Experimental design and Markov decision processes

The following problems

- Shortest path problems.
- Optimal stopping problems.
- Reinforcement learning problems.
- Experiment design (clinical trial) problems
- Advertising.

can be all formalised as Markov decision processes.

Applications

- Robotics.
- Economics.
- Automatic control.
- Resource allocation

Contents

The n meteorologists problem

Setting

- n meteorologists.

The n meteorologists problem

Setting

- n meteorologists.
- At time t:

The n meteorologists problem

Setting

- n meteorologists.
- At time t:
- The i-th meteorologist gives a prediction $x_{t,i} \in \{\texttt{dry}, \texttt{wet}\}$ for today's weather, y_t

The n meteorologists problem

Setting

- n meteorologists.
- At time t:
- The i-th meteorologist gives a prediction $x_{t,i} \in \{\texttt{dry}, \texttt{wet}\}$ for today's weather, y_t
- You must decide whether or not to commute by bike that day $a_t \in \{\texttt{bike}, \texttt{tram}\}$

The n meteorologists problem

Setting

- n meteorologists.
- At time t:
- The i-th meteorologist gives a prediction $x_{t,i} \in \{\texttt{dry}, \texttt{wet}\}$ for today's weather, y_t
- You must decide whether or not to commute by bike that day $a_t \in \{\texttt{bike}, \texttt{tram}\}$
- You obtain a reward $r_t = 1$ if it's dry and you bike, $r_t = -1$ if it's wet and you bike, and $r_t = 0$ otherwise.

The n meteorologists problem

Setting

- n meteorologists.
- At time t:
- The i-th meteorologist gives a prediction $x_{t,i} \in \{\texttt{dry}, \texttt{wet}\}$ for today's weather, y_t
- You must decide whether or not to commute by bike that day $a_t \in \{\texttt{bike}, \texttt{tram}\}$
- You obtain a reward $r_t = 1$ if it's dry and you bike, $r_t = -1$ if it's wet and you bike, and $r_t = 0$ otherwise.
- You then store the information y_t about the weather and the meteorologists' predictions.

Utility

$$U = \sum_t r_t$$

The n meteorologists problem is simple, as:

- You always see their predictions, as well as the weather, no matter whether you bike or take the tram (full information)
- Your actions do not influence their predictions (independence events)

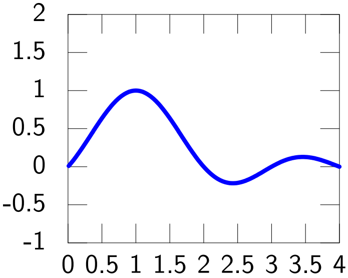
In the remainder, we'll see two settings where decisions are made with either partial information or in a dynamical system. Both of these settings can be formalised with Markov decision processes.

Bandit problems

Bandit problems

Applications

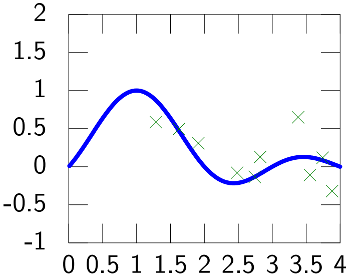
- Efficient optimisation.



Bandit problems

Applications

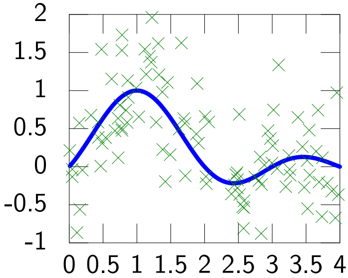
- Efficient optimisation.



Bandit problems

Applications

- Efficient optimisation.



Bandit problems

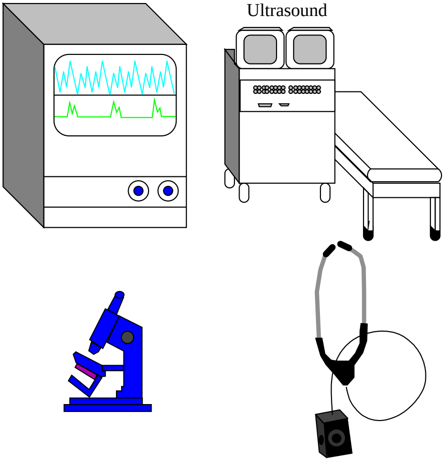
Applications

- Efficient optimisation.
- Online advertising.

Bandit problems

Applications

- Efficient optimisation.
- Online advertising.
- Clinical trials.



Bandit problems

Applications

- Efficient optimisation.
- Online advertising.
- Clinical trials.
- ROBOT SCIENTIST.

The stochastic n-armed bandit problem

Actions and rewards

- A set of actions $\mathcal{A} = \{1, \ldots, n\}$.
- Each action gives you a random reward with distribution $\mathbb{P}(r_t \mid a_t = i)$.
- The expected reward of the i-th arm is $\rho_i \triangleq \mathbb{E}(r_t \mid a_t = i)$.

