Bayesian reinforcement learning

Markov decision processes and approximate Bayesian computation

Christos Dimitrakakis

Chalmers

April 16, 2015

Overview

Objective Probability

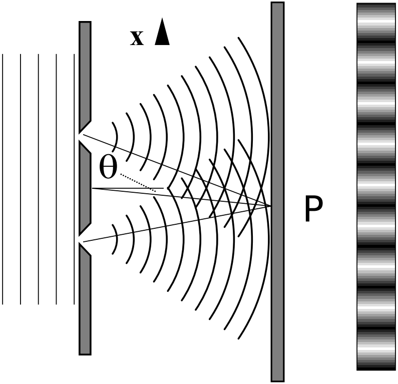


Figure: The double slit experiment

Objective Probability

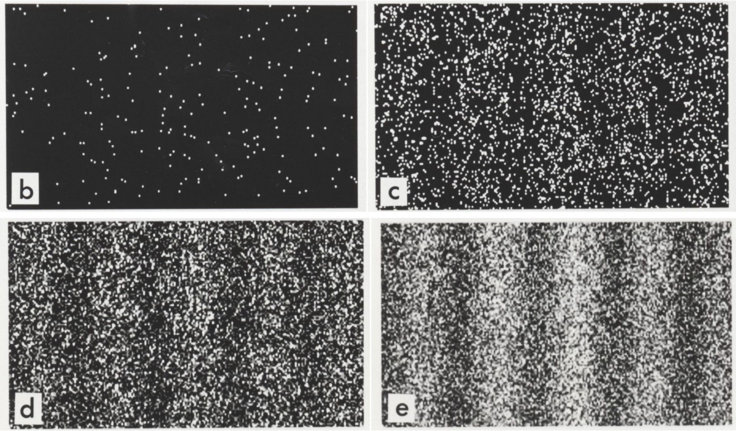


Figure: The double slit experiment

Objective Probability

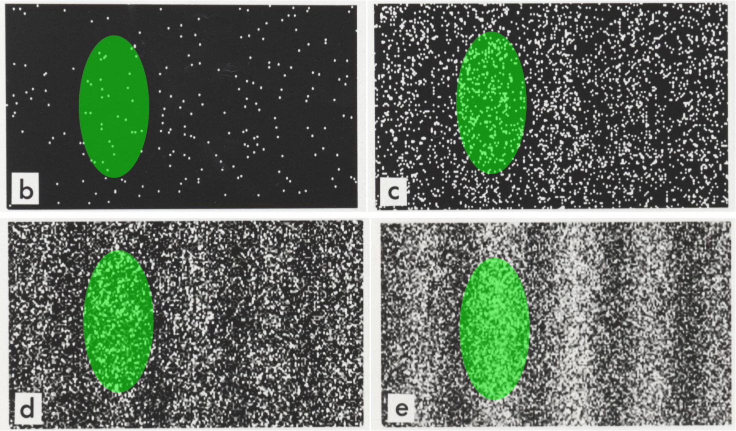


Figure: The double slit experiment

What about everyday life?

Subjective probability

- Making decisions requires making predictions.

Subjective probability

- Making decisions requires making predictions.
- Outcomes of decisions are uncertain.

Subjective probability

- Making decisions requires making predictions.
- Outcomes of decisions are uncertain.
- How can we represent this uncertainty?

Subjective probability

- Making decisions requires making predictions.
- Outcomes of decisions are uncertain.
- How can we represent this uncertainty?

Subjective probability

- Describe which events we think are more likely.
- We quantify this with probability.

Why probability?

- Quantifies uncertainty in a "natural" way.
- A framework for drawing conclusions from data.
- Computationally convenient for decision making.

Rewards

- We are going to receive a reward r from a set R of possible rewards.
- We prefer some rewards to others.

Example 1 (Possible sets of rewards R)

- R is a set of tickets to different musical events.
- R is a set of financial commodities.

When we cannot select rewards directly

- In most problems, we cannot just choose which reward to receive.
- We can only specify a distribution on rewards.

Example 2 (Route selection)

- Each reward $r \in R$ is the time it takes to travel from A to B.
- Route P_1 is faster than P_2 in heavy traffic and vice-versa.
- Which route should be preferred, given a certain probability for heavy traffic?

In order to choose between random rewards, we use the concept of utility.

Definition 3 (Utility)

The utility is a function $U : R \to \mathbb{R}$, such that for all $a, b \in R$

$$a \succsim^* b \quad \text{iff} \quad U(a) \geq U(b), \tag{1.1}$$

The expected utility of a distribution P on R is:

$$\mathbb{E}_P(U) = \sum_{r \in R} U(r) P(r)$$

(1.3)

Definition 3 (Utility)

The utility is a function $U : R \to \mathbb{R}$, such that for all $a, b \in R$

$$a \succsim^* b \quad \text{iff} \quad U(a) \geq U(b), \tag{1.1}$$

The expected utility of a distribution P on R is:

$$\mathbb{E}_P(U) = \sum_{r \in R} U(r) P(r) \tag{1.2}$$

$$= \int_R U(r) \, \mathrm{d}P(r) \tag{1.3}$$

Definition 3 (Utility)

The utility is a function $U : R \to \mathbb{R}$, such that for all $a, b \in R$

$$a \succsim^* b \quad \text{iff} \quad U(a) \geq U(b), \tag{1.1}$$

The expected utility of a distribution P on R is:

$$\mathbb{E}_P(U) = \sum_{r \in R} U(r) P(r) \tag{1.2}$$

$$= \int_R U(r) \, \mathrm{d}P(r) \tag{1.3}$$

Assumption 1 (The expected utility hypothesis)

The utility of P is equal to the expected utility of the reward under P. Consequently,

$$P \succsim^* Q \quad \textit{iff} \quad \mathbb{E}_P(U) \geq \mathbb{E}_Q(U). \tag{1.4}$$

i.e. we prefer P to Q iff the expected utility under P is higher than under Q

The St. Petersburg Paradox

A simple game [Bernoulli, 1713]

- A fair coin is tossed until a head is obtained.
- If the first head is obtained on the n-th toss, our reward will be 2^n currency units.

The St. Petersburg Paradox

A simple game [Bernoulli, 1713]

- A fair coin is tossed until a head is obtained.
- If the first head is obtained on the n-th toss, our reward will be 2^n currency units.

How much are you willing to pay, to play this game once?

The St. Petersburg Paradox

A simple game [Bernoulli, 1713]

- A fair coin is tossed until a head is obtained.
- If the first head is obtained on the n-th toss, our reward will be 2^n currency units.

- The probability to stop at round n is 2^{-n}.

The St. Petersburg Paradox

A simple game [Bernoulli, 1713]

- A fair coin is tossed until a head is obtained.
- If the first head is obtained on the n-th toss, our reward will be 2^n currency units.

- The probability to stop at round n is 2^{-n}.
- Thus, the expected monetary gain of the game is

$$\sum_{n=1}^{\infty} 2^n 2^{-n} = \infty.$$

The St. Petersburg Paradox

A simple game [Bernoulli, 1713]

- A fair coin is tossed until a head is obtained.
- If the first head is obtained on the n-th toss, our reward will be 2^n currency units.

- The probability to stop at round n is 2^{-n}.
- Thus, the expected monetary gain of the game is

$$\sum_{n=1}^{\infty} 2^n 2^{-n} = \infty.$$

- If your utility function were linear ($U(r) = r$) you'd be willing to pay any amount to play.
- You might not internalise the setup of the game (is the coin really fair?)

Summary

- We can subjectively indicate which events we think are more likely.
- We can define a subjective probability P for all events.
- Similarly, we can subjectively indicate preferences for rewards.
- We can determine a utility function for all rewards.
- Hypothesis: we prefer the probability distribution with the highest expected utility.
- This allows us to create algorithms for decision making.

Experimental design and Markov decision processes

The following problems

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

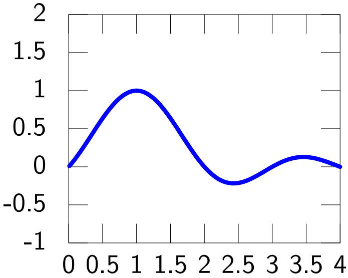
can be all formalised as Markov decision processes.

Bandit problems

Bandit problems

Applications

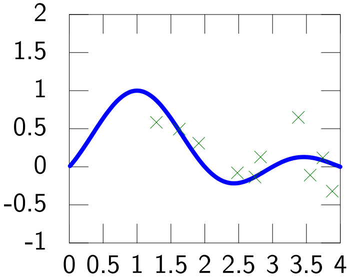
- Efficient optimisation.



Bandit problems

Applications

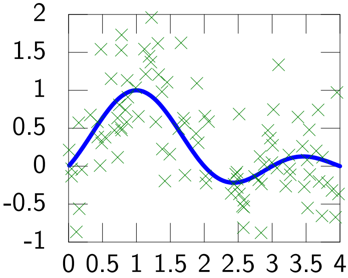
- Efficient optimisation.



Bandit problems

Applications

- Efficient optimisation.



Bandit problems

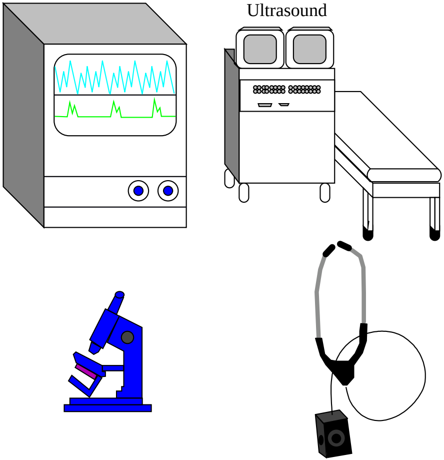
Applications

- 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 $\omega_i \triangleq \mathbb{E}(r_t \mid a_t = i)$.

