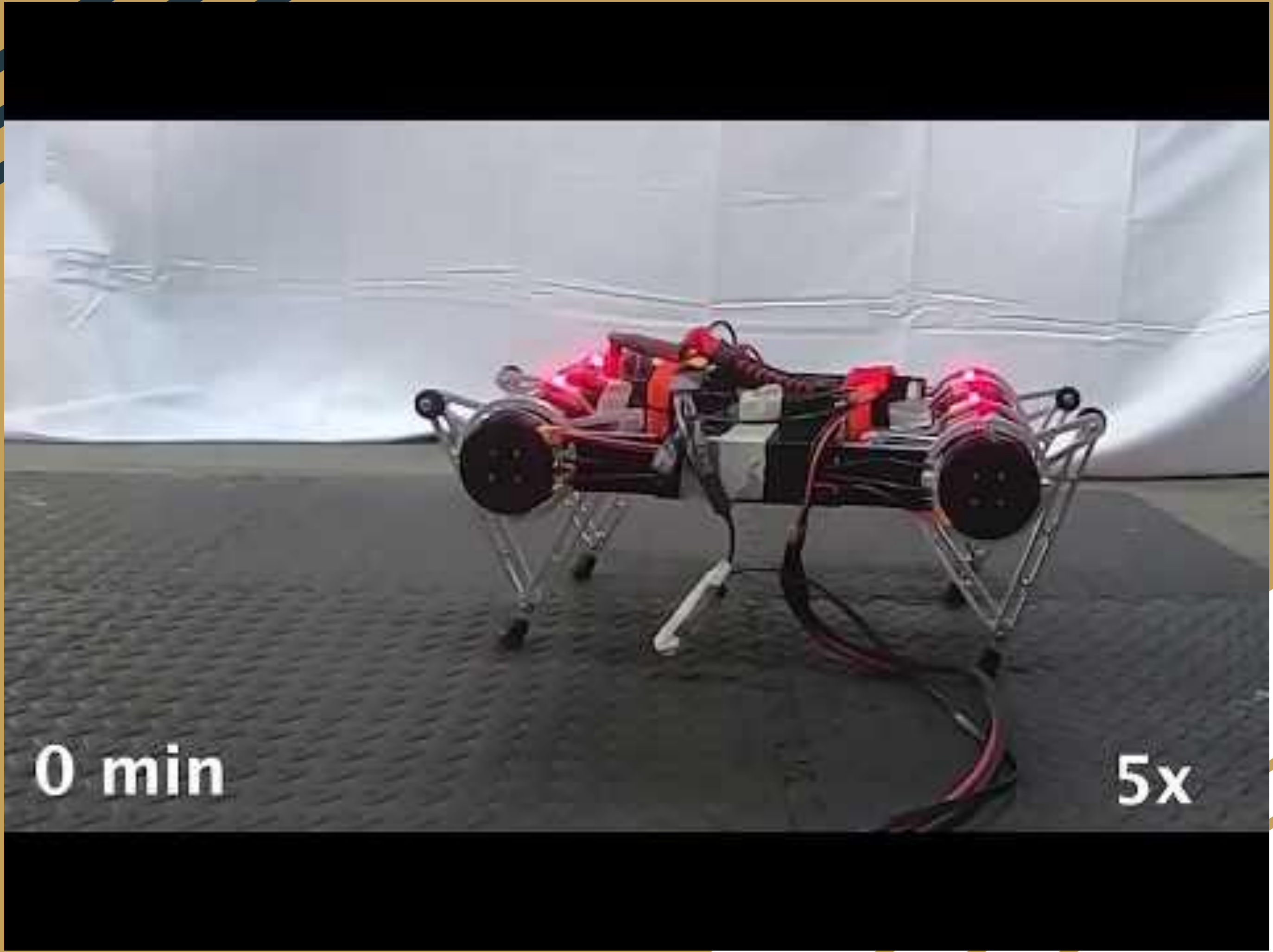
Learning by Demonstration

Session 1

Dr. Anaís Garrell


0 min
5x

Overview - Part 1

The K-Armed Bandit Problem

What to Learn? Estimating Action Values

Exploration vs. Exploitation Tradeoff

Overview - Part 1

The K-Armed Bandit Problem

What to Learn? Estimating Action Values

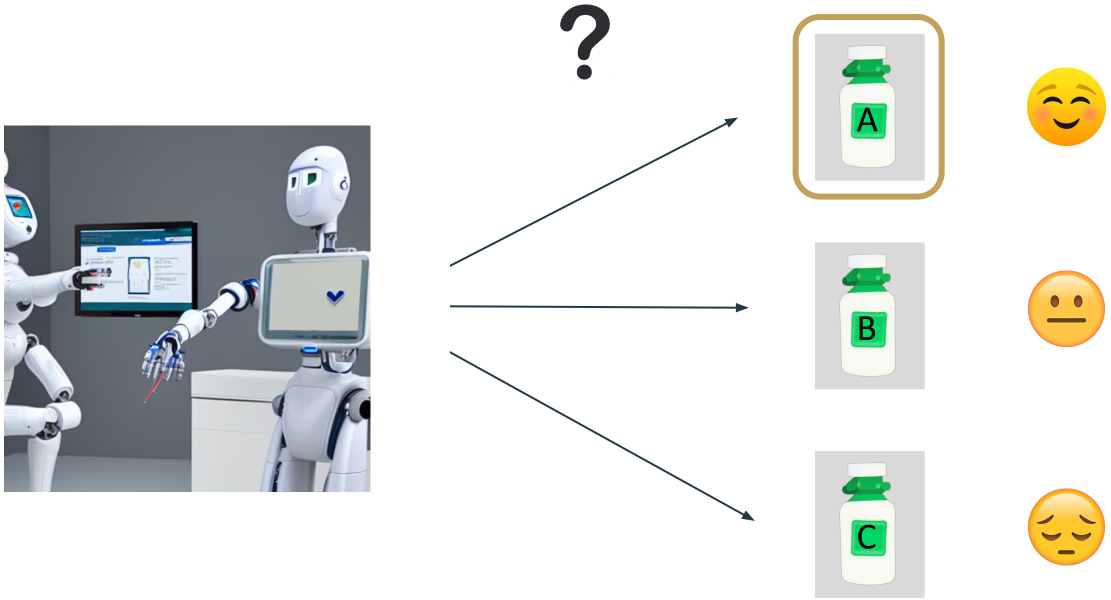
Exploration vs. Exploitation Tradeoff

The K-Armed Bandit Problem

Formalize the problem of decision-making under uncertainty using **k-armed bandits**,

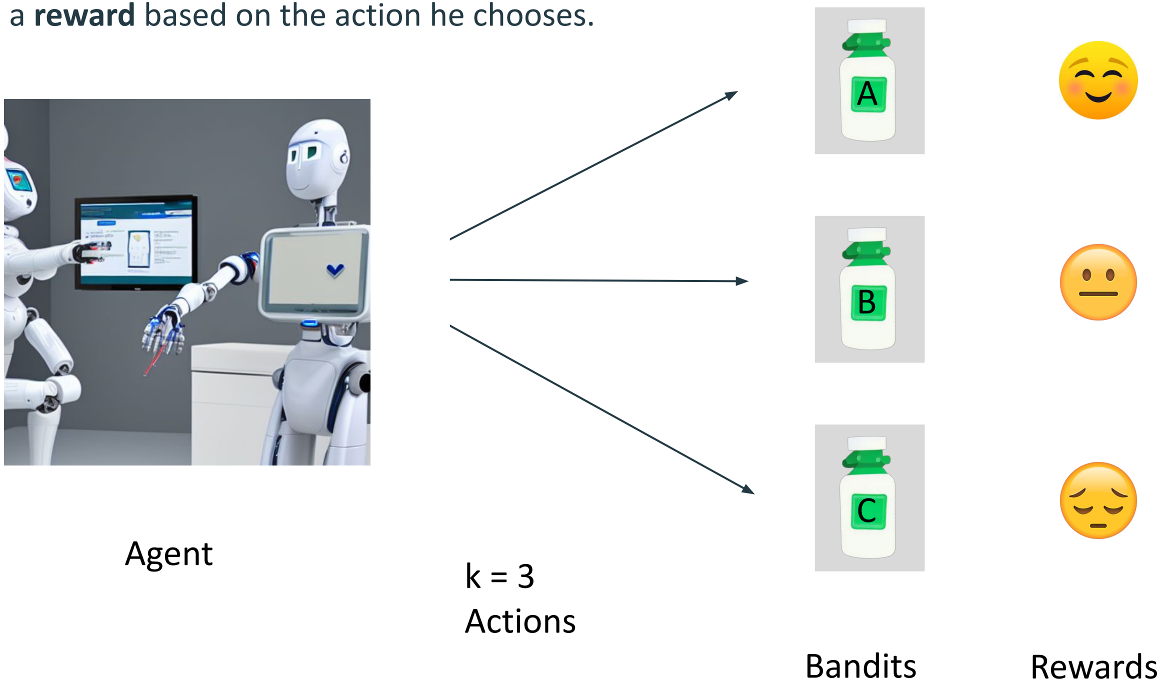
Use this bandit problem to describe fundamental concepts and reinforcement learning, such as **rewards**, **timesteps**, and **values**

The K-Armed Bandit Problem



The K-Armed Bandit Problem

In the k-armed bandit problem, we have a **decision-maker** or agent, who chooses between k different **actions**, and receives a **reward** based on the action he chooses.



The K-Armed Bandit Problem

Action-Values

The **value** is the expected **reward**

$$q_*(a) \doteq \mathbb{E}[R_t \mid A_t = a] \quad \forall a \in \{1, \ldots, k\}$$
$$= \sum_r p(r \mid a)\, r$$

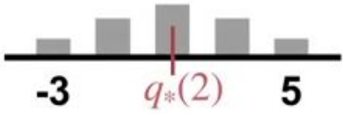
The goal is to **maximize** the expected **reward**

$$\underset{a}{\operatorname{argmax}} \; q_*(a)$$

The K-Armed Bandit Problem

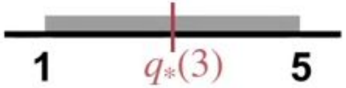
Action-Values

Calculating



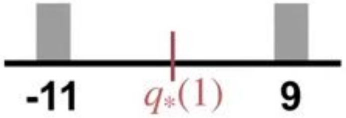
Binomial distribution

$$q_*(a) = 1$$



Uniform distribution

$$q_*(a) = 3$$



Bernoulli distribution

$$q_*(a) = .5 \times -11 + .5 \times 9$$

Overview - Part 1

The K-Armed Bandit Problem

What to Learn? Estimating Action Values

Exploration vs. Exploitation Tradeoff

What to Learn? Estimating Action Values

Value of an action

The value an action is the expected reward when the action is taken

$$q_*(a) \doteq \mathbb{E}[R_t | A_t = a] \quad \forall a \in \{1, \ldots, k\}$$

$q_*(a)$ is not known, we must estimate it

What to Learn? Estimating Action Values

Value of an action

The value an action is the expected reward when the action is taken

$$q_*(a) \doteq \mathbb{E}[R_t | A_t = a] \quad \forall a \in \{1, \ldots, k\}$$

$q_*(a)$ is not known, we must estimate it

Sample-average method

$$Q_t(a) \doteq \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t}$$

$$= \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$

What to Learn? Estimating Action Values

Sample-average method

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$

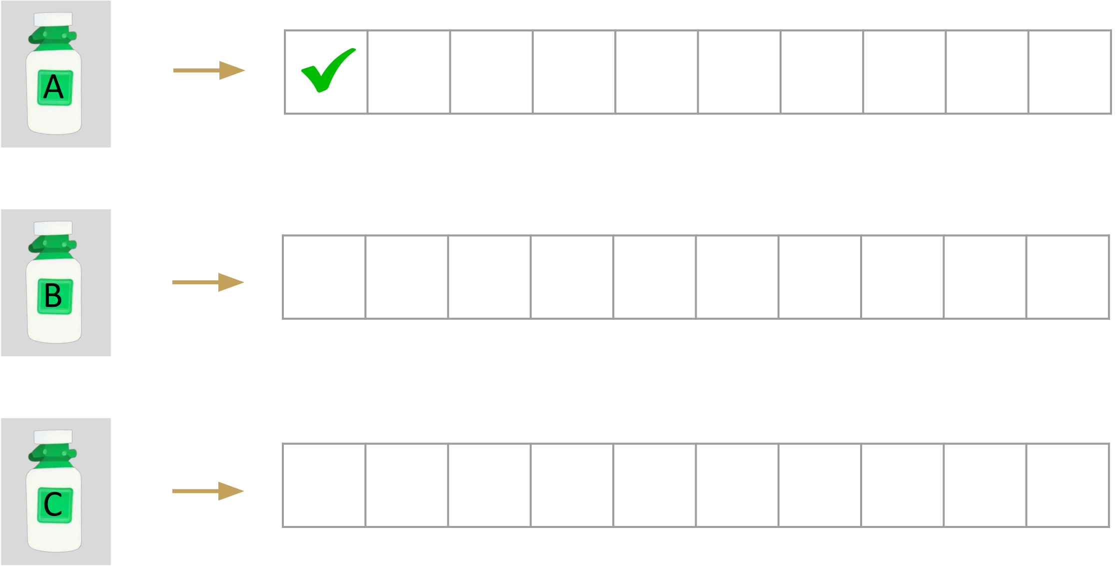
											$Q_t(a)$ =
 →											0
 →											0
 →											0

What to Learn? Estimating Action Values

Sample-average method

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$



$Q_t(a)$ =

1.0

0

0

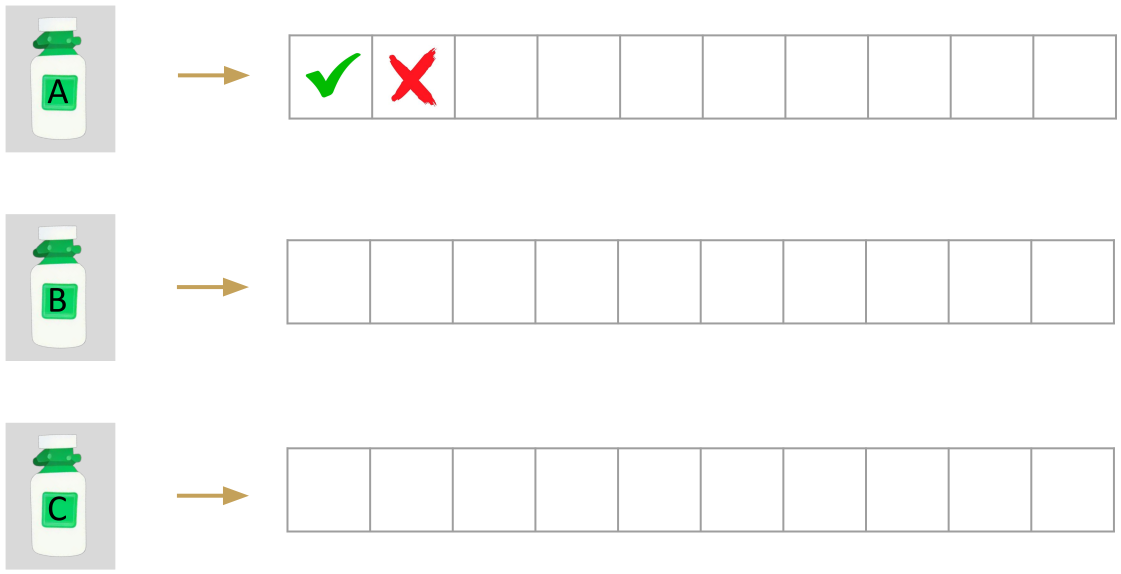
What to Learn? Estimating Action Values

Sample-average method

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

$Q_t(a)$:

