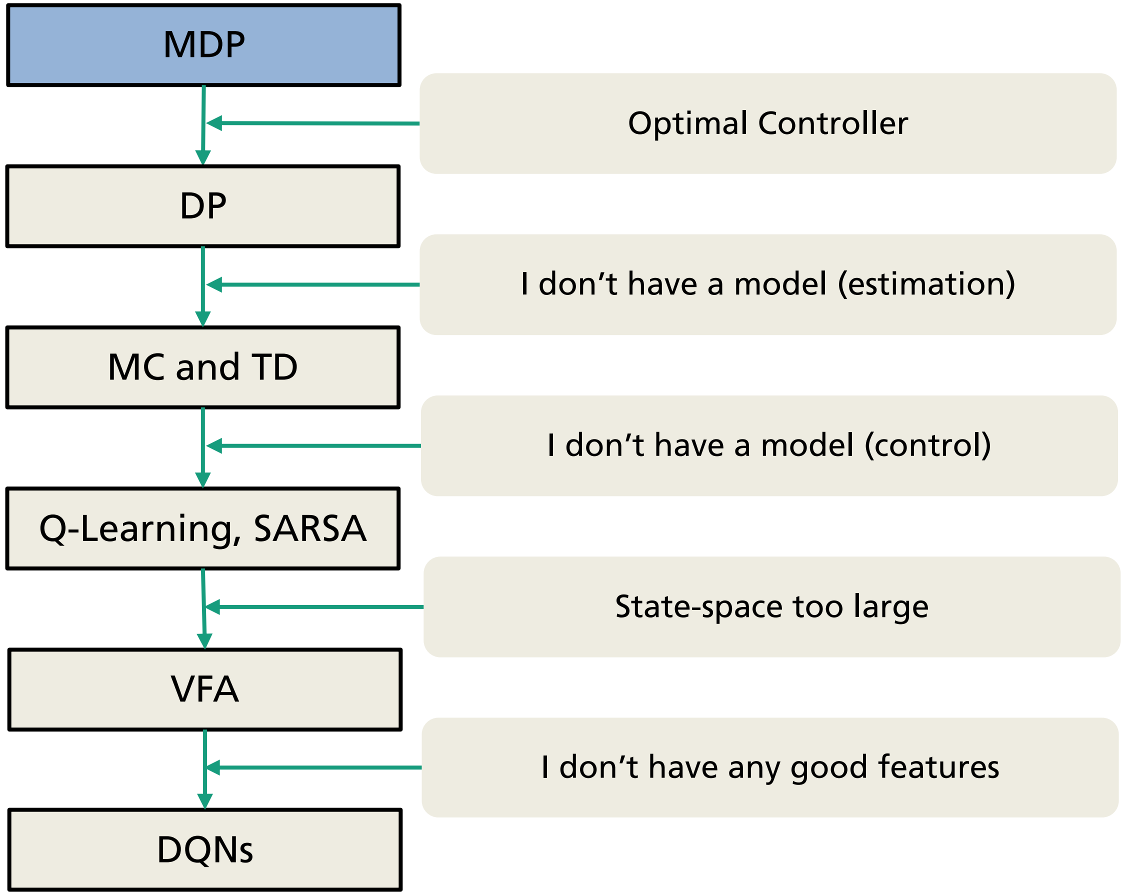
Markov Decision Processes

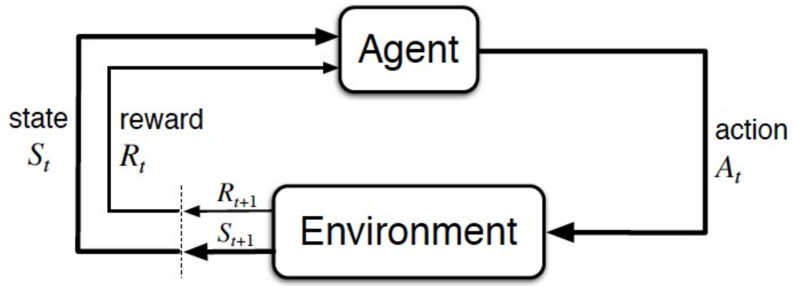
Christopher Mutschler

Overview


MDP
Optimal Controller
DP
I don’t have a model (estimation)
MC and TD
I don’t have a model (control)
Q-Learning, SARSA
State-space too large
VFA
I don’t have any good features
DQNs