Utility

The utility is the sum of the individual rewards $r = r_1, \ldots, r_T$

$$U(r) \triangleq \sum_{t=1}^{T} r_t.$$

Definition 4 (Policies)

A policy π is an algorithm for taking actions given the observed history.

$$\mathbb{P}^\pi(a_{t+1} \mid a_1, r_1, \ldots, a_t, r_t)$$

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

Bernoulli bandits

Example 5 (Bernoulli bandits)

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

$$\mathbb{P}(r_t = 1 \mid a_t = i) = \omega_i \qquad \mathbb{P}(r_t = 0 \mid a_t = i) = 1 - \omega_i \tag{2.1}$$

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

Bernoulli bandits

Example 5 (Bernoulli bandits)

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

$$\mathbb{P}(r_t = 1 \mid a_t = i) = \omega_i \qquad \mathbb{P}(r_t = 0 \mid a_t = i) = 1 - \omega_i \tag{2.1}$$

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

Exercise 1 (The optimal policy)

- *If we know ω_i for all i, what is the best policy?*
- *What if we don't?*

A simple heuristic for the unknown reward case

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

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

- $n_{t,i}$ the number of times you played arm i
- $R_{t,i}$ the total reward received from i.

Whenever you play $a_t = i$:

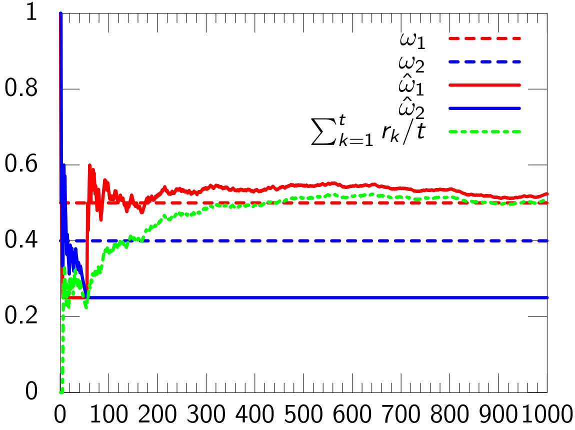
$$R_{t+1,i} = R_{t,i} + r_t, \qquad n_{t+1,i} = n_{t,i} + 1.$$

Greedy policy:

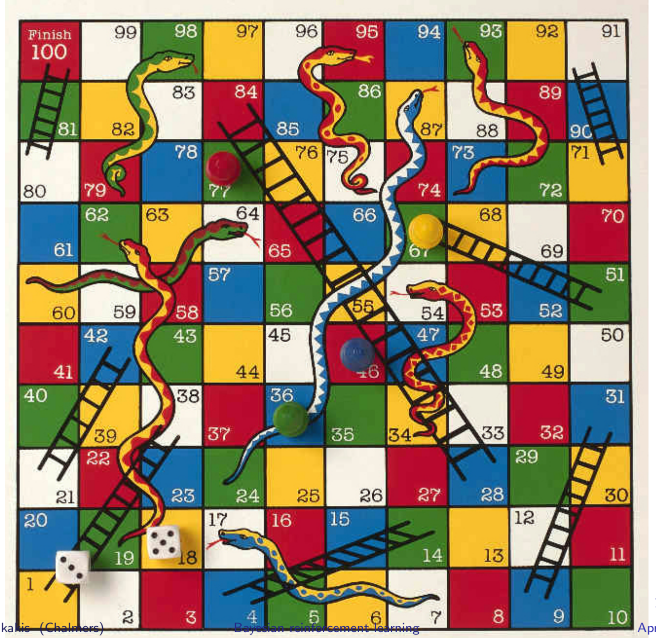
$$a_t = \arg\max_i \hat{\omega}_{t,i}.$$

What should the initial values $n_{0,i}, R_{0,i}$ be?

The greedy policy for $n_{0,i} = R_{0,i} = 1$



A Markov process



Markov process

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

Definition 6 (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

$$\mathbb{P}(s_{t+1} \mid s_t, \ldots, s_1) = \mathbb{P}(s_{t+1} \mid s_t). \tag{3.1}$$

- 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.

The state of an algorithm

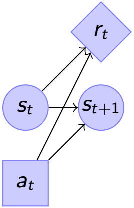
Observe that the R, n vectors of our greedy bandit algorithm form a Markov process. They also summarise our belief about which arm is the best.

Markov decision processes

Markov decision processes (MDP).

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

$$\mathbb{P}^\pi(a_t \mid r_t, s_t, a_t, \ldots, r_1, s_1, a_1) \qquad \text{(history-dependent policy)}$$
$$\mathbb{P}^\pi(a_t \mid s_t) \qquad \text{(Markov policy)}$$

Definition 7 (Utility)

Given a horizon $T \geq 0$, and discount factor $\gamma \in (0, 1]$ the utility can be defined as

$$U_t \triangleq \sum_{k=0}^{T-t} \gamma^k r_{t+k} \tag{3.2}$$

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

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

State value function

$$V^{\pi}_{\mu,t}(s) \triangleq \mathbb{E}^{\pi}_{\mu}(U_t \mid s_t = s) \tag{3.3}$$

The optimal policy π^*

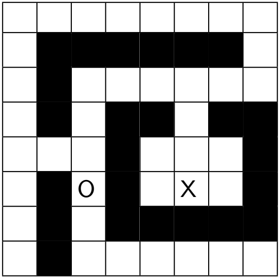
$$\pi^*(\mu) : V^{\pi^*(\mu)}_{t,\mu}(s) \geq V^{\pi}_{t,\mu}(s) \quad \forall \pi, t, s \tag{3.4}$$

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

$$V^*_{t,\mu}(s) \triangleq V^{\pi^*(\mu)}_{t,\mu}(s), \tag{3.5}$$

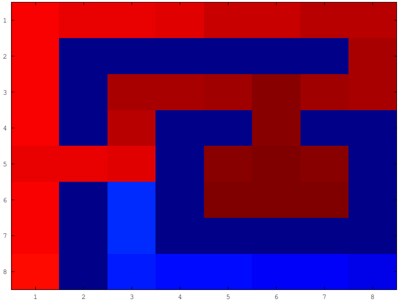
is the value function of the optimal policy π^*.

Stochastic shortest path problem with a pit

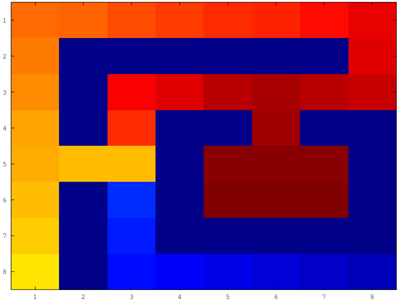


Properties

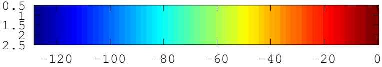
- $T \to \infty$.
- $r_t = -1$, but $r_t = 0$ at X and -100 at O and the problem ends.
- $\mathbb{P}_\mu(s_{t+1} = X | s_t = X) = 1$.
- $\mathcal{A} = \{\mathrm{North}, \mathrm{South}, \mathrm{East}, \mathrm{West}\}$
- Moves to a random direction with probability ω. Walls block.



(a) $\omega = 0.1$



(b) $\omega = 0.5$



(c) value

Figure: Pit maze solutions for two values of ω.

How to evaluate a policy (Case: $\gamma = 1$)

$$V^{\pi}_{\mu,t}(s) \triangleq \mathbb{E}^{\pi}_{\mu}(U_t \mid s_t = s) \tag{3.6}$$

$$\tag{3.7}$$

This derivation directly gives a number of policy evaluation algorithms.

How to evaluate a policy (Case: $\gamma = 1$)

$$V^{\pi}_{\mu,t}(s) \triangleq \mathbb{E}^{\pi}_{\mu}(U_t \mid s_t = s) \tag{3.6}$$

$$= \sum_{k=0}^{T-t} \mathbb{E}^{\pi}_{\mu}(r_{t+k} \mid s_t = s) \tag{3.7}$$

$$\tag{3.8}$$

This derivation directly gives a number of policy evaluation algorithms.

How to evaluate a policy (Case: $\gamma = 1$)

$$
\begin{aligned}
V^{\pi}_{\mu,t}(s) &\triangleq \mathbb{E}^{\pi}_{\mu}(U_t \mid s_t = s) && (3.6)\\
&= \sum_{k=0}^{T-t} \mathbb{E}^{\pi}_{\mu}(r_{t+k} \mid s_t = s) && (3.7)\\
&= \mathbb{E}^{\pi}_{\mu}(r_t \mid s_t = s) + \mathbb{E}^{\pi}_{\mu}(U_{t+1} \mid s_t = s) && (3.8)\\
& && (3.9)
\end{aligned}
$$

This derivation directly gives a number of policy evaluation algorithms.

How to evaluate a policy (Case: $\gamma = 1$)

$$
\begin{aligned}
V^{\pi}_{\mu,t}(s) &\triangleq \mathbb{E}^{\pi}_{\mu}(U_t \mid s_t = s) && (3.6)\\
&= \sum_{k=0}^{T-t} \mathbb{E}^{\pi}_{\mu}(r_{t+k} \mid s_t = s) && (3.7)\\
&= \mathbb{E}^{\pi}_{\mu}(r_t \mid s_t = s) + \mathbb{E}^{\pi}_{\mu}(U_{t+1} \mid s_t = s) && (3.8)\\
&= \mathbb{E}^{\pi}_{\mu}(r_t \mid s_t = s) + \sum_{i \in \mathcal{S}} V^{\pi}_{\mu,t+1}(i)\, \mathbb{P}^{\pi}_{\mu}(s_{t+1} = i | s_t = s). && (3.9)
\end{aligned}
$$

This derivation directly gives a number of policy evaluation algorithms.

