Stockholms universitet

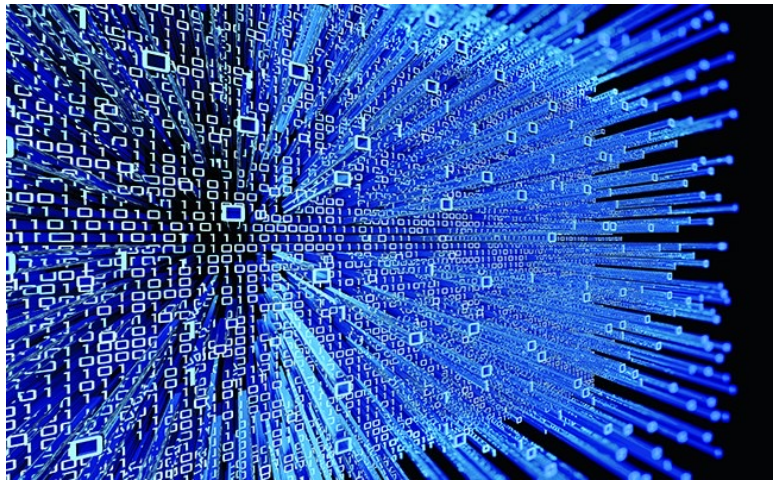
Reinforcement Learning

Data-Driven Decision Making

Machine Learning

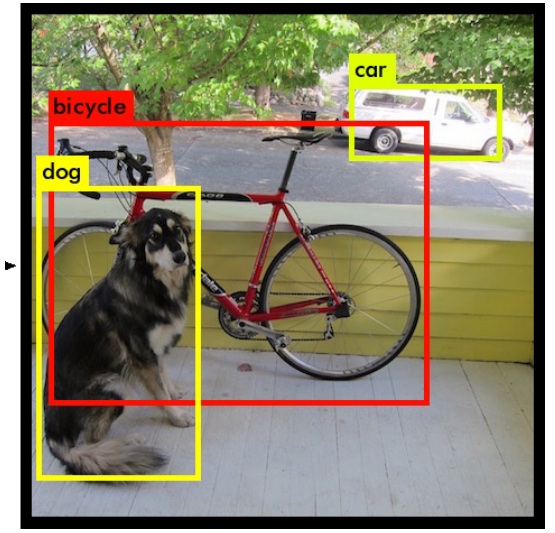
Sindri Magnússon

Machine Learning



Data

Training



Predictive model

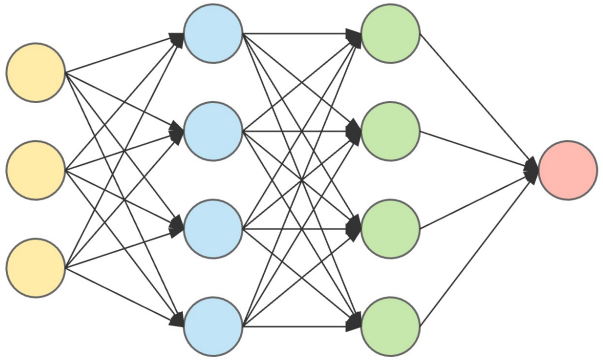
The study of how software agents learn to make good **predictions** or **decisions** from **experience** or **data**

Today: AI/Data-Driven Decision Making

How can a software agents learn to make good **decisions** from **experience** or **data**?

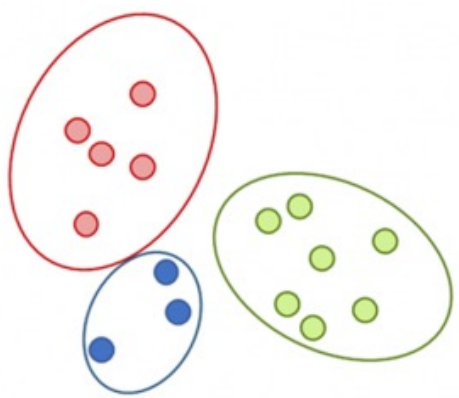
Main Branches of Machine Learning

Supervised learning



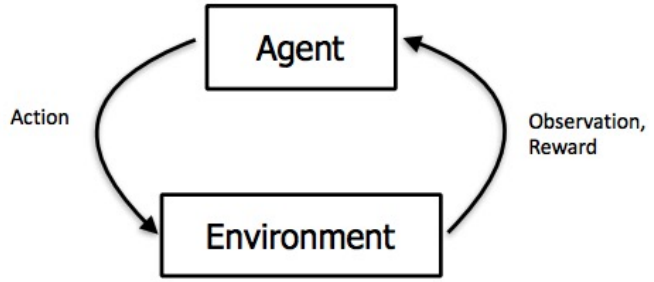
Predictions from labeled data

Unsupervised learning



Patterns in unlabeled data

Reinforcement learning

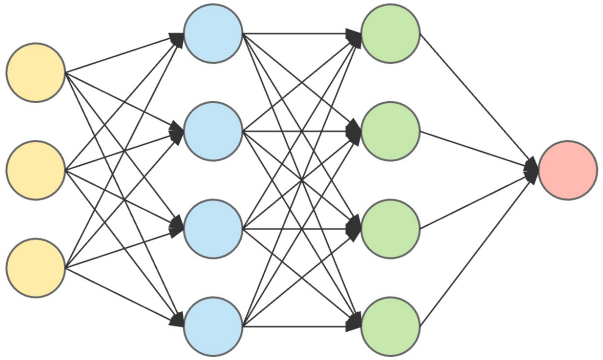


Optimal decisions from data and experience

Main Branches of Machine Learning

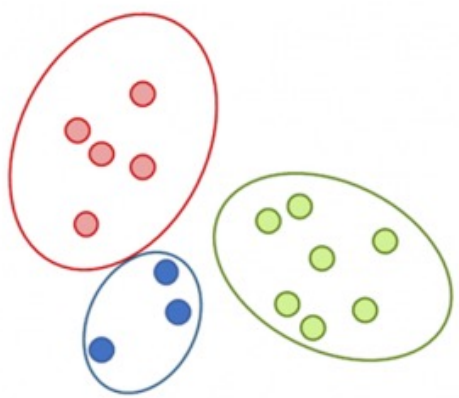
Today

Supervised learning



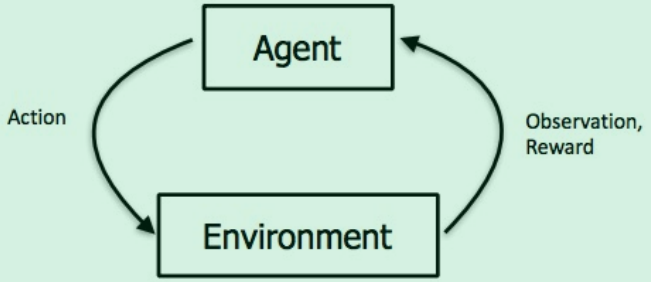
Predictions from labeled data

Unsupervised learning



Patterns in unlabeled data

Reinforcement learning



Optimal decisions from data and experience

Overview

Stockholms universitet

- **Reinforcement Learning - Introduction**
- Main Characteristics of RL and Data-Driven Decisions
- Stateless RL (multi-armed bandits): Exploration vs. Exploitation
- Markov Decision Process: Delayed Consequences
- Deep RL: Generalization
- Conclusions

Reinforcement Learning

Goal: Learn to make optimal sequences of decisions in an unknown enviroment by **Trial and Error**

Reinforcement Learning

Goal: Learn to make optimal **SEQUENCES** of decisions in an unknown enviroment by **Trial and Error**

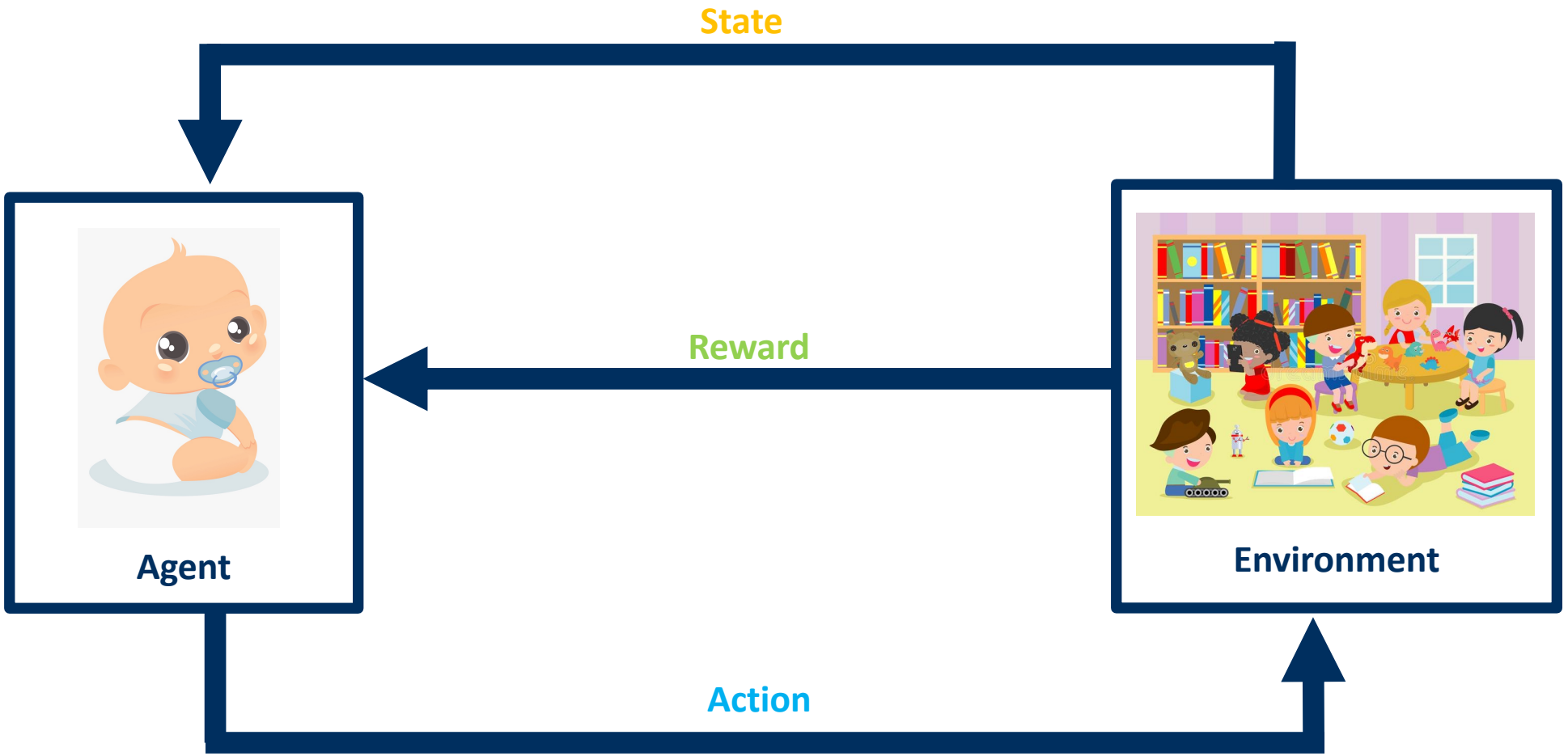
Reinforcement Learning

Goal: Learn to make **OPTIMAL** sequences of decisions in an unknown enviroment by **Trial and Error**

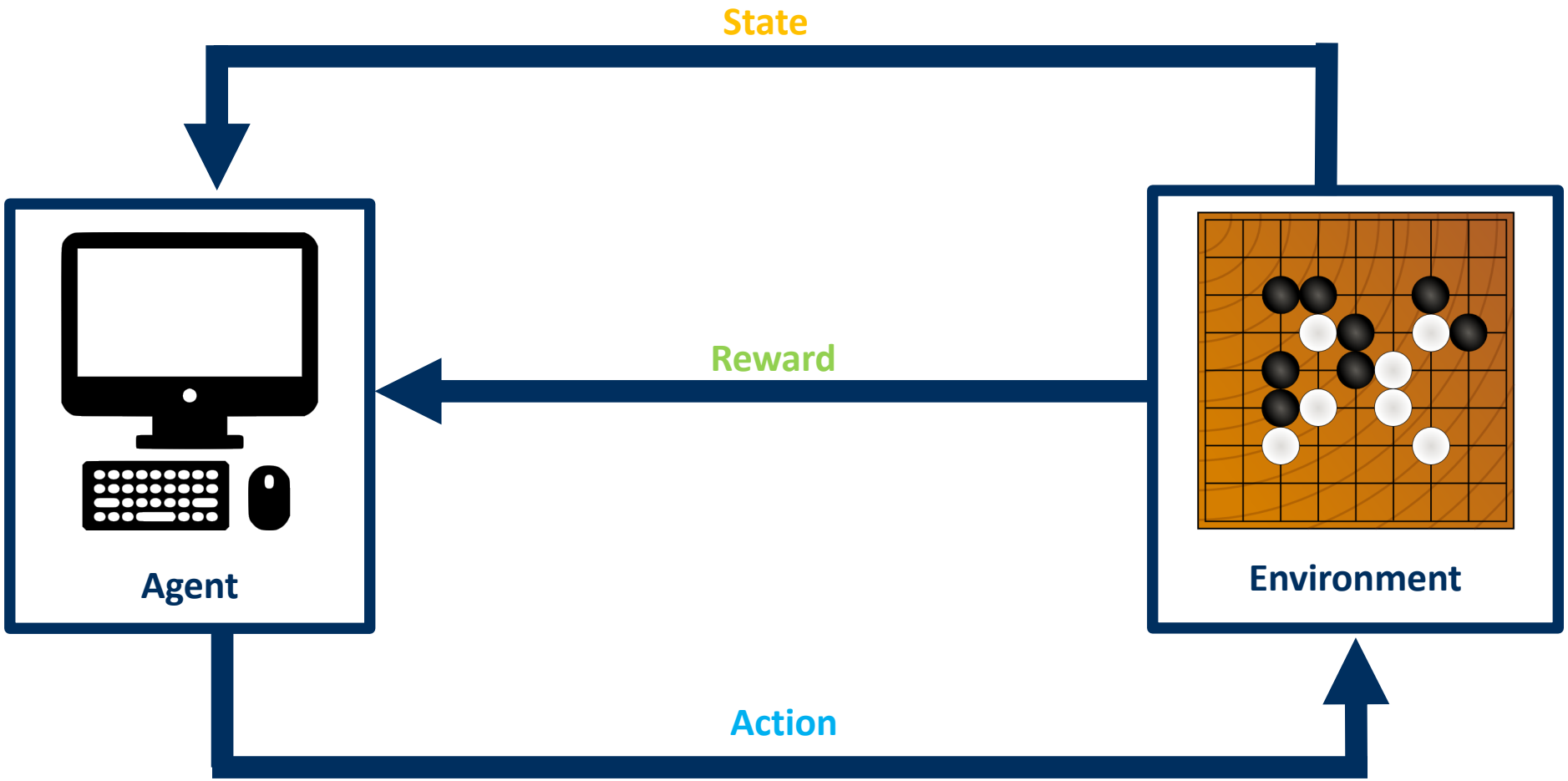
Reinforcement Learning

Goal: LEARN to make optimal sequences of decisions in an unknown enviroment by **Trial and Error**

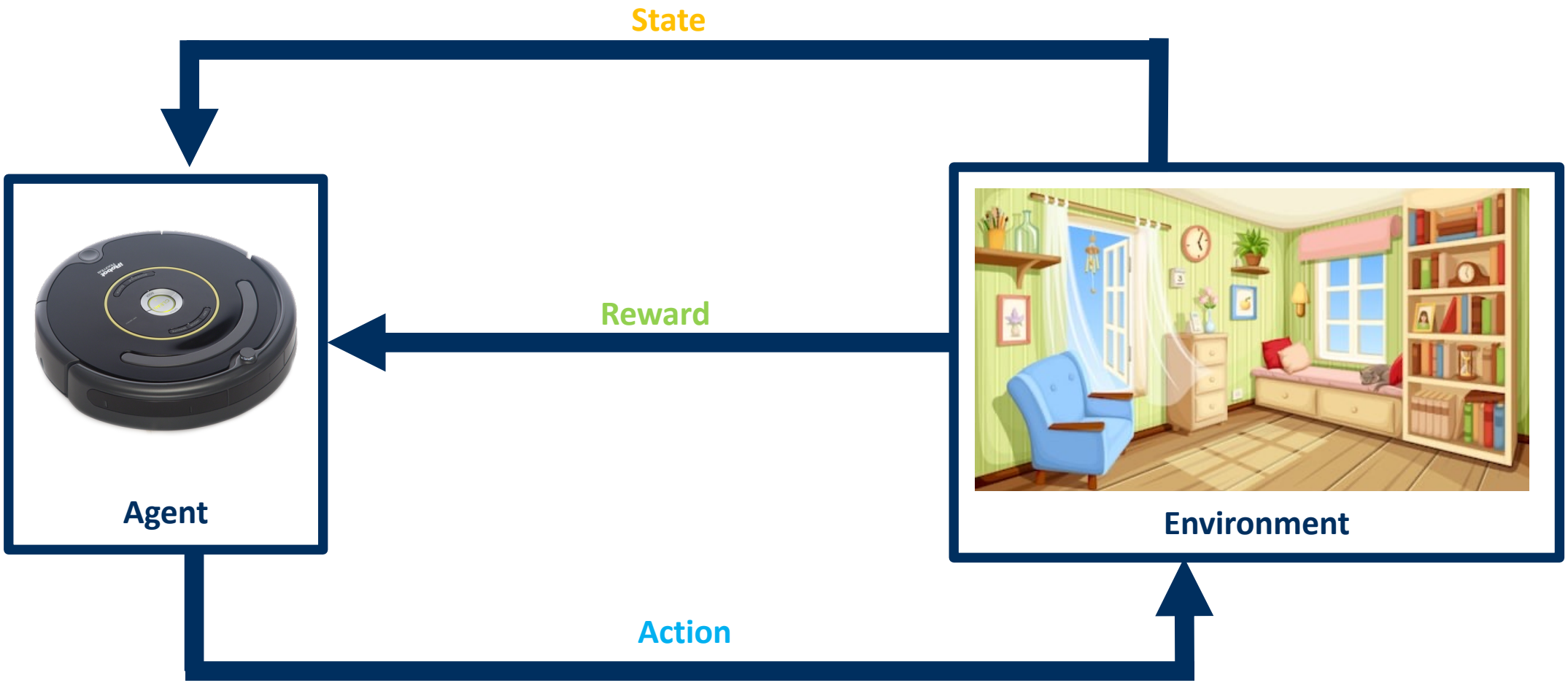
Reinforcement Learning: Human Learning



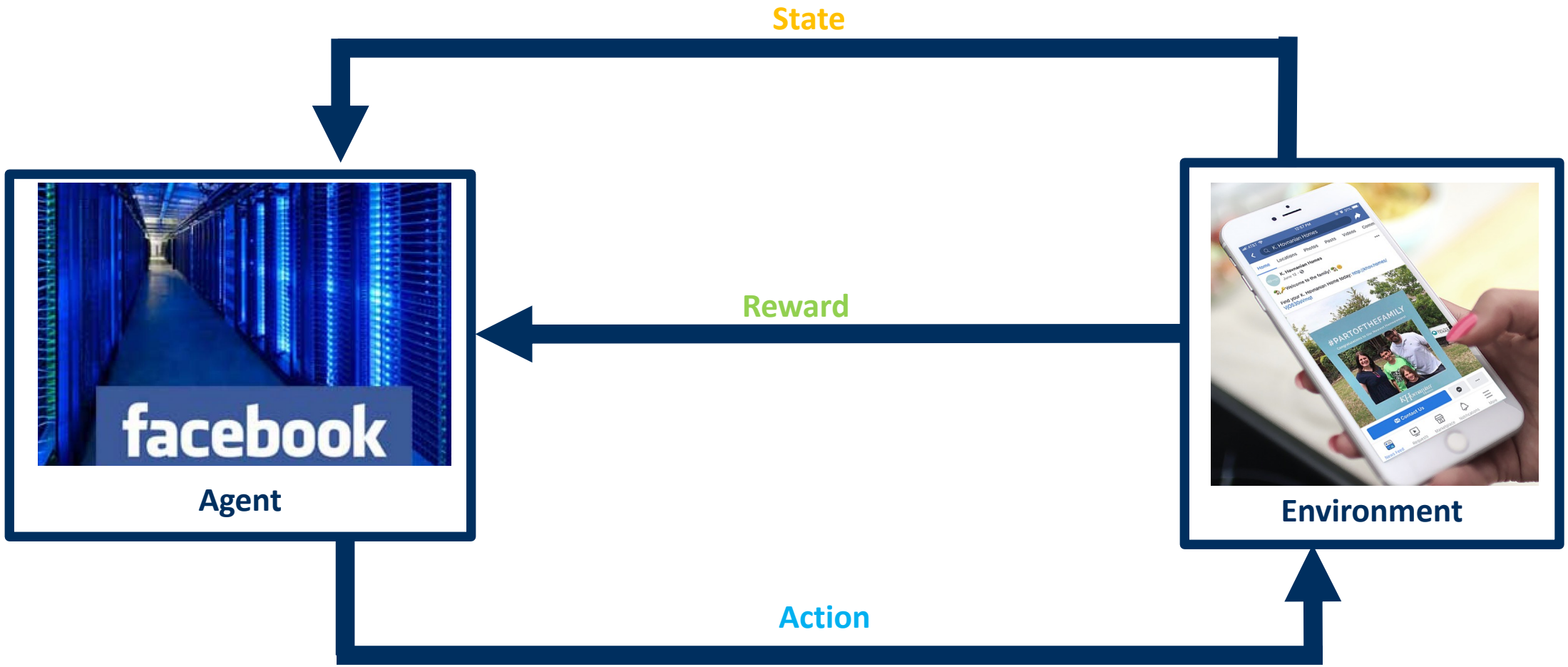
Reinforcement Learning: AlphaGo



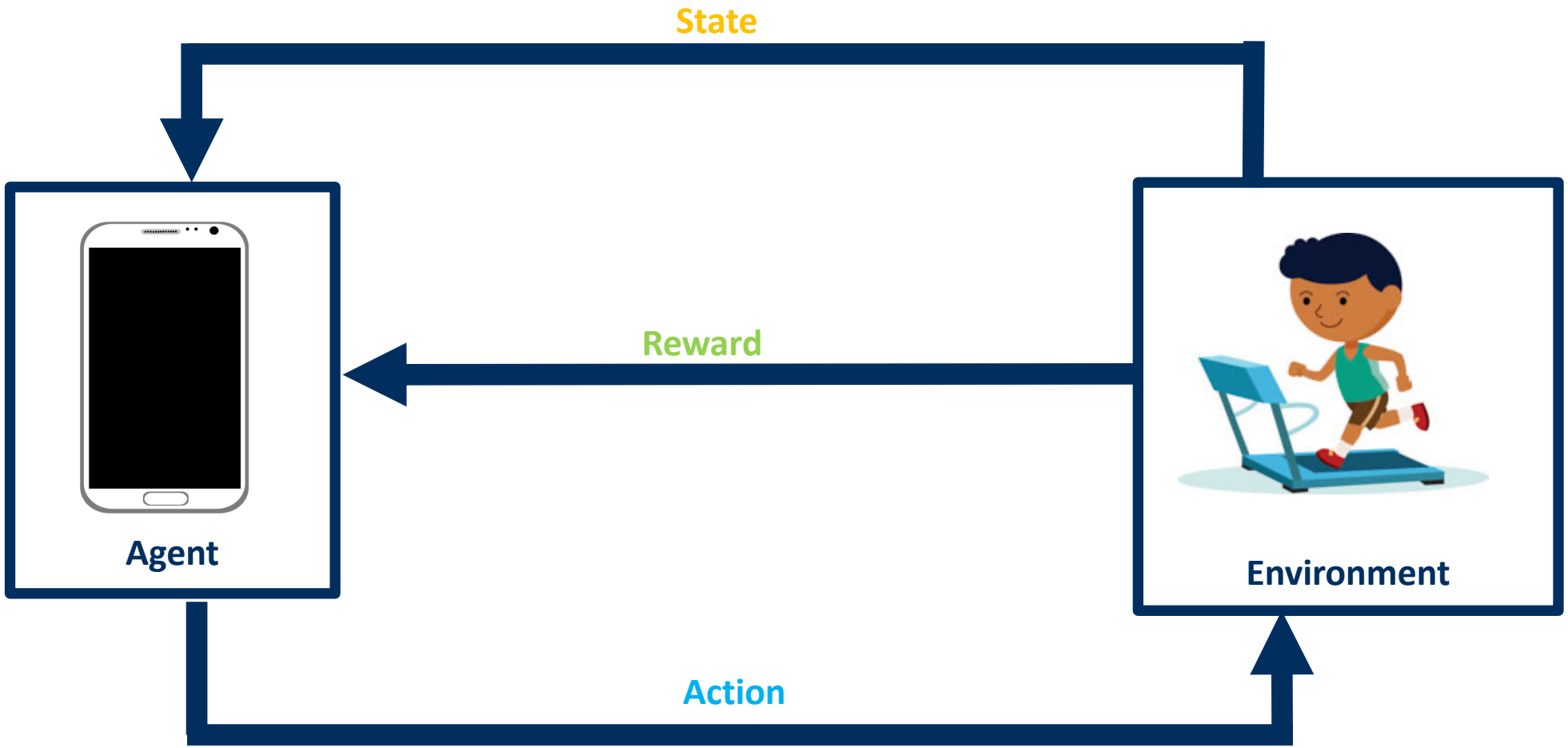
Reinforcement Learning: Robotics



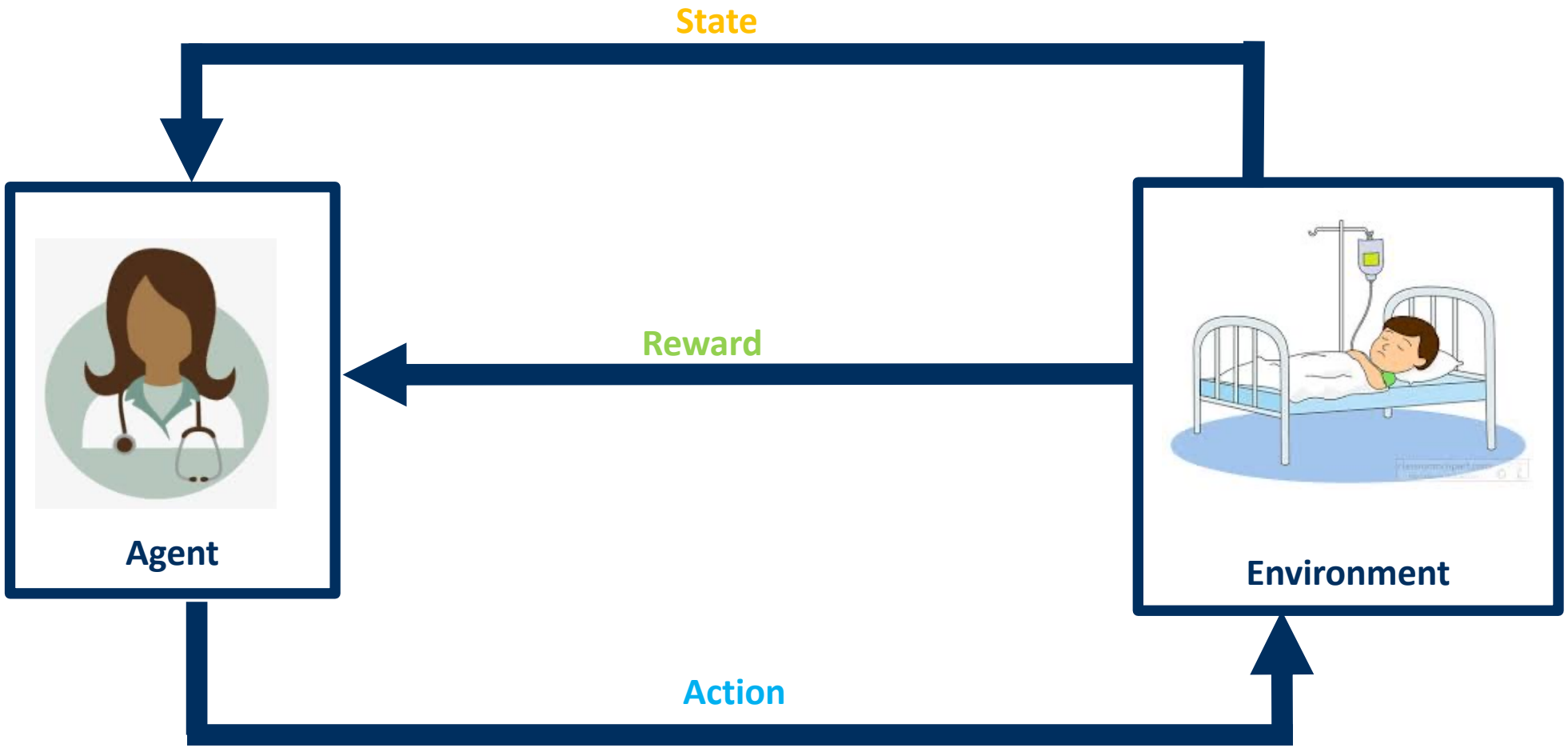
Reinforcement Learning: Target Advertising



