Reinforcement Learning

Dipendra Misra
Cornell University
dkm@cs.cornell.edu

https://dipendramisra.wordpress.com/

Task

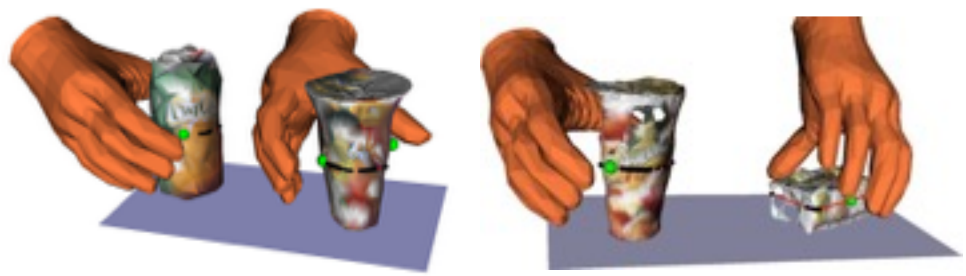
Grasp the green cup.

Output: Sequence of controller actions

Setup from Lenz et. al. 2014

Supervised Learning

Grasp the green cup.



Expert Demonstrations

Setup from Lenz et. al. 2014

Supervised Learning

Grasp the green cup. Problem?

Expert Demonstrations

Setup from Lenz et. al. 2014

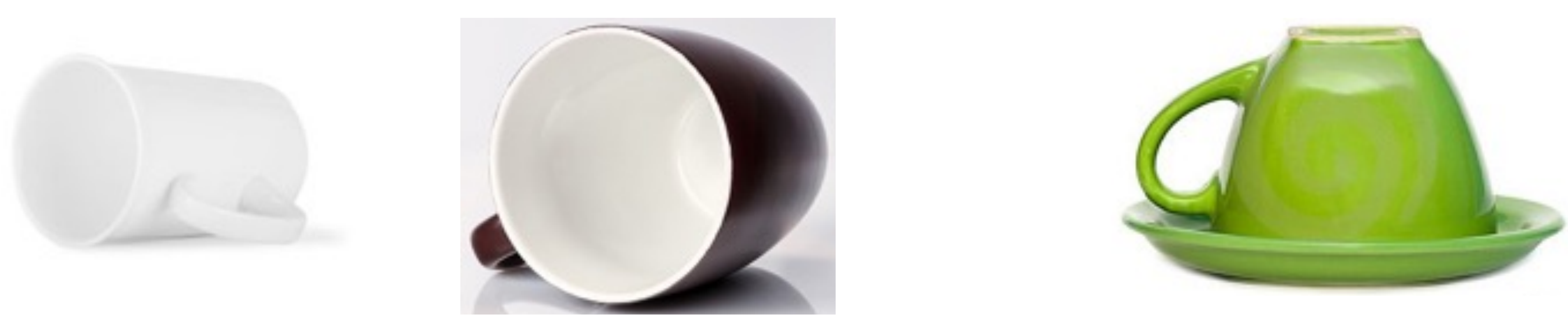
Supervised Learning

Grasp the cup. Problem?

Training data

Test data

No exploration



Exploring the environment

What is reinforcement learning?

"Reinforcement learning is a computation approach that emphasizes on learning by the individual from direct interaction with its environment, without relying on exemplary supervision or complete models of the environment"