How to evaluate a policy (Case: $\gamma = 1$)

$$
\begin{aligned}
V^{\pi}_{\mu,t}(s) &\triangleq \mathbb{E}^{\pi}_{\mu}(U_t \mid s_t = s) && (3.6)\\
&= \sum_{k=0}^{T-t} \mathbb{E}^{\pi}_{\mu}(r_{t+k} \mid s_t = s) && (3.7)\\
&= \mathbb{E}^{\pi}_{\mu}(r_t \mid s_t = s) + \mathbb{E}^{\pi}_{\mu}(U_{t+1} \mid s_t = s) && (3.8)\\
&= \mathbb{E}^{\pi}_{\mu}(r_t \mid s_t = s) + \sum_{i \in \mathcal{S}} V^{\pi}_{\mu,t+1}(i)\, \mathbb{P}^{\pi}_{\mu}(s_{t+1} = i | s_t = s). && (3.9)
\end{aligned}
$$

This derivation directly gives a number of policy evaluation algorithms.

$$\max_{\pi} V^{\pi}_{\mu,t}(s) = \max_{a} \mathbb{E}_{\mu}(r_t \mid s_t = s, a) + \max_{\pi'} \sum_{i \in \mathcal{S}} V^{\pi'}_{\mu,t+1}(i)\, \mathbb{P}^{\pi'}_{\mu}(s_{t+1} = i | s_t = s).$$

gives us the optimal policy value.

Backward induction for discounted infinite horizon problems

- We can also apply backwards induction to the infinite case.
- The resulting policy is stationary.
- So memory does not grow with T.

Value iteration

for $n = 1, 2, \ldots$ and $s \in \mathcal{S}$ **do**
$\quad v_n(s) = \max_a r(s, a) + \gamma \sum_{s' \in \mathcal{S}} P_\mu(s' \mid s, a) v_{n-1}(s')$
end for

Summary

- Markov decision processes model controllable dynamical systems.
- Optimal policies maximise expected utility can be found with:
  - Backwards induction / value iteration.
  - Policy iteration.
- The MDP state can be seen as
  - The state of a dynamic controllable process.
  - The internal state of an agent.

The reinforcement learning problem

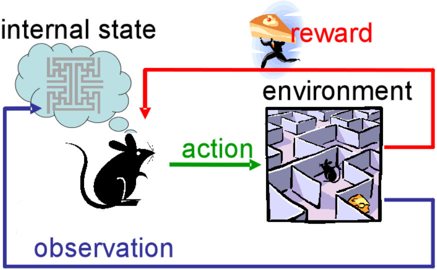
Learning to act in an unknown world, by interaction and reinforcement.

The reinforcement learning problem

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

World μ; Policy π; at time t

- μ generates observation $x_t \in \mathcal{X}$.
- We take action $a_t \in \mathcal{A}$ using π.
- μ gives us reward $r_t \in \mathbb{R}$.



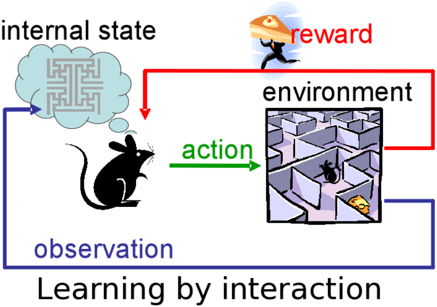
Learning by interaction

The reinforcement learning problem

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

World μ; Policy π; at time t

- μ generates observation $x_t \in \mathcal{X}$.
- We take action $a_t \in \mathcal{A}$ using π.
- μ gives us reward $r_t \in \mathbb{R}$.



Definition 8 (Utility)

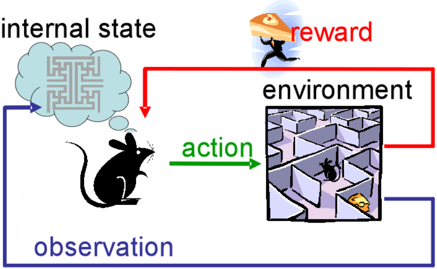
$$U_t = \sum_{k=t}^{T} r_k$$

The reinforcement learning problem

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

World μ; Policy π; at time t

- μ generates observation $x_t \in \mathcal{X}$.
- We take action $a_t \in \mathcal{A}$ using π.
- μ gives us reward $r_t \in \mathbb{R}$.



Learning by interaction

Definition 8 (Expected utility)

$$\mathbb{E}^{\pi}_{\mu} U_t = \mathbb{E}^{\pi}_{\mu} \sum_{k=t}^{T} r_k$$

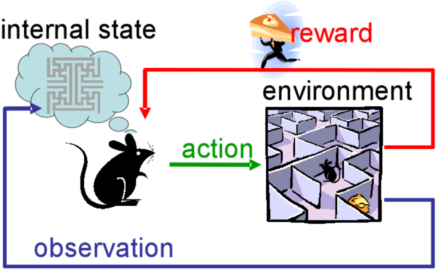
When μ is known, calculate $\max_\pi \mathbb{E}^{\pi}_{\mu} U$.

The reinforcement learning problem

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

World μ; Policy π; at time t

- μ generates observation $x_t \in \mathcal{X}$.
- We take action $a_t \in \mathcal{A}$ using π.
- μ gives us reward $r_t \in \mathbb{R}$.



Learning by interaction

Definition 8 (Expected utility)

$$\mathbb{E}_\mu^\pi U_t = \mathbb{E}_\mu^\pi \sum_{k=t}^{T} r_k$$

Knowing μ is contrary to the problem definition

When μ is not known

Bayesian idea: use a subjective belief $\xi(\mu)$ on $\mathcal{M}$

- Initial belief $\xi(\mu)$.

When μ is not known

Bayesian idea: use a subjective belief $\xi(\mu)$ on $\mathcal{M}$

- Initial belief $\xi(\mu)$.
- The probability of observing history h is $\mathbb{P}^{\pi}_{\mu}(h)$.

When μ is not known

Bayesian idea: use a subjective belief $\xi(\mu)$ on $\mathcal{M}$

- Initial belief $\xi(\mu)$.
- The probability of observing history h is $\mathbb{P}^\pi_\mu(h)$.
- We can use this to adjust our belief via Bayes' theorem:

$$\xi(\mu \mid h, \pi) \propto \mathbb{P}^\pi_\mu(h)\xi(\mu)$$

When μ is not known

Bayesian idea: use a subjective belief $\xi(\mu)$ on $\mathcal{M}$

- Initial belief $\xi(\mu)$.
- The probability of observing history h is $\mathbb{P}^{\pi}_{\mu}(h)$.
- We can use this to adjust our belief via Bayes' theorem:

$$\xi(\mu \mid h, \pi) \propto \mathbb{P}^{\pi}_{\mu}(h)\xi(\mu)$$

- We can thus conclude which μ is more likely.

When μ is not known

Bayesian idea: use a subjective belief $\xi(\mu)$ on $\mathcal{M}$

- Initial belief $\xi(\mu)$.
- The probability of observing history h is $\mathbb{P}^\pi_\mu(h)$.
- We can use this to adjust our belief via Bayes' theorem:

$$\xi(\mu \mid h, \pi) \propto \mathbb{P}^\pi_\mu(h)\xi(\mu)$$

- We can thus conclude which μ is more likely.

The subjective expected utility

$$\mathbb{E}^\pi_\xi U = \sum_\mu \left(\mathbb{E}^\pi_\mu U\right) \xi(\mu).$$

When μ is not known

Bayesian idea: use a subjective belief $\xi(\mu)$ on $\mathcal{M}$

- Initial belief $\xi(\mu)$.
- The probability of observing history h is $\mathbb{P}^\pi_\mu(h)$.
- We can use this to adjust our belief via Bayes' theorem:

$$\xi(\mu \mid h, \pi) \propto \mathbb{P}^\pi_\mu(h) \xi(\mu)$$

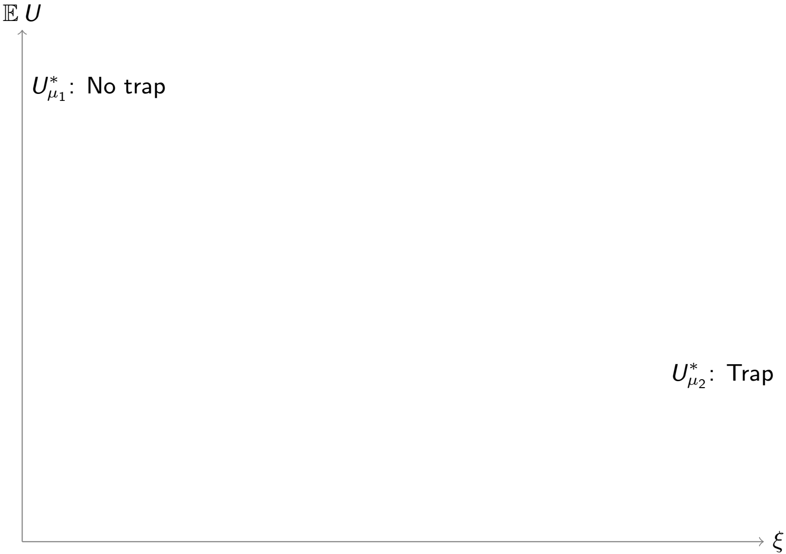
- We can thus conclude which μ is more likely.

