Bayesian reinforcement learning

Decision theory, Markov decision processes and experiment design

Christos Dimitrakakis

4 Sep 2013

Introduction

- Experiment design: examples
- Bandit problems
- Markov decision processes (MDP)

Algorithms for known MDPs

- Evaluating a policy
- Finding the optimal policy
- Some examples

Dealing with unknown MDPs

- Stochastic approximation view
- Decision theoretic view
- Multi-armed bandits
- Belief-augmented MDPs

Example (Clinical testing)

- We have a number of treatments of unknown efficacy.
- When a new patient arrives, we must choose one of them.
- There are two possible, slightly different, goals:
  1. Maximise the number of cured patients.
  2. Discover the best treatment.
- The optimal design is better than randomly assigning patients to treatments.

Example (Maze problem)

- You are given the layout to a maze, containing monsters, traps and treasure.
- Given complete knowledge about the problem, what is the optimal policy?
- What if there is imperfect information?

Experimental design and Markov decision processes

The following problems

- Shortest path problems.
- Optimal stopping problems.
- Reinforcement learning problems.
- Experiment design problems.
- Multi-armed bandit problems.
- Advertising.

can be all formalised as Markov decision processes.

The stochastic multi-armed bandit problem

- Actions $\mathcal{A} = \{1, \ldots, n\}$.
- Each time you take action i you receive a reward $r_t \sim P_i$
- Expected reward $\mathbb{E}(r_t \mid a_t = i) = \mathbb{E}_{P_i} r_t = \theta_i$.
- Select actions to maximise

$$\sum_{t=1}^{T} r_t,$$

horizon $T > 0$.

The stochastic multi-armed bandit problem

- Actions $\mathcal{A} = \{1, \ldots, n\}$.
- Each time you take action i you receive a reward $r_t \sim P_i$
- Expected reward $\mathbb{E}(r_t \mid a_t = i) = \mathbb{E}_{P_i} r_t = \theta_i$.
- Select actions to maximise

$$\sum_{t=1}^{T} r_t,$$

horizon $T > 0$.

What to do when $\boldsymbol{\theta}$ is unknown

- Heuristics.
- Decision-theoretic approaches.

The *n*-armed bandit problem

Algorithm 1: *Q*-learning

- Let $\boldsymbol{q}_t \in \mathbb{R}^n$ be a point estimate at time t:

$$q_{t,i} \triangleq \frac{1}{n_{t,i}} \sum_{k=1}^{t} r_t \, \mathbb{I}\left\{a_t = i\right\},$$

  where $n_{t,i}$ is the number of times arm i has been pulled.
- If we pull each arm infinitely often, $\boldsymbol{q}_t \to \boldsymbol{\theta}$.

However this does not address performance when $t < \infty$.

The *n*-armed bandit problem

Algorithm 1: *Q*-learning

- Let $\boldsymbol{q}_t \in \mathbb{R}^n$ be a point estimate at time t:

$$q_{t,i} \triangleq \frac{1}{n_{t,i}} \sum_{k=1}^{t} r_t \, \mathbb{I}\left\{a_t = i\right\},$$

  where $n_{t,i}$ is the number of times arm i has been pulled.

- If we pull each arm infinitely often, $\boldsymbol{q}_t \to \boldsymbol{\theta}$.

However this does not address performance when $t < \infty$.

Reinforcement learning

The reinforcement learning problem

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

- The environment has a changing state.
- The environment generates observations.
- The agent takes actions based on the observed history.
- The agent receives rewards.

The goal (informally)

Maximise total reward during the agent's lifetime.

Types of environments

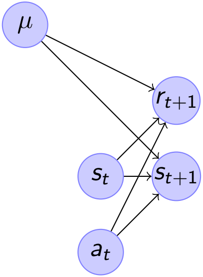
- Markov decision processes (MDPs).
- Partially observable MDPs (POMDPs).
- (Partially observable) (stochastic) Markov games.

Markov decision processes

Markov decision processes (MDP)

We are in some environment μ, where 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

$$\mathbb{P}_\mu(s_{t+1} \in S \mid s_t, a_t) = \mathbb{P}_\mu(s_{t+1} \in S \mid s_1, a_1, \ldots, s_t, a_t) \quad \text{(Transition distribution)}$$
$$\mathbb{P}_\mu(r_{t+1} \in R \mid s_t, a_t) = \mathbb{P}_\mu(r_{t+1} \in R \mid s_1, a_1, \ldots, s_t, a_t), \quad \text{(Reward distribution)}$$

Markov decision processes (MDPs)

The agent's policy π

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

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

Definition (Utility)

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

We wish to find π maximising the expected total future reward

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

to the horizon T .