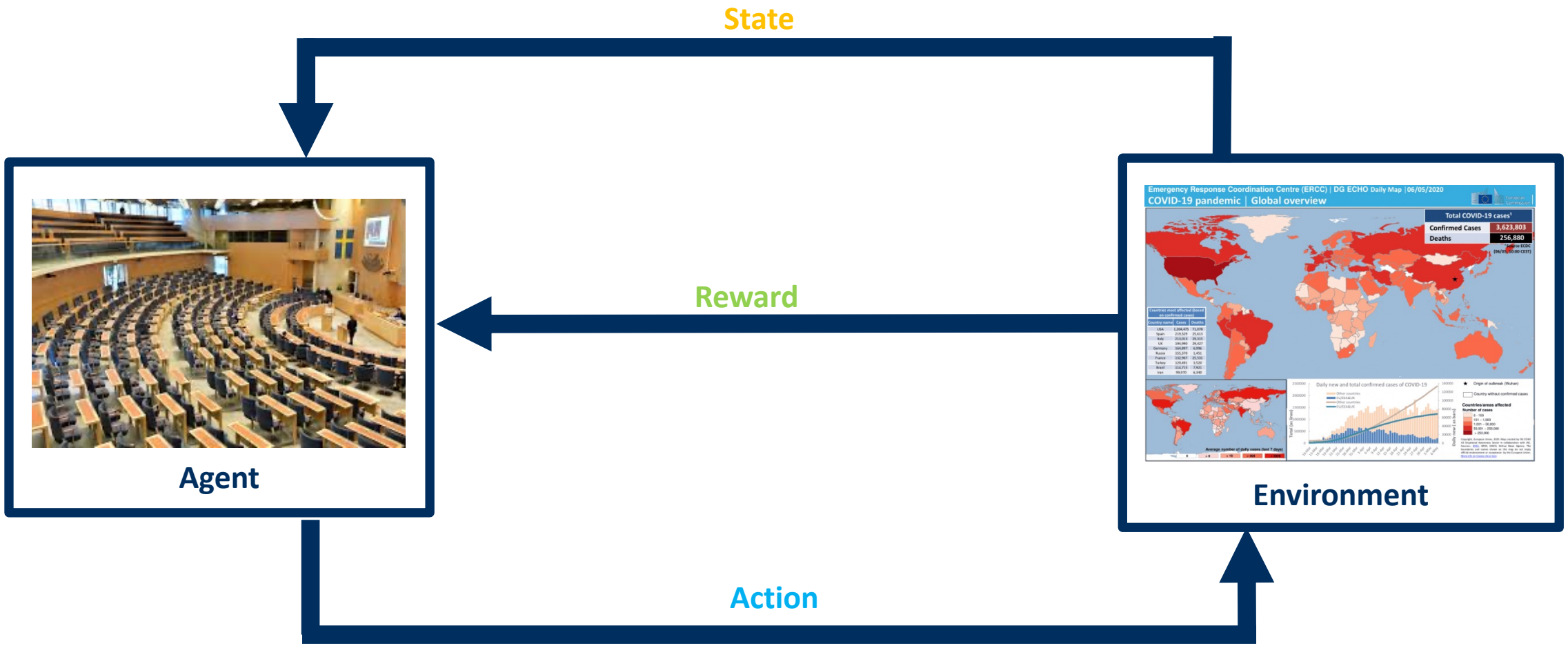
Reinforcement Learning: Mobile Health



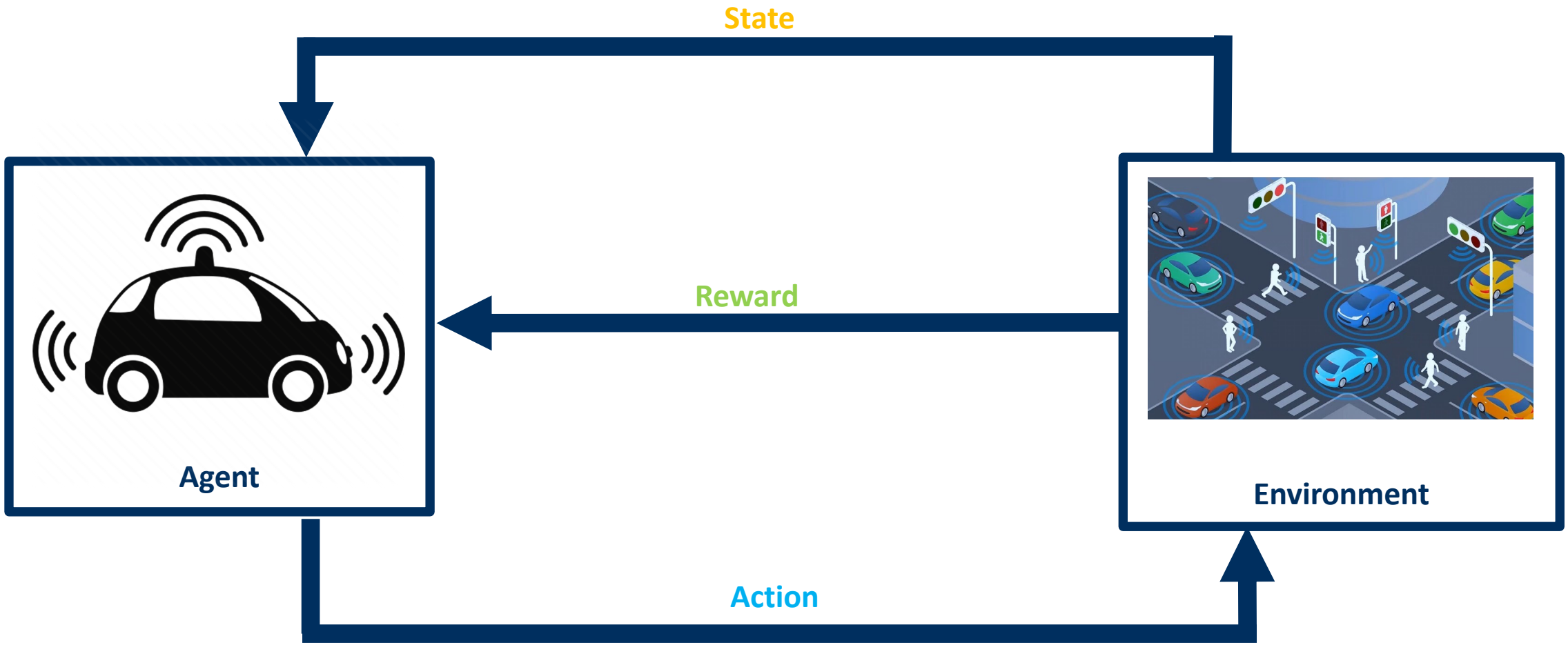
Reinforcement Learning: Treatment Selection



Reinforcement Learning: Pandemic Measures



Reinforcement Learning: Autonomous cars



Reinforcement Learning

Goal: Learn to make **optimal sequences** of decisions in an unknown enviroment by **Trial and Error**

What does the AI AGENT Learn?

Goal:

maximize $\sum_{t=1}^{T} R_t$

What does the AI AGENT Learn?

Goal:

$$\underset{\pi(\cdot) \in \mathbf{\Pi}}{\text{maximize}} \quad \sum_{t=1}^{T} R_t$$

Subject to $a_t = \pi(s_t)$

Policy (function from states to action):

$$a_t = \pi(s_t)$$

Three Basic RL Paradigms

Stockholms universitet

Goal:

$$\underset{\pi(\cdot) \in \mathbf{\Pi}}{\text{maximize}} \quad \sum_{t=1}^{T} R_t$$

Subject to $a_t = \pi(s_t)$

Policy (function from states to action):

$$a_t = \pi(s_t)$$

Multi Armed Bandits

Agent has no information for decision making

$$a_t = \pi(\mathbf{0})$$

Contextual Bandits

Agent has static information for decision making

$$a_t = \pi(s_t)$$

No relationship between s_t and s_τ for $t \neq \tau$

Markov Decision Process

Agent has dynamic information for decision making

$$a_t = \pi(s_t)$$

$$s_{t+1} = T(s_t, a_t)$$

Overview

Stockholms universitet

- Reinforcement Learning - Introduction
- **Main Characteristics of RL and Data-Driven Decisions**
- Stateless RL (multi-armed bandits): Exploration vs. Exploitation
- Markov Decision Process: Delayed Consequences
- Deep RL: Generalization
- Conclusions

Main Characteristics of RL

1. Optimization
2. Exploration-Explotation
3. Delayed Reward
4. Generalization

Main Characteristics of RL

1. Optimization
2. Exploration-Explotation
3. Delayed Reward
4. Generalization

Characteristics of RL: Optimization

Stockholms universitet

Goal:

$$\underset{\pi(\cdot) \in \mathbf{\Pi}}{\text{maximize}} \quad \sum_{t=1}^{T} R_t$$

Subject to $\quad a_t = \pi(s_t)$

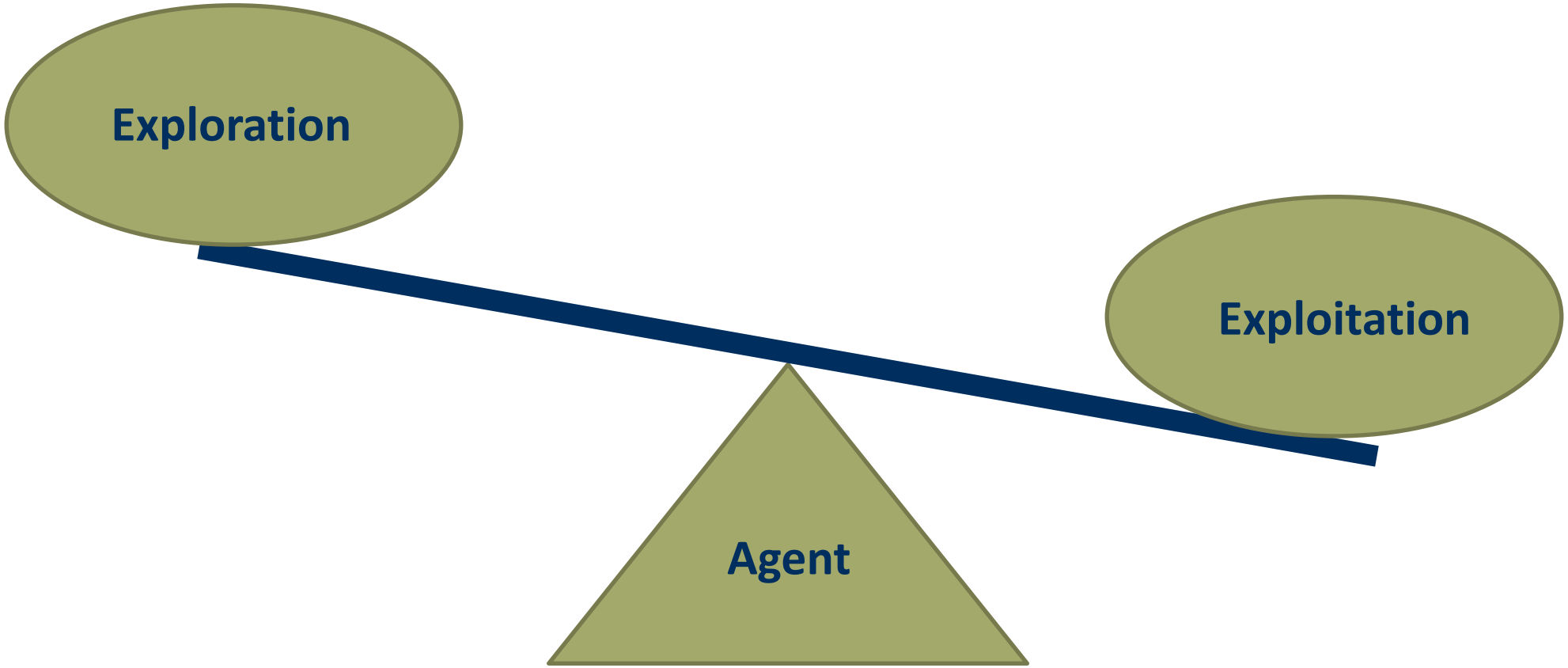
- Optimize some well-defined and measurable objective.
- Reward is an engineering decision: **This is how we tell the AI agent what to focus on!**
- Changing the rewards changes the policy that the agent learns.

Main Characteristics of RL

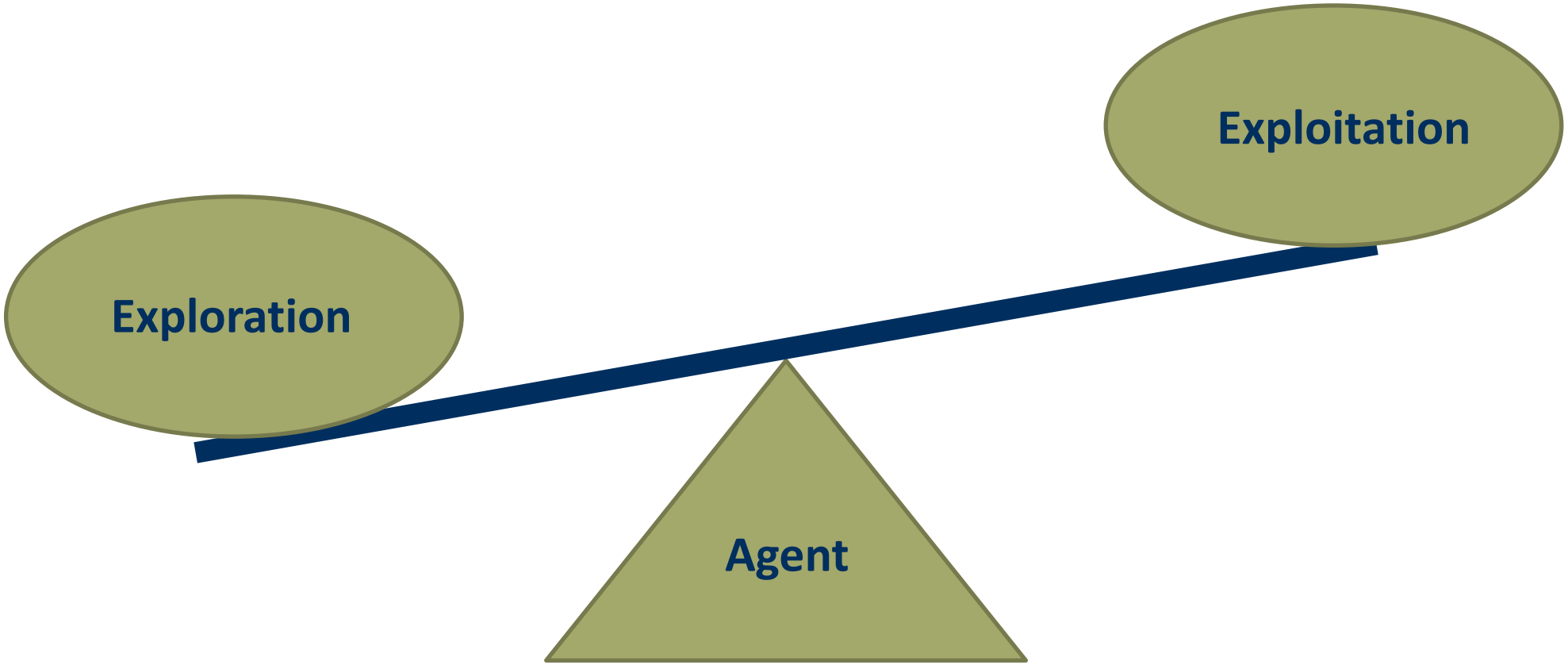
1. Optimization
2. Exploration-Explotation
3. Delayed Reward
4. Generalization

