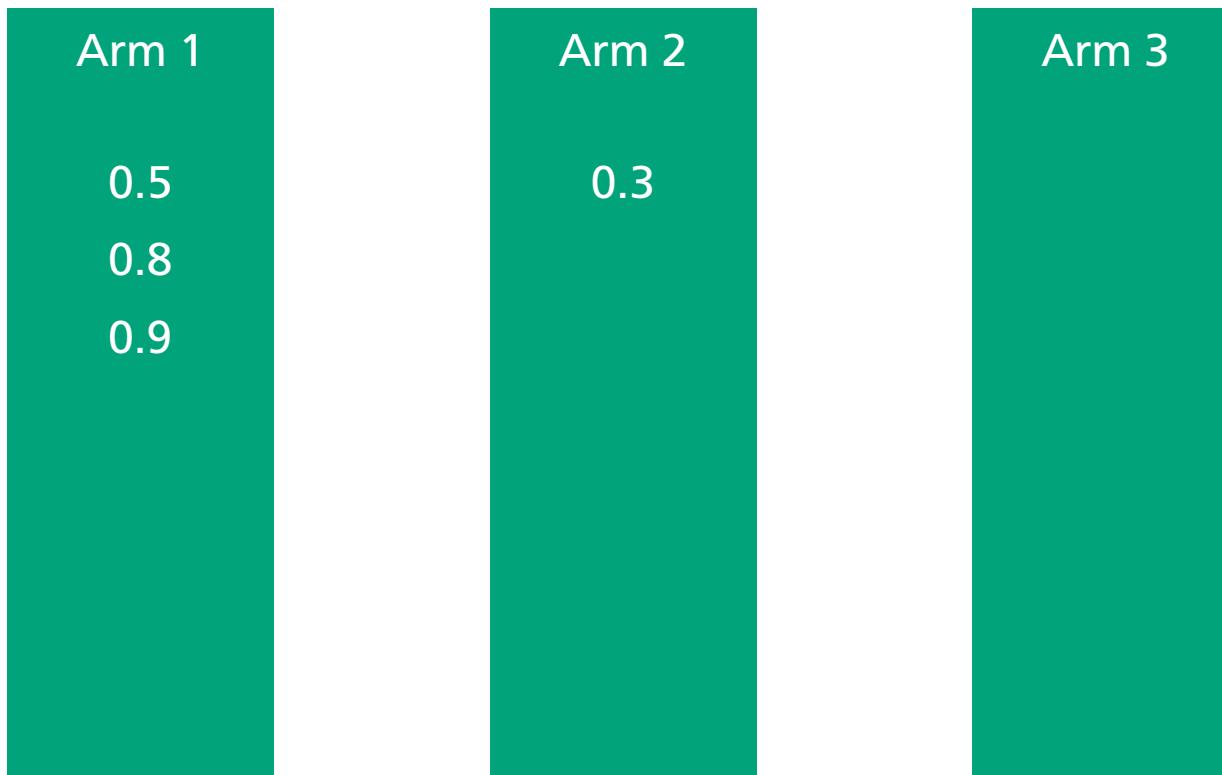
Example



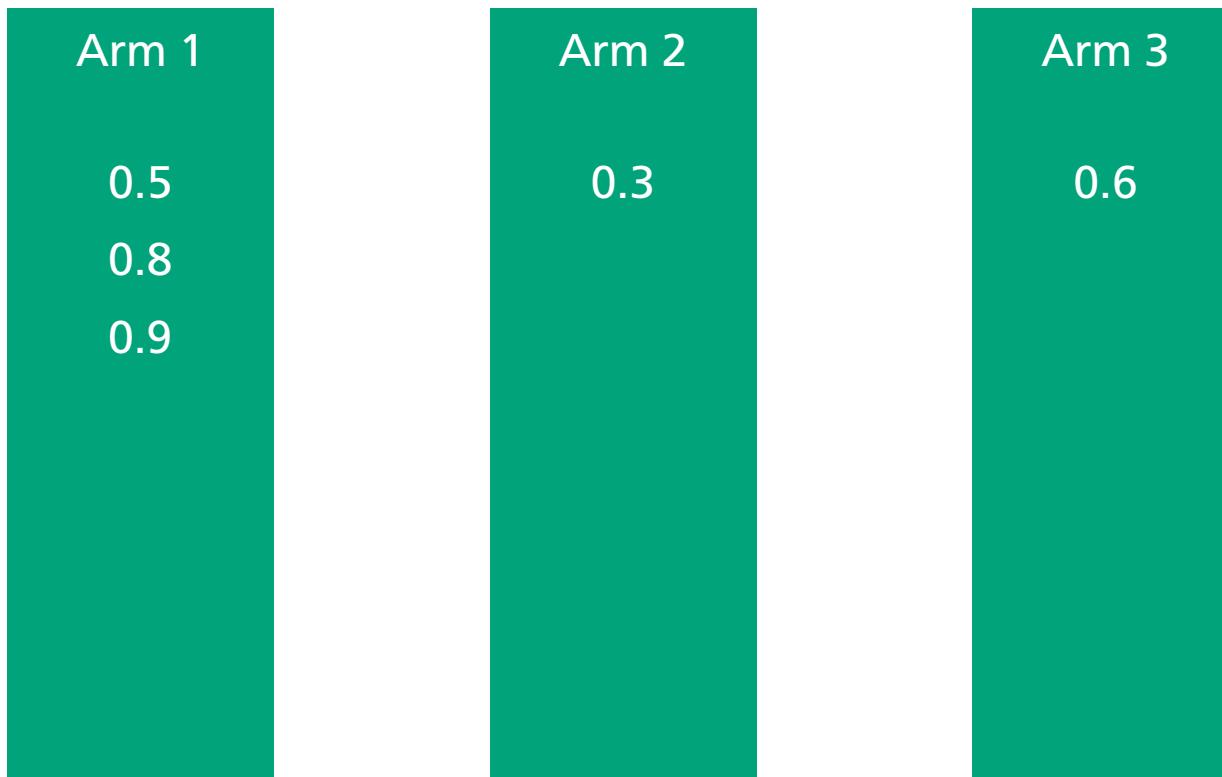
Example



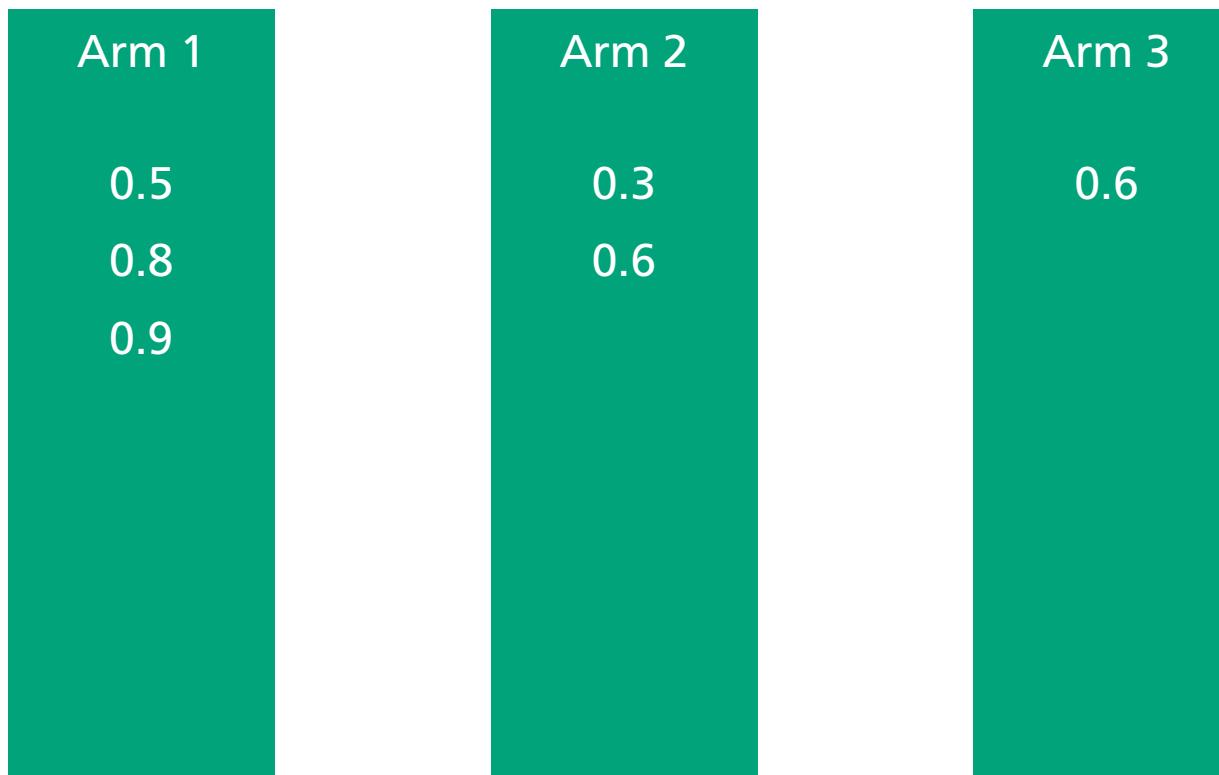
Example



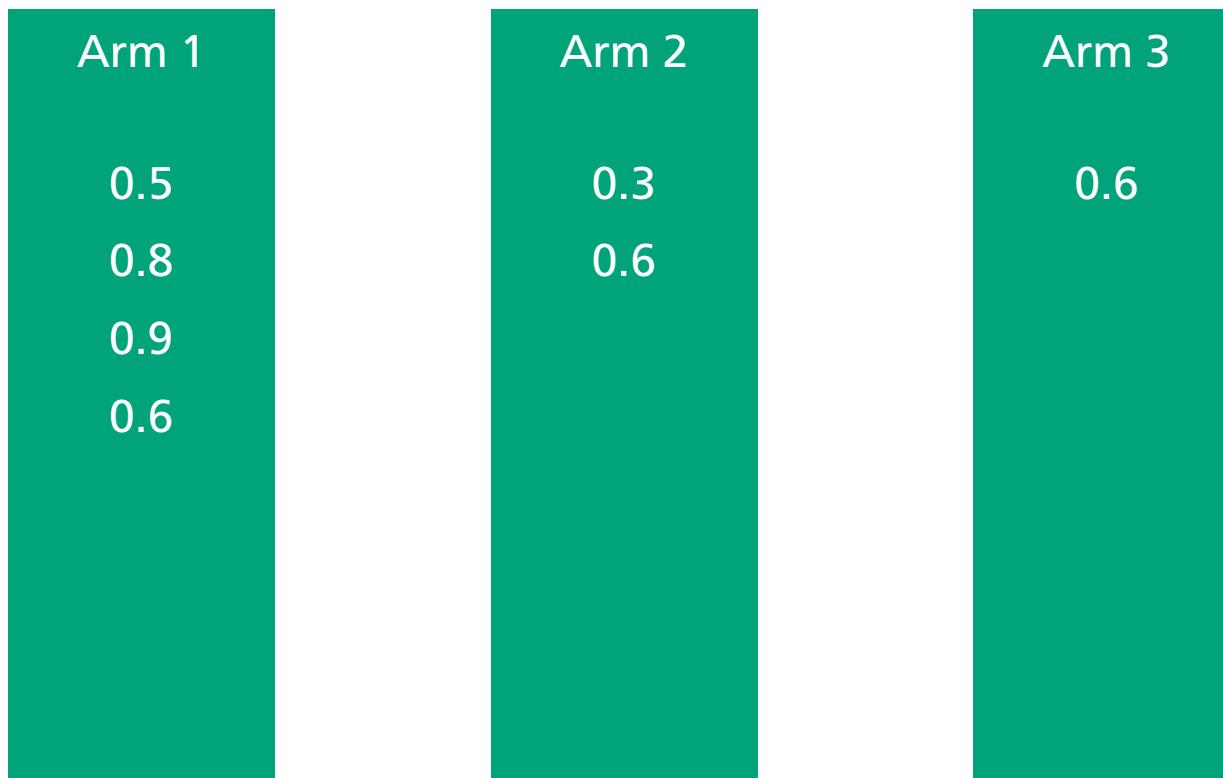
Example



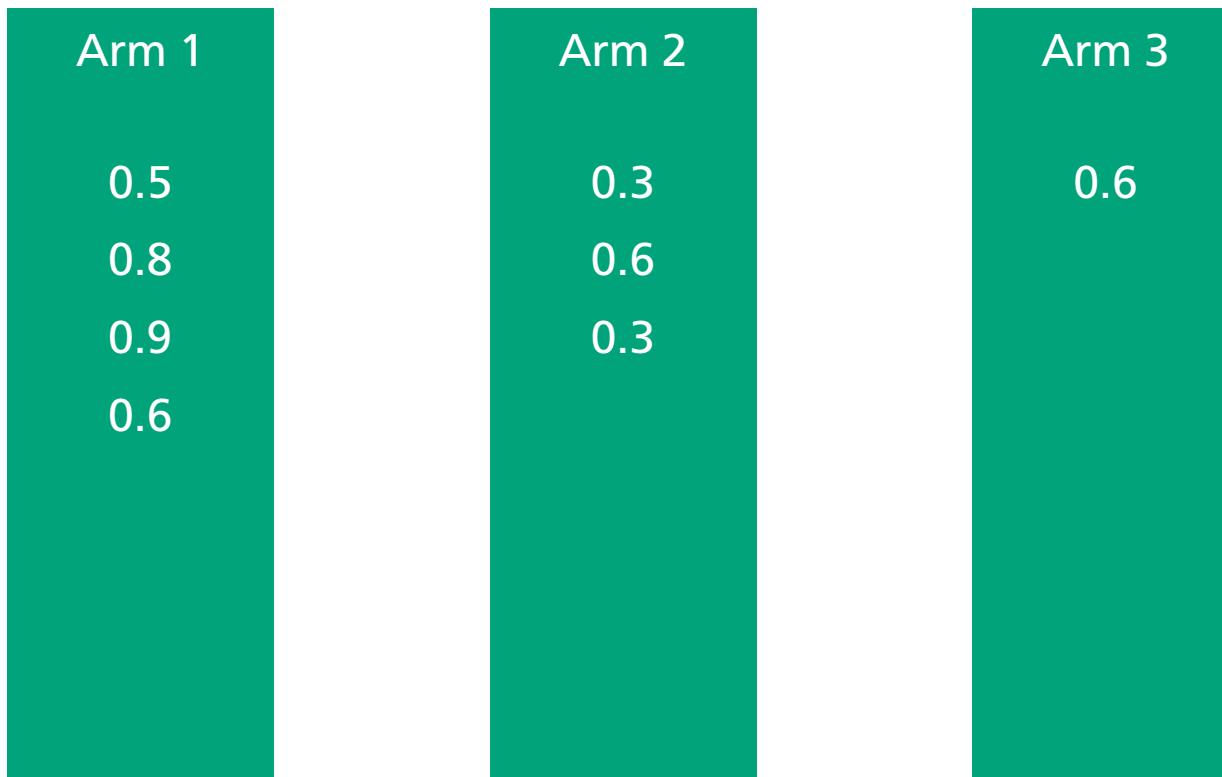
