TÉCNICO LISBOA

Planning, Learning and Intelligent Decision Making

Lecture 6

PADInt 2024

Markov decision process

- **Model** for sequential decision processes
- Described by:
  - State space, $\mathcal{X}$
  - Action space, $\mathcal{A}$
  - Transition probabilities, $\{\mathbf{P}_a, a \in \mathcal{A}\}$
  - Immediate cost function, $\mathbf{c}$

Policy iteration

Value iteration

- Value iteration methods are guaranteed to converge asymptotically

Can we do better?

Our goal

- Our goal is to compute the optimal policy
- How many policies are there?

$$|\mathcal{A}|^{|\mathcal{X}|}$$

Finite number!

- If we search all, we'll finish in a finite number of iterations!

Policy iteration

- Build a sequence of policies,

$$\pi^{(0)}, \pi^{(1)}, \ldots, \pi^{(k)}, \pi^{(k+1)}, \ldots$$

  where each one is better than the previous

Greedy policy

- Given a cost-to-go function J, the greedy policy w.r.t. J is

$$\pi_g(x) = \arg\min_a \left[c(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathbf{P}_a(y \mid x) J(y) \right]$$

- ... for example, the optimal policy is greedy w.r.t. J^*

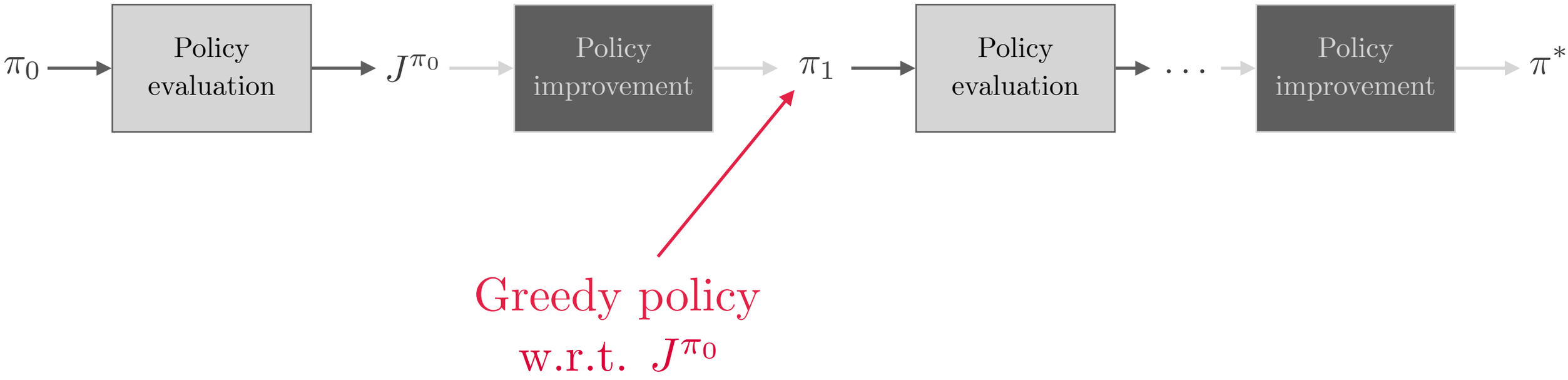
Improvement result

- Given a policy π...
- ... given the cost-to-go function J^{π},
- ... then

$$J^{\pi_g}(x) \leq J^{\pi}(x)$$

Policy π_g is better than π

Policy iteration



Policy iteration

- We get an iterative approach:

Input: MDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \{\mathbf{P}_a\}, c, \gamma)$

1: Initialize $k = 0$

2: Initialize π_0 randomly

3: **repeat**

4: $\quad \mathbf{J} = (\mathbf{I} - \gamma \mathbf{P}_{\pi_\mathbf{k}})^{-1} \mathbf{c}_{\pi_k}$

5: $\quad \pi_{k+1} = \pi_g^J$

6: $\quad k = k + 1$

6: **until** $\pi_k = \pi_{k-1}$

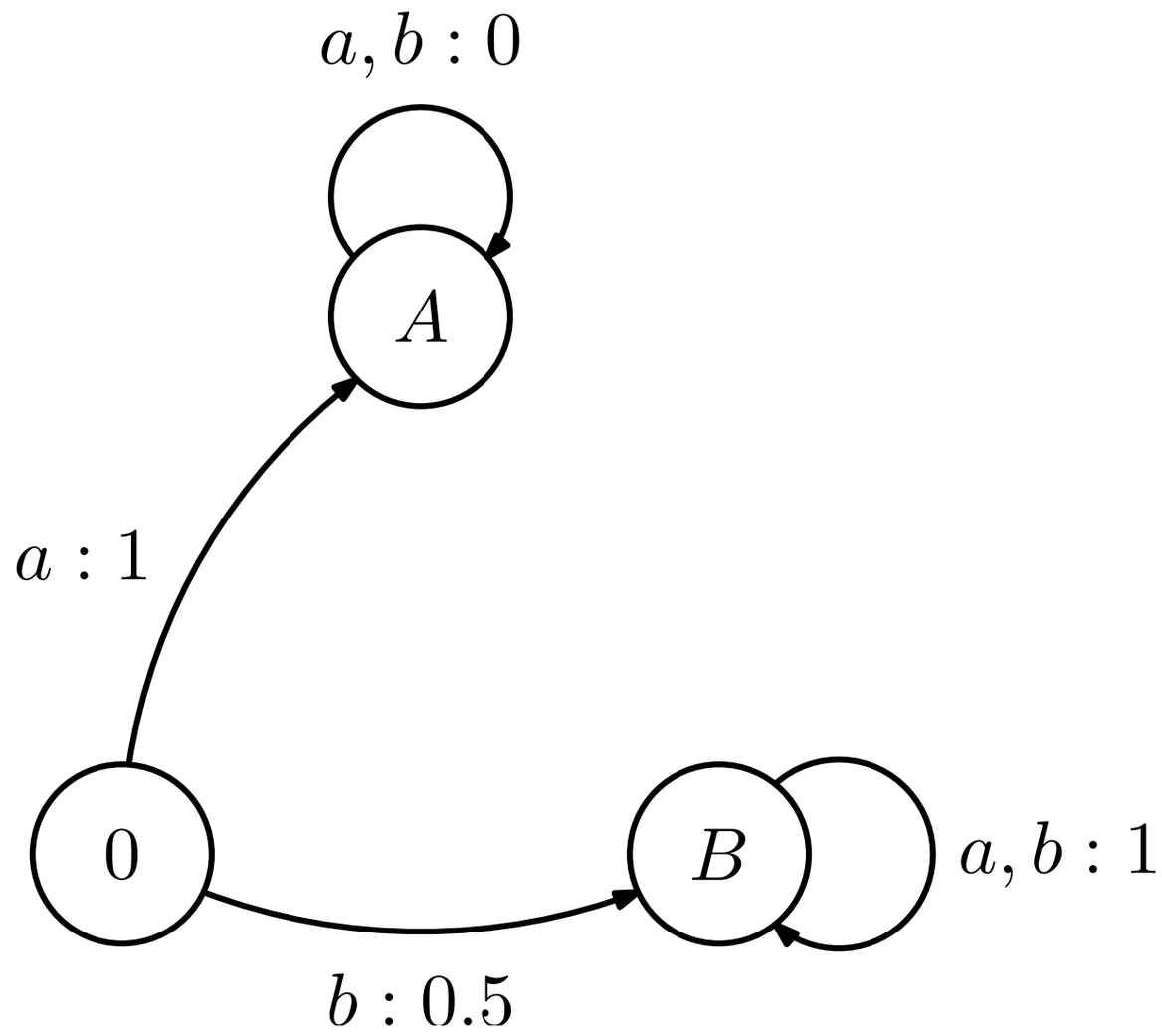
return π_k

- This algorithm is called **policy iteration**

Example

- Let us compute the optimal policy using PI
- Start with the uniform policy

$$\pi_0 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$



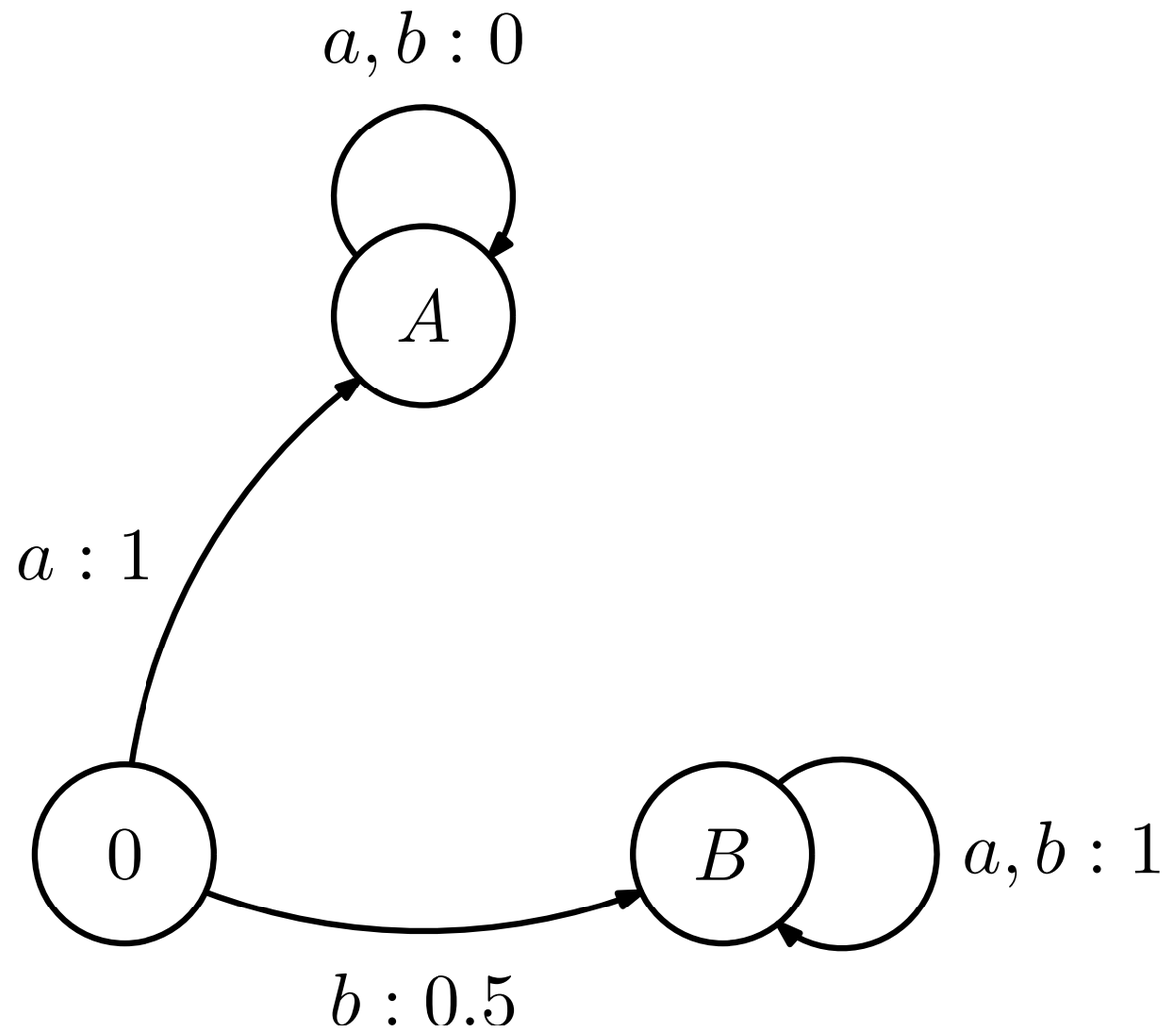
Example

- Step 1: Evaluate π

$$\mathbf{c}_\pi = \begin{bmatrix} 0.75 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{P}_\pi = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{J}^\pi = \begin{bmatrix} 50.25 \\ 0 \\ 100 \end{bmatrix}$$

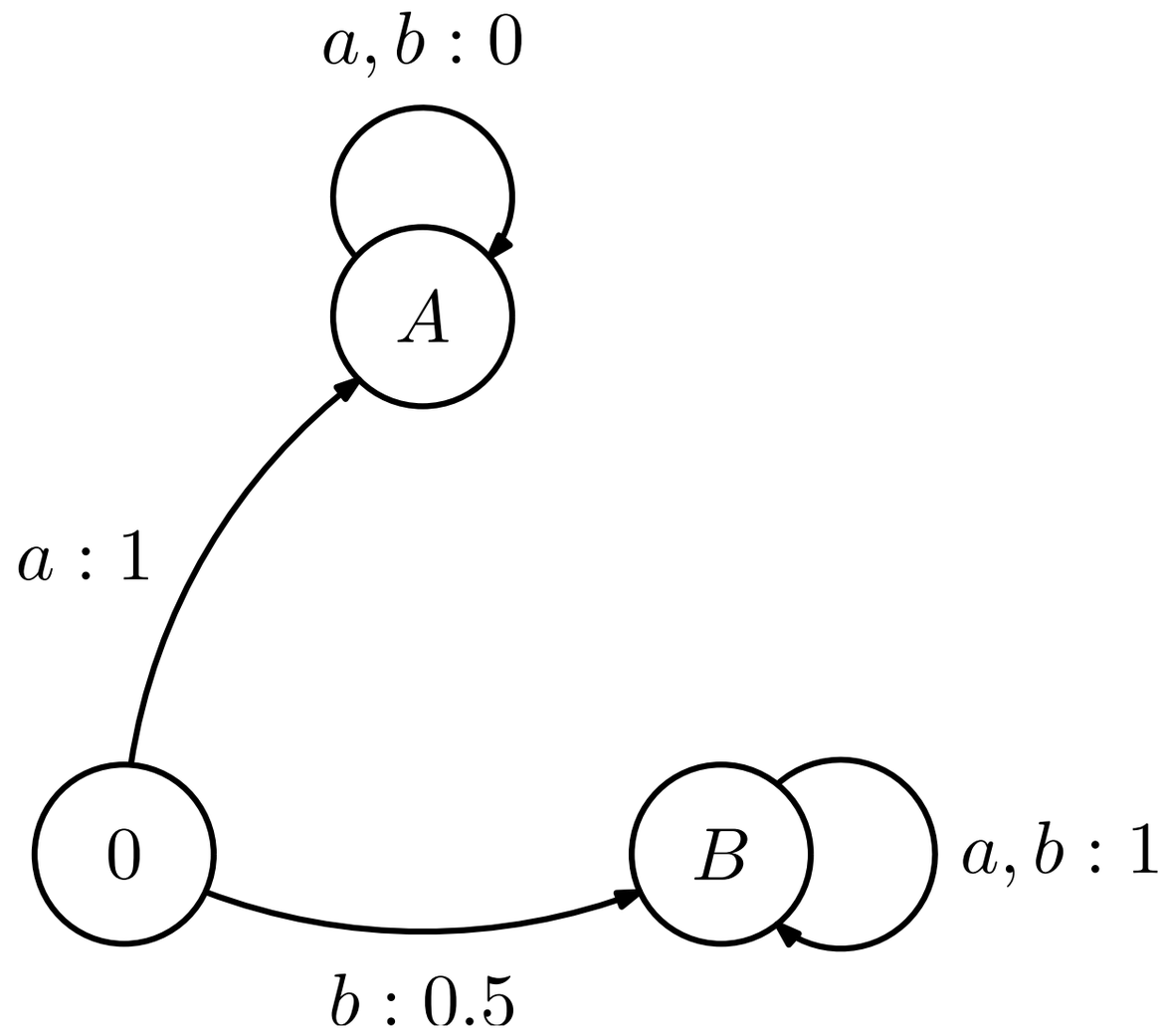


Example

- Step 2: Improve π

$$\mathbf{Q}^{\pi} = \begin{bmatrix} 1 & 99.5 \\ 0 & 0 \\ 100 & 100 \end{bmatrix}$$

$$\pi_1 = \begin{bmatrix} 1 & 0 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$



... and we're done!

Example

```python
def policy_iteration(M):

    # These variables are just to help readibility
    X = M[0]
    A = M[1]
    P = M[2]
    c = M[3]
    gamma = M[4]

    # Initialize pi with the uniform policy
    pol = np.ones((len(X), len(A))) / len(A)
    quit = False
    niter = 0

    while not quit:
        # Auxiliary array to store intermediate values
        Q = np.zeros((len(X), len(A)))

        # Policy evaluation
        cpi = np.sum(c * pol, axis=1, keepdims=True)
        Ppi = pol[:, 0, None] * P[0]

        for a in range(1, len(A)):
            Ppi += pol[:, a, None] * P[a]

        J = np.linalg.inv(np.eye(len(X)) - gamma * Ppi).dot(cpi)

        # Compute Q-values
        for a in range(len(A)):
            Q[:, a, None] = c[:, a, None] + gamma * P[a].dot(J)

        # Compute greedy policy
        Qmin = np.min(Q, axis=1, keepdims=True)

        pnew = np.isclose(Q, Qmin, atol=1e-8, rtol=1e-8).astype(int)
        pnew = pnew / pnew.sum(axis = 1, keepdims = True)

        # Compute stopping condition
        quit = (pol == pnew).all()

        # Update
        pol = pnew
        niter += 1

    print(f'Done after {niter} iterations.')
    return np.round(pol, 3)

pol = policy_iteration(M)
print(pol)
```

```
Done after 2 iterations.
[[1.  0. ]
 [0.5 0.5]
 [0.5 0.5]]
```

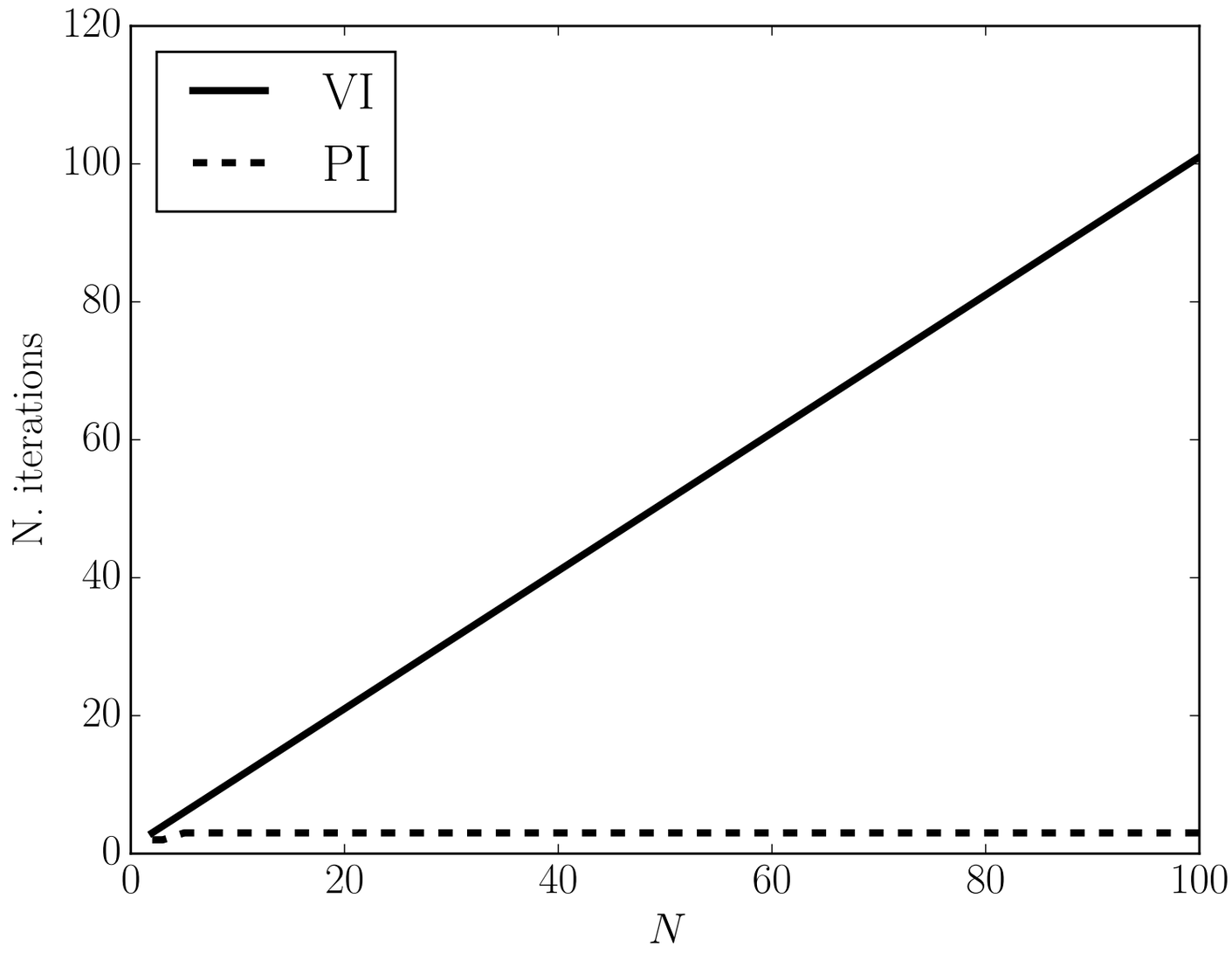
Value iteration took 1834 iterations...