The subjective expected utility

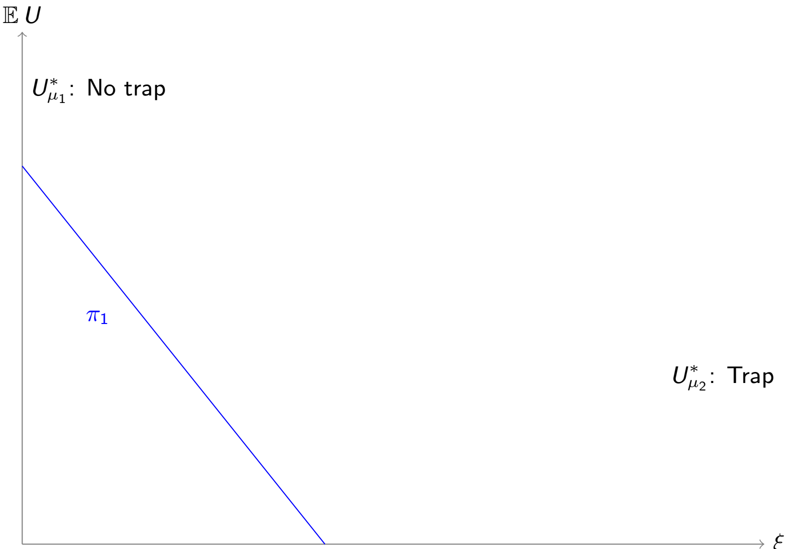
$$U^*_\xi \triangleq \max_\pi \mathbb{E}^\pi_\xi U = \max_\pi \sum_\mu \left(\mathbb{E}^\pi_\mu U\right) \xi(\mu).$$

Integrates planning and learning, and the exploration-exploitation trade-off

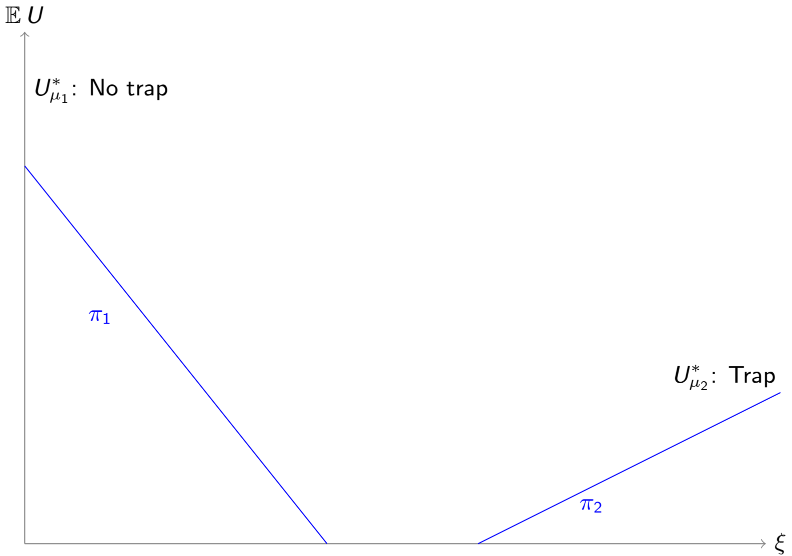
Bounds on the ξ-optimal utility $U^*_\xi \triangleq \max_\pi \mathbb{E}^\pi_\xi U$



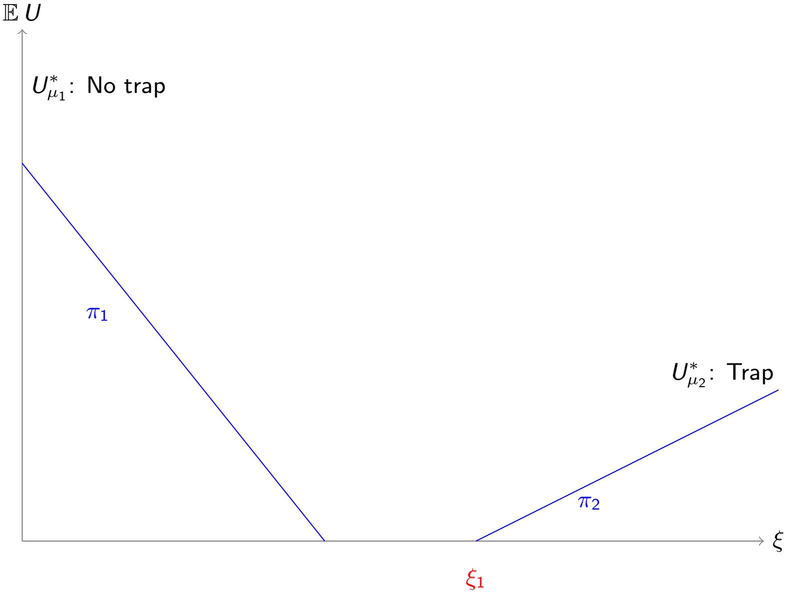
Bounds on the ξ-optimal utility $U^*_\xi \triangleq \max_\pi \mathbb{E}^\pi_\xi U$



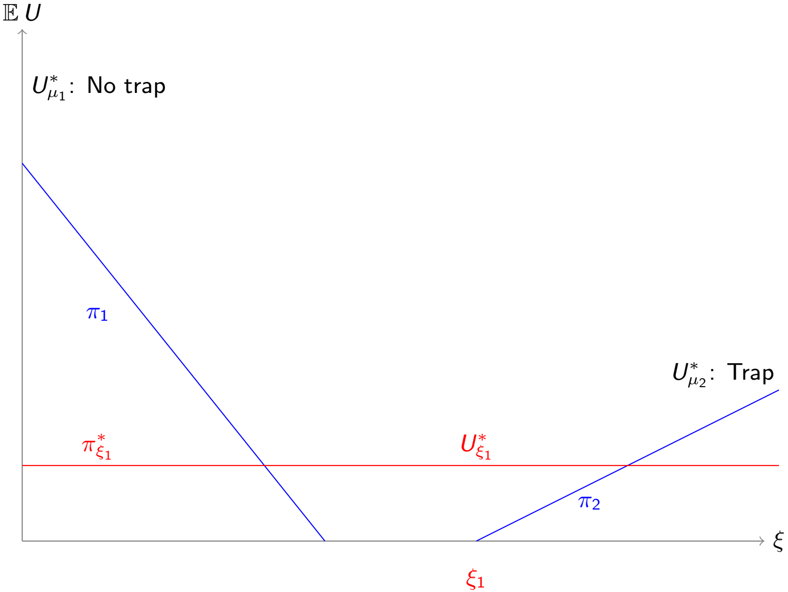
Bounds on the ξ-optimal utility $U^*_\xi \triangleq \max_\pi \mathbb{E}^\pi_\xi U$



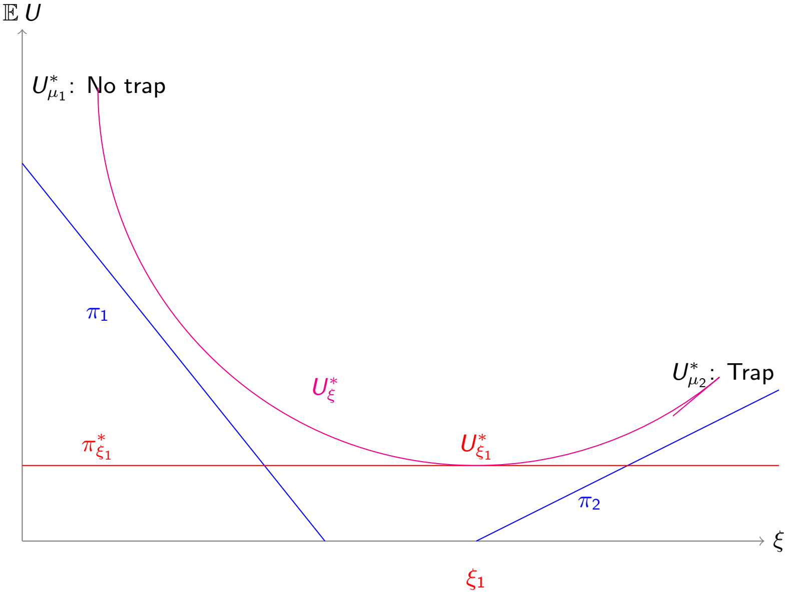
Bounds on the ξ-optimal utility $U^*_\xi \triangleq \max_\pi \mathbb{E}^\pi_\xi U$



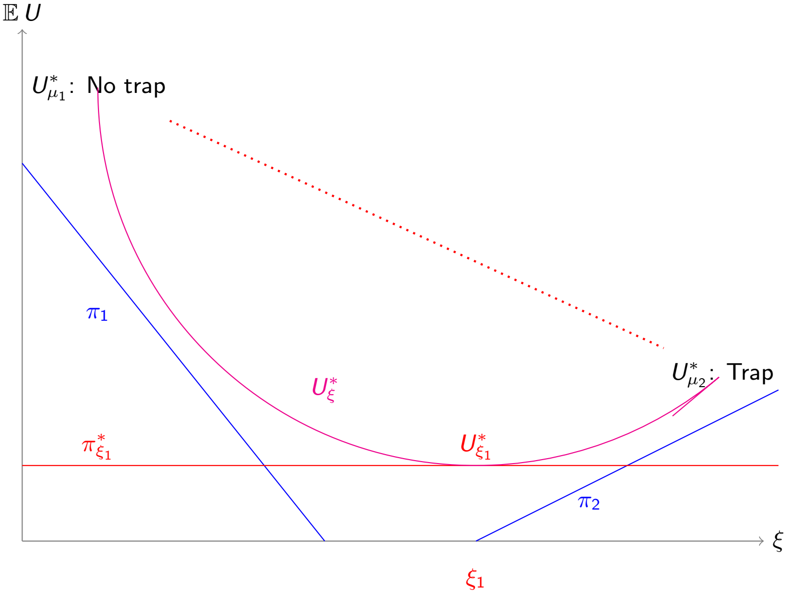
Bounds on the ξ-optimal utility $U^*_\xi \triangleq \max_\pi \mathbb{E}^\pi_\xi U$



Bounds on the ξ-optimal utility $U^*_\xi \triangleq \max_\pi \mathbb{E}^\pi_\xi U$



Bounds on the ξ-optimal utility $U^*_\xi \triangleq \max_\pi \mathbb{E}^\pi_\xi U$



Bernoulli bandits

Decision-theoretic approach

- Assume $r_t \mid a_t = i \sim P_{\omega_i}$, with $\omega_i \in \Omega$.
- Define prior belief ξ_1 on Ω.
- For each step t, select action a_t to maximise

$$\mathbb{E}_{\xi_t}(U_t \mid a_t) = \mathbb{E}_{\xi_t}\left(\sum_{k=1}^{T-t} \gamma^k r_{t+k} \,\middle|\, a_t\right)$$

- Obtain reward r_t.
- Calculate the next belief

$$\xi_{t+1} = \xi_t(\cdot \mid a_t, r_t)$$

How can we implement this?