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$



0.5

0

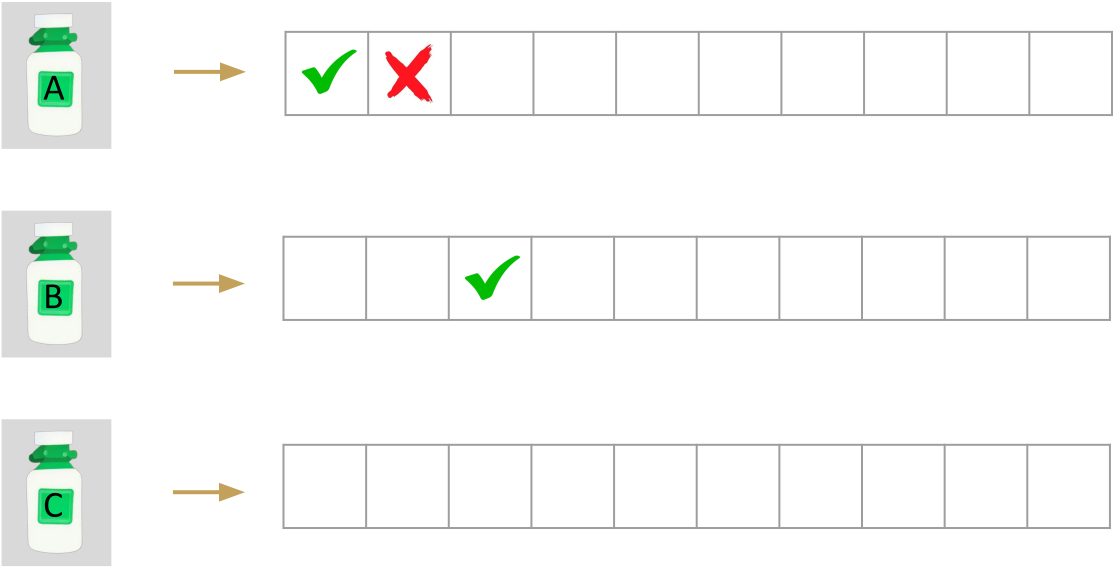
0

What to Learn? Estimating Action Values

Sample-average method

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$



$Q_t(a)$:

0.5

1.0

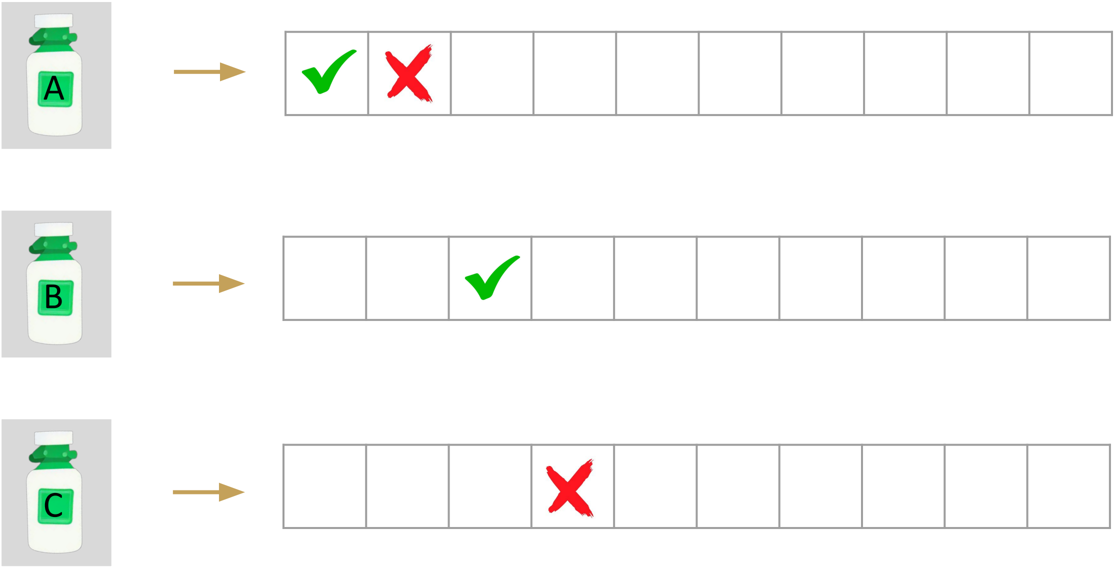
0

What to Learn? Estimating Action Values

Sample-average method

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$



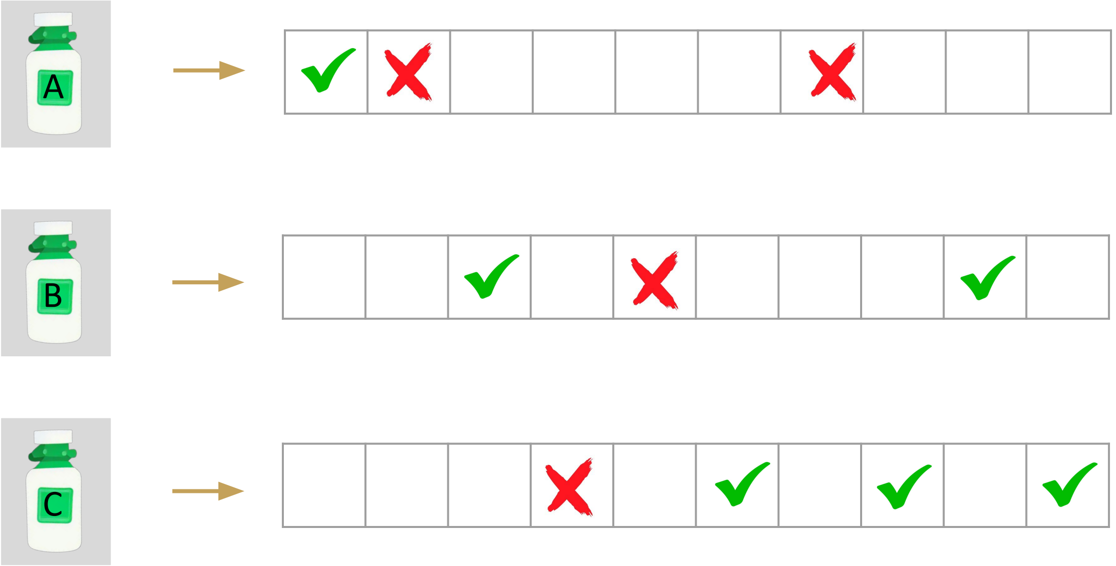
	$Q_t(a)$:
A	0.5
B	1.0
C	0

What to Learn? Estimating Action Values

Sample-average method

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$



$Q_t(a)$:

0.33

0.66

0.75

What to Learn? Estimating Action Values

Action Selection

$Q_t(a)$ =

→ 0.33

→ 0.66

→ 0.75

Non-Greedy actions:

$$a_g = \underset{a}{\operatorname{argmax}}\, Q(a)$$

Greedy action:

What to Learn? Estimating Action Values

Action Selection

$Q_t(a)$:

→ 0.33

→ 0.66

→ 0.75

Non-Greedy actions:

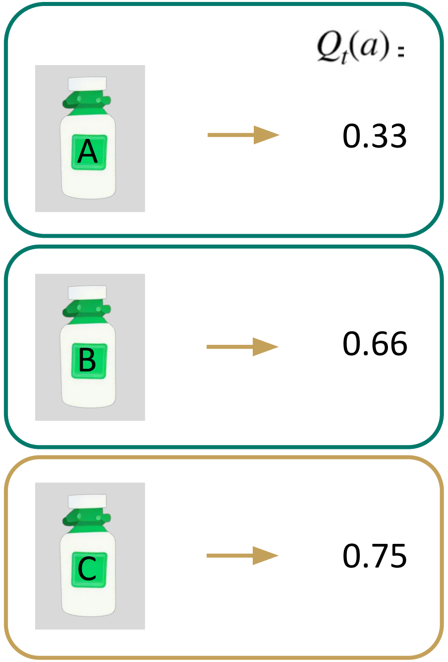
$$a_g = \underset{a}{\text{argmax}}\ Q(a)$$

Greedy action:

- The action that currently has the largest estimated value.
- Selecting the greedy action means the agent is exploiting its current knowledge.
- It is trying to get the most reward it can right now.

What to Learn? Estimating Action Values

Action Selection



Non-Greedy actions:

- The agent would sacrifice immediate reward hoping to gain more information about the other actions.
- The agent can not choose to both **explore** and **exploit** at the same time.
- This is one of the fundamental problems in reinforced learning.

$$a_g = \underset{a}{\operatorname{argmax}}\, Q(a)$$

Greedy action:

- The action that currently has the largest estimated value.
- Selecting the greedy action means the agent is exploiting its current knowledge.
- It is trying to get the most reward it can right now.

What to Learn? Estimating Action Values

Incremental update rule

Q_{n+1}

Recall

$$Q_n = \frac{1}{n-1}\sum_{i=1}^{n-1} R_i$$

What to Learn? Estimating Action Values

Incremental update rule

$$Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i$$

Recall

$$Q_n = \frac{1}{n-1}\sum_{i=1}^{n-1} R_i$$

What to Learn? Estimating Action Values

Incremental update rule

$$Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i$$
$$= \frac{1}{n}(R_n + \sum_{i=1}^{n-1} R_i)$$

Recall

$$Q_n = \frac{1}{n-1}\sum_{i=1}^{n-1} R_i$$

What to Learn? Estimating Action Values

Incremental update rule

$$
\begin{aligned}
Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\
&= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right)
\end{aligned}
$$

Recall

$$Q_n = \frac{1}{n-1}\sum_{i=1}^{n-1} R_i$$

What to Learn? Estimating Action Values

Incremental update rule

Recall

$$Q_n = \frac{1}{n-1}\sum_{i=1}^{n-1} R_i$$

$$\begin{aligned}
Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\
&= \frac{1}{n}(R_n + \sum_{i=1}^{n-1} R_i) \\
&= \frac{1}{n}(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i)
\end{aligned}$$

$$\longrightarrow$$

$$\begin{aligned}
&= \frac{1}{n}(R_n + (n-1)Q_n) \\
&= \frac{1}{n}(R_n + nQ_n - Q_n) \\
&= \boxed{Q_n} + \frac{1}{n}(R_n - Q_n)
\end{aligned}$$

What to Learn? Estimating Action Values

Incremental update rule

$$\text{NewEstimate} \leftarrow \text{OldEstimate} + \text{StepSize}\,[\,\text{Target} - \text{OldEstimate}\,]$$

What to Learn? Estimating Action Values

Incremental update rule

NewEstimate $\leftarrow$ OldEstimate + StepSize $\boxed{[\ \text{Target - OldEstimate}]}$

error in the estimate: the difference between the old estimate and the new target

$$Q_{n+1} = Q_{n} + \frac{1}{n}\boxed{(R_n - Q_n)}$$

What to Learn? Estimating Action Values

Incremental update rule

NewEstimate $\longleftarrow$ OldEstimate + StepSize [$\boxed{\text{Target}}$ - OldEstimate]

New Reward: our target

$$Q_{n+1} = Q_{n-} + \frac{1}{n}(\boxed{R_n} - Q_n)$$

What to Learn? Estimating Action Values

Incremental update rule

NewEstimate $\longleftarrow$ OldEstimate + $\boxed{\text{StepSize}}$ [Target - OldEstimate]

The size of the step: is determined by our step size parameter.

$$Q_{n+1} = Q_{n-} + \boxed{\frac{1}{n}}(R_n - Q_n)$$

What to Learn? Estimating Action Values

Incremental update rule

NewEstimate $\leftarrow$ OldEstimate + StepSize [Target - OldEstimate]

$$Q_{n+1} \doteq Q_n + \alpha_n \Big[R_n - Q_n \Big]$$

$\alpha_n \in (0, 1]$ is constant.

$\alpha_n(a) = \frac{1}{n}$

In the specific case of the sample average

What to Learn? Estimating Action Values

Non-stationary bandit problem

$Q_t(a)$:

→ 0.33

→ 0.66

→ 0.75

$$Q_{n+1} \doteq Q_n + \alpha_n \left[R_n - Q_n \right]$$

α_n

What to Learn? Estimating Action Values

Non-stationary bandit problem

$Q_t(a)$:

→ 0.33

→ 0.66

→ 0.75

$$Q_{n+1} \doteq Q_n + \alpha_n \Big[R_n - Q_n \Big]$$

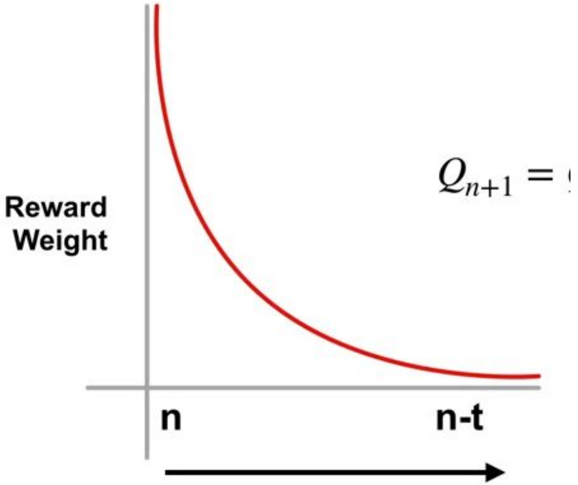
- What if one of the treatments was more effective under certain conditions?
- For instance, treatment B is more effective during the summer months.
- This is an example of a **non-stationary bandit problem**

α_n

What to Learn? Estimating Action Values

Non-stationary bandit problem

$$Q_{n+1} \doteq Q_n + \alpha_n \Big[R_n - Q_n \Big]$$

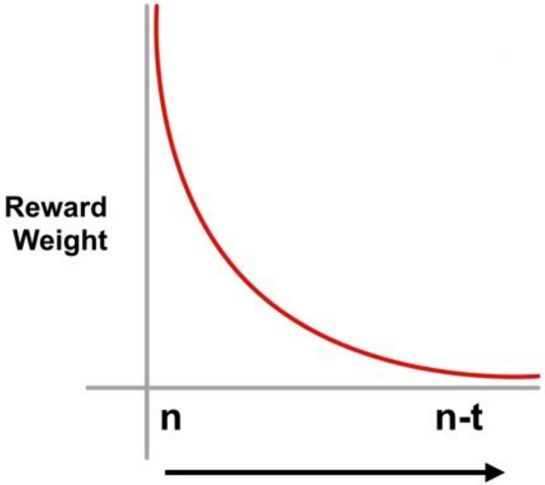


- These problems are like the bandit problems, except the distribution of rewards changes with time.
- The agent is unaware of this change but would like to adapt to it.
- One option is to use a fixed step size. If α_n is constant like 0.1, then the most recent rewards affect the estimate more than older rewards

What to Learn? Estimating Action Values

Non-stationary bandit problem

$$Q_{n+1} \doteq Q_n + \alpha_n \Big[R_n - Q_n \Big]$$



- This graph shows the amount of weight the most recent award receives versus the reward received T time steps ago.
- The weighting fades exponentially with time.
- As we move to the right on the x-axis, we go further back in time.

What to Learn? Estimating Action Values

Non-stationary bandit problem

$$Q_{n+1}$$

What to Learn? Estimating Action Values

Non-stationary bandit problem

$$Q_{n+1} = Q_n + \alpha\Big[R_n - Q_n\Big]$$

What to Learn? Estimating Action Values

Non-stationary bandit problem

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha) Q_n
\end{aligned}
$$

What to Learn? Estimating Action Values

Non-stationary bandit problem

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha) Q_n \\
&= \alpha R_n + (1-\alpha)\left[\alpha R_{n-1} + (1-\alpha) Q_{n-1}\right]
\end{aligned}
$$

What to Learn? Estimating Action Values

Non-stationary bandit problem

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha)Q_n \\
&= \alpha R_n + (1-\alpha)\left[\alpha R_{n-1} + (1-\alpha)Q_{n-1}\right] \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 Q_{n-1}
\end{aligned}
$$

What to Learn? Estimating Action Values

Non-stationary bandit problem

$$\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha)Q_n \\
&= \alpha R_n + (1-\alpha)\left[\alpha R_{n-1} + (1-\alpha)Q_{n-1}\right] \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 Q_{n-1} \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 \alpha R_{n-2} + \\
&\qquad\qquad \cdots + (1-\alpha)^{n-1}\alpha R_1 + (1-\alpha)^n Q_1
\end{aligned}$$

What to Learn? Estimating Action Values

Non-stationary bandit problem

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha)Q_n \\
&= \alpha R_n + (1-\alpha)\left[\alpha R_{n-1} + (1-\alpha)Q_{n-1}\right] \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 Q_{n-1} \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 \alpha R_{n-2} + \\
&\qquad\qquad \cdots + (1-\alpha)^{n-1}\alpha R_1 + (1-\alpha)^n Q_1 \\
&= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i .
\end{aligned}
$$

Q_1 : initial action-value

Overview - Part 1

The K-Armed Bandit Problem

What to Learn? Estimating Action Values

$\boxed{\textbf{Exploration vs. Exploitation Tradeoff}}$

Exploration vs. Exploitation Tradeoff

Exploration and Exploitation

Exploration: improve knowledge for long-term benefit about each action

- By improving the accuracy of the estimated action values, the agent can make more informed decisions in the future.

Exploration vs. Exploitation Tradeoff

Exploration and Exploitation

Exploration: improve knowledge for long-term benefit about each action

- By improving the accuracy of the estimated action values, the agent can make more informed decisions in the future.

Estimated value for picking that treatment.

$q(a) = 3$	$q(a) = 5$	$q(a) = 2$
$N(a) = 3$	$N(a) = 3$	$N(a) = 3$
$q_*(a) = 4$	$q_*(a) = 5$	$q_*(a) = 3$

Exploration vs. Exploitation Tradeoff

Exploration and Exploitation

Exploration: improve knowledge for long-term benefit about each action

- By improving the accuracy of the estimated action values, the agent can make more informed decisions in the future.

