TÉCNICO LISBOA

Planning, Learning and Intelligent Decision Making

Lecture 9

PADInt 2024

Stochastic approximation

Challenge 1

- Amy had the following grades in the first 3 labs+homework:

HW1	HW2	HW3
19.4	14.8	17.1

- What's her current lab average?

$$(19.4 + 14.8 + 17.1) \ / \ 3 = 17.1$$

Great!

Challenge 1

- Her current lab average (after 3 labs) is 17.1
- Her fourth lab grade was 12.7
- What's her updated average?

$$(17.1 \times 3 + 12.7)/4 = 16$$

Previous average
corresponds to 3 grades

What is the average?

- The average is the value θ_N closest to **all** N samples, i.e.,

$$\theta_N = \min_{\theta} \sum_{n=1}^{N} (\theta - x_n)^2$$

Derive and equate to 0

$$2 \sum_{n=1}^{N} (\theta - x_n) = 0$$

What is the average?

- The average is the value θ_N closest to **all** N samples, i.e.,

$$\theta_N = \min_{\theta} \sum_{n=1}^{N} (\theta - x_n)^2$$

$$\sum_{n=1}^{N} (\theta - x_n) = 0$$

What is the average?

- The average is the value θ_N closest to **all** N samples, i.e.,

$$\theta_N = \min_{\theta} \sum_{n=1}^{N} (\theta - x_n)^2$$

$$\sum_{n=1}^{N} \theta = \sum_{n=1}^{N} x_n$$

What is the average?

- The average is the value θ_N closest to **all** N samples, i.e.,

$$\theta_N = \min_{\theta} \sum_{n=1}^{N} (\theta - x_n)^2$$

$$N\theta = \sum_{n=1}^{N} x_n$$

What is the average?

- The average is the value θ_N closest to **all** N samples, i.e.,

$$\theta_N = \min_{\theta} \sum_{n=1}^{N} (\theta - x_n)^2$$

$$N\theta = \sum_{n=1}^{N} x_n$$

What is the average?

- The average is the value θ_N closest to **all** N samples, i.e.,

$$\theta_N = \min_{\theta} \sum_{n=1}^{N} (\theta - x_n)^2$$

$$\theta_N = \frac{1}{N} \sum_{n=1}^{N} x_n$$

What is the average?

- The average is the value θ_N closest to **all** N samples

$$\theta_N = \frac{1}{N} \sum_{n=1}^{N} x_n$$

- From the observed samples, it's the best prediction for the "next sample"

How to recompute the average with a new sample?

New average

- If we observe a new sample x_{N+1}

$$\theta_{N+1} = \frac{1}{N+1}(\theta_N \times N + x_{N+1})$$

Previous average
corresponds to N samples

New average

- If we observe a new sample x_{N+1}

$$\theta_{N+1} = \frac{1}{N+1}(\theta_N \times N + x_{N+1})$$
$$= \frac{N}{N+1}\theta_N + \frac{1}{N+1}x_{N+1}$$

Add and subtract 1

New average

- If we observe a new sample x_{N+1}

$$\begin{aligned}\theta_{N+1} &= \frac{1}{N+1}(\theta_N \times N + x_{N+1}) \\ &= \frac{N+1-1}{N+1}\theta_N + \frac{1}{N+1}x_{N+1}\end{aligned}$$

New average

- If we observe a new sample x_{N+1}

$$\theta_{N+1} = \frac{1}{N+1}(\theta_N \times N + x_{N+1})$$
$$= \frac{N+1-1}{N+1}\theta_N + \frac{1}{N+1}x_{N+1}$$
$$= \left(1 - \frac{1}{N+1}\right)\theta_N + \frac{1}{N+1}x_{N+1}$$

Gather these two terms

New average

- If we observe a new sample x_{N+1}

$$\begin{aligned}
\theta_{N+1} &= \frac{1}{N+1}(\theta_N \times N + x_{N+1}) \\
&= \frac{N+1-1}{N+1}\theta_N + \frac{1}{N+1}x_{N+1} \\
&= \left(1 - \frac{1}{N+1}\right)\theta_N + \frac{1}{N+1}x_{N+1} \\
&= \theta_N + \frac{1}{N+1}(x_{N+1} - \theta_N)
\end{aligned}$$

New average

- If we observe a new sample x_{N+1}

$$\theta_{N+1} = \theta_N + \frac{1}{N+1}(x_{N+1} - \theta_N)$$

New estimate

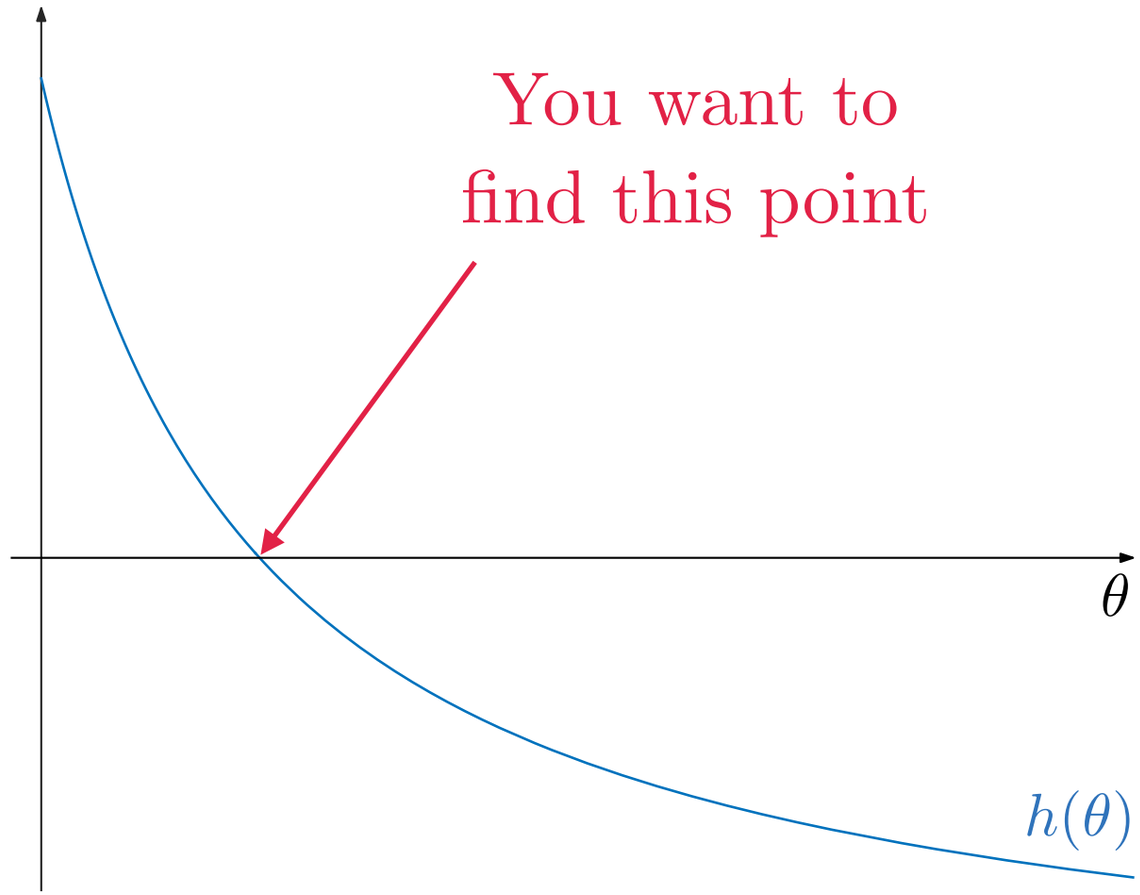
Previous estimate

Step

Error

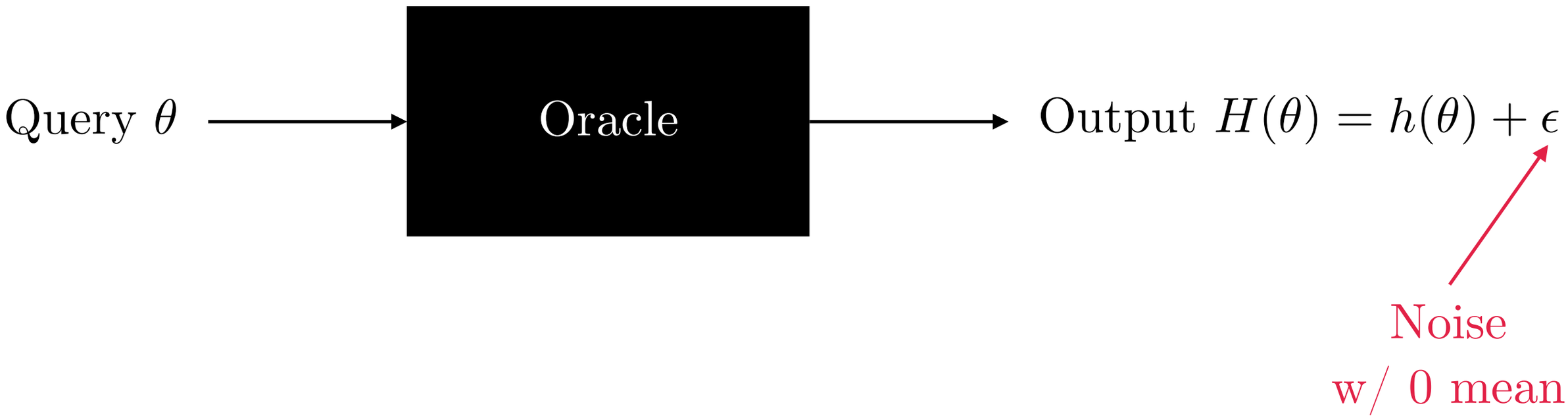
Challenge 2

- Consider the function:



Challenge 2

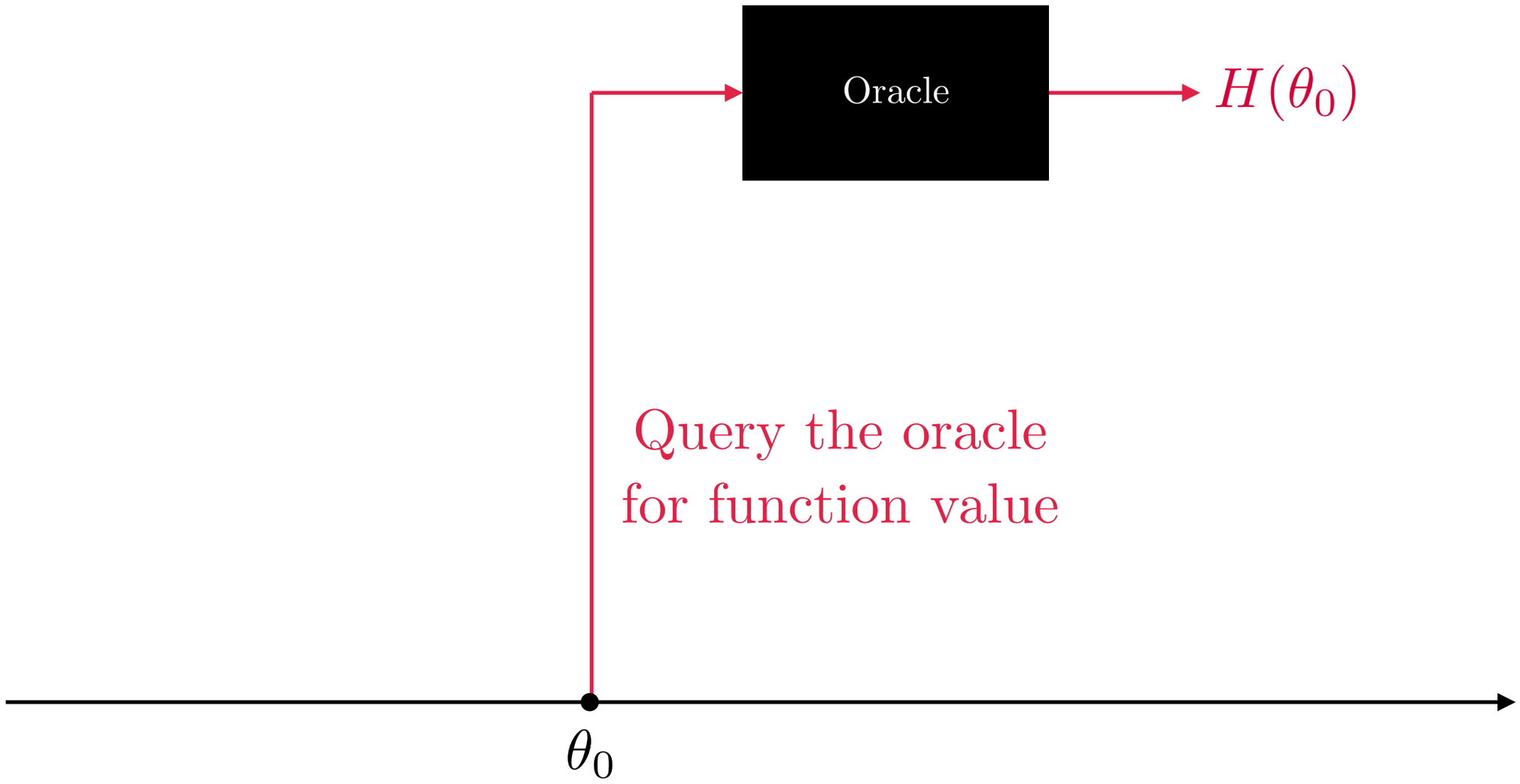
- You can query a black box:



How do you solve this?

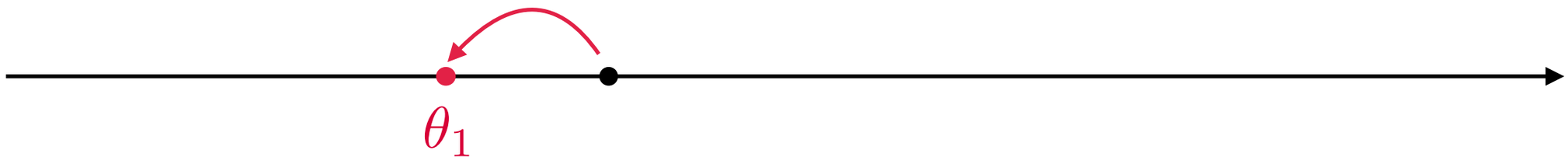
Idea

Idea



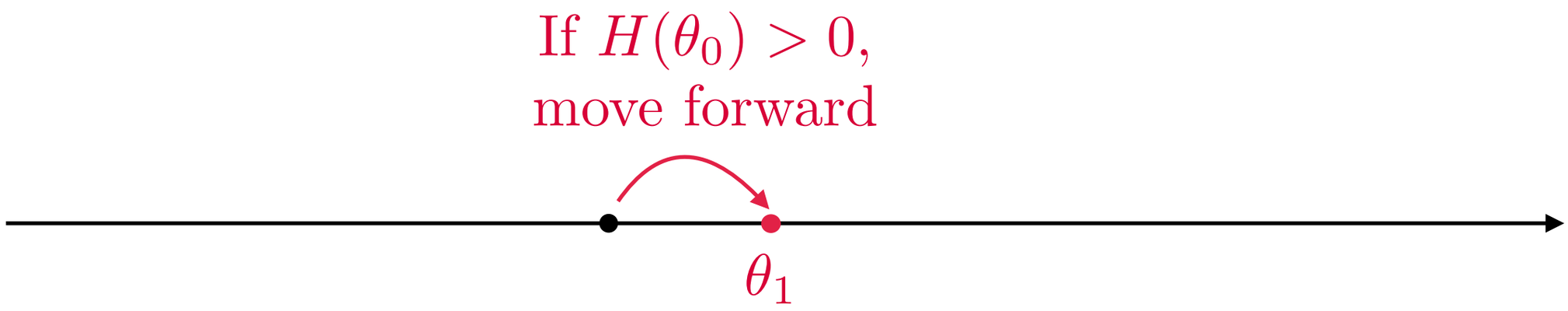
Idea

If $H(\theta_0) < 0$,
move back



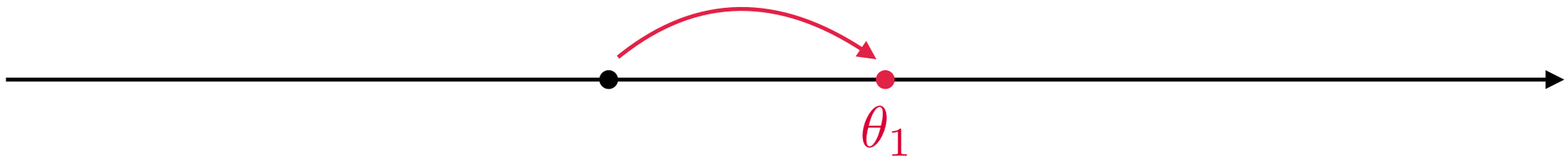
Idea

Idea



Idea

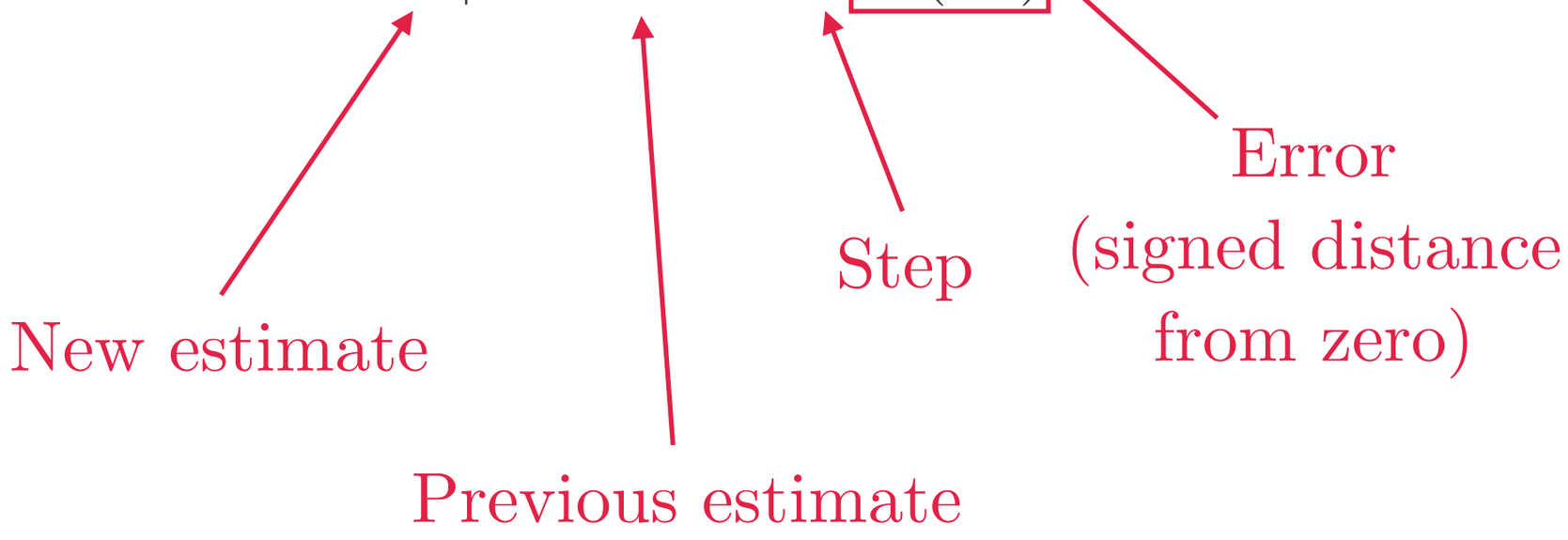
If $H(\theta_0) \gg 0$,
move **far** forward