Bayesian inference on Bernoulli bandits

- Likelihood: $\mathbb{P}_\omega(r_t = 1) = \omega$.
- Prior: $\xi(\omega) \propto \omega^{\alpha-1}(1-\omega)^{\beta-1}$ (i.e. $\mathcal{B}eta(\alpha, \beta)$).

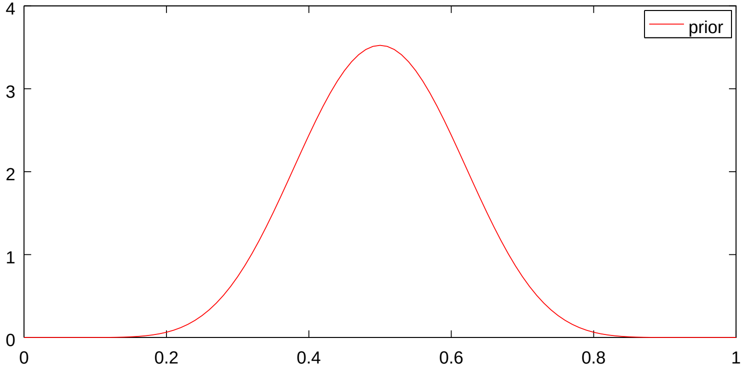


Figure: Prior belief ξ about the mean reward ω.

Bayesian inference on Bernoulli bandits

For a sequence $r = r_1, \ldots, r_n$, $\Rightarrow P_\omega(r) \propto \omega_i^{\#1(\mathrm{r})}(1-\omega_i)^{\#0(\mathrm{r})}$

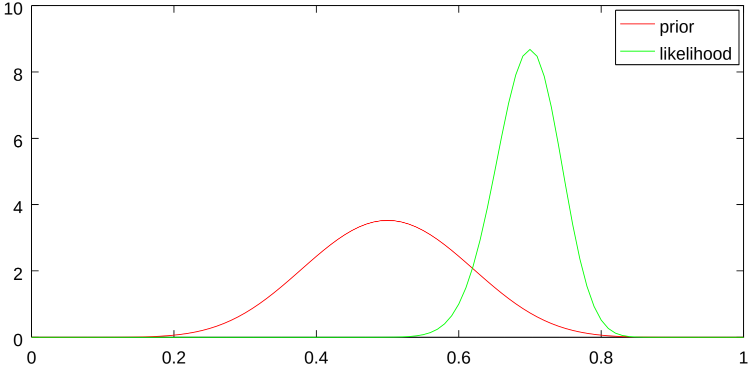


Figure: Prior belief ξ about ω and likelihood of ω for 100 plays with 70 1s.

Bayesian inference on Bernoulli bandits

Posterior: $\mathcal{B}eta(\alpha + \#1(\mathrm{r}), \beta + \#0(\mathrm{r}))$.

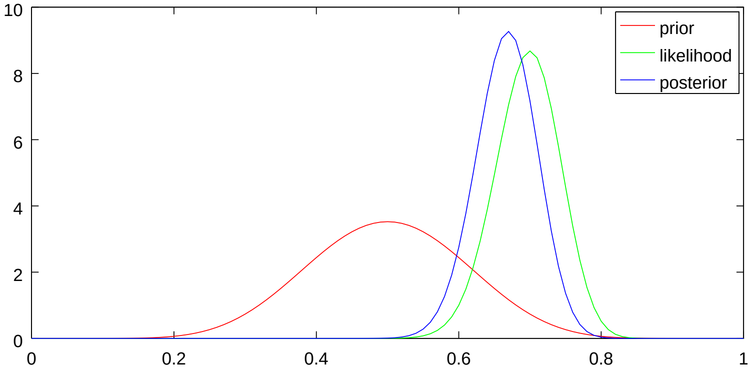


Figure: Prior belief $\xi(\omega)$ about ω, likelihood of ω for the data r, and posterior belief $\xi(\omega \mid r)$

Bernoulli example.

Consider n Bernoulli distributions with unknown parameters ω_i ($i = 1, \ldots, n$) such that

$$r_t \mid a_t = i \sim \mathcal{B}ern(\omega_i), \qquad \mathbb{E}(r_t \mid a_t = i) = \omega_i. \tag{4.1}$$

Our belief for each parameter ω_i is $\mathcal{B}eta(\alpha_i, \beta_i)$, with density $f(\omega \mid \alpha_i, \beta_i)$ so that

$$\xi(\omega_1, \ldots, \omega_n) = \prod_{i=1}^{n} f(\omega_i \mid \alpha_i, \beta_i). \qquad \text{(a priori independent)}$$

$$N_{t,i} \triangleq \sum_{k=1}^{t} \mathbb{I}\{a_k = i\}, \qquad \hat{r}_{t,i} \triangleq \frac{1}{N_{t,i}} \sum_{k=1}^{t} r_t \, \mathbb{I}\{a_k = i\}$$

Then, the posterior distribution for the parameter of arm i is

$$\xi_t = \mathcal{B}eta(\alpha_i + N_{t,i}\hat{r}_{t,i} \ , \ \beta_i + N_{t,i}(1 - \hat{r}_{t,i})).$$

Since $r_t \in \{0, 1\}$ there are $O((2n)^T)$ possible belief states for a T-step bandit problem.

Belief states

- The state of the decision-theoretic bandit problem is the state of our belief.
- A sufficient statistic is the number of plays and total rewards.
- Our belief state ξ_t is described by the priors α, β and the vectors

$$N_t = (N_{t,1}, \ldots, N_{t,i}) \tag{4.2}$$
$$\hat{r}_t = (\hat{r}_{t,1}, \ldots, \hat{r}_{t,i}). \tag{4.3}$$

- The next-state probabilities are defined as:

$$\mathbb{P}(r_t = 1 \mid a_t = i, \xi_t) = \frac{\alpha_i + N_{t,i}\hat{r}_{t,i}}{\alpha_i + \beta_i + N_{t,i}}$$

  as ξ_{t+1} is a deterministic function of ξ_t, r_t and a_t
- So the bandit problem can be formalised as a Markov decision process.

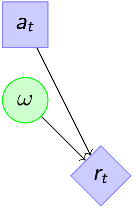


Figure: The basic bandit MDP. The decision maker selects a_t, while the parameter ω of the process is hidden. It then obtains reward r_t. The process repeats for $t = 1, \ldots, T$.

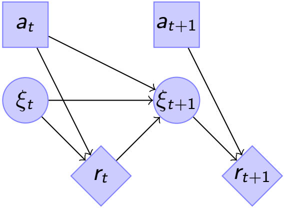


Figure: The decision-theoretic bandit MDP. While ω is not known, at each time step t we maintain a belief ξ_t on Ω. The reward distribution is then defined through our belief.

Backwards induction (Dynamic programming)

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

$$\mathbb{E}(U_t \mid \xi_t) = \max_{a_t \in \mathcal{A}} \mathbb{E}(r_t \mid \xi_t, a_t) + \gamma \sum_{\xi_{t+1}} \mathbb{P}(\xi_{t+1} \mid \xi_t, a_t) \, \mathbb{E}(U_{t+1} \mid \xi_{t+1})$$

end for

Backwards induction (Dynamic programming)

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

$$\mathbb{E}(U_t \mid \xi_t) = \max_{a_t \in \mathcal{A}} \mathbb{E}(r_t \mid \xi_t, a_t) + \gamma \sum_{\xi_{t+1}} \mathbb{P}(\xi_{t+1} \mid \xi_t, a_t) \, \mathbb{E}(U_{t+1} \mid \xi_{t+1})$$

end for

Exact solution methods: exponential in the horizon

- Dynamic programming (backwards induction etc)
- Policy search.

Approximations

- (Stochastic) branch and bound.
- Upper confidence trees.
- Approximate dynamic programming.
- Local policy search (e.g. gradient based)

Bayesian RL for unknown MDPs

The MDP as an environment.

We are in some environment μ, where at each time, we: step t:

- Observe state $s_t \in \mathcal{S}$.
- Take action $a_t \in \mathcal{A}$.
- Receive reward $r_t \in \mathbb{R}$.

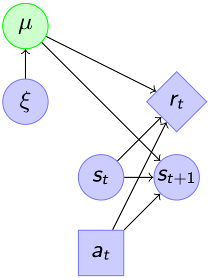


Figure: The unknown Markov decision process

How can we find the Bayes optimal policy for unknown MDPs?

Some heuristics

1. Only change policy at the start of epochs t_i.
2. Calculate the belief ξ_{t_i}.
3. Find a "good" policy π_i for the current belief.
4. Execute it until the next epoch $i + 1$.