N is the number of times you have picked that treatment

$q(a) = 3$	$q(a) = 5$	$q(a) = 2$
$N(a) = 3$	$N(a) = 3$	$N(a) = 3$
$q_*(a) = 4$	$q_*(a) = 5$	$q_*(a) = 3$

Exploration vs. Exploitation Tradeoff

Exploration and Exploitation

Exploration: improve knowledge for long-term benefit about each action

- By improving the accuracy of the estimated action values, the agent can make more informed decisions in the future.

Value of each treatment

$q(a) = 3$	$q(a) = 5$	$q(a) = 2$
$N(a) = 3$	$N(a) = 3$	$N(a) = 3$
$q_*(a) = 4$	$q_*(a) = 5$	$q_*(a) = 3$

Exploration vs. Exploitation Tradeoff

Exploration and Exploitation

Exploration: improve knowledge for long-term benefit about each action

- By improving the accuracy of the estimated action values, the agent can make more informed decisions in the future.

$q(a) = 3$
$N(a) = 3$
$q_*(a) = 4$

$q(a) = 5$
$N(a) = 3$
$q_*(a) = 5$

$q(a) = 2$
$N(a) = 3$
$q_*(a) = 3$

Exploration vs. Exploitation Tradeoff

Exploration and Exploitation

Exploitation: exploit knowledge for short-term benefit about each action

- It exploits the agent's current estimated values.
- It chooses the **greedy** action to try to get the most reward, but by being greedy with respect to estimated values, may not actually get the most reward.

Exploration vs. Exploitation Tradeoff

Exploration and Exploitation

Exploitation: exploit knowledge for short-term benefit about each action

- Example: Pure greedy action selection can lead to sub-optimal behavior.

$q(a) = 0$	$q(a) = 0$	$q(a) = 0$
$N(a) = 0$	$N(a) = 0$	$N(a) = 0$
$q_*(a) = 4$	$q_*(a) = 5$	$q_*(a) = 3$

Exploration vs. Exploitation Tradeoff

Exploration and Exploitation

Exploitation: exploit knowledge for short-term benefit about each action

- Example: Pure greedy action selection can lead to sub-optimal behavior.
-

$q(a) = 3$	$q(a) = 0$	$q(a) = 0$
$N(a) = 5$	$N(a) = 0$	$N(a) = 0$
$q_*(a) = 3$	$q_*(a) = 5$	$q_*(a) = 3$

Exploration vs. Exploitation Tradeoff

Exploration and Exploitation

Exploitation: exploit knowledge for short-term benefit about each action

- Example: Pure greedy action selection can lead to sub-optimal behavior.
-

$q(a) = 3$	$q(a) = 0$	$q(a) = 0$
$N(a) = 5$	$\boxed{N(a) = 0}$	$\boxed{N(a) = 0}$
$q_*(a) = 3$	$q_*(a) = 5$	$q_*(a) = 3$

- The estimated values for the other actions are zero.
- The greedy action is always the same, to pick the first treatment.
- The agent never saw any samples for the other treatments.
- The estimated values for the other two actions remain far from the true values, which means the agent never discovered the best action.

Exploration vs. Exploitation Tradeoff

Exploration vs Exploitation

Exploration: improve knowledge for long-term benefit about each action

more accurate estimates of our values

Exploration vs. Exploitation Tradeoff

Exploration vs Exploitation

Exploration: improve knowledge for long-term benefit about each action

Exploitation: exploit knowledge for short-term benefit about each action

more accurate estimates of our values

more reward

Exploration vs. Exploitation Tradeoff

Exploration vs Exploitation

Exploration: improve knowledge for long-term benefit about each action

Exploitation: exploit knowledge for short-term benefit about each action

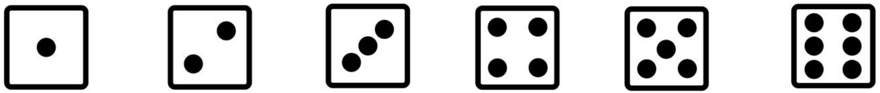
more accurate estimates of our values

more reward

- We cannot choose to do both simultaneously.
- One very simple method for choosing between exploration and exploitation is to choose randomly.
- We could choose to exploit most of the time with a small chance of exploring.

Exploration vs. Exploitation Tradeoff

Epsilon-Greedy Action Selection



explore explore explore explore explore exploit

- **Epsilon:** probability of choosing to explore.
- Here, epsilon = 1/6

Exploration vs. Exploitation Tradeoff

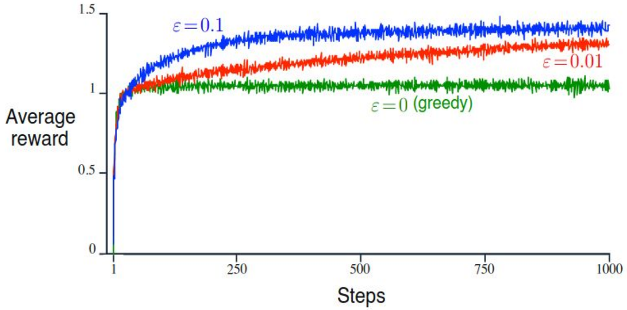
Epsilon-Greedy Action Selection

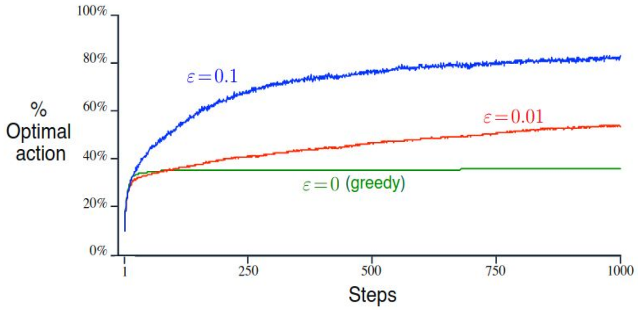
$$A_t \leftarrow \begin{cases} \underset{a}{\operatorname{argmax}}\ Q_t(a) & \textbf{with probability } 1 - \epsilon \\ a \sim Uniform(\{a_1 \ldots a_k\}) & \textbf{with probability } \epsilon \end{cases}$$

Exploration vs. Exploitation Tradeoff

Epsilon-Greedy Action Selection

$$A_t \leftarrow \begin{cases} \underset{a}{\operatorname{argmax}}\ Q_t(a) & \textbf{with probability } 1-\epsilon \\ a \sim Uniform(\{a_1 \dots a_k\}) & \textbf{with probability } \epsilon \end{cases}$$





Exploration vs. Exploitation Tradeoff

Optimistic Initial Values

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

Exploration vs. Exploitation Tradeoff

Optimistic Initial Values

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

- Assumption: each treatment is 100% effective, until shown otherwise.
- The robot would begin prescribing treatments at random, until one of the treatments fails to cure a patient.
- The robot might then choose from the other two treatments at random.
- Again, the robot would continue until one of these treatments fails to work.
- The robot would continue this way, always assuming the treatments are maximally effective, until shown that the estimated values need to be corrected.

Exploration vs. Exploitation Tradeoff

Optimistic Initial Values

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

Q1 (

) → 0

Q1 (

) → 0

Q1 (

) → 0

Exploration vs. Exploitation Tradeoff

Optimistic Initial Values

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

Q1 (

) → 2.0

Q1 (

) → 2.0

Q1 (

) → 2.0

Exploration vs. Exploitation Tradeoff

Optimistic Initial Values

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

$$Q_{n+1} = Q_n + \alpha \Big[R_n - Q_n \Big]$$

$$Q_t(a) \cancel{= \frac{\sum_{i=1}^{t-1} R_i}{t-1}}$$

$Q_t(a)$:

2

2

2

Exploration vs. Exploitation Tradeoff

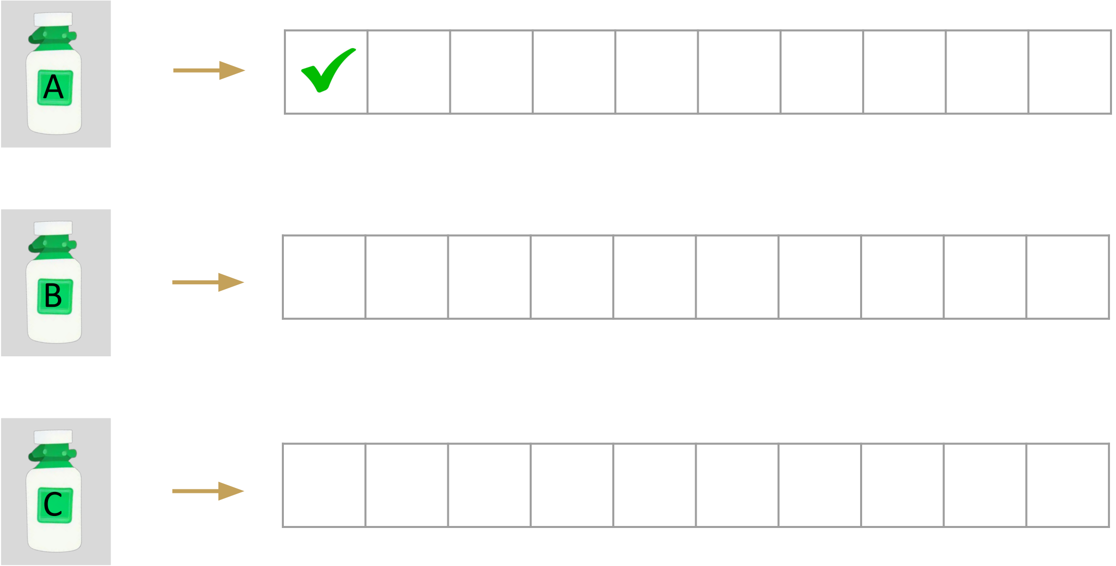
Optimistic Initial Values

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

$$Q_{n+1} = Q_n + \alpha \Big[R_n - Q_n \Big]$$

$$Q_t(a) = \cancel{\frac{\sum_{i=1}^{t-1} R_i}{t-1}}$$

alpha =0.5



$Q_t(a)$:

1.5

2

2

Exploration vs. Exploitation Tradeoff

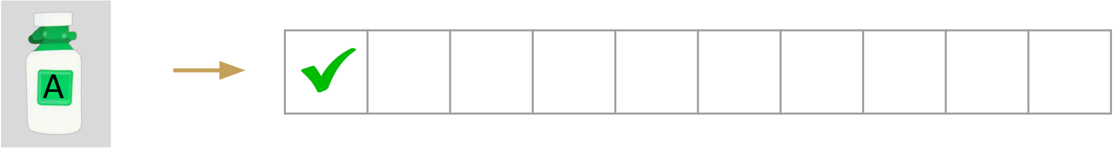
Optimistic Initial Values

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

$$Q_{n+1} = Q_n + \alpha \Big[R_n - Q_n \Big]$$

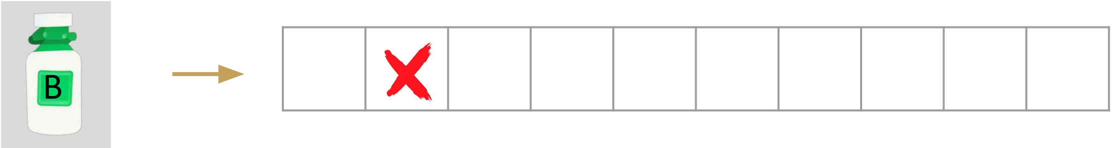
$Q_t(a)$ =

$$\cancel{Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}}$$

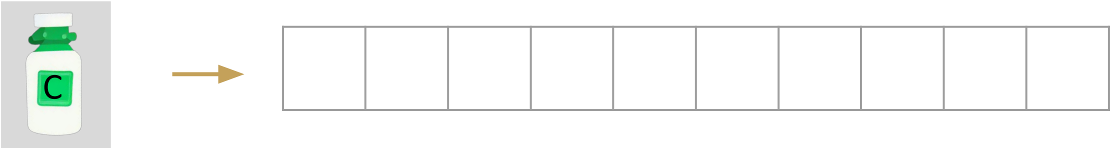


1.5

alpha =0.5



1



2

Exploration vs. Exploitation Tradeoff

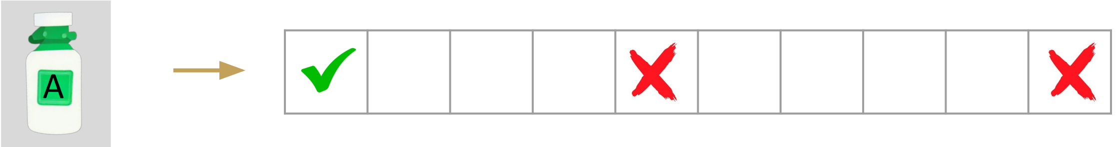
Optimistic Initial Values

If the patient gets better, the robots records a reward of one. Otherwise, the doctor records a reward of zero.

$$Q_{n+1} = Q_n + \alpha \left[R_n - Q_n \right]$$

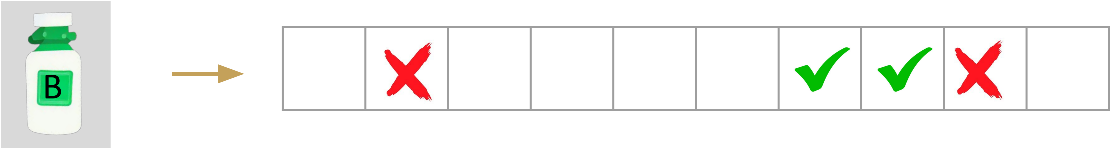
$Q_t(a)$:

$$\cancel{Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}}$$



0.375

alpha =0.5

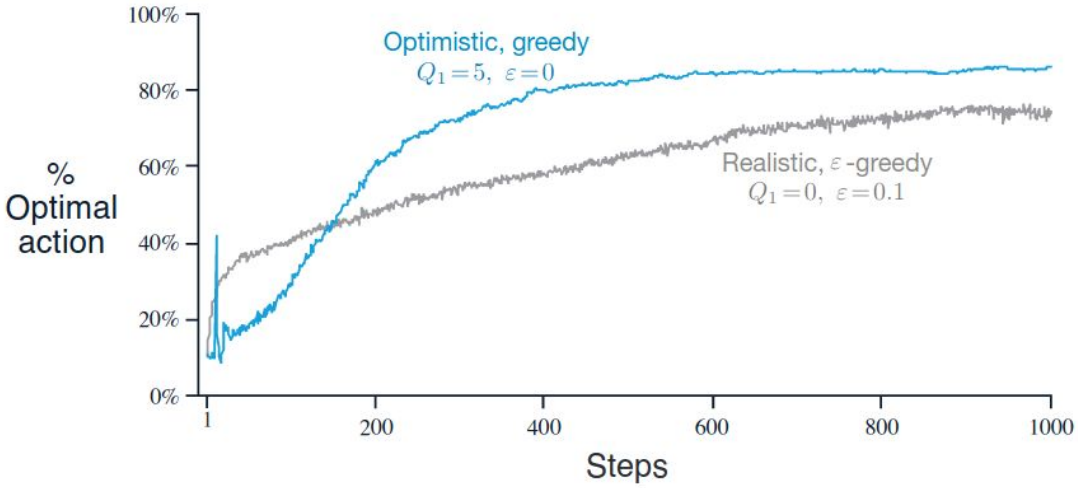


0.5

0.625

Exploration vs. Exploitation Tradeoff

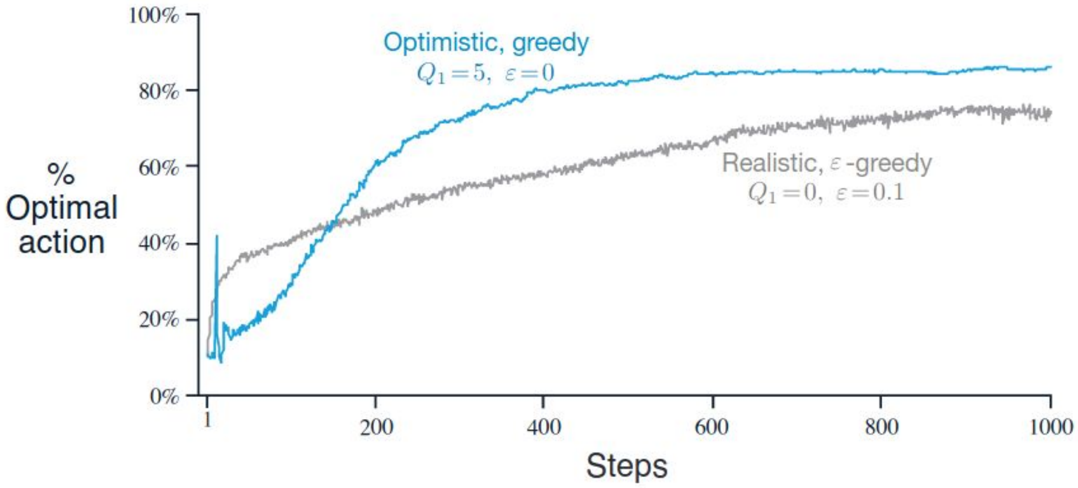
Optimistic Initial Values



The effect of optimistic initial action-value estimates on the 10-amed testbed. Both methods used a constant step-size parameter, alpha = 0.1

Exploration vs. Exploitation Tradeoff

Optimistic Initial Values



The effect of optimistic initial action-value estimates on the 10-armed testbed. Both methods used a constant step-size parameter, alpha = 0.1

- In early learning, the optimistic agent performs worse because it explores more.
- Its exploration decreases with time, because the optimism and its estimates washes out with more samples.