Exploration vs Exploitation

Exploration vs Exploitation

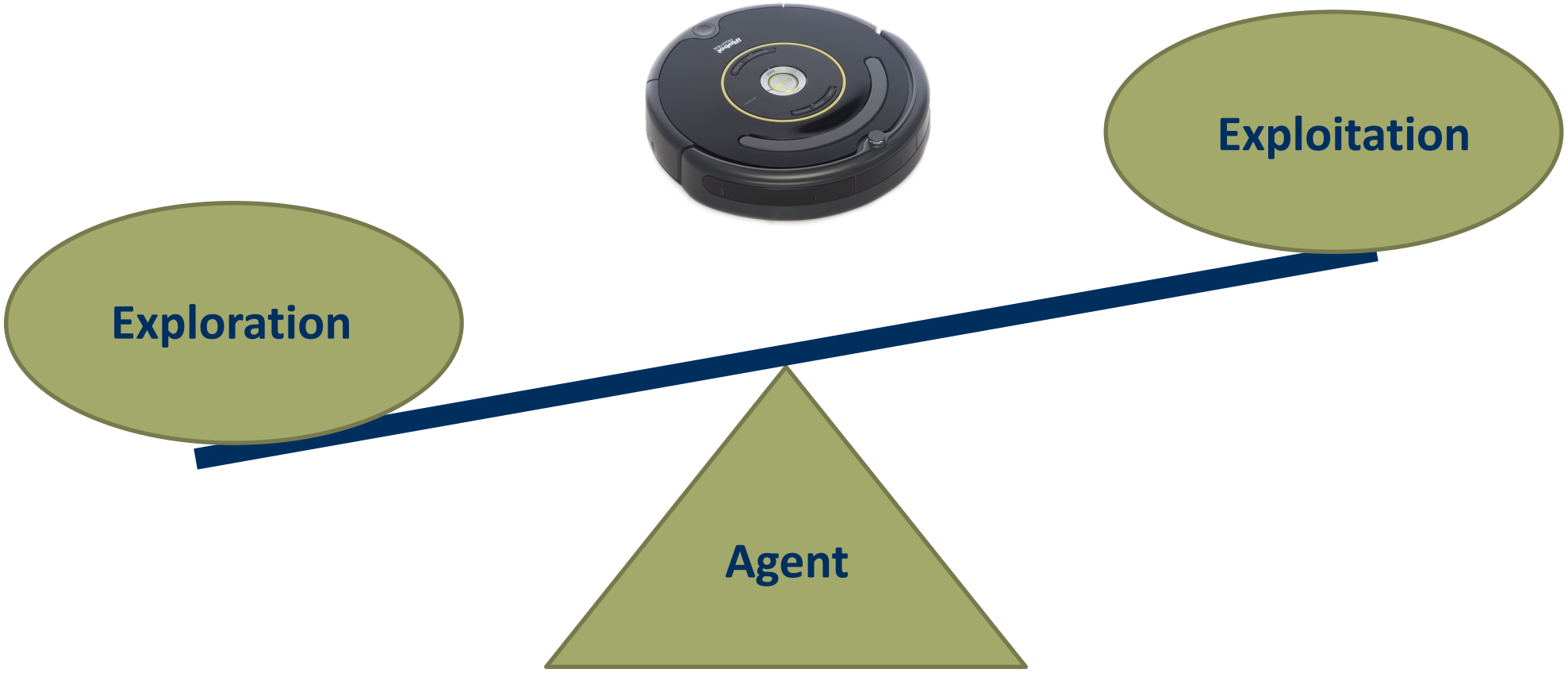


Exploration vs Exploitation

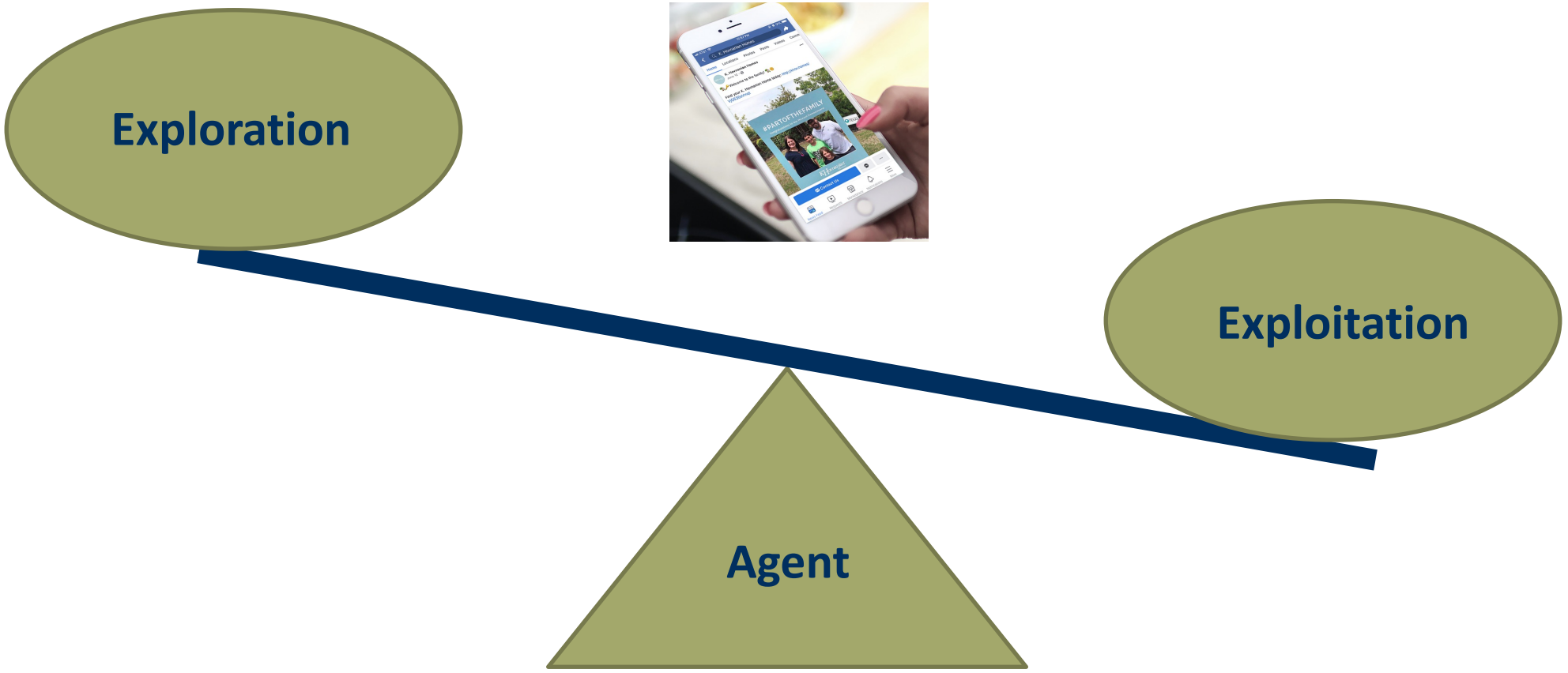


Exploration vs Exploitation

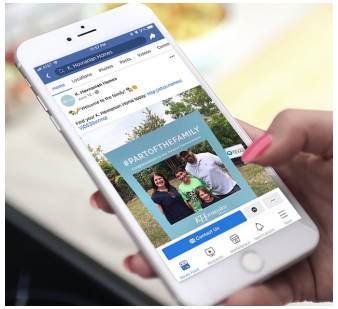
Exploration vs Exploitation



Exploration vs Exploitation



Exploration vs Exploitation

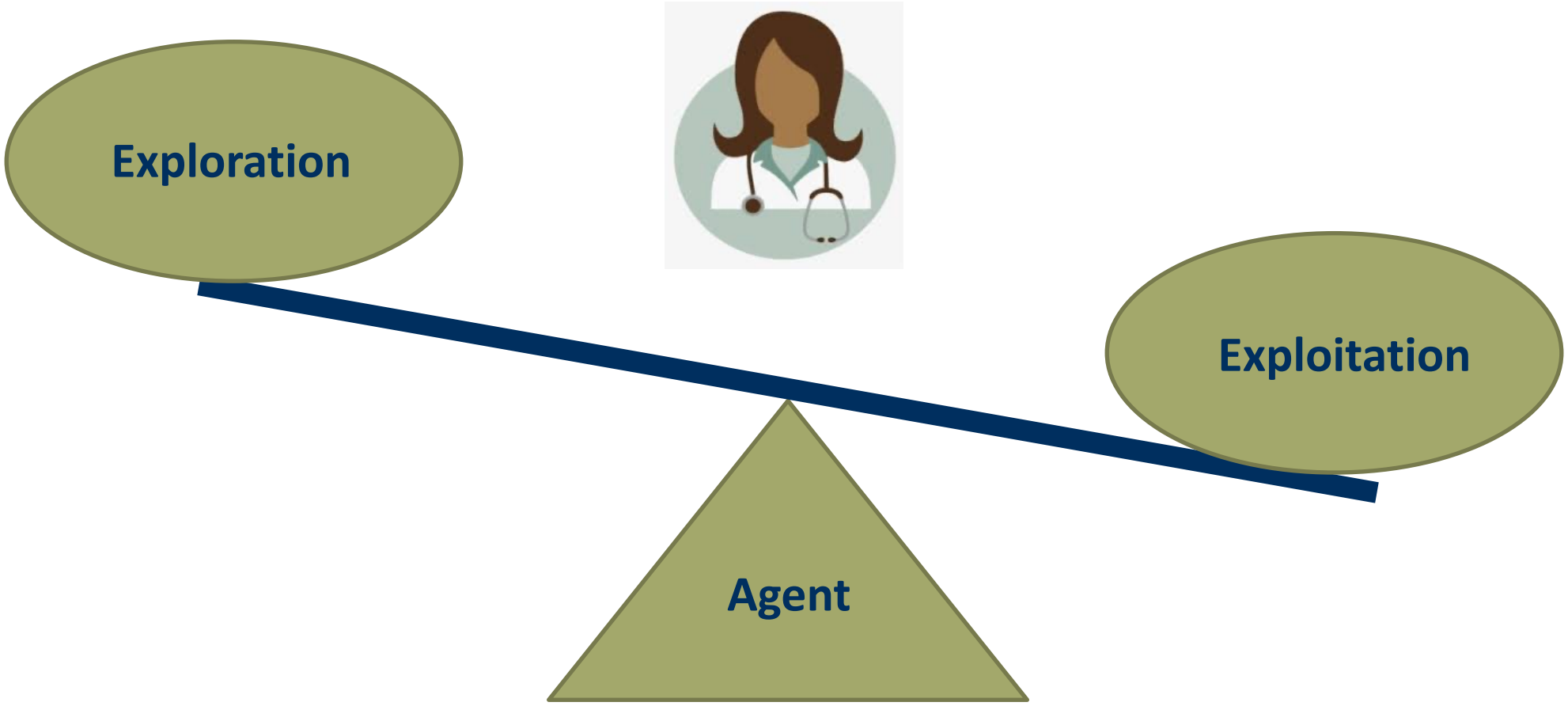


Exploitation

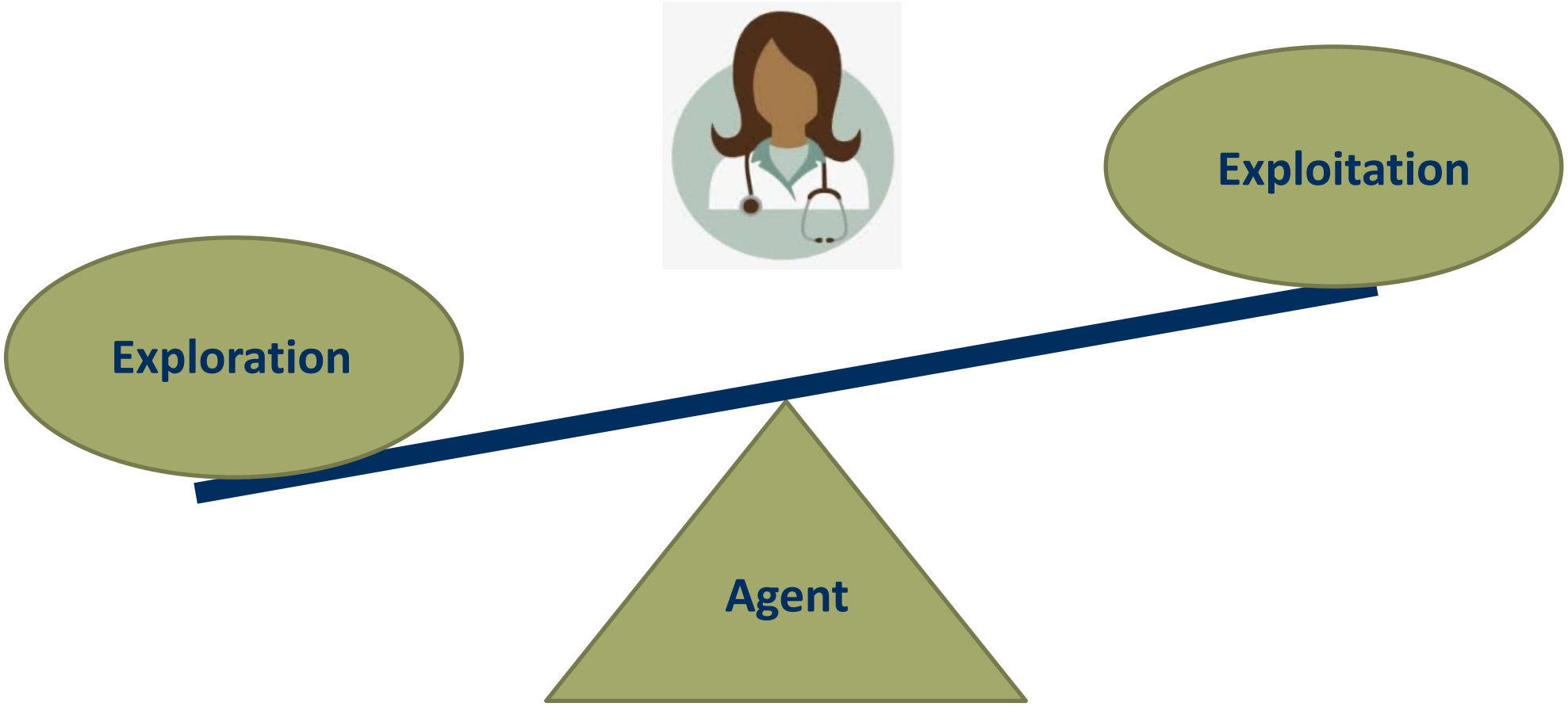
Exploration

Agent

Exploration vs Exploitation



Exploration vs Exploitation

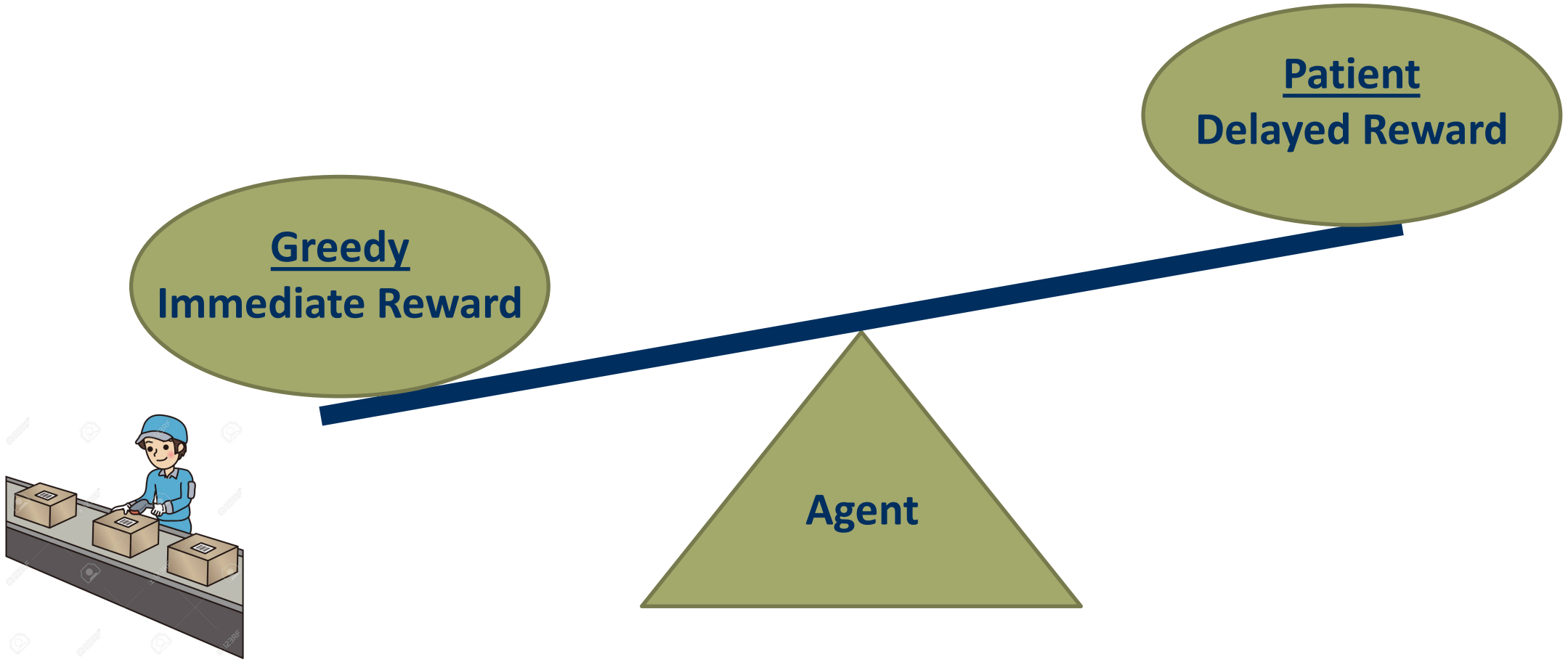


Main Characteristics of RL

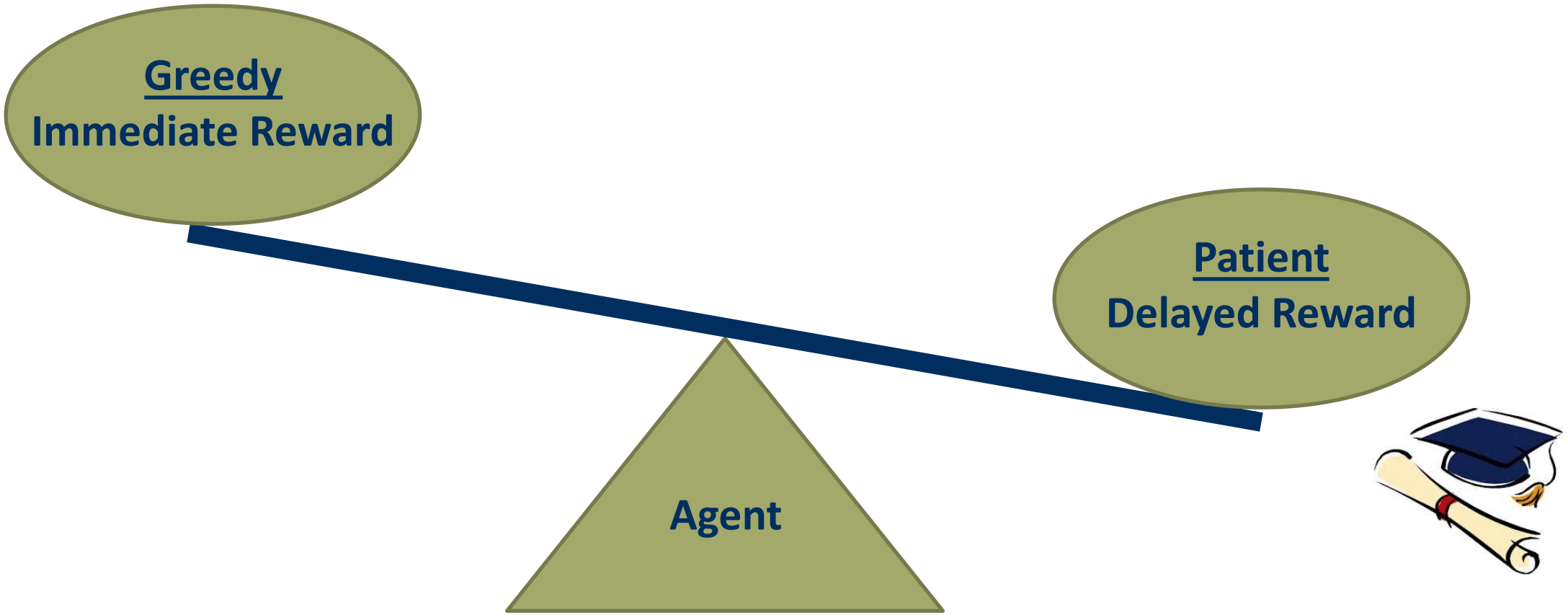
1. Optimization
2. Exploration-Explotation
3. Delayed Reward
4. Generalization

Delayed Reward

Delayed Reward



Delayed Reward



Delayed Reward

Patient
Delayed Reward

Greedy
Immediate Reward

Agent

Delayed Reward

Main Characteristics of RL

1. Optimization
2. Exploration-Explotation
3. Delayed Reward
4. Generalization

Characteristics of RL: Generalization

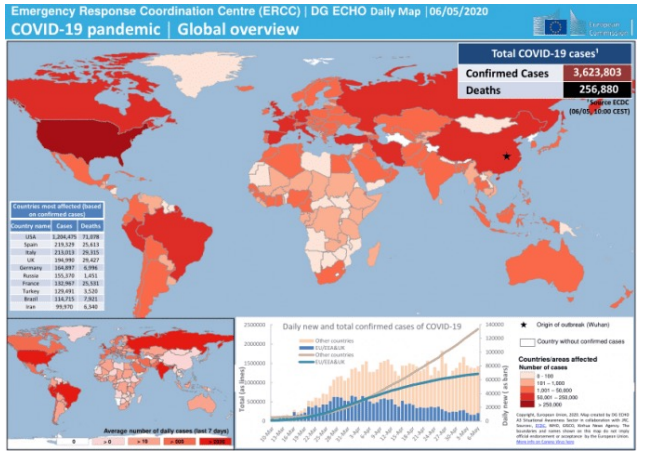
Policy (function from states to action):

$$a_t = \pi(s_t)$$

Impossible to try all state and action pairs (s_t, a_t). Need to generalize from limited samples.

$(256^{100\times300})^3$ images!

Deepmind Nature, 2015



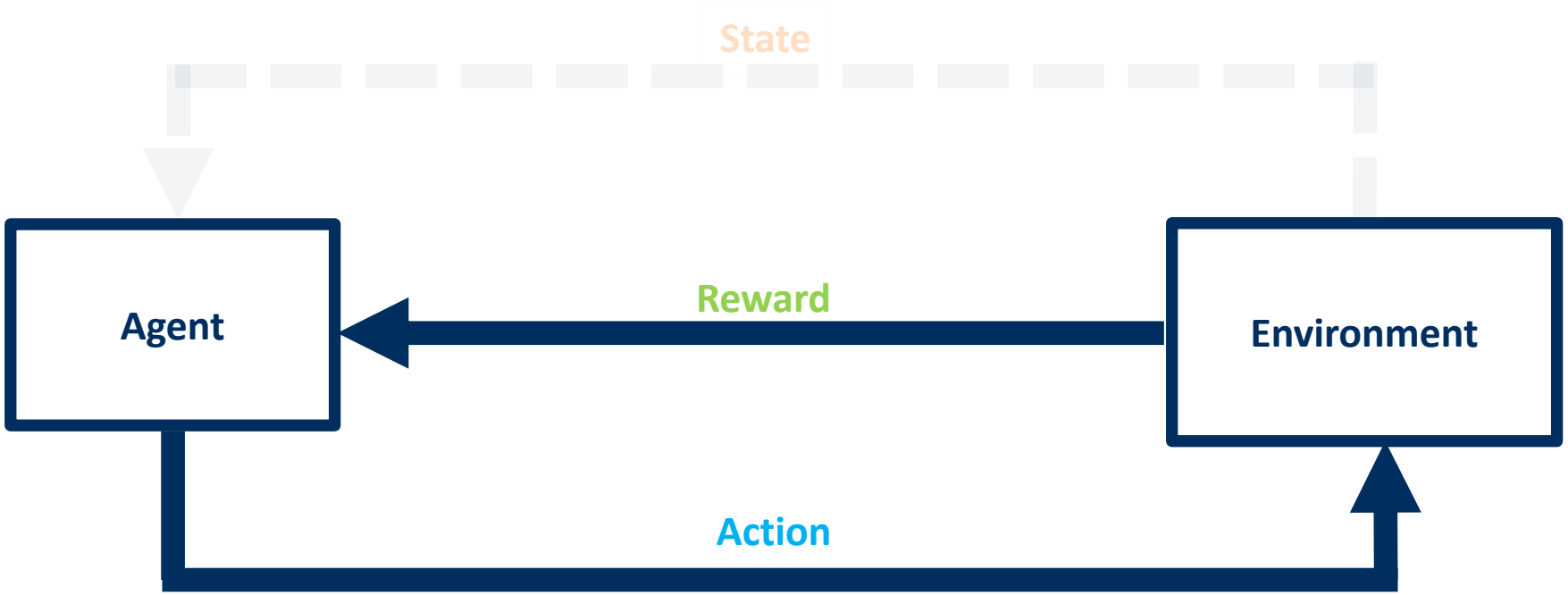
Pandemic

Infinitely many states and actions

Overview

- Reinforcement Learning - Introduction
- Main Characteristics of RL and Data-Driven Decisions
- **Stateless RL (multi-armed bandits): Exploration vs. Exploitation**
- Markov Decision Process: Delayed Consequences
- Deep RL: Generalization
- Conclusions

Multi-Armed Bandits: Exploration vs. Exploitation



Goal: Learn to make good sequences of decisions in an unknown enviroment by **Trial and Error**

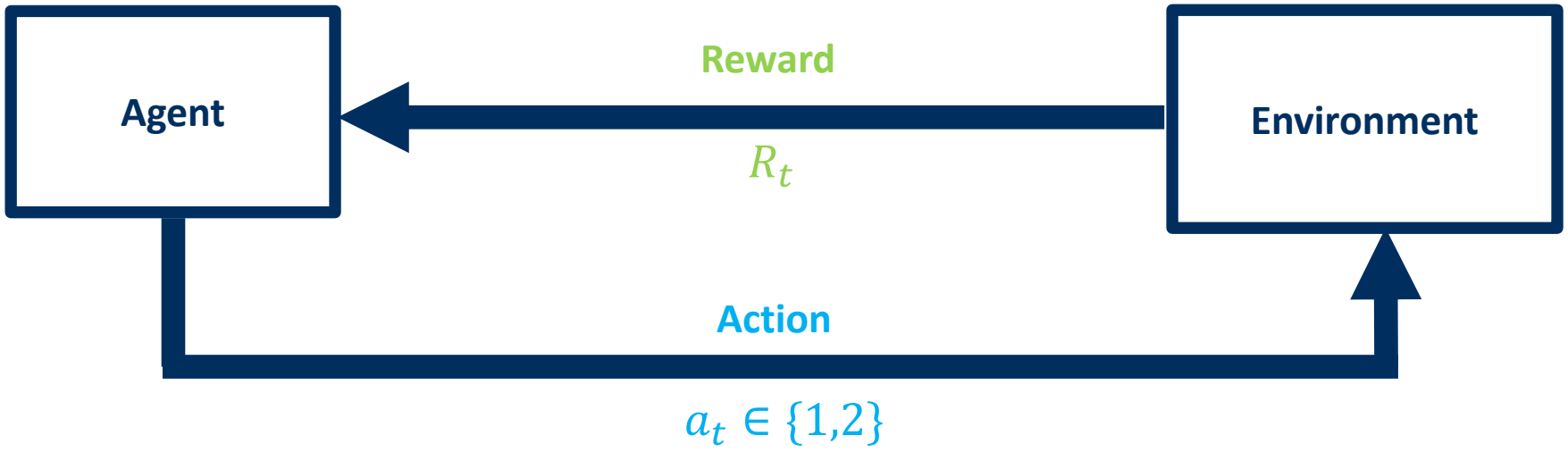
Main Characteristics of RL

- Multi-armed bandits - Setup
- Action Value
- Basic Algorithms
- Regret Analysis

Main Characteristics of RL

- Multi-armed bandits - Setup
- Action Value
- Basic Algorithms
- Regret Analysis

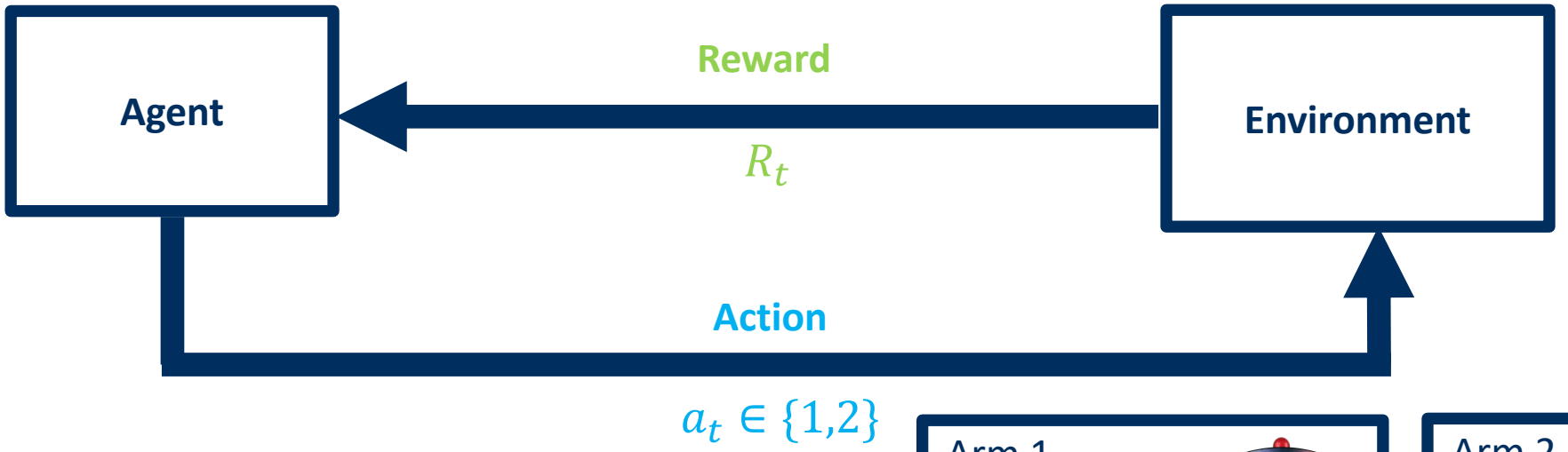
Multi Armed Bandits



Goal:

maximize $\sum_{t=1}^{T} R_t$

Multi Armed Bandits



Arm 1

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Multi Armed Bandits

$a_t \in \{1,2\}$

$t =$	1	2	3	4	5	6	7	8	9	10
$a_t = 1$										
$a_t = 2$										

Arm 1

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Multi Armed Bandits

$t =$	1	2	3	4	5	6	7	8	9	10
$a_t = 1$	0									
$a_t = 2$										

Arm 1

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Multi Armed Bandits

$t =$	1	2	3	4	5	6	7	8	9	10
$a_t = 1$	0									
$a_t = 2$		10								

Arm 1

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Multi Armed Bandits

$t =$	1	2	3	4	5	6	7	8	9	10
$a_t = 1$	0		10							
$a_t = 2$		10								

Arm 1

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Multi Armed Bandits

$t =$	1	2	3	4	5	6	7	8	9	10
$a_t = 1$	0		10	0						
$a_t = 2$		10								

Arm 1

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Multi Armed Bandits

$t =$	1	2	3	4	5	6	7	8	9	10
$a_t = 1$	0		10	0						
$a_t = 2$		10			0					

Arm 1

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Multi Armed Bandits

$t =$	1	2	3	4	5	6	7	8	9	10
$a_t = 1$	0		10	0		0				
$a_t = 2$		10			0					

Arm 1

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Multi Armed Bandits

$t =$	1	2	3	4	5	6	7	8	9	10
$a_t = 1$	0		10	0		0				
$a_t = 2$		10			0		0			

Arm 1

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Multi Armed Bandits

$t =$	1	2	3	4	5	6	7	8	9	10
$a_t = 1$	0		10	0		0				
$a_t = 2$		10			0		0	0		

Arm 1

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Multi Armed Bandits

$t =$	1	2	3	4	5	6	7	8	9	10
$a_t = 1$	0		10	0		0				
$a_t = 2$		10			0		0	0	0	

Arm 1

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Multi Armed Bandits

$t =$	1	2	3	4	5	6	7	8	9	10
$a_t = 1$	0		10	0		0				10
$a_t = 2$		10			0		0	0	0	

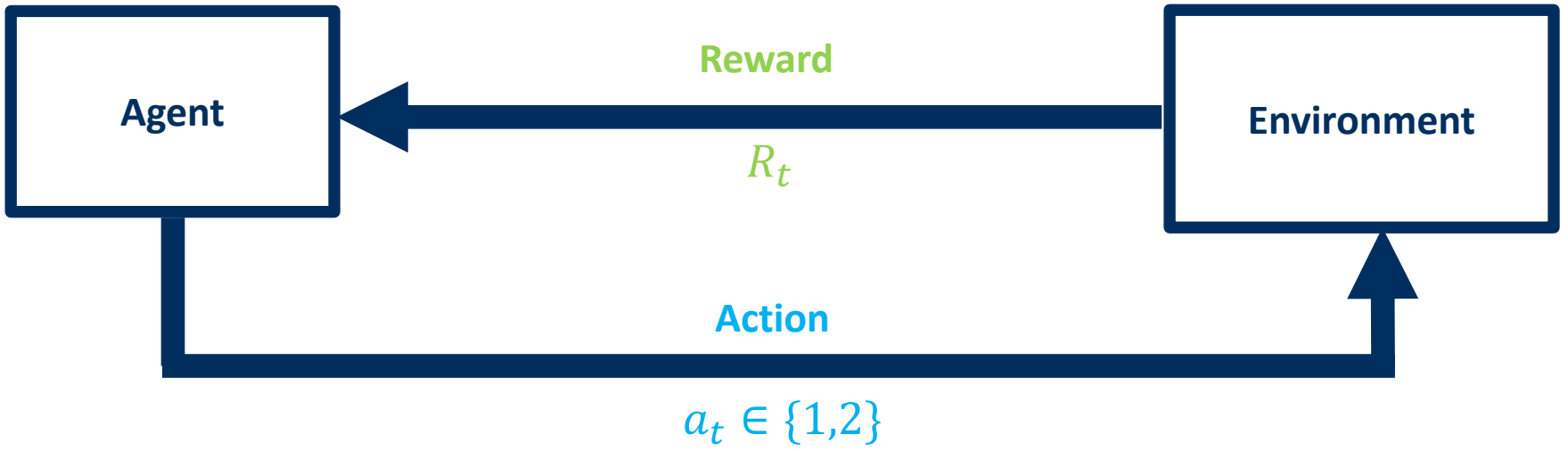
Arm 1

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Multi Armed Bandits: Treatment Selection