Interaction at time t

1. You choose an action $a_t \in \mathcal{A}$.
2. You observe a random reward r_t drawn from the i-th arm.

The utility is the sum of the rewards obtained

$$U \triangleq \sum_t r_t.$$

We must maximise the expected utility, without knowing the values ρ_i.

Policy

Definition 1 (Policies)

A policy π is an algorithm for taking actions given the observed history $h_t \triangleq a_1, r_1, \ldots, a_t, r_t$

$$\mathbb{P}^\pi(a_{t+1} \mid h_t)$$

is the probability of the next action a_{t+1}.

Exercise 1

Why should our action depend on the complete history?

A *The next reward depends on all the actions we have taken.*

B *We don't know which arm gives the highest reward.*

C *The next reward depends on all the previous rewards.*

D *The next reward depends on the complete history.*

E *No idea.*

Policy

Definition 1 (Policies)

A policy π is an algorithm for taking actions given the observed history $h_t \triangleq a_1, r_1, \ldots, a_t, r_t$

$$\mathbb{P}^\pi(a_{t+1} \mid h_t)$$

is the probability of the next action a_{t+1}.

Example 2 (The expected utility of a uniformly random policy)

If $\mathbb{P}^\pi(a_{t+1} \mid \cdot) = 1/n$ for all t, then

Policy

Definition 1 (Policies)

A policy π is an algorithm for taking actions given the observed history $h_t \triangleq a_1, r_1, \ldots, a_t, r_t$

$$\mathbb{P}^\pi(a_{t+1} \mid h_t)$$

is the probability of the next action a_{t+1}.

Example 2 (The expected utility of a uniformly random policy)

If $\mathbb{P}^\pi(a_{t+1} \mid \cdot) = 1/n$ for all t, then

$$\mathbb{E}^\pi U = \mathbb{E}^\pi \left(\sum_{t=1}^T r_t\right) = \sum_{t=1}^T \mathbb{E}^\pi r_t = \sum_{t=1}^T \sum_{i=1}^n \frac{1}{n} \rho_i = \frac{T}{n} \sum_{i=1}^n \rho_i$$

Policy

Definition 1 (Policies)

A policy π is an algorithm for taking actions given the observed history $h_t \triangleq a_1, r_1, \ldots, a_t, r_t$

$$\mathbb{P}^\pi(a_{t+1} \mid h_t)$$

is the probability of the next action a_{t+1}.

The expected utility of a general policy

$$\mathbb{E}^\pi U = \mathbb{E}^\pi \left(\sum_{t=1}^{T} r_t\right)$$

Policy

Definition 1 (Policies)

A policy π is an algorithm for taking actions given the observed history $h_t \triangleq a_1, r_1, \ldots, a_t, r_t$

$$\mathbb{P}^\pi(a_{t+1} \mid h_t)$$

is the probability of the next action a_{t+1}.

The expected utility of a general policy

$$\mathbb{E}^\pi U = \mathbb{E}^\pi \left(\sum_{t=1}^T r_t\right) = \sum_{t=1}^T \mathbb{E}^\pi(r_t) \tag{1.1}$$

Policy

Definition 1 (Policies)

A policy π is an algorithm for taking actions given the observed history $h_t \triangleq a_1, r_1, \ldots, a_t, r_t$

$$\mathbb{P}^\pi(a_{t+1} \mid h_t)$$

is the probability of the next action a_{t+1}.

The expected utility of a general policy

$$\mathbb{E}^\pi U = \mathbb{E}^\pi \left(\sum_{t=1}^T r_t \right) = \sum_{t=1}^T \mathbb{E}^\pi(r_t) \tag{1.1}$$

$$= \sum_{t=1}^T \sum_{a_t \in \mathcal{A}} \mathbb{E}(r_t \mid a_t) \sum_{h_{t-1}} \mathbb{P}^\pi(a_t \mid h_{t-1}) \, \mathbb{P}^\pi(h_{t-1})$$

Contents

Bernoulli bandits

Example 2 (Bernoulli bandits)

Consider n Bernoulli distributions with parameters ω_i ($i = 1, \dots, n$) such that $r_t \mid a_t = i \sim \mathcal{B}ern(\omega_i)$. Then,

$$\mathbb{P}(r_t = 1 \mid a_t = i) = \omega_i \qquad \mathbb{P}(r_t = 0 \mid a_t = i) = 1 - \omega_i \tag{1.2}$$

Then the expected reward for the i-th bandit is $\rho_i \triangleq \mathbb{E}(r_t \mid a_t = i) =?$.