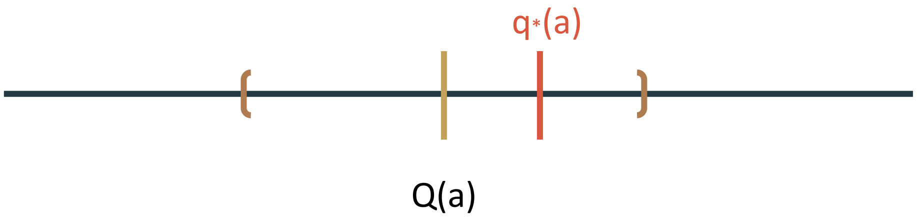
Exploration vs. Exploitation Tradeoff

Upper-Confidence-Bound Action Selection

$$A_t \leftarrow \begin{cases} \underset{a}{\operatorname{argmax}}\ Q_t(a) & \textbf{with probability } 1-\epsilon \\ a \sim Uniform(\{a_1 \ldots a_k\}) & \textbf{with probability } \epsilon \end{cases}$$

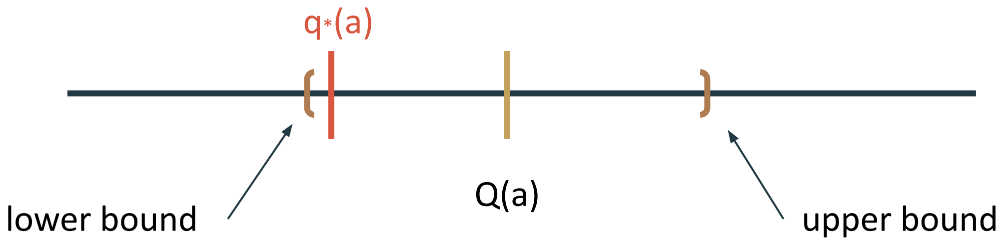
Exploration vs. Exploitation Tradeoff

Upper-Confidence-Bound Action Selection



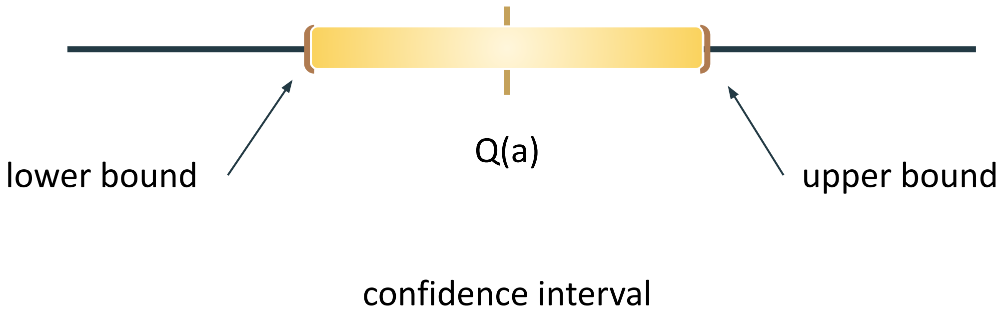
Exploration vs. Exploitation Tradeoff

Upper-Confidence-Bound Action Selection



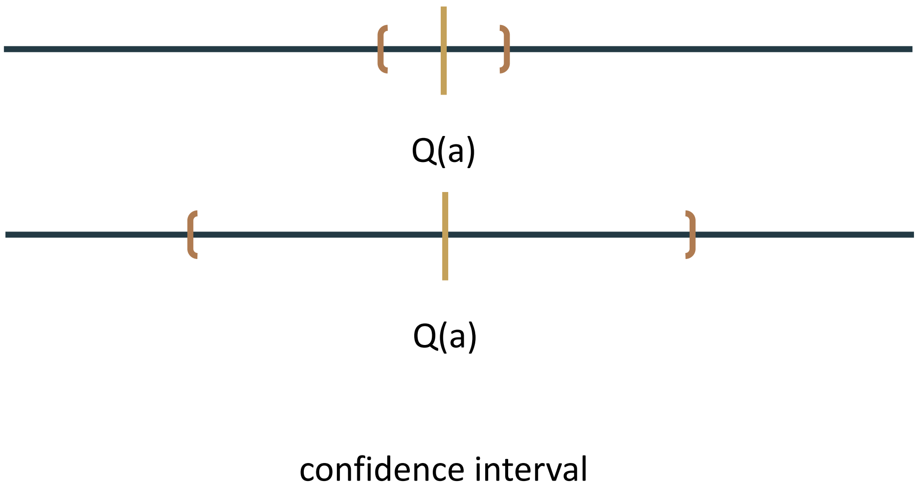
Exploration vs. Exploitation Tradeoff

Upper-Confidence-Bound Action Selection



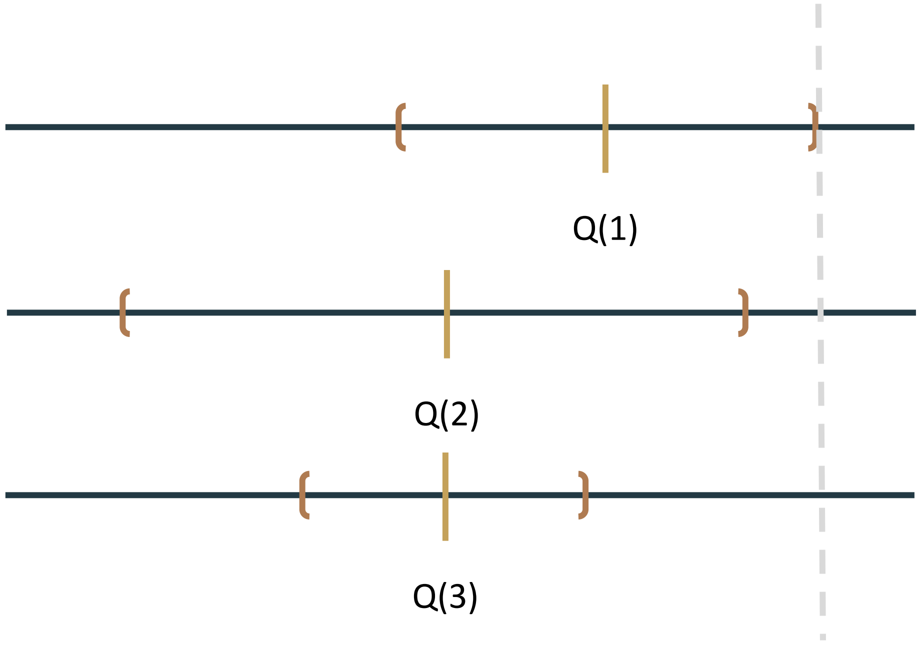
Exploration vs. Exploitation Tradeoff

Upper-Confidence-Bound Action Selection



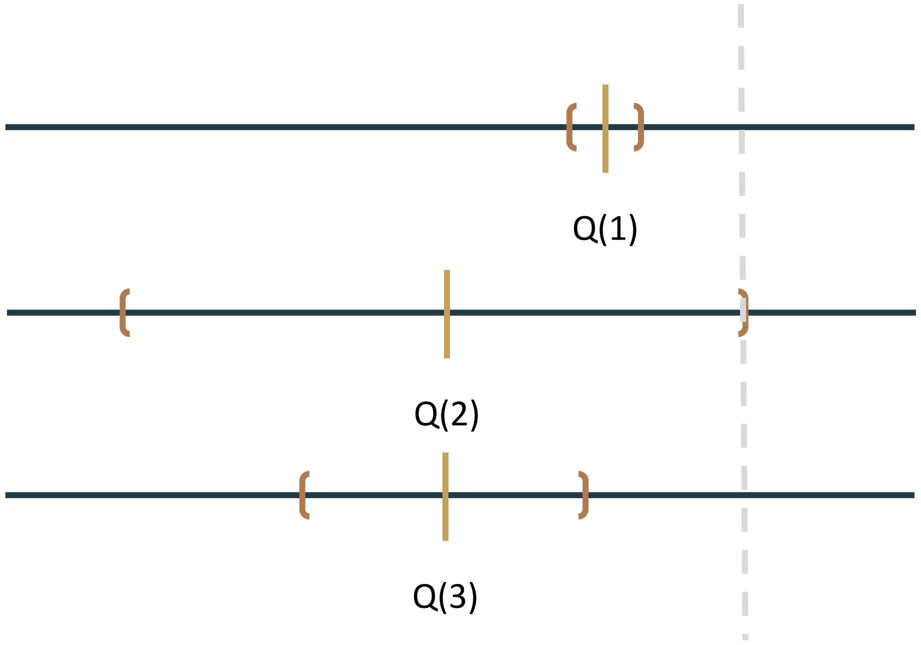
Exploration vs. Exploitation Tradeoff

Optimism in the Face of Uncertainty



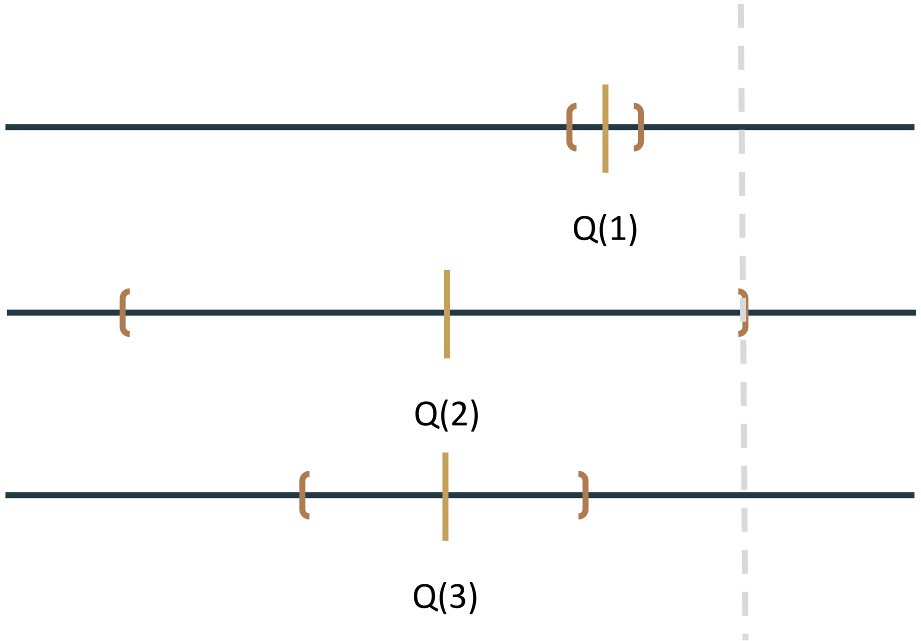
Exploration vs. Exploitation Tradeoff

Optimism in the Face of Uncertainty



Exploration vs. Exploitation Tradeoff

Optimism in the Face of Uncertainty



- This simply means that if we are uncertain about something, we should optimistically assume that it is good.
- Here, our agent has no idea which is best. So it **optimistically** picks the action that has the highest upper bound.
- This makes sense because either it does have the highest value and we get good reward, or by taking it we get to learn about an action.

Exploration vs. Exploitation Tradeoff

Upper - Confidence Bound (UCB) Action Selection

$$A_t \doteq \underset{a}{\arg\max} \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right]$$

- We can use upper-confidence bounds to select actions using the following formula;
- We will select the action that has the highest estimated value + our upper-confidence bound exploration term.
- The upper-bound term can be broken into three parts
- The C parameter as a user-specified parameter that controls the amount of exploration.
- UCB combines exploration and exploitation.
- The first term in the sum represents the **exploitation** part, and the second term represents the **exploration** part.

-

Exploration vs. Exploitation Tradeoff

Upper - Confidence Bound (UCB) Action Selection

$$A_t \doteq \underset{a}{\arg\max} \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right]$$

$$c\sqrt{\frac{ln\ t}{N_t(a)}} \longrightarrow c\sqrt{\frac{ln \text{ timesteps}}{\text{times action } a \text{ taken}}}$$

$$c\sqrt{\frac{ln\ 10000}{5000}} \longrightarrow 0.043c$$

$$c\sqrt{\frac{ln\ 10000}{100}} \longrightarrow 0.303c$$

Overview - Part 2

Introduction to Markov Decision Processes

Goal of Reinforcement Learning

Continuing Tasks

Introduction to Markov Decision Processes

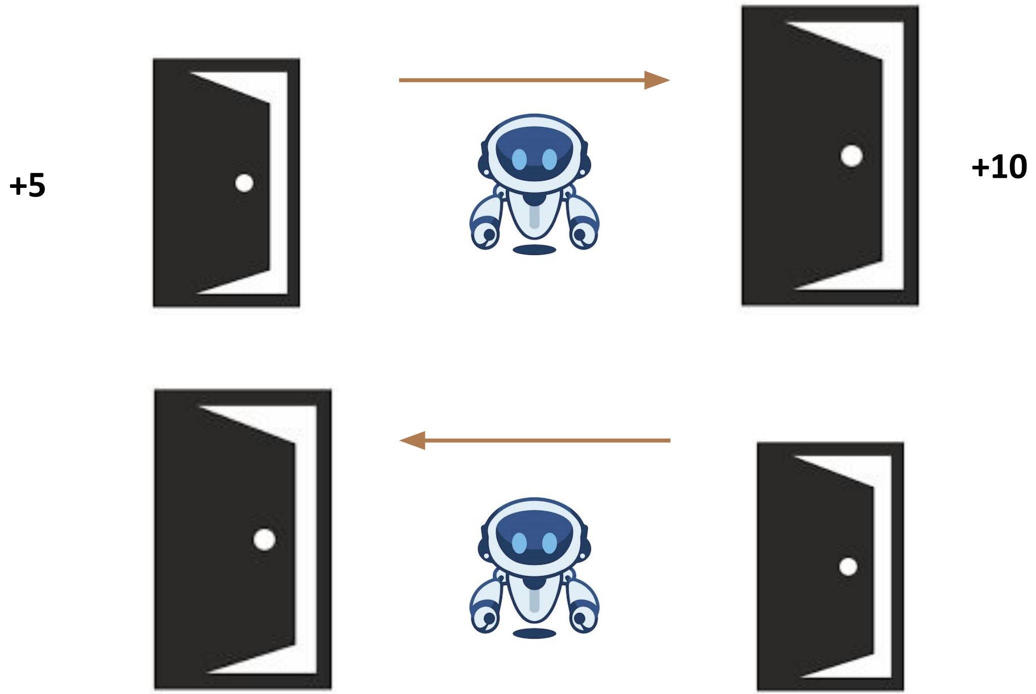
k-Armed Bandit problem :

- The agent is presented with the same situation and each time and the same action is always optimal.
- In many problems, **different situations call for different responses**. The actions we choose now affect the amount of reward we can get into the future.