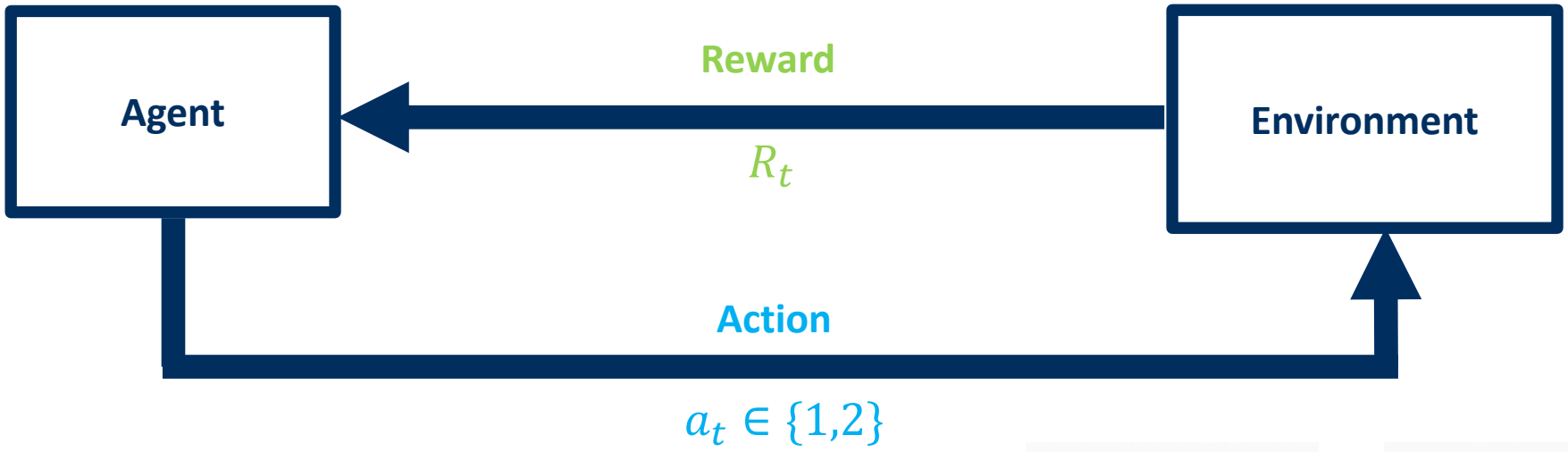


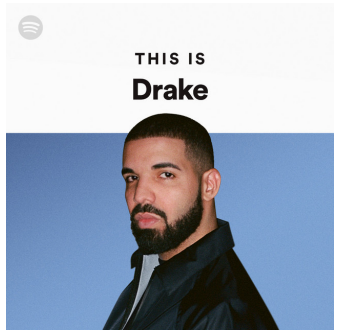
Treadment 1

Treatment 2

W. Thompson, *Biometrika* (1933)

Multi Armed Bandits: Recommendation

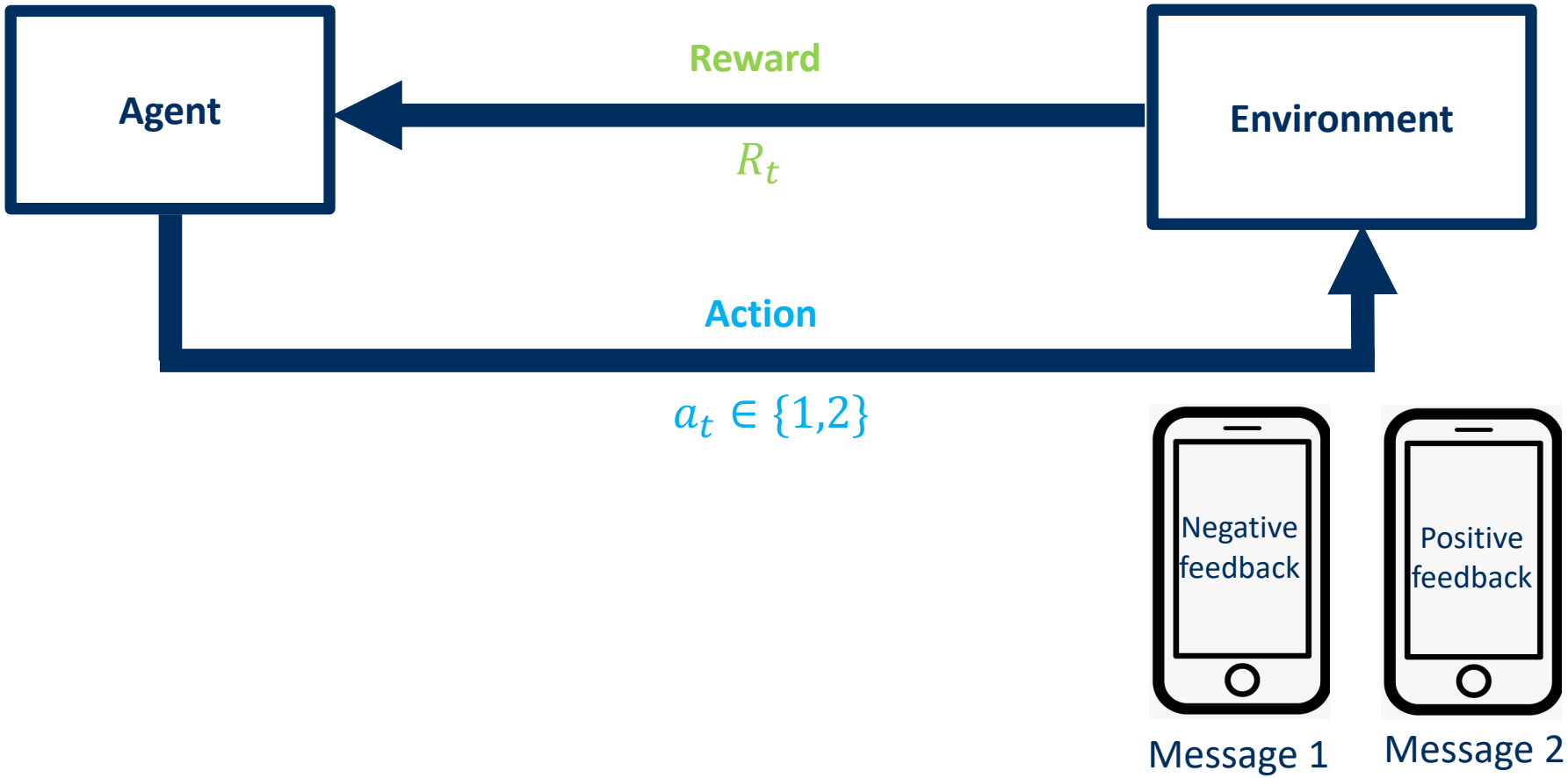




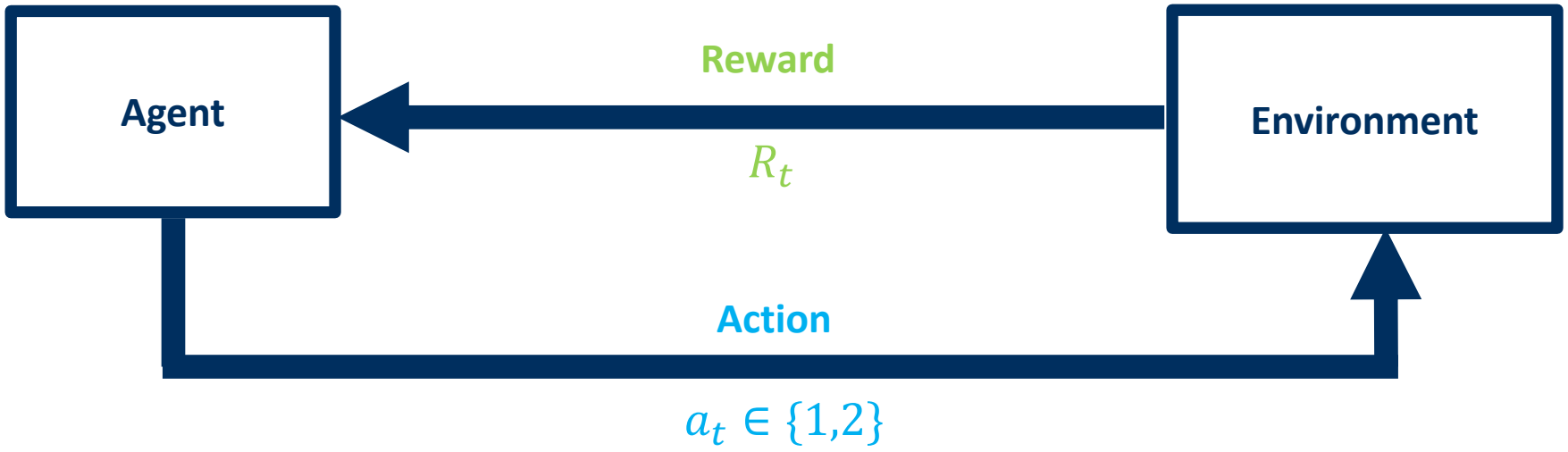
Artist 1

Artist 2

Multi Armed Bandits: mHealth



Multi Armed Bandits



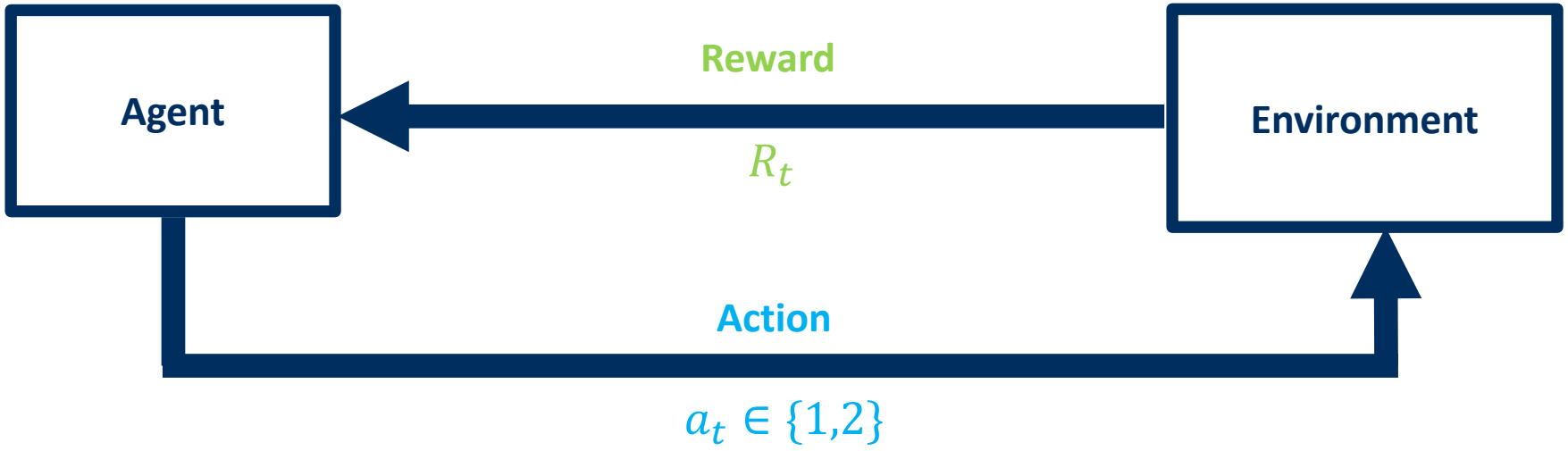
Goal:

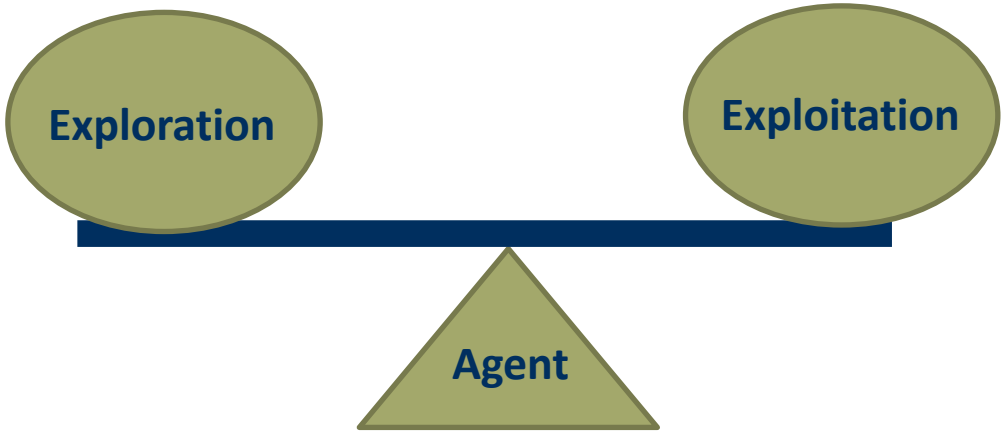
maximize $\sum_{t=1}^{T} R_t$

Main Characteristics of RL

- Multi-armed bandits - Setup
- Action Value
- Basic Algorithms
- Regret Analysis

How to explore? How to exploit?





How to explore? How to exploit?

Action Value

$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$

Arm 1

$Q(1) = \$1$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$Q(2) = \$3$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

How to explore? How to exploit?

Action Value

$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$

Arm 1

$Q(1) = \$1$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$Q(2) = \$3$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

How to explore? How to exploit?

Action Value

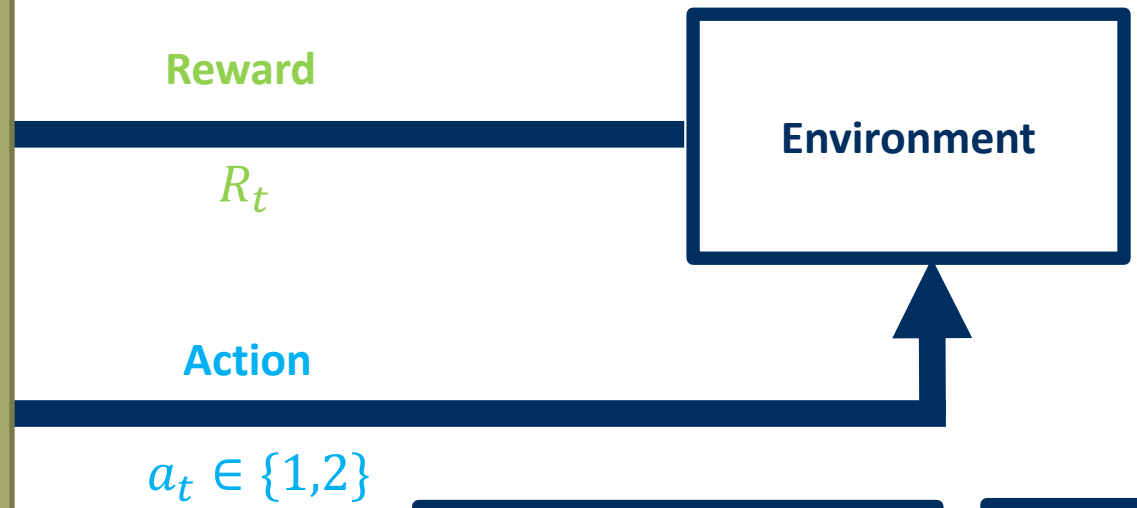
$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$



t	1	2	3	4	5	6	7	8	9	10	$\hat{Q}(a)$
$a_t = 1$											0
$a_t = 2$											0

Arm 1

$Q(1) = \$1$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$Q(2) = \$3$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

How to explore? How to exploit?

Action Value

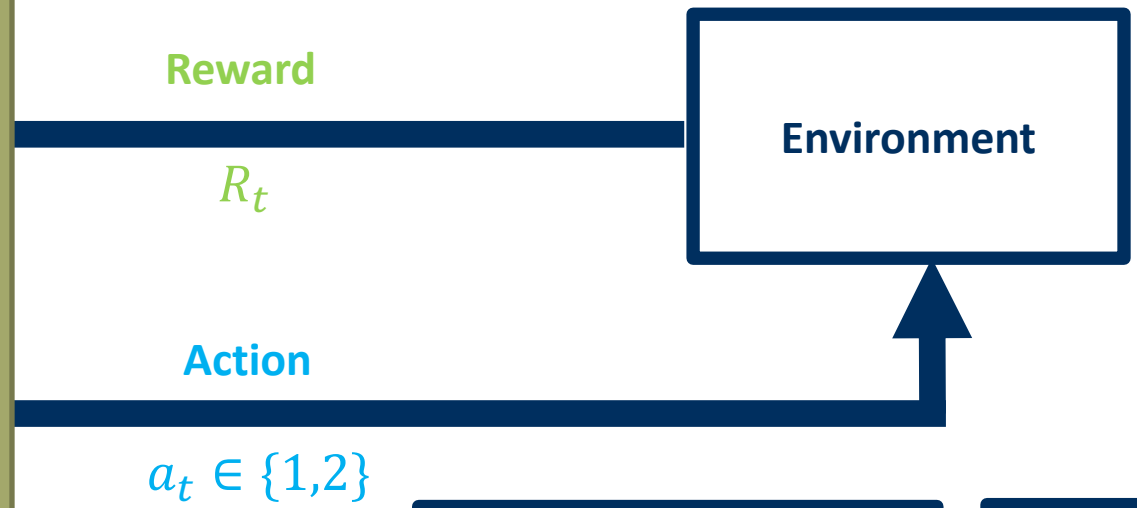
$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$



t	1	2	3	4	5	6	7	8	9	10	$\hat{Q}(a)$
$a_t = 1$	0										0
$a_t = 2$											0

Arm 1

$Q(1) = \$1$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$Q(2) = \$3$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

How to explore? How to exploit?

Action Value

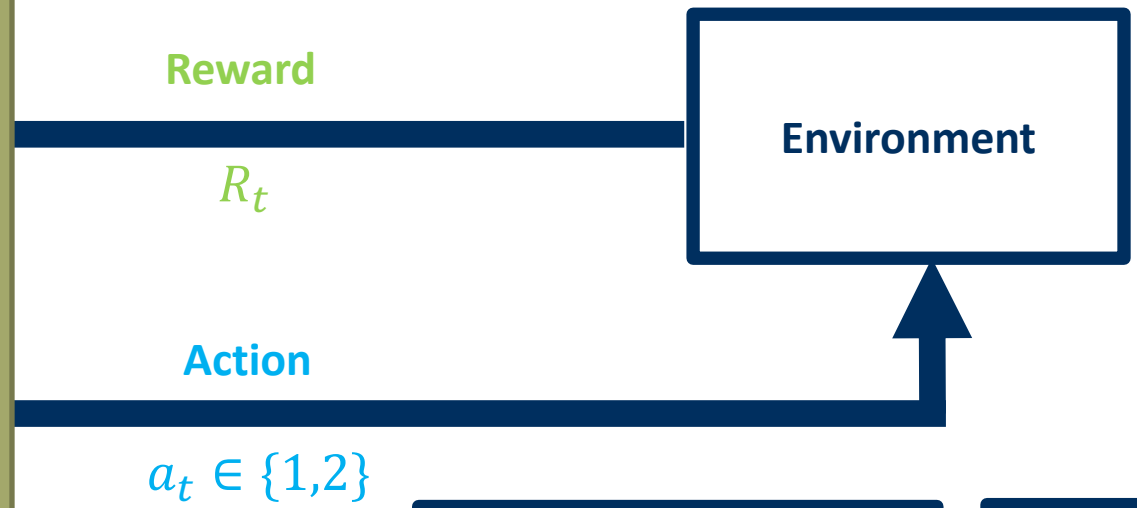
$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$



t	1	2	3	4	5	6	7	8	9	10	$\hat{Q}(a)$
$a_t = 1$	0										0
$a_t = 2$		10									10

Arm 1

$Q(1) = \$1$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$Q(2) = \$3$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

How to explore? How to exploit?

Action Value

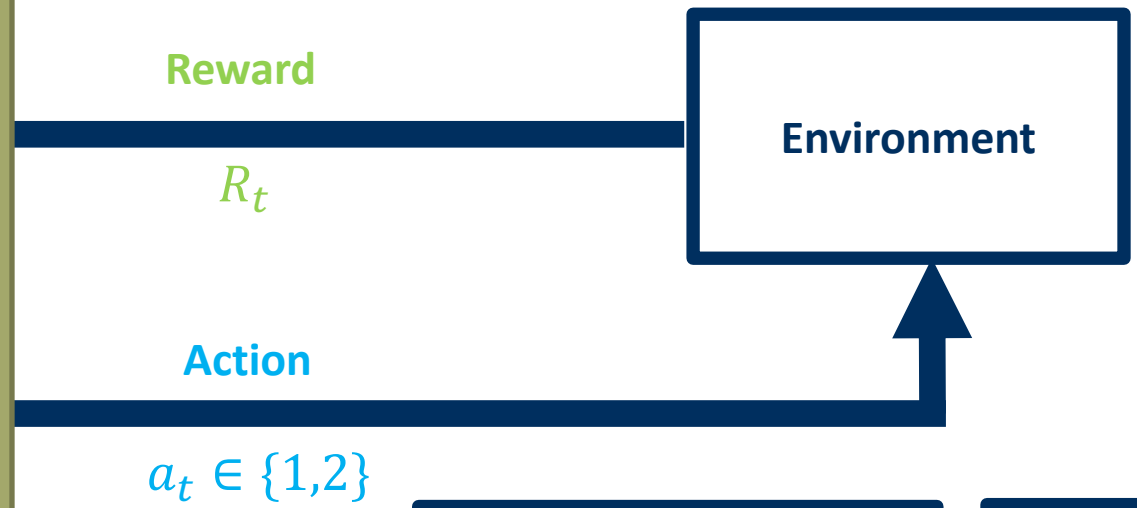
$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$



t	1	2	3	4	5	6	7	8	9	10	$\hat{Q}(a)$
$a_t = 1$	0		10								5
$a_t = 2$		10									10

Arm 1

$Q(1) = \$1$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$Q(2) = \$3$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

How to explore? How to exploit?

Action Value

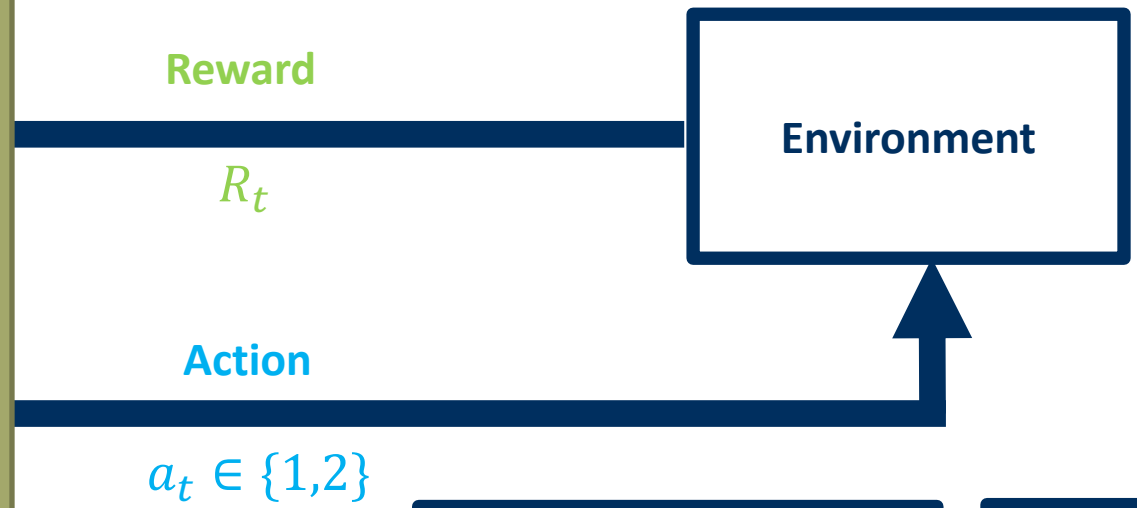
$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$



t	1	2	3	4	5	6	7	8	9	10	$\hat{Q}(a)$
$a_t = 1$	0		10	0							3.333
$a_t = 2$		10									10

Arm 1

$Q(1) = \$1$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$Q(2) = \$3$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

How to explore? How to exploit?

