Introduction to POMDPs

Dr. Stephan Timmer

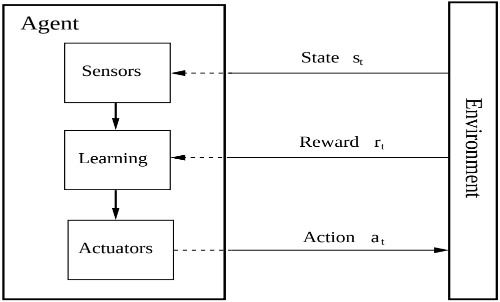
Institute of Cognitive Science, University of Osnabrück

Partially Observable Markov Decision Processes

Uncertainty in Reinforcement Learning

Agent Architecture

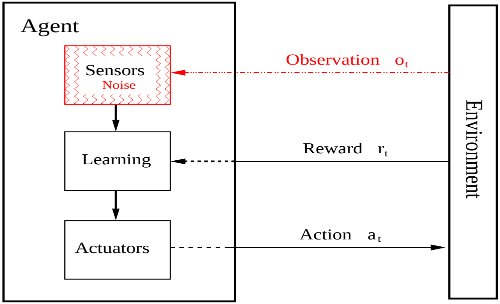
Agent achieves goals by interacting with environment



Uncertainty in Reinforcement Learning

Agent Architecture

Agent achieves goals by interacting with environment



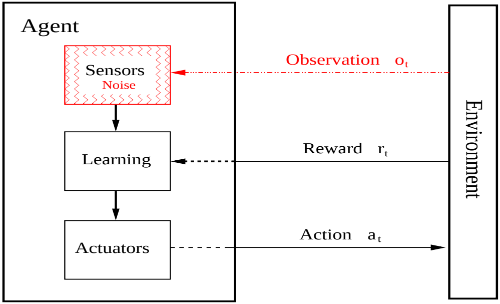
Partial Observability

- Uncertainty about state is induced through noisy sensory measurements

Uncertainty in Reinforcement Learning

Agent Architecture

Agent achieves goals by interacting with environment



Partial Observability

- Uncertainty about state is induced through noisy sensory measurements
- Observations do not reveal complete state information

Problem Specification

Partially Observable Markov Decision Process

A POMDP is given by $M = (T, S, O, A, P_S, P_O, r)$

Problem Specification

Partially Observable Markov Decision Process

A POMDP is given by $M = (T, S, O, A, P_S, P_O, r)$

T: Discretized Time Finite set of decision stages $T := \{0, ..., T_F\}$

Problem Specification

Partially Observable Markov Decision Process

A POMDP is given by $M = (T, S, O, A, P_S, P_O, r)$

T: Discretized Time Finite set of decision stages $T := \{0, ..., T_F\}$

S: State Space Set of environmental states denoted by $s \in S$

Problem Specification

Partially Observable Markov Decision Process

A POMDP is given by $M = (T, S, O, A, P_S, P_O, r)$

T: Discretized Time Finite set of decision stages $T := \{0, ..., T_F\}$

S: State Space Set of environmental states denoted by $s \in S$

O: Observation Space Set of observations denoted by $o \in O$

Problem Specification

Partially Observable Markov Decision Process

A POMDP is given by $M = (T, S, O, A, P_S, P_O, r)$

T: Discretized Time Finite set of decision stages $T := \{0, ..., T_F\}$

S: State Space Set of environmental states denoted by $s \in S$

O: Observation Space Set of observations denoted by $o \in O$

A: Action Space Set of available actions denoted by $a \in A$

Problem Specification

Partially Observable Markov Decision Process

A POMDP is given by $M = (T, S, O, A, P_S, P_O, r)$

T: Discretized Time Finite set of decision stages $T := \{0, ..., T_F\}$

S: State Space Set of environmental states denoted by $s \in S$

O: Observation Space Set of observations denoted by $o \in O$

A: Action Space Set of available actions denoted by $a \in A$

P_S: Transition Model Transition matrix $P_S(s_t \mid s_{t-1}, a_{t-1})$

Problem Specification

Partially Observable Markov Decision Process

A POMDP is given by $M = (T, S, O, A, P_S, P_O, r)$

T: Discretized Time Finite set of decision stages $T := \{0, ..., T_F\}$

S: State Space Set of environmental states denoted by $s \in S$

O: Observation Space Set of observations denoted by $o \in O$

A: Action Space Set of available actions denoted by $a \in A$

P_S: Transition Model Transition matrix $P_S(s_t \mid s_{t-1}, a_{t-1})$

P_O: Observation Model Observation matrix $P_O(o_t \mid s_t, a_{t-1})$

Problem Specification

Partially Observable Markov Decision Process

A POMDP is given by $M = (T, S, O, A, P_S, P_O, r)$

T: Discretized Time Finite set of decision stages $T := \{0, ..., T_F\}$

S: State Space Set of environmental states denoted by $s \in S$

O: Observation Space Set of observations denoted by $o \in O$

A: Action Space Set of available actions denoted by $a \in A$

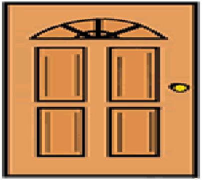
P_S: Transition Model Transition matrix $P_S(s_t \mid s_{t-1}, a_{t-1})$