Markov Decision Processes

- Agent learns by interacting with an environment over many time-steps:
- Markov Decision Process (MDP) is a tool to formulate RL problems
  - Description of an MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:



Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

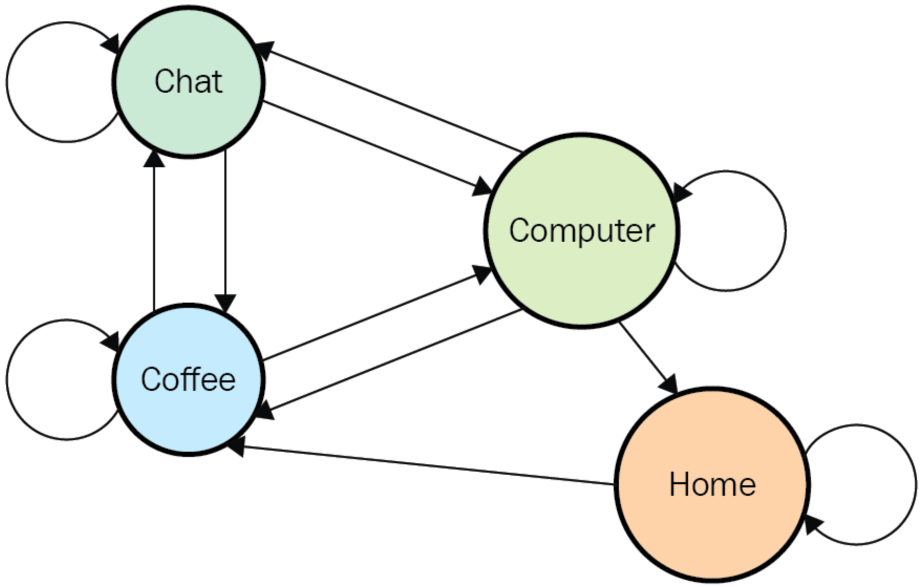
- At each step t, the agent:
  - is at state S_t,
  - performs action A_t,
  - receives reward R_t.

- At each step t, the environment:
  - receives action A_t from the agent,
  - provides reward R_t,
  - moves at state S_{t+1},
  - increments time $t \leftarrow t + 1$.

Note:
If the interaction does stop at some point in time (T) then we have an *episodic RL problem.*

Markov Decision Processes

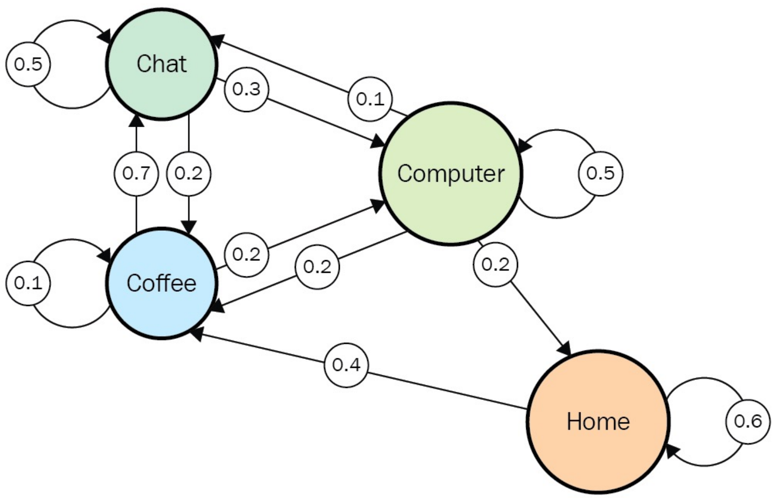
- Markov Process (MP)
  - Description of an MP $(\mathcal{S}, \mathcal{P})$:



Lapan, M. (2018). Deep Reinforcement Learning Hands-On. Packt Publishing Ltd.