Action Value

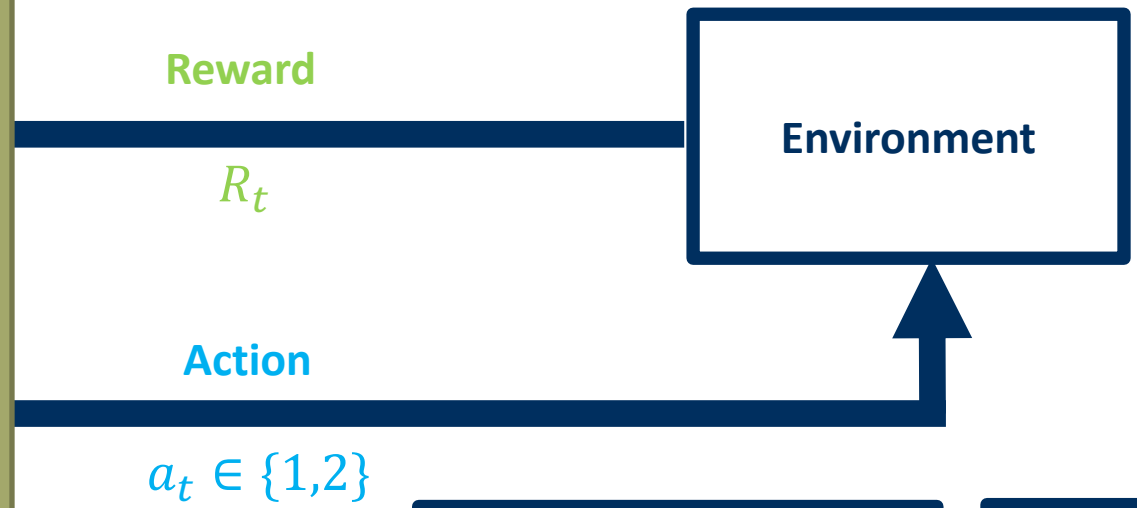
$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$



t	1	2	3	4	5	6	7	8	9	10	$\hat{Q}(a)$
$a_t = 1$	0		10	0							3.333
$a_t = 2$		10			0						5

Arm 1

$Q(1) = \$1$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$Q(2) = \$3$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

How to explore? How to exploit?

Action Value

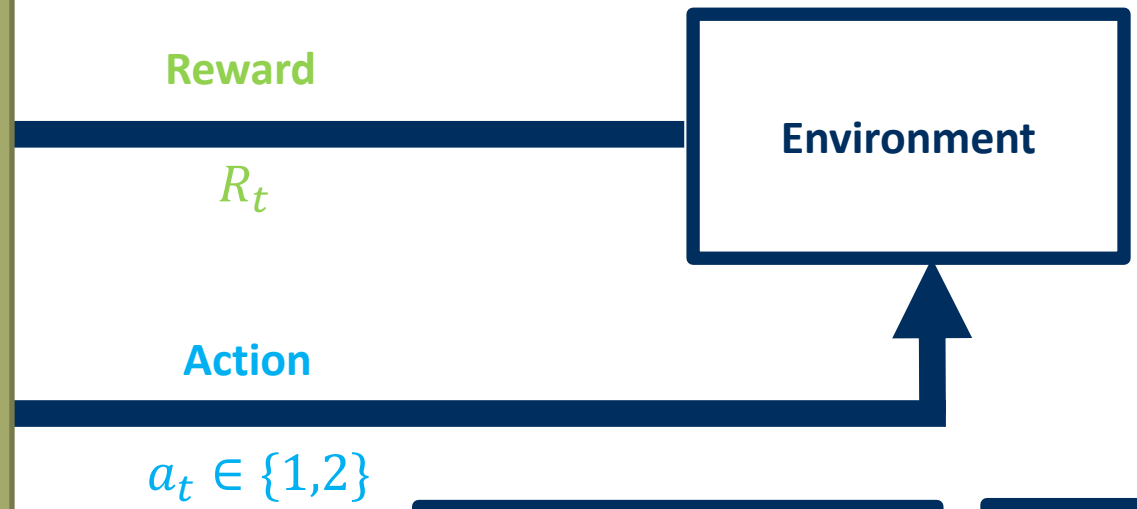
$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$



t	1	2	3	4	5	6	7	8	9	10	$\hat{Q}(a)$
$a_t = 1$	0		10	0		0					2.5
$a_t = 2$		10			0						5

Arm 1

$Q(1) = \$1$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$Q(2) = \$3$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

How to explore? How to exploit?

Action Value

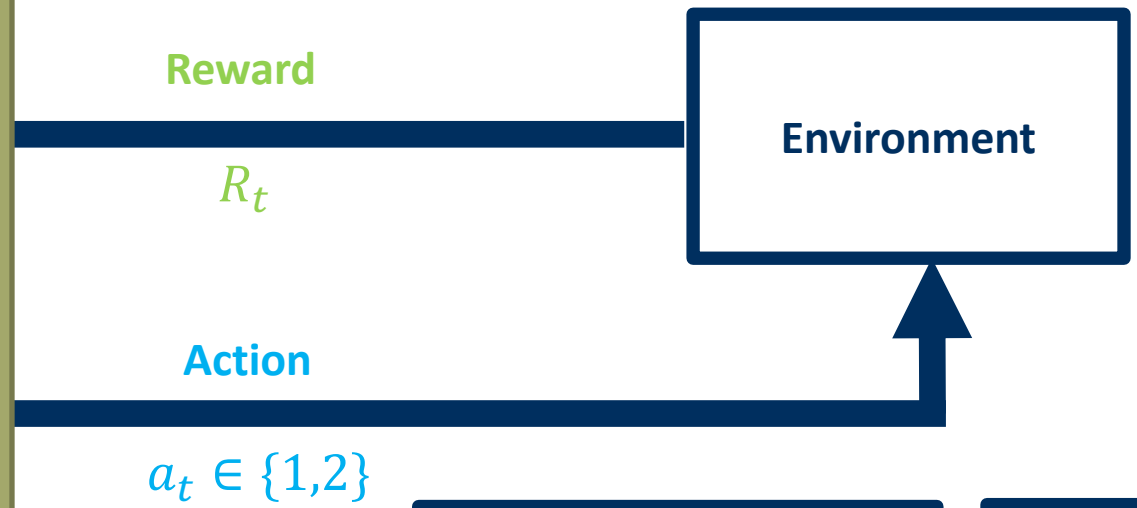
$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$



t	1	2	3	4	5	6	7	8	9	10	$\hat{Q}(a)$
$a_t = 1$	0		10	0		0					2.5
$a_t = 2$		10			0		0				3.333

Arm 1

$Q(1) = \$1$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$Q(2) = \$3$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

How to explore? How to exploit?

Action Value

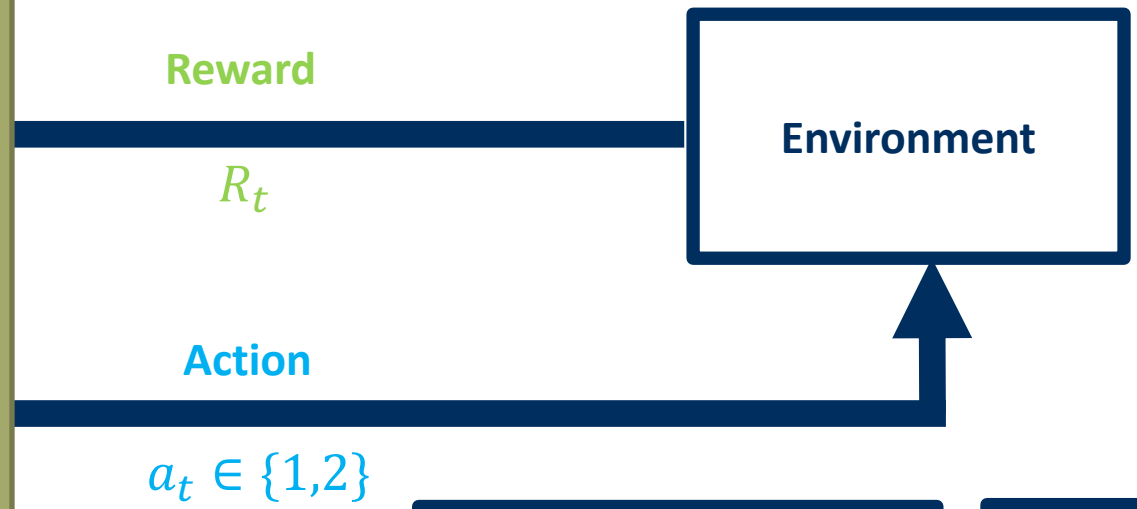
$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$



t	1	2	3	4	5	6	7	8	9	10	$\hat{Q}(a)$
$a_t = 1$	0		10	0		0					2.5
$a_t = 2$		10			0		0	0			2.5

Arm 1

$Q(1) = \$1$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$Q(2) = \$3$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

How to explore? How to exploit?

Action Value

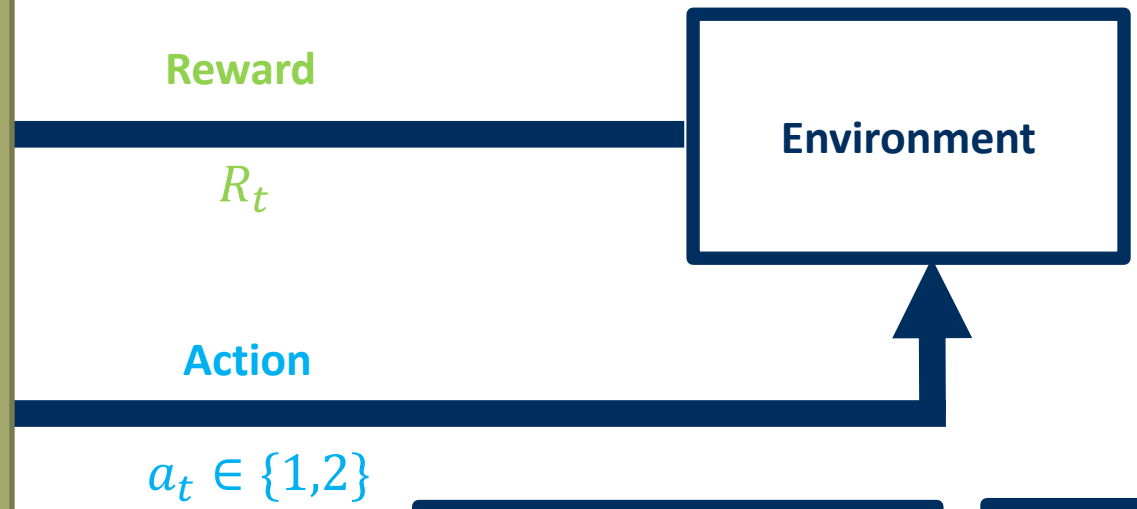
$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$



t	1	2	3	4	5	6	7	8	9	10	$\hat{Q}(a)$
$a_t = 1$	0		10	0		0					2.5
$a_t = 2$		10			0		0	0	0		2

Arm 1

$Q(1) = \$1$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

Arm 2

$Q(2) = \$3$

$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

How to explore? How to exploit?

Action Value

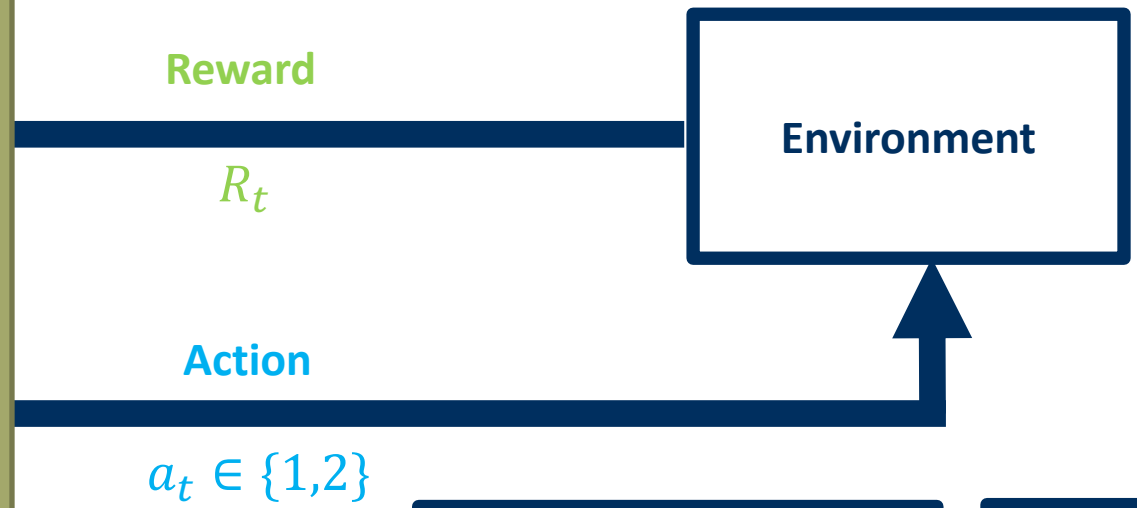
$$Q(a) = \boldsymbol{E}[R_t | a_t = a]$$

Optimal Policy:

$$a_t = argmax_a\, Q(a)$$

Value Estimate:

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$



t	1	2	3	4	5	6	7	8	9	10	$\hat{Q}(a)$
$a_t = 1$	0		10	0		0				10	4
$a_t = 2$		10			0		0	0	0		2

Arm 1

$Q(1) = \$1$

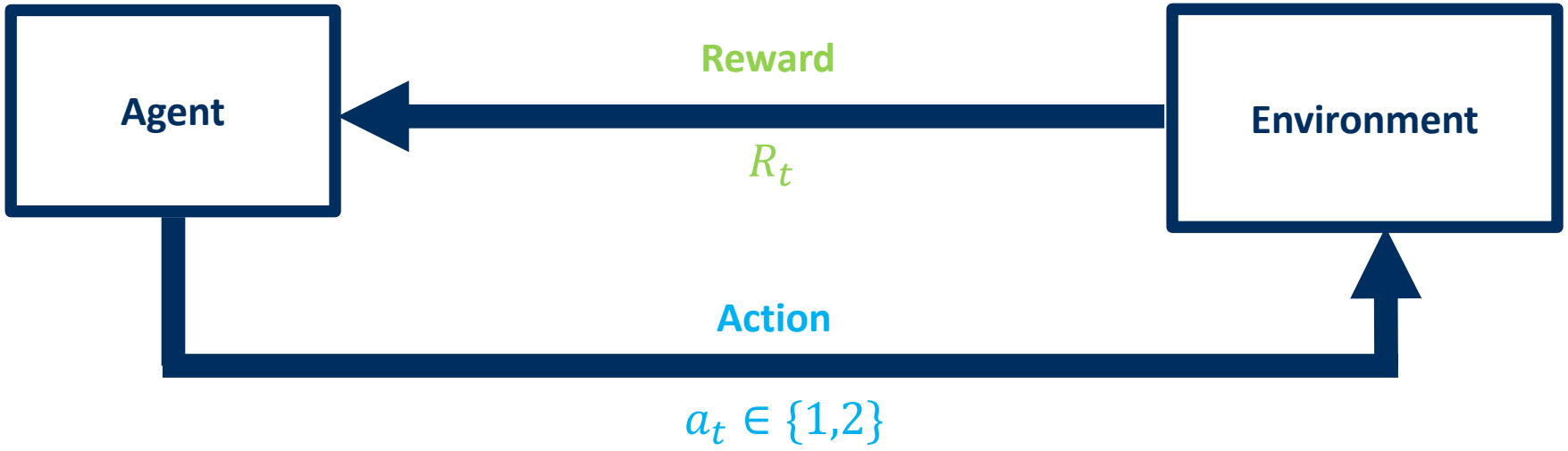
$$R_t = \begin{cases} \$10 & \text{with } \mathbf{10\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

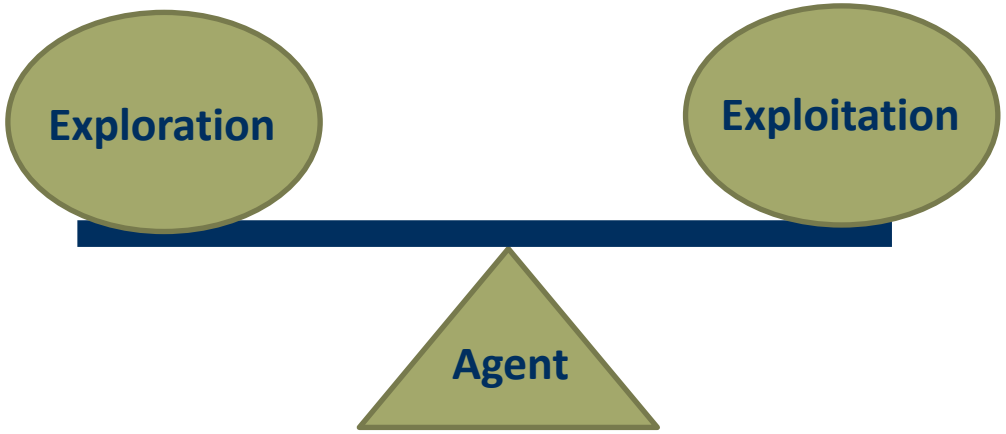
Arm 2

$Q(2) = \$3$

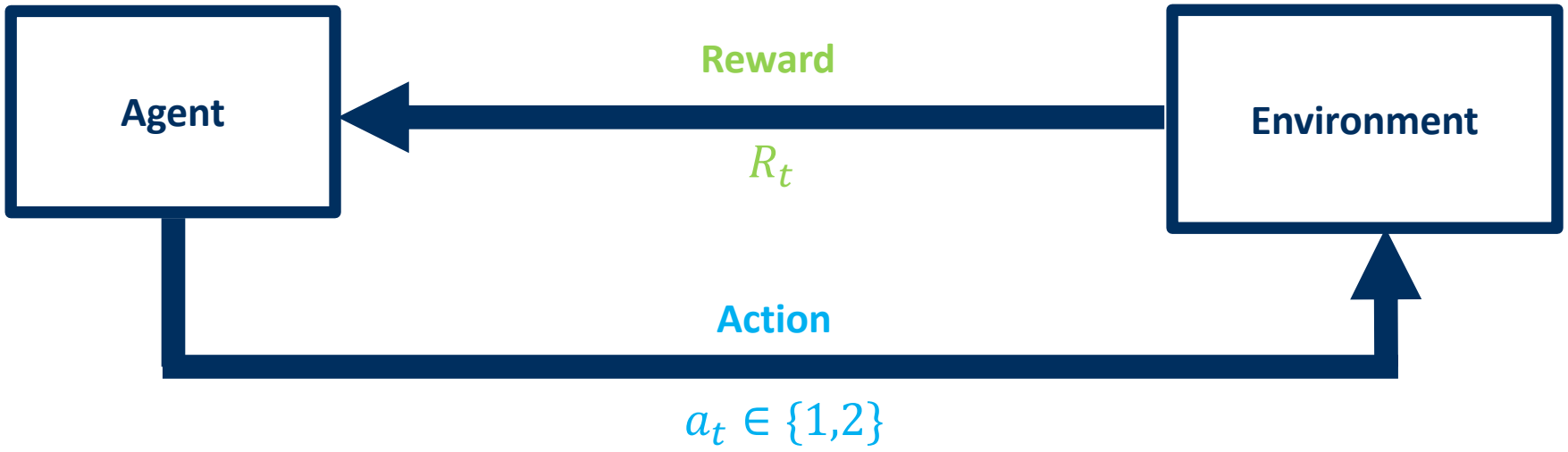
$$R_t = \begin{cases} \$10 & \text{with } \mathbf{30\%} \text{ prob.} \\ \$0 & \text{otherwise} \end{cases}$$

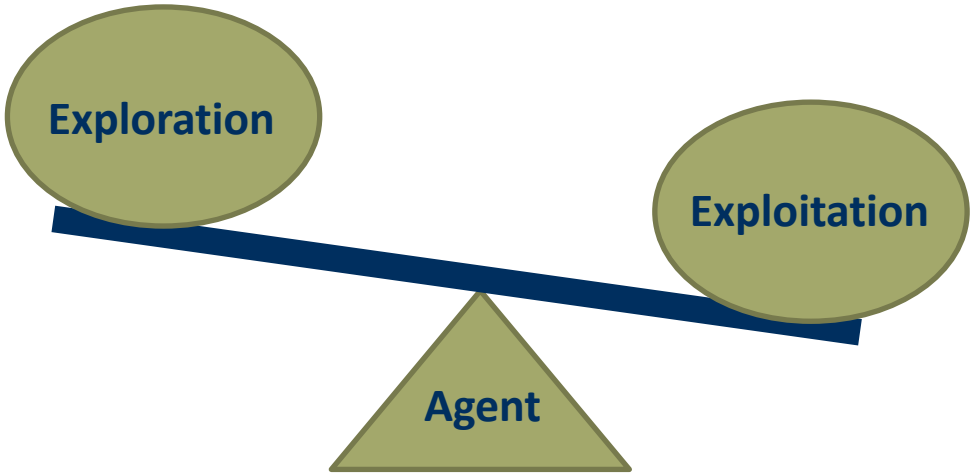
How to explore? How to exploit?





How to explore? How to exploit?

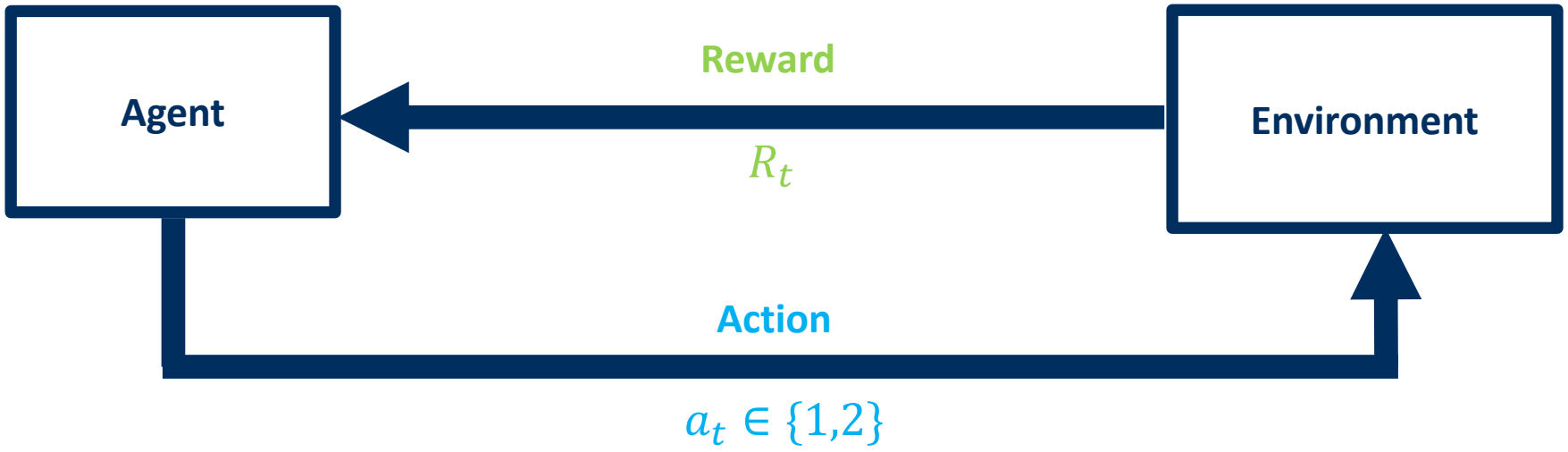




Optimize based on current info

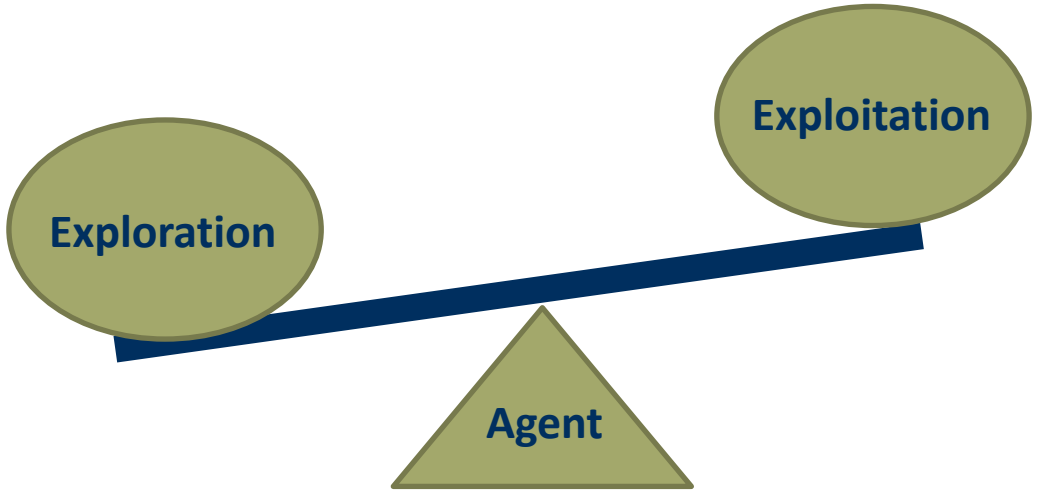
$$a_t = argmax_a \, \hat{Q}(a)$$

How to explore? How to exploit?