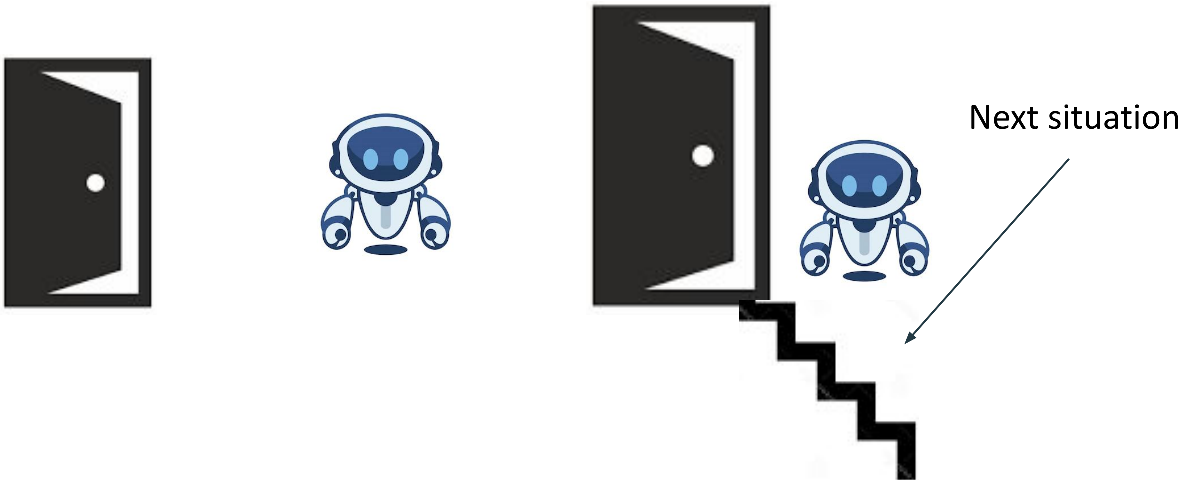
↓

Markov Decision Process

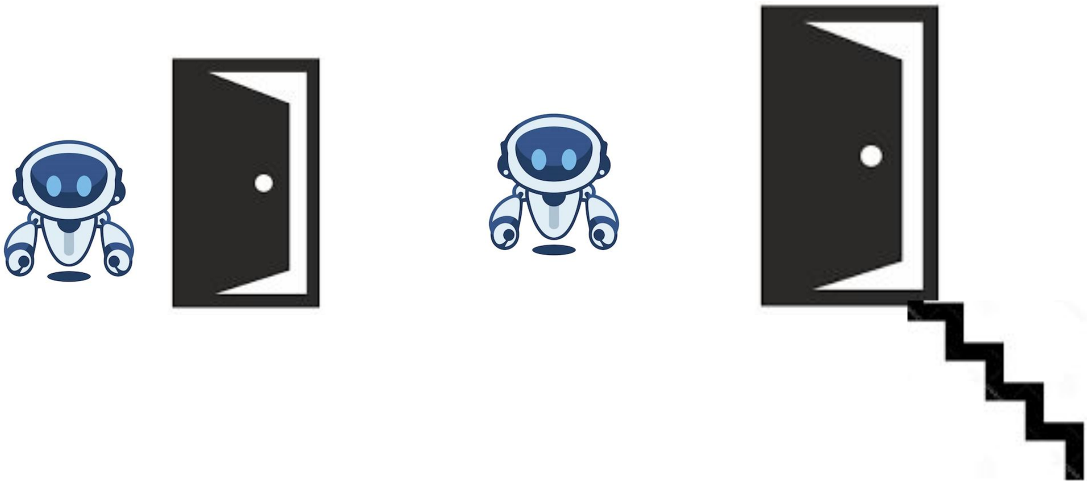
Introduction to Markov Decision Processes



Introduction to Markov Decision Processes

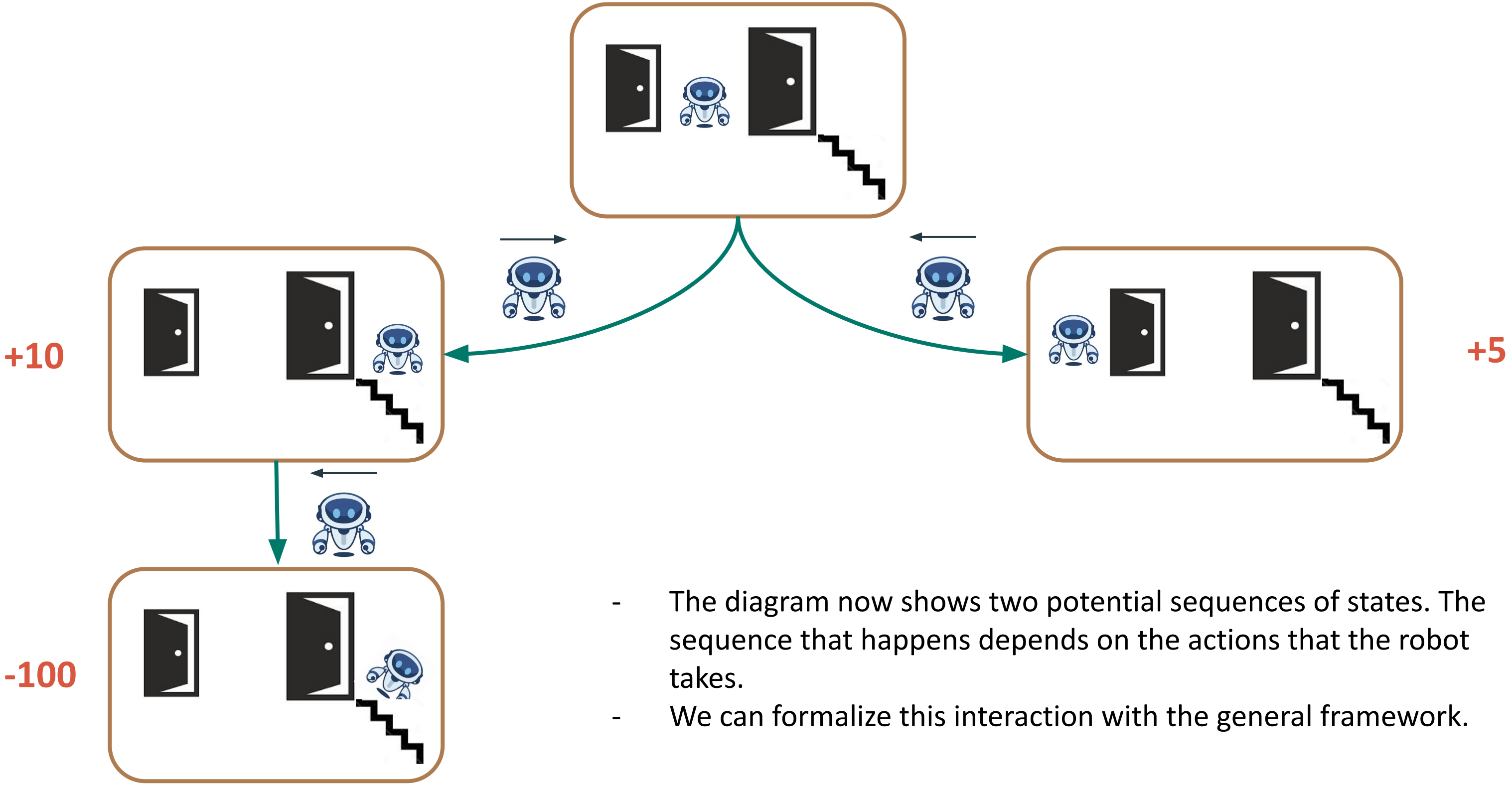


Introduction to Markov Decision Processes



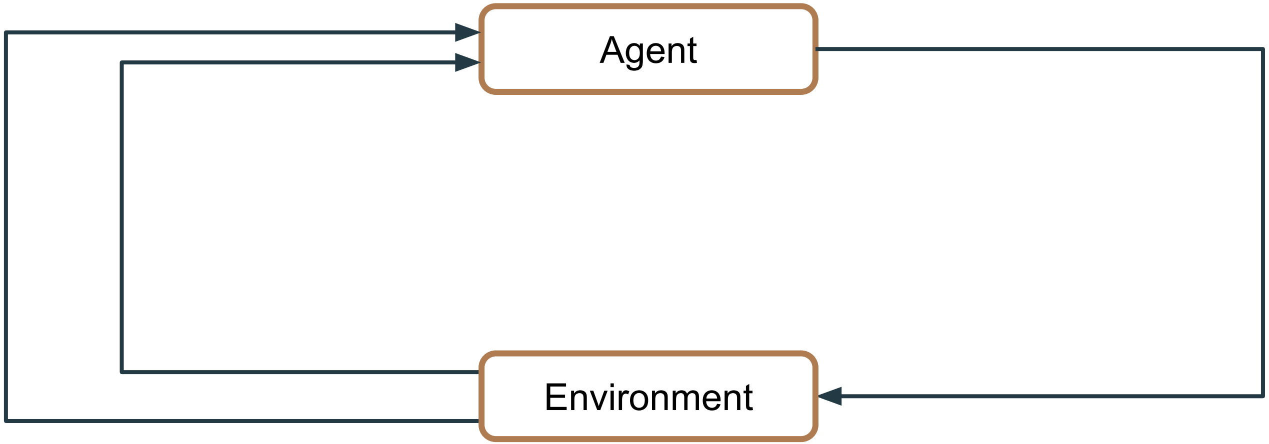
- A bandit robot would only be concerned about immediate reward and so it would go for the largest door.
- But a better decision can be made by considering the **long-term impact** of our decisions.

Introduction to Markov Decision Processes



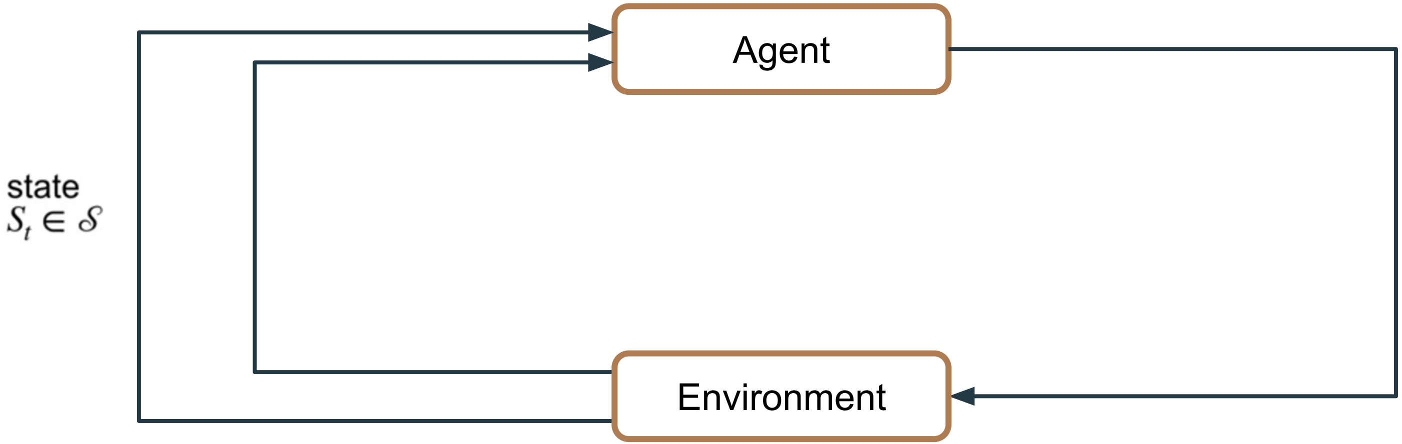
- The diagram now shows two potential sequences of states. The sequence that happens depends on the actions that the robot takes.
- We can formalize this interaction with the general framework.

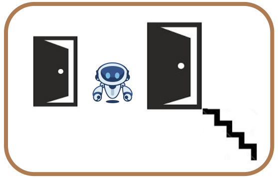
Introduction to Markov Decision Processes



- The **agent** and **environment** interact at **discrete** time steps

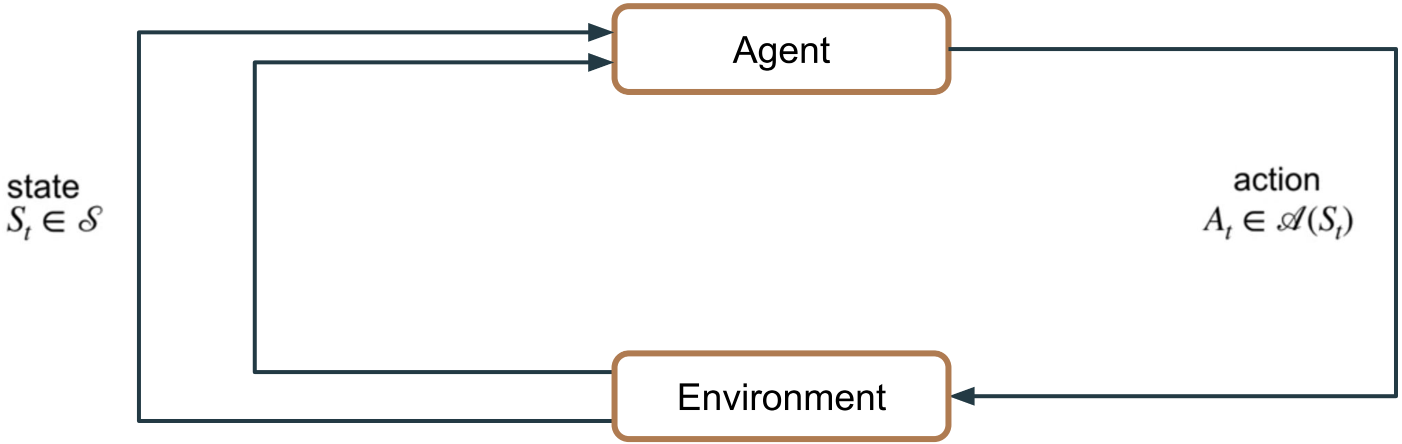
Introduction to Markov Decision Processes

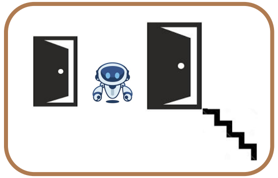




At each time, the agent receives a state St from the environment from a set of possible states.

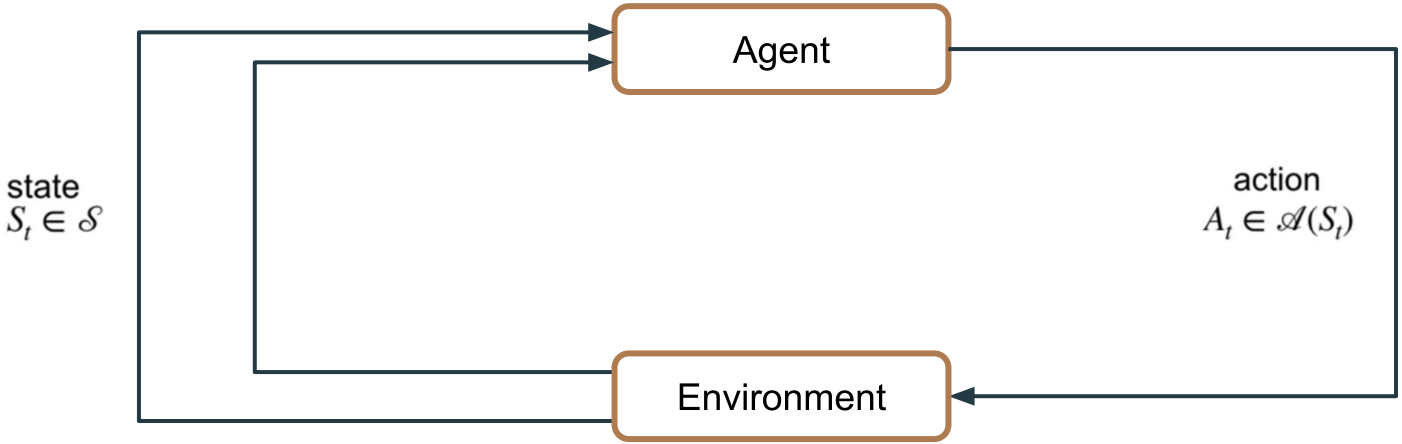
Introduction to Markov Decision Processes

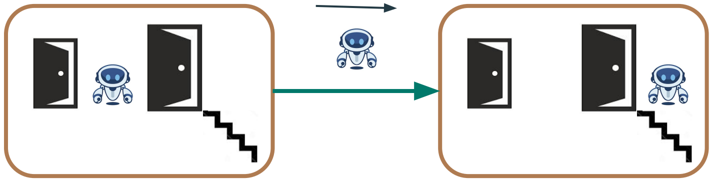




Based on this state the agent selects an action At from a set of possible actions. Script A of St is the set of valid actions in State St.