Bernoulli bandits

Example 2 (Bernoulli bandits)

Consider n Bernoulli distributions with parameters ω_i ($i = 1, \dots, n$) such that $r_t \mid a_t = i \sim \mathcal{B}ern(\omega_i)$. Then,

$$\mathbb{P}(r_t = 1 \mid a_t = i) = \omega_i \qquad \mathbb{P}(r_t = 0 \mid a_t = i) = 1 - \omega_i \tag{1.2}$$

Then the expected reward for the i-th bandit is $\rho_i \triangleq \mathbb{E}(r_t \mid a_t = i) = \omega_i$.

Bernoulli bandits

Example 2 (Bernoulli bandits)

Consider n Bernoulli distributions with parameters ω_i ($i = 1, \ldots, n$) such that $r_t \mid a_t = i \sim \mathcal{Bern}(\omega_i)$. Then,

$$\mathbb{P}(r_t = 1 \mid a_t = i) = \omega_i \qquad \mathbb{P}(r_t = 0 \mid a_t = i) = 1 - \omega_i \tag{1.2}$$

Then the expected reward for the i-th bandit is $\rho_i \triangleq \mathbb{E}(r_t \mid a_t = i) = \omega_i$.

Exercise 1 (The optimal policy under perfect knowledge)

If we know ω_i for all i, what is the best policy?

- A *At every step, play the bandit i with the greatest ω_i.*
- B *Prefer bandits i with larger ω_i, but play them all.*
- C *It depends on the horizon T.*
- D *Prefer bandits i which you have played the least so far, but play them all.*
- E *It is too complicated.*

The unknown reward case

Say you keep a running average of the reward obtained by each arm

$$\hat{\rho}_{t,i} = R_{t,i}/n_{t,i}$$

where $n_{t,i}$ is the number of times you played arm i and $R_{t,i}$ the total reward received from i so that whenever you play $a_t = i$:

$$R_{t+1,i} = R_{t,i} + r_t, \qquad n_{t+1,i} = n_{t,i} + 1.$$

The unknown reward case

Say you keep a running average of the reward obtained by each arm

$$\hat{\rho}_{t,i} = R_{t,i}/n_{t,i}$$

where $n_{t,i}$ is the number of times you played arm i and $R_{t,i}$ the total reward received from i so that whenever you play $a_t = i$:

$$R_{t+1,i} = R_{t,i} + r_t, \qquad n_{t+1,i} = n_{t,i} + 1.$$

Exercise 2 (The optimal policy under imperfect knowledge)

If we just keep travek of the averages $\hat{\rho}_{t,i}$, for all i, what is the best policy?

- A *At every step, play the bandit i with the greatest $\hat{\rho}_{t,i}$.*
- B *Prefer bandits i with larger $\hat{\rho}_{t,i}$, but play them all.*
- C *It depends on the horizon T.*
- D *Prefer bandits i with smaller $n_{t,i}$, but play them all.*
- E *It is too complicated.*

The unknown reward case

Say you keep a running average of the reward obtained by each arm

$$\hat{\rho}_{t,i} = R_{t,i}/n_{t,i}$$

where $n_{t,i}$ is the number of times you played arm i and $R_{t,i}$ the total reward received from i so that whenever you play $a_t = i$:

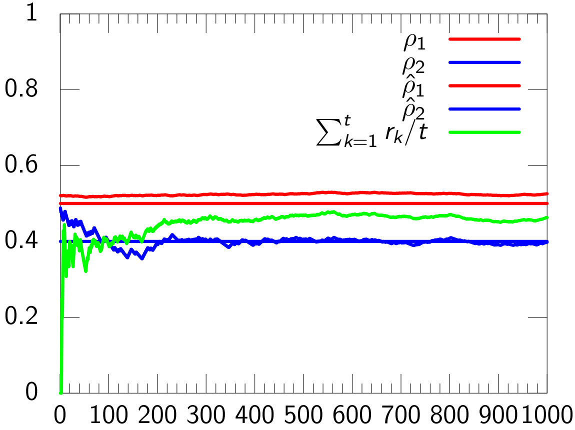
$$R_{t+1,i} = R_{t,i} + r_t, \qquad n_{t+1,i} = n_{t,i} + 1.$$

You could choose to play the strategy

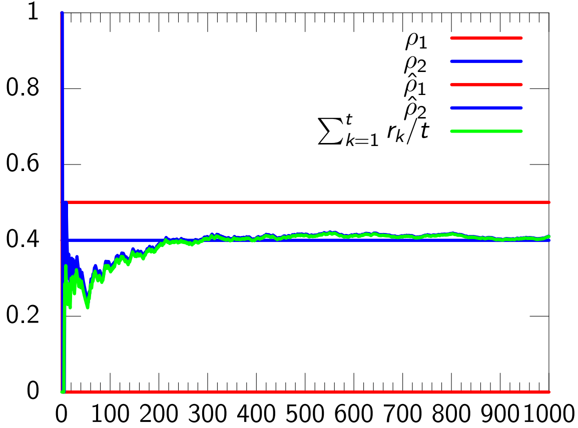
$$a_t = \arg\max_i \hat{\rho}_{t,i}.$$

where we use non-zero initial values $n_{0,i}, R_{0,i}$!