Which should we use?

VI or PI?

Computational efficiency

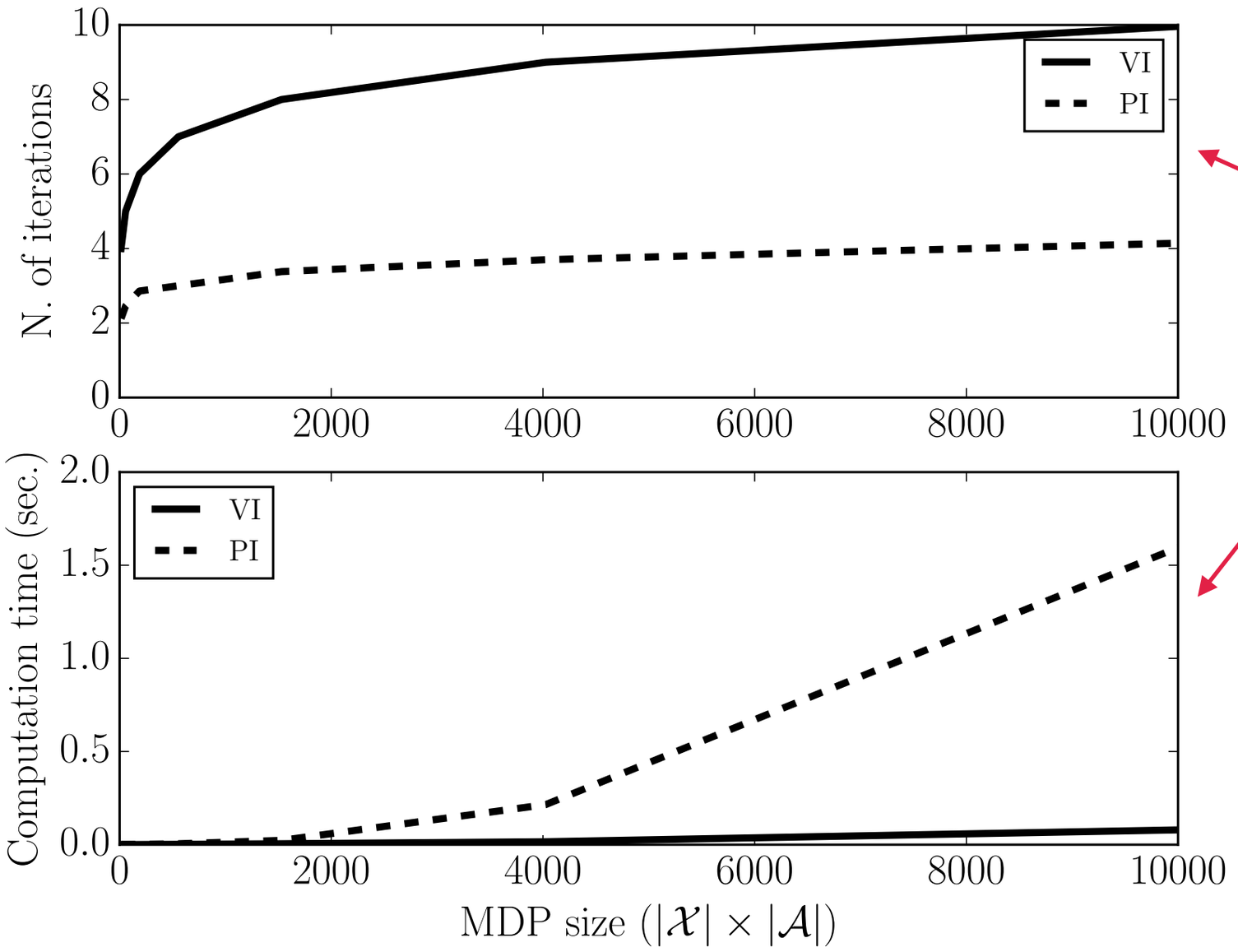
- Results in an MDP with N states:



Careful with this plot!

Computational efficiency

- Another example...

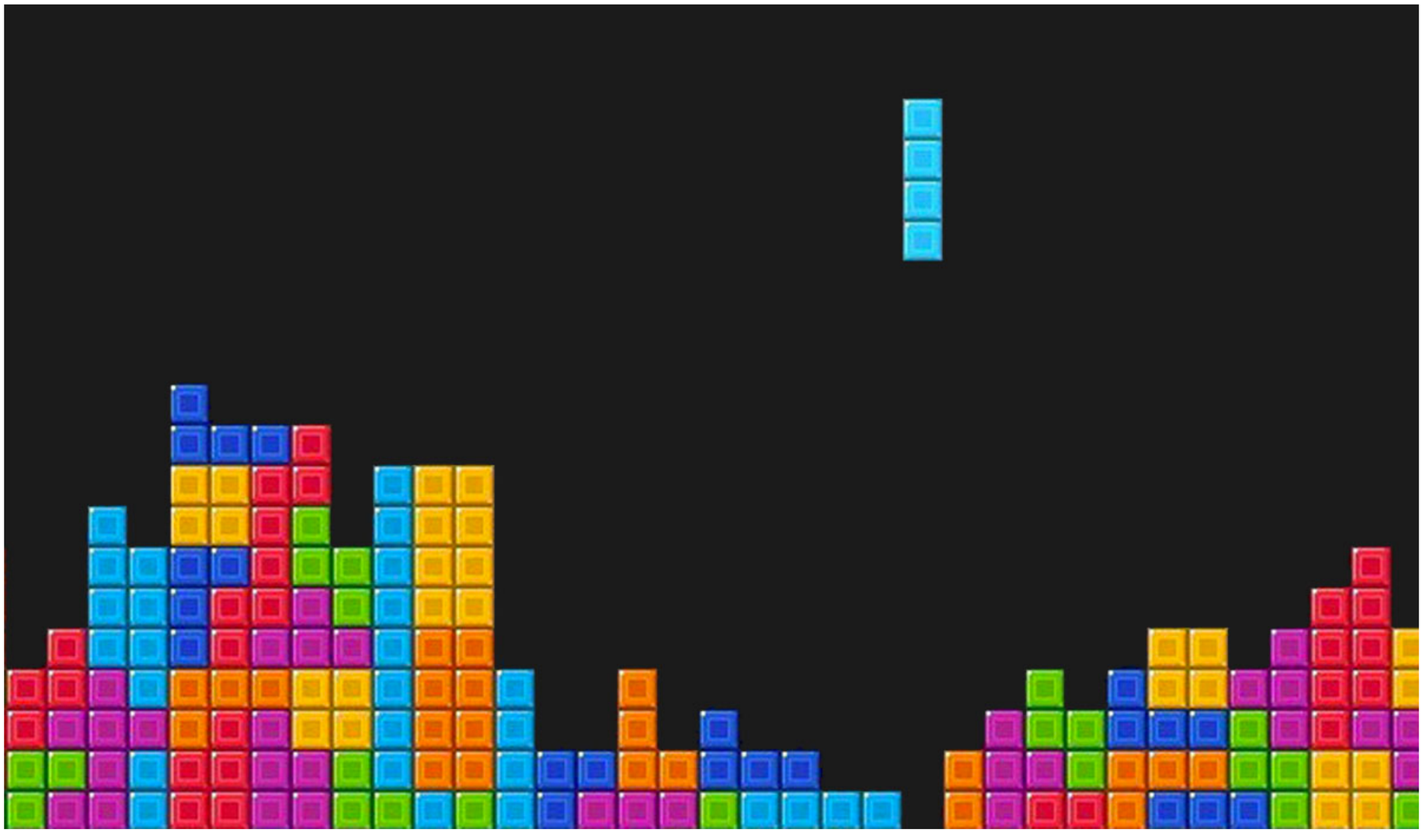


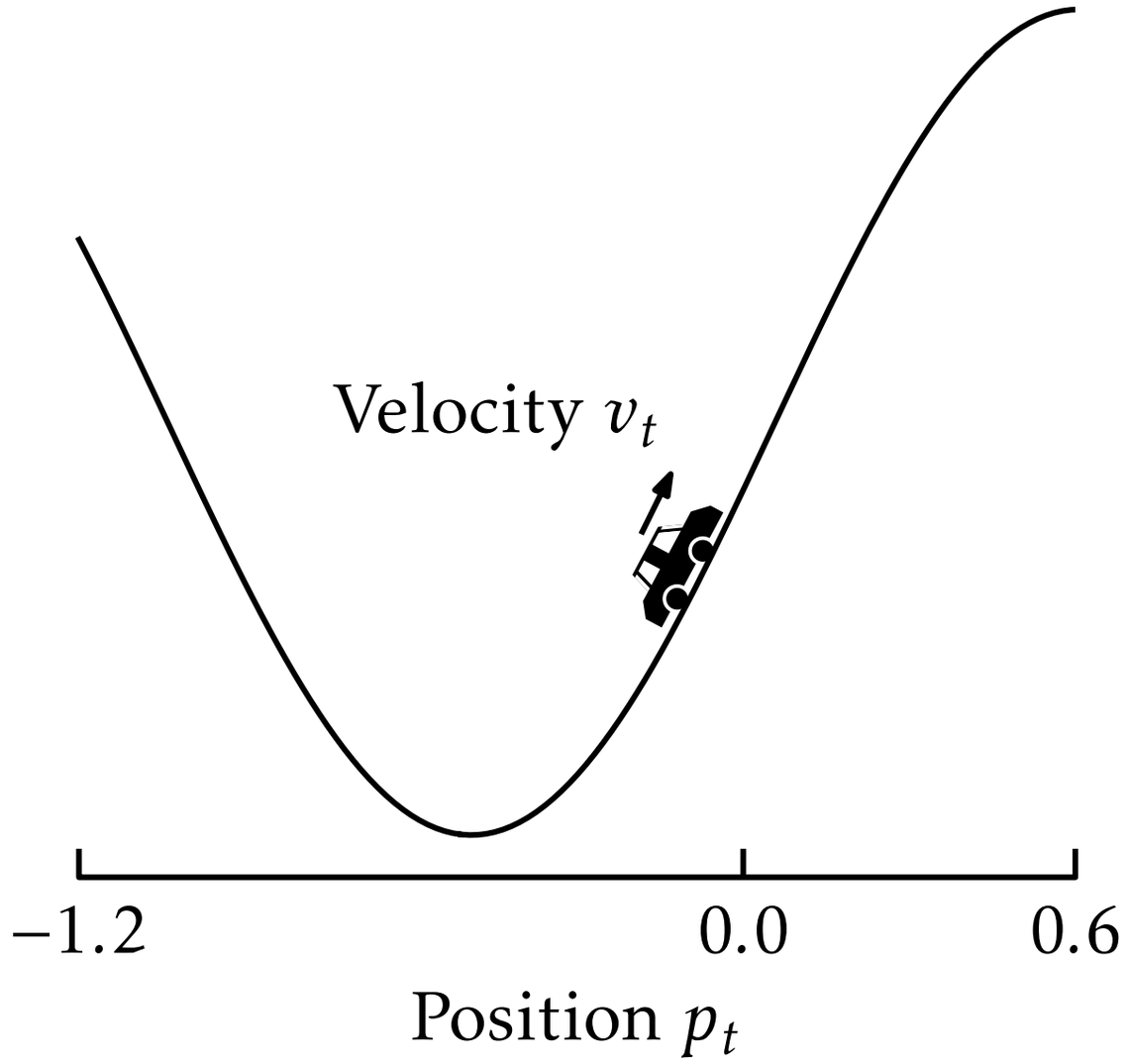
Although VI takes more iterations, it takes less time

Large problems

The curse of dimensionality

- Larger problems cannot be solved exactly

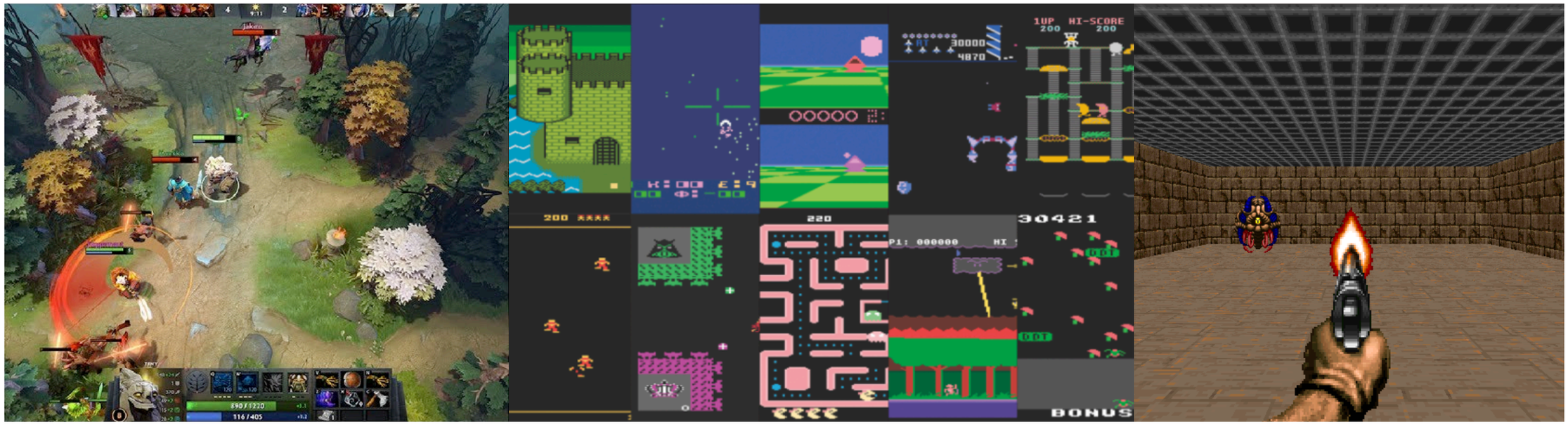




Mountain-car

TÉCNICO LISBOA

Board games

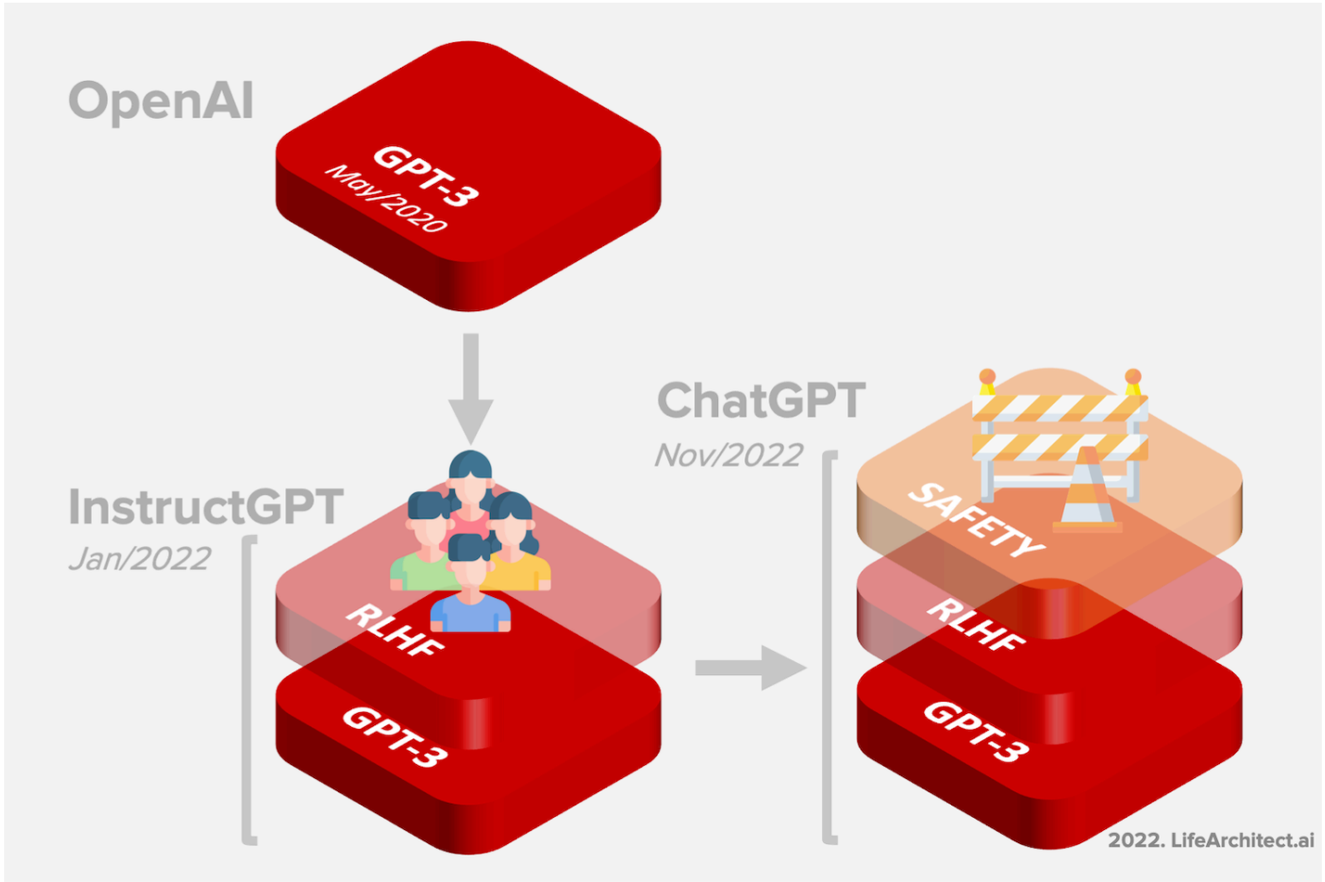


Video games

Poker

TÉCNICO LISBOA

Robotics


OpenAI
GPT-3
May/2020
InstructGPT
Jan/2022
RLHF
GPT-3
ChatGPT
Nov/2022
SAFETY
RLHF
GPT-3
2022. LifeArchitect.ai