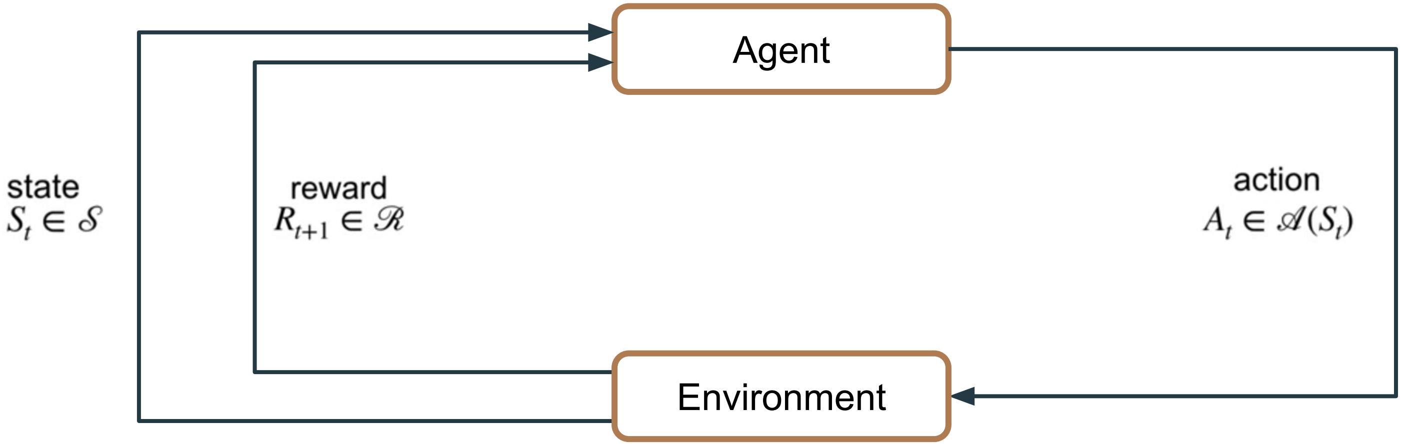
Introduction to Markov Decision Processes

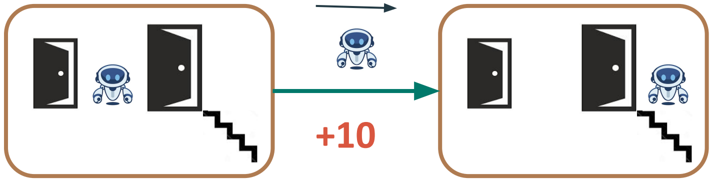




Moving right is an example of an action. One time step later based in part on the agent's action, the agent finds itself in a new state S(t +1).

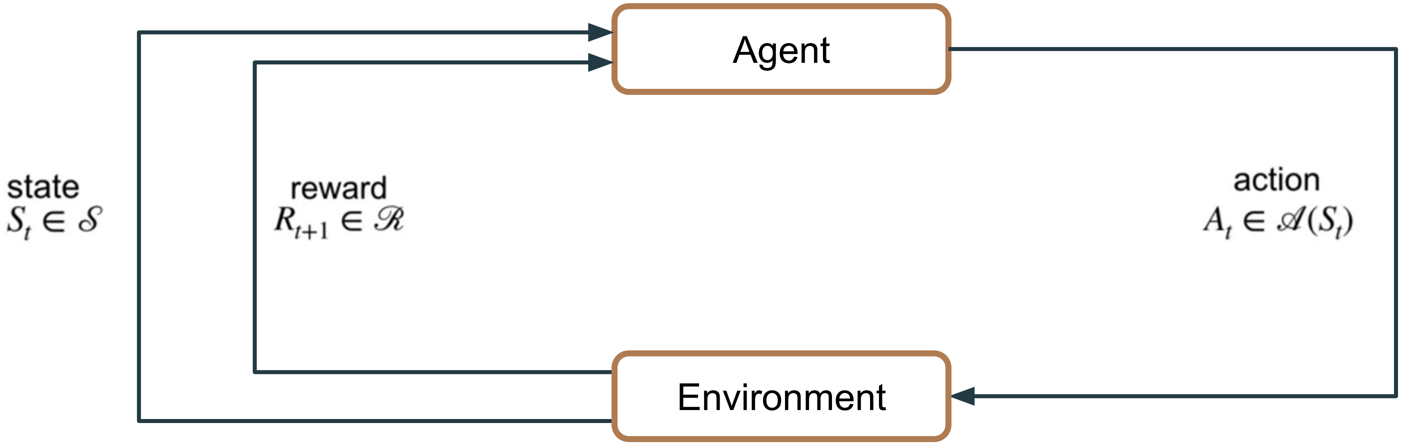
Introduction to Markov Decision Processes

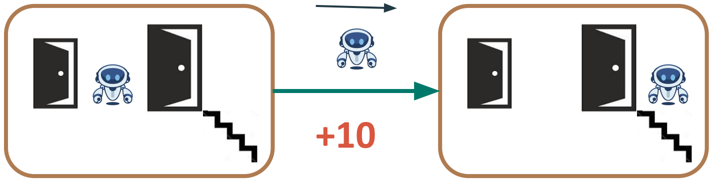




The environment also provides a scalar reward R(t+1) drawn from a set of possible rewards.

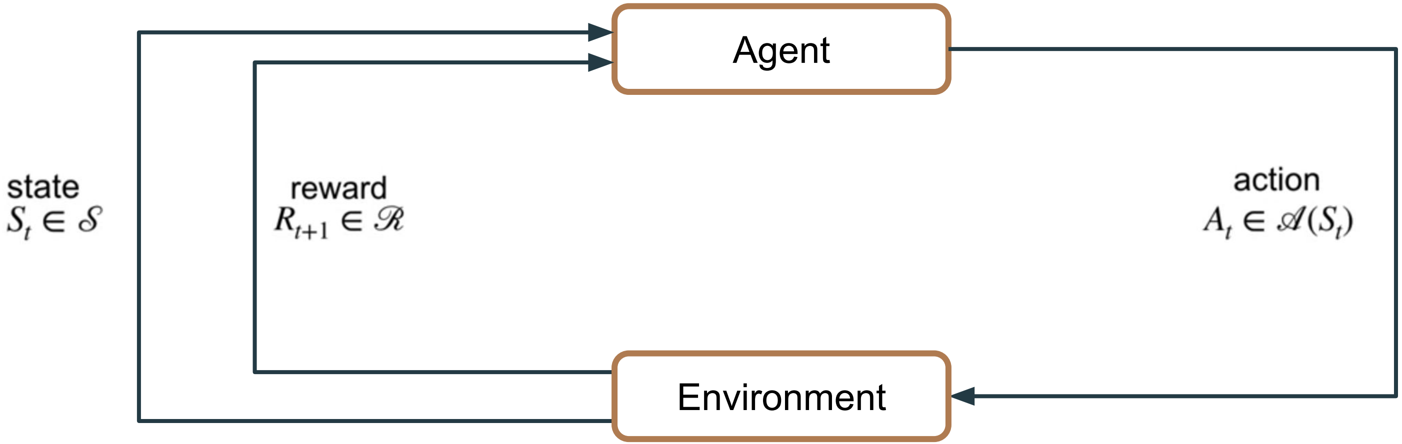
Introduction to Markov Decision Processes

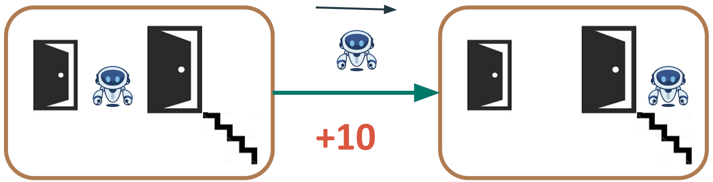


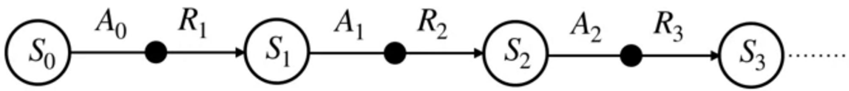


Agent environment interaction in the MDP framework.

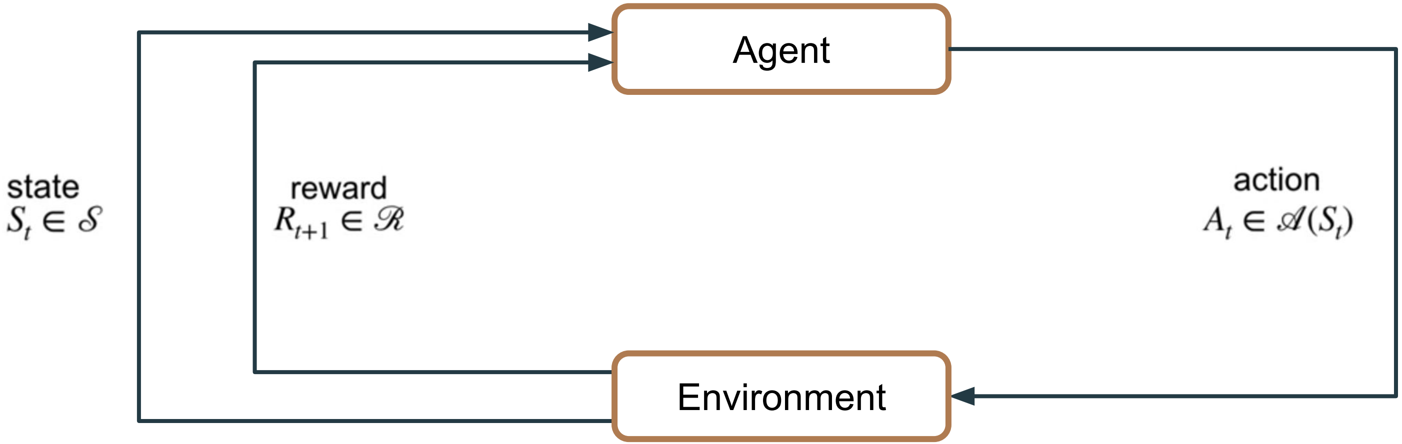
Introduction to Markov Decision Processes



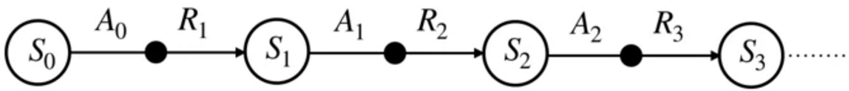




Introduction to Markov Decision Processes

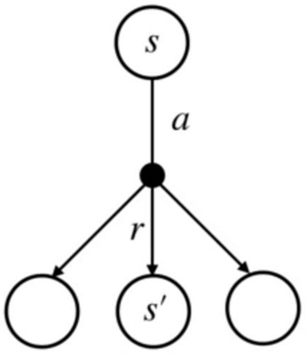


The agent-environment interaction generates a trajectory of experience consisting of **states**, **actions**, and **rewards**.



Introduction to Markov Decision Processes

Dynamics of MDP



$$p(s', r \mid s, a)$$

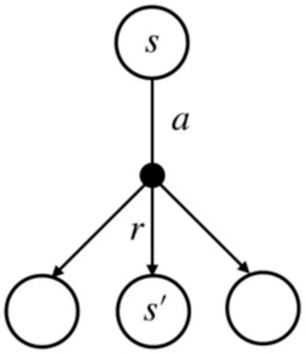
$$p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$$

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) = 1, \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$$

As in bandits, the outcomes are **stochastic**
When the agent takes an action in a state, there are many possible next states and rewards.
The transition dynamics function P, formalizes this notion

Introduction to Markov Decision Processes

Dynamics of MDP



$$p(s', r \mid s, a)$$

$$p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$$

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) = 1, \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$$

Markov property: Future state and reward only depends on the **current state and action**.
The present state is sufficient and remembering earlier states would not improve predictions about the future.

Overview - Part 2

Introduction to Markov Decision Processes

$\boxed{\textbf{Goal of Reinforcement Learning}}$

Continuing Tasks

Goal of Reinforcement Learning

- Agents have long-term goals
- **Goal of an agent: Formal definition**

The return at time step t, is the sum of rewards obtained after time step t.

return $\longrightarrow$ $$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \ldots$$

Random variable because the dynamics of the MDP can be stochastic.

Maximize the expected return

$$\mathbb{E}[G_t] = \mathbb{E}[\, R_{t+1} + R_{t+2} + R_{t+3} + \ldots \,]$$

Goal of Reinforcement Learning

- Agents have long-term goals
- **Goal of an agent: Formal definition**

return $\longrightarrow$ $G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \ldots$

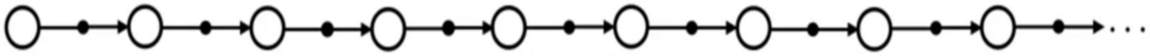
- For this to be well-defined, the sum of rewards must be **finite**.
- Specifically, final time step called capital T where the agent environment interaction ends.

$$\mathbb{E}[G_t] = \mathbb{E}[\, R_{t+1} + R_{t+2} + R_{t+3} + \ldots + \boxed{R_T}\,]$$

final time step

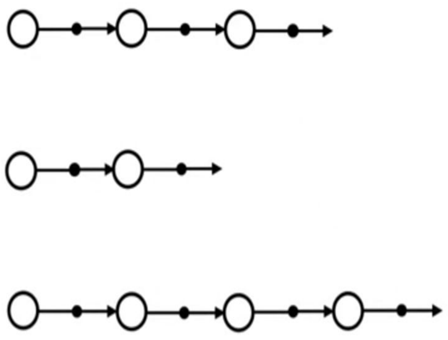
Goal of Reinforcement Learning

- What happens when the interaction ends? In the simplest case, the interaction naturally breaks into chunks called **episodes**.



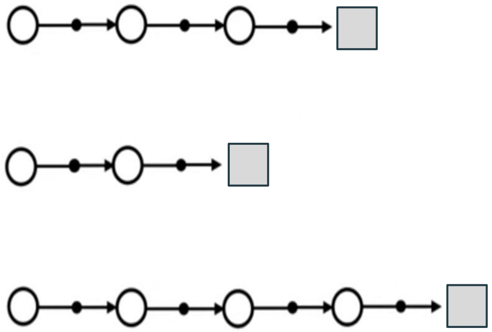
Goal of Reinforcement Learning

- What happens when the interaction ends? In the simplest case, the interaction naturally breaks into chunks called **episodes**.
- Each episode begins **independently** of how the previous one ended.



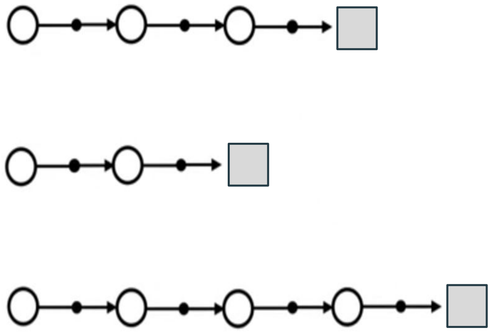
Goal of Reinforcement Learning

- What happens when the interaction ends? In the simplest case, the interaction naturally breaks into chunks called **episodes**.
- Each episode begins **independently** of how the previous one ended.
- At termination, the agent is reset to a start state.
- Every episode has a final state which we call the **terminal state**. We call these tasks **episodic tasks**.



Goal of Reinforcement Learning

- What happens when the interaction ends? In the simplest case, the interaction naturally breaks into chunks called **episodes**.
- Each episode begins **independently** of how the previous one ended.
- At termination, the agent is reset to a start state.
- Every episode has a final state which we call the **terminal state**. We call these tasks **episodic tasks**.



Overview - Part 2

Introduction to Markov Decision Processes

Goal of Reinforcement Learning

Continuing Tasks

Continuing Tasks

EPISODIC TASKS:

- Interaction breaks naturally into episodes.
- Every episode in an episodic task must end in a terminal state.
- Episodes are independent.

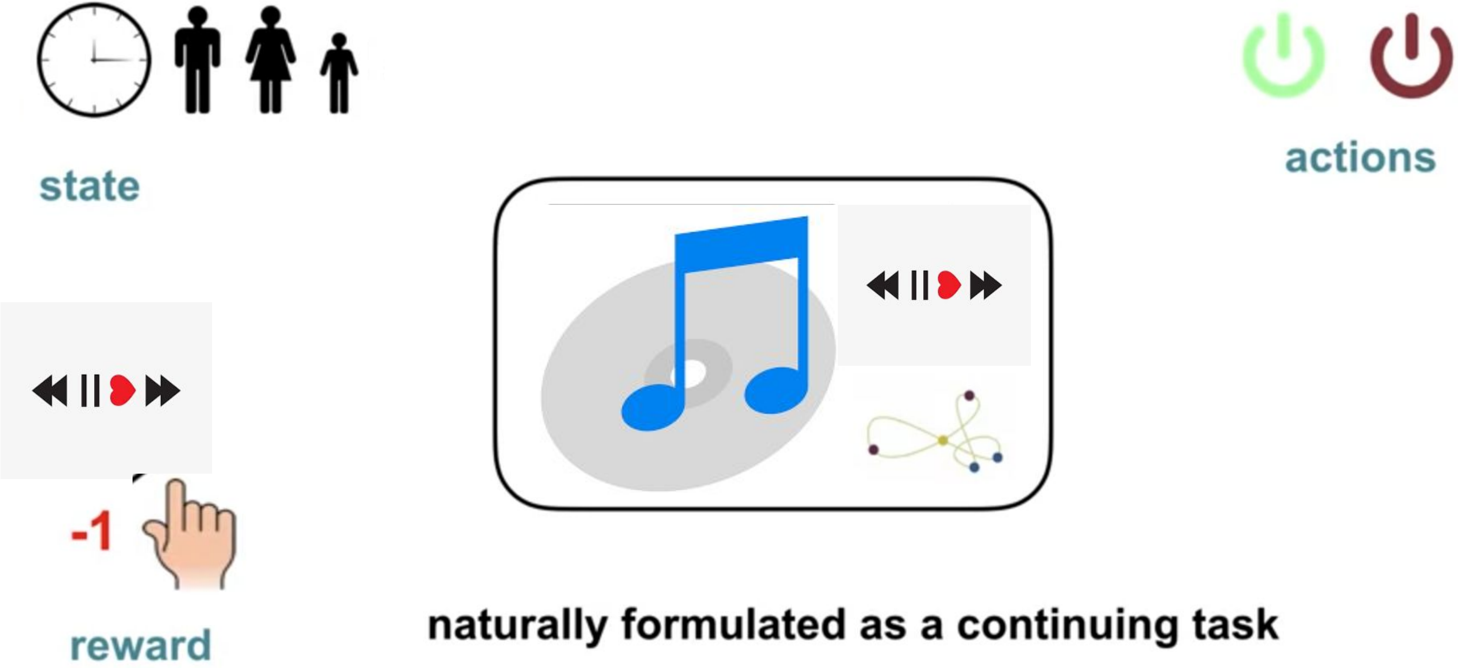
$$\mathbb{E}[G_t] = \mathbb{E}[\, R_{t+1} + R_{t+2} + R_{t+3} + \ldots + \boxed{R_T}\,]$$

final time step

CONTINUING TASKS

- Interaction goes continually (cannot be broken up into independent episodes).
- No terminal states.