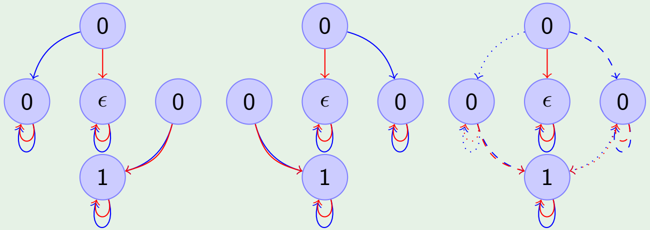
One simple heuristic is to simply calculate the expected MDP for a given belief ξ:

$$\widehat{\mu}_\xi \triangleq \mathbb{E}_\xi \, \mu.$$

Then, we simply calculate the optimal policy for $\widehat{\mu}_\xi$:

$$\pi^*(\widehat{\mu}_\xi) \in \arg\max_{\pi \in \Pi_1} V^\pi_{\widehat{\mu}_\xi},$$

Example 9 (Counterexample)



(a) μ_1 (b) μ_2 (c) $\widehat{\mu}_\xi$

Another heuristic is to get the most probable MDP for a belief ξ:

$$\widehat{\mu}^*_\xi \triangleq \arg\max_\mu \xi(\mu)$$

Then, we simply calculate the optimal policy for $\widehat{\mu}^*_\xi$:

$$\pi^*(\widehat{\mu}_\xi) \in \arg\max_{\pi \in \Pi_1} V^\pi_{\widehat{\mu}_\xi},$$

Example 10

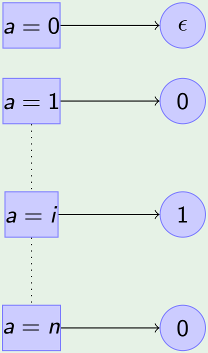


Figure: The MDP μ_i from $|\mathcal{A}| + 1$ MDPs.

Posterior (Thompson) sampling

Another heuristic is to simply sample an MDP from the belief ξ:

$$\mu^{(k)} \sim \xi(\mu)$$

Then, we simply calculate the optimal policy for $\mu^{(k)}$:

$$\pi^*(\widehat{\mu}_\xi) \in \arg\max_{\pi \in \Pi_1} V^\pi_{\mu^{(k)}},$$

Properties

- $\sqrt{T}$ regret. (Direct proof: hard [1]. Easy proof: convert to confidence bound [11])
- Generally applicable for many beliefs.
- Connections to differential privacy [9].
- Generalises to stochastic value function bounds [8].

Belief-Augmented MDPs

- Unknown bandit problems can be converted into MDPs through the belief state.
- We can do the same for MDPs. We just create a hyperstate, composed of the current belief and the current belief state.

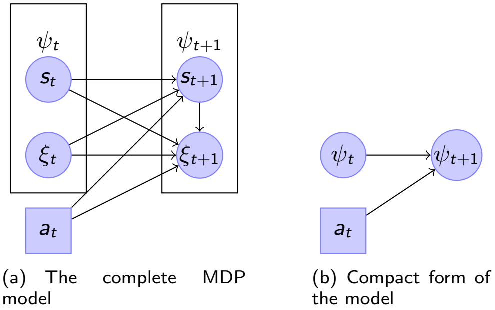


Figure: Belief-augmented MDP

The augmented MDP

$$P(s_{t+1} \in S \mid \xi_t, s_t, a_t) \triangleq \int_S P_\mu(s_{t+1} \in S \mid s_t, a_t) \, \mathrm{d}\xi_t(\mu) \tag{4.4}$$

$$\xi_{t+1}(\cdot) = \xi_t(\cdot \mid s_{t+1}, s_t, a_t) \tag{4.5}$$

- So now we have converted the unknown MDP problem into an MDP.

- So now we have converted the unknown MDP problem into an MDP.
- That means we can use dynamic programming to solve it.

- So now we have converted the unknown MDP problem into an MDP.
- That means we can use dynamic programming to solve it.
- So... are we done?

- So now we have converted the unknown MDP problem into an MDP.
- That means we can use dynamic programming to solve it.
- So... are we done?
- Unfortunately the exact solution is again exponential in the horizon.

ABC (Approximate Bayesian Computation) RL[1]

[1] Dimitrakakis, Tziortiotis. ABC Reinforcement Learning: ICML 2013

ABC (Approximate Bayesian Computation) RL[1]

How to deal with an arbitrary model space $\mathcal{M}$

- The models $\mu \in \mathcal{M}$ may be non-probabilistic simulators.
- We may not know how to choose the simulator parameters.

[1] Dimitrakakis, Tziortiotis. ABC Reinforcement Learning: ICML 2013

ABC (Approximate Bayesian Computation) RL[1]

How to deal with an arbitrary model space $\mathcal{M}$

- The models $\mu \in \mathcal{M}$ may be non-probabilistic simulators.
- We may not know how to choose the simulator parameters.

Overview of the approach

- Place a prior on the simulator parameters.
- Observe some data h on the real system.
- Approximate the posterior by statistics on simulated data.
- Calculate a near-optimal policy for the posterior.

[1] Dimitrakakis, Tziortiotis. ABC Reinforcement Learning: ICML 2013

ABC (Approximate Bayesian Computation) RL[1]

How to deal with an arbitrary model space $\mathcal{M}$

- The models $\mu \in \mathcal{M}$ may be non-probabilistic simulators.
- We may not know how to choose the simulator parameters.

Overview of the approach

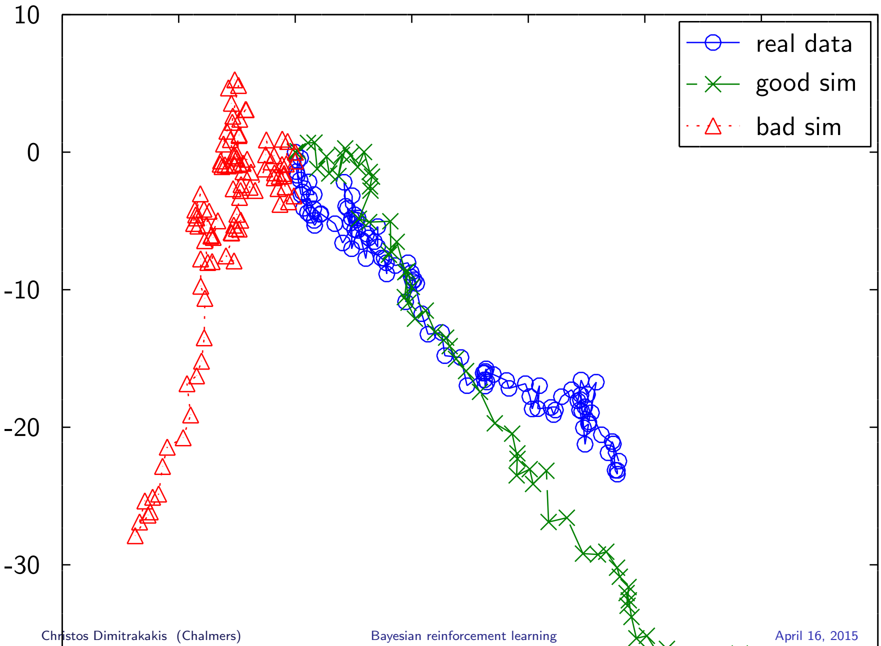
- Place a prior on the simulator parameters.
- Observe some data h on the real system.
- Approximate the posterior by statistics on simulated data.
- Calculate a near-optimal policy for the posterior.

Results

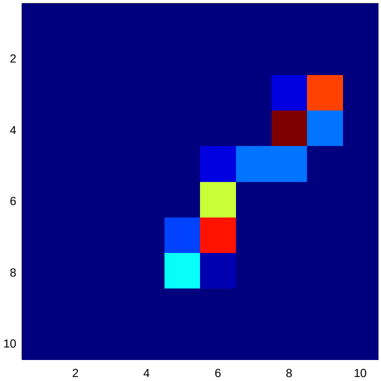
- Soundness depends on properties of the statistics.
- In practice, can require much less data than a general model.

[1] Dimitrakakis, Tziortiotis. ABC Reinforcement Learning: ICML 2013

A set of trajectories



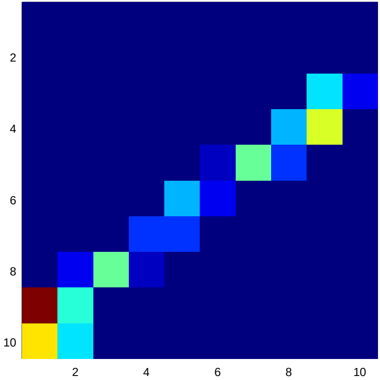
A set of trajectories



- Trajectories are easy to generate.
- How to compare?
- Use a *statistic*.

Cumulative features of real data

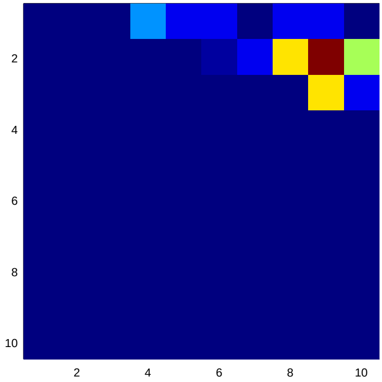
A set of trajectories



- Trajectories are easy to generate.
- How to compare?
- Use a *statistic*.

Cumulative features of good sim

A set of trajectories



Cumulative features of bad sim

- Trajectories are easy to generate.
- How to compare?
- Use a *statistic*.

ABC (Approximate Bayesian Computation)

When there is no probabilistic model ($\mathbb{P}_\mu$ is not available): ABC!