NLP

The curse of dimensionality

- We must resort to some form of approximation
- Instead of allowing arbitrary functions, methods restrict to pre-specified families of functions
  - Families provide good representations → methods perform well
  - Families provide bad representations → methods perform poorly

Examples of approximations

- **State aggregation**
  - States are "aggregated" into "chunks"
  - Each chunk is treated as a "super-state"
  - Very limited representation power

Methods retain convergence guarantees

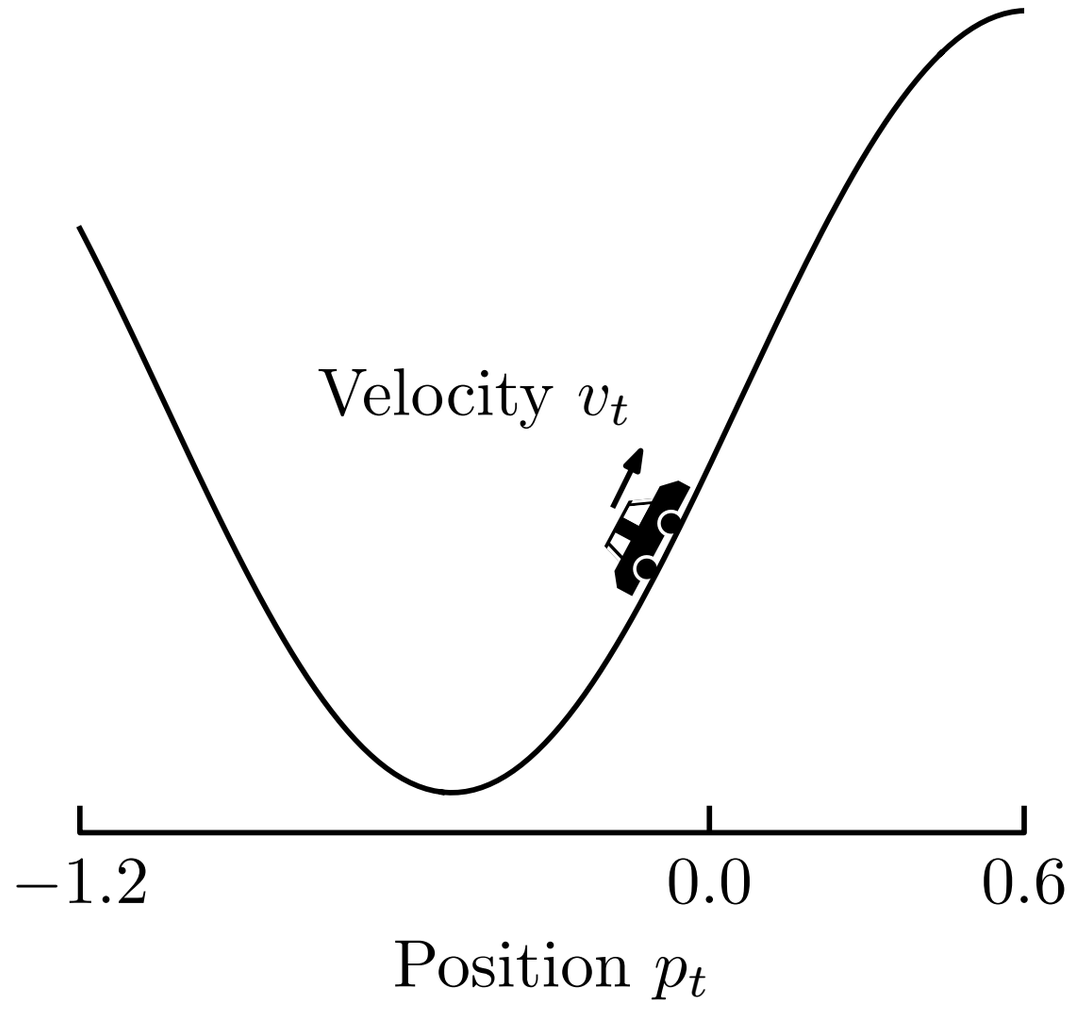
Examples of approximations

- **Linear approximations**
  - States described by vector of "features"
  - J and Q are represented as combinations of features
  - Good representation power; difficult to choose good features

Methods **do not**
retain convergence
guarantees

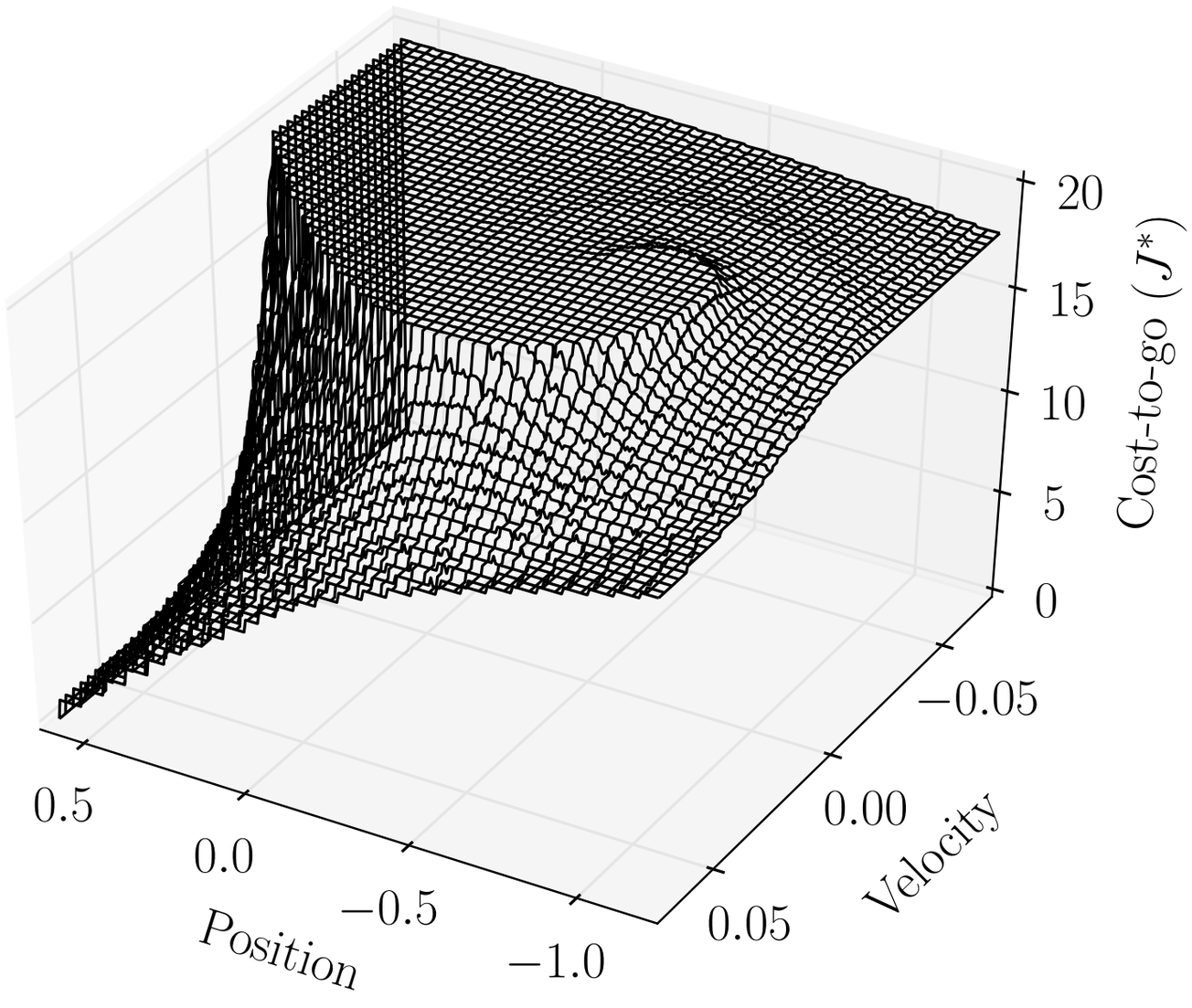
Example: The mountain car

- A car goes up a mountain
- Engine is not strong enough to go all the way up
- Car must run back and forth



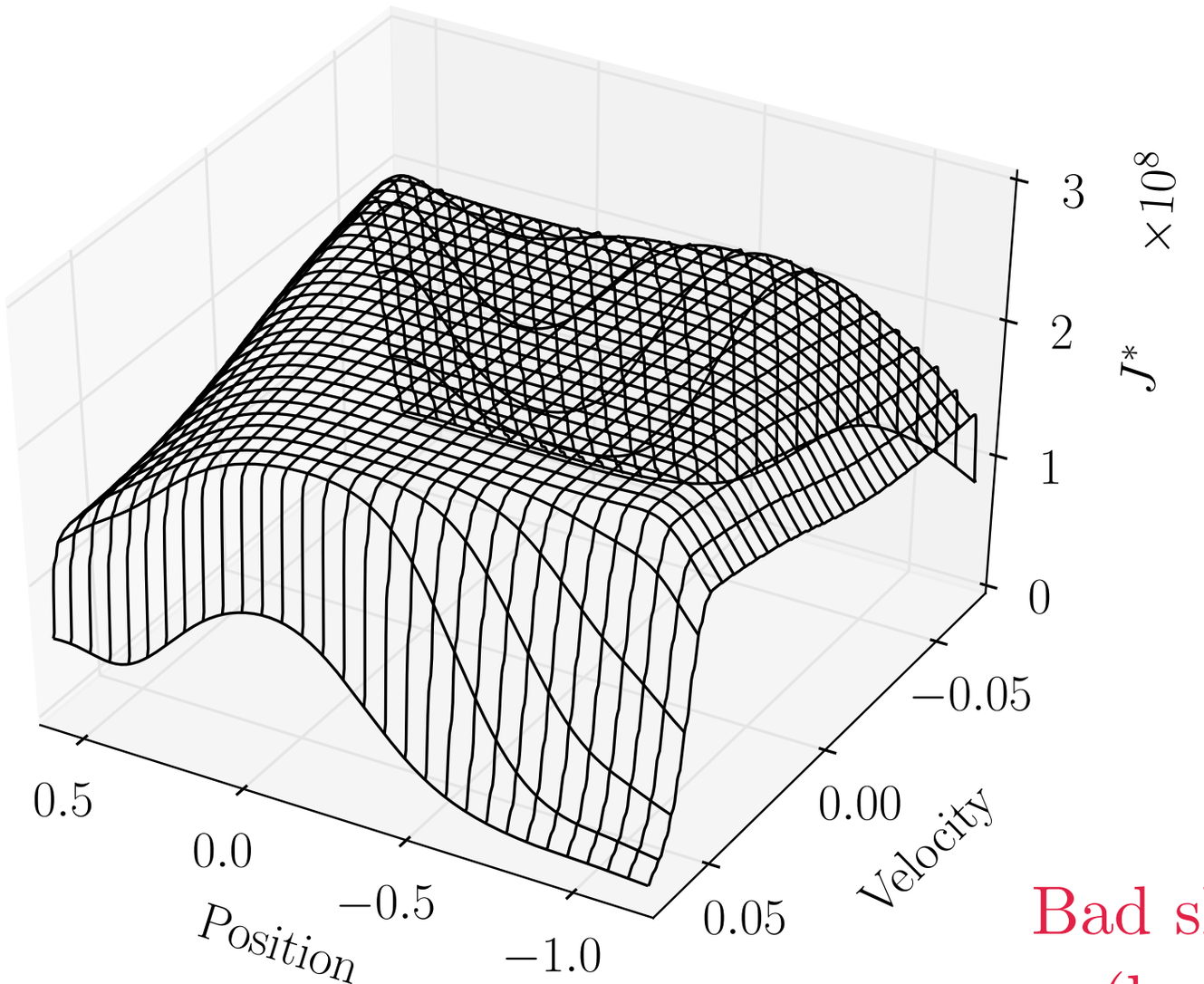
Example: The mountain car

- The actual cost-to-go:



Example: The mountain car

- A naive (linear) approximation:



Bad shape and scale
(look at z-axis!)

Examples of approximations

- **Averagers**
  - Approximations that do not **extrapolate**
  - Such architectures are known as **averagers**

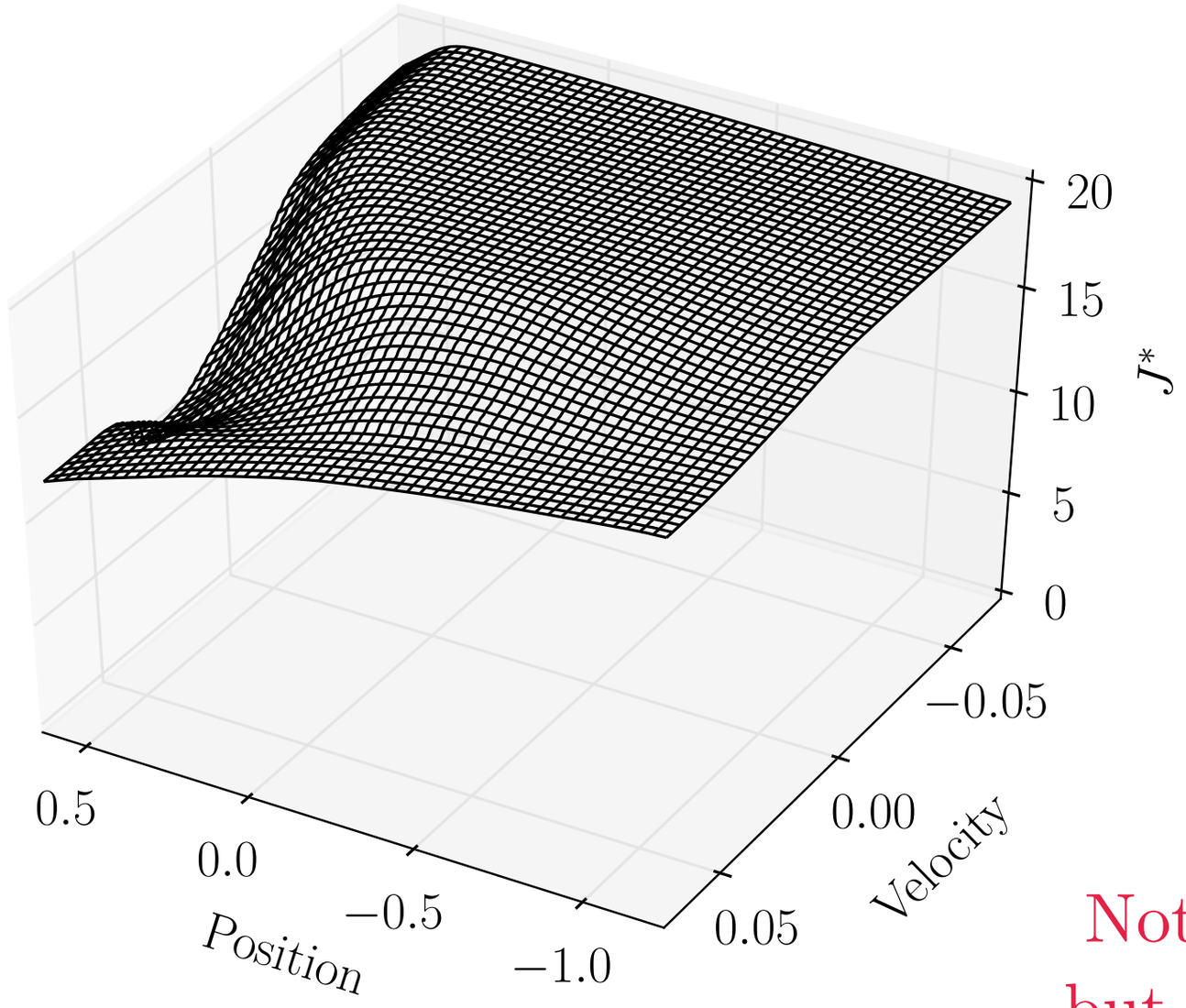
Methods **do retain**

convergence

guarantees

Example: The mountain car

- An averager:



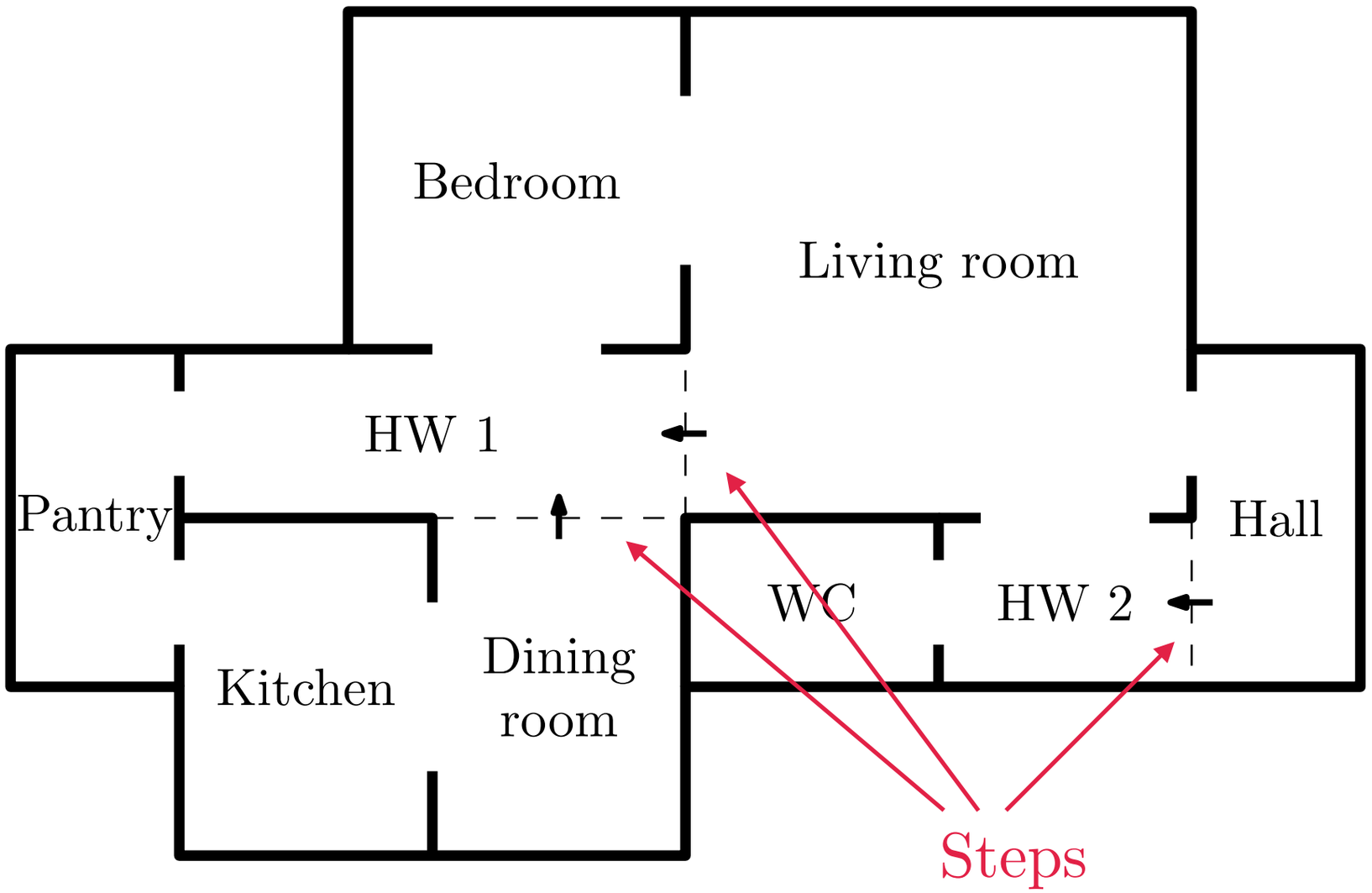
Not all the detail,
but the shape is ok

… Enter partial observability…


The household robot...
again!