Example



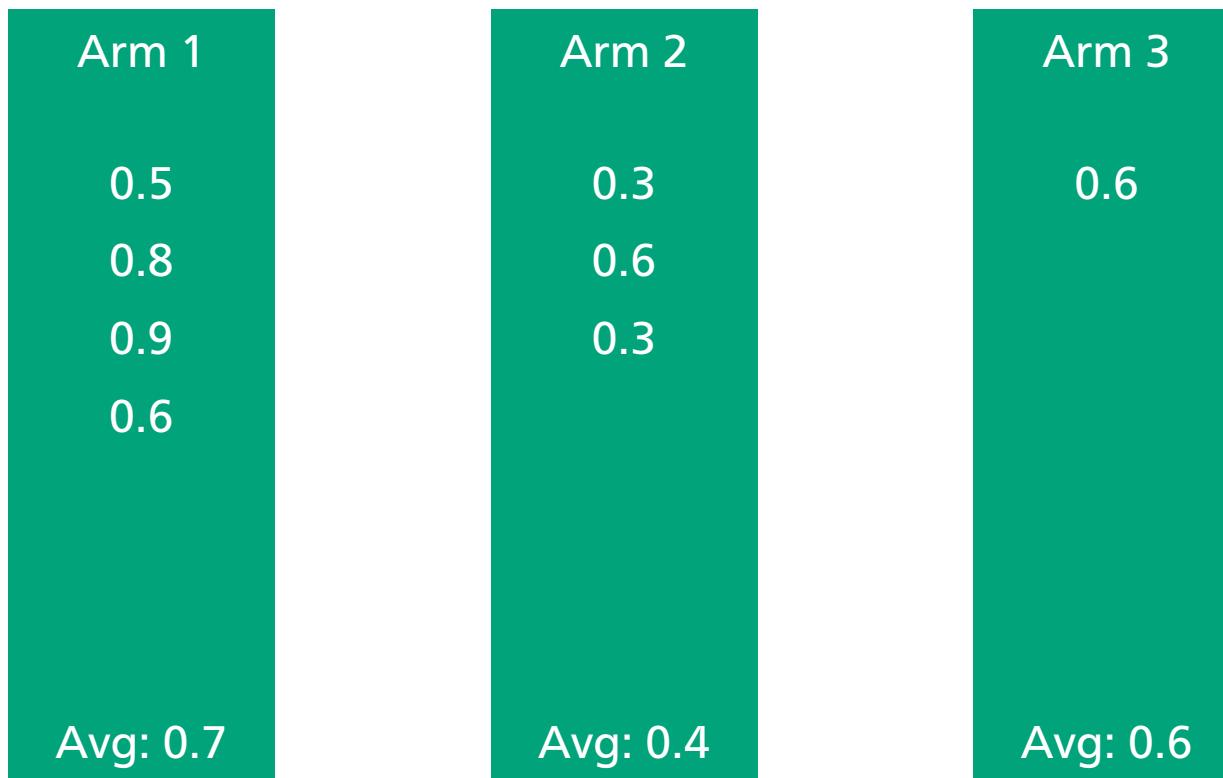
Example



Example



Example



Exploration and exploitation

If we only care about getting the best reward now, we can pick the arm with highest average

However, we might overlooked an even better action