- R. Sutton and A. Barto

Interaction with the environment

action

reward

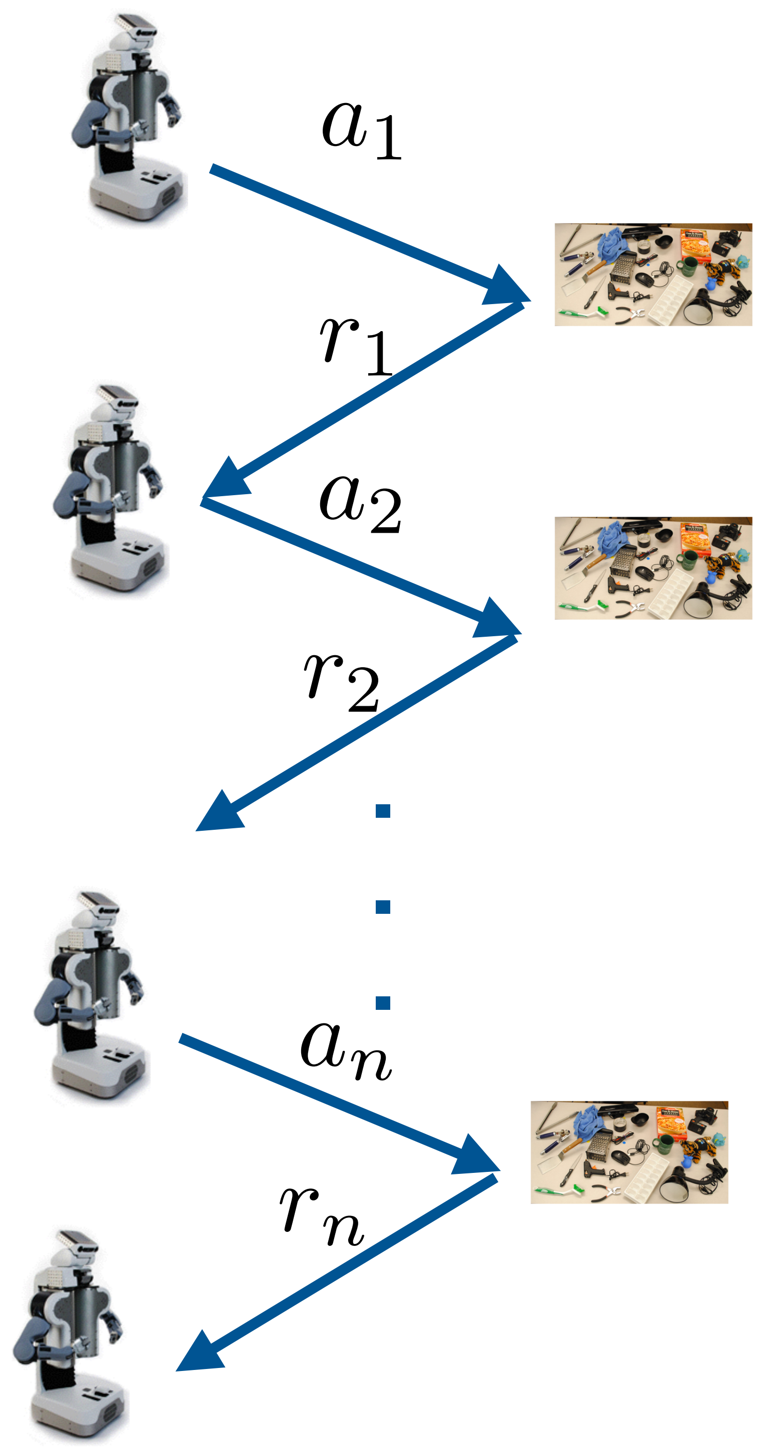
+

new environment

Scalar reward

Setup from Lenz et. al. 2014

Interaction with the environment

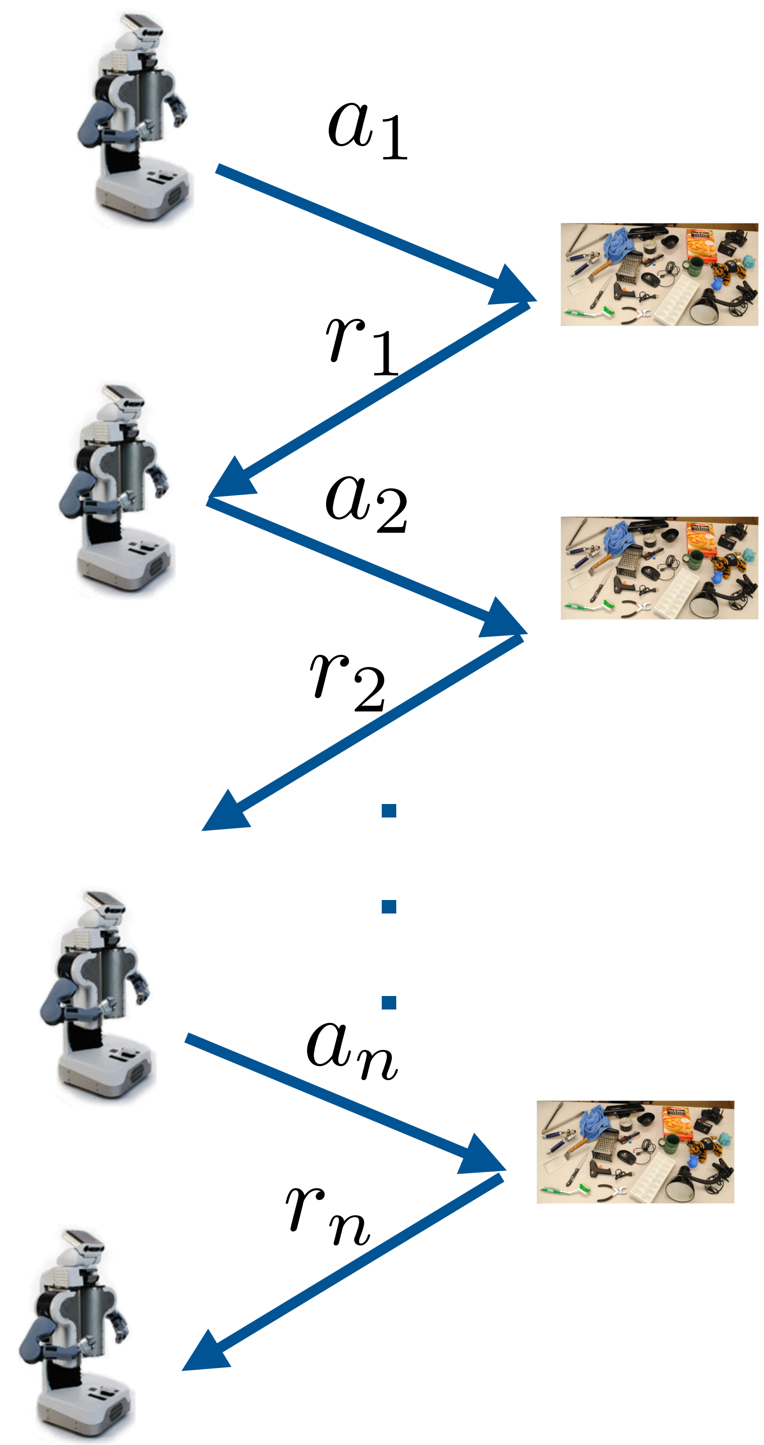


Episodic
vs
Non-Episodic

Setup from Lenz et. al. 2014

Rollout

$$\langle s_1, a_1, r_1, s_2, a_2, r_2, s_3, \cdots a_n, r_n, s_n \rangle$$



Setup from Lenz et. al. 2014

Setup

$$s_t \xrightarrow[r_t]{a_t} s_{t+1}$$

e.g.,1$

Policy

$$\pi(s, a) = 0.9$$

Interaction with the environment

action

reward
+
new environment

Objective?

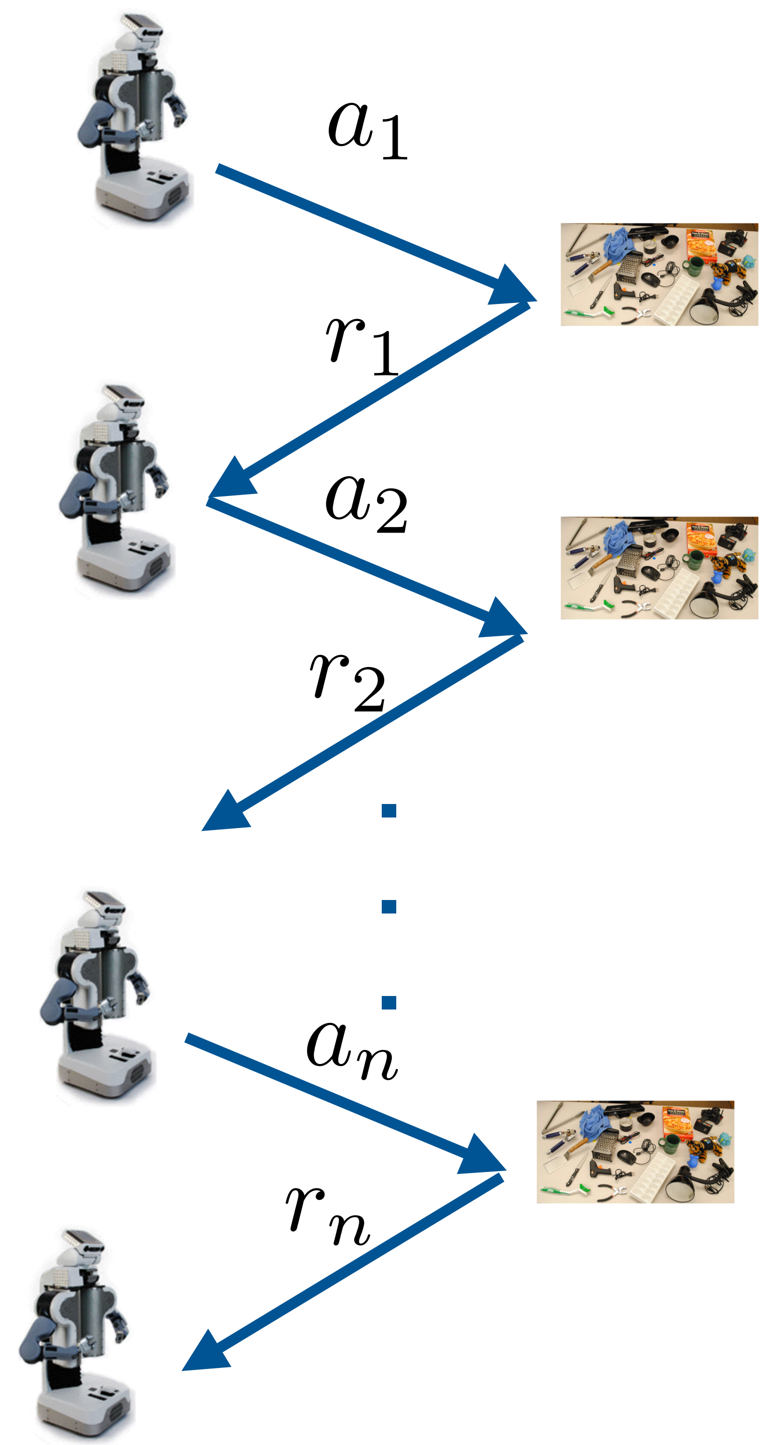
Setup from Lenz et. al. 2014

Objective

$$\langle s_1, a_1, r_1, s_2, a_2, r_2, s_3, \cdots a_n, r_n, s_n \rangle$$

maximize
expected reward

$$E\left[\sum_{t=1}^{n} r_t\right]$$



Problem?