Try an uncertain action

Update $\hat{Q}(a_t)$



Optimize based on current info

$a_t = argmax_a \hat{Q}(a)$

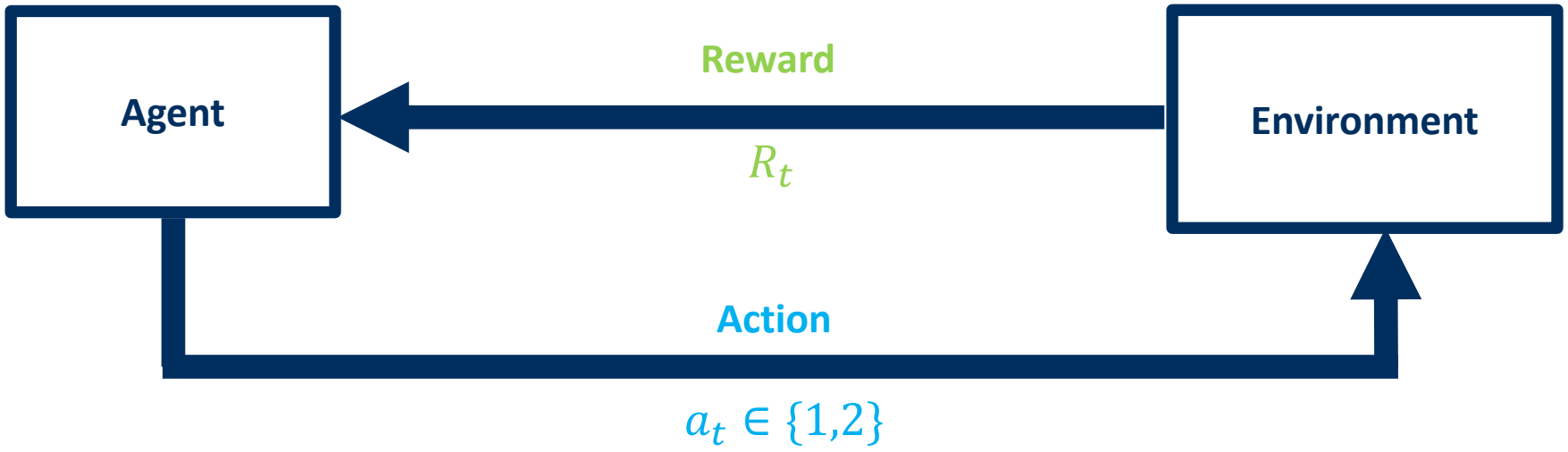
Main Characteristics of RL

- Multi-armed bandits - Setup
- Action Value
- Basic Algorithms
- Regret Analysis

Multi-Armed Bandit Algorithms

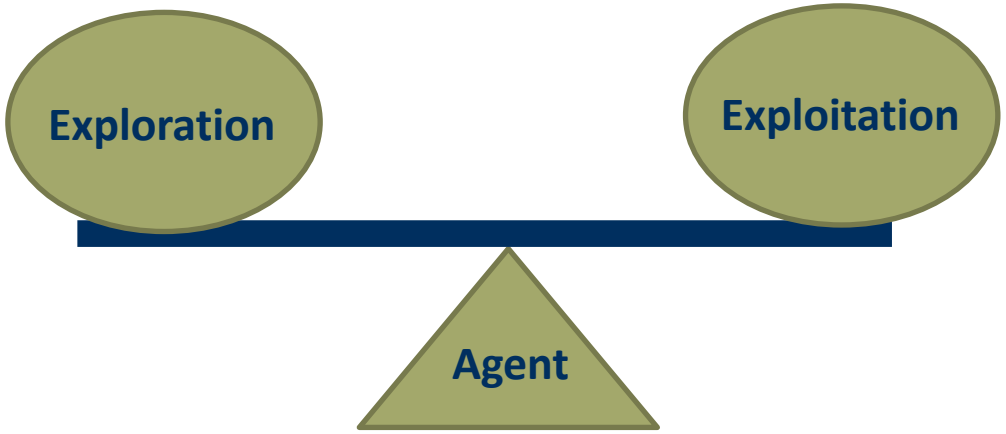
- ϵ-Greedy
- Upper Confidence Bounds (UCB) Algorithm

Simple Exploration: ϵ-greedy



With probability ϵ

Random action



With probability $1 - \epsilon$

$$a_t = argmax_a \, \hat{Q}(a)$$

Simple Exploration: ϵ-greedy

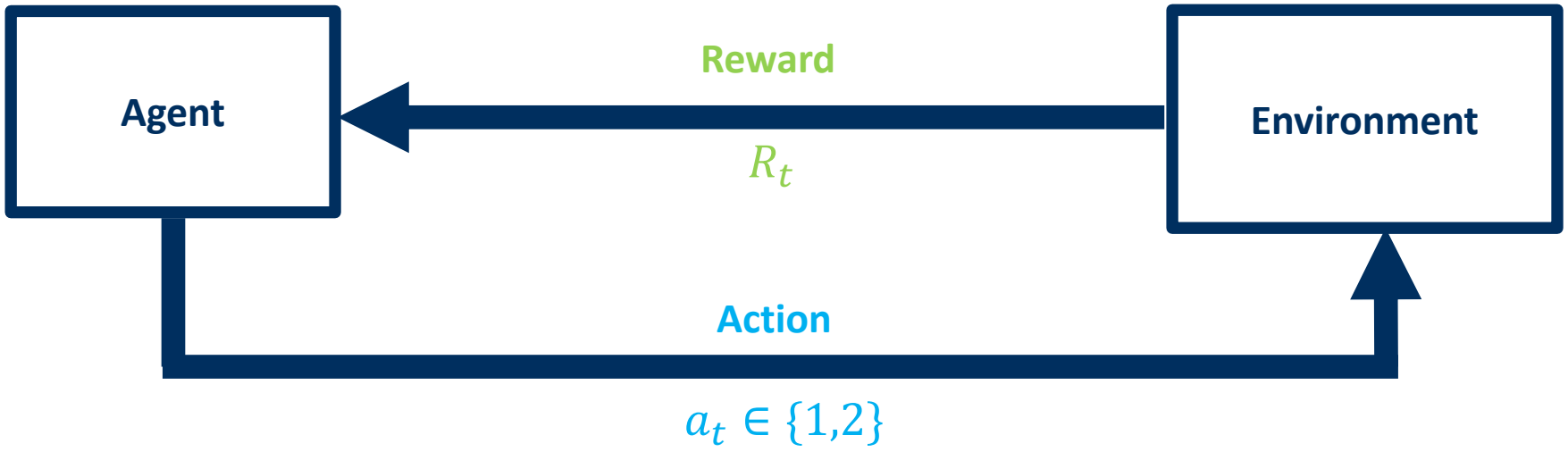
Stockholms universitet

For $t = 1,2,3, \ldots$

$$a_t = \begin{cases} \text{random} & \text{with probability } \epsilon \\ \arg\max_{a \in \{1,2\}} \hat{Q}_t(a) & \text{with probability } 1 - \epsilon \end{cases}$$

$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$

Simple Exploration: ϵ-greedy



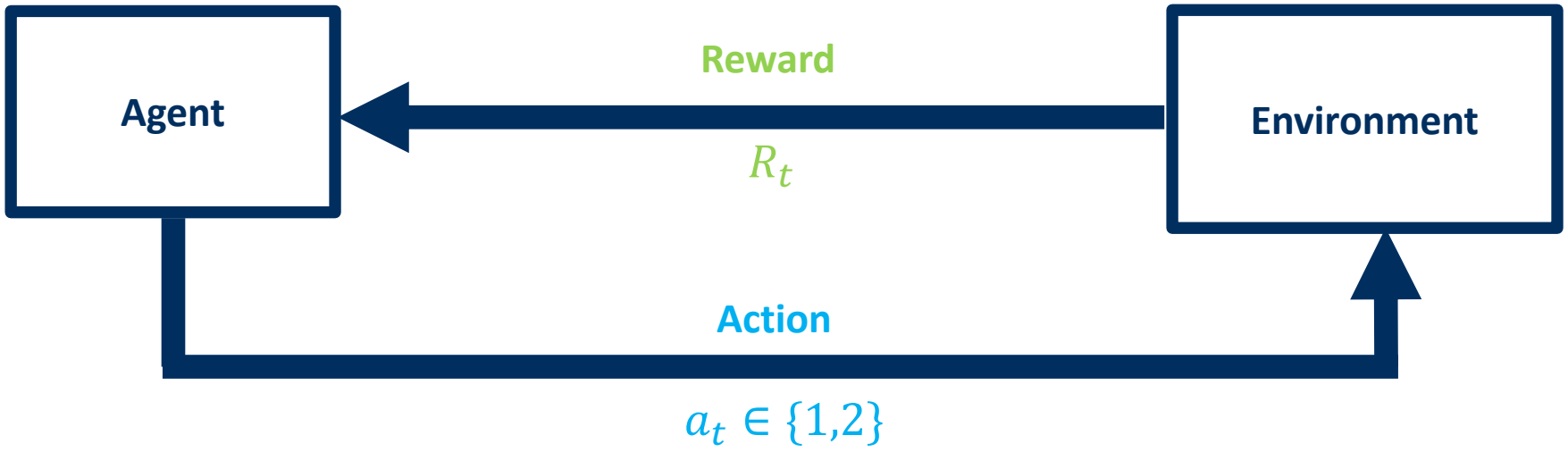
Problems:

- **Inefficient exploration:** Spend to much/little time exploring
- **Inefficient selection:** Spend to much/little time on clearly bad/good actions

Multi-Armed Bandit Algorithms

- ϵ-Greedy
- Upper Confidence Bounds (UCB) Algorithm

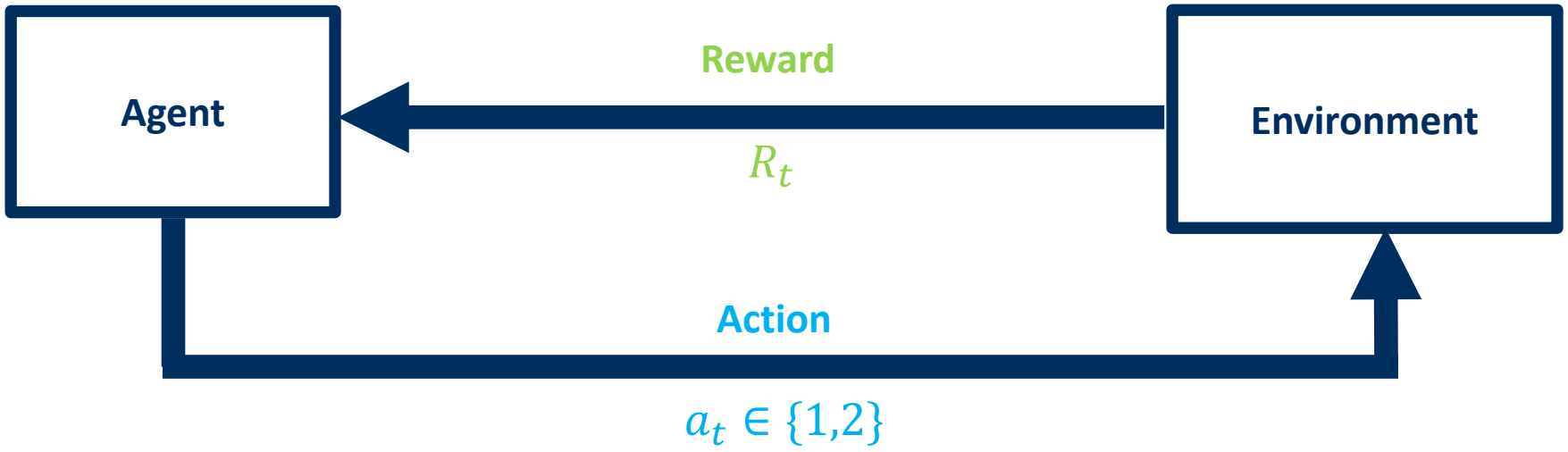
Upper Confidence Bounds (UCB) Algorithm



Take action a if

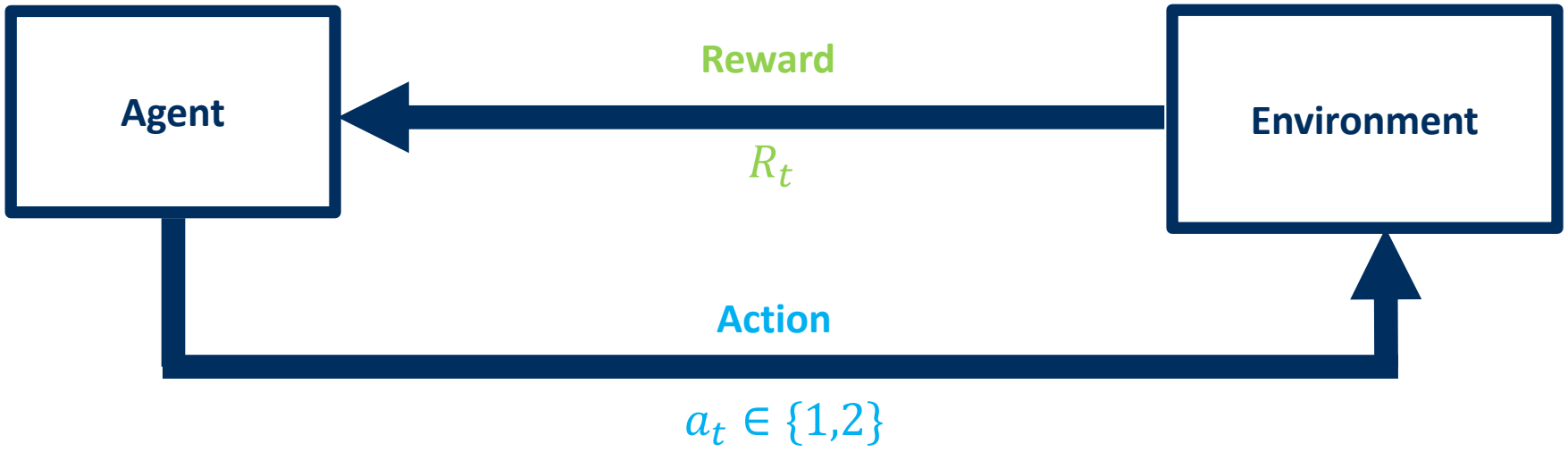
- Reward seems high for a, i.e., $\hat{Q}(a)$ is large
- We are not confident about about $\hat{Q}(a)$

Upper Confidence Bounds (UCB) Algorithm



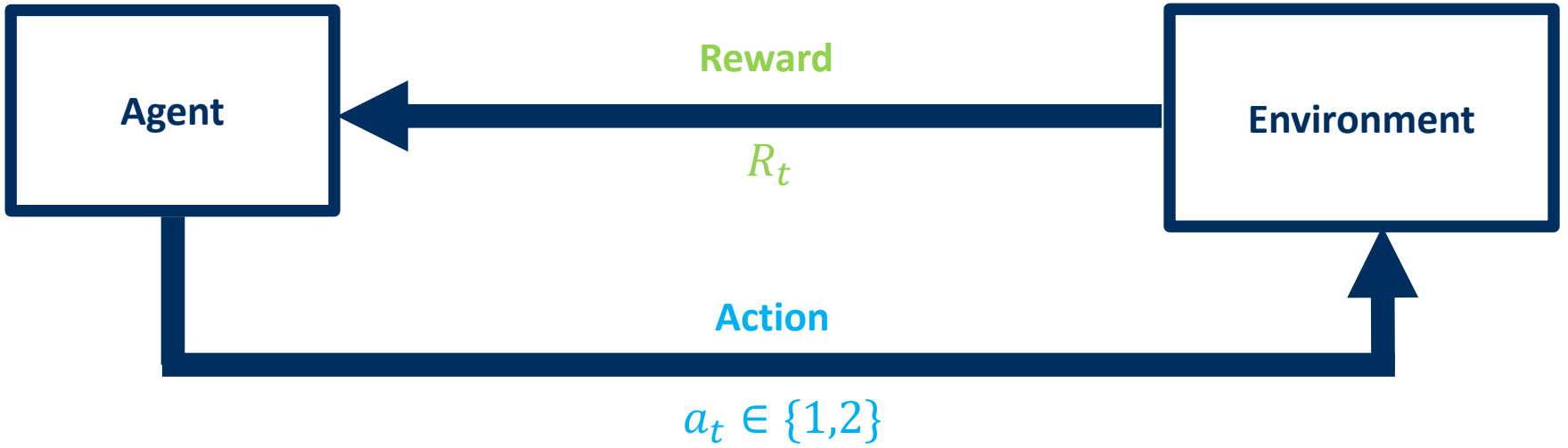
1. Optimism in face of uncertainty
2. Statistical confidence bounds

Optimisim in face of uncertainty



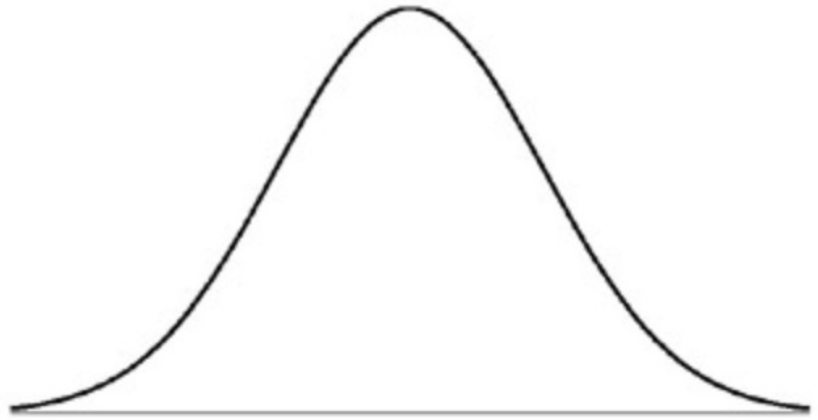
- Act as if the environment is as plausible as possible
- If unsure, then overestimate $\hat{Q}(a)$

Statistical upper confidence bounds

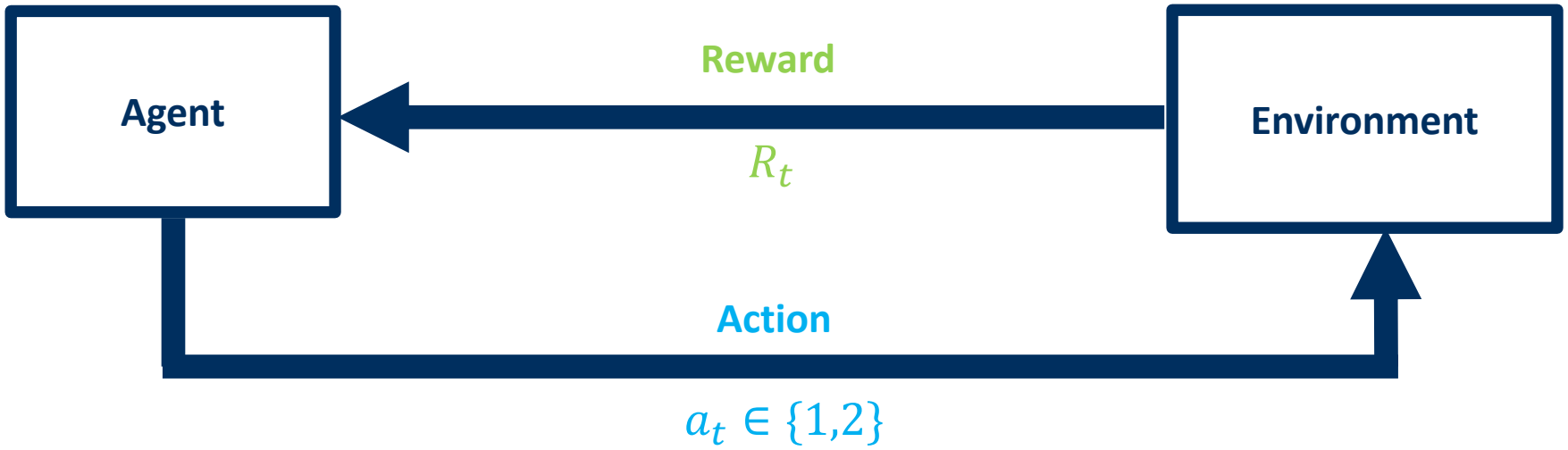


If $\bar{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i$ then

$$\mathbf{P}\left(\bar{x} \geq \bar{x}_n + \sqrt{\frac{2\log 1/\delta}{n}}\right) \leq \delta$$



Statistical upper confidence bounds



If $\bar{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i$ then

$$\mathbf{P}\left(\bar{x} \geq \bar{x}_n + \sqrt{\frac{2\log 1/\delta}{n}}\right) \leq \delta$$

$$\mathrm{UCB}(a,t,\delta) = \hat{Q}_t(a) + \sqrt{\frac{2\log 1/\delta}{N_t(a)}}$$

Upper Confidence Bounds (UCB) Algorithm

Stockholms universitet

For $t = 1,2,3, \ldots$

$$a_t = argmax_a \ \mathrm{UCB}(a, t, \delta)$$

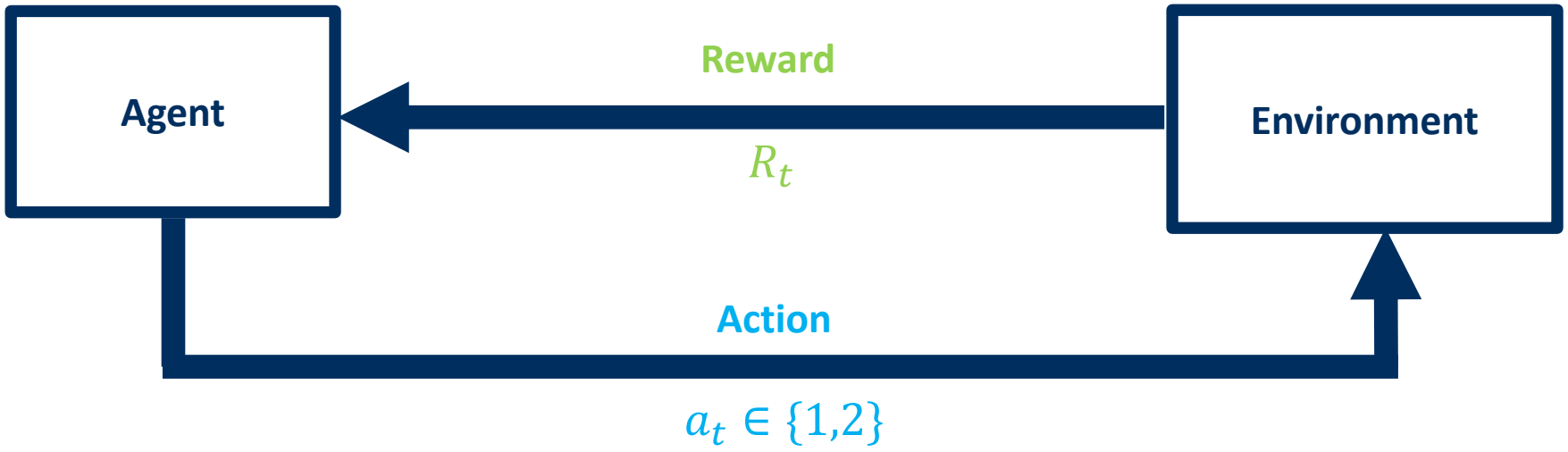
$$\hat{Q}_{t+1}(a_t) = \hat{Q}_t(a_t) + \frac{1}{N(a_t)}(R_t - \hat{Q}_t(a_t))$$

Main Characteristics of RL

- Multi-armed bandits - Setup
- Action Value
- Basic Algorithms
- Regret Analysis

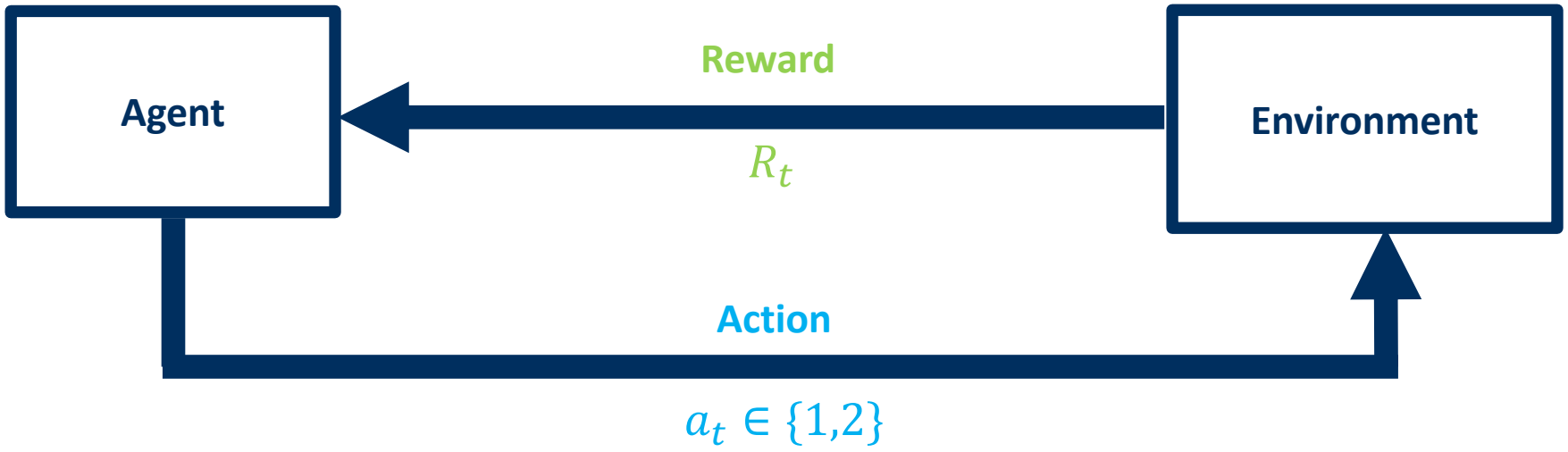
Stockholms universitet

How to mesure the exploration efficency of an Algorithm?