Markov Decision Processes

- Markov Process (MP)
  - Description of an MP $(\mathcal{S}, \mathcal{P})$:



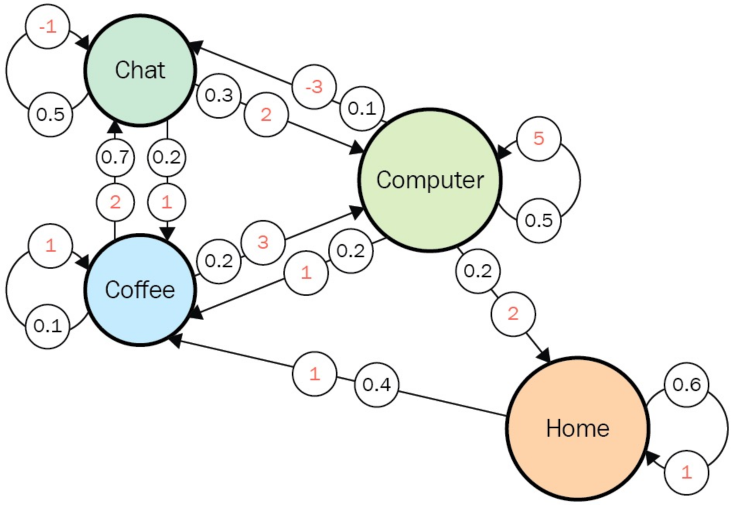
	Home	Coffee	Chat	Computer
Home	60%	40%	0%	0%
Coffee	0%	10%	70%	20%
Chat	0%	20%	50%	30%
Computer	20%	20%	10%	50%

Lapan, M. (2018). Deep Reinforcement Learning Hands-On. Packt Publishing Ltd.

Markov Decision Processes

- Markov Reward Process (MRP)
  - Description of an MRP $(\mathcal{S}, \mathcal{P}, \mathcal{R})$:
  - $\mathcal{R}$ is a reward function:

$$\mathcal{R}_s = \mathbb{E}[R_{t+1} | S_t = s]$$

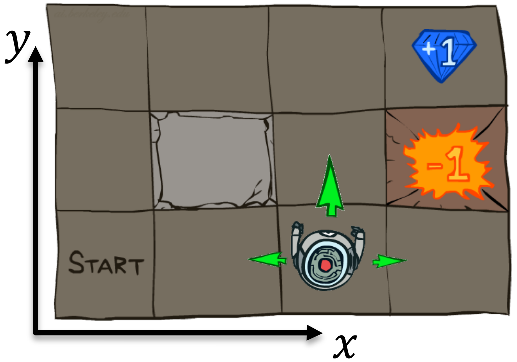


	Home	Coffee	Chat	Computer
Home	1	1		
Coffee		1	2	3
Chat		1	-1	2
Computer	2	1	-3	5

Lapan, M. (2018). Deep Reinforcement Learning Hands-On. Packt Publishing Ltd.