P_O: Observation Model Observation matrix $P_O(o_t \mid s_t, a_{t-1})$

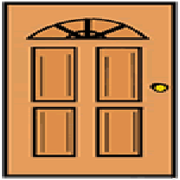
r: Reward Model Reward function $r : S \times A \to \mathbb{R}$

The Famous Tiger Problem

State Space



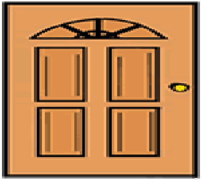
$S = \text{LEFT}$



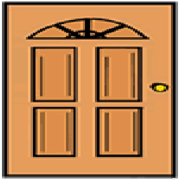
$S = \text{RIGHT}$

The Famous Tiger Problem

State Space



S = LEFT

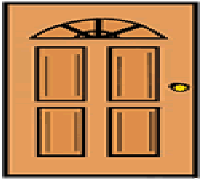


S = RIGHT

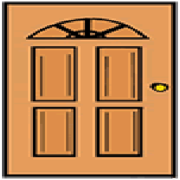
Actions and Rewards

The Famous Tiger Problem

State Space

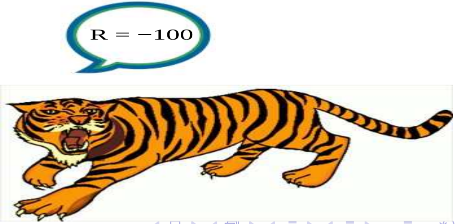


S = LEFT



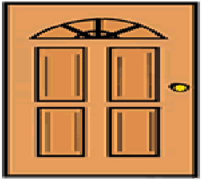
S = RIGHT

Actions and Rewards

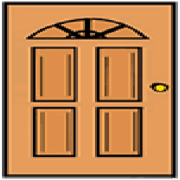


The Famous Tiger Problem

State Space

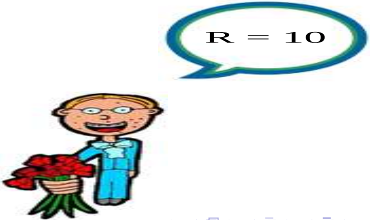


S = LEFT



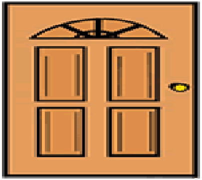
S = RIGHT

Actions and Rewards

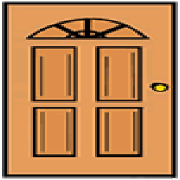


The Famous Tiger Problem

State Space



S = LEFT

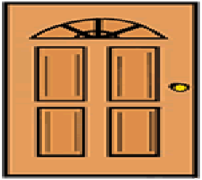


S = RIGHT

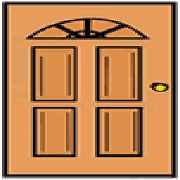
Observations (Listening for $R = -1$)

The Famous Tiger Problem

State Space



$\mathrm{S} = \mathrm{LEFT}$



$\mathrm{S} = \mathrm{RIGHT}$

Observations (Listening for $R = -1$)

$$
\begin{aligned}
P(o = HL \mid s = LEFT) &= 0.85 \\
P(o = HR \mid s = LEFT) &= 0.15 \\
P(o = HL \mid s = RIGHT) &= 0.15 \\
P(o = HR \mid s = RIGHT) &= 0.85
\end{aligned}
$$

Finite Horizon Policies

Fully Observable Processes

An optimal policy is given by a sequence of mappings $\pi^* := (\pi^*_t)_{(0 \le t < T_F)}$ each constituting a rule $\pi_t : S \longrightarrow A$ for choosing actions

Finite Horizon Policies

Fully Observable Processes

An optimal policy is given by a sequence of mappings $\pi^* := (\pi_t^*)_{(0 \leq t < T_F)}$ each constituting a rule $\pi_t : S \longrightarrow A$ for choosing actions

$$\pi^* \in \arg\max_{\pi \in \Pi} E[\sum_{t=0}^{T_F - 1} r(s_t, \pi_t(s_t)) + r(s_{T_F}, \cdot)]$$

Finite Horizon Policies

Fully Observable Processes

An optimal policy is given by a sequence of mappings $\pi^* := (\pi^*_t)_{(0 \leq t < T_F)}$ each constituting a rule $\pi_t : S \longrightarrow A$ for choosing actions

$$\pi^* \in \arg\max_{\pi \in \Pi} E[\sum_{t=0}^{T_F - 1} r(s_t, \pi_t(s_t)) + r(s_{T_F}, \cdot)]$$

Partially Observable Processes

Given that current state s_t is unknown, what information is available in order to choose optimal actions?