Idea

- We thus build a sequence

$$\theta_{n+1} = \theta_n + \alpha_n H(\theta_n)$$



Stochastic approximation

- Iterative algorithms to compute the solution to the equation

$$\mathbb{E}[\mathrm{H}(\theta)] = 0$$

where H is some function that can be queried

- Take the general form

$$\theta_{n+1} = \theta_n + \alpha_n H(\theta_n)$$

$$= \theta_n + \alpha_n h(\theta_n) + \alpha_n (H(\theta_n) - h(\theta_n))$$

Value with noise

Value without noise

Zero-mean noise ϵ

Example 1. Computing a mean

- We want to compute θ so that

$$\theta = \boxed{\mathbb{E}[\mathrm{x}]}$$

Mean (unknown)

Example 1. Computing a mean

- We want to compute θ so that

$$\theta = \mathbb{E}[\mathrm{x}]$$

- Equivalently,

$$\mathbb{E}[\mathrm{x}] - \theta = 0$$

Example 1. Computing a mean

- We want to compute θ so that

$$\theta = \mathbb{E}[\mathrm{x}]$$

- Equivalently,

$$\boxed{\mathbb{E}[\mathrm{x} - \theta]} = 0$$

$$h(\theta)$$

Example 1. Computing a mean

- We want to compute θ so that

$$\theta = \mathbb{E}[\mathrm{x}]$$

- Equivalently,

$$\mathbb{E}[\mathrm{x} - \theta] = 0$$

$$\mathrm{H}(\theta)$$

Example 1. Computing a mean

- We want to compute θ so that

$$\theta = \mathbb{E}[\mathrm{x}]$$

- Equivalently,

$$\mathbb{E}[\mathrm{H}(\theta)] = 0$$

with

$$\mathrm{H}(\theta) = \mathrm{x} - \theta$$

Example 1. Computing a mean

- We have:

$$\mathrm{H}(\theta) = \mathrm{x} - \theta$$

- Using our general expression

$$\theta_{n+1} = \theta_n + \alpha_n \boxed{H(\theta_n)}$$

Sample of $\mathrm{H}(\theta_n)$

Example 1. Computing a mean

- We have:

$$\mathrm{H}(\theta) = \mathrm{x} - \theta$$

- Using our general expression

$$\theta_{n+1} = \theta_n + \alpha_n \boxed{(x_{n+1} - \theta_n)}$$

Sample of $\mathrm{H}(\theta_n)$

Example 1. Computing a mean

- We have:

$$\mathrm{H}(\theta) = \mathrm{x} - \theta$$

- Using our general expression

$$\theta_{n+1} = \theta_n + \alpha_n(x_{n+1} - \theta_n)$$

Compare with our previous expression

$$\theta_{N+1} = \theta_N + \frac{1}{N+1}(x_{N+1} - \theta_N)$$

Example 2. Computing a probability

- We want to compute θ so that

$$\theta = \boxed{\mathbb{P}[\mathrm{x} = x]}$$

Probability (unknown)

Example 2. Computing a probability

- We want to compute θ so that

$$\theta = \mathbb{P}[\mathrm{x} = x]$$

- This is not an expectation...

Example 2. Computing a probability

- A different perspective:
  - How frequently (percentage of occurrences) is $\mathrm{x} = x$?
  - Answer: $\mathbb{P}[\mathrm{x} = x]$
- In other words...

$$\mathbb{I}[\mathrm{x} = x]$$

Example 2. Computing a probability

- A different perspective:
  - How frequently (percentage of occurrences) is $\mathrm{x} = x$?
  - Answer: $\mathbb{P}[\mathrm{x} = x]$

- In other words...

$$\mathbb{P}[\mathrm{x} = x] = \mathbb{E}[\mathbb{I}[\mathrm{x} = x]]$$

$$\mathbb{E}[\mathbb{I}[\mathrm{x} = x]]$$
$$= \sum_y \mathbb{I}[\mathrm{x} = x]\mathbb{P}[\mathrm{x} = y]$$
$$= \mathbb{P}[\mathrm{x} = x]$$

Example 2. Computing a probability

- We want to compute θ so that

$$\theta = \mathbb{E}[\mathbb{I}[\mathrm{x} = x]]$$

Expectation!

- Equivalently,

$$\mathbb{E}[\mathbb{I}[\mathrm{x} = x]] - \theta = 0$$

Example 2. Computing a probability

- We want to compute θ so that

$$\theta = \mathbb{E}[\mathbb{I}[\mathrm{x} = x]]$$

- Equivalently,

$$\mathbb{E}[\mathbb{I}[\mathrm{x} = x] - \theta] = 0$$

$h(\theta)$

Example 2. Computing a probability

- We want to compute θ so that

$$\theta = \mathbb{E}[\mathbb{I}[\mathrm{x} = x]]$$

- Equivalently,

$$\mathbb{E}[\mathbb{I}[\mathrm{x} = x] - \theta] = 0$$

$\mathrm{H}(\theta)$

Example 2. Computing a probability

- We want to compute θ so that

$$\theta = \mathbb{E}[\mathbb{I}[\mathrm{x} = x]]$$

- Equivalently,

$$\mathbb{E}[\mathrm{H}(\theta)] = 0$$

with

$$\mathrm{H}(\theta) = \mathbb{I}[\mathrm{x} = x] - \theta$$

Example 2. Computing a probability

- We have:

$$\mathrm{H}(\theta) = \mathbb{I}[\mathrm{x} = x] - \theta$$

- Using our general expression

$$\theta_{n+1} = \theta_n + \alpha_n \boxed{H(\theta_n)}$$

Sample of $\mathrm{H}(\theta_n)$

Example 2. Computing a probability

- We have:

$$\mathrm{H}(\theta) = \mathbb{I}[\mathrm{x} = x] - \theta$$

- Using our general expression

$$\theta_{n+1} = \theta_n + \alpha_n \boxed{(\mathbb{I}[x_{n+1} = x] - \theta_n)}$$

Sample of $\mathrm{H}(\theta_n)$

Example 3. Computing a FP

- A fixed point (FP) of a function f is the solution to

$$\theta = f(\theta)$$

or, equivalently,

$$f(\theta) - \theta = 0$$

Example 3. Computing a FP

- We want to compute θ so that

$$\theta = \boxed{\mathbb{E}[F(\mathrm{x}, \theta)]}$$

$$\downarrow$$

$$f(\theta)$$

$$\downarrow$$

Solving an
FP equation

Example 3. Computing a FP

- We want to compute θ so that

$$\theta = \mathbb{E}[F(\mathrm{x}, \theta)]$$

Expectation depends on θ!

- Equivalently,

$$\mathbb{E}[F(\mathrm{x}, \theta) - \theta] = 0$$

$h(\theta)$

Example 3. Computing a FP

- We want to compute θ so that

$$\theta = \mathbb{E}[F(\mathrm{x}, \theta)]$$

- Equivalently,

$$\mathbb{E}[\boxed{F(\mathrm{x}, \theta) - \theta}] = 0$$

$$\mathrm{H}(\theta)$$

Example 3. Computing a FP

- We want to compute θ so that

$$\theta = \mathbb{E}[F(\mathrm{x}, \theta)]$$

- Equivalently,

$$\mathbb{E}[\mathrm{H}(\theta)] = 0$$

with

$$\mathrm{H}(\theta) = F(\mathrm{x}, \theta) - \theta$$

Example 3. Computing a FP

- We have:

$$\mathrm{H}(\theta) = F(\mathrm{x}, \theta) - \theta$$

- Using our general expression

$$\theta_{n+1} = \theta_n + \alpha_n \boxed{H(\theta_n)}$$

Sample of $\mathrm{H}(\theta_n)$

Example 3. Computing a FP

- We have:

$$\mathrm{H}(\theta) = F(\mathrm{x}, \theta) - \theta$$

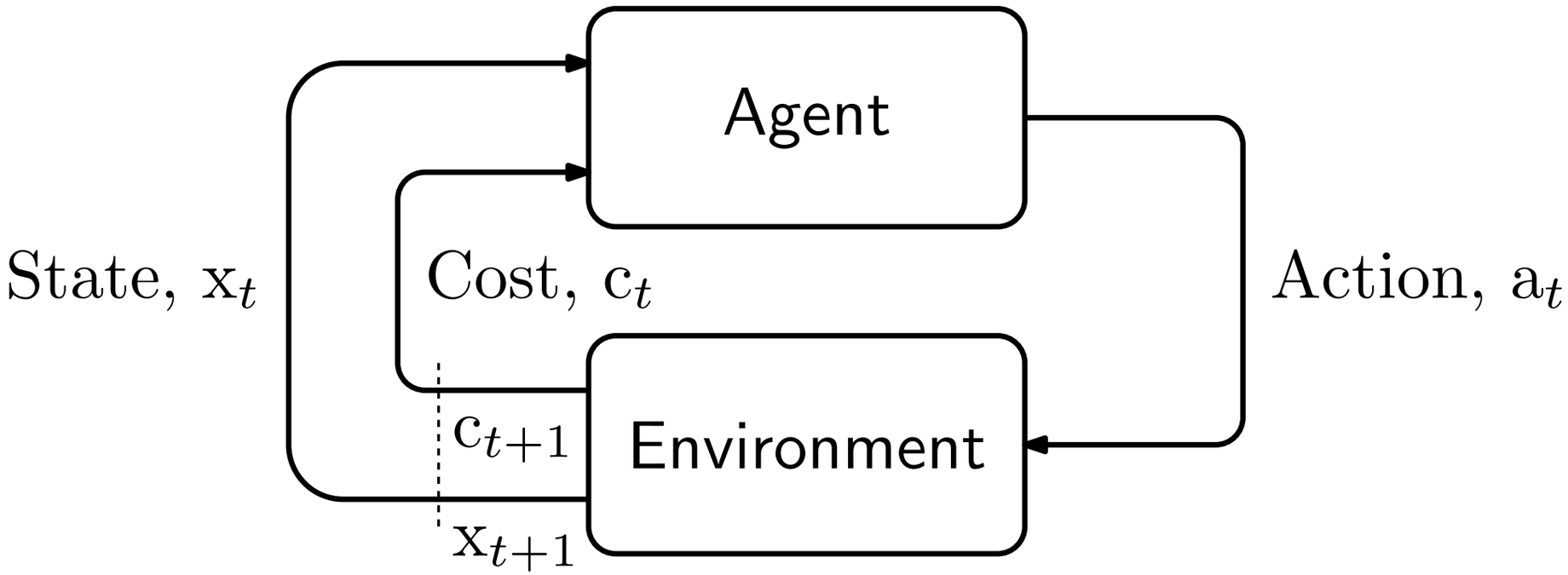
- Using our general expression

$$\theta_{n+1} = \theta_n + \alpha_n \boxed{(F(x_{n+1}, \theta_n) - \theta_n)}$$

Sample of $\mathrm{H}(\theta_n)$

Stochastic approximation

Markov decision process



Computing J^π

- We have that

$$J^\pi(x) = c_\pi(x) + \gamma \sum_{y \in \mathcal{X}} \mathbf{P}_\pi(y \mid x) J^\pi(y)$$

which is equivalent to

$$J^\pi(x) = \mathbb{E}_{\mathrm{a} \sim \pi(x)} \left[c(x, \mathrm{a}) + \gamma \sum_{y \in \mathcal{X}} \mathbf{P}_{\mathrm{a}}(y \mid x) J^\pi(y) \right]$$