that we underestimate, and we will never find it by being greedy

Much time left / less certain: do more exploration

Less time left / more certain: do more exploitation

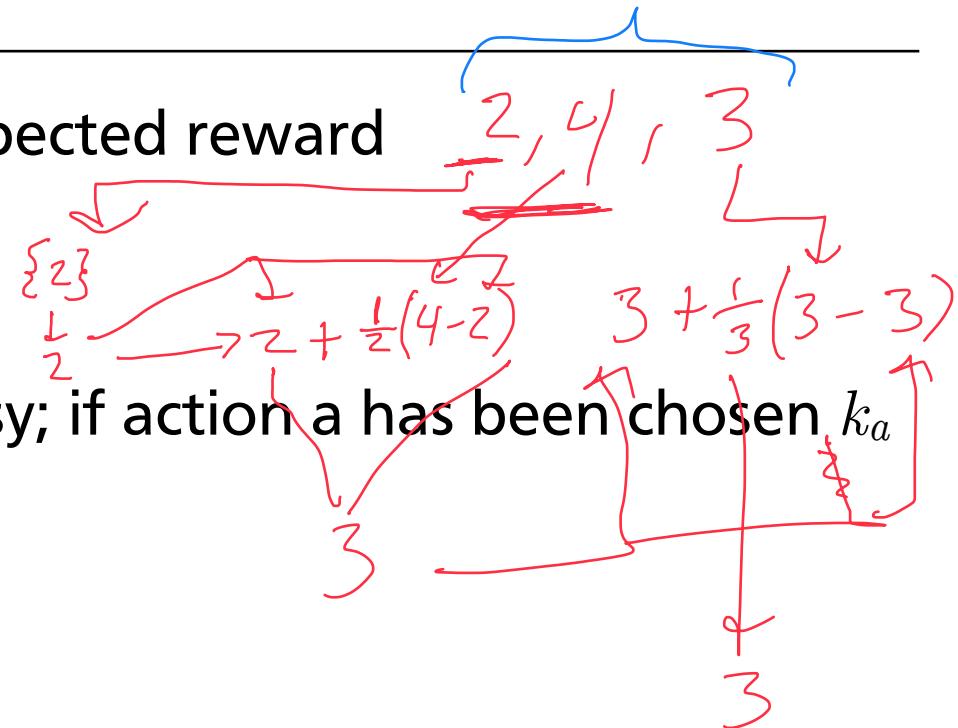
How to estimate the value of each arm?