Finite Horizon Policies

Fully Observable Processes

An optimal policy is given by a sequence of mappings $\pi^* := (\pi_t^*)_{(0 \leq t < T_F)}$ each constituting a rule $\pi_t : S \longrightarrow A$ for choosing actions

$$\pi^* \in \arg\max_{\pi \in \Pi} E[\sum_{t=0}^{T_F - 1} r(s_t, \pi_t(s_t)) + r(s_{T_F}, \cdot)]$$

Partially Observable Processes

Given that current state s_t is unknown, what information is available in order to choose optimal actions?

The complete sequence of past actions and observations

Finite Horizon Policies

Fully Observable Processes

An optimal policy is given by a sequence of mappings $\pi^* := (\pi_t^*)_{(0 \leq t < T_F)}$ each constituting a rule $\pi_t : S \longrightarrow A$ for choosing actions

$$\pi^* \in \arg\max_{\pi \in \Pi} E[\sum_{t=0}^{T_F - 1} r(s_t, \pi_t(s_t)) + r(s_{T_F}, \cdot)]$$

Partially Observable Processes

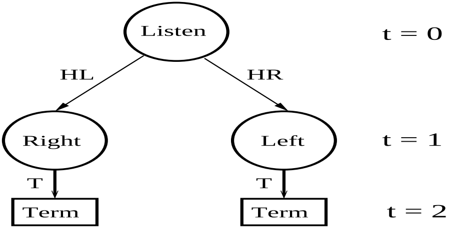
Given that current state s_t is unknown, what information is available in order to choose optimal actions?

The complete sequence of past actions and observations

$\Rightarrow$ Optimal policy π^* depends on past actions and observations

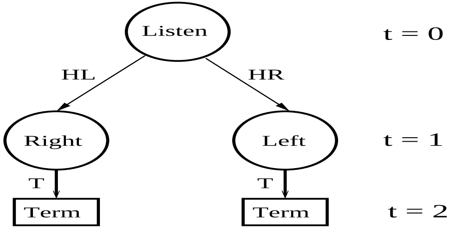
Example Policy for $T_F = 2$

Policy $\equiv$ Tree



Example Policy for $T_F = 2$

Policy $\equiv$ Tree



Policy $\equiv$ Mapping

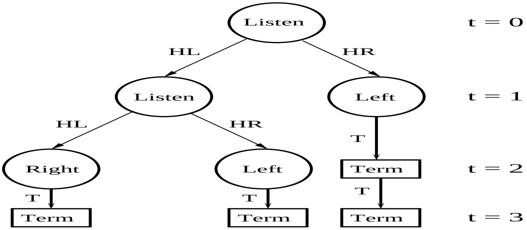
Policy $\pi : (A \times O)^* \rightarrow A$

$$\pi([]) = Listen$$

$$\pi([Listen, HL]) = Right \quad \pi([Listen, HR]) = Left$$

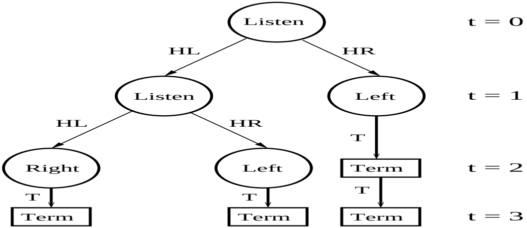
Example Policy for $T_F = 3$

Policy $\equiv$ Tree



Example Policy for $T_F = 3$

Policy $\equiv$ Tree



Policy $\equiv$ Mapping

$$\pi([]) = Listen$$

$$\pi([Listen, HL] = Listen \qquad \pi([Listen, HR]) = Left$$

$$\pi([Listen, HL, Listen, HL]) = Right \qquad \pi([Listen, HL, Listen, HR]) = Left$$

Information State Space

How to solve POMDPs?

Question: How is it possible to compute an optimal policy tree?

Information State Space

How to solve POMDPs?

Question: How is it possible to compute an optimal policy tree?