We must know the cost

We must know the transition probabilities

Computing Q^*

- We have that

$$Q^*(x,a) = \boxed{c(x,a)} + \gamma \sum_{x' \in \mathcal{X}} \boxed{\mathbf{P}_a(x' \mid x)} \min_{a'} Q^*(x',a')$$

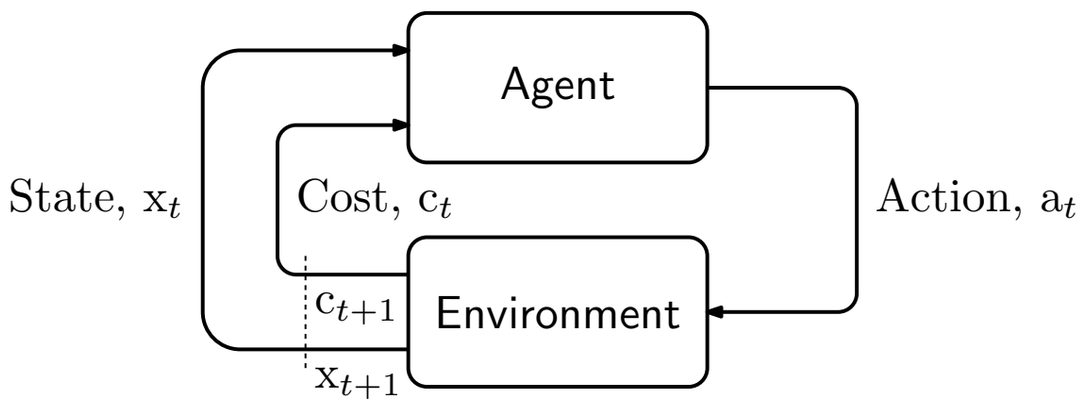
We must know the cost

We must know the transition probabilities

What if we don’t?

Interactive learning

- We let the agent into the environment
- At each moment, the agent observes the state x_t
- The agent then selects an action a_t
- The agent observes the resulting cost c_t
- The process repeats



Interactive learning

- At each step, the agent collects a "data point":

$$(x_t, a_t, c_t, x_{t+1})$$

- Agent must compute the optimal policy by collecting many such data points

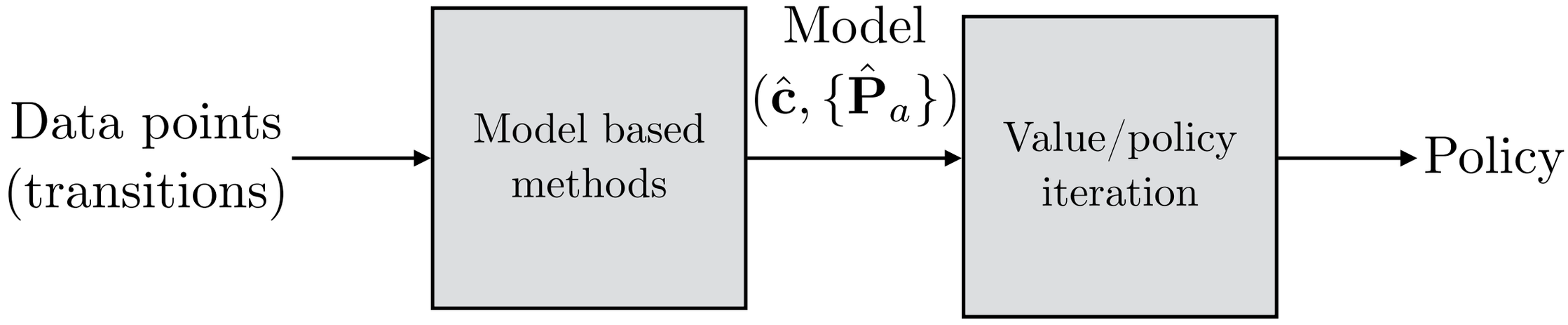
- The agent learns from "reward and punishment"

  - This form of learning is called **reinforcement learning**

How?

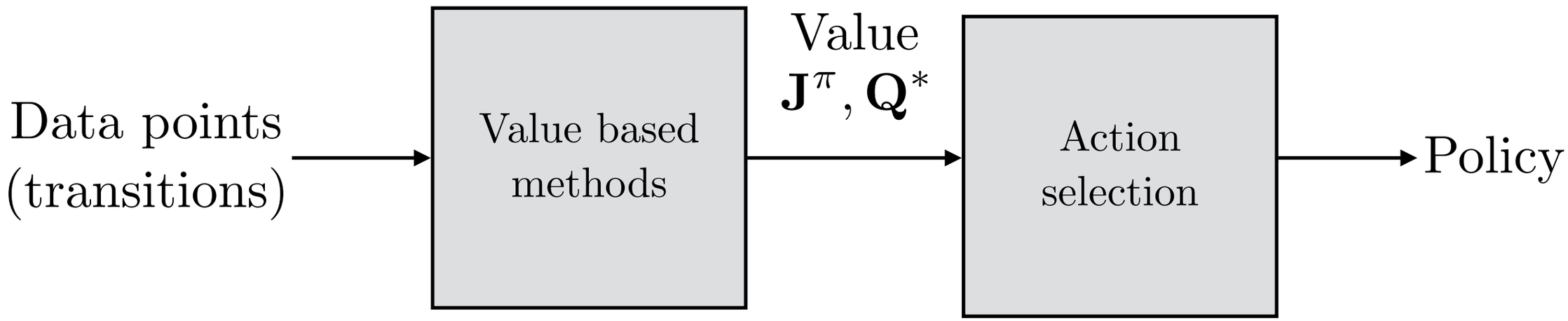
Three families of approaches

- Model-based methods:



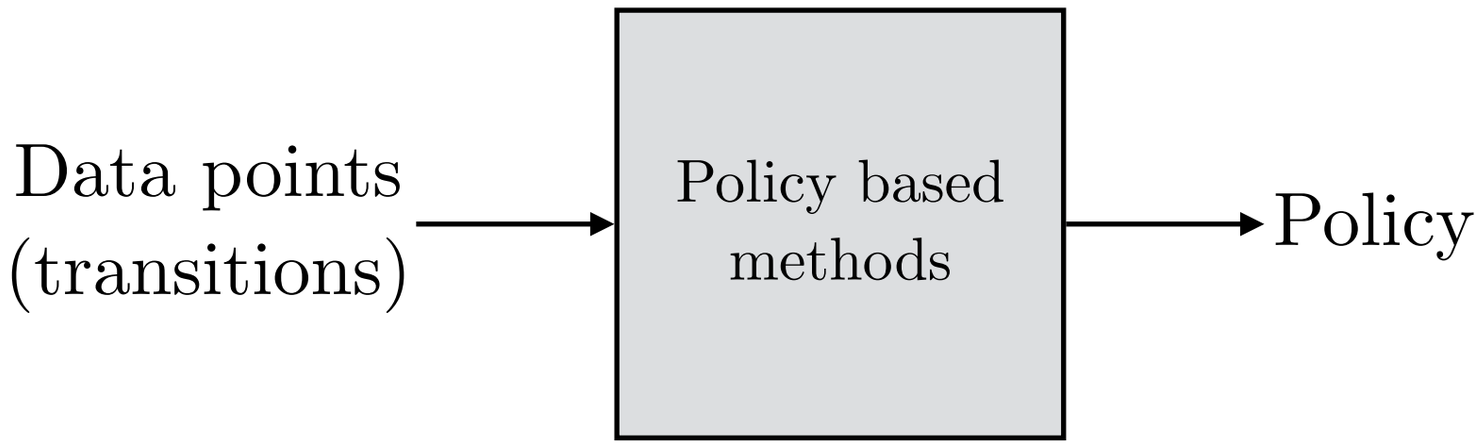
Three families of approaches

- Value-based methods:



Three families of approaches

- Policy-based methods:



Model-based methods

Estimating c

- We can estimate the cost c by keeping track of the observed costs at different states and actions
- At each step t, we just set

$$\hat{c}(x_t, a_t) = c_t$$

Use ^ to indicate that this is learned

What if there is noise in the costs?

Estimating c

- We can estimate the cost c by keeping track of the observed costs at different states and actions

- At each step t, we would like to have

$$\hat{c}(x_t, a_t) = \mathbb{E}[\mathrm{c}_t]$$

This is our θ for each x and a

This is unknown

Estimating c

- We can estimate the cost c by keeping track of the observed costs at different states and actions
- At each step t, we would like to have

$$\hat{c}(x_t, a_t) = \mathbb{E}[\mathrm{c}_t]$$

$$H(\theta) = c_t - \hat{c}(x_t, a_t)$$

Estimating c

- We can estimate the cost c by keeping track of the observed costs at different states and actions
- At each step t, we just set

$$\hat{c}(x_t, a_t) = \hat{c}(x_t, a_t) + \alpha_t(c_t - \hat{c}(x_t, a_t))$$

We are just computing the mean!