Continuing Tasks



naturally formulated as a continuing task

Continuing Tasks

EPISODIC TASKS:

- Interaction breaks naturally into episodes.
- Every episode in an episodic task must end in a terminal state.
- Episodes are independent.

$$\mathbb{E}[G_t] = \mathbb{E}[\ R_{t+1} + R_{t+2} + R_{t+3} + \dots + \boxed{R_T}\ ]$$

final time step

CONTINUING TASKS

- Interaction goes continually (cannot be broken up into independent episodes).
- No terminal states.

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \dots \quad = \infty ?$$

Continuing Tasks

CONTINUING TASKS

- Interaction goes continually (cannot be broken up into independent episodes).
- No terminal states.

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \ldots + R_{t+k} + \ldots$$

How to make sure G_t is finite?

Continuing Tasks

CONTINUING TASKS

- Interaction goes continually (cannot be broken up into independent episodes).
- No terminal states.

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \ldots + R_{t+k} + \ldots$$

How to make sure G_t **is finite?**

Discount rate:

- $0 \leq \gamma < 1$
- Immediate rewards contribute more to the some. Rewards far into the future contribute less because they are multiplied by Gamma.

Continuing Tasks

CONTINUING TASKS

- Interaction goes continually (cannot be broken up into independent episodes).
- No terminal states.

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \ldots + R_{t+k} + \ldots$$

How to make sure G_t **is finite?**

Discount rate:

$$\boxed{G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{k-1} R_{t+k} + \ldots}$$

- $0 \leq \gamma < 1$
- Immediate rewards contribute more to the some. Rewards far into the future contribute less because they are multiplied by Gamma.

Continuing Tasks

How to make sure G_t **is finite?**

Discount rate:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{k-1} R_{t+k} + \ldots$$

- $0 \leq \gamma < 1$
- Immediate rewards contribute more to the some. Rewards far into the future contribute less because they are multiplied by Gamma.

Continuing Tasks

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{k-1} R_{t+k} + \ldots$$

How to make sure G_t is finite?

Discount rate:

- $0 \leq \gamma < 1$
- Immediate rewards contribute more to the some. Rewards far into the future contribute less because they are multiplied by Gamma.

Continuing Tasks

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{k-1} R_{t+k} + \ldots$$

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Finite as long as $0 \leq \gamma < 1$

How to make sure G_t **is finite?**

Discount rate:

- $0 \leq \gamma < 1$
- Immediate rewards contribute more to the some. Rewards far into the future contribute less because they are multiplied by Gamma.

Continuing Tasks

Effect of γ on agent behavior

$$\boxed{G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{k-1} R_{t+k} + \ldots}$$

If $\gamma = 0$

$$= R_{t+1} + 0 R_{t+2} + 0^2 R_{t+3} + \ldots + 0^{k-1} R_{t+k} + \ldots$$
$$= R_{t+1}$$

The agent is shortsighted and only cares about immediate expected reward:
Short-sighted agent

Continuing Tasks

Effect of γ on agent behavior

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{k-1} R_{t+k} + \ldots$$

If $\gamma \longrightarrow 1$

The immediate and future rewards are weighted nearly equally in the return. The agent in this case is more **far-sighted Agent.**

Continuing Tasks

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \ \dots$$

Continuing Tasks

$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \ \dots \\ &= R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \ \dots \right) \end{aligned}$$

This is just G_{t+1}

Continuing Tasks

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \ \dots$$

$$= R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \ \dots)$$

$$G_t = R_{t+1} + \gamma G_{t+1}$$

This is just G_{t+1}

Overview - Part 3

Policies and Value Functions

Bellman Equations

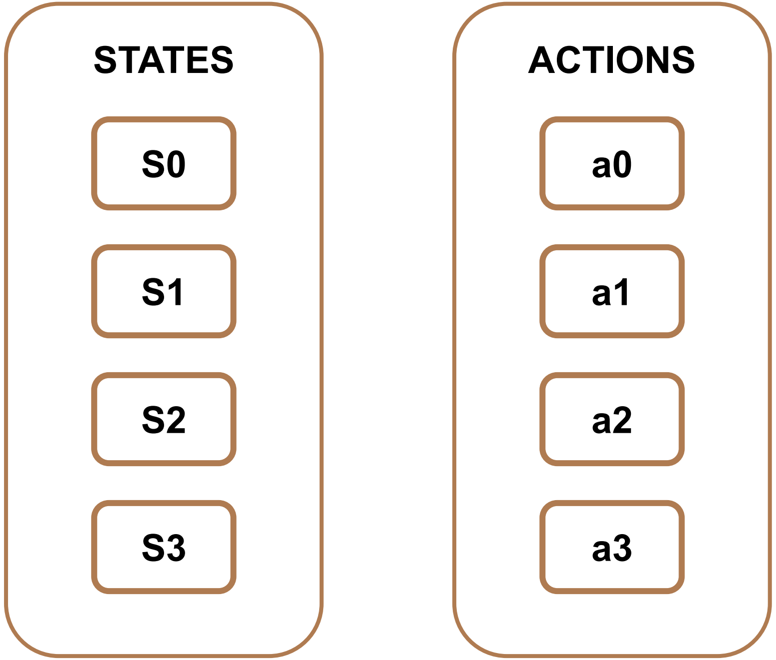
Optimality (Optimal Policies & Value Functions)

Policies and Value Functions

Deterministic Policy Notation $\pi(s) = a$ ← Action selected in state s by the policy Pi.

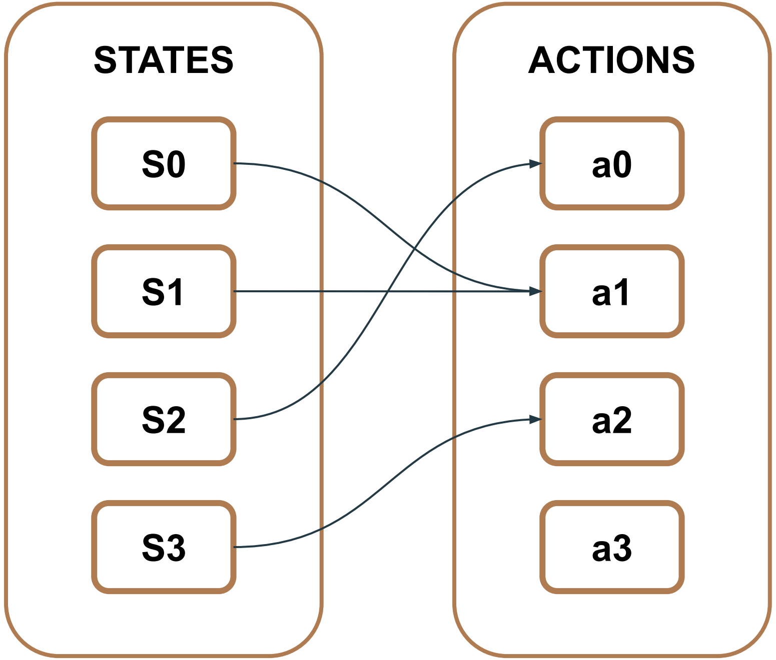
Policies and Value Functions

Deterministic Policy Notation $\pi(s) = a$ ← Action selected in state s by the policy Pi.



Policies and Value Functions

Deterministic Policy Notation $\pi(s) = a$ ← Action selected in state s by the policy Pi.



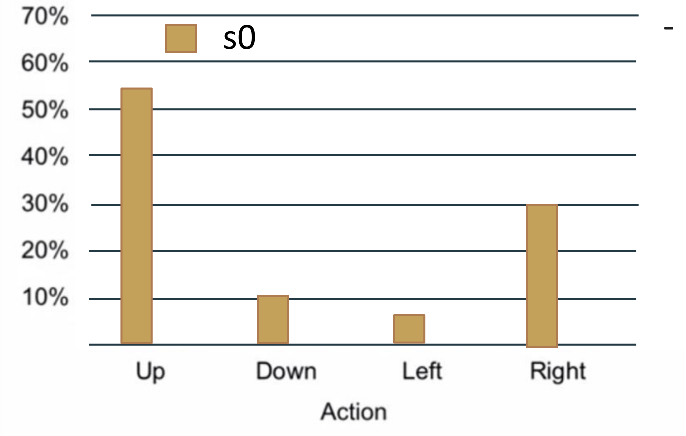
STATE	ACTION
S0	a1
S1	a1
S2	a0
S3	a2

Policies and Value Functions

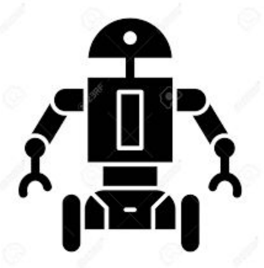
Stochastic Policy Notation

$$\pi(a \mid s)$$

- Probability of selecting action a in a state s.
- **Stochastic policy:** multiple actions may be selected with non-zero probability.



- Distribution over actions for state s0 according to π.

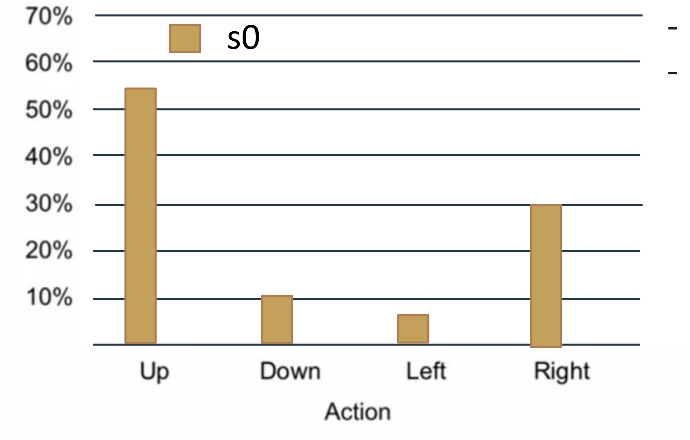


Policies and Value Functions

Stochastic Policy Notation

$$\pi(a \mid s)$$

- Probability of selecting action a in a state s.
- **Stochastic policy:** multiple actions may be selected with non-zero probability.



- Distribution over actions for state s0 according to π.
- π specifies a separate distribution over actions for each state.

$$\sum_{a \in \mathcal{A}(s)} \pi(a \mid s) = 1$$

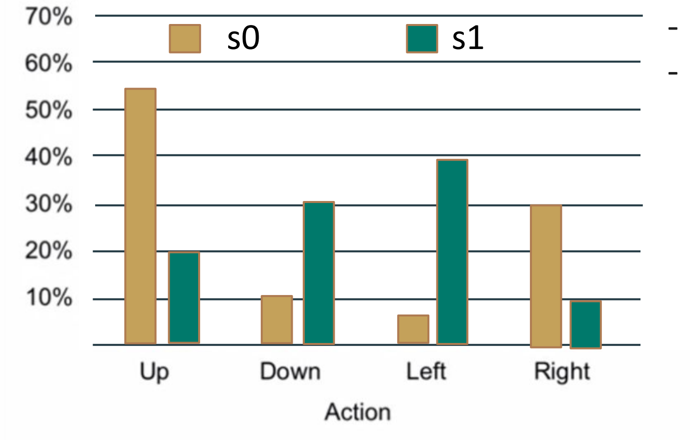
$$\pi(a \mid s) \geq 0$$

Policies and Value Functions

Stochastic Policy Notation

$$\pi(a \mid s)$$

- Probability of selecting action a in a state s.
- **Stochastic policy:** multiple actions may be selected with non-zero probability.



- Distribution over actions for state s0 according to π.
- π specifies a separate distribution over actions for each state.

$$\sum_{a \in \mathcal{A}(s)} \pi(a \mid s) = 1$$

$$\pi(a \mid s) \geq 0$$

Policies and Value Functions

Important!! Policies

The most important things to remember:

- Agent's behavior is specified by a policy that maps the state to a probability distribution over actions
- The policy can depend only on the current state, and not other things like time or previous states.

Policies and Value Functions

Recall that

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

Value functions

The objective, is to learn a **policy** that achieves the **most reward in the long run**.

$$v(s) \doteq \mathbb{E}\left[G_t \mid S_t = s\right]$$

- **State value function:** future award an agent can expect to receive starting from a particular state.
- The state value function is the expected return from a given state.
- The agent's behavior will also determine how much total reward it can expect.

Policies and Value Functions

Recall that

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

Value functions

The objective, is to learn a **policy** that achieves the **most reward in the long run**.

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s]$$

- **State value function:** future award an agent can expect to receive starting from a particular state.
- The state value function is the expected return from a given state.
- The agent's behavior will also determine how much total reward it can expect.
- A value function is defined with respect to a given policy.
- The subscript π indicates the value function is contingent on the agent selecting actions according to π.

Policies and Value Functions

Recall that

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

Action- Value functions

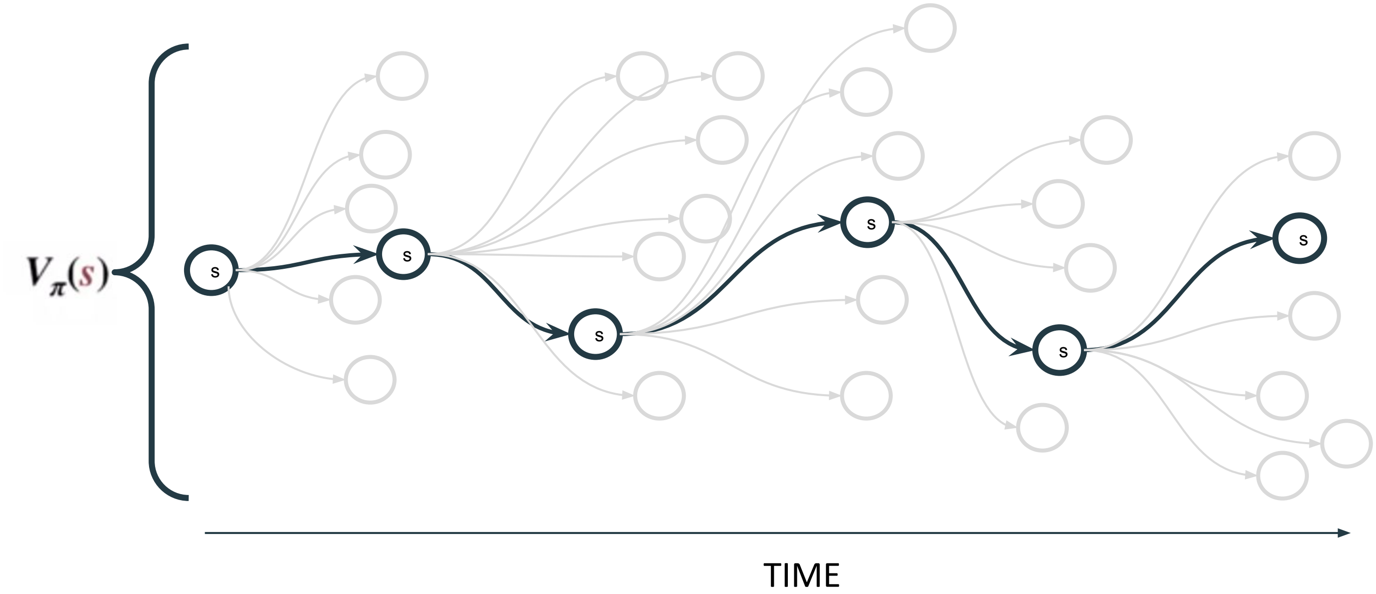
$$q_\pi(s,a) \doteq \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right]$$

- An **action value** describes what happens when the agent first selects a particular action.
- The action value of a state is the expected return if the agent selects action ***a*** and then follows policy **π**.
- Value functions are crucial in reinforce learning, they allow an agent to query the quality of its current situation instead of waiting to observe the long-term outcome.