- A prior ξ on a class of simulators $\mathcal{M}$
- History $h \in \mathcal{H}$ from policy π.
- Statistic $f : \mathcal{H} \to (\mathcal{W}, \|\cdot\|)$
- Threshold $\epsilon > 0$.

ABC (Approximate Bayesian Computation)

When there is no probabilistic model ($\mathbb{P}_\mu$ is not available): ABC!

- A prior ξ on a class of simulators $\mathcal{M}$
- History $h \in \mathcal{H}$ from policy π.
- Statistic $f : \mathcal{H} \to (\mathcal{W}, \| \cdot \|)$
- Threshold $\epsilon > 0$.

ABC (Approximate Bayesian Computation)

When there is no probabilistic model ($\mathbb{P}_\mu$ is not available): ABC!

- A prior ξ on a class of simulators $\mathcal{M}$
- History $h \in \mathcal{H}$ from policy π.
- Statistic $f : \mathcal{H} \to (\mathcal{W}, \|\cdot\|)$
- Threshold $\epsilon > 0$.

Example 11 (Cumulative features)

Feature function $\phi : \mathcal{X} \to \mathbb{R}^k$.

$$f(h) \triangleq \sum_t \phi(x_t)$$

ABC (Approximate Bayesian Computation)

When there is no probabilistic model ($\mathbb{P}_\mu$ is not available): ABC!

- A prior ξ on a class of simulators $\mathcal{M}$
- History $h \in \mathcal{H}$ from policy π.
- Statistic $f : \mathcal{H} \to (\mathcal{W}, \|\cdot\|)$
- Threshold $\epsilon > 0$.

Example 11 (Utility)

$$f(h) \triangleq \sum_t r_t$$

ABC (Approximate Bayesian Computation)

When there is no probabilistic model ($\mathbb{P}_\mu$ is not available): ABC!

- A prior ξ on a class of simulators $\mathcal{M}$
- History $h \in \mathcal{H}$ from policy π.
- Statistic $f : \mathcal{H} \to (\mathcal{W}, \|\cdot\|)$
- Threshold $\epsilon > 0$.

ABC (Approximate Bayesian Computation)

When there is no probabilistic model ($\mathbb{P}_\mu$ is not available): ABC!

- A prior ξ on a class of simulators $\mathcal{M}$
- History $h \in \mathcal{H}$ from policy π.
- Statistic $f : \mathcal{H} \to (\mathcal{W}, \|\cdot\|)$
- Threshold $\epsilon > 0$.

ABC-RL using Thompson sampling

- **do** $\hat{\mu} \sim \xi$, $h' \sim \mathbb{P}^\pi_{\hat{\mu}}$ // sample a model and history
- **until** $\|f(h') - f(h)\| \le \epsilon$ // until the statistics are close
- $\mu^{(k)} = \hat{\mu}$ // approximate posterior sample $\mu^{(k)} \sim \xi_\epsilon(\cdot \mid h_t)$
- $\pi^{(k)} \approx \arg\max \mathbb{E}^\pi_{\mu^{(k)}} U_t$ // approximate optimal policy for sample

ABC (Approximate Bayesian Computation)

When there is no probabilistic model ($\mathbb{P}_\mu$ is not available): ABC!

- A prior ξ on a class of simulators $\mathcal{M}$
- History $h \in \mathcal{H}$ from policy π.
- Statistic $f : \mathcal{H} \to (\mathcal{W}, \|\cdot\|)$
- Threshold $\epsilon > 0$.

ABC-RL using Thompson sampling

- **do** $\hat{\mu} \sim \xi$, $h' \sim \mathbb{P}^\pi_{\hat{\mu}}$ // sample a model and history
- **until** $\|f(h') - f(h)\| \leq \epsilon$ // until the statistics are close
- $\mu^{(k)} = \hat{\mu}$ // approximate posterior sample $\mu^{(k)} \sim \xi_\epsilon(\cdot \mid h_t)$
- $\pi^{(k)} \approx \arg\max \mathbb{E}^\pi_{\mu^{(k)}} U_t$ // approximate optimal policy for sample

ABC (Approximate Bayesian Computation)

When there is no probabilistic model ($\mathbb{P}_\mu$ is not available): ABC!

- A prior ξ on a class of simulators $\mathcal{M}$
- History $h \in \mathcal{H}$ from policy π.
- Statistic $f : \mathcal{H} \to (\mathcal{W}, \| \cdot \|)$
- Threshold $\epsilon > 0$.

ABC-RL using Thompson sampling

- **do** $\hat{\mu} \sim \xi$, $h' \sim \mathbb{P}^\pi_{\hat{\mu}}$ // sample a model and history
- **until** $\|f(h') - f(h)\| \leq \epsilon$ // until the statistics are close
- $\mu^{(k)} = \hat{\mu}$ // approximate posterior sample $\mu^{(k)} \sim \xi_\epsilon(\cdot \mid h_t)$
- $\pi^{(k)} \approx \arg\max \mathbb{E}^\pi_{\mu^{(k)}} U_t$ // approximate optimal policy for sample

ABC (Approximate Bayesian Computation)

When there is no probabilistic model ($\mathbb{P}_\mu$ is not available): ABC!

- A prior ξ on a class of simulators $\mathcal{M}$
- History $h \in \mathcal{H}$ from policy π.
- Statistic $f : \mathcal{H} \to (\mathcal{W}, \|\cdot\|)$
- Threshold $\epsilon > 0$.

ABC-RL using Thompson sampling

- **do** $\hat{\mu} \sim \xi$, $h' \sim \mathbb{P}^\pi_{\hat{\mu}}$ // sample a model and history
- **until** $\|f(h') - f(h)\| \leq \epsilon$ // until the statistics are close
- $\mu^{(k)} = \hat{\mu}$ // approximate posterior sample $\mu^{(k)} \sim \xi_\epsilon(\cdot \mid h_t)$
- $\pi^{(k)} \approx \arg\max \mathbb{E}^\pi_{\mu^{(k)}} U_t$ // approximate optimal policy for sample

The approximate posterior $\xi_\epsilon(\cdot \mid h)$

Corollary 11

If f is a sufficient statistic and $\epsilon = 0$, then $\xi(\cdot \mid h) = \xi_\epsilon(\cdot \mid h)$.

The approximate posterior $\xi_\epsilon(\cdot \mid h)$

Corollary 11

If f is a sufficient statistic and $\epsilon = 0$, then $\xi(\cdot \mid h) = \xi_\epsilon(\cdot \mid h)$.

Assumption 2 (A1. Lipschitz log-probabilities)

For the policy π, $\exists L > 0$ s.t. $\forall h, h' \in \mathcal{H}$ and $\forall \mu \in \mathcal{M}$

$$\left| \ln \left[\mathbb{P}^{\pi}_{\mu}(h) / \mathbb{P}^{\pi}_{\mu}(h') \right] \right| \leq L \| f(h) - f(h') \|$$

The approximate posterior $\xi_\epsilon(\cdot \mid h)$

Corollary 11

If f is a sufficient statistic and $\epsilon = 0$, then $\xi(\cdot \mid h) = \xi_\epsilon(\cdot \mid h)$.

Assumption 2 (A1. Lipschitz log-probabilities)

For the policy π, $\exists L > 0$ s.t. $\forall h, h' \in \mathcal{H}$ and $\forall \mu \in \mathcal{M}$

$$\left|\ln\left[\mathbb{P}^\pi_\mu(h) / \mathbb{P}^\pi_\mu(h')\right]\right| \leq L \|f(h) - f(h')\|$$

Theorem 12 (The approximate posterior $\xi_\epsilon(\cdot \mid h)$ is close to $\xi(\cdot \mid h)$)

If A1 holds then $\forall \epsilon > 0$:

$$D\left(\xi(\cdot \mid h) \,\|\, \xi_\epsilon(\cdot \mid h)\right) \leq 2L\epsilon + \ln |A^h_\epsilon|, \tag{4.6}$$

where $A^h_\epsilon \triangleq \{z \in \mathcal{H} \mid \|f(z) - f(h)\| \leq \epsilon\}$.

The approximate posterior $\xi_\epsilon(\cdot \mid h)$

Corollary 11

If f is a sufficient statistic and $\epsilon = 0$, then $\xi(\cdot \mid h) = \xi_\epsilon(\cdot \mid h)$.

Assumption 2 (A1. Lipschitz log-probabilities)

For the policy π, $\exists L > 0$ s.t. $\forall h, h' \in \mathcal{H}$ and $\forall \mu \in \mathcal{M}$

$$\left|\ln\left[\mathbb{P}^\pi_\mu(h) / \mathbb{P}^\pi_\mu(h')\right]\right| \leq L \|f(h) - f(h')\|$$

Theorem 12 (The approximate posterior $\xi_\epsilon(\cdot \mid h)$ is close to $\xi(\cdot \mid h)$)

If A1 holds then $\forall \epsilon > 0$:

$$D\left(\xi(\cdot \mid h) \parallel \xi_\epsilon(\cdot \mid h)\right) \leq 2L\epsilon + \ln |A^h_\epsilon|, \tag{4.6}$$

where $A^h_\epsilon \triangleq \{z \in \mathcal{H} \mid \|f(z) - f(h)\| \leq \epsilon\}$.

Summary

- Unknown MDPs can be handled in a Bayesian framework.
- This defines a belief-augmented MDP with
  - A state for the MDP.
  - A state for the agent's belief.
- The Bayes-optimal utility is convex, enabling approximations.
- A big problem in specifying the “right” prior.

Questions?