Answer: Perform value iteration on branches of policy trees

Information State Space

How to solve POMDPs?

Question: How is it possible to compute an optimal policy tree?
Answer: Perform value iteration on branches of policy trees

$\Rightarrow$ Branches correspond to sequences of actions and observations

Information State Space

How to solve POMDPs?

Question: How is it possible to compute an optimal policy tree?
Answer: Perform value iteration on branches of policy trees

$\Rightarrow$ Branches correspond to sequences of actions and observations

Definition (Information States)

The information state I_t is defined to be the complete sequence of actions and observations $[a_0, o_1, a_1, o_2, ..., a_{t-1}, o_t]$ until time t.
The information state space I contains all possible information states.

Information State MDP

Given a POMDP $M := (T, S, A, O, P_S, P_O, r)$, we compute an optimal policy for M by transforming M into an MDP called *Information State MDP*

Information State MDP

Given a POMDP $M := (T, S, A, O, P_S, P_O, r)$, we compute an optimal policy for M by transforming M into an MDP called *Information State MDP*

Definition (Information State MDP)

Discretized Time Does not change

Information State MDP

Given a POMDP $M := (T, S, A, O, P_S, P_O, r)$, we compute an optimal policy for M by transforming M into an MDP called *Information State MDP*

Definition (Information State MDP)

Discretized Time Does not change

State Space Given by information state space I

Information State MDP

Given a POMDP $M := (T, S, A, O, P_S, P_O, r)$, we compute an optimal policy for M by transforming M into an MDP called *Information State MDP*

Definition (Information State MDP)

Discretized Time Does not change

State Space Given by information state space I

Action Space Does not change

Information State MDP

Given a POMDP $M := (T, S, A, O, P_S, P_O, r)$, we compute an optimal policy for M by transforming M into an MDP called *Information State MDP*

Definition (Information State MDP)

Discretized Time: Does not change

State Space: Given by information state space I

Action Space: Does not change

Transitions: $P_I(I_{t+1} = [a_0, o_1, ..., a_t, o_{t+1}] \mid a_t, I_t) = p(o_{t+1}|a_t, I_t)$

Information State MDP

Given a POMDP $M := (T, S, A, O, P_S, P_O, r)$, we compute an optimal policy for M by transforming M into an MDP called *Information State MDP*

Definition (Information State MDP)

Discretized Time Does not change

State Space Given by information state space I

Action Space Does not change

Transitions $P_I(I_{t+1} = [a_0, o_1, ..., a_t, o_{t+1}] \mid a_t, I_t) = p(o_{t+1}|a_t, I_t)$

Rewards $r_I(I_t, a_t) = \sum_{s \in S} p(s_t = s \mid I_t) r(s, a)$

Markov Property

Lemma

Information states constitute a markovian state space. It holds that $P_I(I_{t+1} \mid a_t, I_t) = P_I(I_{t+1} \mid a_t, I_t, I_{t-1}, \dots, I_0)$ *(memoryless process)*

Markov Property

Lemma

Information states constitute a markovian state space. It holds that $P_I(I_{t+1} \mid a_t, I_t) = P_I(I_{t+1} \mid a_t, I_t, I_{t-1}, ..., I_0)$ *(memoryless process)*

Proof.

$I_k = [a_0, o_1, ..., a_{k-1}, o_k] \subset [a_0, o_1, ..., a_{t-1}, o_t] = I_t \ (0 \leq k \leq t-1)$

Thus, it follows $P_I(I_{t+1} \mid a_t, I_t) = P_I(I_{t+1} \mid a_t, I_t, I_{t-1}, ..., I_0)$ □

Value Iteration on Information States

Value Functions on Information States

Compute sequence of value functions $(V_n)_{0 \leq n \leq T_F}$ defined on information states, $V_n : (A \times O)^* \longrightarrow \mathbb{R}$

Value Iteration on Information States

Value Functions on Information States

Compute sequence of value functions $(V_n)_{0 \leq n \leq T_F}$ defined on information states, $V_n : (A \times O)^* \longrightarrow \mathbb{R}$

Algorithm

1. Initialization

$$V_{T_F}(I_{T_F}) = \sum_{s \in S} p(s_{T_F} = s \mid I_{T_F}) r(s, \cdot)$$

Value Iteration on Information States

Value Functions on Information States

Compute sequence of value functions $(V_n)_{0 \leq n \leq T_F}$ defined on information states, $V_n : (A \times O)^* \longrightarrow \mathbb{R}$