Household robot

- Consider the household



Household robot

- Robot moves in the environment, assisting human users
- When at the Hall, receives a request from the Kitchen

One "movement", one decision

Household robot

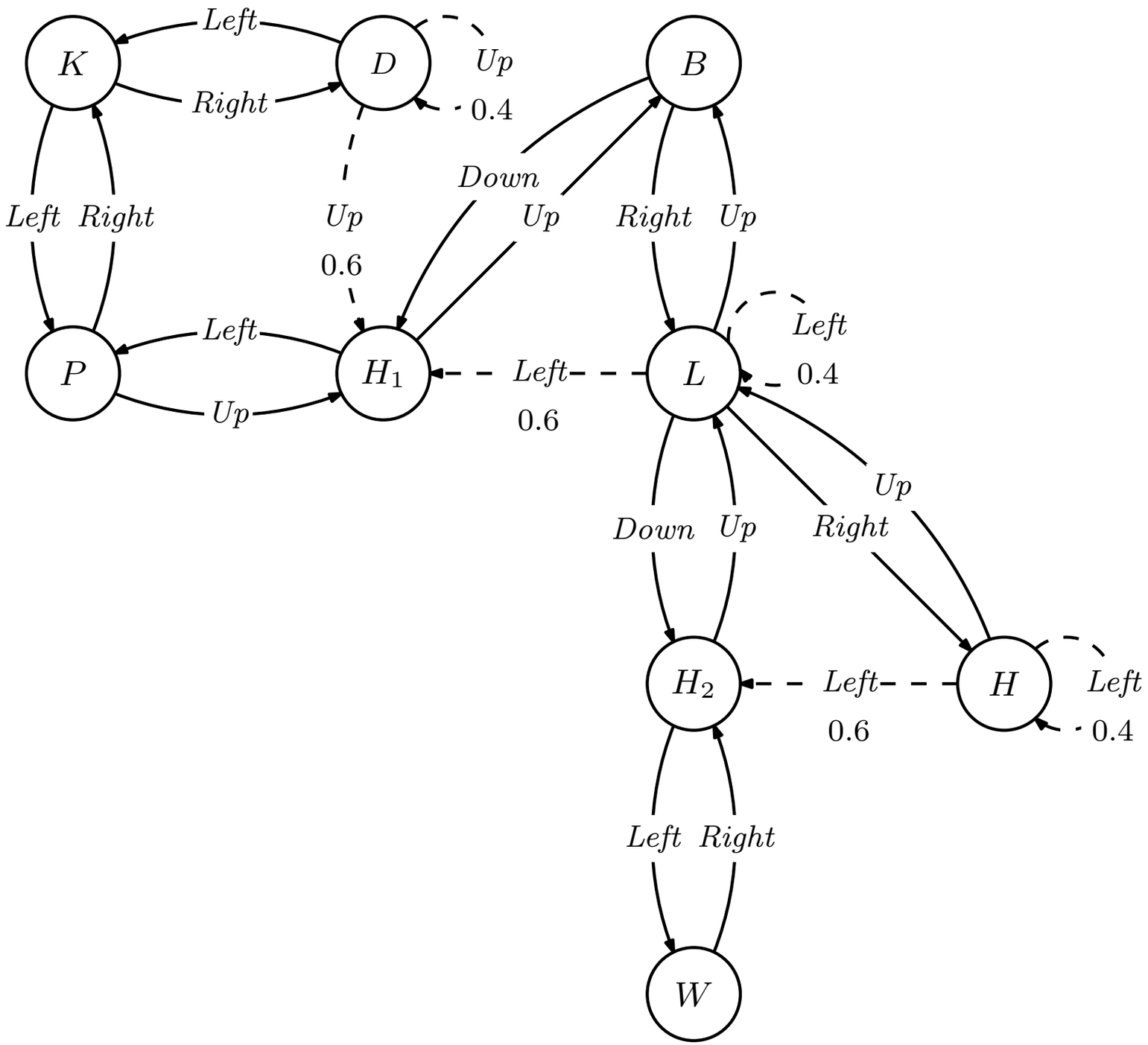
- At each step, the robot has available a set of actions:

$$\mathcal{A} = \{\text{U(p), D(own), L(eft), R(ight), S(tay)}\}$$

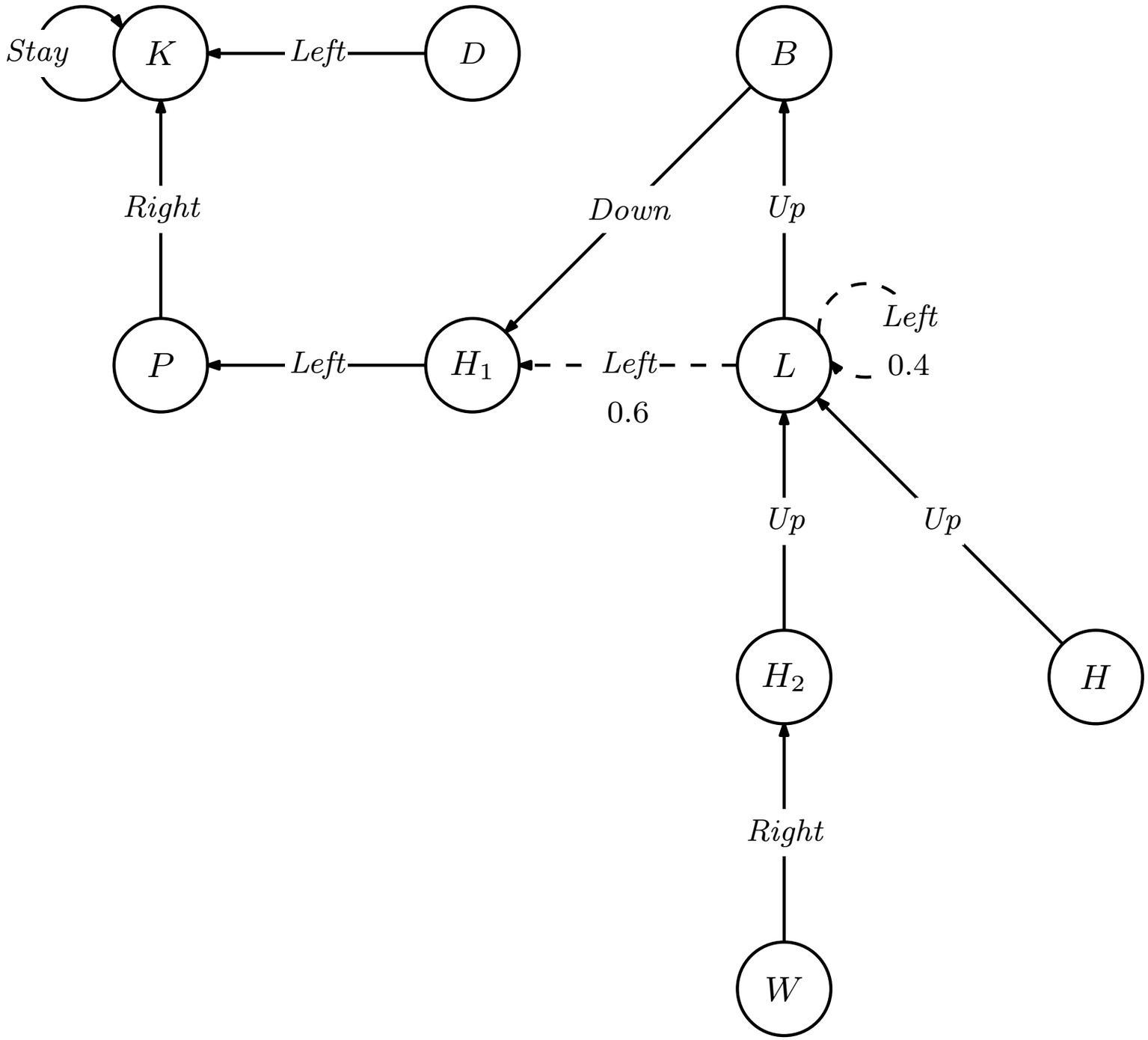
Household robot

- Motions across a step fail with probability 0.4

Movement of the robot

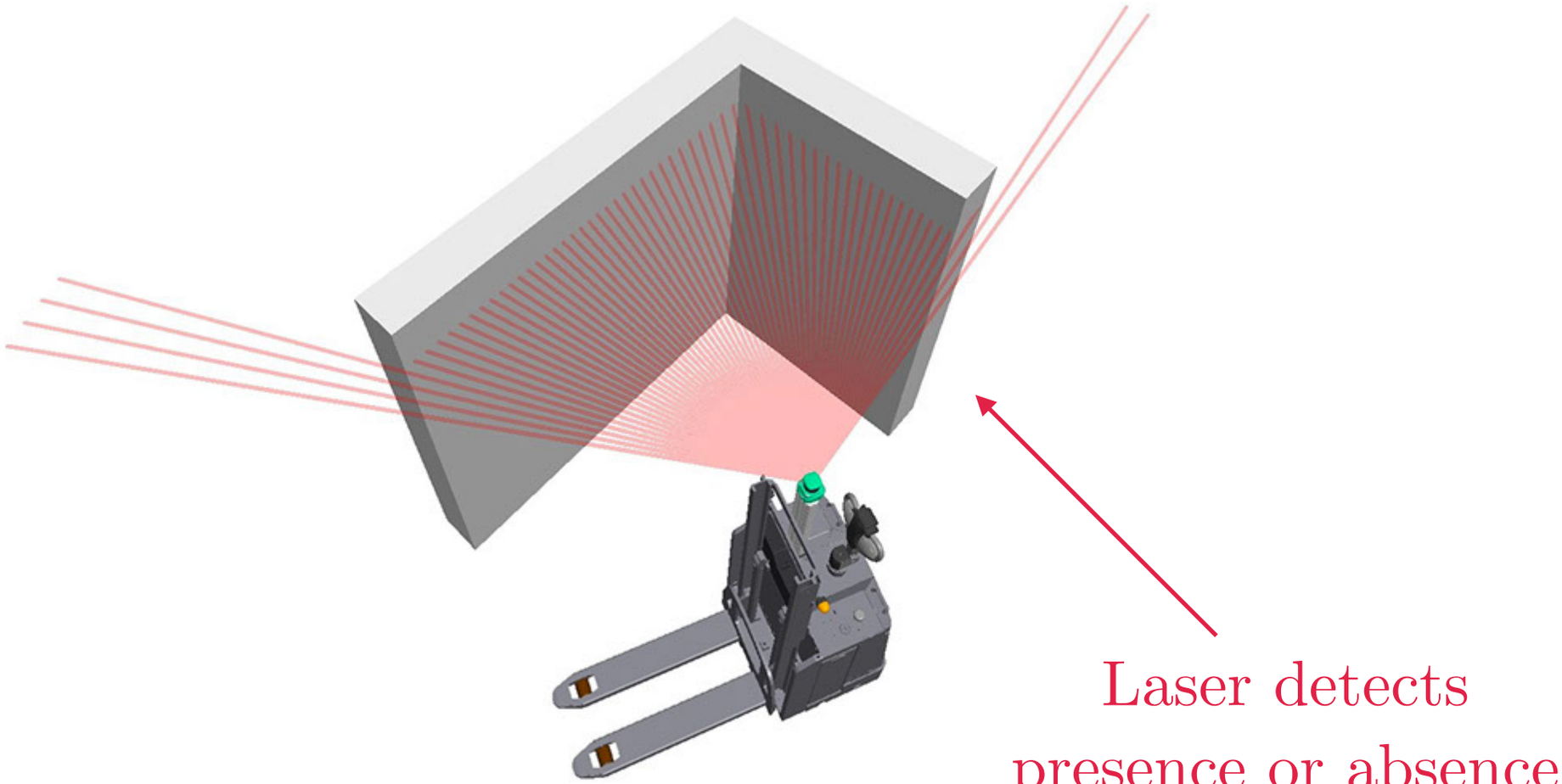


Movement of the robot



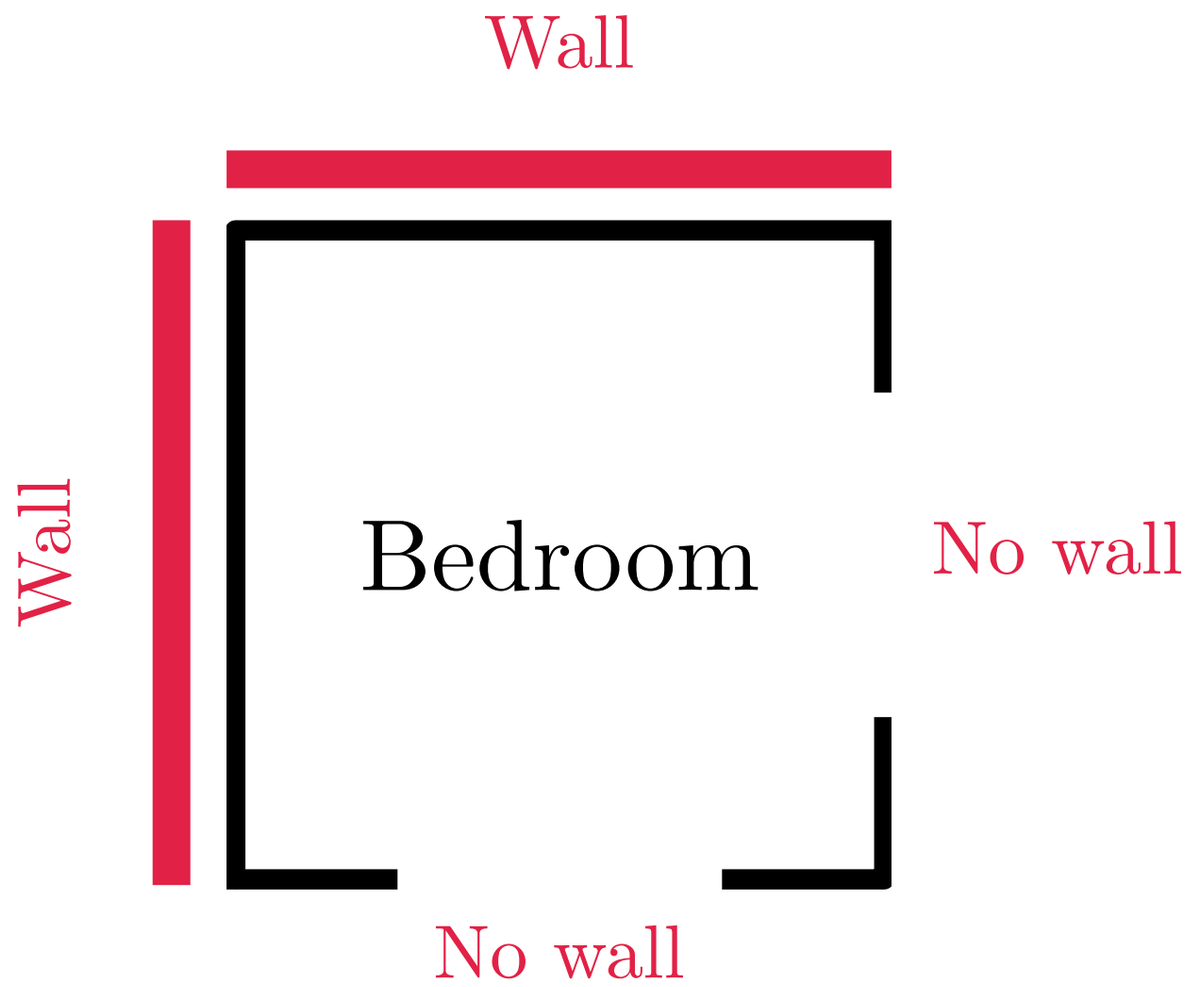
Household robot

- Robot navigates using a laser



Household robot

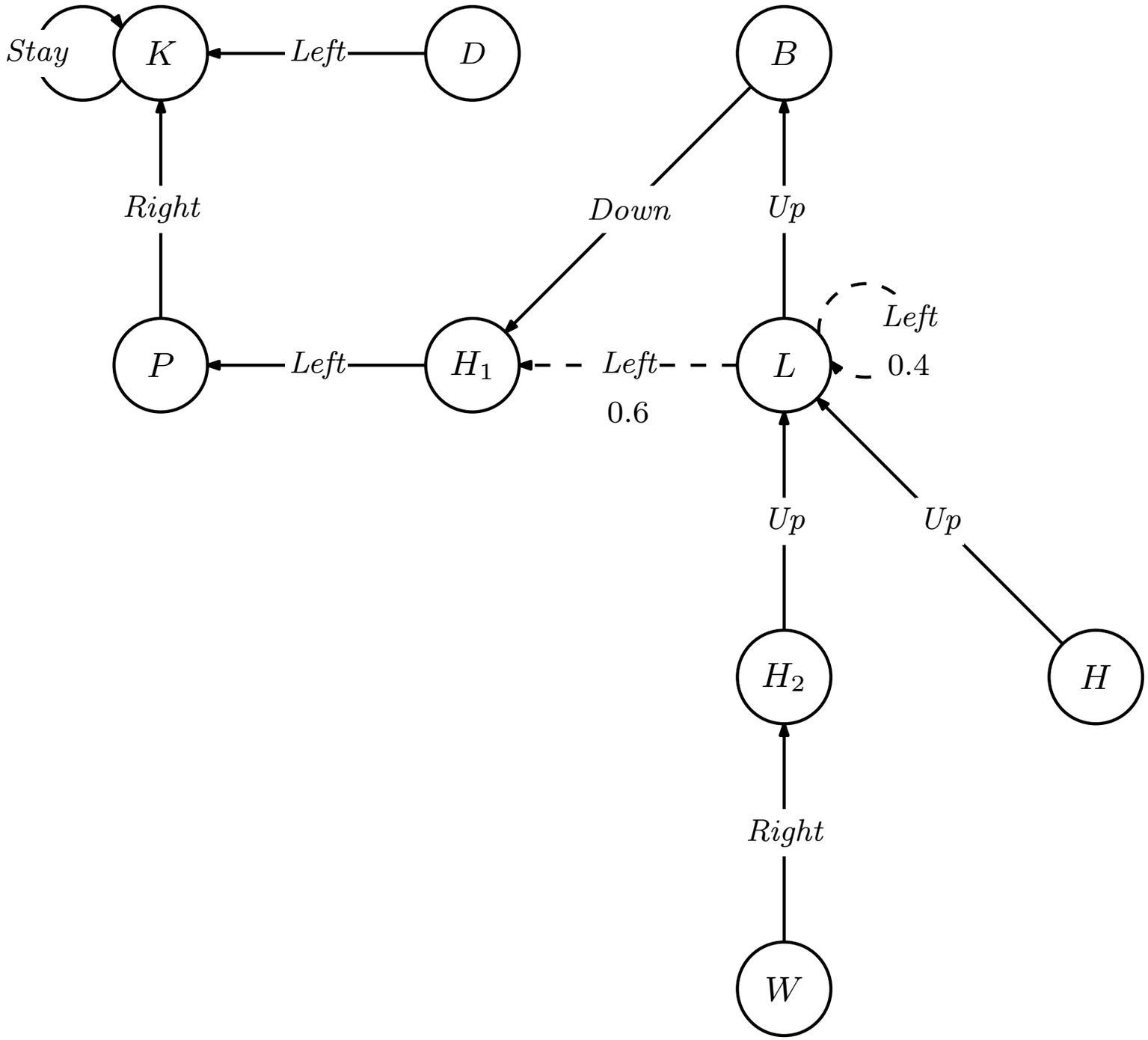
- For example:



Household robot

- However, laser is not perfect
  - It fails to detect existing walls with 5% probability
  - It detects non-existing walls with 10% probability (in some situations with 20% probability)
  - Detection of a wall independent of adjacent walls

Movement of the robot



Unfortunately...