Markov Decision Processes

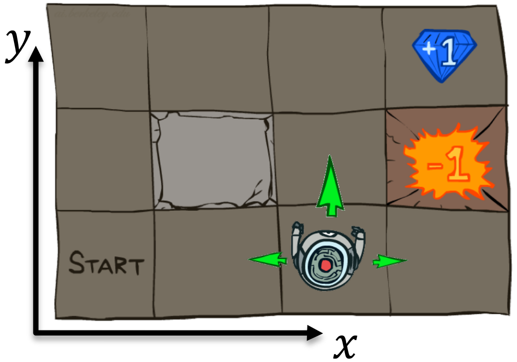
- Markov Decision Process (MDP)
  - Description of an MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:

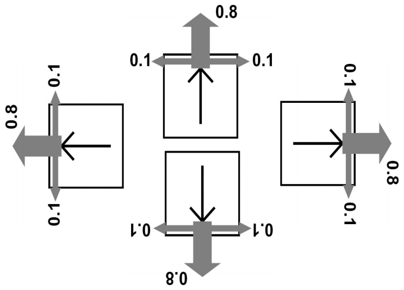


http://ai.berkeley.edu/lecture_slides.html

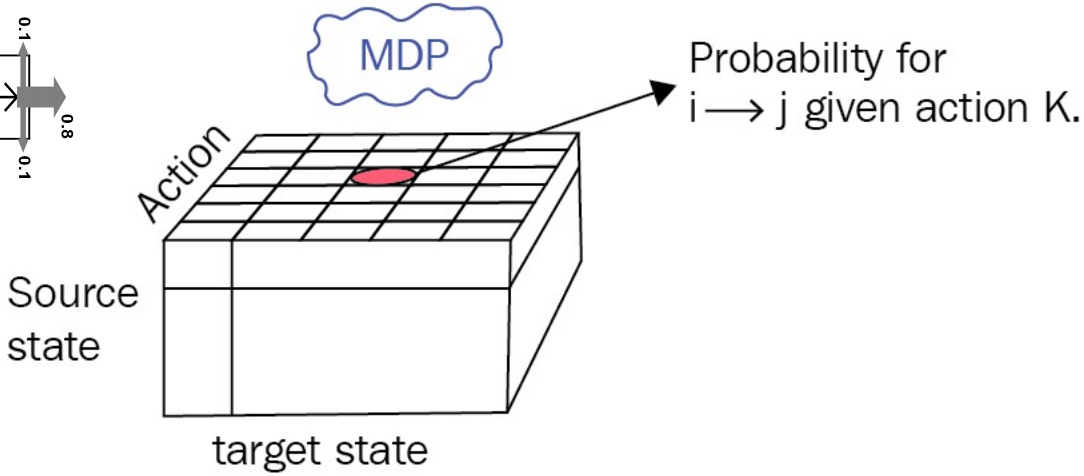
Markov Decision Processes

- Markov Decision Process (MDP)
  - Description of an MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:
  - State transition model:
    - A state transition probability matrix $\mathcal{P}$ helps to model the true state transition function $T(S_{t+1} | S_t, A_t)$ of a real-world environment.
    - For each action $A^i \in \mathcal{A}$, we have a state transition matrix $\mathcal{P}^{A^i}$ at any time-step t





http://ai.berkeley.edu/lecture_slides.html



Lapan, M. (2018). Deep Reinforcement Learning Hands-On. Packt Publishing Ltd.

Markov Decision Processes

- Markov Decision Process (MDP)
  - Description of an MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:
  - State transition model:
    - A state transition probability matrix $\mathcal{P}$ helps to model the true state transition function $T(S_{t+1} | S_t, A_t)$ of a real-world environment.
    - For each action $A^i \in \mathcal{A}$, we have a state transition matrix $\mathcal{P}^{A^i}$ at any time-step t as follows:

$$\begin{bmatrix} \mathcal{P}_{11} & \cdots & \mathcal{P}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{P}_{n1} & \cdots & \mathcal{P}_{nn} \end{bmatrix}$$

Notes:
- Rows sum up to 1.0.
- $\mathcal{P}$ could change over time.