Setup from Lenz et. al. 2014

Discounted Reward

maximize
expected reward

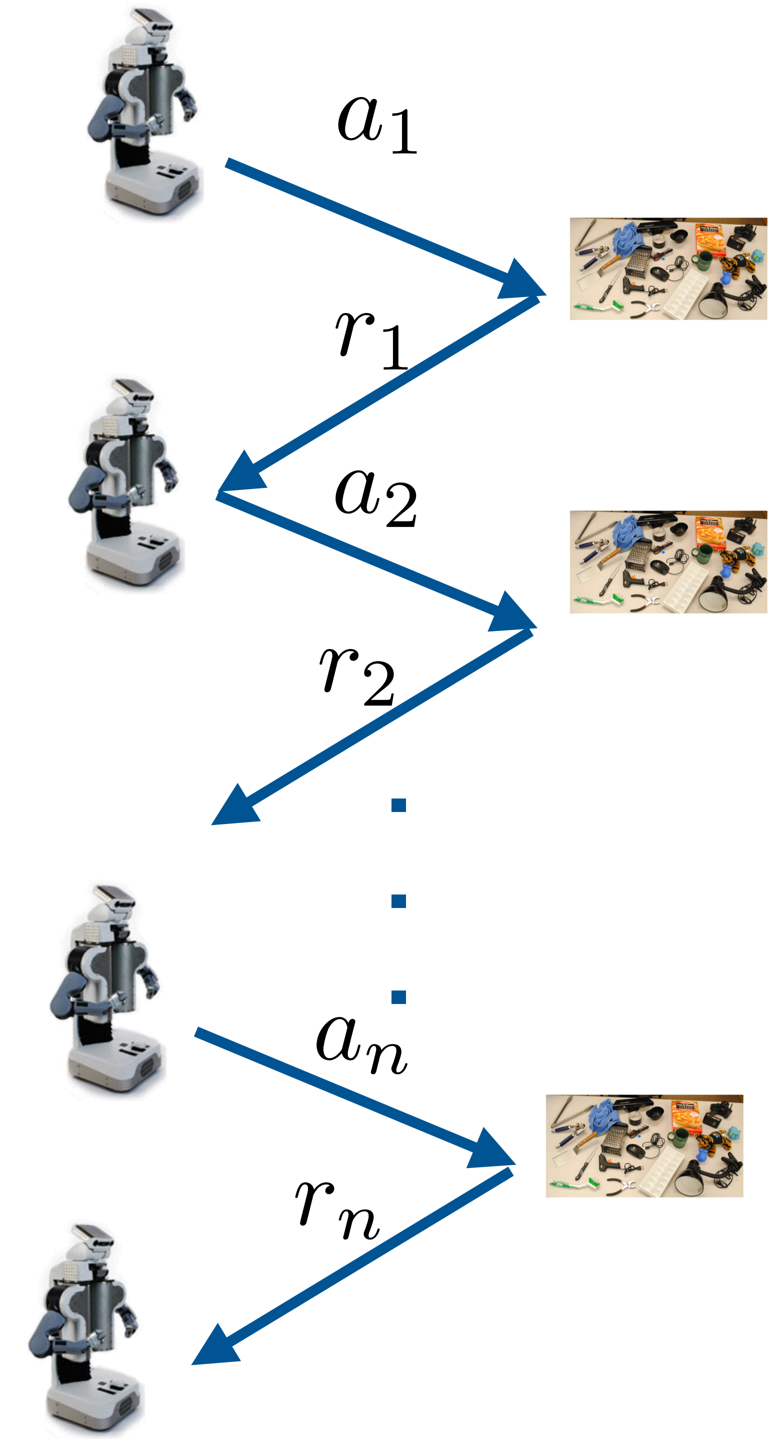
$$E\left[\sum_{t=0}^{\infty} r_{t+1}\right]$$

Problem?

unbounded

discount future reward

$$E\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1}\right] \qquad \gamma \in [0, 1)$$



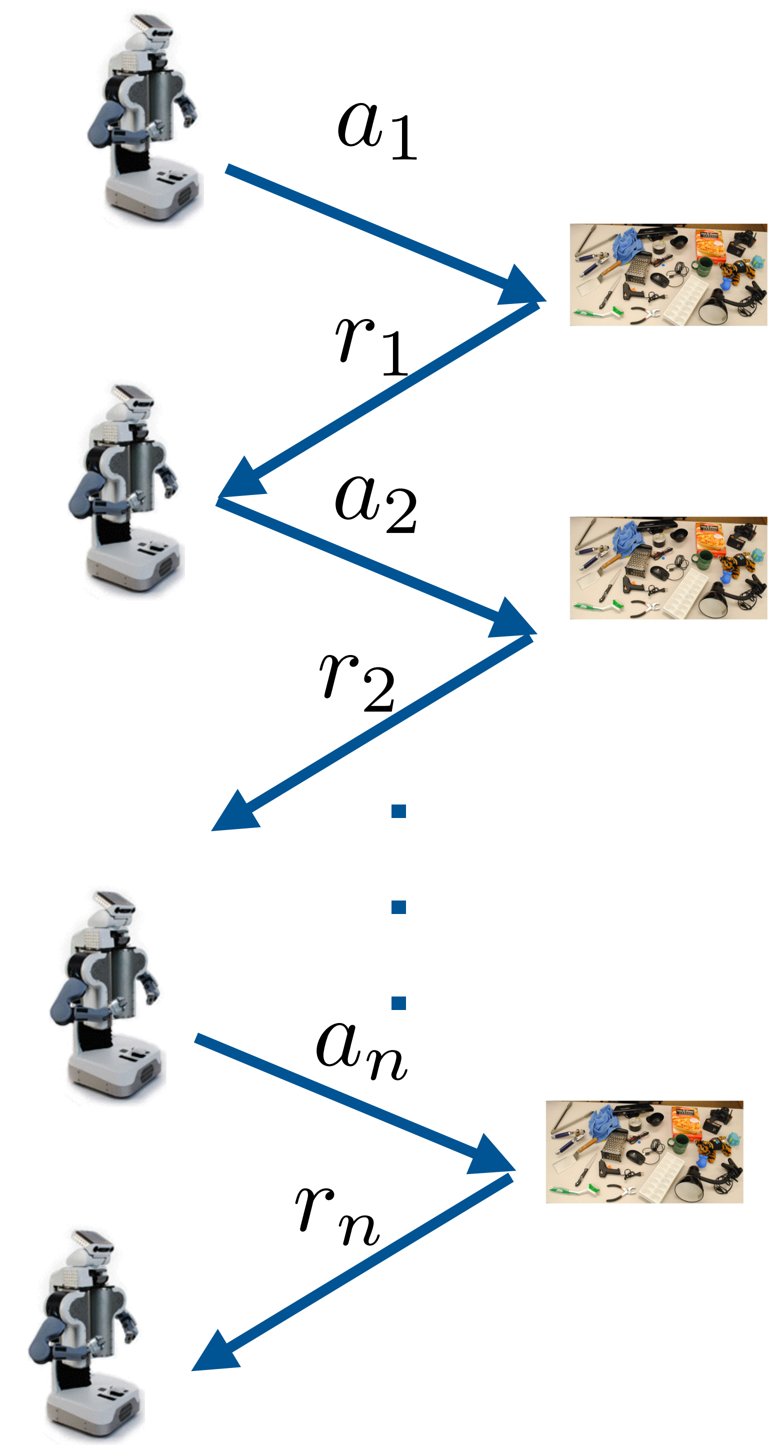
Discounted Reward

maximize discounted
expected reward

$$E\left[\sum_{t=0}^{n-1} \gamma^t r_{t+1}\right]$$

if $r \leq M$ and $\gamma \in [0, 1)$

$$E\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1}\right] \leq \sum_{t=0}^{\infty} \gamma^t M = \frac{M}{1-\gamma}$$