- At each step, what does the decision of the robot depend on?
  - Position of the robot

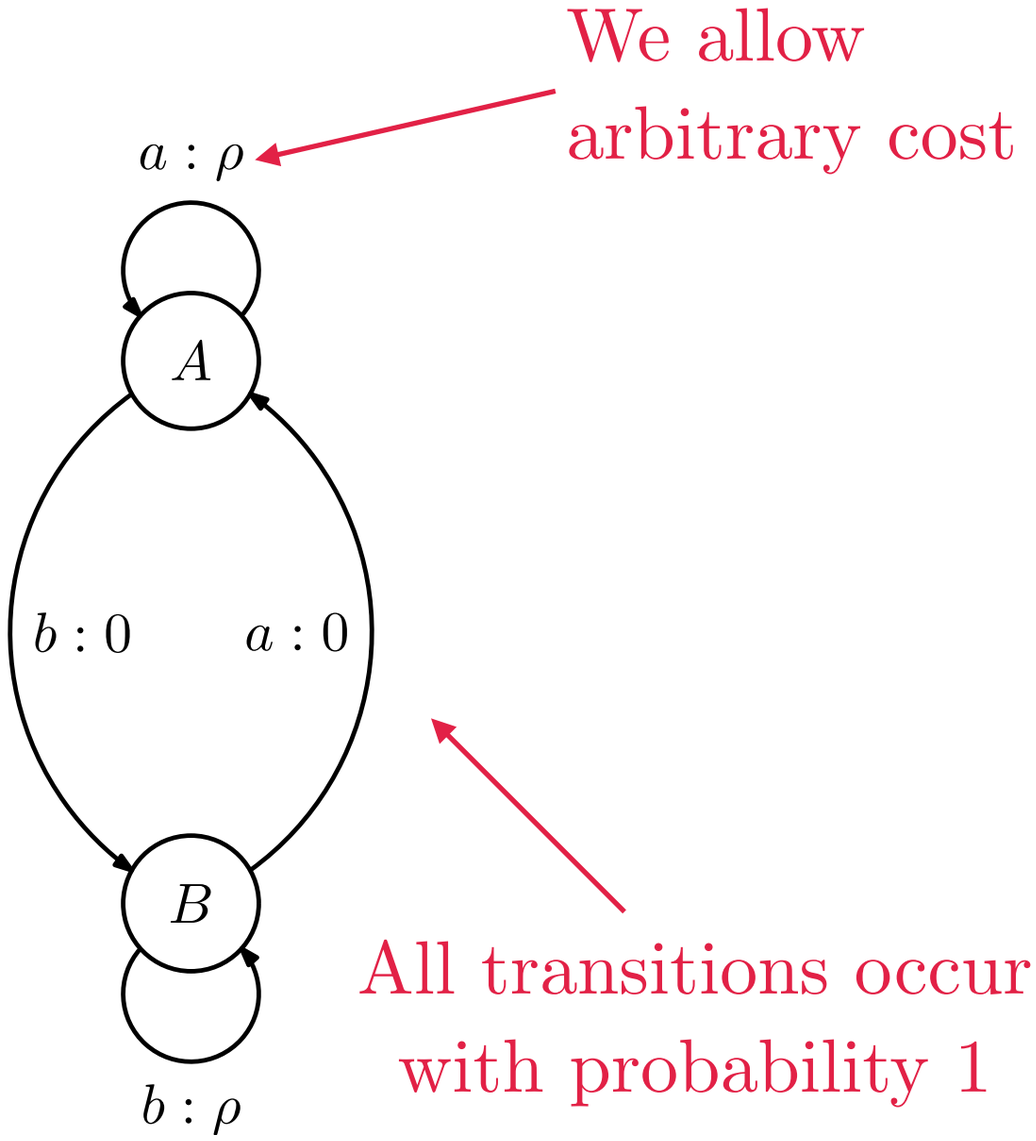
Position is not
directly observable!

Decision must be
based on observations!

The two-state nightmare

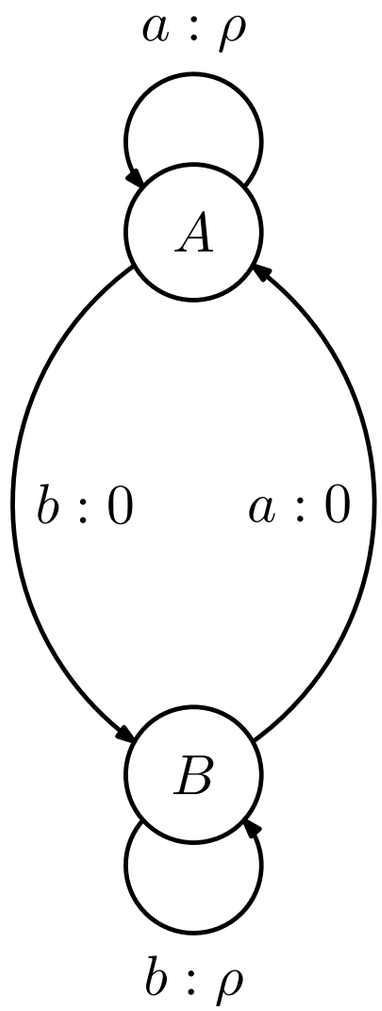
2-state problem

- Consider the following problem:



2-state problem

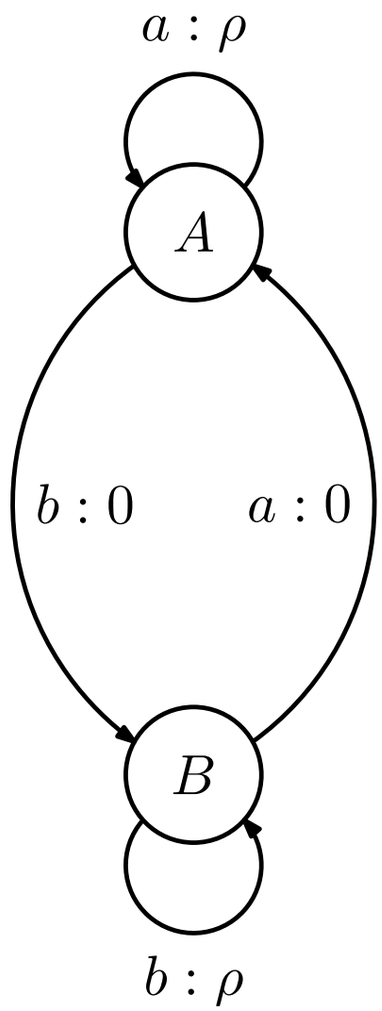
- What is the optimal policy?
  - Select action b in state A
  - Select action a in state B



2-state problem

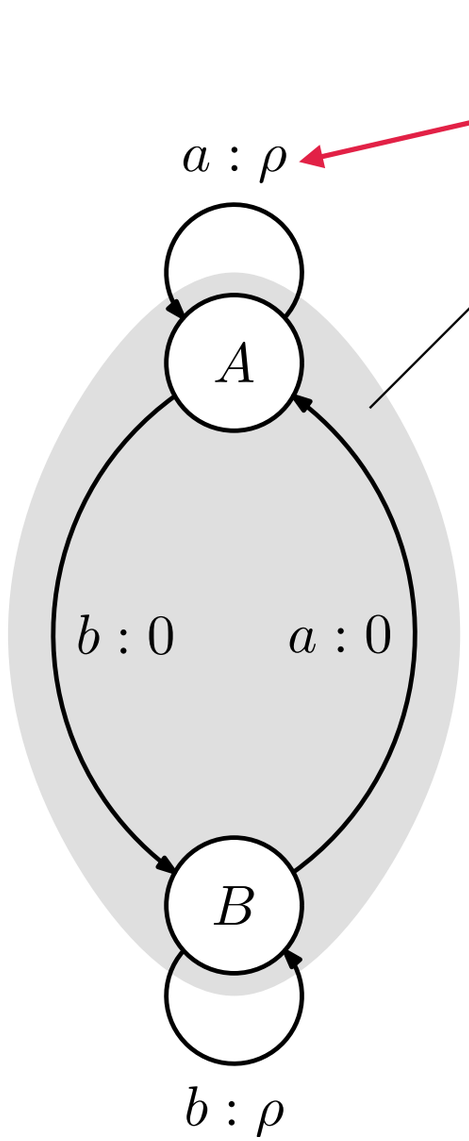
- What is the optimal cost-to-go?
  - Every step a cost of zero, so:

$$J^*(x) = 0$$



2-state problem, v. 2.0

- Consider the following problem:



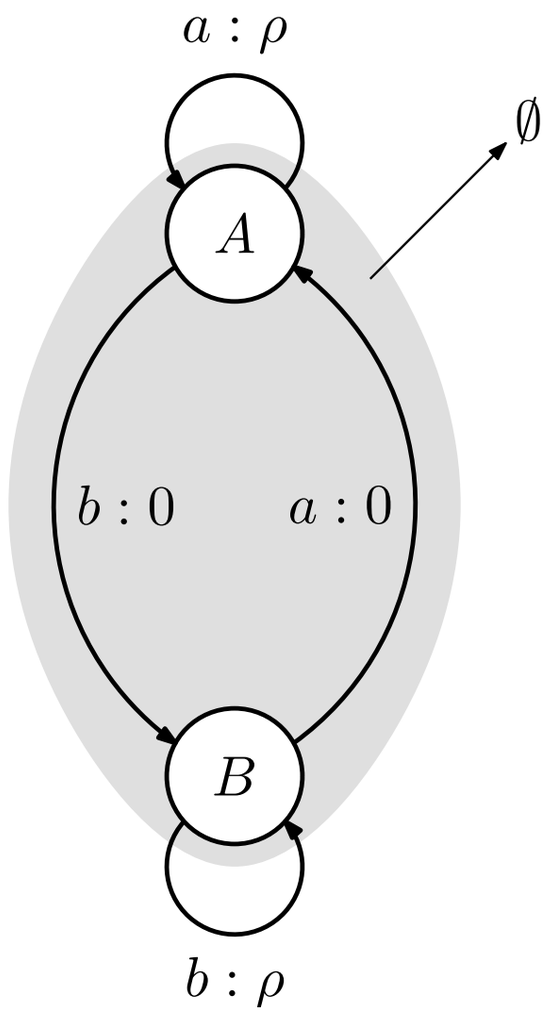
We allow arbitrary cost

No observations

All transitions occur with probability 1

2-state problem

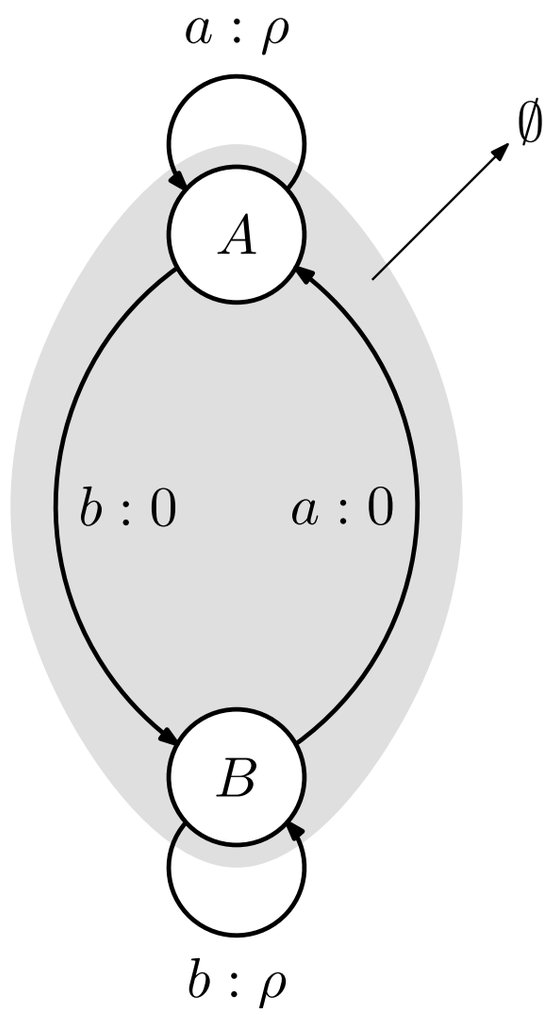
- What is the optimal policy?
  - Not obvious...



2-state problem

Tentative 1:

- Ignore partial observability
- Select actions deterministically
  - "Memoryless policy"

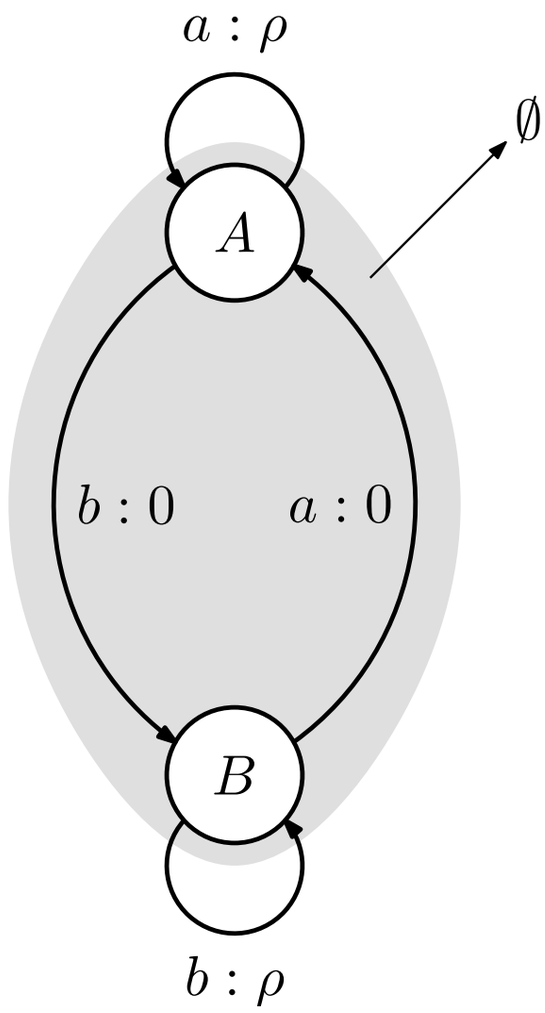


Decision must be constant!

2-state problem

- What is the cost-to-go?
  - Always select a
  - Best case: 0 followed by infinite ρs

$$J(x) = 0 + \gamma\rho + \gamma^2\rho + \dots$$
$$= \frac{\gamma\rho}{1-\gamma}$$



2-state problem

- Comparing with the optimal one:

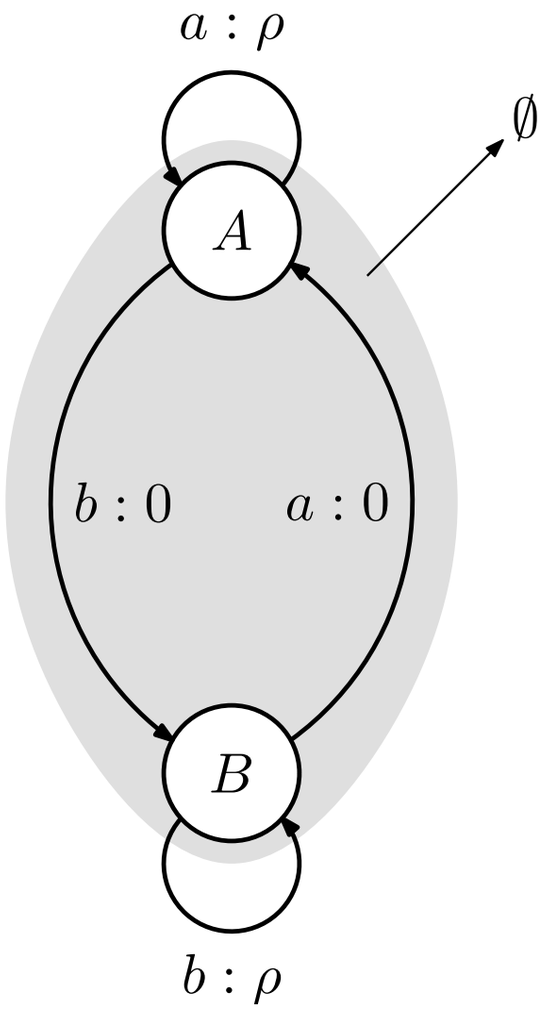
$$\frac{\gamma\rho}{1-\gamma} > 0$$

The best memoryless policy can be
arbitrarily worse than the best MDP policy!