Policies and Value Functions

Action- Value functions

Value functions predict rewards into the future



Policies and Value Functions

Action- Value functions

Value functions predict rewards into the future

- The return is not immediately available
- The return may be random due to stochasticity in both the policy and environment dynamics.
- The value function summarizes all the possible futures by averaging over returns.
- Value function enable us to judge the quality of different policies.

Policies and Value Functions

Action- Value functions : Example

Reward: + 1 if winning
0 if otherwise

- Chess has an episodic MDP.
- The **state** is given by the positions of all the pieces on the board, the **actions** are the legal moves, and **termination** occurs when the game ends in either a win, loss, or draw.

Policies and Value Functions

Action- Value functions : Example

Reward: + 1 if winning
0 if otherwise

- Chess has an episodic MDP.
- The **state** is given by the positions of all the pieces on the board, the **actions** are the legal moves, and **termination** occurs when the game ends in either a win, loss, or draw.
- This reward does not tell us much about how well the agent is playing during the match, we'll have to wait until the end of the game to see any non-zero reward.

Policies and Value Functions

Action- Value functions : Example

$$P(\text{win}) = V_\pi(s)$$

s

Policies and Value Functions

Recall that

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

Action- Value functions : Example

s

$$P(\text{win}) = V_\pi(s)$$

$$v(s) \doteq \mathbb{E}\left[G_t \mid S_t = s\right]$$

- The value function tells us much more.
- The state value is equal to the expected sum of future rewards.
- Since the only possible non-zero reward is +1 for winning, the state value is simply the probability of winning if we follow the current policy Pi.

Policies and Value Functions

Recall that

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

Action- Value functions : Example

$$P(\text{win}) = V_\pi(s)$$

$$v\ (s) \doteq \mathbb{E}\ [G_t \mid S_t = s]$$

- In this two player game, the opponent's move is part of the state transition.
- New movements puts the board into a new state, S'.

Policies and Value Functions

Recall that

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

Action- Value functions : Example

$P(\text{win}) = V_\pi(s) = 0.34$

$V_\pi(s') = 0.31$

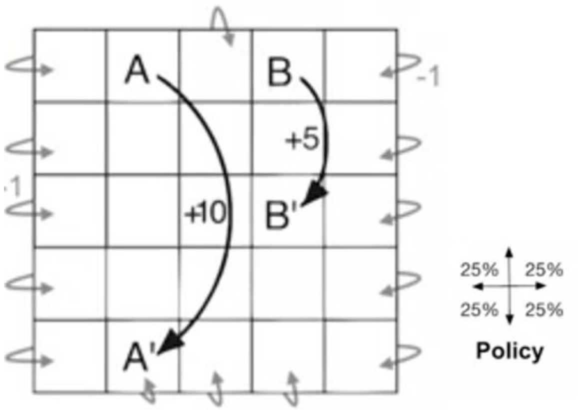
$$P(\text{win}) = V_\pi(s)$$

$$v\ (s) \doteq \mathbb{E}\ [G_t \mid S_t = s]$$

- Note, the value of state S’ is lower than the value of state S.
- We are less likely to win the game from this new state assuming we continue following policy Pi.
- An action value function would allow us to assess the probability of winning for each possible move given we follow the policy Pi for the rest of the game.

Policies and Value Functions

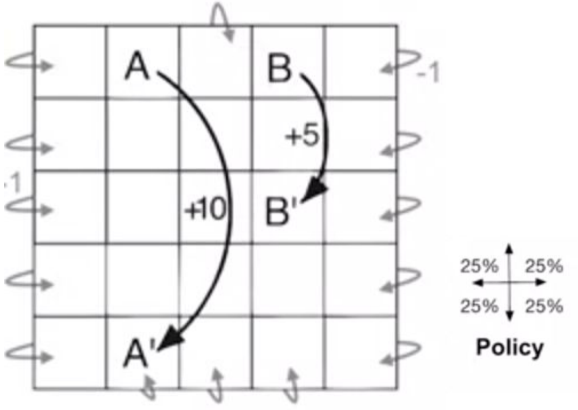
Action- Value functions : Example simple continuing MDP



- The states are defined by the locations on the grid, the actions move the agent up, down, left, or right.
- The agent cannot move off the grid and bumping generates a reward of -1.
- Most other actions yield no reward.

Policies and Value Functions

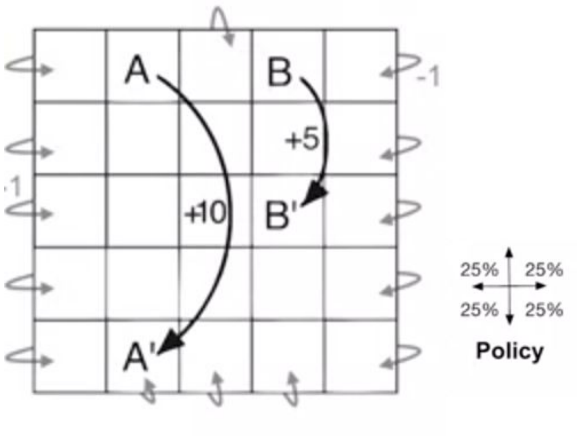
Action- Value functions : Example simple continuing MDP



- There are two special states: A and B
- Every action in state A yields + 10 reward and + five reward in state B.
- We must specify the policy before we can figure out what the value function is.

Policies and Value Functions

Action- Value functions : Example simple continuing MDP



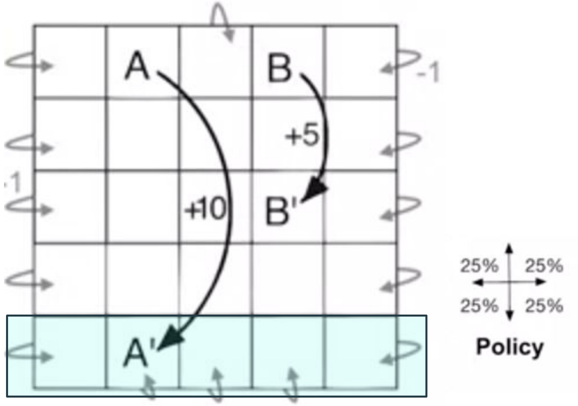
$\gamma = 0.9$

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

- Uniform random policy.
- Since this is a continuing task, we need to specify Gamma, let's go with 0.9.
- Later, we will learn several ways to compute and estimate the value function.

Policies and Value Functions

Action- Value functions : Example simple continuing MDP



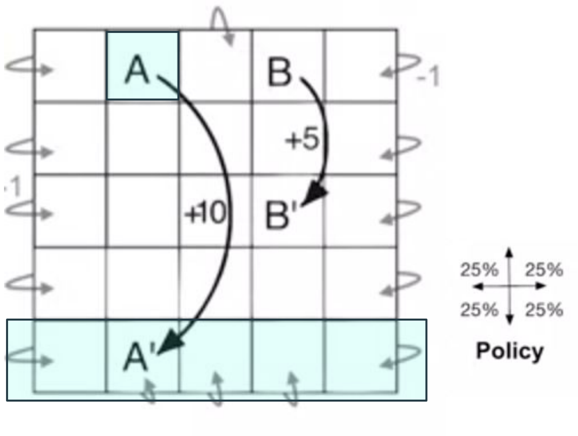
$\gamma = 0.9$

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

- Negative values near the bottom, these values are low because the agent is likely to bump into the wall before reaching the states A and B.
- A and B are both the only sources of positive reward in this MDP.

Policies and Value Functions

Action- Value functions : Example simple continuing MDP



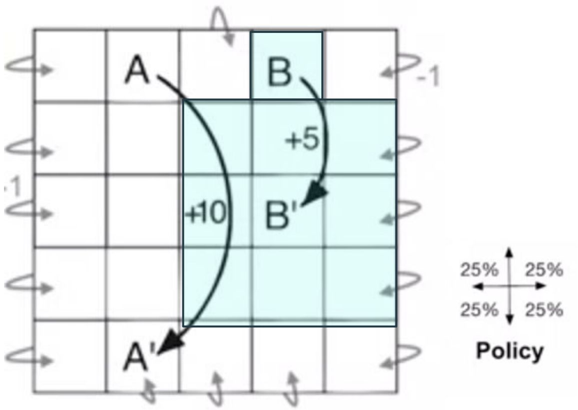
$\gamma = 0.9$

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

- State A has the highest value.
- The value is less than 10 even if every action from state A generates a reward of +10.
- Every transition from A moves the agent close to the lower wall and here the random policy is likely to bump and get negative reward. .

Policies and Value Functions

Action- Value functions : Example simple continuing MDP



$\gamma = 0.9$

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

- The value of state B is slightly greater than five.
- The transition from B moves the agent to the middle.
- In the middle, the agent is unlikely to bump and is close to the high-valued states A and B.

Overview - Part 3

Policies and Value Functions

Bellman Equations

Optimality (Optimal Policies & Value Functions)

Bellman Equations

In reinforcement learning we can relate the value of the current state to the value of future states without waiting to observe all the future rewards.

We use **Bellman equations** to formalize this connection between the value of a state and its possible successors.

Bellman Equations

Recall that

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

State - Value Bellman functions

The Bellman equation for the state value function defines a relationship between the value of a state and the value of his possible successor states.

$$v_\pi(s) \doteq \mathbb{E}_\pi \left[G_t \mid S_t = s \right]$$

Recall that the return is defined as the discounted sum of future rewards.

$$= \mathbb{E}_\pi \left[R_{t+1} + \gamma G_{t+1} \mid S_t = s \right]$$

Action choice depends only on the current state, while the next state and reward depend only on the current state and action.

$$= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a) \left[r + \gamma \mathbb{E}_\pi \left[G_{t+1} \mid S_{t+1} = s' \right] \right]$$

$$= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a) \left[r + \gamma \sum_{a'} \pi(a' \mid s') \sum_{s''} \sum_{r'} p(s'', r' \mid s', a') \left[r' + \gamma \mathbb{E}_\pi \left[G_{t+2} \mid S_{t+2} = s'' \right] \right] \right]$$

Bellman Equations

Recall that

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

State - Value Bellman functions

The Bellman equation for the state value function defines a relationship between the value of a state and the value of his possible successor states.

Recall that the return is defined as the discounted sum of future rewards.

$$v_\pi(s) \doteq \mathbb{E}_\pi \left[G_t \mid S_t = s \right]$$

$$= \mathbb{E}_\pi \left[R_{t+1} + \gamma G_{t+1} \mid S_t = s \right]$$

EXERCICE!!

$$= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a) \left[r + \gamma \mathbb{E}_\pi \left[G_{t+1} \mid S_{t+1} = s' \right] \right]$$

$$= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a) \left[r + \gamma \sum_{a'} \pi(a' \mid s') \sum_{s''} \sum_{r'} p(s'', r' \mid s', a') \left[r' + \gamma \mathbb{E}_\pi \left[G_{t+2} \mid S_{t+2} = s'' \right] \right] \right]$$

Bellman Equations

Recall that

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

State - Value Bellman functions

The Bellman equation for the state value function defines a relationship between the value of a state and the value of his possible successor states.

Recall that the return is defined as the discounted sum of future rewards.

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi\left[G_t \mid S_t = s\right] \\ &= \mathbb{E}_\pi\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\ &= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma \mathbb{E}_\pi\left[G_{t+1} \mid S_{t+1} = s'\right]\right] \\ &= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma \sum_{a'} \pi(a' \mid s') \sum_{s''} \sum_{r'} p(s'', r' \mid s', a')\left[r' + \gamma \mathbb{E}_\pi\left[G_{t+2} \mid S_{t+2} = s''\right]\right]\right] \end{aligned}$$

The expected return depends on states and rewards infinitely far into the future.

Bellman Equations

Recall that

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

State - Value Bellman functions

The Bellman equation for the state value function defines a relationship between the value of a state and the value of his possible successor states.

Recall that the return is defined as the discounted sum of future rewards.

$$v_\pi(s) \doteq \mathbb{E}_\pi \left[G_t \mid S_t = s \right]$$

$$= \mathbb{E}_\pi \left[R_{t+1} + \gamma G_{t+1} \mid S_t = s \right]$$

$$= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a) \left[r + \gamma \mathbb{E}_\pi \left[G_{t+1} \mid S_{t+1} = s' \right] \right]$$

$$= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a) \left[r + \gamma \sum_{a'} \pi(a' \mid s') \sum_{s''} \sum_{r'} p(s'', r' \mid s', a') \left[r' + \gamma \mathbb{E}_\pi \left[G_{t+2} \mid S_{t+2} = s'' \right] \right] \right]$$

Expected return is also the definition of the value function for state S’. The only difference is that the time index is t+1 instead of t. This is not an issue because neither the policy nor PI depends on time.

Bellman Equations

Recall that

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

State - Value Bellman functions

The Bellman equation for the state value function defines a relationship between the value of a state and the value of his possible successor states.

$$\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi \left[G_t \mid S_t = s \right] \\
&= \mathbb{E}_\pi \left[R_{t+1} + \gamma G_{t+1} \mid S_t = s \right] \\
&= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a) \left[r + \gamma \mathbb{E}_\pi \left[G_{t+1} \mid S_{t+1} = s' \right] \right] \\
&= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a) \left[r + \gamma v_\pi(s') \right]
\end{aligned}$$

Recall that the return is defined as the discounted sum of future rewards.

we can use them as a stand-in for the average of an infinite number of possible futures.

Bellman Equations

Recall that

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

Action - Value Bellman functions

$$q_\pi(s,a) \doteq \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right]$$

- It will be a recursive equation for the value of a state action pair in terms of its possible successors **state-action pairs**.
- The equation does not begin with the policy selecting an action. This is because the action is already fixed as part of the state-action pair. Instead.

Bellman Equations

Recall that

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

Action - Value Bellman functions

$$q_\pi(s,a) \doteq \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right]$$

$$= \sum_{s'} \sum_{r} p(s',r \mid s,a)\Big[r + \gamma \mathbb{E}_\pi\left[G_{t+1} \mid S_{t+1} = s'\right]\Big]$$

Bellman Equations

Recall that

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

Action - Value Bellman functions

$$q_\pi(s,a) \doteq \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right]$$

$$= \sum_{s'} \sum_{r} p(s',r \mid s,a) \Big[r + \gamma \mathbb{E}_\pi \left[G_{t+1} \mid S_{t+1} = s' \right] \Big]$$

$$= \sum_{s'} \sum_{r} p(s',r \mid s,a) \Big[r + \gamma \sum_{a'} \pi(a' \mid s') \mathbb{E}_\pi \left[G_{t+1} \mid S_{t+1} = s', A_{t+1} = a' \right] \Big]$$

Bellman Equations

Recall that

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

Action - Value Bellman functions

$$
\begin{aligned}
q_\pi(s,a) &\doteq \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right] \\
&= \sum_{s'}\sum_{r} p(s',r \mid s,a)\Big[r + \gamma \mathbb{E}_\pi\left[G_{t+1} \mid S_{t+1} = s'\right]\Big] \\
&= \sum_{s'}\sum_{r} p(s',r \mid s,a)\left[r + \gamma \sum_{a'} \pi(a' \mid s')\mathbb{E}_\pi\left[G_{t+1} \mid S_{t+1} = s', A_{t+1} = a'\right]\right] \\
&= \sum_{s'}\sum_{r} p(s',r \mid s,a)\left[r + \gamma \sum_{a'} \pi(a' \mid s') q_\pi(s',a')\right]
\end{aligned}
$$

Bellman Equations

Action - Value Bellman functions

Key ideas

- Again, we have a weighted sum over terms consisting of immediate reward plus expected future return given a specific next state little s prime.
- We want to recursive equation for the value of one state-action pair in terms of the next state-action pair.
- We have the expected return given only the next state. To change this, we can express the expected return from the next state as a sum of the agents possible action choices.
- So we have covered how to derive the Bellman equations for state and action value functions.
- These equations provide relationships between the values of a state or state-action pair and the possible next states or next state action pairs.
- **The Bellman equations capture an important structure of the reinforcement learning problem.**

Bellman Equations

Why Bellman equations

Next session on Learning

- Why Bellman equation?
- Optimal policies
- Policy Evaluation
- Policy Iteration
- Monte Carlo
- Temporal Difference Learning