Algorithm

1. Initialization

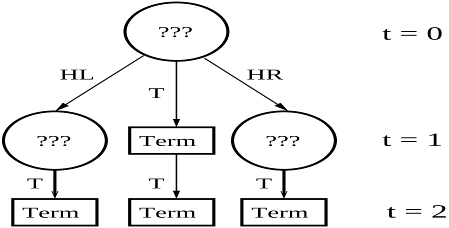
$$V_{T_F}(I_{T_F}) = \sum_{s \in S} p(s_{T_F} = s \mid I_{T_F}) r(s, \cdot)$$

2. Bellman Equation

$$V_n^*(I_t) = \max_{a \in A} [\sum_{s \in S} p(s_t = s | I_t) r(s, a)$$
$$+ \beta \sum_{o \in O} p(o_{t+1} = o | I_t, a) V_{n+1}^*(I_{t+1} = \{a_0, .., a_t = a, o_{t+1} = o\})]$$

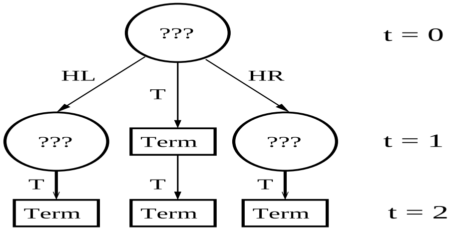
Value Iteration (Example)

Value Iteration



Value Iteration (Example)

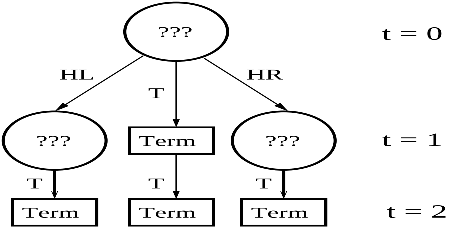
Value Iteration



$$V_0([]) \quad = \quad \max\{r(Left) + V_1([Left, T]),$$

Value Iteration (Example)

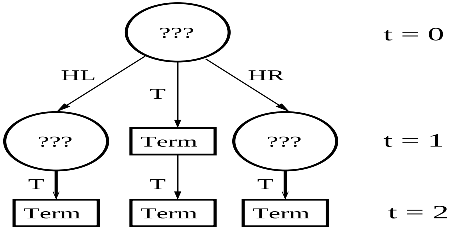
Value Iteration



$$V_0([]) \quad = \quad \max\{r(Left) + V_1([Left, T]),\ r(Right) + V_1([Right, T]),$$

Value Iteration (Example)

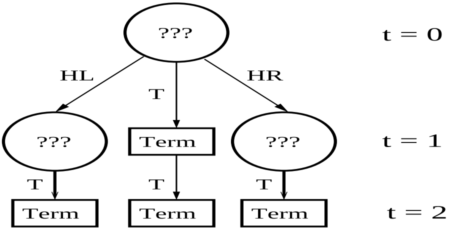
Value Iteration



$$V_0([]) = \max\{r(Left) + V_1([Left, T]),\ r(Right) + V_1([Right, T]),\ r(Listen) + \sum_{o \in \{HL,HR\}} p(o \mid [Listen]) V_1([Listen, o])\}$$

Value Iteration (Example)

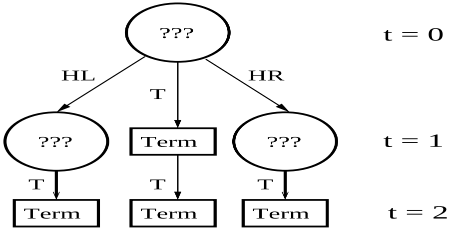
Value Iteration



$$
\begin{aligned}
V_1([Left, T]) &= V_2([Left, T, \cdot, T]) \\
V_1([Right, T]) &= V_2([Right, T, \cdot, T])
\end{aligned}
$$

Value Iteration (Example)

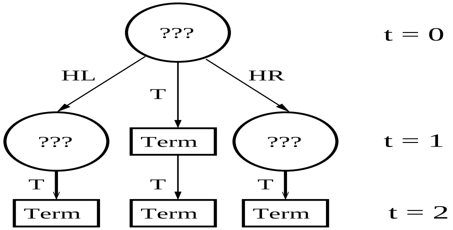
Value Iteration



$$\begin{aligned} V_1([Listen, HL]) &= \max\{r(Left) + V_2([Listen, HL, Left, T]), \\ &\quad r(Right) + V_2([Listen, HL, Right, T]), \\ &\quad r(Listen) + V_2([Listen, HL, Listen, T])\} \end{aligned}$$

Value Iteration (Example)

Value Iteration



$$
\begin{aligned}
V_2(I) &= \sum_{s \in S} p(s \mid I) r(s, \cdot) \\
&= 0
\end{aligned}
$$