The uniform policy

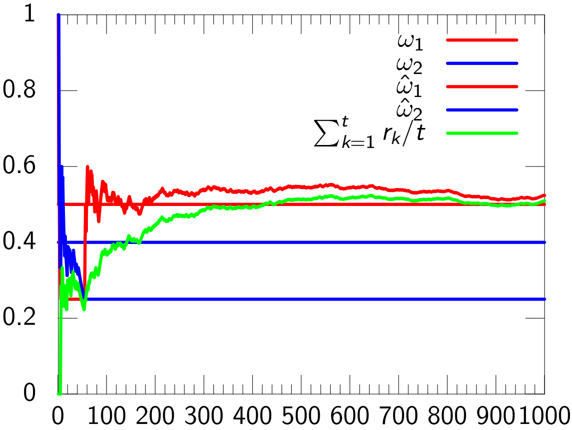


The greedy policy



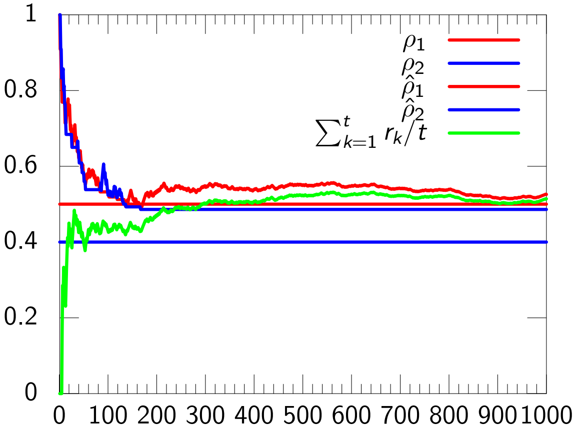
For $n_{0,i} = R_{0,i} = 0$

The greedy policy



For $n_{0,i} = R_{0,i} = 1$

The greedy policy



For $n_{0,i} = R_{0,i} = 10$

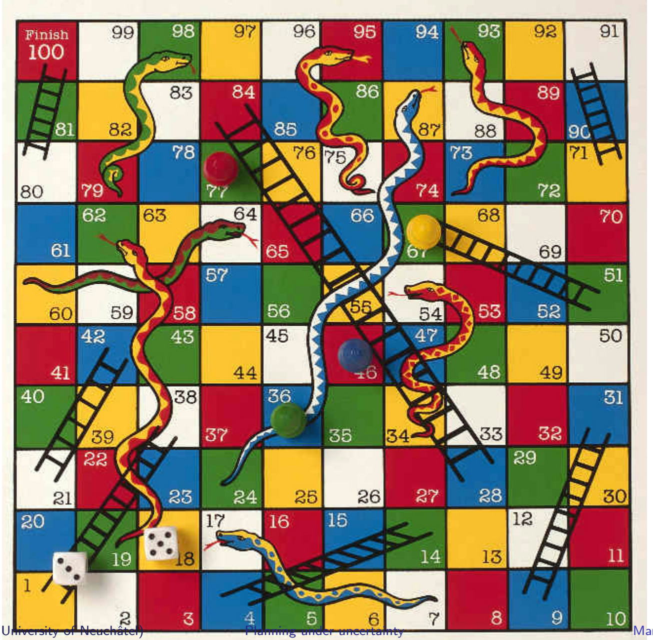
Summary

- Bandit problems are the simplest type of partial information problems.
- Learning policies for such problems must remember the complete history.
- If we know the problem parameters, simple stationary policies are optimal.
- If we don't, then our policies must carefuly balance:
  - Exploration: Learning more about the problem.
  - Exploitation: Using what is already known.

From now on, we focus on the case where the problem is perfectly known.

Contents

A gentle introduction to Markov processes



Markov process

$$s_{t-1} \longrightarrow s_t \longrightarrow s_{t+1}$$

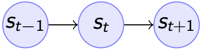
Definition 3 (Markov Process – or Markov Chain)

The sequence $\{s_t \mid t = 1, \ldots\}$ of random variables $s_t : \Omega \to \mathcal{S}$ is a Markov process if

$$\mathbb{P}(s_{t+1} \mid s_t, \ldots, s_1) = \mathbb{P}(s_{t+1} \mid s_t). \tag{2.1}$$

- s_t is state of the Markov process at time t.
- $\mathbb{P}(s_{t+1} \mid s_t)$ is the transition kernel of the process.