$$\textbf{Regret}(t) = t\mu_\star - \boldsymbol{E}\sum_{i=1}^{t} R_i$$

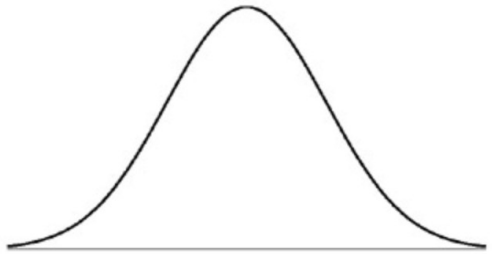
Environemnt



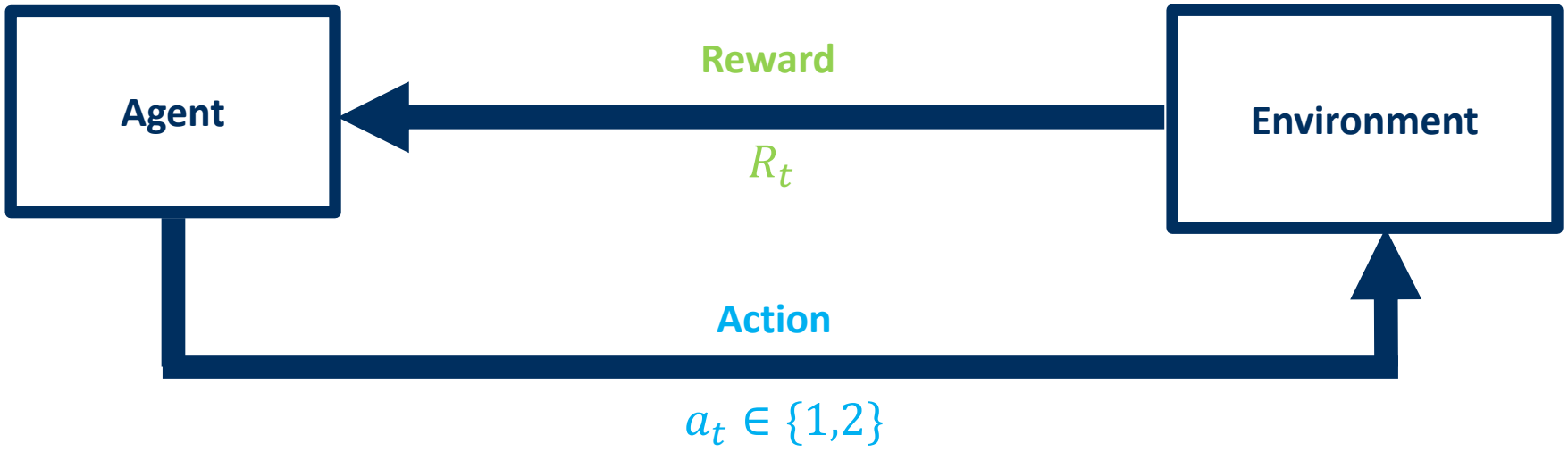
Environmental Assumption:

$$\boldsymbol{E}[R_t|a_t = 1] = \mu_1$$
$$\boldsymbol{E}[R_t|a_t = 2] = \mu_2$$

$$\mu_1 > \mu_2$$



Regret for ϵ-Greedy



ϵ-Greedy:

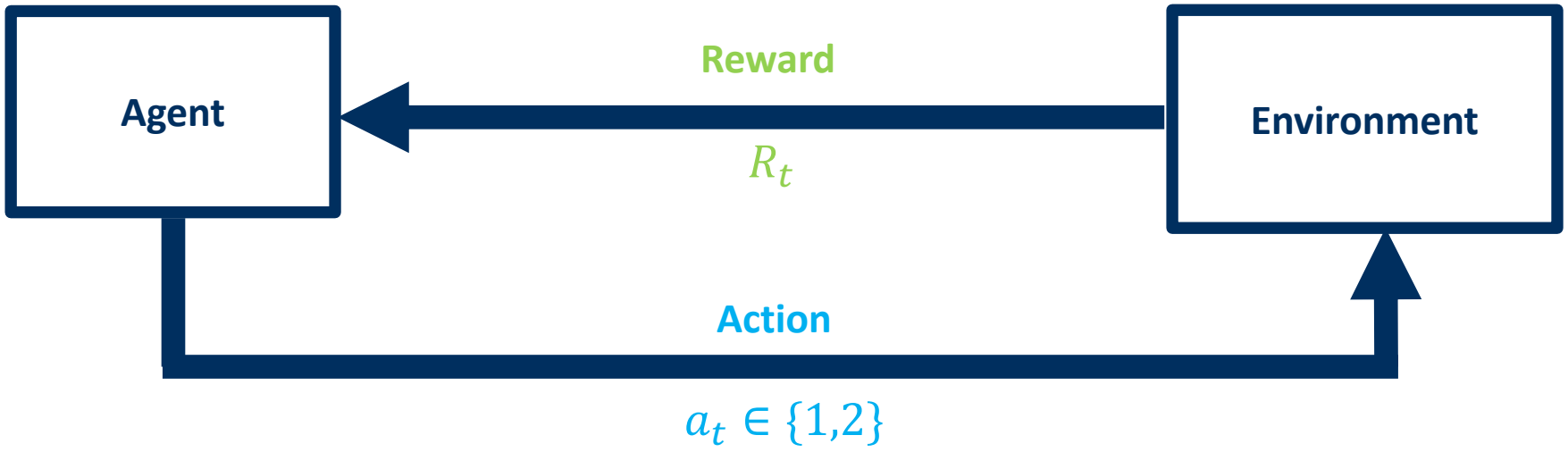
$$\textbf{Regret}(t) \geq \frac{\epsilon\Delta}{2} t$$

$$\textbf{Regret}(t) = \Omega(t)$$

Linear regret unavoidable ☹

$$\Delta = \mu_2 - \mu_1$$

Regret for UCB – Suboptimality Gap

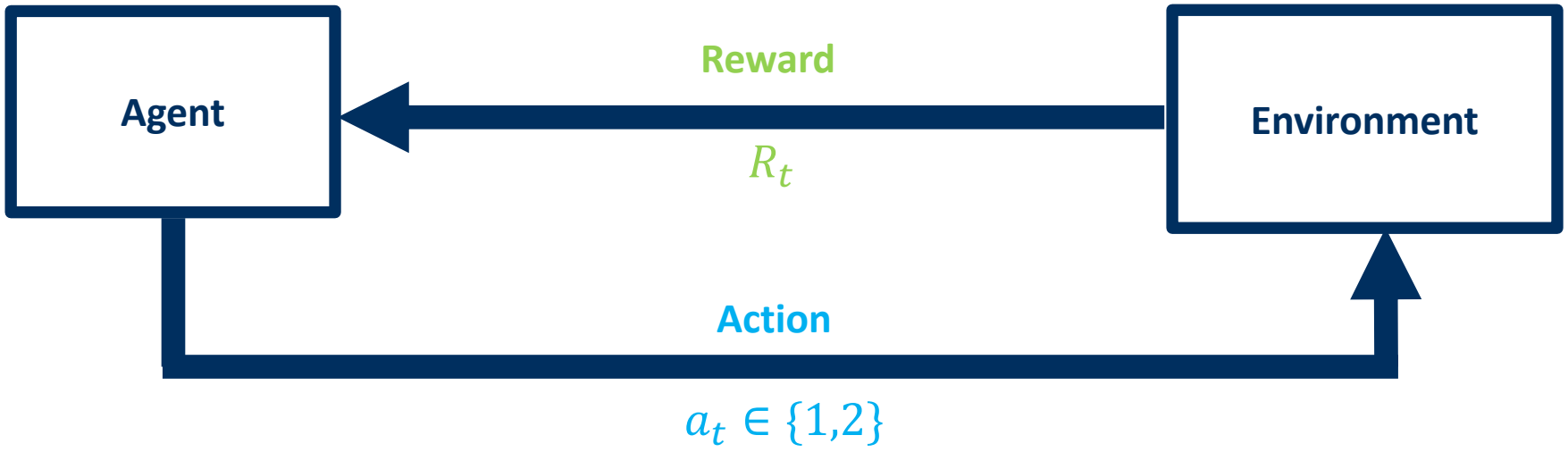


UCB Exploration:

$$\textbf{Regret}(t) \leq 3\Delta + \frac{16 \log t}{\Delta} = O(\log t)$$

$$\Delta = \mu_2 - \mu_1$$

Regret for UCB



UCB Exploration:

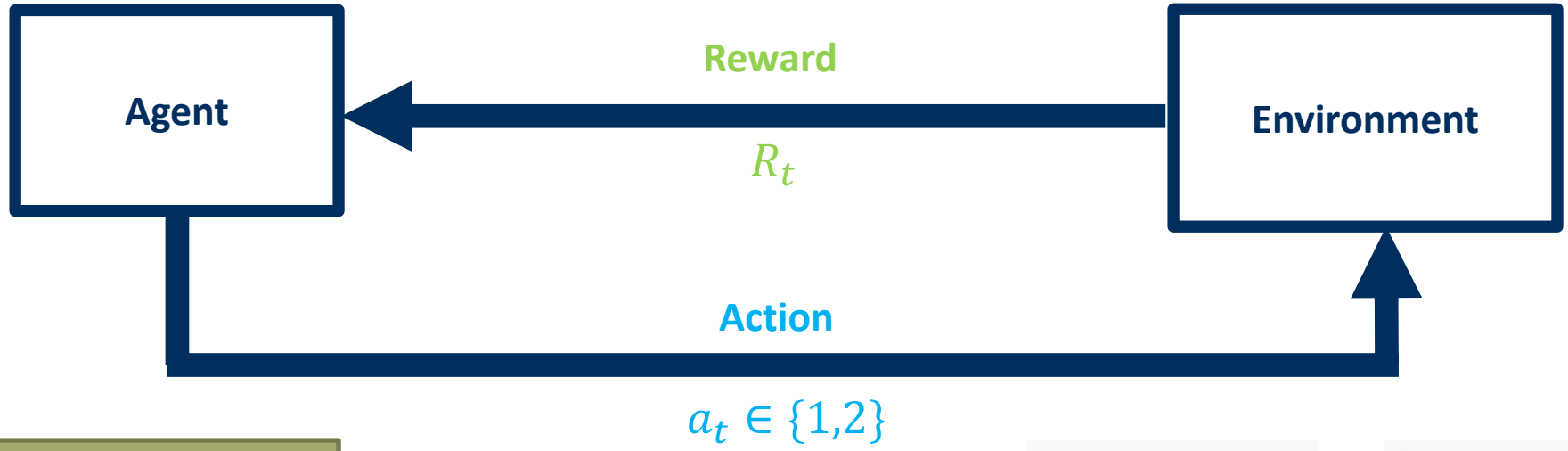
$$\textbf{Regret}(t) \leq 3\Delta + \sqrt{2t \log t}$$

Possible to improve to:

$$\textbf{Regret}(t) = O(\sqrt{t})$$

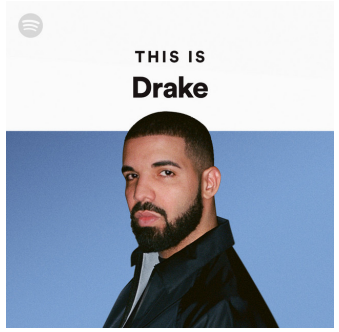
$$\Delta = \mu_2 - \mu_1$$

Next steps: Contextual Bandits



Context – How is listening?

- Age
- Demographics
- Browsing history



Artist 1

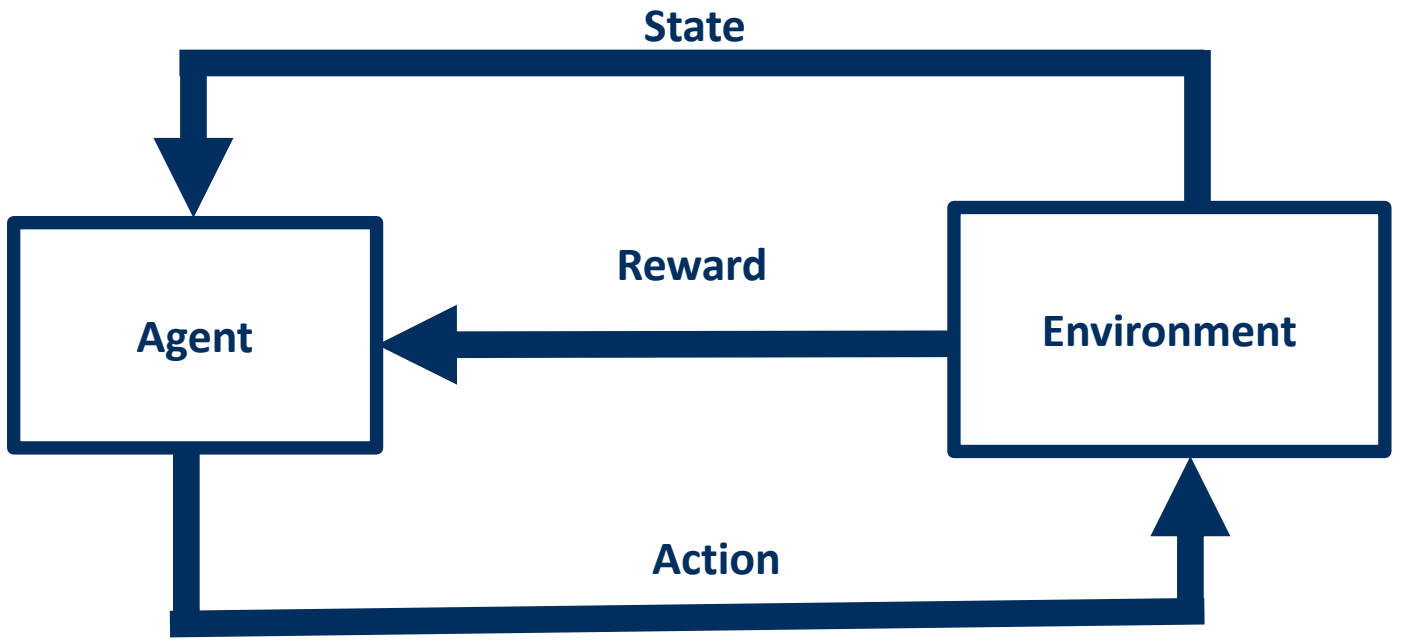
Artist 2

Overview

- Reinforcement Learning - Introduction
- Main Characteristics of RL and Data-Driven Decisions
- Stateless RL (multi-armed bandits): Exploration vs. Exploitation
- **Markov Decision Process: Delayed Consequences**
- Deep RL: Generalization
- Conclusions

Delayed Consequences (Covid Measures)

- **Agent:** Swedish Government
- **Action:** lockdown or not lockdown
- **Reward:**
  - Infected
  - Death-rate
  - Economic Impact
- **Goal:** Optimize total reward



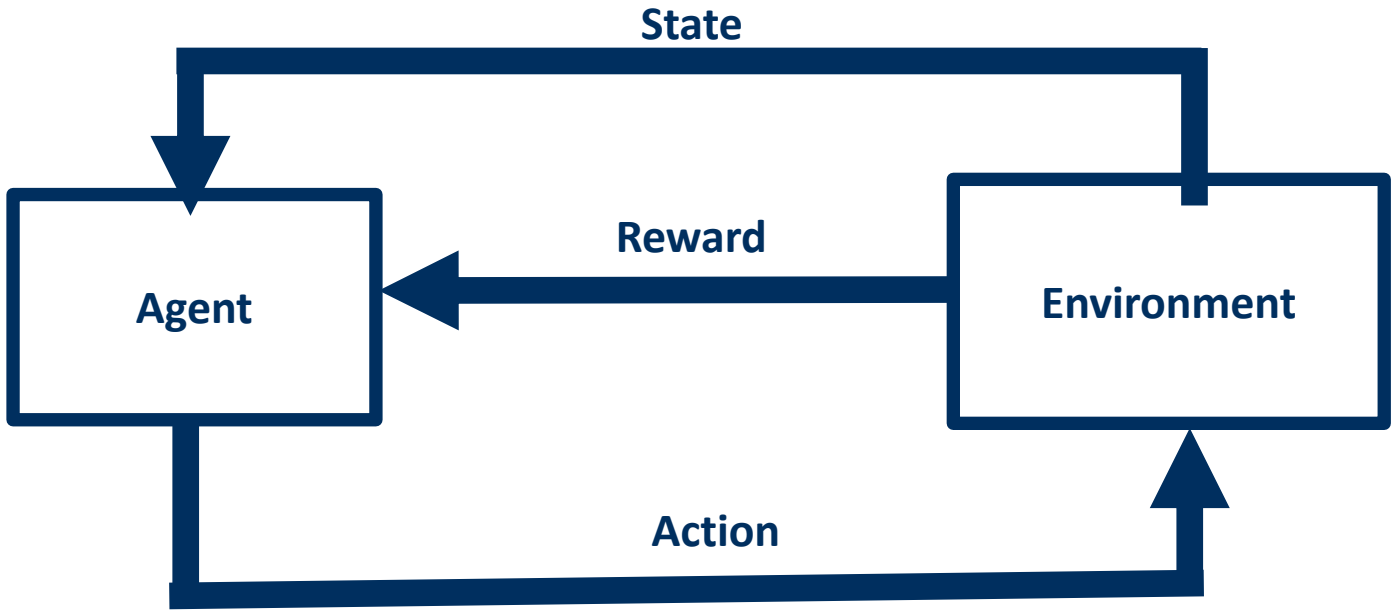
Lockdown

Not lockdown

Dynamics: Actions affect future states!

Delayed Consequences (Treatment Selection)

- **Agent:** Doctor
- **Action:** Treatment 1 or 2
- **Reward:**
  - 1 if successful
  - -1 if not successful
- **State:**
  - bloodpressure
  - previus medication
  - time since last medication
  - underlying diseases



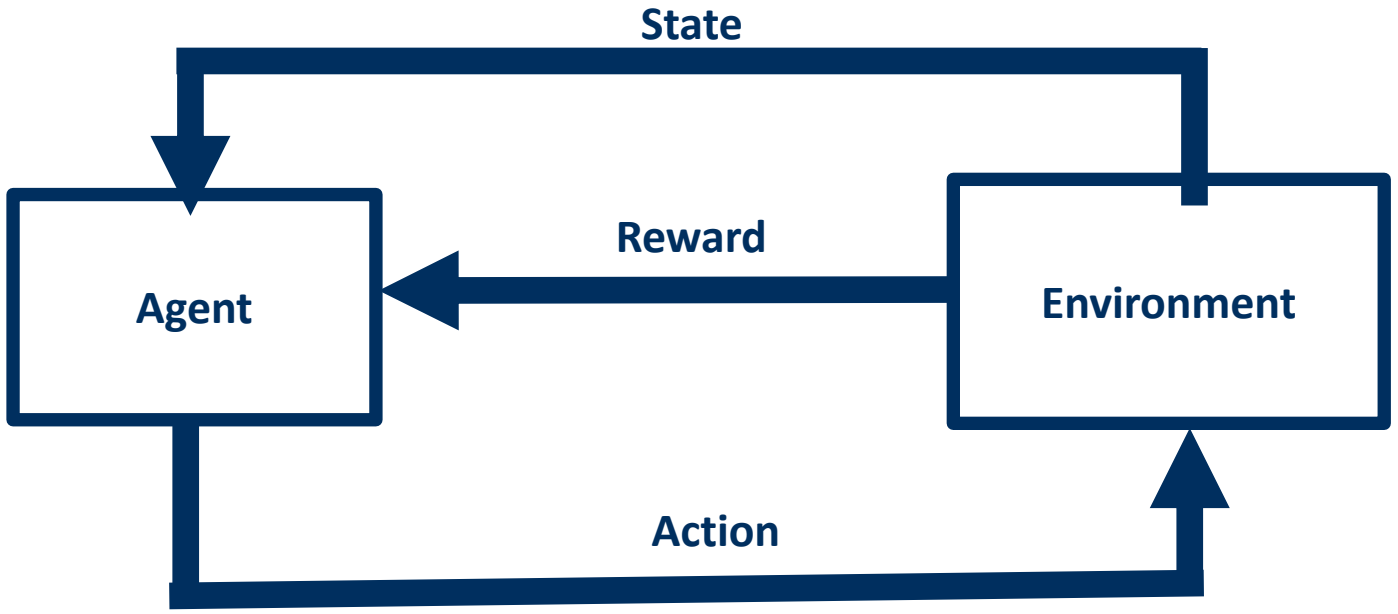
Treadment 1

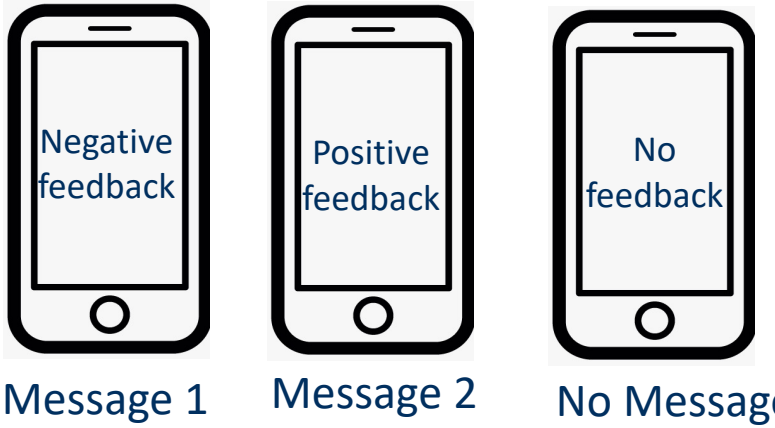
Treatment 2

No Treatment

Delayed Consequences (mHealth)

- **Agent:** mHealth app
- **Action:** Choose message 1 or 2
- **Reward:** +1/-1
  - Message 1, patient responds with 10% prob.
  - Message 2, patient responds with 30% prob.
- **State:**
  - How many notification were sent recently
  - heart rate
  - tiredness
  - mood

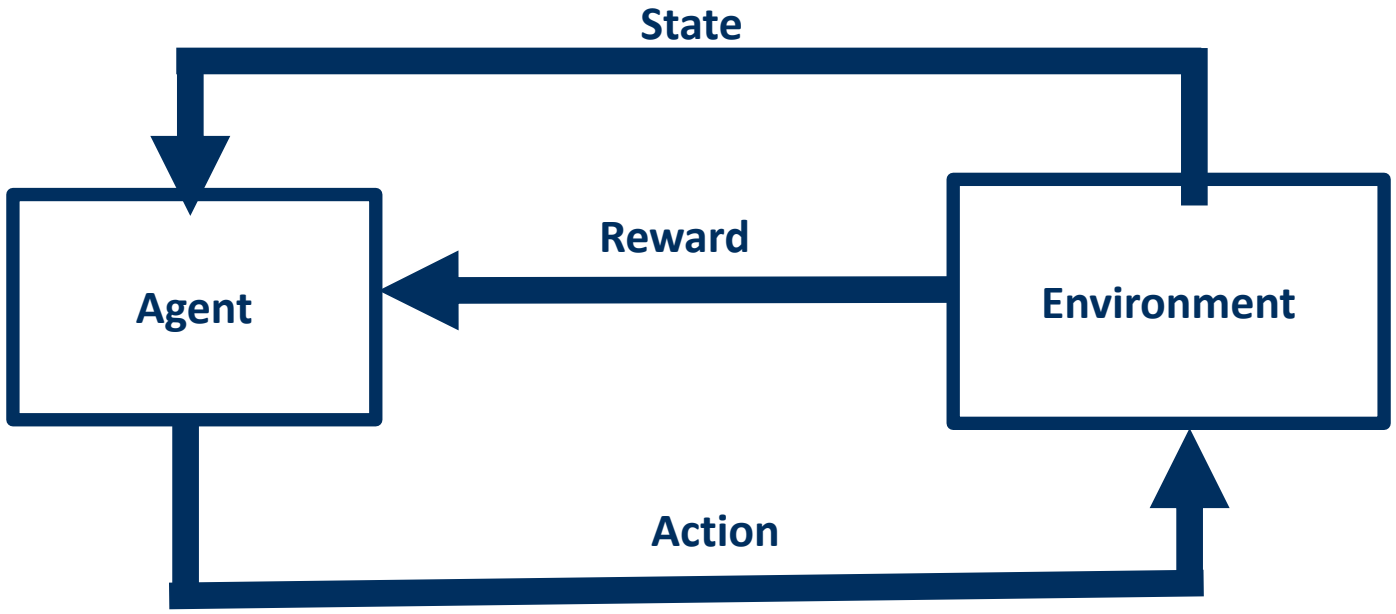




Markov Decision Process

- **Trajetory:** $s_0, a_0, R_1, s_1, a_1, R_2, \ldots$
- **Dynamics:** $p(s_{t+1}, |\underbrace{s_t, a_t, \ldots, s_0, a_0}_{\text{History}})$
- **Markovian Assumption:**

$$p(s_{t+1} |s_t, a_t, \ldots, s_0, a_0) = p(s_{t+1} |s_t, a_t)$$



Time	1	2	3	4	5
State	Happy	Happy	Happy	Sad	Sad
Action	Positive	Negative	none	Positive	Postive
Revard	0	0	1	1	1