2-state problem

Tentative 2:

- Ignore partial observability
- Select actions stochastically

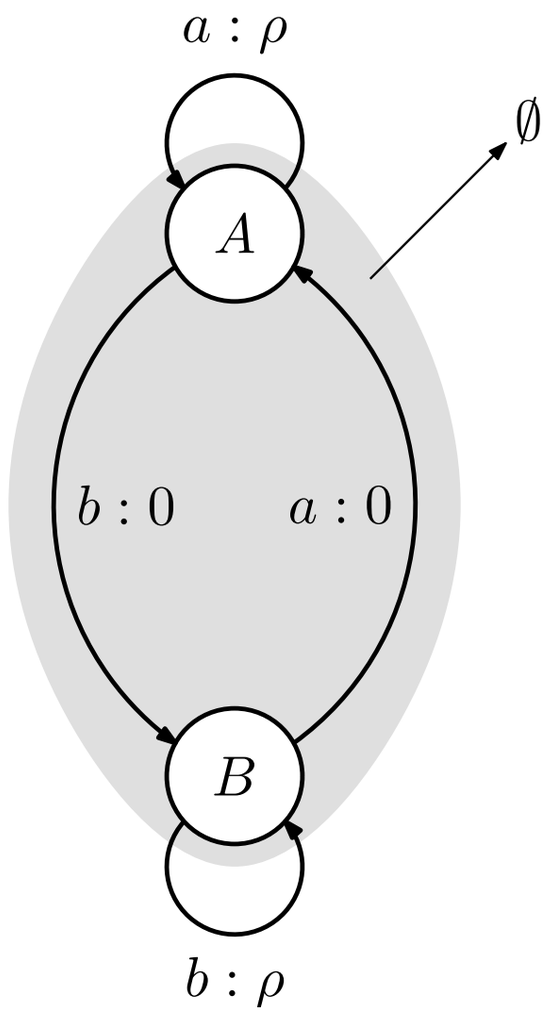


Select each action with probability 0.5

2-state problem

- What is the cost-to-go?
  - In each step incur an average cost of $\rho/2$
  - Therefore:

$$J(x) = \frac{\rho}{2} + \gamma\frac{\rho}{2} + \gamma^2\frac{\rho}{2} + \dots$$
$$= \frac{\rho}{2(1-\gamma)}$$



2-state problem

- Comparing with the deterministic one:

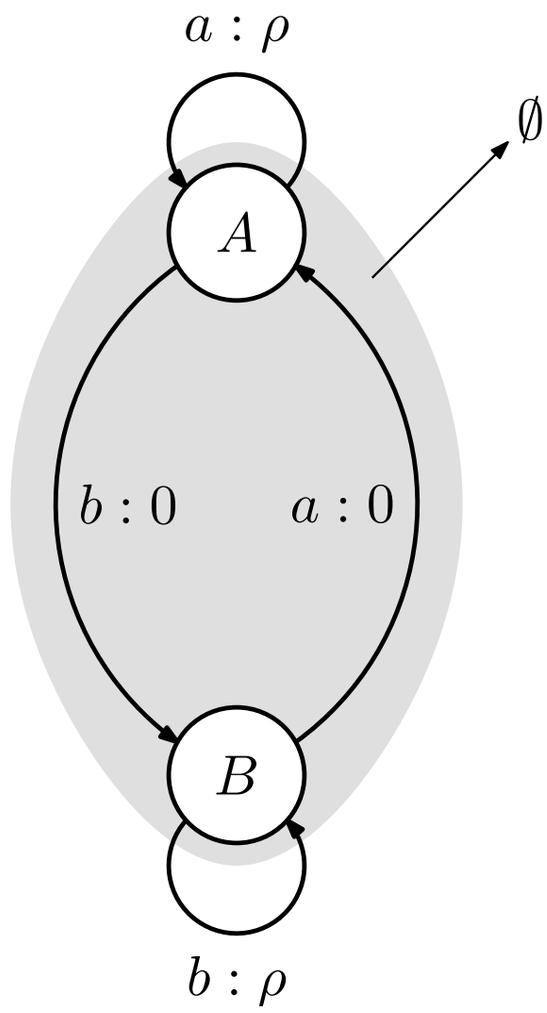
$$\frac{\rho}{2(1-\gamma)} < \frac{\gamma\rho}{1-\gamma} \qquad \text{if } \gamma > \frac{1}{2}$$

The best deterministic policy can be
arbitrarily worse than the best stochastic policy!

2-state problem

Tentative 3:

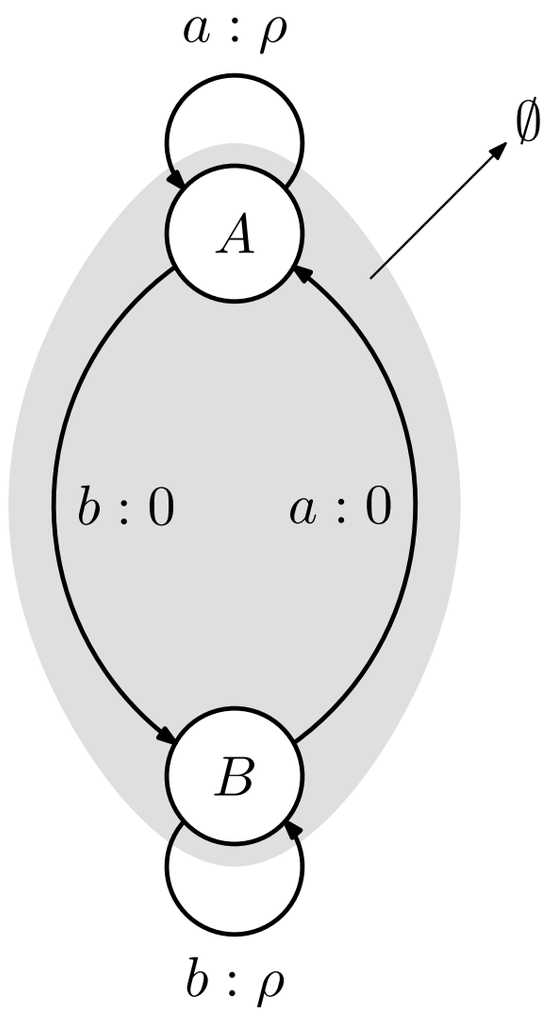
- Non-stationary policy:
  - Alternate action selection



2-state problem

- What is the cost-to-go?
  - Worst case: ρ followed by infinite 0s
  - Therefore:

$$J(x) = \rho + \gamma 0 + \gamma^2 0 + \dots$$
$$= \rho$$



2-state problem

- Comparing with the deterministic one:

$$\rho < \frac{\gamma\rho}{1-\gamma} \qquad \text{if } \gamma > \frac{1}{2}$$

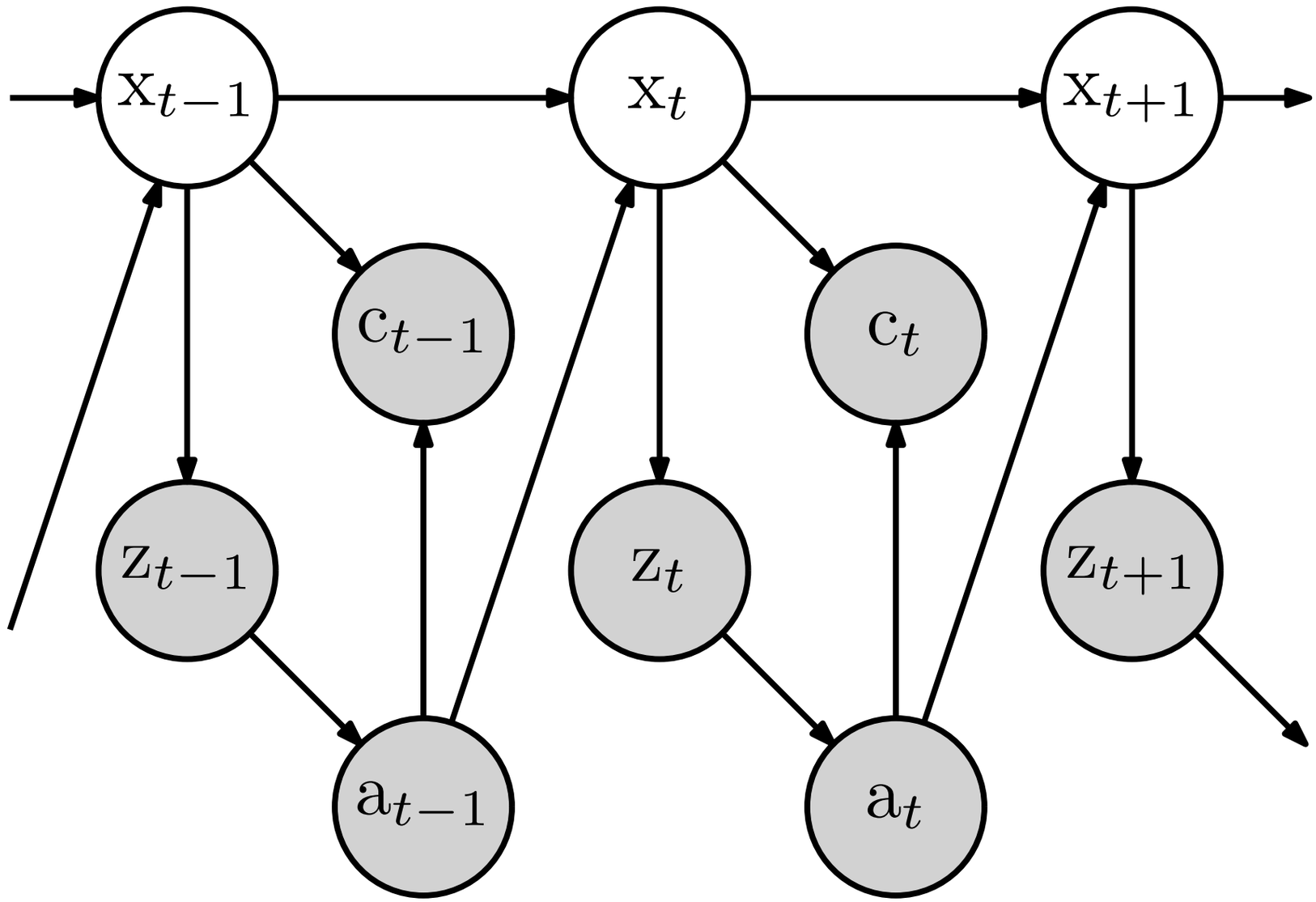
The best deterministic policy can be arbitrarily worse than the best non-stationary policy!

2-state problem

- Comparing with the stochastic one:

$$\rho < \frac{\rho}{2(1-\gamma)} \qquad \text{if } \gamma > \frac{1}{2}$$

The best stochastic policy can be arbitrarily worse than the best non-stationary policy!



Partially observable MDPs

States

- Relevant information for decision making
- We represent the state at time t as x_t
- Set of possible states is $\mathcal{X}$ (finite, most of the time)
- Each step, the agent makes a decision (**decision epoch**)

Action

- Means by which the agent influences the "environment"
- We represent the action at time t as a_t
- Set of possible actions is $\mathcal{A}$ (finite)

Dynamics

- Describe how the state evolves as a consequence of the agent's actions
- We assume that it verifies the **Markov property**

Markov property

Key Property: Markov property

The state at instant $t+1$ depends only on the state and action at time step t, i.e.,

$$\mathbb{P}\left[\mathrm{x}_{t+1} = y \mid \mathbf{x}_{0:t} = \boldsymbol{x}_{0:t}, \mathbf{a}_{0:t} = \boldsymbol{a}_{0:t}\right] = \mathbb{P}\left[\mathrm{x}_{t+1} = y \mid \mathrm{x}_t = x_t, \mathrm{a}_t = a_t\right]$$

Additional assumptions:

- The probabilities $\boxed{\mathbb{P}[\mathrm{x}_{t+1} = y \mid \mathrm{x}_t = x, \mathrm{a}_t = a]}$ do not depend on t

Transition probability from x
to y given a

- For each action $a \in \mathcal{A}$, we store the transition probabilities in a **matrix**

$$[\mathbf{P}_a]_{xy} = \mathbb{P}\left[\mathrm{x}_{t+1} = y \mid \mathrm{x}_t = x, \mathrm{a}_t = a\right]$$

Immediate costs

- Instantaneously evaluates **state and action**
- Represented as a function $c : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$
- For simplicity, we assume that $c(x, a) \in [0, 1]$

So far, everything looks like an MDP...

Observations

- Information that the agent actually "**sees**"
- We represent the observation at time t as z_t
- Set of possible observations is $\mathcal{Z}$ (finite)
- Observations depend on current state **and previous action**

Perception

- Describe how the observations depend on the state and the agent's actions
- We assume that observations depend only on the state and (previous) action

State-dependent observations

State-dependent observations

The state at instant t and action at instant $t-1$ are enough to predict the observation at instant t:

$$\mathbb{P}[\mathrm{z}_t = z \mid \mathbf{x}_{0:t} = \boldsymbol{x}_{0:t}, \mathbf{a}_{0:t-1} = \boldsymbol{a}_{0:t-1}, \mathbf{z}_{0:t-1} = \boldsymbol{z}_{0:t-1}] = \\ = \mathbb{P}[\mathrm{z}_t = z \mid \mathrm{x}_t = x_t, \mathrm{a}_{t-1} = a_{t-1}]$$

Depends only on x_t and a_{t-1}

Additional assumptions:

- The probabilities $\boxed{\mathbb{P}[\mathrm{z}_t = z \mid \mathrm{x}_t = x, \mathrm{a}_{t-1} = a]}$ do not depend on t

Probability of observing z in x given a

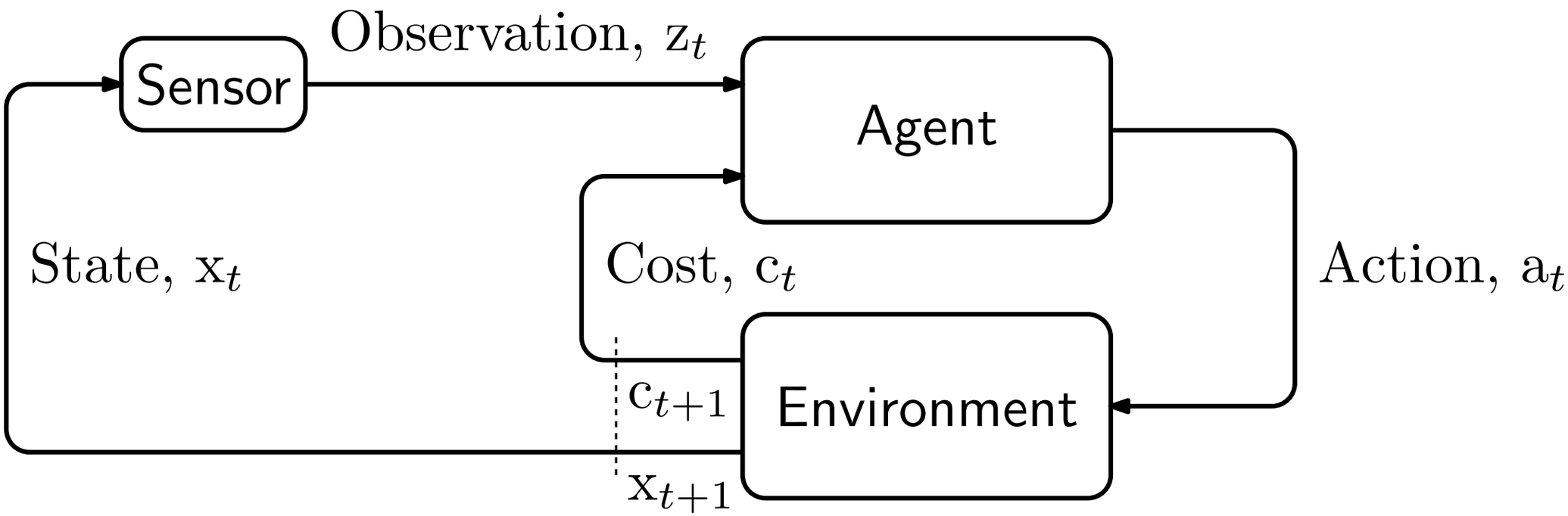
- For each action $a \in \mathcal{A}$, we store the observation probabilities in a matrix

$$[\mathbf{O}_a]_{xz} = \mathbb{P}\left[\mathrm{z}_t = z \mid \mathrm{x}_t = x, \mathrm{a}_{t-1} = a\right]$$

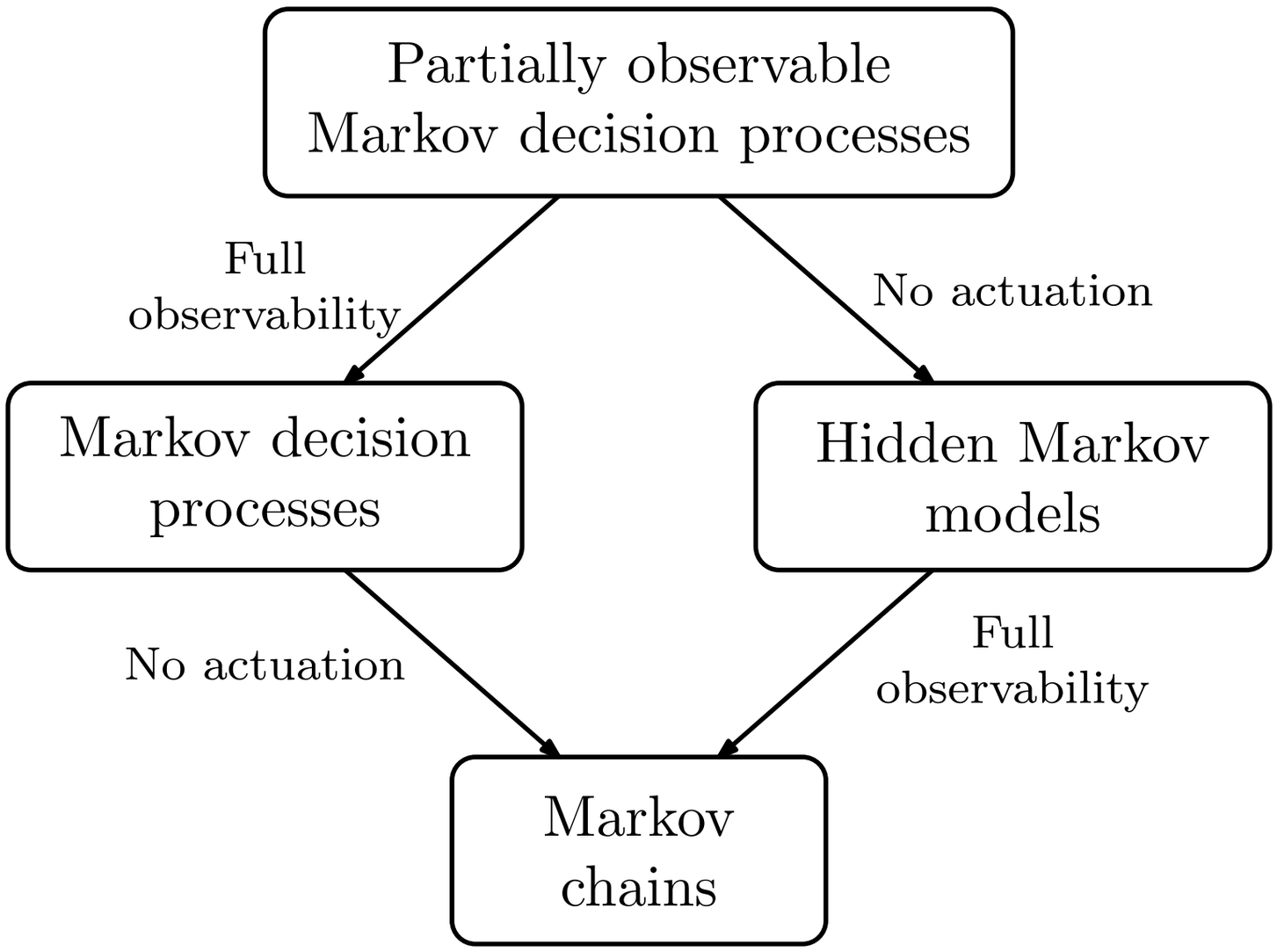
Partially observable MDP

- Described by:
  - State space, $\mathcal{X}$
  - Action space, $\mathcal{A}$
  - Observation space, $\mathcal{Z}$
  - Transition probabilities, $\{\mathbf{P}_a, a \in \mathcal{A}\}$
  - Observation probabilities, $\{\mathbf{O}_a, a \in \mathcal{A}\}$
  - Immediate cost function, $\mathbf{c}$