Open Problems

1. Model of Information state MDP unknown

Open Problems

1. Model of Information state MDP unknown
   $\Rightarrow$ Formulas needed for $p(o_t \mid a_t, I_t)$ and $p(s_t \mid I_t)$

Open Problems

1. Model of Information state MDP unknown
   $\Rightarrow$ Formulas needed for $p(o_t \mid a_t, I_t)$ and $p(s_t \mid I_t)$
2. Length of information states grows linearly with horizon T_F

Open Problems

1. Model of Information state MDP unknown
   $\Rightarrow$ Formulas needed for $p(o_t \mid a_t, I_t)$ and $p(s_t \mid I_t)$
2. Length of information states grows linearly with horizon T_F
3. Number of information states grows exponentially with horizon T_F

Belief States

How can we represent information states by a (data)-structure of constant size?

Belief States

How can we represent information states by a (data)-structure of constant size?

Definition (Belief State)

The belief state b_t is an $|S|$-dimensional vector such that $b_t(s) := p(s_t = s \mid I_t)$. Belief states therefore form probability distributions over states.

Value Functions on Belief States

Theorem (Equivalence of Information States and Belief States)

Given a POMDP $(T, S, A, O, P_S, P_O, r)$ *and a finite horizon* T_F*, it is possible to rewrite the sequence of (optimal) value functions* $(V_n^*)_{(0 \leq n \leq T_F)}$ *in terms of belief states*

Value Functions on Belief States

Theorem (Equivalence of Information States and Belief States)

Given a POMDP $(T, S, A, O, P_S, P_O, r)$ *and a finite horizon* T_F*, it is possible to rewrite the sequence of (optimal) value functions* $(V_n^*)_{(0 \leq n \leq T_F)}$ *in terms of belief states*

$$V_n^*(b) := \max_{a \in A} [\sum_{s \in S} b(s) r(s, a) + \beta \sum_{o \in O} \sum_{s' \in s, s'' \in S} P_O(o \mid s'', a) \cdot P_S(s'' \mid s', a) b(s') V_{n+1}^*(b_o^a)]$$

Value Functions on Belief States

Theorem (Equivalence of Information States and Belief States)

Given a POMDP $(T, S, A, O, P_S, P_O, r)$ and a finite horizon T_F, it is possible to rewrite the sequence of (optimal) value functions $(V_n^*)_{(0 \leq n \leq T_F)}$ in terms of belief states

$$V_n^*(b) := \max_{a \in A} [\sum_{s \in S} b(s) r(s,a) + \beta \sum_{o \in O} \sum_{s' \in s, s'' \in S} P_O(o \mid s'', a) \cdot P_S(s'' \mid s', a) b(s') V_{n+1}^*(b_o^a)]$$

The belief state b_o^a denotes the successor belief of b after executing action $a \in A$ and making observation $o \in O$

$$b_o^a(s) := \frac{P_O(o \mid s, a) \sum_{s' \in S} P_S(s \mid s', a) b(s')}{\sum_{s' \in S, s'' \in S} P_S(s'' \mid s', a) P_O(o \mid s'', a) b(s')}$$

Successor Beliefs

Lemma

Given that $b := b_t$, $a := a_t$, $o := o_{t+1}$, it holds that

$$b_o^a(s) = b_{t+1} = \frac{P_O(o \mid s, a) \sum_{s' \in S} P_S(s \mid s', a) b_t(s')}{\sum_{s' \in S, s'' \in S} P_S(s'' \mid s', a) P_O(o \mid s'', a) b_t(s')}$$

Successor Beliefs (2)

Proof.

$$b_{t+1}(s) = p(s_{t+1} = s \mid I_{t+1})$$

□

Successor Beliefs (2)

Proof.

$$\begin{aligned} b_{t+1}(s) &= p(s_{t+1} = s \mid I_{t+1}) \\ &= p(s_{t+1} = s \mid I_t, o_{t+1} = o, a_t = a) \end{aligned}$$

$\square$

Successor Beliefs (2)

Proof.

$$
\begin{aligned}
b_{t+1}(s) &= p(s_{t+1} = s \mid I_{t+1}) \\
&= p(s_{t+1} = s \mid I_t, o_{t+1} = o, a_t = a) \\
&= \frac{p(s_{t+1} = s, o_{t+1} = o, I_t, a_t = a)}{p(o_{t+1} = o, I_t, a_t = a)}
\end{aligned}
$$

□

Successor Beliefs (2)

Proof.

$$
\begin{aligned}
b_{t+1}(s) &= p(s_{t+1} = s \mid I_{t+1}) \\
&= p(s_{t+1} = s \mid I_t, o_{t+1} = o, a_t = a) \\
&= \frac{p(s_{t+1} = s, o_{t+1} = o, I_t, a_t = a)}{p(o_{t+1} = o, I_t, a_t = a)} \\
&= \frac{p(s_{t+1} = s, o_{t+1} = o \mid I_t, a_t = a)}{p(o_{t+1} = o \mid I_t, a_t = a)}
\end{aligned}
$$

$\square$

Successor Beliefs (3)

Proof.