Markov process



Definition 3 (Markov Process – or Markov Chain)

The sequence $\{s_t \mid t = 1, \ldots\}$ of random variables $s_t : \Omega \to \mathcal{S}$ is a Markov process if

$$\mathbb{P}(s_{t+1} \mid s_t, \ldots, s_1) = \mathbb{P}(s_{t+1} \mid s_t). \tag{2.1}$$

- s_t is state of the Markov process at time t.
- $\mathbb{P}(s_{t+1} \mid s_t)$ is the transition kernel of the process.

Exercise 2 (Finite state machine with random input)

Let $\omega = \omega_1, \ldots, \omega_t$ be an infinitely long random string of bits, $\Omega = \mathcal{S} = \{0, 1\}$ and:

$$s_{t+1} = s_t \oplus \omega_t.$$

Is s_t a Markov process?

Contents

Reinforcement learning

The reinforcement learning problem.

Learning to act in an unknown environment, by interaction and reinforcement.

- The environment has a changing state s_t.
- The agents observes the state s_t.
- The agent takes action a_t.
- It receives rewards r_t.

The goal (informally)

Maximise total reward $\sum_t r_t$

Types of environments

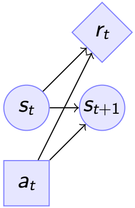
- Markov decision processes (MDPs).
- Partially observable MDPs (POMDPs).
- (Partially observable) Markov games.

Markov decision processes

Markov decision processes (MDP) μ.

At each time step t:

- We observe **state** $s_t \in \mathcal{S}$.
- We take **action** $a_t \in \mathcal{A}$.
- We receive a **reward** $r_t \in \mathbb{R}$.



Markov property of the reward and state distribution

$$\mathbb{P}_\mu(s_{t+1} \mid s_t, a_t) \qquad \text{(Transition distribution)}$$

$$\mathbb{P}_\mu(r_t \mid s_t, a_t) \qquad \text{(Reward distribution)}$$

The agent

The agent's policy π

$$\pi(a_t \mid s_t, \ldots, s_1, a_{t-1}, \ldots, a_1) \qquad \text{(history-dependent policy)}$$

$$\pi(a_t \mid s_t) \qquad \text{(Markov policy)}$$

Definition 4 (Utility)

Given a horizon T, the utility can be defined as

$$U_t \triangleq \sum_{k=0}^{T-t} r_{t+k} \tag{3.1}$$

The agent wants to to find π maximising the expected total future reward

$$\mathbb{E}^{\pi}_{\mu} U_t = \mathbb{E}^{\pi}_{\mu} \sum_{k=0}^{T-t} r_{t+k}. \qquad \text{(expected utility)}$$

Contents

State value function

$$V^{\pi}_{\mu,t}(s) \triangleq \mathbb{E}^{\pi}_{\mu}(U_t \mid s_t = s) \tag{3.2}$$

The optimal policy π^*

$$\pi^*(\mu) : V^{\pi^*(\mu)}_{t,\mu}(s) \geq V^{\pi}_{t,\mu}(s) \quad \forall \pi, t, s \tag{3.3}$$

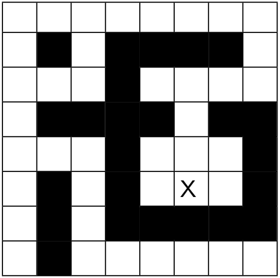
dominates all other policies π everywhere in $\mathcal{S}$.
The optimal value function V^*

$$V^{*}_{t,\mu}(s) \triangleq V^{\pi^*(\mu)}_{t,\mu}(s), \tag{3.4}$$

is the value function of the optimal policy π^*.

Contents

Deterministic shortest-path problems



Properties

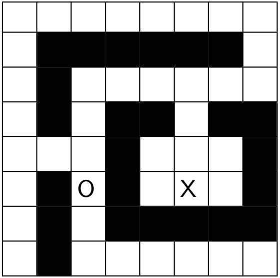
- $T \to \infty$.
- $r_t = -1$ unless $s_t = X$, in which case $r_t = 0$.
- $\mathbb{P}_\mu(s_{t+1} = X | s_t = X) = 1$.
- $\mathcal{A} = \{\mathrm{North}, \mathrm{South}, \mathrm{East}, \mathrm{West}\}$
- Transitions are deterministic and walls block.

14	13	12	11	10	9	8	7
15		13					6
16	15	14		4	3	4	5
17					2		
18	19	20		2	1	2	
19		21		1	0	1	
20		22					
21		23	24	25	26	27	28

Properties

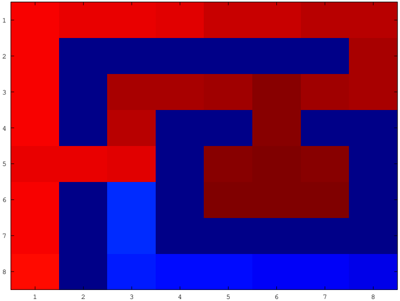
- $\gamma = 1, \; T \to \infty$.
- $r_t = -1$ unless $s_t = X$, in which case $r_t = 0$.
- The length of the shortest path from s equals the negative value of the optimal policy.
- Also called *cost-to-go*.