Markov Decision Processes ($\mathcal{S}$)

- History is the sequence of observations, actions, rewards:

$$H_t = O_0, A_0, R_0, O_1, A_1, R_1, O_2, \dots, O_{t-1}, A_{t-1}, R_{t-1}, O_t$$

- 3 different definitions of s_t

  - (Full) Environmental state S_t^e
    - Private to the environment, not visible, maybe irrelevant information
    - Uses H_t to pick observation and reward

  - Agent state S_t^a (actually used)
    - Private to the agent, history of observations, rewards, and actions
    - Uses function of history $S_t^a = f(H_t)$ to select next action

  - Information state (we will define it soon)
    - Basically, S_t^a with special constraints in $f(H_t)$

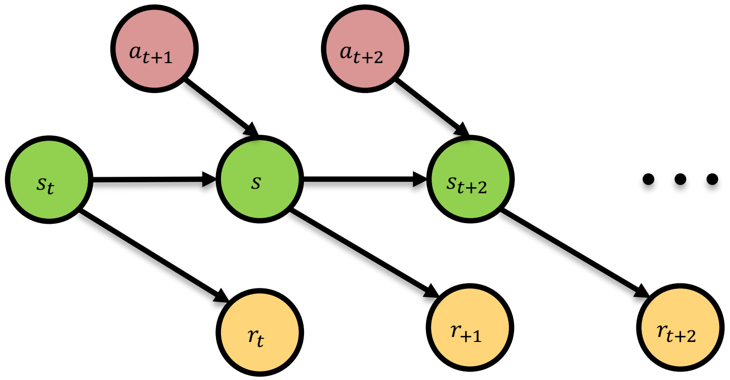
Markov Decision Processes ($\mathcal{S}$)

- Assumption of MDPs: Markov Property
  - A state S_t is Markov if and only if

$$\mathbb{P}[S_{t+1} \,|S_1, \dots, S_{t-1}, S_t] = \mathbb{P}[S_{t+1} \,|\, S_t]$$

  - Past states $S_1, \dots, S_{t-1}$ do not change the outcome for the next state S_{t+1}.
  - The current state S_t captures all relevant information from the history.
  - "The future is independent of the past given the present"

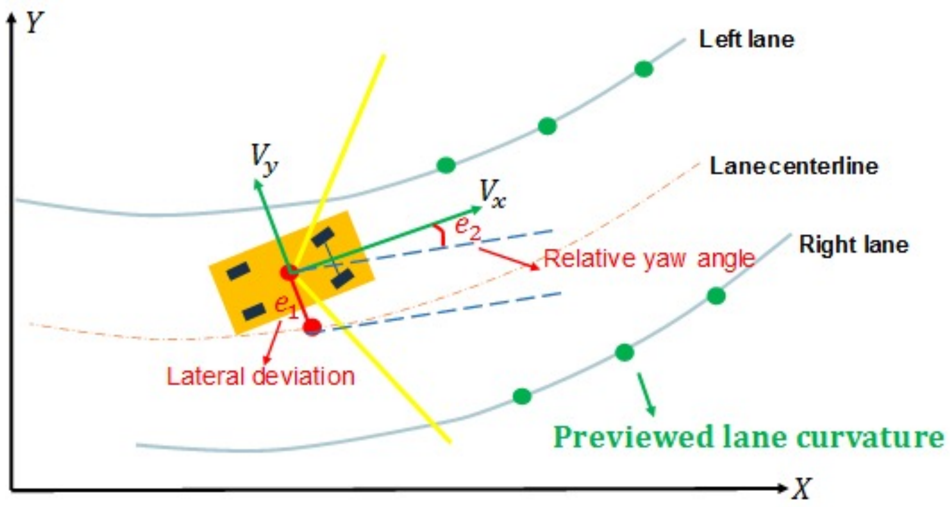
$$H_{1:t} \rightarrow S_t \rightarrow H_{t+1:\infty}$$



- State is the information used to determine what happens next
  - Direct (fully observable): $O_t = S_t^e$
  - Indirect (partially observable): $O_t = f(S_t^e)$

Markov Decision Processes ($\mathcal{S}$)

- Assumption of MDPs: Markov Property
  - How can we ensure/construct such a Markov state?