Estimating P

- What about $\mathbf{P}$?
- We have that

$$\mathbf{P}(x' \mid x_t, a_t) = \mathbb{P}[\mathrm{x}_{t+1} = x' \mid \mathrm{x}_t = x_t, \mathrm{a}_t = a_t]$$

This is our θ for each x, x' and a

This is unknown

Estimating P

- What about $\mathbf{P}$?
- We have that

$$\mathbf{P}(x' \mid x_t, a_t) = \mathbb{E}[\mathbb{I}[\mathrm{x}_{t+1} = x'] \mid \mathrm{x}_t = x_t, \mathrm{a}_t = a_t]$$

$$H(\theta) = \mathbb{I}[x_{t+1} = x'] - \hat{\mathbf{P}}(x' \mid x_t, a_t)$$

Estimating P

- We can estimate the transition probabilities $\mathbf{P}$ by keeping track of the how often we transition between states
- At each step t, we just set

$$\hat{\mathbf{P}}(x' \mid x_t, a_t) = \hat{\mathbf{P}}(x' \mid x_t, a_t) + \alpha_t(\mathbb{I}[x_{t+1} = x'] - \hat{\mathbf{P}}(x' \mid x_t, a_t))$$

Use VI or PI with the model

- Once you have estimates for $\mathbf{P}$ and $\mathbf{c}$
  - You can use VI to compute

$$J^{\pi}(x) = \hat{c}_{\pi}(x) + \gamma \sum_{y \in \mathcal{X}} \hat{\mathbf{P}}_{\pi}(y \mid x) J^{\pi}(y)$$

or

$$Q^{*}(x, a) = \hat{c}(x, a) + \gamma \sum_{x' \in \mathcal{X}} \hat{\mathbf{P}}_{a}(x' \mid x) \min_{a'} Q^{*}(x', a')$$

Use VI or PI with the model

- Once you have estimates for $\mathbf{P}$ and $\mathbf{c}$
  - You can use PI to compute

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

Does this work?

- We are computing means:
  - We estimate $c(x, a)$ as a mean (for each x and a)
  - We estimate $\mathbf{P}(x' \mid x, a)$ as mean (for each x, x', and a)
- How many "data points" do we need for each x and a?
  - An infinite number!

The model-based approach described converges to the true parameters $\mathbf{P}$ *and* $\mathbf{c}$ *as long as every state and action are visited infinitely often.*

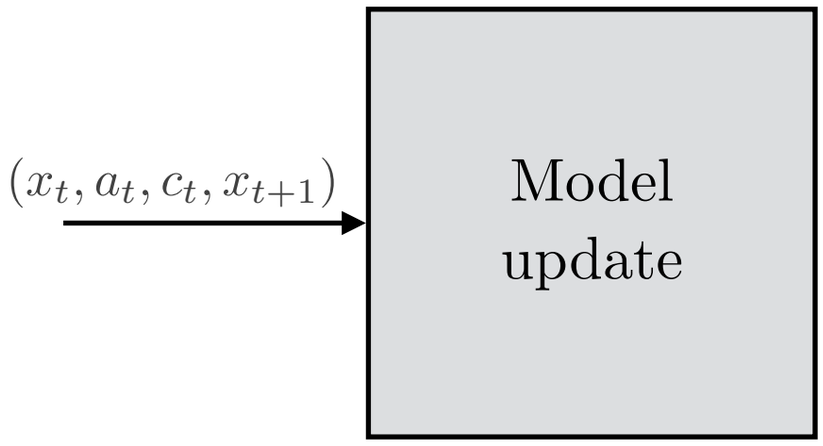
So when do we run VI?

Model based RL

- In practice, we interleave steps of **model updating** with steps of **value/policy iteration**

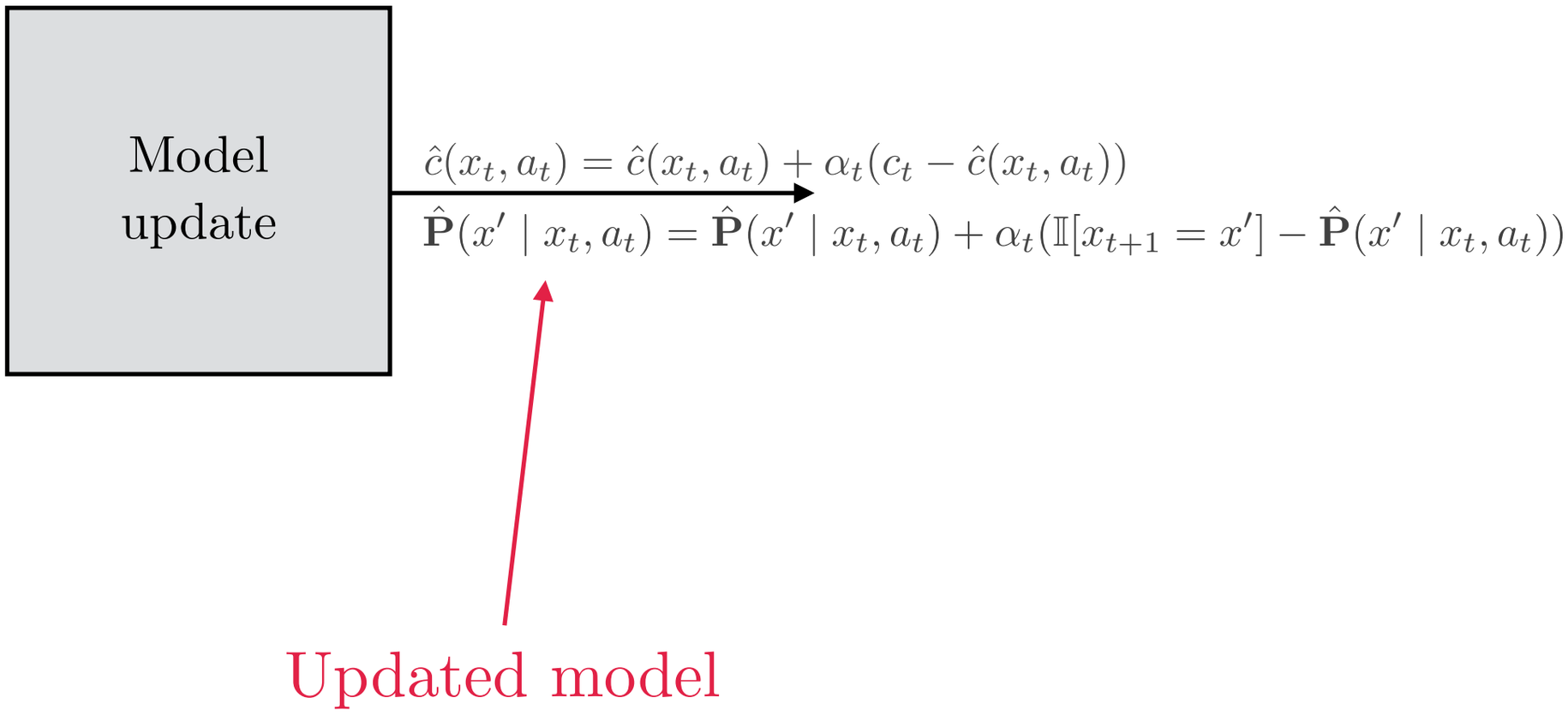
Model based RL

- In practice, we interleave steps of model learning with steps of value/policy iteration



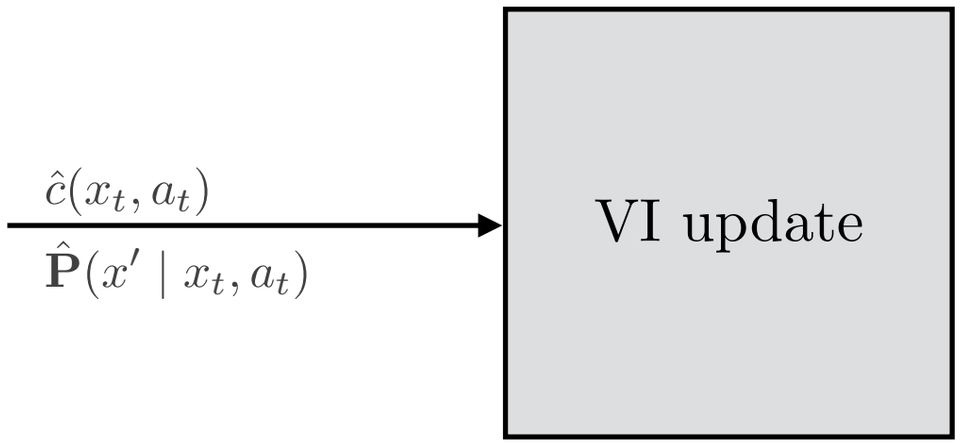
Model based RL

- In practice, we interleave steps of model learning with steps of value/policy iteration



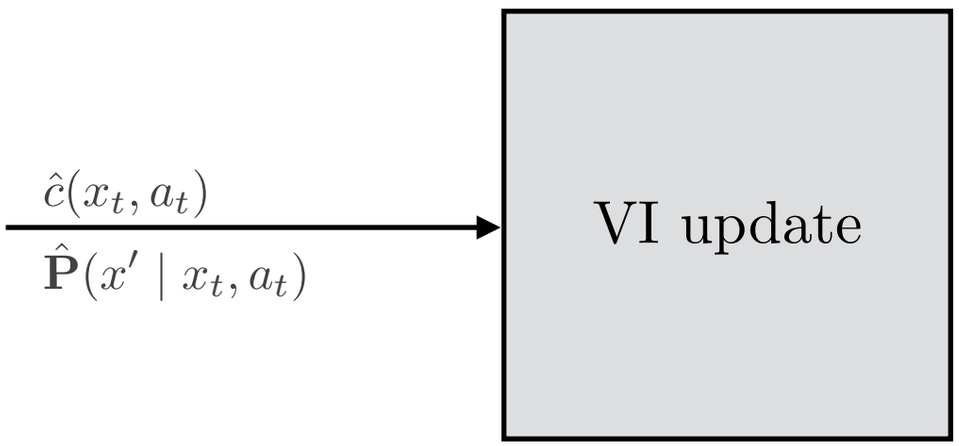
Model based RL

- In practice, we interleave steps of model learning with steps of value/policy iteration