Stochastic shortest path problem with a pit

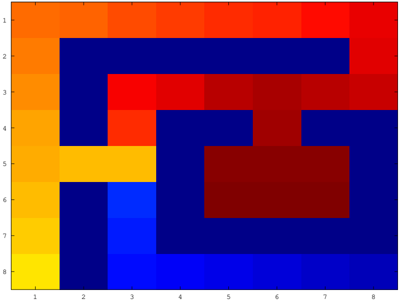


Properties

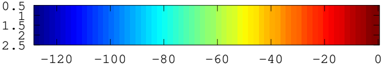
- $T \to \infty$.
- $r_t = -1$, but $r_t = 0$ at X and -100 at O and the problem ends.
- $\mathbb{P}_\mu(s_{t+1} = X | s_t = X) = 1$.
- $\mathcal{A} = \{\text{North}, \text{South}, \text{East}, \text{West}\}$
- Moves to a random direction with probability ω. Walls block.



(a) $\omega = 0.1$



(b) $\omega = 0.5$



(c) value

Figure: Pit maze solutions for two values of ω.

Exercise 3

- *Why should we only take the shortcut in (a)?*
- *Why does the agent commit suicide at the bottom?*

Contents

How to evaluate a policy

$$V^{\pi}_{\mu,t}(s) \triangleq \mathbb{E}^{\pi}_{\mu}(U_t \mid s_t = s) \tag{4.1}$$

$$\tag{4.2}$$

This derivation directly gives a number of policy evaluation algorithms.

How to evaluate a policy

$$V^{\pi}_{\mu,t}(s) \triangleq \mathbb{E}^{\pi}_{\mu}(U_t \mid s_t = s) \tag{4.1}$$

$$= \sum_{k=0}^{T-t} \mathbb{E}^{\pi}_{\mu}(r_{t+k} \mid s_t = s) \tag{4.2}$$

$$\tag{4.3}$$

This derivation directly gives a number of policy evaluation algorithms.

How to evaluate a policy

$$V^{\pi}_{\mu,t}(s) \triangleq \mathbb{E}^{\pi}_{\mu}(U_t \mid s_t = s) \tag{4.1}$$

$$= \sum_{k=0}^{T-t} \mathbb{E}^{\pi}_{\mu}(r_{t+k} \mid s_t = s) \tag{4.2}$$

$$= \mathbb{E}^{\pi}_{\mu}(r_t \mid s_t = s) + \mathbb{E}^{\pi}_{\mu}(U_{t+1} \mid s_t = s) \tag{4.3}$$

$$\tag{4.4}$$

This derivation directly gives a number of policy evaluation algorithms.

How to evaluate a policy

$$
\begin{aligned}
V^{\pi}_{\mu,t}(s) &\triangleq \mathbb{E}^{\pi}_{\mu}(U_t \mid s_t = s) && (4.1)\\
&= \sum_{k=0}^{T-t} \mathbb{E}^{\pi}_{\mu}(r_{t+k} \mid s_t = s) && (4.2)\\
&= \mathbb{E}^{\pi}_{\mu}(r_t \mid s_t = s) + \mathbb{E}^{\pi}_{\mu}(U_{t+1} \mid s_t = s) && (4.3)\\
&= \mathbb{E}^{\pi}_{\mu}(r_t \mid s_t = s) + \sum_{i \in \mathcal{S}} V^{\pi}_{\mu,t+1}(i)\, \mathbb{P}^{\pi}_{\mu}(s_{t+1} = i | s_t = s). && (4.4)
\end{aligned}
$$

This derivation directly gives a number of policy evaluation algorithms.

Monte-Carlo Policy evaluation

for $s \in \mathcal{S}$ **do**

end for

Monte-Carlo Policy evaluation

for $s \in \mathcal{S}$ **do**
  for $k = 1, \ldots, K$ **do**
    Execute policy π and record total reward K times:

$$\hat{R}_k(s) = \sum_{t=1}^{T} r_{t,k}.$$

  end for
end for

Monte-Carlo Policy evaluation

for $s \in \mathcal{S}$ **do**
  for $k = 1, \ldots, K$ **do**
    Execute policy π and record total reward K times:

$$\hat{R}_k(s) = \sum_{t=1}^{T} r_{t,k}.$$

  end for
  Calculate estimate:

$$\boldsymbol{v}_t(s) = \frac{1}{K} \sum_{k=1}^{K} \hat{R}_k(s).$$

end for

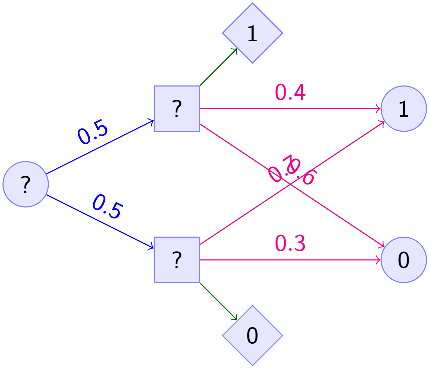
Backwards induction policy evaluation

for State $s \in S$, $t = T, \ldots, 1$ **do**

Update values of states:

$$v_t(s_t) = \sum_{a_t \in \mathcal{A}} \mathbb{P}^\pi(a_t \mid s_t) \left\{ \mathbb{E}_\mu(r_t \mid s_t, a_t) + \sum_{s_{t+1} \in \mathcal{S}} \mathbb{P}_\mu(s_{t+1} \mid s_t, a_t) v_{t+1}(s_{t+1}) \right\}$$

end for



Exercise 4

What is the value $v_t(s_t)$ of the first state?

A *1.4*

B *1.05*

C *1.0*

D *0.7*

E *0*

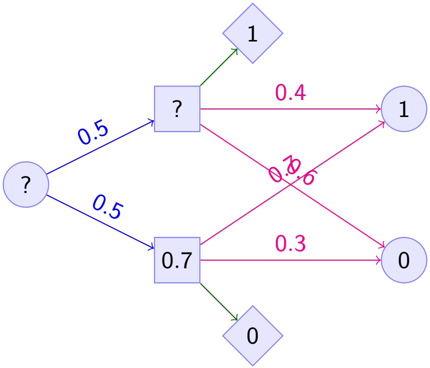
Backwards induction policy evaluation

for State $s \in S$, $t = T, \ldots, 1$ **do**

Update values of states:

$$v_t(s_t) = \sum_{a_t \in \mathcal{A}} \mathbb{P}^\pi(a_t \mid s_t) \left\{ \mathbb{E}_\mu(r_t \mid s_t, a_t) + \sum_{s_{t+1} \in \mathcal{S}} \mathbb{P}_\mu(s_{t+1} \mid s_t, a_t) v_{t+1}(s_{t+1}) \right\}$$

end for



Exercise 4

What is the value $v_t(s_t)$ of the first state?

A *1.4*

B *1.05*

C *1.0*

D *0.7*

E *0*

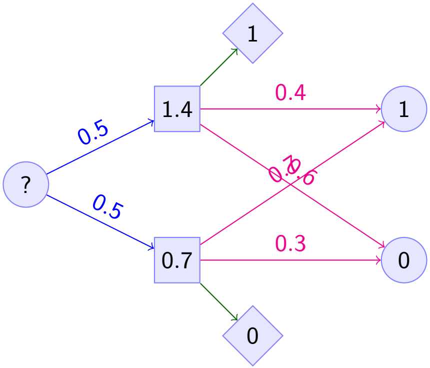
Backwards induction policy evaluation

for State $s \in S$, $t = T, \ldots, 1$ **do**

Update values of states:

$$v_t(s_t) = \sum_{a_t \in \mathcal{A}} \mathbb{P}^{\pi}(a_t \mid s_t) \left\{ \mathbb{E}_{\mu}(r_t \mid s_t, a_t) + \sum_{s_{t+1} \in \mathcal{S}} \mathbb{P}_{\mu}(s_{t+1} \mid s_t, a_t) v_{t+1}(s_{t+1}) \right\}$$

end for



Exercise 4

What is the value $v_t(s_t)$ of the first state?

A *1.4*

B *1.05*

C *1.0*

D *0.7*

E *0*

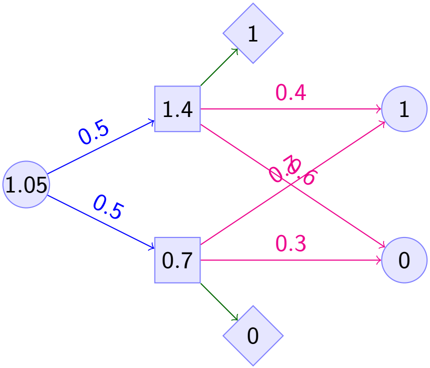
Backwards induction policy evaluation

for State $s \in S$, $t = T, \ldots, 1$ **do**

Update values of states:

$$v_t(s_t) = \sum_{a_t \in \mathcal{A}} \mathbb{P}^{\pi}(a_t \mid s_t) \left\{ \mathbb{E}_{\mu}(r_t \mid s_t, a_t) + \sum_{s_{t+1} \in \mathcal{S}} \mathbb{P}_{\mu}(s_{t+1} \mid s_t, a_t) v_{t+1}(s_{t+1}) \right\}$$

end for



Exercise 4

What is the value $v_t(s_t)$ of the first state?