The value of an action is the expected reward

$$Q_t(a) = \mathbb{E}[r_t | a_t]$$

Estimating an expectation is easy; if action a has been chosen k_a times, yielding rewards r_1, \dots, r_{k_a} :

$$\hat{Q}_t(a) = \frac{\sum_{i=1}^{k_a} r_i}{k_a}$$



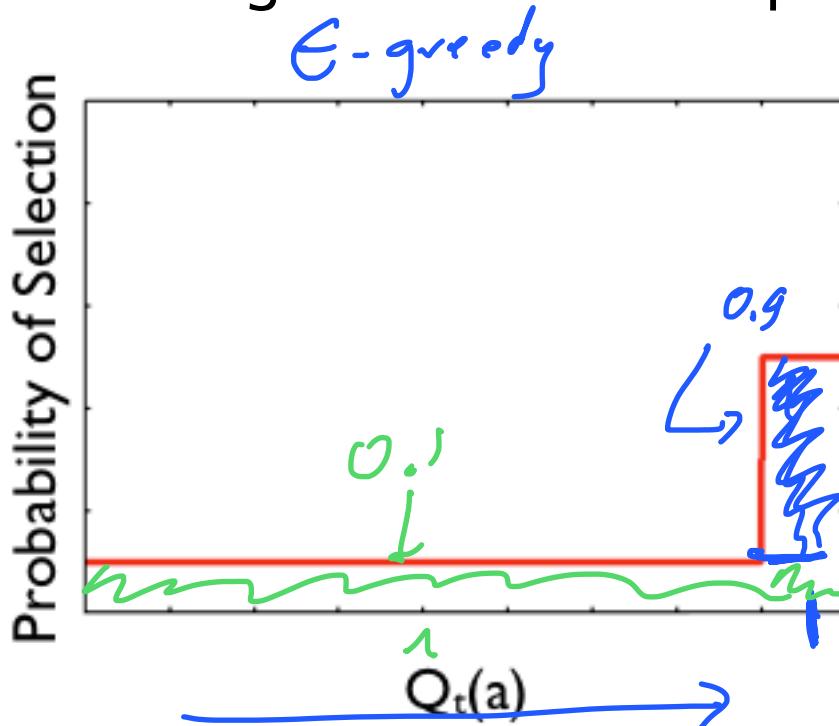
But this requires storing all rewards, ever. Incremental solution:

$$\hat{Q}_{t+1}(a) = \hat{Q}_t(a) + \frac{1}{k_a + 1} [r_t - \hat{Q}_t(a)]$$

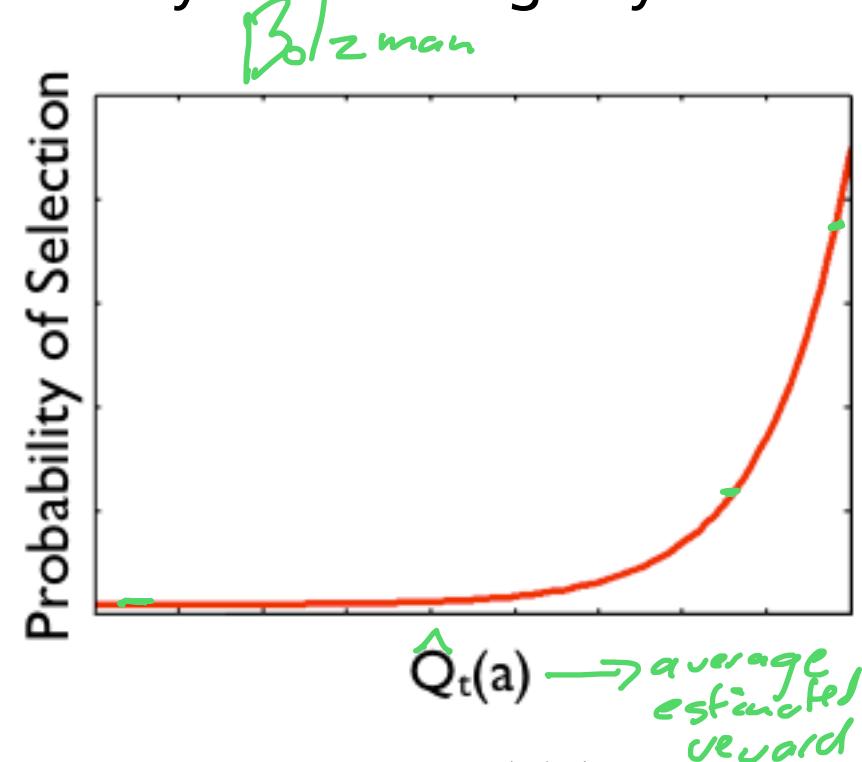
#times before ↑

Exploration strategies

Soft strategies: ensure non-0 probability of choosing any action



Choose action deemed best
with probability $1-\varepsilon$, random
otherwise



$$p(a) = \frac{e^{Q(a)/\tau}}{\sum_{a'} e^{Q(a')/\tau}}$$

Figures: Sutton&Barto, RL:AI
Reinforcement Learning

Exploration strategies

Optimism:

Use mean and uncertainty
(Upper confidence bound)

What happens now when
optimal action wasn't tried
yet?

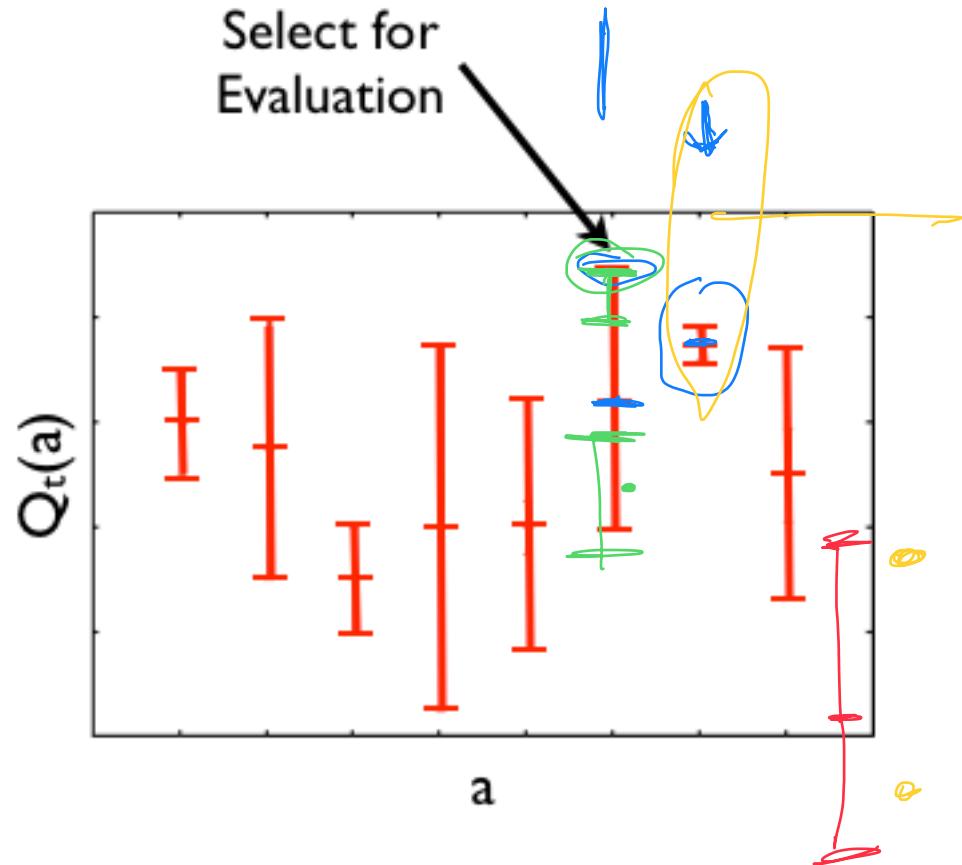


Figure: Sutton&Barto, RL:AI

Exploration strategies

Optimism:

Use mean and uncertainty
(Upper confidence bound)

What happens now when
optimal action wasn't tried
yet?