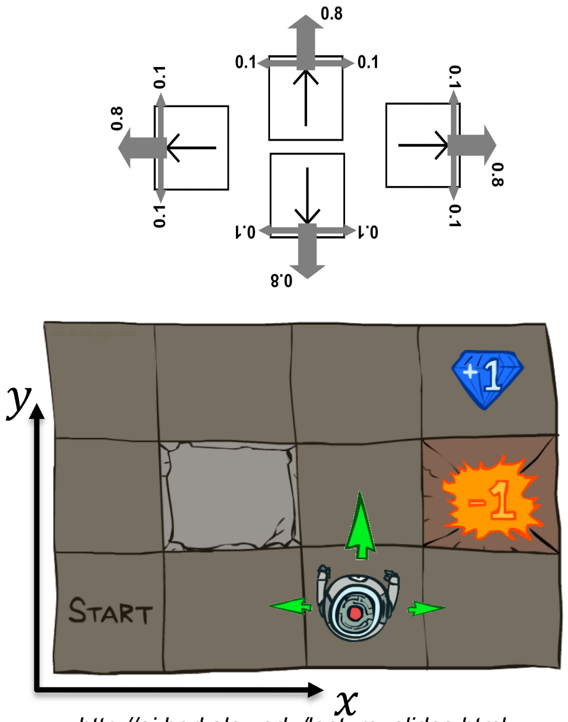


Sensor Measurements:
- Speed, Angle

Requirements:
- Lateral acceleration
- Angular velocity

Markov Decision Processes ($\mathcal{A}$)

- MDP example: Gridworld, episodic task

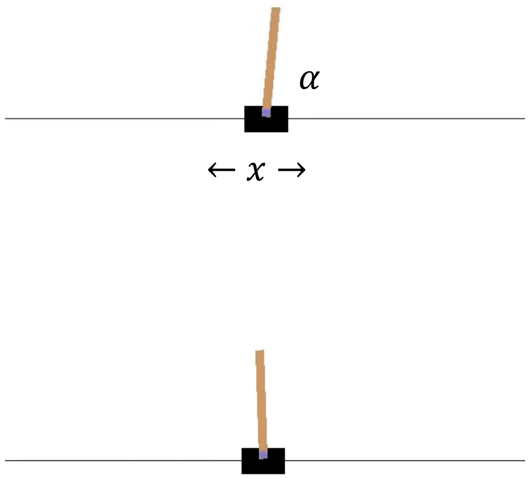


http://ai.berkeley.edu/lecture_slides.html

	Values
$\mathcal{S}$	(x, y) with $x \in \{0, 1, 2, 3\}$ and $y \in \{0, 1, 2\}$
$\mathcal{A}$	LEFT, RIGHT, UP, DOWN,

Markov Decision Processes ($\mathcal{A}$)

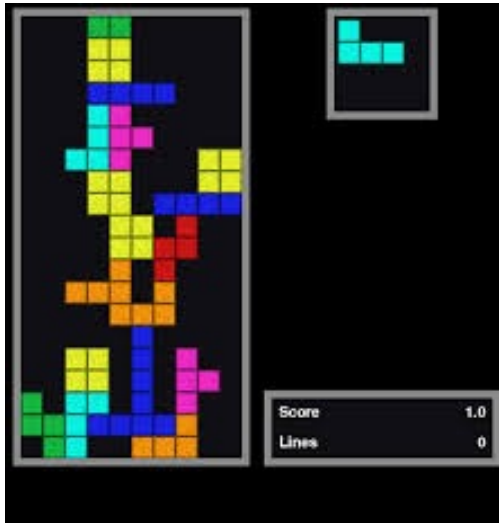
- MDP example: Cartpole, episodic or continuing task



	Values
$\mathcal{S}$	$(x, \theta, \dot{x}, \dot{\theta})$ with $x \in \mathbb{R}$ and $\alpha \in [0°, 360°]$
$\mathcal{A}$	LEFT, RIGHT