A *1.4*

B *1.05*

C *1.0*

D *0.7*

E *0*

Contents

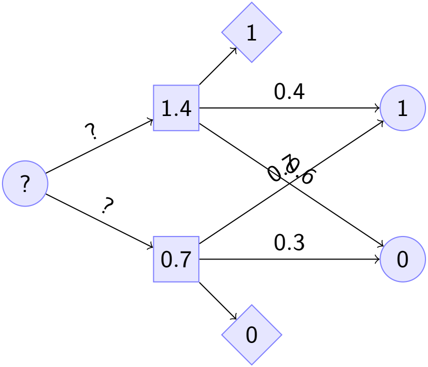
Backwards induction policy optimization

for State $s \in S$, $t = T, \ldots, 1$ **do**

Update values

$$v_t(s_t) = \max_{a_t \in \mathcal{A}} \left\{ \mathbb{E}_\mu(r_t \mid s_t, a_t) + \sum_{s_{t+1} \in \mathcal{S}} \mathbb{P}_\mu(s_{t+1} \mid s_t, a_t) v_{t+1}(s_{t+1}) \right\}$$

end for



Exercise 5

What is the value $v_t(s_t)$ of the first state?

A *1.4*

B *1.05*

C *1.0*

D *0.7*

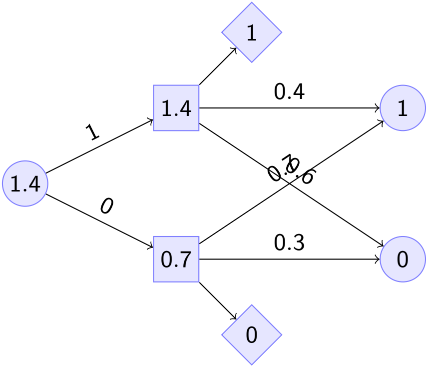
E *0*

Backwards induction policy optimization

for State $s \in S$, $t = T, \ldots, 1$ **do**
  Update values

$$v_t(s_t) = \max_{a_t \in \mathcal{A}} \left\{ \mathbb{E}_\mu(r_t \mid s_t, a_t) + \sum_{s_{t+1} \in \mathcal{S}} \mathbb{P}_\mu(s_{t+1} \mid s_t, a_t) v_{t+1}(s_{t+1}) \right\}$$

end for



Exercise 5

What is the value $v_t(s_t)$ of the first state?

A *1.4*

B *1.05*

C *1.0*

D *0.7*

E *0*

Contents

Discounted total reward.

$$U_t = \lim_{T \to \infty} \sum_{k=t}^{T} \gamma^k r_k, \qquad \gamma \in (0, 1)$$

Definition 5

A policy π is stationary if $\pi(a_t \mid s_t)$ does not depend on t.

Remark 1

We can use the Markov chain kernel $\boldsymbol{P}_{\mu,\pi}$ *to write the expected reward vector as*

$$\boldsymbol{v}^\pi = \sum_{t=0}^{\infty} \gamma^t \boldsymbol{P}_{\mu,\pi}^t \boldsymbol{r} \tag{5.1}$$

Theorem 6

For any stationary policy π, $\boldsymbol{v}^\pi$ is the unique solution of

$$\boldsymbol{v} = \boldsymbol{r} + \gamma \boldsymbol{P}_{\mu,\pi} \boldsymbol{v}. \quad \leftarrow \textit{fixed point} \tag{5.2}$$

In addition, the solution is:

$$\boldsymbol{v}^\pi = (\boldsymbol{I} - \gamma \boldsymbol{P}_{\mu,\pi})^{-1} \boldsymbol{r}. \tag{5.3}$$

Contents

Value iteration

for $n = 1, 2, \ldots$ and $s \in \mathcal{S}$ **do**

$v_n(s) = \max_a r(s, a) + \gamma \sum_{s' \in \mathcal{S}} P_\mu(s' \mid s, a) v_{n-1}(s')$

end for

Policy Iteration

Input μ, $\mathcal{S}$.
Initialise $\boldsymbol{v}_0$.
for $n = 1, 2, \ldots$ **do**
  $\pi_{n+1} = \arg\max_\pi \{\boldsymbol{r} + \gamma \boldsymbol{P}_\pi \boldsymbol{v}_n\}$ `// policy improvement`
  $\boldsymbol{v}_{n+1} = V_\mu^{\pi_{n+1}}$ `// policy evaluation`
  break if $\pi_{n+1} = \pi_n$.
end for
Return $\pi_n, \boldsymbol{v}_n$.

[1] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[2] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2001.

[3] Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[4] Herman Chernoff. Sequential design of experiments. *Annals of Mathematical Statistics*, 30(3):755–770, 1959.

[5] Herman Chernoff. Sequential models for clinical trials. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol.4*, pages 805–812. University of California Press, 1966.

[6] Morris H. DeGroot. *Optimal Statistical Decisions*. John Wiley & Sons, 1970.

[7] Marting L. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New Jersey, US, 1994.

[8] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *ICML 2010*, 2010.

[9] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.