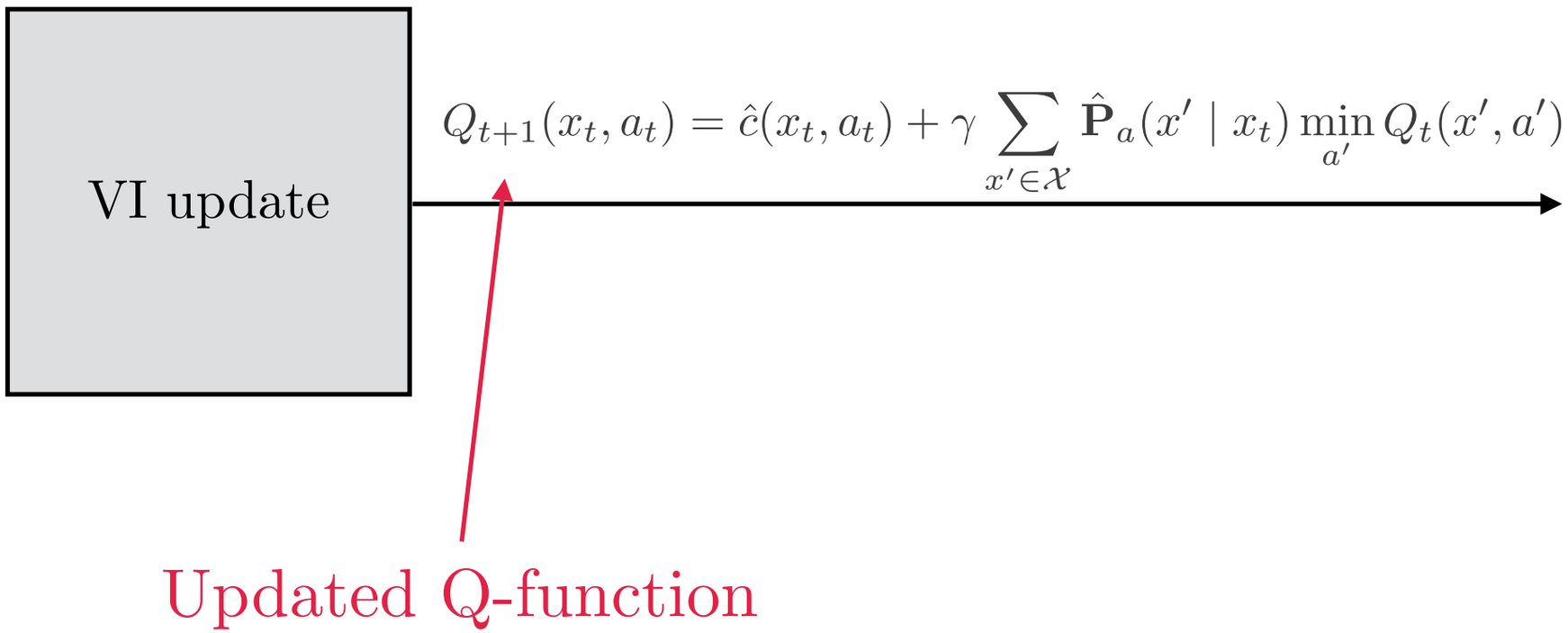
Model based RL

- In practice, we interleave steps of model learning with steps of value/policy iteration



Model based RL

- In practice, we interleave steps of model learning with steps of value/policy iteration



Summarizing…

Computing J^π

- Given a sample (x_t, c_t, x_{t+1}), where the action was selected from π
- Perform the updates

Estimating $\mathbf{c}_\pi$

$$\hat{c}(x_t) = \hat{c}(x_t) + \alpha_t(c_t - \hat{c}(x_t))$$

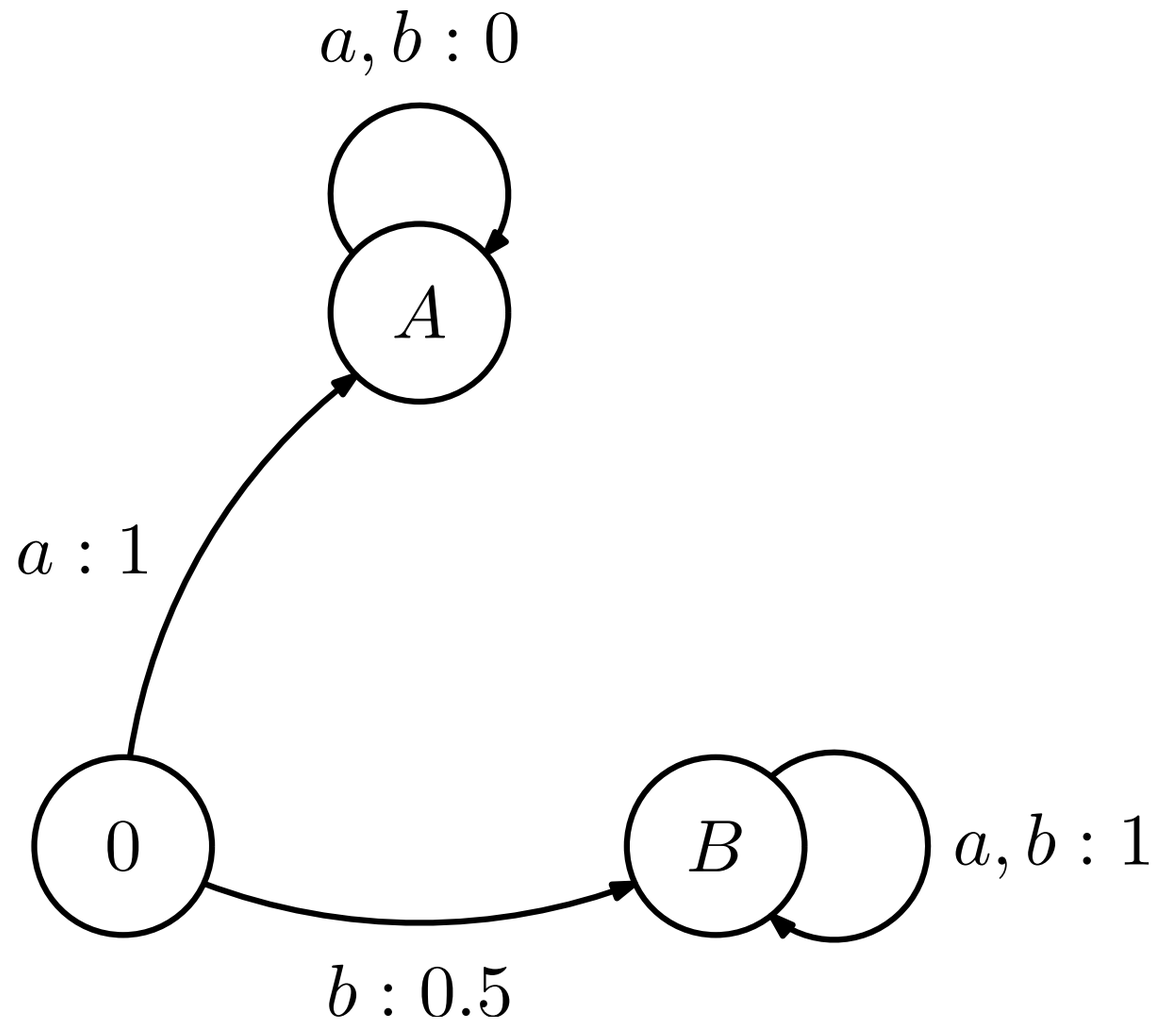
Estimating $\mathbf{P}_\pi$

$$\hat{\mathbf{P}}(x' \mid x_t) = \hat{\mathbf{P}}(x' \mid x_t) + \alpha_t(\mathbb{I}[x_{t+1} = x'] - \hat{\mathbf{P}}(x' \mid x_t))$$

$$J_{t+1}(x_t) = \hat{c}(x_t) + \gamma \sum_{x' \in \mathcal{X}} \hat{\mathbf{P}}(x' \mid x_t) J_t(x')$$