Markov decision processes (MDPs)

The agent's policy π

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

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

Definition (Utility)

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

We wish to find π maximising the expected total future reward

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

to the horizon T with discount factor $\gamma \in (0, 1]$.

Introduction

Algorithms for known MDPs

- Evaluating a policy
- Finding the optimal policy
- Some examples

Dealing with unknown MDPs

Policy evaluation

An optimal policy

An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. – Bellman.

The value function of a policy π (for $\gamma = 1$, $T < \infty$)

$$V^{\pi}_{\mu,t}(s) \triangleq \mathbb{E}_{\pi,\mu}(U_t \mid s_t = s) \tag{2.1}$$

(2.2)

Directly gives policy evaluation algorithms.

Policy evaluation

The value function of a policy π (for $\gamma = 1$, $T < \infty$)

$$V^{\pi}_{\mu,t}(s) \triangleq \mathbb{E}_{\pi,\mu}(U_t \mid s_t = s) \tag{2.1}$$

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

$$\tag{2.2}$$

Directly gives policy evaluation algorithms.

Policy evaluation

The value function of a policy π (for $\gamma = 1$, $T < \infty$)

$$
\begin{aligned}
V^{\pi}_{\mu,t}(s) &\triangleq \mathbb{E}_{\pi,\mu}(U_t \mid s_t = s) && (2.1)\\
&= \sum_{k=0}^{T-t} \mathbb{E}_{\pi,\mu}(r_{t+k} \mid s_t = s), \quad U_{t+1} = \sum_{k=1}^{T-t} r_{t+k}. && \text{(rollout)}\\
&= \mathbb{E}_{\pi,\mu}(r_t \mid s_t = s) + \mathbb{E}_{\pi,\mu}(U_{t+1} \mid s_t = s) && (2.2)\\
& && (2.3)
\end{aligned}
$$

Directly gives policy evaluation algorithms.

Policy evaluation

The value function of a policy π (for $\gamma = 1$, $T < \infty$)

$$
\begin{aligned}
V^{\pi}_{\mu,t}(s) &\triangleq \mathbb{E}_{\pi,\mu}(U_t \mid s_t = s) && (2.1)\\
&= \sum_{k=0}^{T-t} \mathbb{E}_{\pi,\mu}(r_{t+k} \mid s_t = s) && (\text{rollout})\\
&= \mathbb{E}_{\pi,\mu}(r_t \mid s_t = s) + \mathbb{E}_{\pi,\mu}(U_{t+1} \mid s_t = s) && (2.2)\\
&= \mathbb{E}_{\mu,\pi}(r_t \mid s_t = s) + \sum_{i \in \mathcal{S}} V^{\pi}_{\mu,t+1}(i)\,\mathbb{P}_{\mu,\pi}(s_{t+1} = i | s_t = s). && (2.3)\\
& && (2.4)
\end{aligned}
$$

Directly gives policy evaluation algorithms.

Algorithm 2. Policy evaluation using backwards induction

For each state $s \in S$, for $t = 1, \ldots, T - 1$:

$$\boldsymbol{v}_t(s) = \boldsymbol{r}(s) + \sum_{j \in S} \mathbb{P}_{\mu,\pi}(s_{t+1} = j \mid s_t = s)\boldsymbol{v}_{t+1}(j), \tag{2.5}$$

with $\boldsymbol{v}_T(s) = \boldsymbol{r}(s)$.