Numerator

$$\sum_{s' \in S} b_t(s')$$

□

Successor Beliefs (3)

Proof.

Numerator

$$\sum_{s' \in S} P_S(s \mid s', a) b_t(s')$$

Successor Beliefs (3)

Proof.

Numerator

$$P_O(o \mid s, a) \sum_{s' \in S} P_S(s \mid s', a) b_t(s')$$

□

Successor Beliefs (3)

Proof.

Numerator

$$p(s_{t+1} = s, o_{t+1} = o \mid I_t, a_t = a) = P_O(o \mid s, a) \sum_{s' \in S} P_S(s \mid s', a) b_t(s')$$

$\square$

Successor Beliefs (3)

Proof.

Numerator

$$p(s_{t+1} = s, o_{t+1} = o \mid I_t, a_t = a) = P_O(o \mid s, a) \sum_{s' \in S} P_S(s \mid s', a) b_t(s')$$

Denominator

$$\sum_{s' \in S} b_t(s')$$

$\square$

Successor Beliefs (3)

Proof.

Numerator

$$p(s_{t+1} = s, o_{t+1} = o \mid I_t, a_t = a) = P_O(o \mid s, a) \sum_{s' \in S} P_S(s \mid s', a) b_t(s')$$

Denominator

$$\sum_{s'' \in S} \sum_{s' \in S} P_S(s'' \mid s', a) b_t(s')$$

$\square$

Successor Beliefs (3)

Proof.

Numerator

$$p(s_{t+1} = s, o_{t+1} = o \mid I_t, a_t = a) = P_O(o \mid s, a) \sum_{s' \in S} P_S(s \mid s', a) b_t(s')$$

Denominator

$$\sum_{s'' \in S} \sum_{s' \in S} P_O(o \mid s'', a) P_S(s'' \mid s', a) b_t(s')$$

$\square$

Successor Beliefs (3)

Proof.

Numerator

$$p(s_{t+1} = s, o_{t+1} = o \mid I_t, a_t = a) = P_O(o \mid s, a) \sum_{s' \in S} P_S(s \mid s', a) b_t(s')$$

Denominator

$$p(o_{t+1} = o \mid I_t, a_t = a) = \sum_{s'' \in S} \sum_{s' \in S} P_O(o \mid s'', a) P_S(s'' \mid s', a) b_t(s')$$

Value Functions on Belief States (2)

Theorem

The sequence of (optimal) value functions $(V_n^*)_{(0 \leq n \leq T_F)}$ can be rewritten in terms of belief states such that
$\forall\, 0 \leq n \leq T_F,\ \forall t \in T : V_n^*(I_t) = V_n^*(b_t)$

Value Functions on Belief States (2)

Theorem

The sequence of (optimal) value functions $(V_n^*)_{(0 \leq n \leq T_F)}$ *can be rewritten in terms of belief states such that*
$\forall\, 0 \leq n \leq T_F,\ \forall t \in T : V_n^*(I_t) = V_n^*(b_t)$

Proof.

By (backward) induction over n starting with $n = T_F$:

$\square$

Value Functions on Belief States (2)

Theorem

The sequence of (optimal) value functions $(V_n^*)_{(0 \leq n \leq T_F)}$ *can be rewritten in terms of belief states such that*
$\forall\, 0 \leq n \leq T_F,\ \forall t \in T : V_n^*(I_t) = V_n^*(b_t)$

Proof.

By (backward) induction over n starting with $n = T_F$:

$$V_{T_F}^*(I_t) = \sum_{s \in S} p(s_t = s \mid I_t) r(s, \cdot)$$

□

Value Functions on Belief States (2)

Theorem

The sequence of (optimal) value functions $(V_n^*)_{(0 \leq n \leq T_F)}$ *can be rewritten in terms of belief states such that*
$\forall\, 0 \leq n \leq T_F,\ \forall t \in T : V_n^*(I_t) = V_n^*(b_t)$

Proof.

By (backward) induction over n starting with $n = T_F$:

$$V_{T_F}^*(I_t) = \sum_{s \in S} p(s_t = s \mid I_t) r(s, \cdot)$$
$$= \sum_{s \in S} b_t(s) r(s, \cdot)$$

Value Functions on Belief States (2)

Theorem

The sequence of (optimal) value functions $(V_n^*)_{(0 \leq n \leq T_F)}$ *can be rewritten in terms of belief states such that*
$\forall\, 0 \leq n \leq T_F,\ \forall t \in T : V_n^*(I_t) = V_n^*(b_t)$

Proof.