Alternative: Initialize all Q
values optimistically, then only
allow them to change slowly

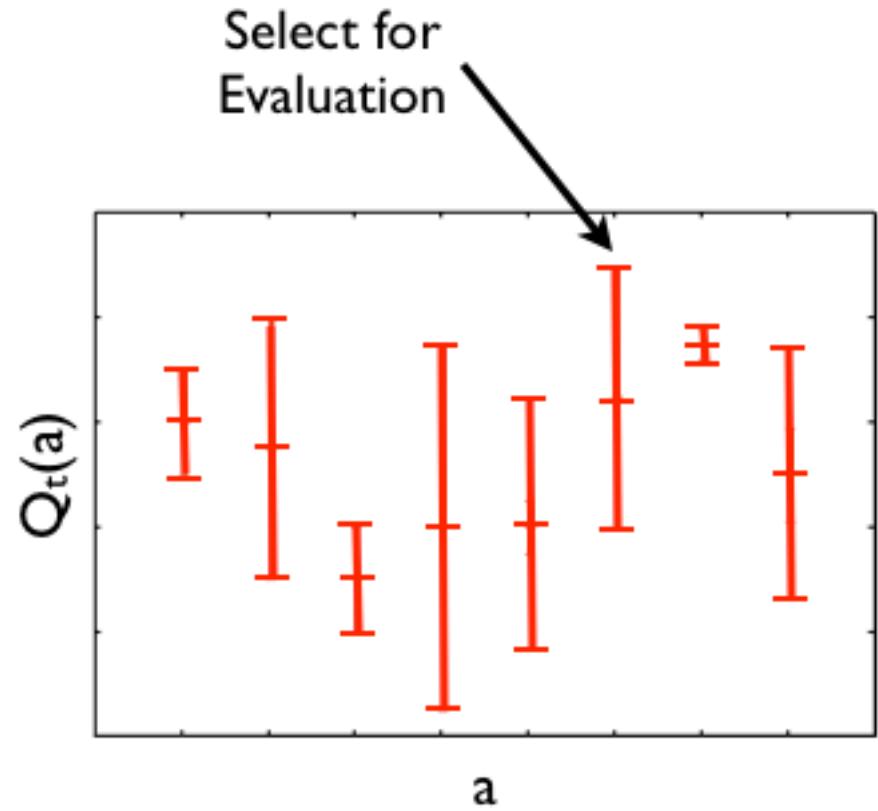


Figure: Sutton&Barto, RL:AI

Most decision problems are different...

In many decision problems, we aren't just searching for one 'optimal' action

- There are many different situations (*states s*)
- The best action depends on the situation
- The bandit framework can be extended to learn policies

$$a = \underline{\pi}(s)$$

Most decision problems are different...

In many decision problems, we aren't just searching for one 'optimal' action

- There are many different situations (*states s*)
- The best action depends on the situation
- The bandit framework can be extended to learn policies

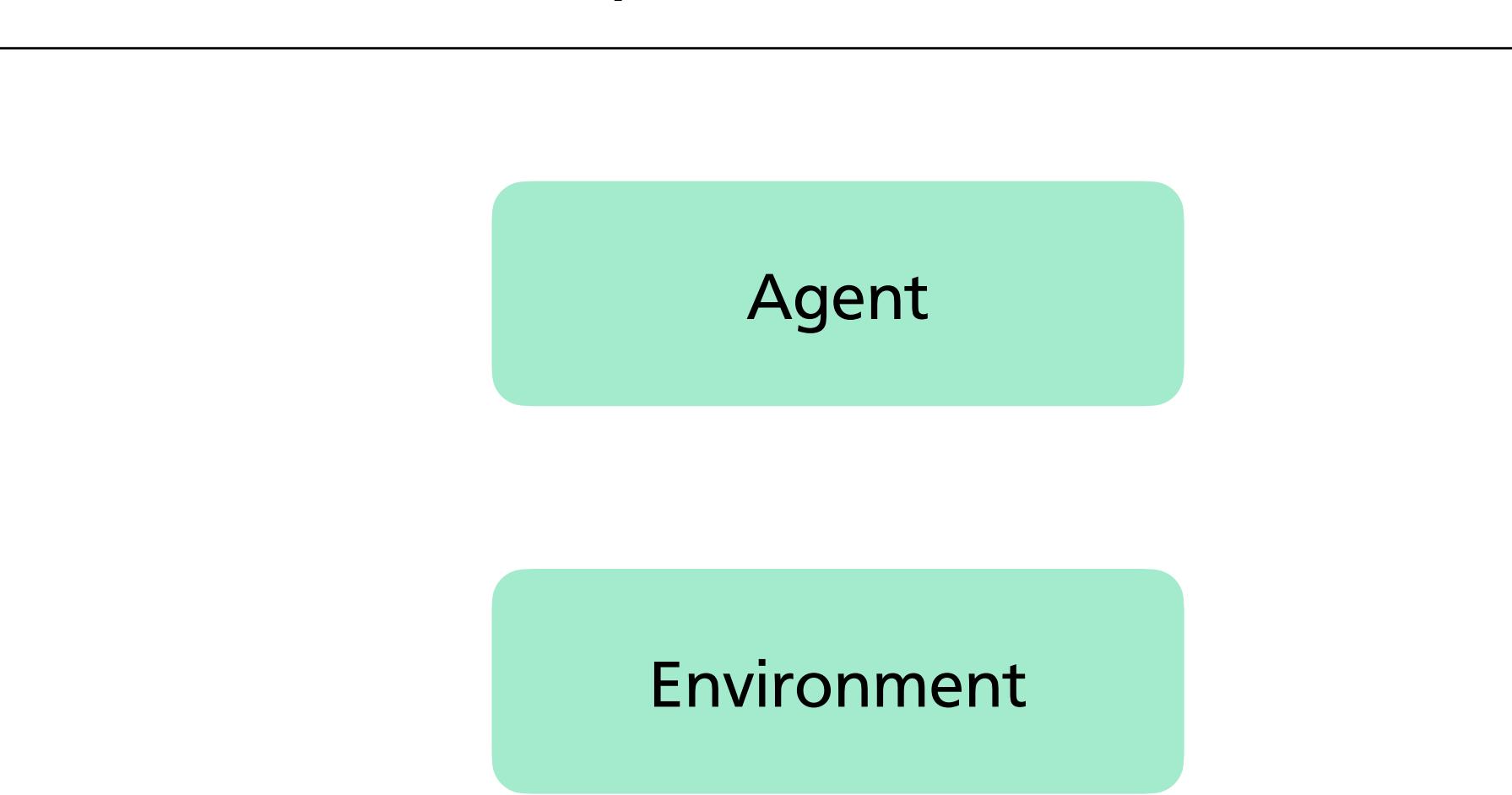
$$a = \pi(s)$$

contextual bandit.