Need for discounting

- To keep the problem well formed
- Evidence that humans discount future reward

Markov Decision Process

MDP is a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ where

- $\mathcal{S}$ is a set of finite or infinite states
- $\mathcal{A}$ is a set of finite or infinite actions
- For the transition $s \xrightarrow{a} s'$
- $P_{s,s'}^{a} \in P$ is the transition probability
- $R_{s,s'}^{a} \in R$ is the reward for the transition
- $\gamma \in [0, 1]$ is the discounted factor

} Markov Asmp. (brace grouping the transition probability and reward items)

MDP Example

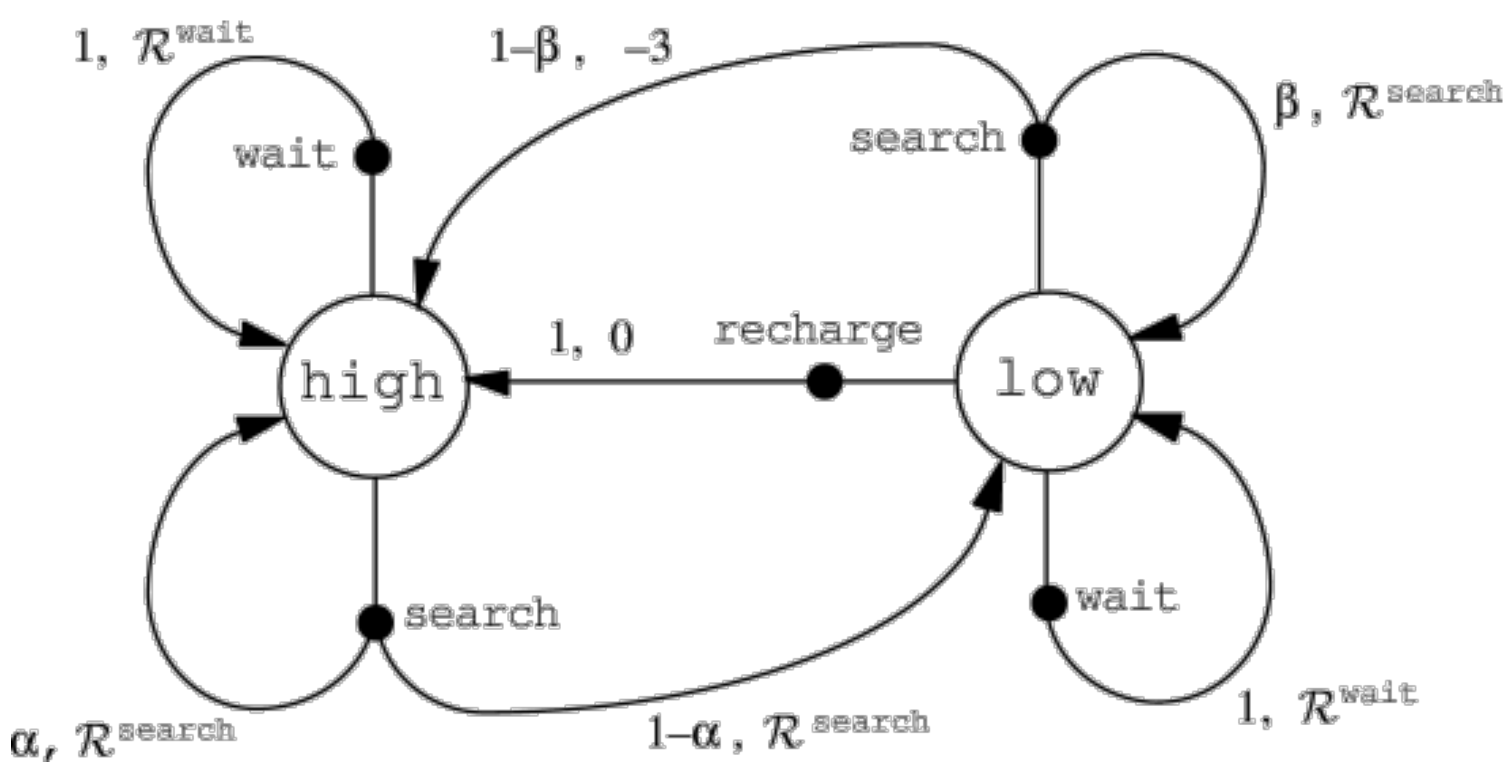


Table 3.1: Transition probabilities and expected rewards for the finite MDP of the recycling robot example. There is a row for each possible combination of current state, s, next state, s', and action possible in the current state, $a \in \mathcal{A}(s)$.

$s = s_t$	$s' = s_{t+1}$	$a = a_t$	$\mathcal{P}^a_{ss'}$	$\mathcal{R}^a_{ss'}$
high	high	search	α	$\mathcal{R}^{\text{search}}$
high	low	search	$1 - \alpha$	$\mathcal{R}^{\text{search}}$
low	high	search	$1 - \beta$	-3
low	low	search	β	$\mathcal{R}^{\text{search}}$
high	high	wait	1	$\mathcal{R}^{\text{wait}}$
high	low	wait	0	$\mathcal{R}^{\text{wait}}$
low	high	wait	0	$\mathcal{R}^{\text{wait}}$
low	low	wait	1	$\mathcal{R}^{\text{wait}}$
low	high	recharge	1	0
low	low	recharge	0	0.