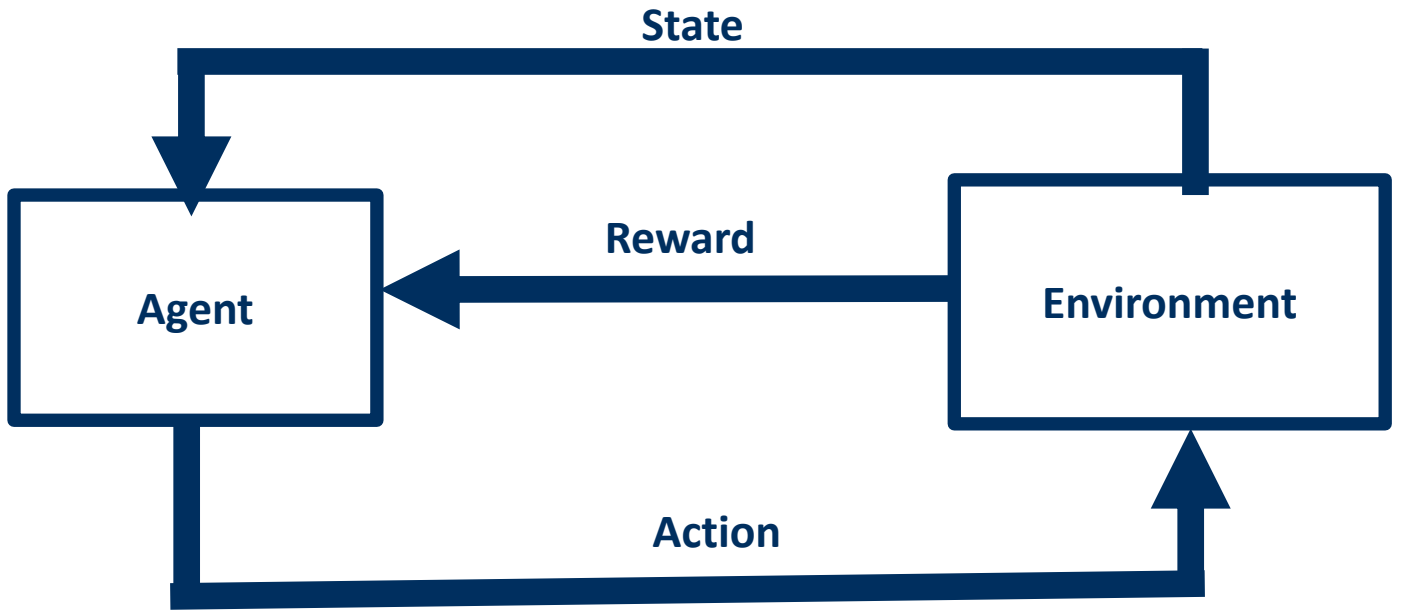
Message 1 Message 2 No Message

Markov Decision Process: Policy

- **Policy:** $\pi(s) = a$ with distribution $p^{\pi}(a|s)$

π(Happy)=
- Positive with probability 0.25
- Negative with probability 0.25
- None with probability 0.5

π(sad)=
- Positive with probability 0.5
- Negative with probability 0.25
- None with probability 0.25

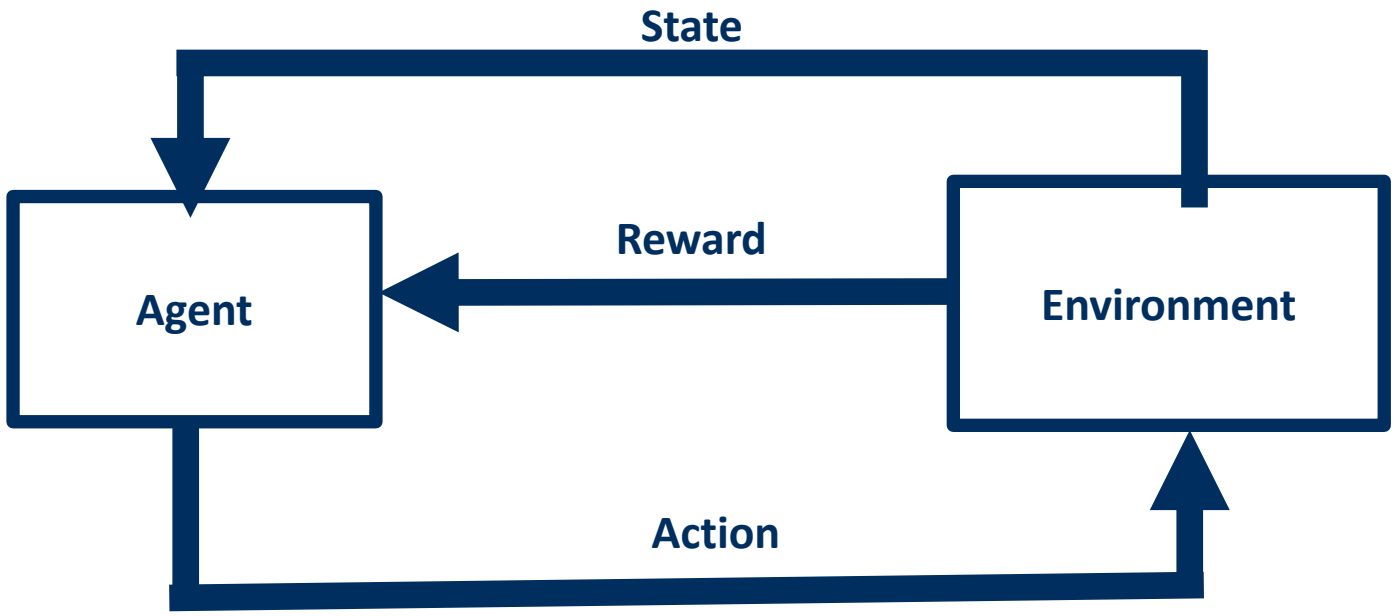


Goal: Find Optimal $\pi(\cdot)$

$$\max_{\pi} \sum_{t} R_t$$

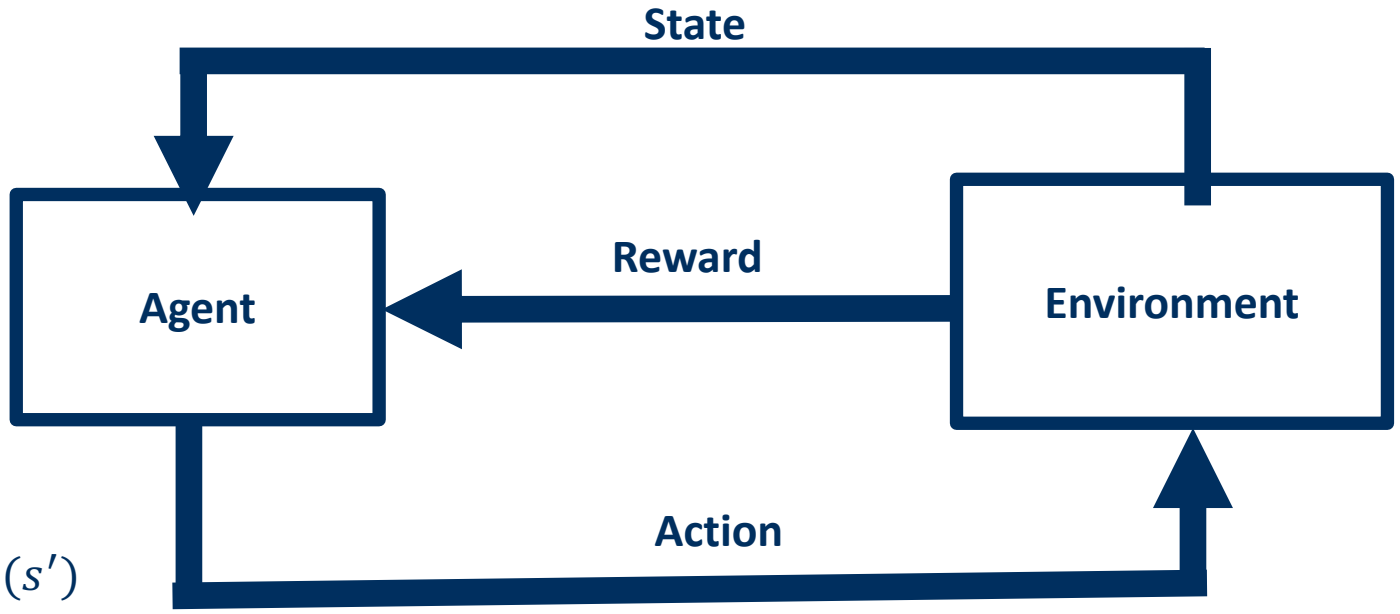
Markov Decision Process

- **Policy:** $\pi(s) = a$ with distribution $p^{\pi}(a|s)$
- **Reward:** R_t
  - $r(s,a) = \boldsymbol{E}[R_{t+1}|s_t = s, a_t = a]$
- **Value:**
  - $V^{\pi}(s) = \boldsymbol{E}\left[\sum_{t=0}^{\infty} R_t \mid s_0 = s\right]$
- **Action-value:**
  - $Q^{\pi}(s,a) = \boldsymbol{E}\left[\sum_{t=0}^{\infty} R_t \mid s_0 = s, a_0 = a\right]$



Dynamic Programming: Known Dynamics and Rewards

- **Know:**
  - $p(s_{t+1}, R_{t+1}|s_t, a_t)$
  - $r(s, a)$
- **Goal:** Learn optimal policy
  - $\pi(\cdot)$
  - $p^{\pi}(a|s)$
- **Optimal Bellmann Equation:**
  - $V^*(s) = \max_a r(s, a) + \sum_{s'} p(s'|s, a)V^*(s')$



Dynamic Programming: Known Dynamics and Rewards

Stockholms universitet

- **Know:**
  - $p(s_{t+1}, R_{t+1}|s_t, a_t)$
  - $r(s, a)$
- **Goal:** Learn optimal policy
  - $\pi(\cdot)$
  - $p^{\pi}(a|s)$
- **Optimal Bellmann Equation:**
  - $V^*(s) = \max_a r(s, a) + \sum_{s'} p(s'|s, a)V^*(s')$

- **Value Iteration:**

  Initialize Value function: $\hat{V}_0(s)$ for all s

  for $i = 1,2,3 \ldots$
  for s$\in S$
  $\hat{V}_{i+1}(s) = \max_a r(s, a) + \sum_{s'} p(s'|s, a)\, \hat{V}_i(s')$

- **Converges to $V^*(\cdot)$:**

$$\lim_{i \to \infty} \hat{V}_i(\cdot) = V^*(\cdot)$$

- **Optimal Policy:**

$$\pi(s) = arg\max_a r(s, a) + \sum_{s'} p(s'|s, a)\, \hat{V}_{end}(s')$$

Dynamic Programming: Known Dynamics and Rewards

Stockholms universitet

- **Know:**
  - $p(s_{t+1}, R_{t+1}|s_t, a_t)$
  - $r(s, a)$
- **Goal:** Learn optimal policy
  - $\pi(\cdot)$
  - $p^{\pi}(a|s)$
- **Optimal Bellmann Equation:**
  - $V^*(s) = \max_a r(s, a) + \sum_{s'} p(s'|s, a)V^*(s')$

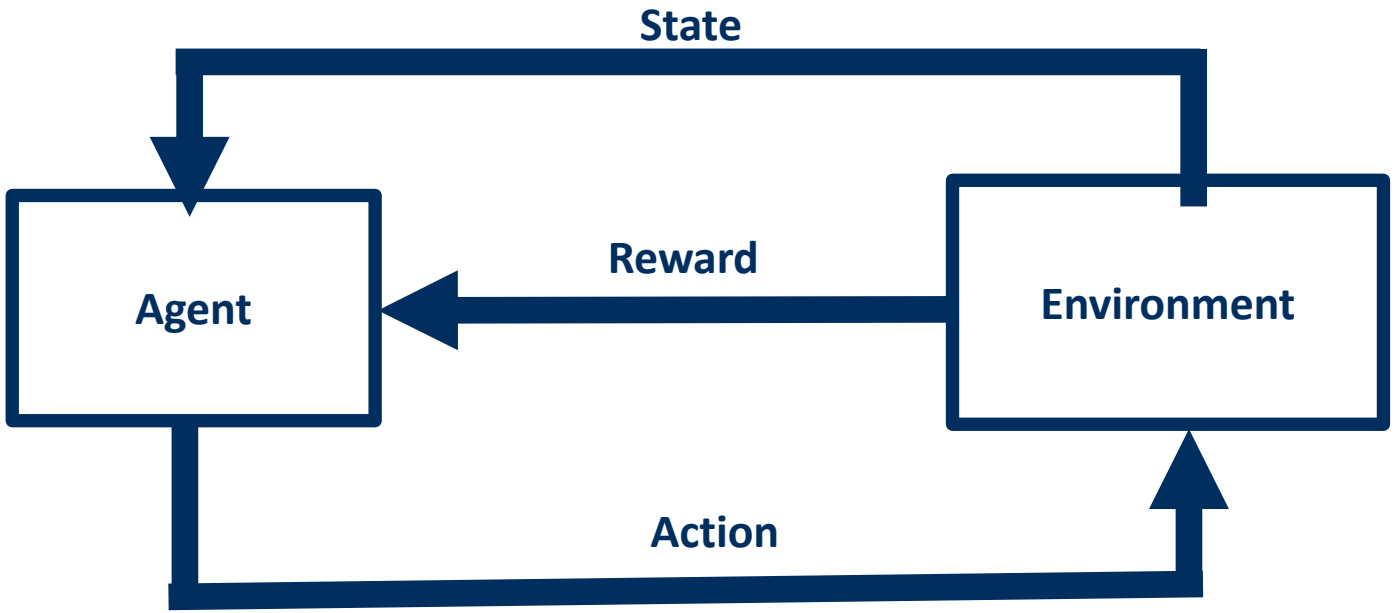
Problem:

We usually don't know the dynamics or the rewards

Dynamic Programming: Unknown Dynamics/Rewards

Stockholms universitet

- **Do not Know:**
  - $p(s_{t+1}, R_{t+1}|s_t, a_t)$
  - $r(s, a)$
- **Goal:** Learn optimal policy
  - $\pi(\cdot)$
  - $p^{\pi}(a|s)$
- **Optimal Action Value Function:**
  - $Q^*(s, a) = \quad r(s, a) + \max_{a'} \sum_{s'} p(s'|s, a) Q^*(s', a')$



Dynamic Programming: Unknown Dynamics/Rewards

Stockholms universitet

- **Know:**
  - $p(s_{t+1}, R_{t+1}|s_t, a_t)$
  - $r(s, a)$
- **Goal:** Learn optimal policy
  - $\pi(\cdot)$
  - $p^{\pi}(a|s)$
- **Optimal Action Value Function:**
  - $Q^*(s,a) = r(s,a) + \max_a \sum_{s'} p(s'|s,a) Q^*(s',a')$

- **Q Learning:**

  Initialize Value function: $\hat{Q}\ \ (s,a)$ for all s and a
  Choose initial state s_0

  for t= 0,1,2,3 …
  Coose action a_i from $\hat{Q}$ (e.g. ϵ Greedy)
  Take action a_t and observe R_{t+1} and s_{t+1}

  $$\hat{Q}(s_i, a_i) = \hat{Q}(s_i, a_i) + \alpha[R_{t+1} + \max_a \hat{Q}(s_{t+1}, a) - \hat{Q}(s_t, a_t)]$$

- **Converges to $Q^*(\cdot)$:**

  $$\lim_{t\to\infty} \hat{Q}\ \ (\cdot) = Q^*(\cdot)$$

- **Optimal Policy:**

  $$\pi(s) = arg\max_a\ \hat{Q}(s,a)$$

Dynamic Programming: Unknown Dynamics/Rewards

Stockholms universitet

- **Know:**
  - $p(s_{t+1}, R_{t+1}|s_t, a_t)$
  - $r(s, a)$
- **Goal:** Learn optimal policy
  - $\pi(\cdot)$
  - $p^{\pi}(a|s)$
- **Optimal Action Value Function:**
  - $Q^*(s, a) = r(s, a) + \max_a \sum_{s'} p(s'|s, a) Q^*(s', a')$

Challenge:

State/Action-Spaces are usually Huge!

Overview

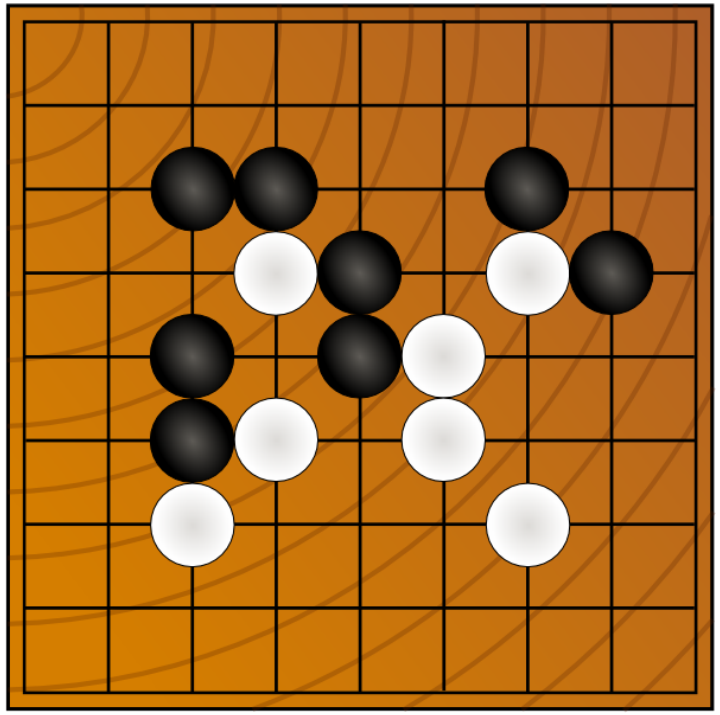
Stockholms universitet

- Reinforcement Learning - Introduction
- Main Characteristics of RL and Data-Driven Decisions
- Stateless RL (multi-armed bandits): Exploration vs. Exploitation
- Markov Decision Process: Delayed Consequences
- **Deep Reinforcement Learning: Generalization**
- Conclusions

Deep RL: Generalization

- State-spaces often huge

$(256^{100\times300})^3$ states



3^{361} states

Deep RL: Generalization

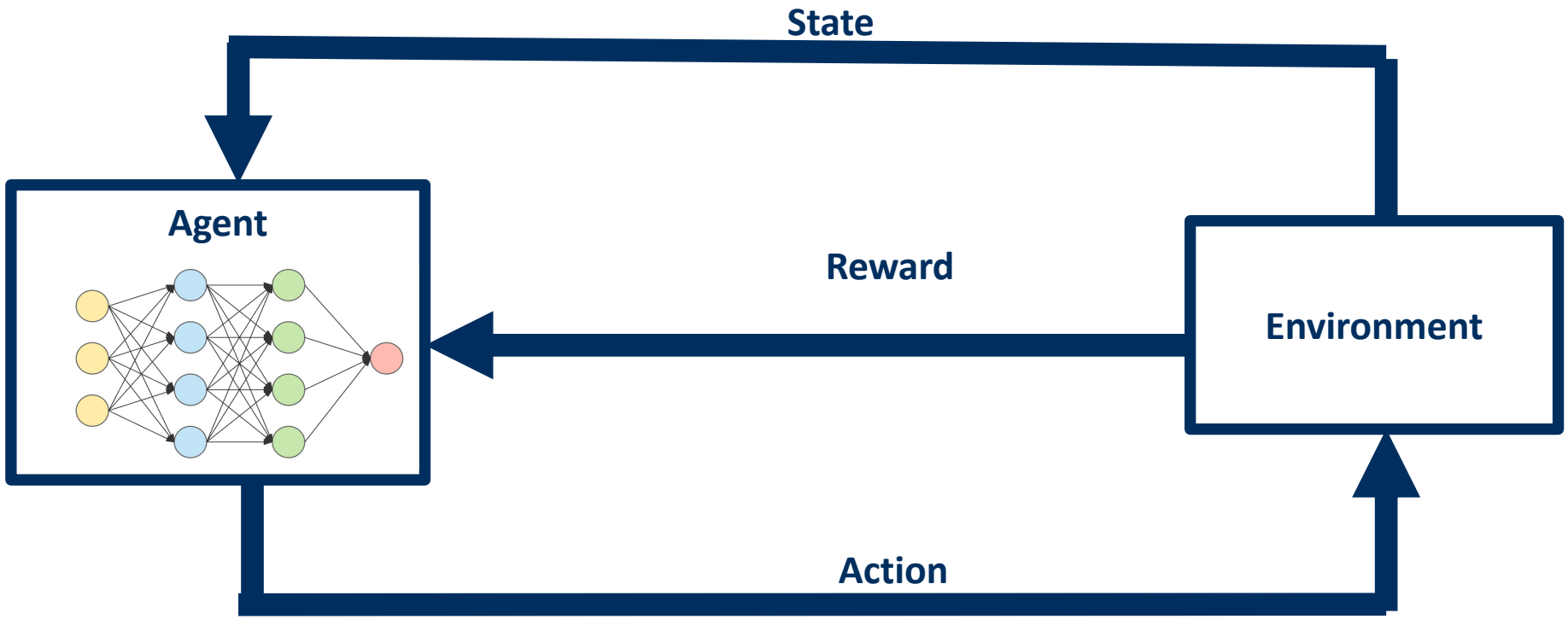
Stockholms universitet

- State-spaces grows exponentially with features

			Number of features	Number of states
Blood-pressure	High	Low	1	2
Gender	Man	Woman	2	$2^2 = 4$
Age	Old	Young	3	$2^3 = 8$
Heart-rate	High	High	4	$2^4 = 16$
-	-	-	5	$2^5 = 32$
⋮	⋮	⋮	⋮	⋮
-	-	-	20	$2^{20} = 1048576$

Deep RL: Generalization

- Cannot learn $Q^*(s, a)$ by trying all state action pairs!
- **Solution:** Learn approximate $Q^*(s, a)$ by a deep neural network
- Train $Q_w(s, a) \approx Q^*(s, a)$ where w are the weights of a neural networ

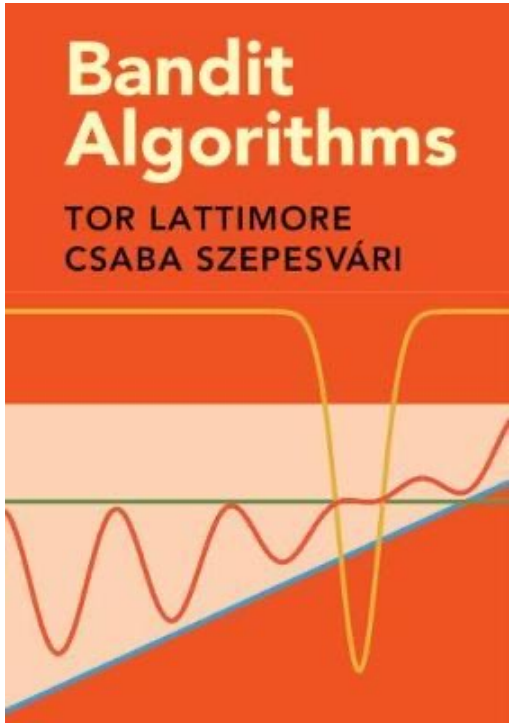


Overview

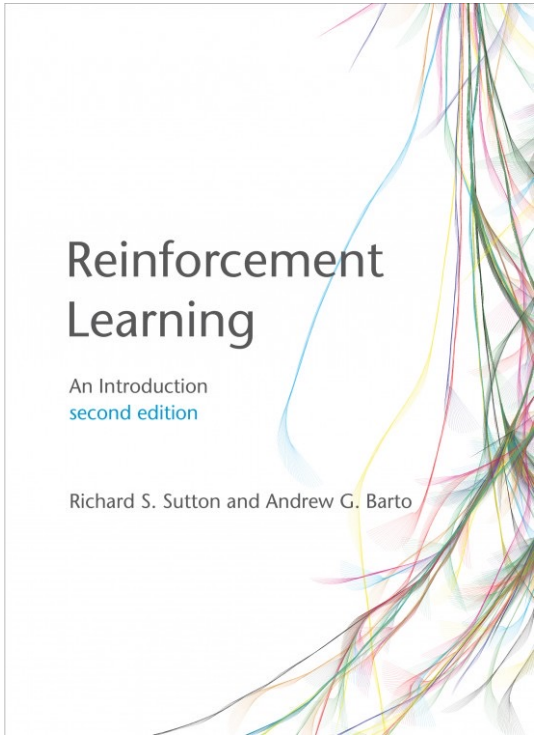
Stockholms universitet

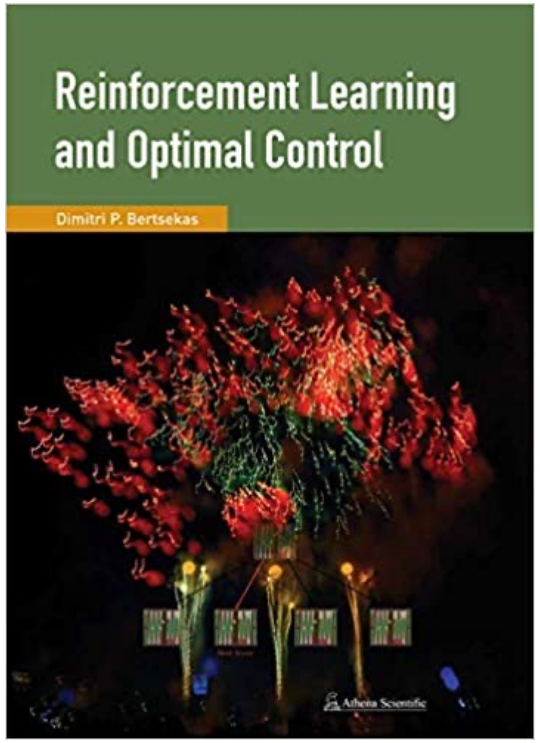
- Reinforcement Learning - Introduction
- Main Characteristics of RL and Data-Driven Decisions
- Stateless RL (multi-armed bandits): Exploration vs. Exploitation
- Markov Decision Process: Delayed Consequences
- Deep Reinforcement Learning: Generalization
- **Conclusions**

Literature


Bandit Algorithms
TOR LATTIMORE
CSABA SZEPESVÁRI


Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems


Reinforcement Learning
An Introduction
second edition
Richard S. Sutton and Andrew G. Barto


Reinforcement Learning and Optimal Control
Dimitri P. Bertsekas