More complex: actions can change the current situation, and affect which action should be chosen next

- Now we have a decision process
- Let's look at a simple model for such processes
- This will be the main topic for the rest of the course!

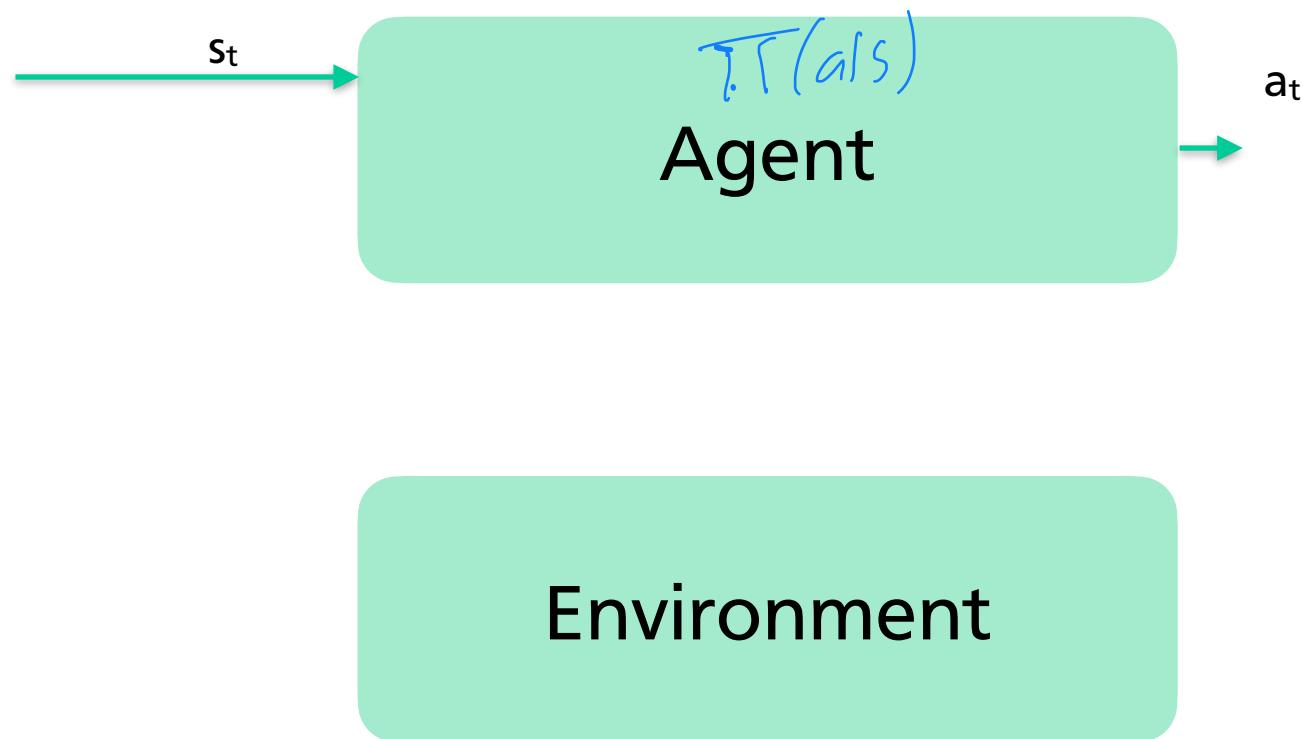