Example from Sutton and Barto 1998

Summary

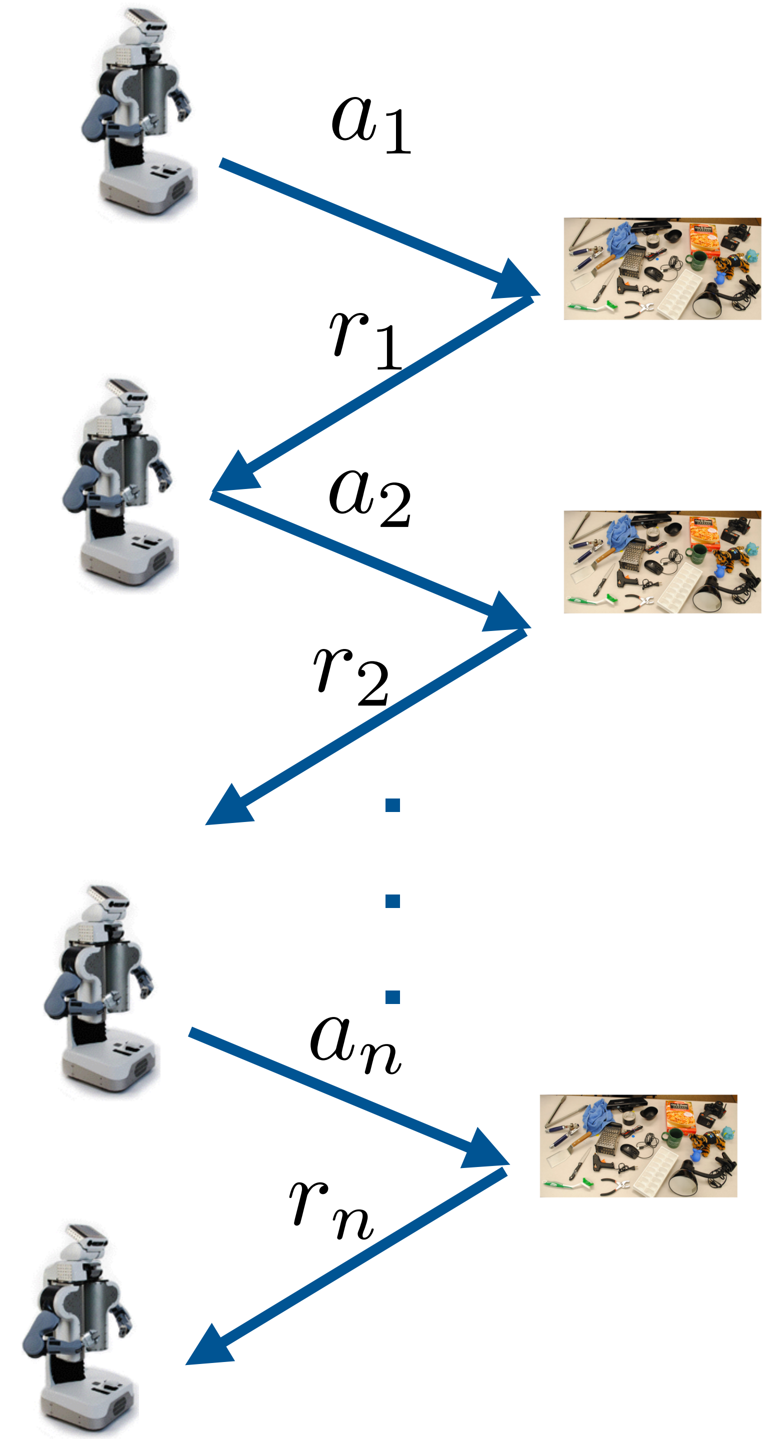
MDP is a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$

Maximize discounted expected reward

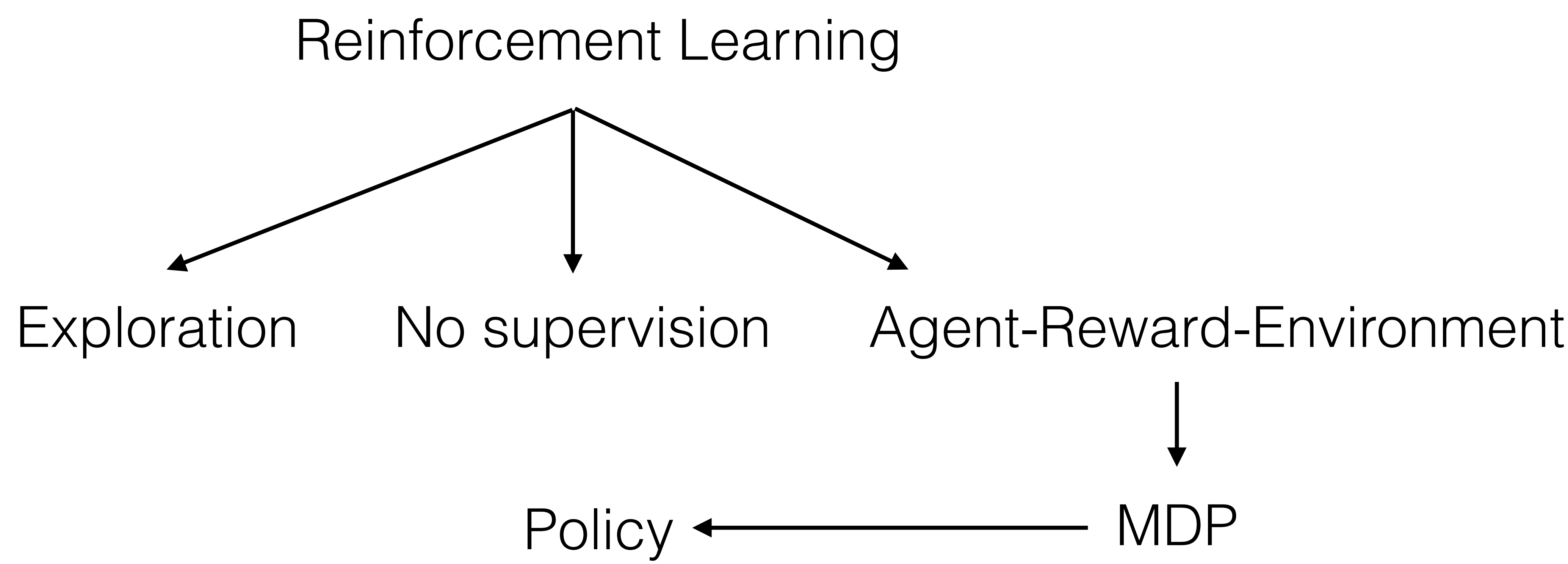
$$E\left[\sum_{t=0}^{n-1} \gamma^t r_{t+1}\right]$$

Agent controls the policy

$$\pi(s, a)$$



What we learned



Value functions

- Expected reward from following a policy

State value function

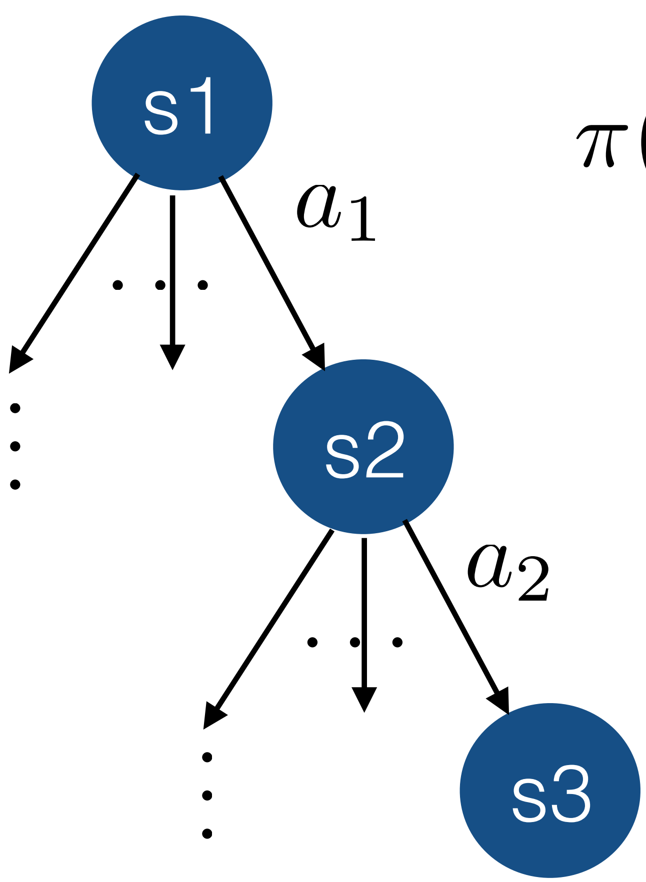
$$V^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_1 = s, \pi\right]$$

State action value function

$$Q^\pi(s,a) = E\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_1 = s, a_1 = a, \pi\right]$$