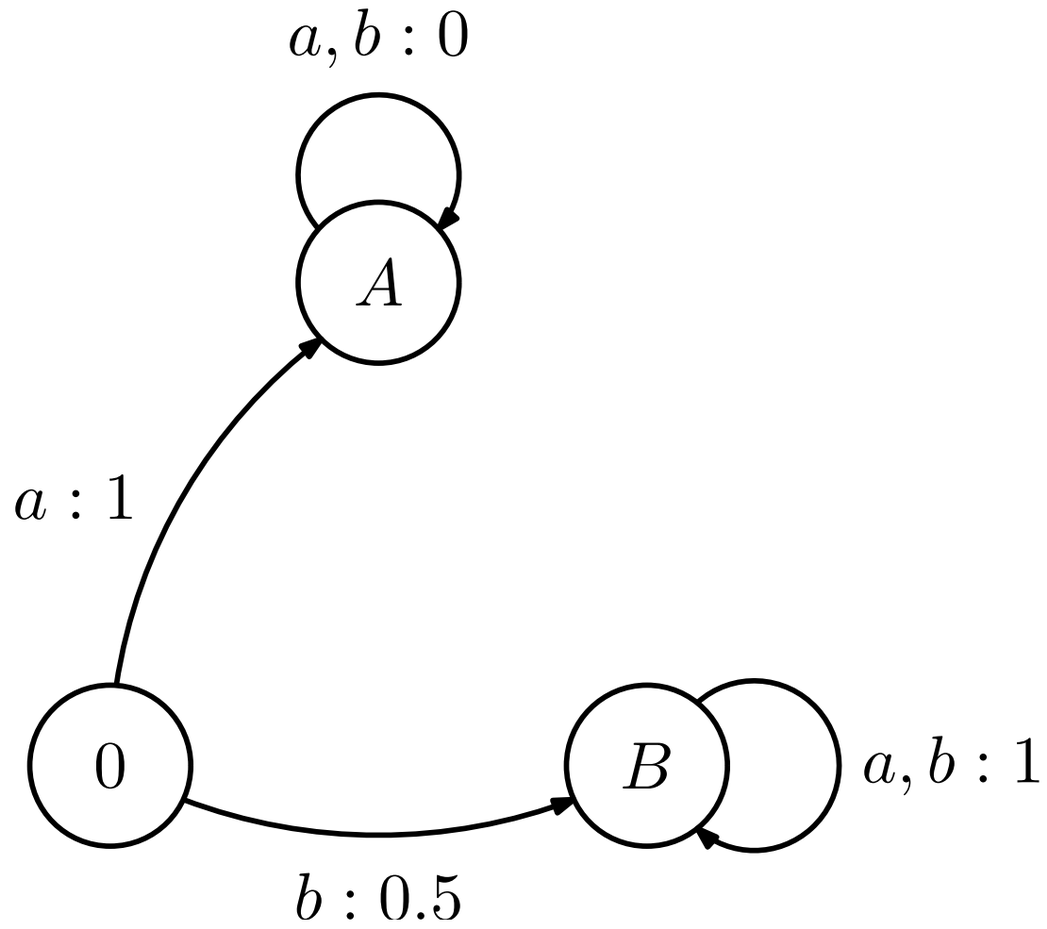
Update only affected entries

Example



Example

- Suppose that the agent does not know $\mathbf{c}$ nor $\mathbf{P}$
- The agent selects actions according to the random policy π
- We want to compute J^{π}, for $\gamma = 0.9$



Example

- We want to use the recursion

$$J^{\pi}(x) = c_{\pi}(x) + \gamma \sum_{y \in \mathcal{X}} \mathbf{P}_{\pi}(y \mid x) J^{\pi}(y)$$

We must know the cost c_{π}

We must know the transition probabilities $\mathbf{P}_{\pi}$

Example

- Suppose we observe the following transitions:

$$\tau = \{(0, 1.0, A), (A, 0.0, A), (A, 0.0, A), (A, 0.0, A)\}$$

- We initialize:

$$\hat{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \hat{\mathbf{P}} = \begin{bmatrix} 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \end{bmatrix}$$

Example

- For the first transition $(0,\ 1.0,\ A)$, using $\alpha = 0.2$, we get

$$\hat{c}(0) = 0 + \alpha \times (1 - 0) = 0.2$$

and

$$\hat{\mathbf{P}}(0 \mid 0) = 0.3 + \alpha \times (0 - 0.3) = 0.27$$
$$\hat{\mathbf{P}}(A \mid 0) = 0.3 + \alpha \times (1 - 0.3) = 0.47$$
$$\hat{\mathbf{P}}(B \mid 0) = 0.3 + \alpha \times (0 - 0.3) = 0.27$$

Example

- After updating,

$$\hat{c} = \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix} \qquad \hat{\mathbf{P}} = \begin{bmatrix} 0.27 & 0.47 & 0.27 \\ 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \end{bmatrix}$$

- We can now update J, to get

$$J(0) = \hat{c}(0) + 0.9 \times \mathbf{P}(y \mid 0) J(y) = 0.2$$

Example

- After processing the whole trajectory,

$$\hat{c} = \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix} \qquad \hat{\mathbf{P}} = \begin{bmatrix} 0.27 & 0.47 & 0.27 \\ 0.17 & 0.66 & 0.17 \\ 0.33 & 0.33 & 0.33 \end{bmatrix}$$

and

$$\hat{J} = \begin{bmatrix} 0.2 \\ 0.1 \\ 0.0 \end{bmatrix}$$

Example

- After 20 trajectories (episodes) of 4 steps each, we get

$$c_\pi = \begin{bmatrix} 0.75 \\ 0.0 \\ 1.0 \end{bmatrix} \qquad \mathbf{P}_\pi = \begin{bmatrix} 0.0 & 0.43 & 0.57 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \qquad J^\pi = \begin{bmatrix} 5.7 \\ 0.0 \\ 10.0 \end{bmatrix}$$

Actual value

$$c_\pi = \begin{bmatrix} 0.75 \\ 0.0 \\ 1.0 \end{bmatrix} \qquad \mathbf{P}_\pi = \begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \qquad J^\pi = \begin{bmatrix} 5.25 \\ 0.0 \\ 10.0 \end{bmatrix}$$

Computing Q^*

- Given a sample (x_t, a_t, c_t, x_{t+1})
- Perform the updates

$$\hat{c}(x_t, a_t) = \hat{c}(x_t, a_t) + \alpha_t(c_t - \hat{c}(x_t, a_t))$$

$$\hat{\mathbf{P}}(x' \mid x_t, a_t) = \hat{\mathbf{P}}(x' \mid x_t, a_t) + \alpha_t(\mathbb{I}[x_{t+1} = x'] - \hat{\mathbf{P}}(x' \mid x_t, a_t))$$

$$Q_{t+1}(x_t, a_t) = \hat{c}(x_t, a_t) + \gamma \sum_{x' \in \mathcal{X}} \hat{\mathbf{P}}_a(x' \mid x_t) \min_{a'} Q_t(x', a')$$

Update only affected entries

Does this work?

- Both methods

Theorem: The two approaches converge w.p.1 to J^π and Q^*, respectively, as long as every state (for J^π) or every state-action pair (for Q^*) is visited infinitely often.

Monte Carlo RL

Computing J^π

- We have that

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

We follow policy π

Computing J^π

- We have that

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

- We are computing an expectation over an infinitely long trajectory starting in state x

Computing J^π

- Let τ denote a (long) trajectory obtained using policy π:

$$\tau = \{x_0, c_0, x_1, c_1, \ldots, c_{T-1}, x_T\}$$

Actions selected using π

Computing J^π

- Let τ denote a (long) trajectory obtained using policy π:

$$\tau = \{x_0, c_0, x_1, c_1, \ldots, c_{T-1}, x_T\}$$

- We define the loss of trajectory τ as

$$L(\tau) = \sum_{t=0}^{T-1} \gamma^t c_t$$

Random quantity

Computing J^π

- Two important observations:
  - For T large enough, the difference

$$\sum_{t=0}^{\infty} \gamma^t c_t - \sum_{t=0}^{T-1} \gamma^t c_t$$

    is negligible

  - We have that

$$J^\pi(x_0) \approx \mathbb{E}[L(\tau)]$$

Computing an expectation!

Computing J^π

- Given a trajectory obtained with policy π

$$\tau_k = \{x_{k,0}, c_{k,0}, x_{k,1}, c_{k,1}, \ldots, c_{k,T-1}, x_{k,T}\}$$

- Compute loss

$$L(\tau_k) = \sum_{t=0}^{T-1} \gamma^t c_{k,t}$$

- Update

$$J_{k+1}(x_{k,0}) = J_k(x_{k,0}) + \alpha_k(L(\tau_k) - J_k(x_{k,0}))$$

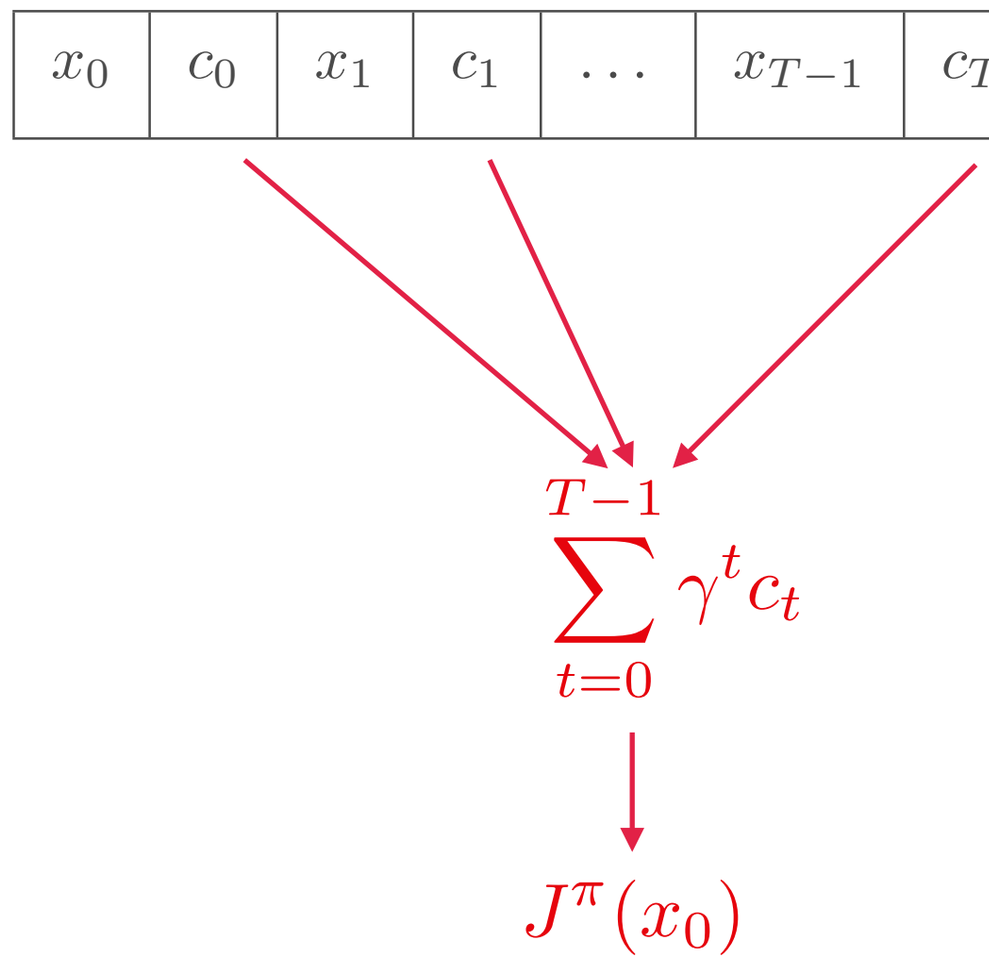
Does this work?