Belief updates

Discounted reward MDPs

Backwards induction

Updating the belief in discrete MDPs

Let $D_t = \left\langle s^t, a^{t-1}, r^{t-1} \right\rangle$ be the observed data to time t. Then

$$\xi(B \mid D_t, \pi) = \frac{\int_B \mathbb{P}^\pi_\mu(D_t) \,\mathrm{d}\xi(\mu)}{\int_{\mathcal{M}} \mathbb{P}^\pi_\mu(D_t) \,\mathrm{d}\xi(\mu)}. \tag{5.1}$$

$$\begin{aligned}
\xi_{t+1}(B) \triangleq \xi(B \mid D_{t+1}) &= \frac{\int_B \mathbb{P}^\pi_\mu(D_t) \,\mathrm{d}\xi(\mu)}{\int_{\mathcal{M}} \mathbb{P}^\pi_\mu(D_t) \,\mathrm{d}\xi(\mu)} && (5.2)\\
&= \frac{\int_B \mathbb{P}_\mu(s_{t+1}, r_t \mid s_t, a_t)\pi(a_t \mid s^t, a^{t-1}, r^{t-1}) \,\mathrm{d}\xi(\mu \mid D_t)}{\int_{\mathcal{M}} \mathbb{P}_\mu(s_{t+1}, r_t \mid s_t, a_t)\pi(a_t \mid s^t, a^{t-1}, r^{t-1}) \,\mathrm{d}\xi(\mu \mid D_t)} && (5.3)\\
&= \frac{\int_B \mathbb{P}_\mu(s_{t+1}, r_t \mid s_t, a_t) \,\mathrm{d}\xi_t(\mu)}{\int_{\mathcal{M}} \mathbb{P}_\mu(s_{t+1}, r_t \mid s_t, a_t) \,\mathrm{d}\xi_t(\mu)} && (5.4)
\end{aligned}$$

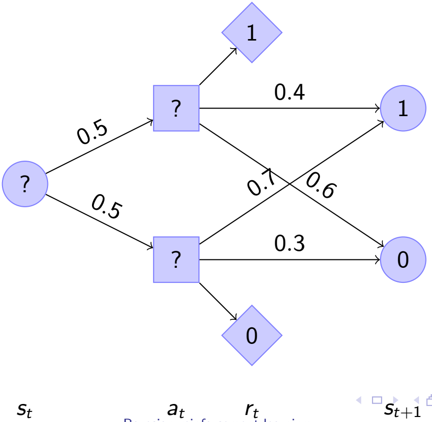
Backwards induction policy evaluation

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

Update values

$$v_t(s) = \mathbb{E}^{\pi}_{\mu}(r_t \mid s_t = s) + \sum_{j \in S} \mathbb{P}^{\pi}_{\mu}(s_{t+1} = j \mid s_t = s) v_{t+1}(j), \tag{5.5}$$

end for



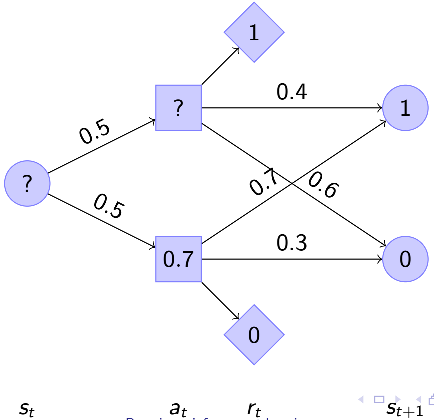
Backwards induction policy evaluation

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

Update values

$$v_t(s) = \mathbb{E}^{\pi}_{\mu}(r_t \mid s_t = s) + \sum_{j \in S} \mathbb{P}^{\pi}_{\mu}(s_{t+1} = j \mid s_t = s) v_{t+1}(j), \tag{5.5}$$

end for

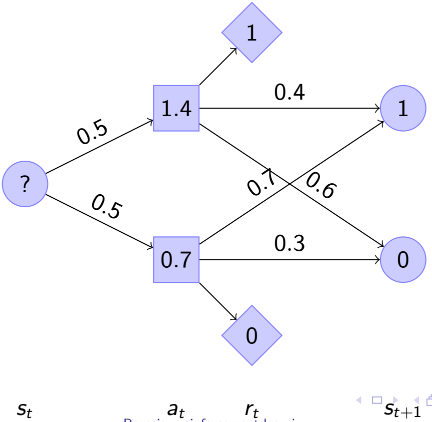


Backwards induction policy evaluation

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

$$v_t(s) = \mathbb{E}^\pi_\mu(r_t \mid s_t = s) + \sum_{j \in S} \mathbb{P}^\pi_\mu(s_{t+1} = j \mid s_t = s) v_{t+1}(j), \tag{5.5}$$

end for



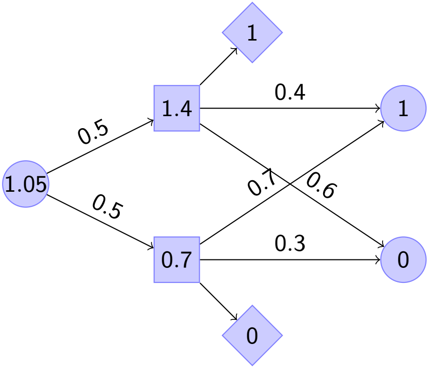
Backwards induction policy evaluation

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

Update values

$$v_t(s) = \mathbb{E}^{\pi}_{\mu}(r_t \mid s_t = s) + \sum_{j \in S} \mathbb{P}^{\pi}_{\mu}(s_{t+1} = j \mid s_t = s) v_{t+1}(j), \tag{5.5}$$

end for



Belief updates

Discounted reward MDPs

Backwards induction

Discounted total reward.

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

Definition 13

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 utility vector as

$$\boldsymbol{v}^{\pi} = \sum_{t=0}^{\infty} \gamma^t \boldsymbol{P}_{\mu,\pi}^t \boldsymbol{r} \tag{6.1}$$

Theorem 14

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

$$\boldsymbol{v} = \boldsymbol{r} + \gamma \boldsymbol{P}_{\mu,\pi} \boldsymbol{v}. \quad \leftarrow \textit{fixed point} \tag{6.2}$$

In addition, the solution is:

$$\boldsymbol{v}^\pi = (\boldsymbol{I} - \gamma \boldsymbol{P}_{\mu,\pi})^{-1} \boldsymbol{r}. \tag{6.3}$$

Example 15

Similar to the geometric series:

$$\sum_{t=0}^{\infty} \alpha^t = \frac{1}{1-\alpha}$$

Policy iteration

Algorithm 1 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`
  $\boldsymbol{v}_{n+1} = (\boldsymbol{I} - \gamma \boldsymbol{P}_{\mu,\pi_{n+1}})^{-1} \boldsymbol{r}$ `// policy evaluation`
  break if $\pi_{n+1} = \pi_n$.
end for
Return $\pi_n, \boldsymbol{v}_n$.

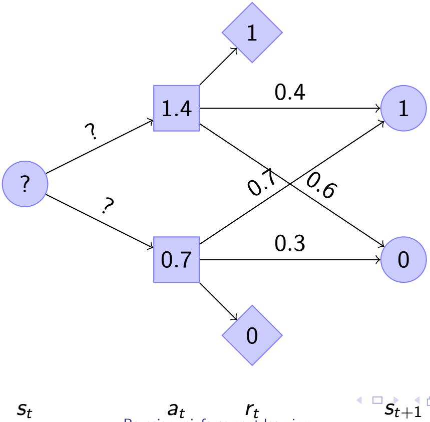
Backwards induction policy optimization

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

Update values

$$v_t(s) = \max_a \mathbb{E}_\mu(r_t \mid s_t = s, a_t = a) + \sum_{j \in S} \mathbb{P}_\mu(s_{t+1} = j \mid s_t = s, a_t = a) v_{t+1}(j), \quad (6.4)$$

end for



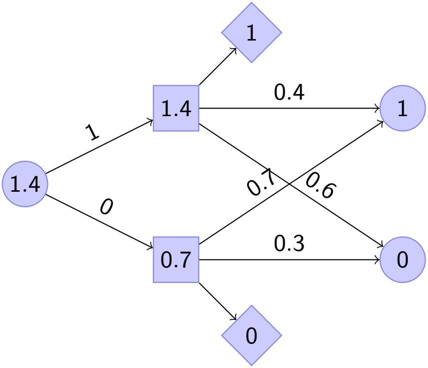
Backwards induction policy optimization

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

Update values

$$v_t(s) = \max_a \mathbb{E}_\mu(r_t \mid s_t = s, a_t = a) + \sum_{j \in S} \mathbb{P}_\mu(s_{t+1} = j \mid s_t = s, a_t = a) v_{t+1}(j), \quad (6.4)$$

end for



[1] Shipra Agrawal and Navi Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *COLT 2012*, 2012.

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

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

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

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

[6] Herman Chernoff. Sequential Models for Clinical Trials. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol.4*, pages 805–812. Univ. of Calif Press, 1966.

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

[8] Christos Dimitrakakis. Monte-carlo utility estimates for bayesian reinforcement learning. In *IEEE 52nd Annual Conference on Decision and Control (CDC 2013)*, 2013. arXiv:1303.2506.

[9] Christos Dimitrakakis, Blaine Nelson, Aikaterini Mitrokotsa, and Benjamin Rubinstein. Robust and private Bayesian inference. In *Algorithmic Learning Theory*, 2014.

[10] Milton Friedman and Leonard J. Savage. The expected-utility hypothesis and the measurability of utility. *The Journal of Political Economy*, 60(6):463, 1952.

[11] Emilie Kaufmanna, Nathaniel Korda, and Rémi Munos. Thompson sampling: An optimal finite time analysis. In *ALT-2012*, 2012.

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

[13] Leonard J. Savage. *The Foundations of Statistics*. Dover Publications, 1972.

[14] 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.

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