State Value function

$$V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_1 = s, \pi\right]$$



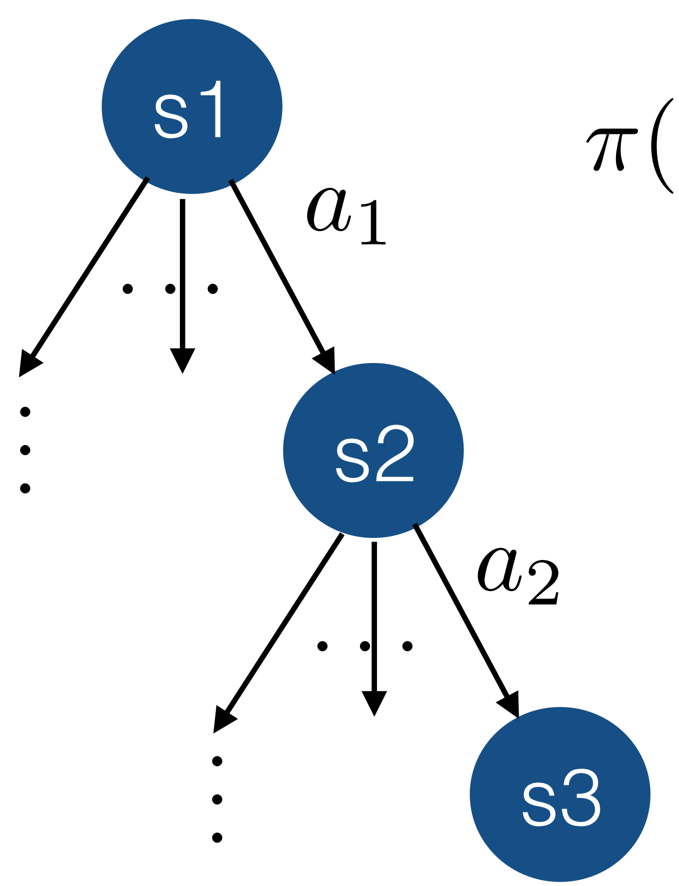
$$\pi(s_1, a_1) P^{a_1}_{s_1, s_2} \qquad R^{a_1}_{s_1, s_2}$$

$$\pi(s_2, a_2) P^{a_2}_{s_2, s_3} \qquad R^{a_2}_{s_2, s_3}$$

State Value function

$$V^{\pi}(s_1) = E\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1}\right]$$

$$= \sum_t (r_1 + \gamma r_2 \cdots) p(t) \quad \text{where} \quad t = \langle s_1, a_1, s_2, a_2 \cdots \rangle = \langle s_1, a_1, s_2 \rangle : t'$$



$$\pi(s_1, a_1) P^{a_1}_{s_1, s_2} \qquad R^{a_1}_{s_1, s_2}$$

$$\pi(s_2, a_2) P^{a_2}_{s_2, s_3} \qquad R^{a_2}_{s_2, s_3}$$

State Value function

$$V^{\pi}(s_1) = E\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1}\right]$$

$$= \sum_t (r_1 + \gamma r_2 \cdots) p(t) \quad \text{where} \quad t = \langle s_1, a_1, s_2, a_2 \cdots \rangle = \langle s_1, a_1, s_2 \rangle : t'$$

$$= \sum_{a_1, s_2} \sum_{t'} P(s_1, a_1, s_2) P(t' \mid s_1, a_1, s_2) \left\{ R^{a_1}_{s_1, s_2} + \gamma (r_2 \cdots) \right\}$$

$$= \sum_{a_1, s_2} P(s_1, a_1, s_2) \{ R^{a_1}_{s_1, s_2} + \gamma \underbrace{\sum_{t'} P(t' \mid s_1, a_1, s_2)(r_2 \cdots)}_{V^{\pi}(s_2)} \}$$

$$= \sum_{a_1} \pi(s_1, a_2) \sum_{s_2} P^{a_1}_{s_1, s_2} \left\{ R^{a_1}_{s_1, s_2} + \gamma V^{\pi}(s_2) \right\}$$

Bellman Self-Consistency Eqn

$$V^\pi(s) = \sum_a \pi(s,a) \sum_{s'} P^a_{s,s'} \left\{ R^a_{s,s'} + \gamma V^\pi(s') \right\}$$

similarly

$$Q^\pi(s,a) = \sum_{s'} P^a_{s,s'} \left\{ R^a_{s,s'} + \gamma V^\pi(s') \right\}$$

$$Q^\pi(s,a) = \sum_{s'} P^a_{s,s'} \left\{ R^a_{s,s'} + \gamma \sum_{a'} \pi(s',a') Q^\pi(s',a') \right\}$$

Bellman Self-Consistency Eqn

$$V^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} P^{a}_{s,s'} \left\{ R^{a}_{s,s'} + \gamma V^{\pi}(s') \right\}$$

Given N states, we have N equations in N variables

Solve the above equation

Does it have a unique solution?

Yes, it does. Exercise: Prove it.

Optimal Policy

$$V^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} P^{a}_{s,s'} \left\{ R^{a}_{s,s'} + \gamma V^{\pi}(s') \right\}$$

Given a state s

policy π_1 is as good as π_2 (den. $\pi_1 \geq \pi_2$) if:

$$V^{\pi_1}(s) \geq V^{\pi_2}(s)$$

How to define a globally optimal policy?

Optimal Policy

policy π_1 is as good as π_2 (den. $\pi_1 \geq \pi_2$) if:

$$V^{\pi_1}(s) \geq V^{\pi_2}(s)$$

How to define a globally optimal policy?

π^* is an optimal policy if:

$$V^{\pi^*}(s) \geq V^{\pi}(s) \quad \forall s \in \mathcal{S}, \pi$$

Does it always exists?

Yes it always does.

Existence of Optimal Policy

Leader policy for every state s is: $\pi_s = arg\max_\pi V^\pi(s)$

Define: $\pi^*(s,a) = \pi_s(s,a) \quad \forall s, a$

To show π^* is optimal or equivalently:

$$\delta(s) = V^{\pi^*}(s) - V^{\pi_s}(s) \geq 0$$

$$V^{\pi^*}(s) = \sum_a \pi_s(s,a) \sum_{s'} P^a_{s,s'}\{R^a_{s,s'} + \gamma V^{\pi^*}(s')\}$$

$$V^{\pi^s}(s) = \sum_a \pi_s(s,a) \sum_{s'} P^a_{s,s'}\{R^a_{s,s'} + \gamma V^{\pi_s}(s')\}$$

Existence of Optimal Policy

Leader policy for every state $\boldsymbol{S}$ is: $\pi_s = arg\max_{\pi} V^{\pi}(s)$

Define: $\pi^*(s,a) = \pi_s(s,a) \quad \forall s, a$

$$V^{\pi^*}(s) = \sum_a \pi_s(s,a) \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V^{\pi^*}(s')\}$$

$$V^{\pi^s}(s) = \sum_a \pi_s(s,a) \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V^{\pi_s}(s')\}$$

$$\delta(s) = V^{\pi^*}(s) - V^{\pi_s}(s) = \gamma \sum_a \pi_s(s,a) \sum_{s'} P^a_{s,s'} \underbrace{\{V^{\pi^*}(s') - V^{\pi_s}(s')\}}_{\geq \delta(s')}$$

$$\geq \gamma \sum_a \pi_s(s,a) \sum_{s'} P^a_{s,s'} \{\delta(s')\} = \gamma \texttt{conv}(\delta(s'))$$