Partially observable MDP



An overview


Partially observable
Markov decision processes
Full
observability
No actuation
Markov decision
processes
Hidden Markov
models
No actuation
Full
observability
Markov
chains

TÉCNICO LISBOA

Decisions with POMDPs

Policy

- A **policy** is a (maybe random) "rule" that tells an agent/decision-maker what actions to choose in each step

History

- The **history** at time step t...
  - ... is a random variable, h_t
  - ... contains all that the agent **saw** up to time step t:

$$\mathrm{h}_t = \{\mathrm{z}_0, \mathrm{a}_0, \mathrm{z}_1, \mathrm{a}_1, \ldots, \mathrm{a}_{t-1}, \mathrm{z}_t\}$$

Observations,
not states

- The set of t-length histories (histories up to time t) is denoted as $\mathcal{H}_t$

Policies

- A **policy** is a (random?) mapping π from **histories** to **actions**
- A **policy** is a mapping $\pi : \mathcal{H} \to \Delta(\mathcal{A})$

Distributions over actions

- A policy chooses each action $a \in \mathcal{A}$ with a probability that depends on the history ht:

$$\pi(a \mid h_t) = \mathbb{P}[\mathrm{a}_t = a \mid \mathrm{h}_t = h_t]$$

- In general, policies are **stochastic** (random) and **history-dependent**

Types of policies

- A policy is **memoryless**...
  - ... if the distribution over actions given the history **depends only on the last observation** (and t)
    - If $h_t = \{z_0, a_0, \ldots, a_{t-1}, z_t\}$, then

$$\pi(a \mid h_t) = \mathbb{P}[\mathrm{a}_t = a \mid \mathrm{h}_t = h_t] = \mathbb{P}[\mathrm{a}_t = a \mid \mathrm{z}_t = z_t]$$

Depends only on z_t
(and eventually t)

The ideal case

- The policy that we are looking for is (ideally):

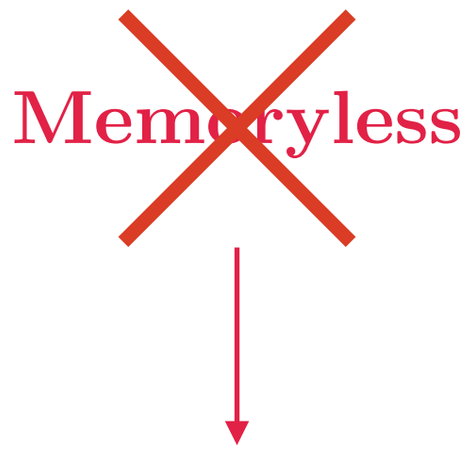
Memoryless

~~**Deterministic**~~

~~**Stationary**~~

The ideal case

- The policy that we are looking for is (ideally):



We are looking for a policy that depends on the whole history!

Discounted cost-to-go

- Assumptions:
  - The agent lives forever (we don't know number of decisions)
  - There is an inflation rate (costs in the future are not as bad as costs now)
  - Agent wants to pay as little as possible

Discounted cost-to-go

- Discounted cost-to-go:

$$DC \stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathrm{c}_t\right]$$

Cost-to-go function

- Cost-to-go function:
  - Fix a policy π
  - Deploy the agent according to some initial distribution $\boldsymbol{b}_0$
  - Let the agent go
  - Keep track of all costs to pay

Cost-to-go function

- How much will the agent pay (in average)?

$$\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t c_t\right]$$

Discounted cost-to-go

Expectation (the "in average")

Cost-to-go function

- How much will the agent pay (in average)?
  - Depends on the **initial distribution**

$$\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t \mathrm{c}_t \mid \mu_0 = \boldsymbol{b}_0\right]$$

Initial distribution

Cost-to-go function

- How much will the agent pay (in average)?
  - Depends on the initial state x
  - Depends on the policy π

$$\mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}\mathrm{c}_{t} \mid \mu_0 = \boldsymbol{b}_0\right]$$

Policy

Cost-to-go function

- How much will the agent pay (in average)?
  - Depends on the initial state x
  - Depends on the policy π

$$J^{\pi}(\boldsymbol{b}_0) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \mathrm{c}_t \mid \mu_0 = \boldsymbol{b}_0\right]$$

Cost-to-go
given distribution $\boldsymbol{b}_0$
for policy π

Cost-to-go function

- J^π maps **distributions** over $\mathcal{X}$ to real numbers (the discounted cost-to-go from that distribution)
- J^π is the **cost-to-go function** associated with policy π

Examples

The tiger problem

The tiger problem

- You are a prisoner, trying to escape a dungeon
- At a point in your escape, you face two doors

The tiger problem

- Behind one of the doors (you don't know which) lies your freedom

The tiger problem

- Behind the other door (you don't know which) lies a fearsome tiger

The tiger problem

- You can try to open one of the doors or listen behind the doors
  - When you listen, you hear the tiger behind the correct door with 85% probability
  - You waste time and may be caught
- When you open a door, you either go free or die

The tiger problem

- You are cursed to keep repeating this forever...

The POMDP model

- Can you model this problem as a POMDP?

States

- What are the states?
  - Tiger on the left
  - Tiger on the right

Actions

- What are the actions?
  - Open left
  - Open right
  - Listen

Observations

- What are the observations?
  - Tiger left
  - Tiger right
  - Nothing

Observations

- What are the observations?
  - Tiger left
  - Tiger right
  - ~~Nothing~~

Transition probabilities

- What are the transition probabilities?
  - Depend on the action
  - When listening, position of the tiger doesn't change

$$\mathbf{P}_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Transition probabilities

- What are the transition probabilities?
  - Depend on the action
  - When opening a door, you either die or go free
  - The world "resets" (it's the curse!...)

$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

Observation probabilities

- What are the observation probabilities?
  - Depend on the action
  - When listening, you hear the tiger in the correct position with probability 0.85

$$\mathbf{O}_L = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix}$$

Observation probabilities

- What are the observation probabilities?
  - Depend on the action
  - When opening a door, you either die or go free
  - The world "resets", but you hear nothing (model it as a random observation)

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

Cost

- What is the cost function?
  - Maximum cost for dying
  - Minimum cost for going free
  - **Depends on the action!**

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0.1 \\ 0 & 1 & 0.1 \end{bmatrix}$$

Small cost for listening

But how do we select actions?

Challenges

- Deterministic memoryless policies are not good enough
- Optimal policy may need to keep track of the history...

An old trick

- At time t = 0, you don't know where the tiger is
- You execute "Listen" 2 times
- You observe "Right", "Right"
- How can you use this information?

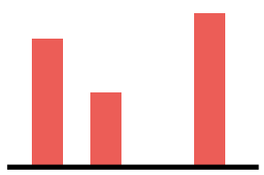
$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_{L} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

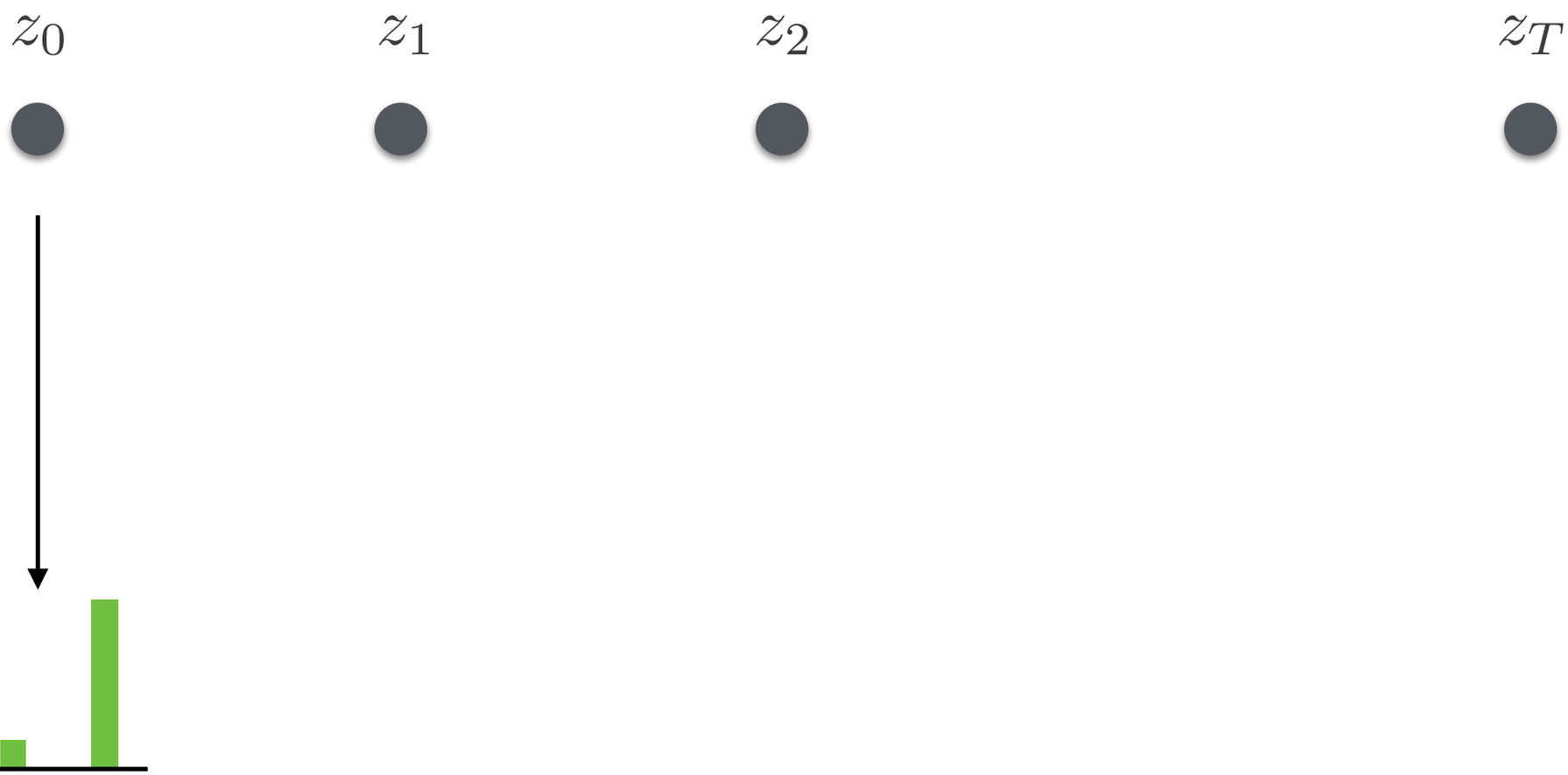
$$\mathbf{O}_{L} = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix}$$