Theorem

Algorithm 2 results in estimates with the property:

$$\boldsymbol{v}_t(s) = V^{\pi}_{\mu,t}(s) \tag{2.6}$$

Algorithm 3. Policy optimisation using backwards induction

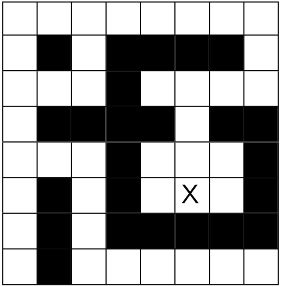
```
Input $\mu$, $\mathcal{S}_T$.
Initialise $\boldsymbol{v}_T(s)$, for all $s \in \mathcal{S}_T$.
for $n = T-1, T-2, \ldots, 1$ do
    for $s \in \mathcal{S}_n$ do
        $\pi_n(s) = \arg\max_a \mathbb{P}_\mu(s'|s,a)[\mathbb{E}_\mu(r|s',s) + \boldsymbol{v}_{n+1}(s')]$
        $\boldsymbol{v}_n(s) = \sum_{s' \in \mathcal{S}_{n+1}} \mathbb{P}_\mu(s'|s,\pi_n(s))[\mathbb{E}_\mu(r|s',s) + \boldsymbol{v}_{n+1}(s')]$
    end for
end for
Return $\pi = (\pi_n)_{n=1}^T$.
```

Theorem

For a T-horizon problems, backwards induction is optimal, i.e.

$$\boldsymbol{v}_n(s) = V^*_{\mu,n}(s) \tag{2.7}$$

Deterministic shortest-path problems



Properties

- $\gamma = 1, \ T \to \infty$.
- $r_t = -1$ 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.

What is the shortest path to the destination from any point?

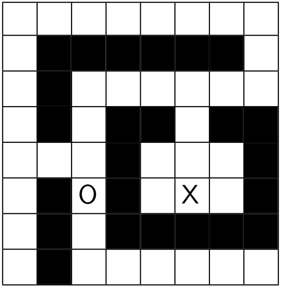
Shortest-path problem solution

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*.
- Remember Dijkstra's algorithm?

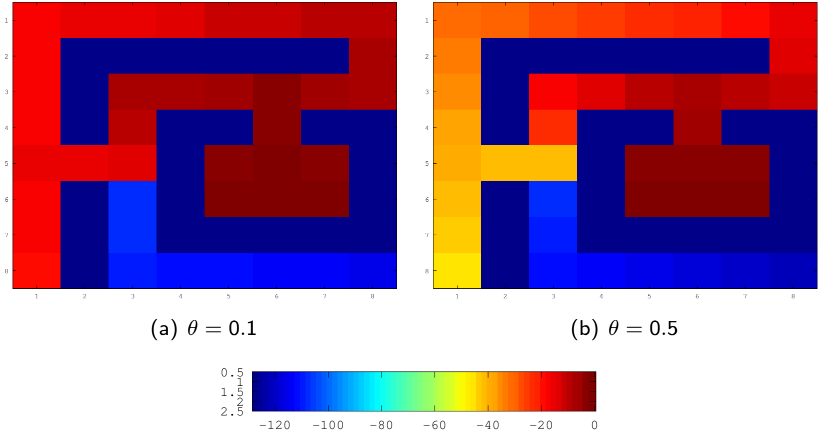
Stochastic shortest path problem, with a pit



Properties

- $\gamma = 1$, $T \to \infty$.
- $r_t = -1$, but $r_t = 0$ at X and -100 at O and episode 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 θ. Walls block.

For what value of θ is it better to take the dangerous shortcut? (However, if we want to take into account risk explicitly we must modify the agent's utility function)



(a) $\theta = 0.1$

(b) $\theta = 0.5$

(c) value

Figure: Pit maze solutions for two values of θ.

- Now we can find the optimal policy for any known μ.
- What if μ is unknown?

Introduction

Algorithms for known MDPs

Dealing with unknown MDPs
- Stochastic approximation view
- Decision theoretic view
- Multi-armed bandits
- Belief-augmented MDPs

Stochastic approximation for unknown μ

Replace $\mathbb{P}_\mu$ with P_t and $\mathbb{E}_\mu$ with E_t:

- The empirical distribution at time t?
- Combine with gradient descent?

Algorithm 4. Stochastic backwards induction

```
Initialise $v_0(s)$.
for $t = 1, 2, \ldots$ do
    $\pi_t(s) = \arg\max_a P_t(s'|s, a)[E_t(r|s', s) + v_{n+1}(s')]$
    $v_n(s) = \sum_{s' \in \mathcal{S}_{n+1}} P_t(s'|s, \pi_n(s))[E_t(r|s', s) + v_{n+1}(s')]$
    Update $P_t, E_t$ model with $(s_t, a_t, r_{t+1} s_{t+1})$.
end for
Return $\pi = (\pi_n)_{n=1}^T$.
```

Unfortunately, these do not take into account uncertainty about μ.