Markov decision process



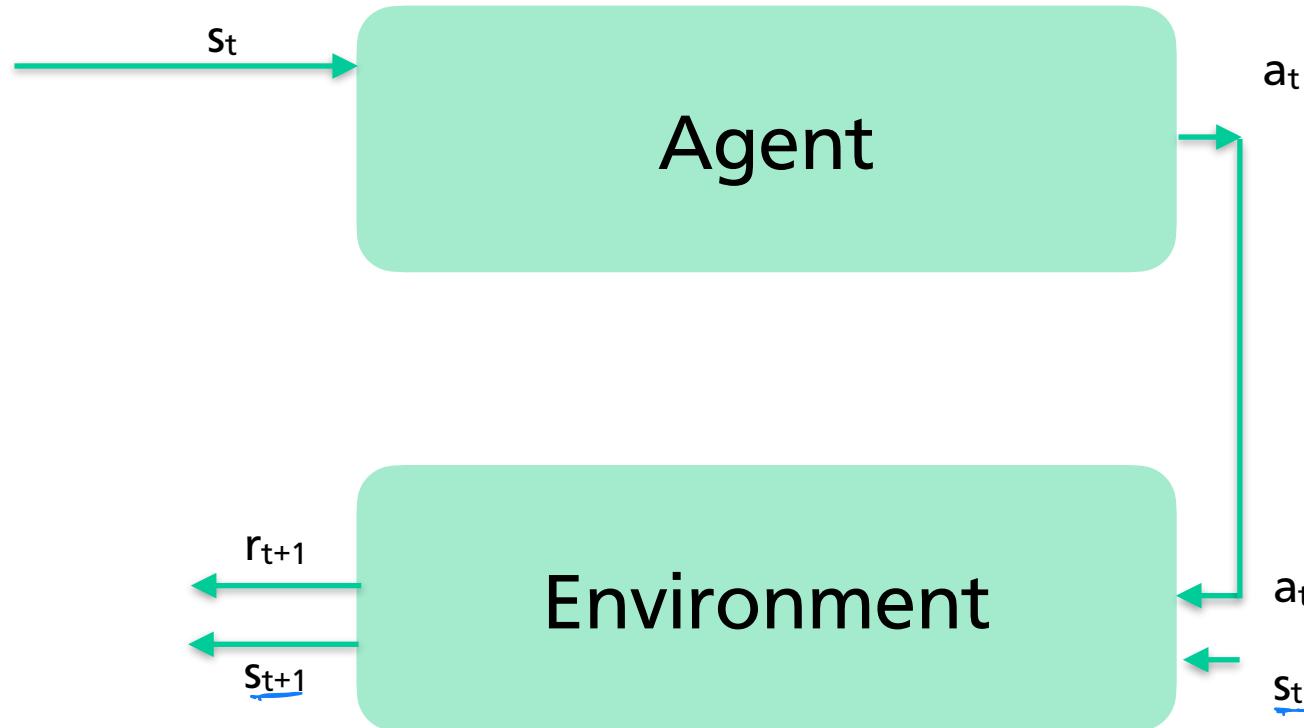
Agent

Environment

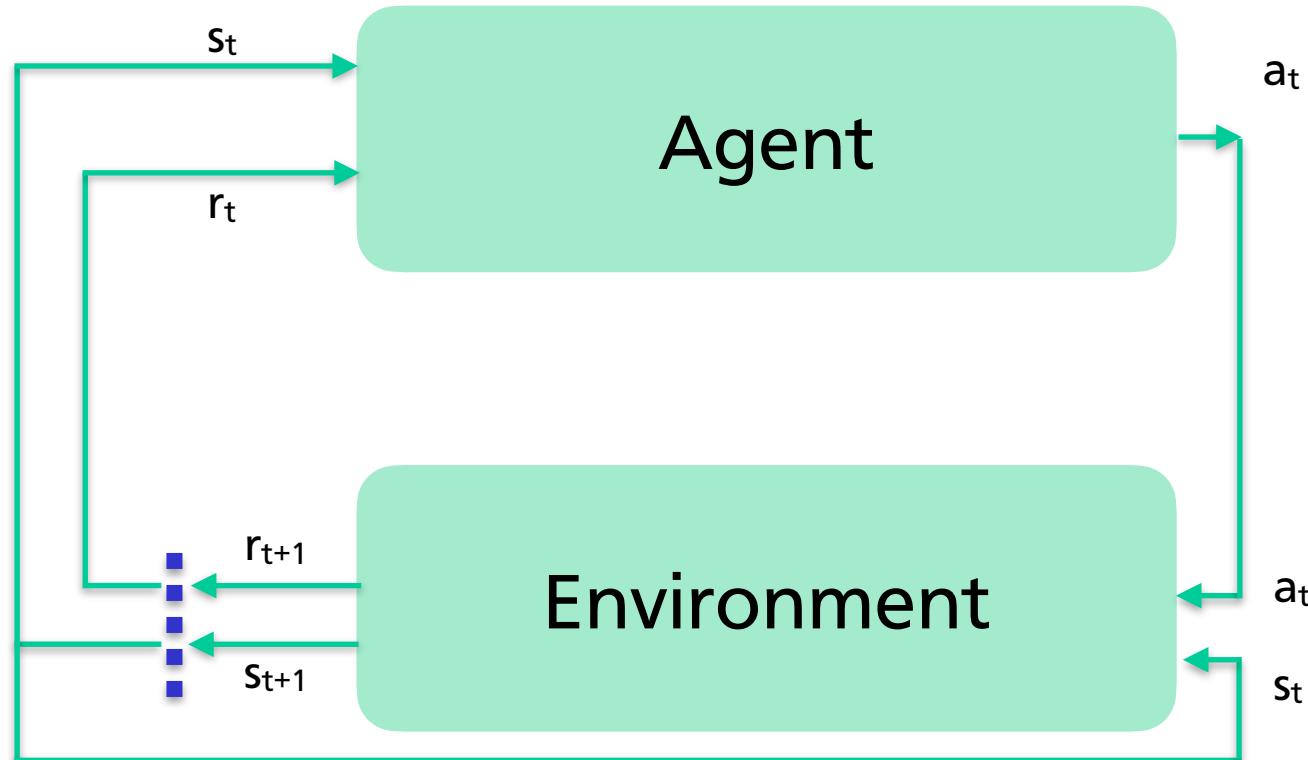
Markov decision process



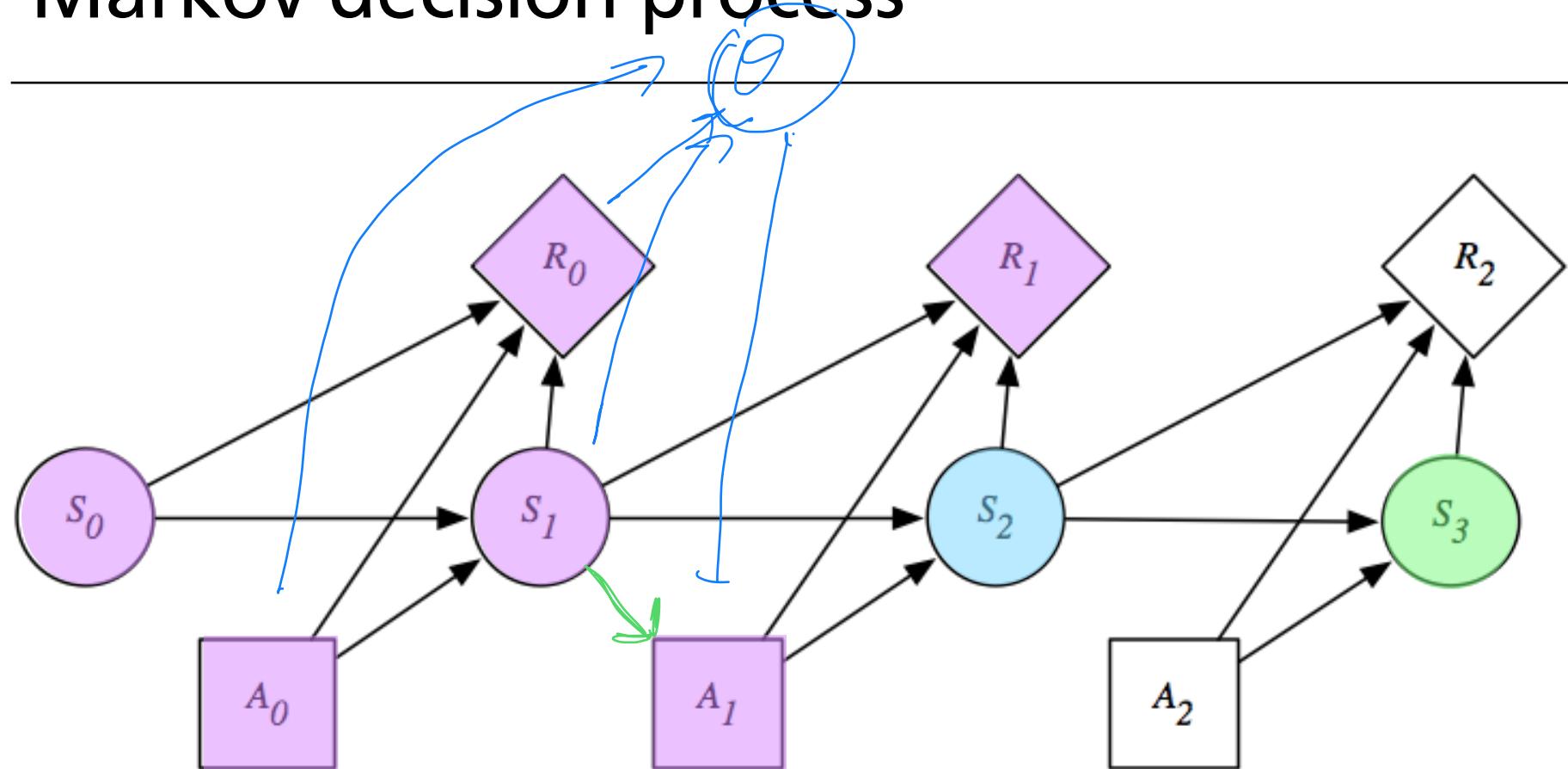
Markov decision process



Markov decision process



Markov decision process



Markov decision processes (MDPs)