Existence of Optimal Policy

Leader policy for every state $\boldsymbol{S}$ is: $\pi_s = arg\max_{\pi} V^{\pi}(s)$

Define: $\pi^*(s,a) = \pi_s(s,a) \;\; \forall s, a$

$$\delta(s) = V^{\pi^*}(s) - V^{\pi_s}(s) = \gamma \sum_a \pi_s(s,a) \sum_{s'} P^a_{s,s'} \{V^{\pi^*}(s') - V^{\pi_s}(s')\}$$

$$\geq \gamma \sum_a \pi_s(s,a) \sum_{s'} P^a_{s,s'} \{\delta(s')\} = \gamma \texttt{conv}(\delta(s'))$$

$$\delta(s) \geq \gamma \min \delta(s')$$

$$\min \delta(s) \geq \gamma \min \delta(s')$$

$$\gamma \in [0,1) \Rightarrow \min \delta(s) \geq 0$$

Hence proved

Bellman's Optimality Condition

Define $V^*(s) = V^{\pi^*}(s)$ and $Q^*(s,a) = Q^{\pi^*}(s,a)$

$$V^{\pi^*}(s) = \sum_a \pi^*(s,a) Q^{\pi^*}(s,a) \leq \max_a Q^{\pi^*}(s,a)$$

Claim: $V^{\pi^*}(s) = \max_a Q^{\pi^*}(s,a)$

Let $V^{\pi^*}(s) < \max_a Q^{\pi^*}(s,a)$

Define $\pi'(s) = \arg\max_a Q^{\pi^*}(s,a)$

Bellman's Optimality Condition

$$\pi'(s) = \arg\max_a Q^{\pi^*}(s,a)$$

$$V^{\pi^*}(s) = \sum_a \pi^*(s,a) Q^{\pi^*}(s,a)$$

$$V^{\pi'}(s) = Q^{\pi'}(s, \pi'(s))$$

$$\delta(s) = V^{\pi'}(s) - V^{\pi^*}(s) = Q^{\pi'}(s,\pi'(s)) - \sum_a \pi^*(s,a) Q^{\pi^*}(s,a)$$
$$\geq Q^{\pi'}(s,\pi'(s)) - Q^{\pi^*}(s,\pi'(s)) = \gamma \sum_{s'} P_{s,s'}^{\pi'(s)} \delta(s')$$

$$\delta(s) \geq 0$$

π^* is not optimal

$\exists s'$ such that $\delta(s') > 0$

Bellman's Optimality Condition

$$V^*(s) = \max_a Q^*(s,a)$$

$$V^*(s) = \max_a \sum_{s'} P^a_{s,s'}\{R^a_{s,s'} + \gamma V^*(s')\}$$

similarly

$$Q^*(s,a) = \sum_{s'} P^a_{s,s'}\{R^a_{s,s'} + \gamma \max_{a'} Q^*(s',a')\}$$

Optimal policy from Q value

Given $Q^*(s,a)$ an optimal policy is given by:

$$\pi^*(s) = \arg\max_a Q^*(s,a)$$

Corollary: Every MDP has a deterministic optimal policy

Summary

An optimal policy π^* exists such that:

$$V^{\pi^*}(s) \geq V^{\pi}(s) \quad \forall s \in \mathcal{S}, \pi$$

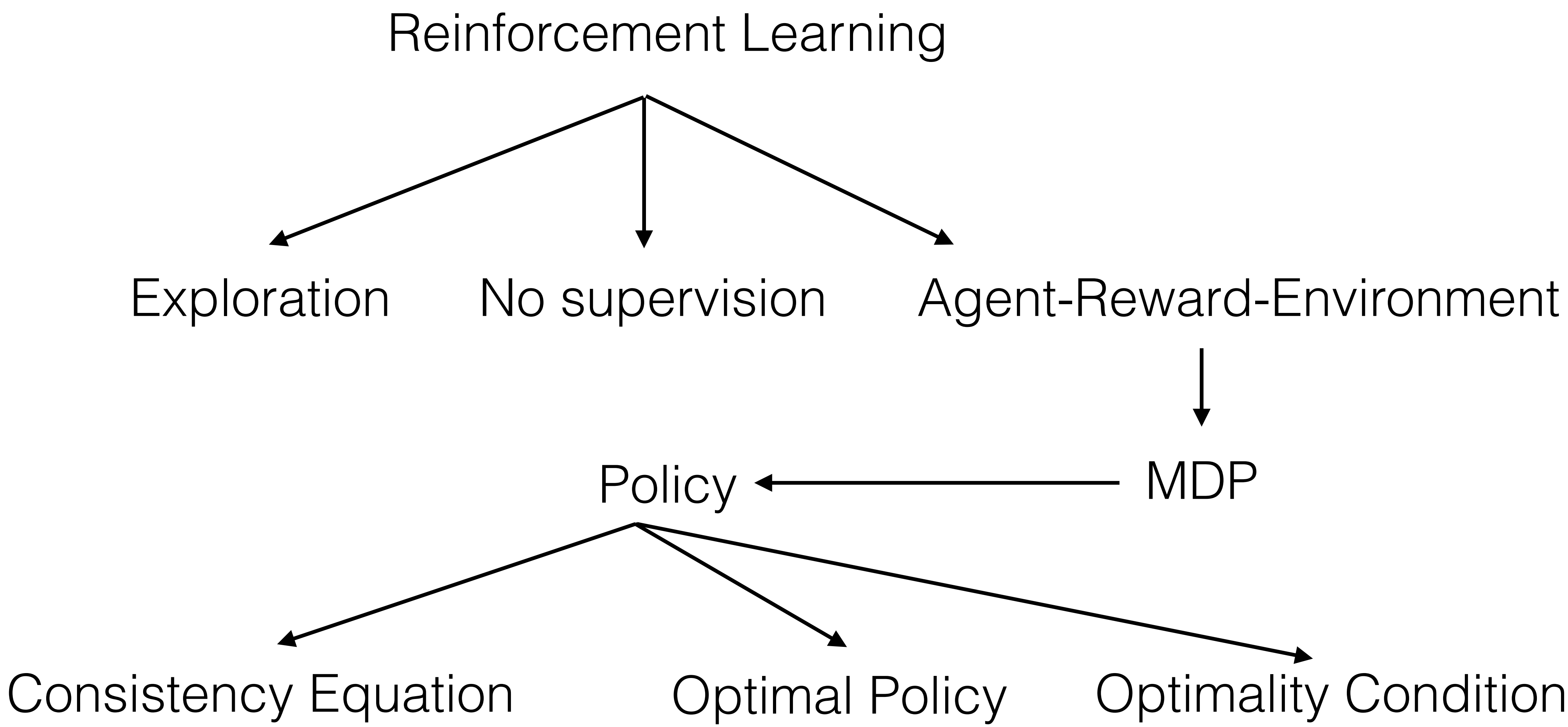
Bellman's self-consistency equation

$$V^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} P^{a}_{s,s'} \left\{ R^{a}_{s,s'} + \gamma V^{\pi}(s') \right\}$$

Bellman's optimality condition

$$V^{*}(s) = \max_{a} \sum_{s'} P^{a}_{s,s'} \{ R^{a}_{s,s'} + \gamma V^{*}(s') \}$$

What we learned



Solving MDP

To solve an MDP is to find an optimal policy

Bellman's Optimality Condition

$$V^*(s) = \max_a \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V^*(s')\}$$

Iteratively solve the above equation

Bellman Backup Operator

$$V^*(s) = \max_a \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V^*(s')\}$$

$$T : V \to V$$

$$(TV)(s) = \max_a \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V(s')\}$$