Decision-theoretic view

Bayesian framework

- Assume the true MDP $\mu^* \in \mathcal{M}$.
- Each $\mu \in \mathcal{M}$ defines: $\mathbb{P}_\mu(s_{t+1} \mid s_t, a_t)$, $\mathbb{E}_\mu(r_t \mid s_t)$.
- Choose a subjective prior probability ξ_0 on $\mathcal{M}$.

Decision-theoretic view

Bayesian framework

- Assume the true MDP $\mu^* \in \mathcal{M}$.
- Each $\mu \in \mathcal{M}$ defines: $\mathbb{P}_\mu(s_{t+1} \mid s_t, a_t)$, $\mathbb{E}_\mu(r_t \mid s_t)$.
- Choose a subjective prior probability ξ_0 on $\mathcal{M}$.

Optimal policy for a given belief ξ_t

$$\pi^*(\xi_t) \triangleq \operatorname*{arg\,max}_{\pi \in \Pi} \mathbb{E}^\pi_{\xi_t} U_t$$

$$\mathbb{E}^\pi_{\xi_t} U_t = \sum_{\mu \in \mathcal{M}} \left(\mathbb{E}^\pi_\mu U_t\right) \xi_t(\mu)$$

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

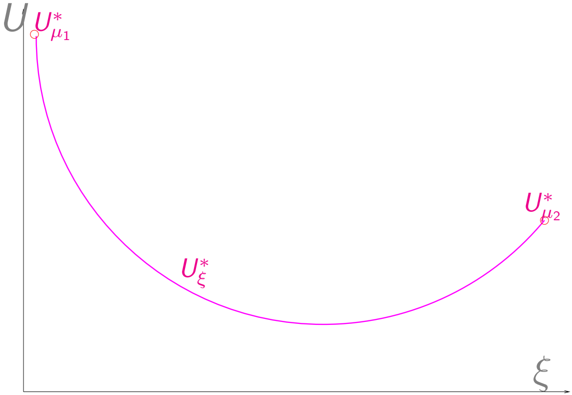


Figure: A geometric view of the bounds

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

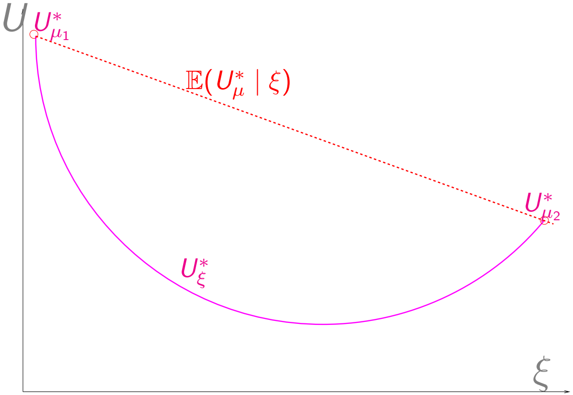


Figure: A geometric view of the bounds

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

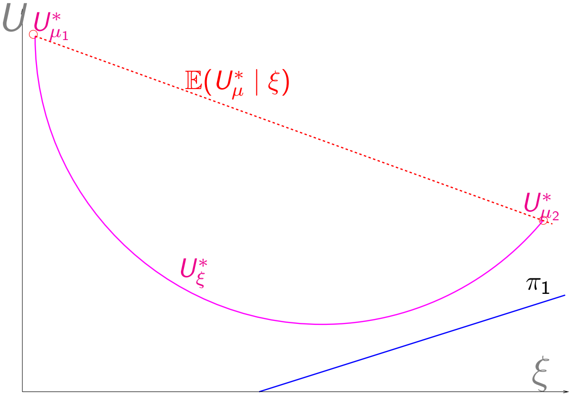


Figure: A geometric view of the bounds

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

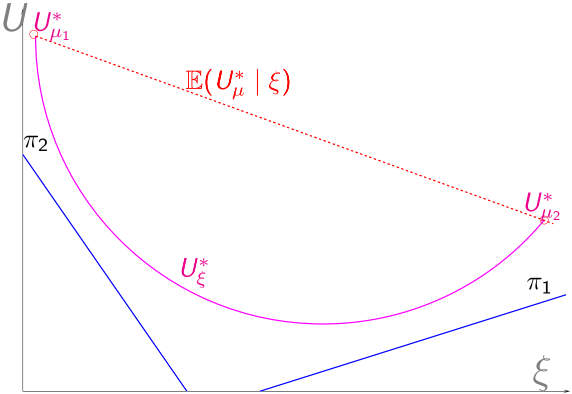


Figure: A geometric view of the bounds

ξ-optimal utility $U_\xi^* \triangleq \max_\pi \mathbb{E}_\xi^\pi U$

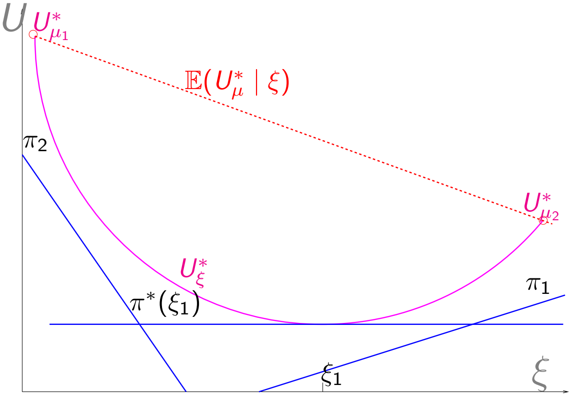