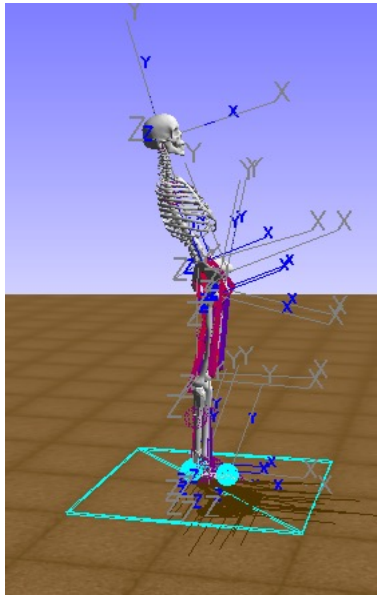
Markov Decision Processes

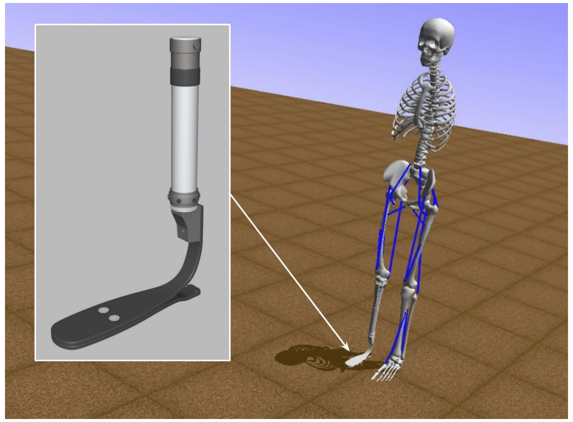
- MDP example: Tetris, episodic task



Markov Decision Processes

- MDP example: Running with a prosthetic leg, episodic task



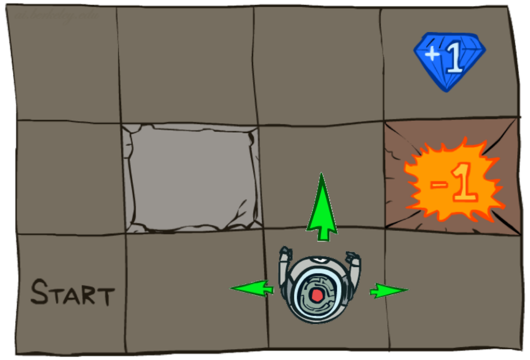


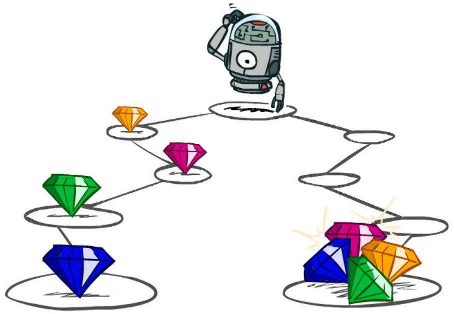
# of muscles	19
# degrees of freedom	14
reward	negative distance from requested velocity

Markov Decision Processes

- Markov Decision Process (MDP) is a tool to formulate RL problems
  - Description of MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$
  - Recall: Actions have consequences!
  - Choosing an action $A^i \in \mathcal{A}$ for A_t at timestep t yields different reward sequences
  - How do we know which sequence to prefer?

  - Idea: Decay value of rewards over time.
    - γ is a discount factor: $\gamma \in [0,1]$





http://ai.berkeley.edu/lecture_slides.html

Markov Decision Processes

- We want to “solve” the MDP, by maximizing future rewards.
  - We see the episodes in the form of

$$S_0 \xrightarrow{(A_0, R_0)} S_1 \xrightarrow{(A_1, R_1)} S_2 \xrightarrow{(A_2, R_2)} S_3 \dots \;\; S_{t-1} \xrightarrow{(A_{t-1}, R_{t-1})} S_t$$

  - **Question:** what happens if our problem never stops (i.e., $T = \infty$)?
    - Examples: data center cooling, recommender systems, etc.