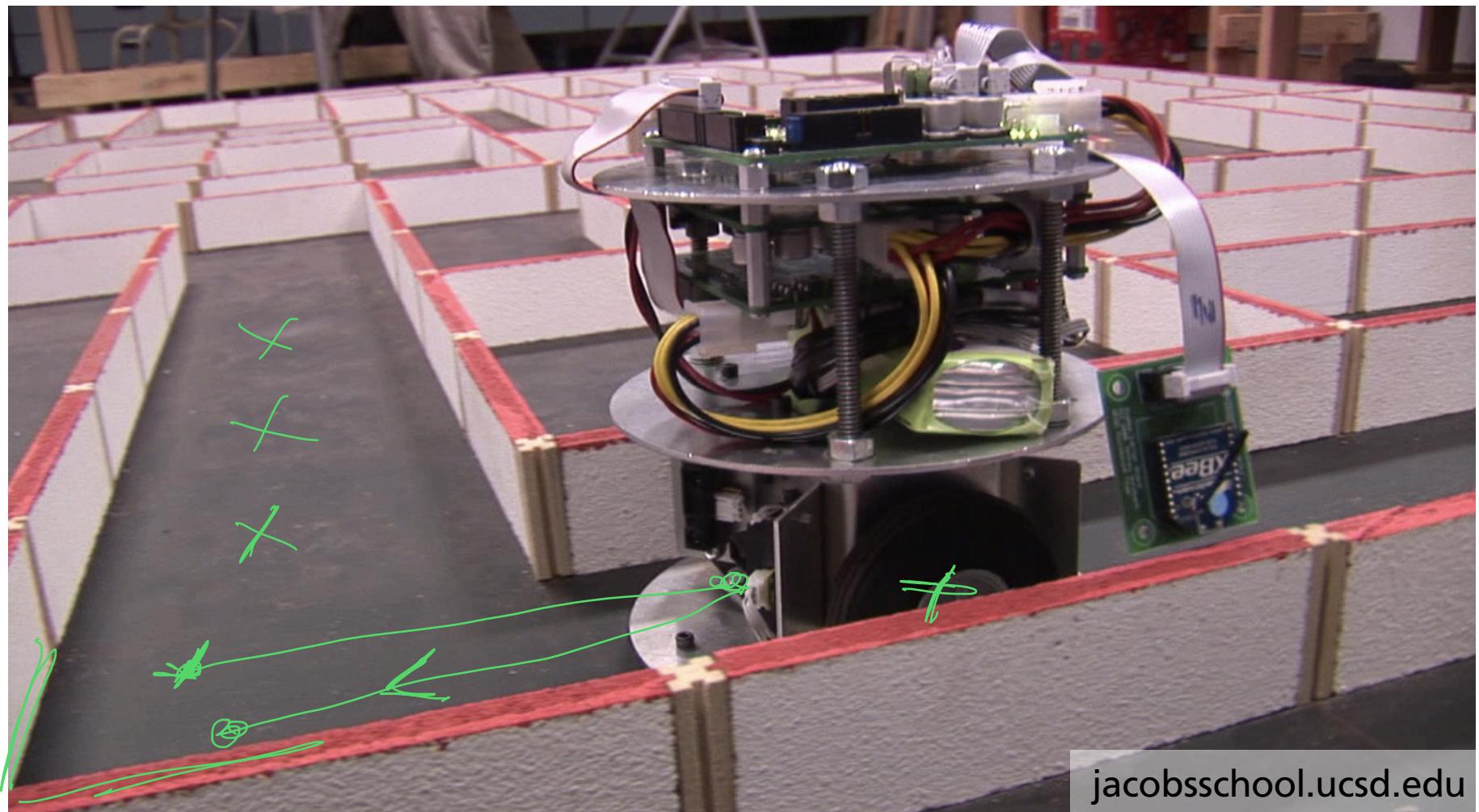
Assumptions:

- The next state depends **only** on the current state and action, and not on any states from longer ago
- Rewards depends **only** on (some of) the state, action, and next state
- Discrete time steps

This implies:

- Environment is fully observable (we always know exactly in which state we are). There is no hidden information.
- Unless the time is part of the state, transitions, reward, or termination cannot depend on the time. The environment is **stationary**.

Markov decision process



jacobsschool.ucsd.edu

Markov decision processes (MDPs)

Most work in RL is focused on MDPs. Why are they so hard?

- Credit assignment is hard:

Suppose we get a good reward. Was the last action the most important in getting it? Or did an earlier action put us in a good position?

Suppose we get a bad reward. Maybe, the last action was the best from a bad situation, and it could have been even worse...

Markov decision processes (MDPs)

Most work in RL is focused on MDPs. Why are they so hard?

- Credit assignment is hard:

Suppose we get a good reward. Was the last action the most important in getting it? Or did an earlier action put us in a good position?

Suppose we get a bad reward. Maybe, the last action was the best from a bad situation, and it could have been even worse...

- Data is non-iid and depends on the actions chosen

Data comes in sequences. This violates common ML assumptions

Initial actions might be very bad, and stay in bad states. When we learn to take better actions, suddenly we get to good states but we don't know what to do...

Markov decision processes

Think of the decision processes we mentioned earlier, or new examples.

- Which of these are Markov? Which aren't?
- Does it depend on uncertain details? How?
- Can you think of one other example in each category?

Chess (with / without opponent's fixed reactive policy)

Robot in maze (with / without exact position sensing)

Autonomous driving

Ad serving

Markov decision processes

Formally, a finite MDP consists of:

- A finite set of states
- A finite set of actions for each state (often the same in all states)
- A dynamics function

$$p(\underbrace{s', r}_{\text{transition}}, \underbrace{s, a}_{\text{action}}) \doteq \Pr \{S_t = s', R_t = r | S_{t-1} = s, A_{t-1} = a\}$$

Sometimes written as:
transition function $p(s' | s, a)$
reward function $p(r | s, a, s')$

- A discount factor $\gamma \in [0, 1)$

Return and horizons

Rewards encode what to achieve

We want to maximise the expected cumulative reward

Cumulative reward is called the **return**

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

where T is length of an episode (one game, one run in maze, ...)

This only makes sense if the task is episodic
(every episode ends in a terminal state)

Return and horizons

In continuing tasks the return is problematic!

We'll use the notion of **discounted return**

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Rest of the course: finding optimal policies

Next lecture:

define optimal policies & find them for known MDPs



After that:

learn policies for **unknown** MDPs

Various conditions, e.g. discrete / continuous states and actions

Thanks for your attention!

Feedback?

h.c.vanhoof@uva.nl