Dynamic Programming Solution

Initialize V^0 randomly

do

$$V^{t+1} = TV^t$$

until $\|V^{t+1} - V^t\|_\infty > \epsilon$

return V^{t+1}

$$V^{t+1}(s) = \max_a \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V^t(s')\}$$

Convergence

$$(TV)(s) = \max_a \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V(s')\}$$

Theorem: $\|TV_1 - TV_2\|_\infty \leq \gamma \|V_1 - V_2\|_\infty$

where $\|x\|_\infty = \max\{|x_1|, |x_2| \cdots |x_k|\};\ x \in R^k$

Proof:

$$|(TV_1)(s) - (TV_2)(s)| = |\max_a \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V_1(s')\} -$$

$$- \max_a \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V_2(s')\}|$$

using $|\max_x f(x) - \max_x g(x)| \leq \max_x |f(x) - g(x)|$

Convergence

Theorem: $$\|TV_1 - TV_2\|_\infty = \gamma\|V_1 - V_2\|_\infty$$

where $\|x\|_\infty = \max\{|x_1|, |x_2| \cdots |x_k|\};\ x \in R^k$

Proof:

$$\begin{aligned}
|(TV_1)(s) - (TV_2)(s)| &\le \max_a \gamma |\sum_{s'} P^a_{s,s'}(V_1(s') - V_2(s'))| \\
&\le \max_a \gamma \sum_{s'} P^a_{s,s'} |(V_1(s') - V_2(s'))| \\
&\le \max_a \max_{s'} |V_1(s') - V_2(s')| \\
&\le \gamma \|V_1 - V_2\|_\infty
\end{aligned}$$

$$\Rightarrow \|TV_1 - TV_2\|_\infty \le \gamma\|V_1 - V_2\|_\infty$$

Optimal is a fixed point

$$V^* = \max_a \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V^*(s')\} = TV^*$$

V^* is a fixed point of T

Optimal is the fixed point

$$V^* = \max_a \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V^*(s')\} = TV^*$$

V^* is a fixed point of T

Theorem: V^* is the only fixed point of T

Proof: $TV_1 = V_1$ $\quad$ $TV_2 = V_2$

$$\|V_1 - V_2\|_\infty = \|TV_1 - TV_2\|_\infty \leq \gamma \|V_1 - V_2\|_\infty$$

As $\gamma \in [0, 1)$ therefore $\|V_1 - V_2\|_\infty = 0 \Rightarrow V_1 = V_2$

Dynamic Programming Solution

Initialize V^0 randomly

do

$$V^{t+1} = TV^t$$

until $\|V^{t+1} - V^t\|_\infty > \epsilon$

Problem?

return V^{t+1}

Theorem: algorithm converges for all V^0

Proof: $\|V^{t+1} - V^*\|_\infty = \|TV^t - TV^*\|_\infty \leq \gamma \|V^t - V^*\|_\infty$

$$\|V^t - V^*\|_\infty \leq \gamma^t \|V^0 - V^*\|_\infty$$

$$\lim_{t\to\infty} \|V^t - V^*\|_\infty \leq \lim_{t\to\infty} \gamma^t \|V^0 - V^*\|_\infty = 0$$

Summary

Iteratively solving optimality condition

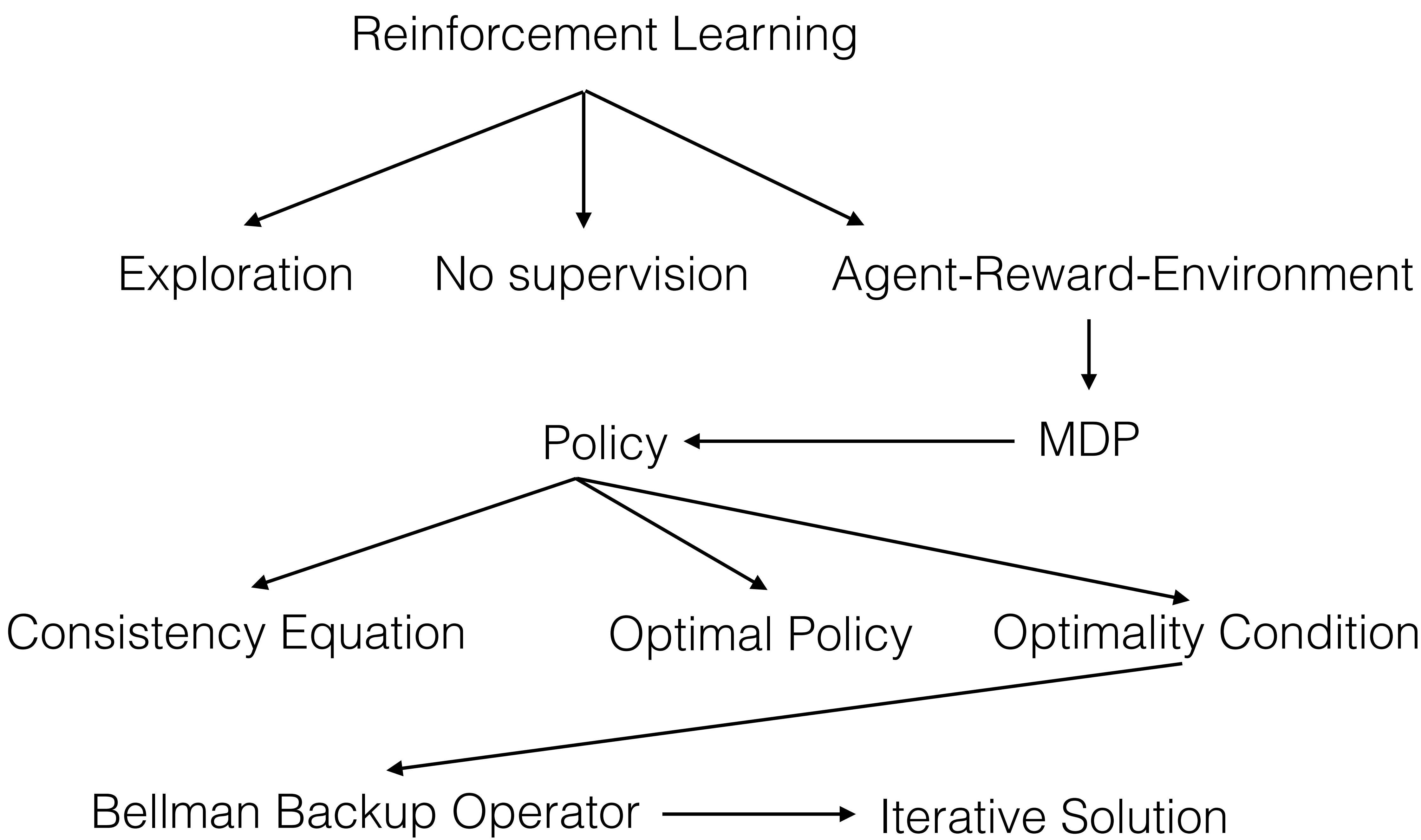
$$V^{t+1}(s) = \max_a \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V^t(s')\}$$

Bellman Backup Operator

$$(TV)(s) = \max_a \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V(s')\}$$

Convergence of the iterative solution

What we learned



In next tutorial

- Value and Policy Iteration
- Monte Carlo Solution
- SARSA and Q-Learning
- Policy Gradient Methods
- Learning to search OR Atari game paper?