Theorem: For T large enough, Monte Carlo policy evaluation converge w.p.1 to J^π, as long as every state is visited infinitely often.

Let's revisit this

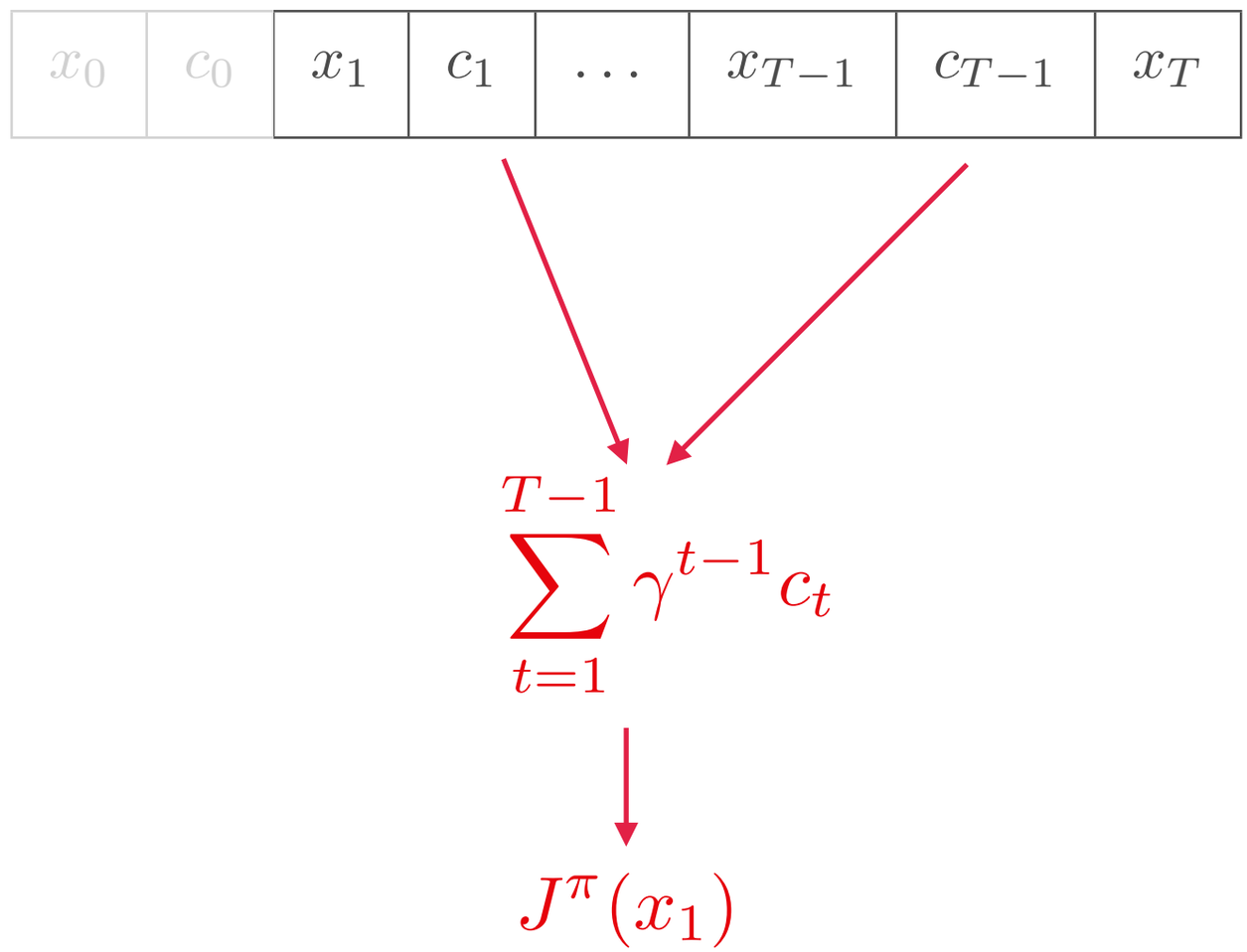
Exploration in Monte Carlo RL

- We can update the cost-to-go for every state visited along a trajectory



Exploration in Monte Carlo RL

- We can update the cost-to-go for every state visited along a trajectory



Exploration in Monte Carlo RL

- We can update the cost-to-go for every state visited along a trajectory
  - Consider only the first time a state appears: First-visit MC
  - Consider all the times a state appears: Every-visit MC

- Even with first-visit/every-visit MC, sufficient exploration generally requires exploring starts

Computing Q^*

- We have that

$$Q^*(x,a) = \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty} \gamma^t \mathrm{c}_t \mid \mathrm{x}_0 = x, \mathrm{a}_0 = a\right]$$

We follow optimal policy π^*

Computing Q^*

- We have that

$$Q^*(x, a) = \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty} \gamma^t \mathrm{c}_t \mid \mathrm{x}_0 = x, \mathrm{a}_0 = a\right]$$

- We compute an expectation over an infinitely long trajectory starting in state x and action a
- The trajectory is obtained from the optimal policy

We don't know the optimal policy!

Monte Carlo Policy Optimization

- We use the principle of policy iteration
  - We start with random policy π_0
  - We evaluate the policy (compute Q^{π_0})
  - We compute an improved policy π_1
  - Repeat

Computing Q^π

- Let τ denote a (long) trajectory obtained using policy π:

$$\tau_k = \{x_0, a_0, c_0, x_1, a_1, \ldots, c_{T-1}, x_T\}$$

Actions selected
using π
(except maybe a_0)

Computing Q^π

- Let τ denote a (long) trajectory obtained using policy π:

$$\tau = \{x_0, a_0, c_0, x_1, a_1, \ldots, c_{T-1}, x_T\}$$

- We define the loss of trajectory τ as

$$L(\tau) = \sum_{t=0}^{T-1} \gamma^t c_t$$

Computing Q^π

- Two important observations:
  - For T large enough, the difference

$$\sum_{t=0}^{\infty} \gamma^t c_t - \sum_{t=0}^{T-1} \gamma^t c_t$$

    is negligible

  - We have that

Computing an expectation!

$$Q^\pi(x_0, a_0) \approx \mathbb{E}[L(\tau)]$$

Computing Q^π

- Given a trajectory obtained with policy π

$$\tau_k = \{x_{k,0}, a_{k,0}, c_{k,0}, x_{k,1}, a_{k,1}, c_{k,1}, \ldots, a_{k,T-1}, c_{k,T-1}, x_{k,T}\}$$

- Compute loss

$$L(\tau_k) = \sum_{t=0}^{T-1} \gamma^t c_{k,t}$$

- Update

$$Q_{k+1}(x_{k,0}, a_{k,0}) = Q_k(x_{k,0}, a_{k,0}) + \alpha_k(L(\tau_k) - Q_k(x_{k,0}, a_{k,0}))$$

Exploration in Monte Carlo RL

- We can update the cost-to-go for every state-action pair visited along a trajectory

- Sufficient exploration generally requires exploring starts (for all state-action pairs)

Temporal difference methods

Computing J^π

- We have that

$$J^\pi(x) = c_\pi(x) + \gamma \sum_{y \in \mathcal{X}} \mathbf{P}_\pi(y \mid x) J^\pi(y)$$

which, back in lecture 6, we wrote as

$$\mathbf{J}^\pi = \mathbf{T}_\pi \mathbf{J}^\pi$$

$\mathbf{J}^\pi$ is a fixed point

Computing J^π

- Alternatively, for each state x, we can write

$$J^\pi(x) = \mathbb{E}_{\mathrm{a}\sim\pi(x),\mathrm{y}\sim\mathbf{P}(x,a)}\left[c(x,\mathrm{a}) + \gamma J^\pi(\mathrm{y})\right]$$

Random variables

Computing J^π

- Alternatively, for each state x, we can write

$$\mathbb{E}_{\mathrm{a}\sim\pi(x),\mathrm{y}\sim\mathbf{P}(x,a)}\left[c(x,\mathrm{a})+\gamma J^\pi(\mathrm{y})-J^\pi(x)\right]=0$$

J^π is the zero of this equation

Computing J^π

- Alternatively, for each state x, we can write

$$\mathbb{E}_{\mathrm{a}\sim\pi(x),\mathrm{y}\sim\mathbf{P}(x,a)}\left[c(x,\mathrm{a})+\gamma J^\pi(\mathrm{y})-J^\pi(x)\right]=0$$

- Using our stochastic approximation trick,

$$J_{t+1}(x_t)=J_t(x_t)+\alpha_t\boxed{\left[c_t+\gamma J_t(x_{t+1})-J_t(x_t)\right]}$$

Sample of $\mathrm{H}(\theta)$

Temporal difference

- The algorithm

$$J_{t+1}(x_t) = J_t(x_t) + \alpha_t \left[c_t + \gamma J_t(x_{t+1}) - J_t(x_t)\right]$$

is called TD-learning (temporal-difference learning) or TD(0)

- The quantity

$$\delta_t = \boxed{c_t + \gamma J_t(x_{t+1})} - J_t(x_t)$$

Estimate at time $t+1$

is called the **temporal difference** at time step t

Temporal difference

- The algorithm

$$J_{t+1}(x_t) = J_t(x_t) + \alpha_t \left[c_t + \gamma J_t(x_{t+1}) - J_t(x_t)\right]$$

is called TD-learning (temporal-difference learning) or TD(0)

- The quantity

Estimate at time t

$$\delta_t = c_t + \gamma J_t(x_{t+1}) - J_t(x_t)$$

is called the **temporal difference** at time step t

TD(0)

- Given a sample (x_t, c_t, x_{t+1}), where the action was selected from π

- Compute

$$J_{t+1}(x_t) = J_t(x_t) + \alpha_t \left[c_t + \gamma J_t(x_{t+1}) - J_t(x_t)\right]$$