Figure: A geometric view of the bounds

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

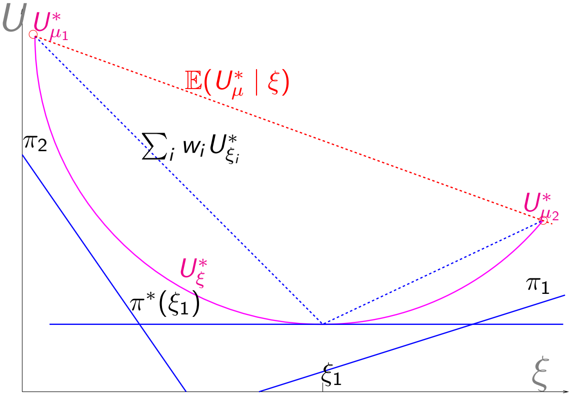


Figure: A geometric view of the bounds

Updating the belief

Belief update for finite $\mathcal{M}$

$$\xi_{t+1}(\mu) \triangleq \frac{\mathbb{P}_\mu(r_{t+1}, s_{t+1} \mid s_t, a_t)\xi_t(\mu)}{\sum_{\mu' \in \mathcal{M}} \mathbb{P}_{\mu'}(r_{t+1}, s_{t+1} \mid s_t, a_t)\xi_t(\mu')}$$

Closed-form posterior calculation

- Finite $\mathcal{M}$.
- Conjugate distributions (i.e. Dirichlet-multinomial).

Does this mean we can compute the optimal policy in Π?

Updating the belief

Belief update for finite $\mathcal{M}$

$$\xi_{t+1}(\mu) \triangleq \frac{\mathbb{P}_{\mu}(r_{t+1}, s_{t+1} \mid s_t, a_t)\xi_t(\mu)}{\sum_{\mu' \in \mathcal{M}} \mathbb{P}_{\mu'}(r_{t+1}, s_{t+1} \mid s_t, a_t)\xi_t(\mu')}$$

Closed-form posterior calculation

- Finite $\mathcal{M}$.
- Conjugate distributions (i.e. Dirichlet-multinomial).

Does this mean we can compute the optimal policy in Π?

- No.

Updating the belief

Belief update for finite $\mathcal{M}$

$$\xi_{t+1}(\mu) \triangleq \frac{\mathbb{P}_{\mu}(r_{t+1}, s_{t+1} \mid s_t, a_t)\xi_t(\mu)}{\sum_{\mu' \in \mathcal{M}} \mathbb{P}_{\mu'}(r_{t+1}, s_{t+1} \mid s_t, a_t)\xi_t(\mu')}$$

Closed-form posterior calculation

- Finite $\mathcal{M}$.
- Conjugate distributions (i.e. Dirichlet-multinomial).

Does this mean we can compute the optimal policy in Π?

- No.
- Unless Π is small.