An old trick



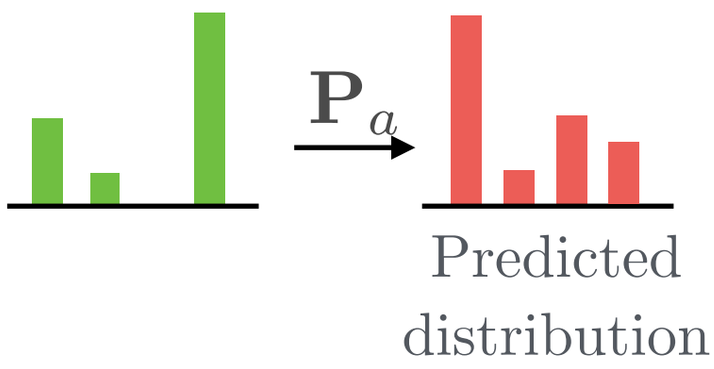
Initial distribution

An old trick

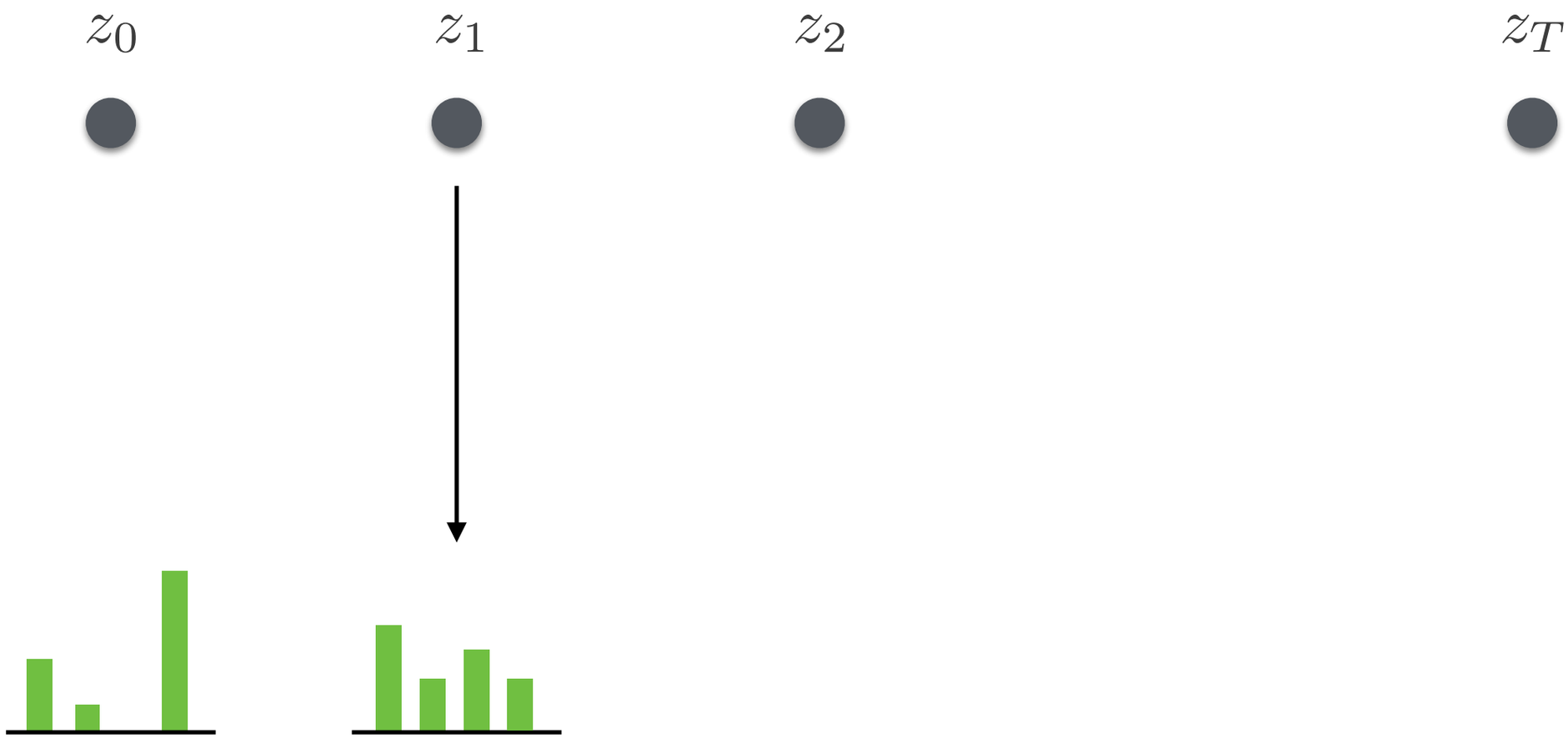


"Improved" distribution

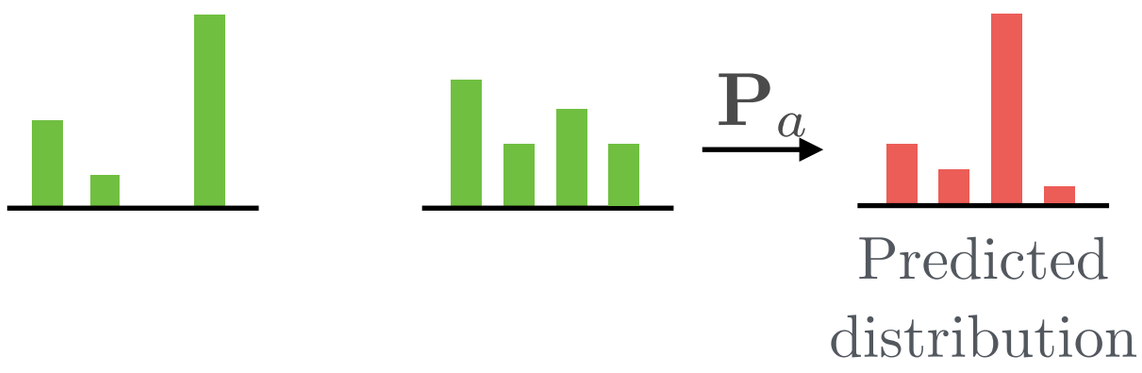
An old trick



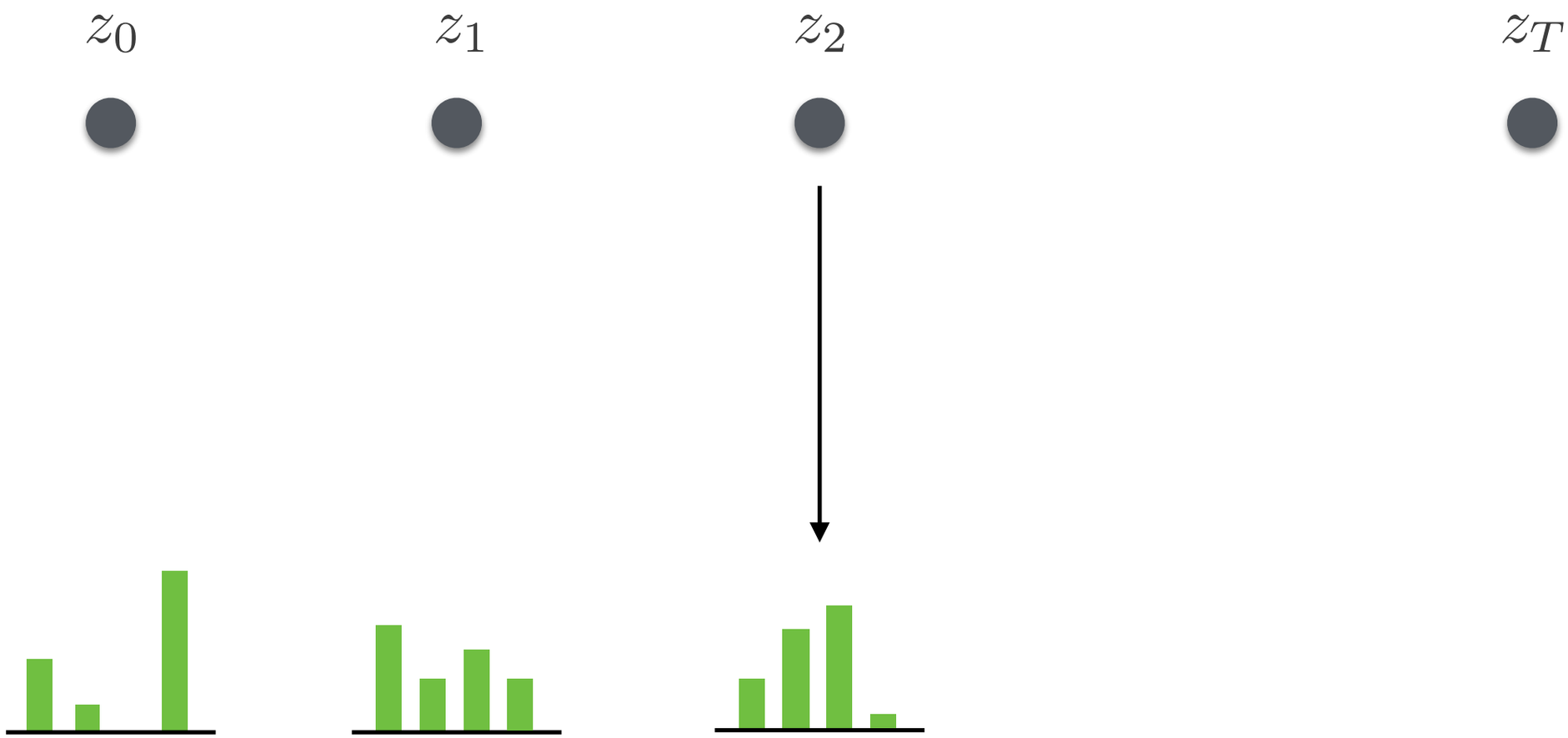
An old trick



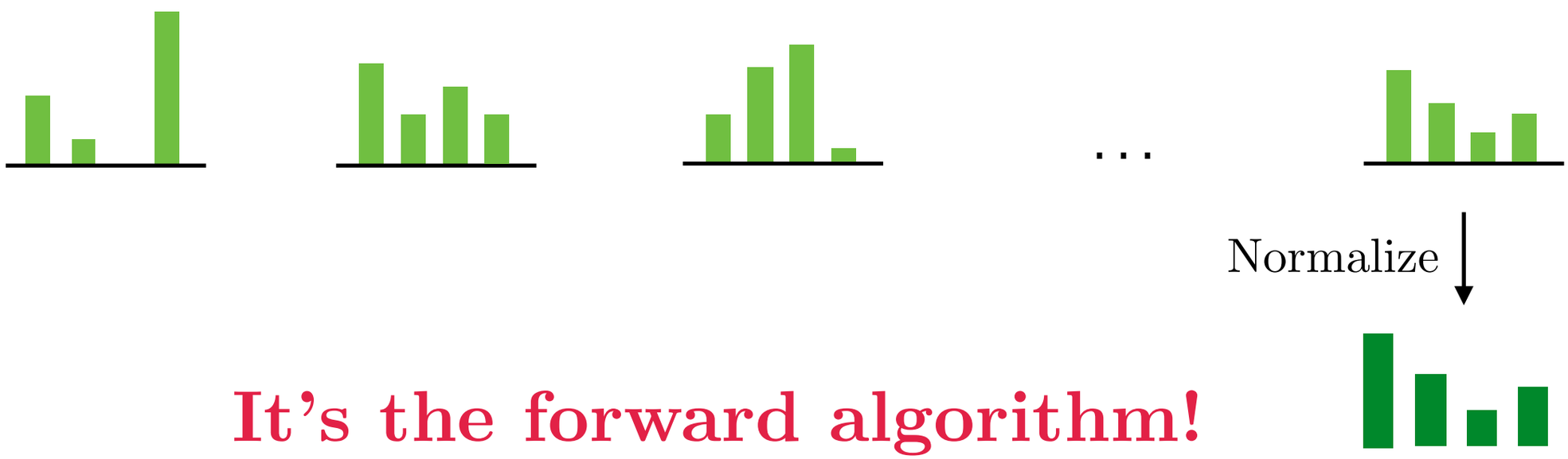
An old trick



An old trick



An old trick



It's the forward algorithm!

Let's do this:

- Initial distribution:

$$\boldsymbol{\alpha}_0 = \boldsymbol{\mu}_0 = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{O}_L = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix}$$

Let's do this:

- Take action "Listen":

$$\hat{\boldsymbol{\alpha}}_1 = \boldsymbol{\alpha}_0^\top \mathbf{P}_L$$

$$= \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{O}_L = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix}$$

Let's do this:

- Consider observation "R":

$$\boldsymbol{\alpha}_1^\top = \hat{\boldsymbol{\alpha}}_1 \text{diag}(\mathbf{O}_L(R \mid \cdot))$$

$$= \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0.15 & 0 \\ 0 & 0.85 \end{bmatrix}$$

$$= \begin{bmatrix} 0.075 & 0.425 \end{bmatrix}$$

$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{O}_L = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix}$$

Let's do this:

- Again, action "Listen":

$$\hat{\boldsymbol{\alpha}}_2 = \boldsymbol{\alpha}_1^\top \mathbf{P}_L$$

$$= \begin{bmatrix} 0.075 & 0.425 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.075 & 0.425 \end{bmatrix}$$

$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{O}_L = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix}$$

Let's do this:

- ... and observation "R":

$$\boldsymbol{\alpha}_2^\top = \hat{\boldsymbol{\alpha}}_2 \text{diag}(\mathbf{O}_L(R \mid \cdot))$$

$$= \begin{bmatrix} 0.075 & 0.425 \end{bmatrix} \begin{bmatrix} 0.15 & 0 \\ 0 & 0.85 \end{bmatrix}$$

$$= \begin{bmatrix} 0.011 & 0.361 \end{bmatrix}$$

$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{O}_L = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix}$$

Let's do this:

- Finally, normalize:

$$\boldsymbol{\mu}_{2|0:2} = \frac{\boldsymbol{\alpha}_2^\top}{\|\boldsymbol{\alpha}_2\|_1}$$

$$= \frac{1}{0.373}\begin{bmatrix}0.011 & 0.361\end{bmatrix}$$

$$= \begin{bmatrix}0.03 & 0.97\end{bmatrix}$$

$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix}0.5 & 0.5\\ 0.5 & 0.5\end{bmatrix}$$

$$\mathbf{P}_L = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix}0.5 & 0.5\\ 0.5 & 0.5\end{bmatrix}$$

$$\mathbf{O}_L = \begin{bmatrix}0.85 & 0.15\\ 0.15 & 0.85\end{bmatrix}$$

One extra observation?

- What if we make one extra observation?
  - This time, "L"?

Step

$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_{L} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{O}_{L} = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix}$$

One extra observation?

- What if we make one extra observation?
  - This time, "L"?

$\boldsymbol{\mu}_{2|0:2} \rightarrow \boldsymbol{\mu}_{2|0:2}\mathbf{P}_L$

↓

Obs.

$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{O}_L = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix}$$

One extra observation?

- What if we make one extra observation?
  - This time, "L"?

$$\boldsymbol{\mu}_{2|0:2} \rightarrow \boldsymbol{\mu}_{2|0:2}\mathbf{P}_L$$

$$\rightarrow \boldsymbol{\mu}_{2|0:2}\mathbf{P}_L \text{diag}(\mathbf{O}_L(L \mid \cdot))$$

↓

Norm.

$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{O}_L = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix}$$

One extra observation?

- What if we make one extra observation?
  - This time, "L"?

$$\boldsymbol{\mu}_{2|0:2} \rightarrow \boldsymbol{\mu}_{2|0:2}\mathbf{P}_L$$

$$\rightarrow \boldsymbol{\mu}_{2|0:2}\mathbf{P}_L \text{diag}(\mathbf{O}_L(L \mid \cdot))$$

$$\rightarrow \frac{\boldsymbol{\mu}_{2|0:2}\mathbf{P}_L \text{diag}(\mathbf{O}_L(L \mid \cdot))}{\|\boldsymbol{\mu}_{2|0:2}\mathbf{P}_L \text{diag}(\mathbf{O}_L(L \mid \cdot))\|_1}$$

$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{O}_L = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix}$$

One extra observation?

- What if we make one extra observation?
  - This time, "L"?

$$\boldsymbol{\mu}_{3|0:3} = \begin{bmatrix} 0.15 & 0.85 \end{bmatrix}$$

Probability that tiger is on the right

$$\mathbf{P}_{OL} = \mathbf{P}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{P}_{L} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{O}_{OL} = \mathbf{O}_{OR} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{O}_{L} = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix}$$

The belief

- We call the distribution $\boldsymbol{\mu}_{t|0:t}$ the **belief** at time t
  - We will denote it as $\boldsymbol{b}_t$
  - $\boldsymbol{b}_t$ is a distribution over $\mathcal{X}$
  - $\boldsymbol{b}_t(x)$ is the agent's belief that $\mathrm{x}_t = x$, given all the history, i.e.,

$$\boldsymbol{b}_t(x) = \mathbb{P}\left[\mathrm{x}_t = x \mid \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}, \mathbf{a}_{0:t-1} = \boldsymbol{a}_{0:t-1}\right]$$

The belief

- We can update the belief using the previous equation
  - ... after executing action a...
  - ... after making observation z...

$$\boldsymbol{b}_{t+1} = \frac{\boldsymbol{b}_t \mathbf{P}_a \text{diag}(\mathbf{O}_a(z \mid \cdot))}{\|\boldsymbol{b}_t \mathbf{P}_a \text{diag}(\mathbf{O}_a(z \mid \cdot))\|_1}$$

or, component-wise, ...

$$\boldsymbol{b}_{t+1}(x') = \frac{\sum_{x \in \mathcal{X}} \boldsymbol{b}_t(x) \mathbf{P}_a(x' \mid x) \mathbf{O}_a(z \mid x')}{\sum_{x,x'' \in \mathcal{X}} \boldsymbol{b}_t(x) \mathbf{P}_a(x'' \mid x) \mathbf{O}_a(z \mid x'')}$$

Belief update

The belief

- We can update the belief using the previous equation
  - ... after executing action a...
  - ... after making observation z...

$$\boldsymbol{b}_{t+1} = \frac{\boldsymbol{b}_t \mathbf{P}_a \text{diag}(\mathbf{O}_a(z \mid \cdot))}{\|\boldsymbol{b}_t \mathbf{P}_a \text{diag}(\mathbf{O}_a(z \mid \cdot))\|_1}$$

or, component-wise, ...

$$\boldsymbol{b}_{t+1}(x') = \frac{\sum_{x \in \mathcal{X}} \boldsymbol{b}_t(x) \mathbf{P}_a(x' \mid x) \mathbf{O}_a(z \mid x')}{\sum_{x,x'' \in \mathcal{X}} \boldsymbol{b}_t(x) \mathbf{P}_a(x'' \mid x) \mathbf{O}_a(z \mid x'')} \quad \mathbf{B}(\boldsymbol{b}_t, z, a)$$

The belief

- The belief at time-step $t+1$ depends only on:

$$\boldsymbol{b}_{t+1}(x') = \frac{\sum_{x \in \mathcal{X}} \boldsymbol{b}_t(x) \mathbf{P}_{a_t}(x' \mid x) \mathbf{O}_{a_t}(z_{t+1} \mid x')}{\sum_{x,x'' \in \mathcal{X}} \boldsymbol{b}_t(x) \mathbf{P}_{a_t}(x'' \mid x) \mathbf{O}_{a_t}(z_{t+1} \mid x'')}$$

The belief at time step t

The action at time step t

The observation at time step $t+1$

... only quantities from time t

The belief

- The belief at time-step $t+1$ depends only on:

$$\boldsymbol{b}_{t+1}(x') = \frac{\sum_{x \in \mathcal{X}} \boldsymbol{b}_t(x) \mathbf{P}_{a_t}(x' \mid x) \mathbf{O}_{a_t}(z_{t+1} \mid x')}{\sum_{x,x'' \in \mathcal{X}} \boldsymbol{b}_t(x) \mathbf{P}_{a_t}(x'' \mid x) \mathbf{O}_{a_t}(z_{t+1} \mid x'')}$$

  - ... the belief at time t
  - ... the action at time t
  - ... the observation at time $t+1$

- The belief at time t **summarizes the history** up to time t!

The belief is Markov!

Partially observable MDP

- Described by:
  - State space, $\mathcal{X}$
  - Action space, $\mathcal{A}$
  - Observation space, $\mathcal{Z}$
  - Transition probabilities, $\{\mathbf{P}_a, a \in \mathcal{A}\}$
  - Observation probabilities, $\{\mathbf{O}_a, a \in \mathcal{A}\}$
  - Immediate cost function, $\mathbf{c}$

Belief MDP

- Described by:
  - State space, $\mathcal{B}$ — Set of all beliefs
  - Action space, $\mathcal{A}$ — Same set of actions
  - Transition probabilities $\mathbf{P}_B$ (from the belief update)
  - Immediate cost function c_B

$$c_B(\boldsymbol{b}, a) = \sum_{x \in \mathcal{X}} \boldsymbol{b}(x) c(x, a)$$

Optimality

Cost-to-go function

- **We can adapt MDP results to POMDPs through belief-MDPs**
- Optimal cost-to-go:

$$J^*(\boldsymbol{b}) = \min_{a \in \mathcal{A}} \left[c_B(\boldsymbol{b}, a) + \gamma \sum_{\boldsymbol{b}'} \mathbf{P}_B(\boldsymbol{b}' \mid \boldsymbol{b}, a) J^*(\boldsymbol{b}') \right]$$

$$c_B(\boldsymbol{b}, a) = \sum_{x \in \mathcal{X}} \boldsymbol{b}(x) c(x, a)$$

Cost-to-go function

- **We can adapt MDP results to POMDPs through belief-MDPs**
- Optimal cost-to-go:

$$J^*(\boldsymbol{b}) = \min_{a \in \mathcal{A}} \left[c_B(\boldsymbol{b}, a) + \gamma \sum_{\boldsymbol{b}'} \mathbf{P}_B(\boldsymbol{b}' \mid \boldsymbol{b}, a) J^*(\boldsymbol{b}') \right]$$

$$\mathbf{P}_B(\boldsymbol{b}' \mid \boldsymbol{b}, a) = \sum_{z \in \mathcal{Z}} \sum_{x, x' \in \mathcal{X}} \boldsymbol{b}(x) \mathbf{P}_a(x' \mid x, a) \mathbf{O}_a(z \mid x', a) \mathbb{I}[\boldsymbol{b}' = \mathbf{B}(\boldsymbol{b}, z, a)]$$

Cost-to-go function

- **We can adapt MDP results to POMDPs through belief-MDPs**
- Optimal cost-to-go:

$$J^*(\boldsymbol{b}) = \min_{a \in \mathcal{A}} \underbrace{\sum_{x \in \mathcal{X}} \boldsymbol{b}(x) \Big[c(x,a) }_{c_B(\boldsymbol{b},a)} + \gamma \sum_{x' \in \mathcal{X}} \sum_{z \in \mathcal{Z}} \mathbf{P}_a(x' \mid x) \mathbf{O}_a(z \mid x') J^*(\mathbf{B}(\boldsymbol{b}, z, a)) \Big]$$

Cost-to-go function

- **We can adapt MDP results to POMDPs through belief-MDPs**
- Optimal cost-to-go:

$$J^*(\boldsymbol{b}) = \min_{a \in \mathcal{A}} \sum_{x \in \mathcal{X}} \boldsymbol{b}(x) \left[c(x,a) + \gamma \sum_{x' \in \mathcal{X}} \sum_{z \in \mathcal{Z}} \mathbf{P}_a(x' \mid x) \mathbf{O}_a(z \mid x') J^*(\mathbf{B}(\boldsymbol{b}, z, a)) \right]$$

$$\mathbf{P}_B(\boldsymbol{b}' \mid \boldsymbol{b}, a)$$

Cost-to-go function

- **We can adapt MDP results to POMDPs through belief-MDPs**
- Optimal cost-to-go:

$$J^*(\boldsymbol{b}) = \min_{a \in \mathcal{A}} \sum_{x \in \mathcal{X}} \boldsymbol{b}(x) \left[c(x,a) + \gamma \sum_{x' \in \mathcal{X}} \sum_{z \in \mathcal{Z}} \mathbf{P}_a(x' \mid x) \mathbf{O}_a(z \mid x') J^*(\mathbf{B}(\boldsymbol{b}, z, a) \right]$$

$\mathbf{T}$

- Operator $\mathbf{T}$ transforms arbitrary Js into new Js:

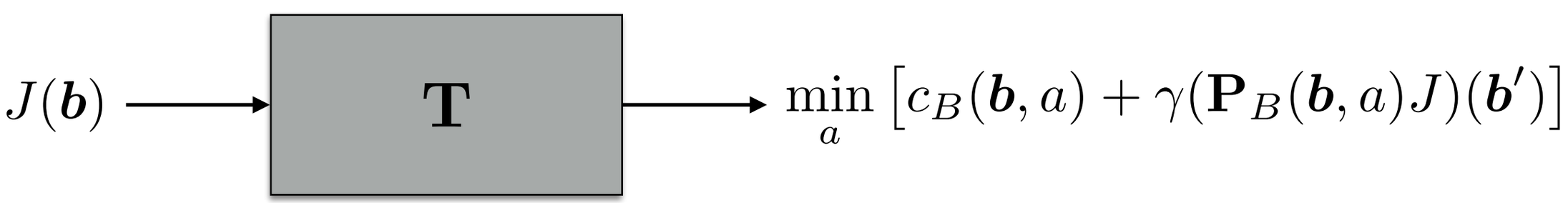
$$(\mathbf{T}J)(\boldsymbol{b}) = \min_a \left[c_B(\boldsymbol{b}, a) + \gamma (\mathbf{P}_B(\boldsymbol{b}, a) J)(\boldsymbol{b}') \right]$$

Cost-to-go function

- **We can adapt MDP results to POMDPs through belief-MDPs**
- Optimal cost-to-go:

$$J^*(\boldsymbol{b}) = \mathbf{T}\left[\min_{a\in\mathcal{A}} \sum_{x\in\mathcal{X}} \boldsymbol{b}(x)\left[c(x,a) + \gamma \sum_{x'\in\mathcal{X}} \sum_{z\in\mathcal{Z}} \mathbf{P}_a(x' \mid x)\mathbf{O}_a(z \mid x')J^*(\mathbf{B}(\boldsymbol{b},z,a)\right]\right]$$

- Operator $\mathbf{T}$ transforms arbitrary Js into new Js:



… however…

$$J^*(\boldsymbol{b}) = \min_{a \in \mathcal{A}} \left[c_B(\boldsymbol{b}, a) + \gamma \sum_{\boldsymbol{b}'} \mathbf{P}_B(\boldsymbol{b}' \mid \boldsymbol{b}, a) J^*(\boldsymbol{b}') \right]$$

How do we represent this?

Representing J*

- J^* is a function defined in $\mathbb{R}^n$ (belief)
- We cannot represent it explicitly
- Therefore,

Input: Belief MDP $\mathcal{M} = (\mathcal{B}, \mathcal{A}, \{\mathbf{P}_B(a)\}, c_B, \gamma)$

Input: ...erance ϵ

1: Initialize $k = 0$

2: Initialize $J_0(\boldsymbol{b})$ ... 0, for all $\boldsymbol{b}$

3: **repeat**

4: $\quad J_{k+1}(\boldsymbol{b})$ ... $= (\mathbf{T}\mathbf{J}_k)(\boldsymbol{b})$, for all $\boldsymbol{b}$

5: $\quad k = k + 1$

6: **until** $\|J_k$ ... $J_{k-1}\| < \epsilon$

return J_k