  - Total discounted (γ) reward (**return**) (of one sample)

$$\mathrm{G} = R_0 + \gamma R_1 + \gamma^2 R_2 + \gamma^3 R_3 + \cdots = \sum_{t=0}^{\infty} \gamma^t R_t$$

Markov Decision Processes

- Markov Decision Process (MDP) is a tool to formulate RL problems
  - Description of MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$

- Why discount rewards with γ?
  - Mathematically convenient to discount rewards (true reason).
  - Avoids infinite returns in non-episodic problems
    - Datacenter cooling
    - Recommender system
  - Uncertainty about the future may not be fully represented (model uncertainty, our model is not perfect).

- Can I use $\gamma = 1$?
  - Yes, if you have an episodic setting or you definitely know that there is a terminal absorbing state.
- Should I use $\gamma = 1$?
  - **NO!**

Markov Decision Processes: Policy

- Expected long-term value of state s:

$$v(s) = \mathbb{E}(\mathrm{G}) = \mathbb{E}(R_0 + \gamma R_1 + \gamma^2 R_2 + \gamma^3 R_3 + \ldots + \gamma^t R_t)$$

- **Goal: maximize the expected return $\mathbb{E}(\mathrm{G})$.**

- We need a controller that helps us select the actions to maximize $\mathbb{E}(G)$.
- A policy π represents this controller:
  - π determines the agent's behavior, i.e., its way of acting
  - π is a mapping from state space $\mathcal{S}$ to action space $\mathcal{A}$

$$\pi : \mathcal{S} \mapsto \mathcal{A}$$

  - Two types of policies:
    - Deterministic policy: $a = \pi(s)$.
    - Stochastic policy: $\pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s]$.

- **New goal: find a policy that maximizes the expected return!**

Markov Decision Processes: Policy

- Some remarks on terminology

$\mathbf{s}_t$ – state
$\mathbf{a}_t$ – action
$r(\mathbf{s}, \mathbf{a})$ – reward function

$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action
$c(\mathbf{x}, \mathbf{u})$ – cost function

$$r(\mathbf{s}, \mathbf{a}) = -c(\mathbf{x}, \mathbf{u})$$

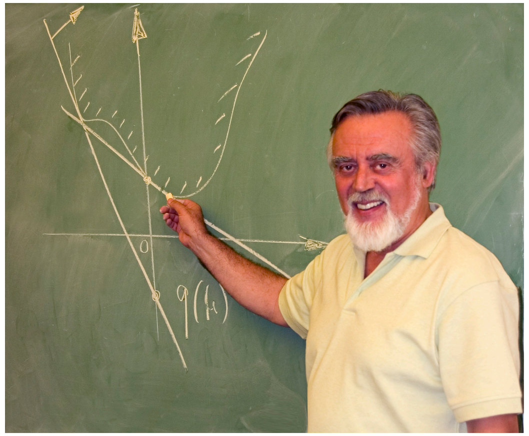
Richard Bellman

Lev Pontryagin

http://rail.eecs.berkeley.edu/deeprlcourse/static/slides/lec-2.pdf

Markov Decision Processes: Policy

- Some remarks on terminology



Bertsekas, 2019:

RL uses Max/Value, DP uses Min/Cost

- Reward of a stage = (Opposite of) Cost of a stage.
- State value = (Opposite of) State cost.
- Value (or state-value) function = (Opposite of) Cost function.

Controlled system terminology

- Agent = Decision maker or controller.
- Action = Decision or control.
- Environment = Dynamic system.

Methods terminology

- Learning = Solving a DP-related problem using simulation.
- Self-learning (or self-play in the context of games) = Solving a DP problem using simulation-based policy iteration.
- Planning vs Learning distinction = Solving a DP problem with model-based vs model-free simulation.