Updating the belief

Belief update for finite $\mathcal{M}$

$$\xi_{t+1}(\mu) \triangleq \frac{\mathbb{P}_{\mu}(r_{t+1}, s_{t+1} \mid s_t, a_t)\xi_t(\mu)}{\sum_{\mu' \in \mathcal{M}} \mathbb{P}_{\mu'}(r_{t+1}, s_{t+1} \mid s_t, a_t)\xi_t(\mu')}$$

Closed-form posterior calculation

- Finite $\mathcal{M}$.
- Conjugate distributions (i.e. Dirichlet-multinomial).

Does this mean we can compute the optimal policy in Π?

- No.
- Unless Π is small.
- Typically, Π is exponential-sized.

The stochastic n-armed bandit problem revisited

- Actions $\mathcal{A} = \{1, \ldots, n\}$.
- Selecting i results in a random reward $r_t \sim P_i$.
- Select actions to maximise

$$\sum_{t=0}^{T} r_t,$$

  horizon $T \geq 0$.
- P_i is unknown.

Bernoulli bandits

Consider n Bernoulli bandits with unknown parameters θ_i, $i = 1, \ldots, n$ such that

$$r_t \mid a_t = i \sim \mathcal{B}ern(\theta_i), \qquad \mathbb{E}(r_t \mid a_t = i) = \theta_i. \tag{3.1}$$

Bernoulli bandits

Consider n Bernoulli bandits with unknown parameters θ_i, $i = 1, \ldots, n$ such that

$$r_t \mid a_t = i \sim \mathcal{Bern}(\theta_i), \qquad \mathbb{E}(r_t \mid a_t = i) = \theta_i. \tag{3.1}$$

Prior belief: Beta distribution $\mathcal{Beta}(\alpha_i, \beta_i)$, with density $f(\theta \mid \alpha_i, \beta_i)$ so that

$$\xi(\theta_1, \ldots, \theta_n) = \prod_{i=1}^{n} f(\theta_i \mid \alpha_i, \beta_i).$$

Bernoulli bandits

Consider n Bernoulli bandits with unknown parameters θ_i, $i = 1, \ldots, n$ such that

$$r_t \mid a_t = i \sim \mathcal{B}ern(\theta_i), \qquad \mathbb{E}(r_t \mid a_t = i) = \theta_i. \tag{3.1}$$

Prior belief: Beta distribution $\mathcal{B}eta(\alpha_i, \beta_i)$, with density $f(\theta \mid \alpha_i, \beta_i)$ so that

$$\xi(\theta_1, \ldots, \theta_n) = \prod_{i=1}^{n} f(\theta_i \mid \alpha_i, \beta_i).$$

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

Bernoulli bandits

Consider n Bernoulli bandits with unknown parameters θ_i, $i = 1, \ldots, n$ such that

$$r_t \mid a_t = i \sim \mathcal{B}ern(\theta_i), \qquad \mathbb{E}(r_t \mid a_t = i) = \theta_i. \tag{3.1}$$

Prior belief: Beta distribution $\mathcal{B}eta(\alpha_i, \beta_i)$, with density $f(\theta \mid \alpha_i, \beta_i)$ so that

$$\xi(\theta_1, \ldots, \theta_n) = \prod_{i=1}^{n} f(\theta_i \mid \alpha_i, \beta_i).$$

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

Posterior distribution for the parameter of arm i:

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

Belief states

- The state of the bandit problem is the state of our belief.
- A sufficient statistic for our belief is the number of times we played each bandit and the total reward from each bandit.
- Thus, our state at time t is entirely described our priors α, β (the initial state) and the vectors

$$N_t = (N_{t,1}, \ldots, N_{t,i}) \tag{3.2}$$

$$\hat{r}_t = (\hat{r}_{t,1}, \ldots, \hat{r}_{t,i}). \tag{3.3}$$

- At any time t, we can calculate the probability of observing $r_t = 1$ or $r_t = 0$ if we pull arm i as:

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

- The next state is well-defined and depends only on the current state.
- For this reason, the decision-theoretic n-armed bandit problem can be formalised as a Markov decision process.

The information-state Markov decision process

Exercise

- *Write the transition kernel for MDP defined by a bandit process with two arms:*
  1. *Receive reward* 1 *w.p.* θ, *reward* -1 *w.p.* $-\theta$.
  2. *Receive* 0 *reward and quit.*
- *What is the optimal decision for* $T = 1$*?*
- *What about* $T = 2$*?*

Exercise

- *Write the transition kernel for information-state MDP for the above process.*
- *What is the optimal decision for* $T = 1$*?*
- *What about* $T = 2$*?*

Bayesian RL algorithms

Augmented MDP approaches

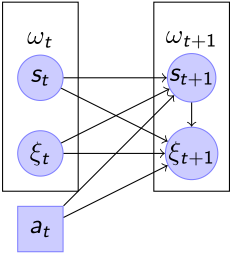
- Augmente the original MDP state s_t with the belief state ξ_t.
- Construct a new augmented MDP with state $\omega_t = (s_t, \xi_t)$.
- Solve this MDP using tree search, Monte Carlo search etc.

Value function bounds

- Use the convexity of the Bayes-optimal value function.
- Improve bounds iteratively via search.

Open question

Which methods are best for which settings?



The augmented MDP

The optimal policy for the augmented MDP is ξ-optimal for the original problem.

$$\mathbb{P}(s_{t+1} \in S \mid \xi_t, s_t, a_t) \triangleq \int_S \mathbb{P}_\mu(s_{t+1} \in S \mid s_t, a_t) \, \mathrm{d}\xi_t(\mu) \tag{3.4}$$

$$\xi_{t+1}(\cdot) = \xi_t(\cdot \mid s_{t+1}, s_t, a_t) \tag{3.5}$$

Value function bounds and Thompson sampling

$$\frac{1}{K}\sum_{k=1}^{K} V^{\pi}_{\mu_k} \lesssim V^*_{\xi} \lesssim \frac{1}{K}\sum_{k=1}^{K} V^*_{\mu_k}, \qquad \mu_k \sim \xi$$

Algorithm 5. Thompson sampling ($K = 1$).

```
Input prior $\xi_0$ on $\mathcal{M}$.
for episode $k$ do

end for
```

Value function bounds and Thompson sampling

$$\frac{1}{K}\sum_{k=1}^{K} V^{\pi}_{\mu_k} \lesssim V^*_{\xi} \lesssim \frac{1}{K}\sum_{k=1}^{K} V^*_{\mu_k}, \qquad \mu_k \sim \xi$$

Algorithm 5. Thompson sampling ($K = 1$).

```
Input prior $\xi_0$ on $\mathcal{M}$.
for episode $k$ do
    // – Thompson sampling – //
    $\mu^{(k)} \sim \xi_{t_k}(\mu)$                                  // generate MDP from posterior
    $\pi^{(k)} \approx \arg\max_{\pi} \mathbb{E}^{\pi}_{\mu^{(k)}} U$                   // Get new policy using ADP
end for
```

Value function bounds and Thompson sampling

$$\frac{1}{K}\sum_{k=1}^{K} V^{\pi}_{\mu_k} \lesssim V^{*}_{\xi} \lesssim \frac{1}{K}\sum_{k=1}^{K} V^{*}_{\mu_k}, \qquad \mu_k \sim \xi$$

Algorithm 5. Thompson sampling ($K = 1$).

```
Input prior $\xi_0$ on $\mathcal{M}$.
for episode $k$ do
    // – Thompson sampling – //
    $\mu^{(k)} \sim \xi_{t_k}(\mu)$                                   // generate MDP from posterior
    $\pi^{(k)} \approx \arg\max_{\pi} \mathbb{E}^{\pi}_{\mu^{(k)}} U$                // Get new policy using ADP
    // – Run policy and collect data – //
    for $t = t_k, \ldots, t_{k+1} - 1$ do
        $a_t \mid s_t = s \sim \pi^{(k)}(a \mid s)$                              // Take action
        $\xi_{t+1}(\mu) = \xi_t(\mu \mid s_{t+1}, a_t, s_t)$                 // Update posterior
    end for
end for
```

Summary

Markov decision processes

Can represent : Shortest path problems, Stopping problems, Experiment design problems, Multi-armed bandit problems, Reinforcement learning problems.

Backwards induction

- In the class of dynamic programming algorithms.
- Tractable when either the state space $\mathcal{S}$ or the horizon T are small.

Optimal decisions and Bayesian reinforcement learning

- A known environment is represented as an MDP.
- Bandit problems can be solved by representing them as infinite-state MDPs.
- Any unknown environment can be represented as a distribution over MDPs.
- The decision problem can again be formulated as an infinite-state MDP.

Open questions

Modelling

- General vs. task-specific priors.
- Efficient inference.

Planning

- Theoretical performance of optimal solution.
- Performance gap of approximations.