By (backward) induction over n starting with $n = T_F$:

$$\begin{aligned} V_{T_F}^*(I_t) &= \sum_{s \in S} p(s_t = s \mid I_t) r(s, \cdot) \\ &= \sum_{s \in S} b_t(s) r(s, \cdot) \\ &= V_{T_F}^*(b_t) \end{aligned}$$

□

Value Functions on Belief States (3)

Proof.

For $n < T_F$, it holds that

$\square$

Value Functions on Belief States (3)

Proof.

For $n < T_F$, it holds that

$$V_n^*(I_t) = \max_{a \in A}[\sum_{s \in S} p(s_t = s|I_t)r(s,a) + \beta \sum_{o \in O} p(o_{t+1} = o|I_t, a)V_{n+1}^*(I_{t+1})]$$

$\square$

Value Functions on Belief States (3)

Proof.

For $n < T_F$, it holds that

$$V_n^*(I_t) = \max_{a \in A}[\sum_{s \in S} p(s_t = s|I_t)r(s,a) + \beta \sum_{o \in O} p(o_{t+1} = o|I_t, a)V_{n+1}^*(I_{t+1})]$$
$$= \max_{a \in A}[\sum_{s \in S} b_t(s)r(s,a) + \beta \sum_{o \in O} p(o_{t+1} = o|I_t, a)V_{n+1}^*(b_{t+1})]$$

□

Value Functions on Belief States (3)

Proof.

For $n < T_F$, it holds that

$$
\begin{aligned}
V_n^*(I_t) &= \max_{a \in A}[\sum_{s \in S} p(s_t = s | I_t) r(s,a) + \beta \sum_{o \in O} p(o_{t+1} = o | I_t, a) V_{n+1}^*(I_{t+1})] \\
&= \max_{a \in A}[\sum_{s \in S} b_t(s) r(s,a) + \beta \sum_{o \in O} p(o_{t+1} = o | I_t, a) V_{n+1}^*(b_{t+1})] \\
&= \max_{a \in A}[\sum_{s \in S} b_t(s) r(s,a) + \beta \sum_{o \in O} p(o_{t+1} = o \mid I_t, a) V_{n+1}^*(b_o^a)]
\end{aligned}
$$

$\square$

Value Functions on Belief States (3)

Proof.

For $n < T_F$, it holds that

$$
\begin{aligned}
V_n^*(I_t) &= \max_{a \in A}[\sum_{s \in S} p(s_t = s | I_t) r(s,a) + \beta \sum_{o \in O} p(o_{t+1} = o | I_t, a) V_{n+1}^*(I_{t+1})] \\
&= \max_{a \in A}[\sum_{s \in S} b_t(s) r(s,a) + \beta \sum_{o \in O} p(o_{t+1} = o | I_t, a) V_{n+1}^*(b_{t+1})] \\
&= \max_{a \in A}[\sum_{s \in S} b_t(s) r(s,a) + \beta \sum_{o \in O} p(o_{t+1} = o \mid I_t, a) V_{n+1}^*(b_o^a)] \\
&= \max_{a \in A}[\sum_{s \in S} b_t(s) r(s,a) + \beta \sum_{o \in O} \sum_{s' \in s, s'' \in S} P_O(o \mid s'', a) \\
&\quad \cdot P_S(s'' \mid s', a) b_t(s') V_{n+1}^*(b_o^a)]
\end{aligned}
$$

Value Functions on Belief States (3)

Proof.

For $n < T_F$, it holds that

$$
\begin{aligned}
V_n^*(I_t) &= \max_{a\in A}[\sum_{s\in S} p(s_t = s|I_t)r(s,a) + \beta \sum_{o\in O} p(o_{t+1} = o|I_t, a)V_{n+1}^*(I_{t+1})] \\
&= \max_{a\in A}[\sum_{s\in S} b_t(s)r(s,a) + \beta \sum_{o\in O} p(o_{t+1} = o|I_t, a)V_{n+1}^*(b_{t+1})] \\
&= \max_{a\in A}[\sum_{s\in S} b_t(s)r(s,a) + \beta \sum_{o\in O} p(o_{t+1} = o \mid I_t, a)V_{n+1}^*(b_o^a)] \\
&= \max_{a\in A}[\sum_{s\in S} b_t(s)r(s,a) + \beta \sum_{o\in O} \sum_{s'\in s, s''\in S} P_O(o \mid s'', a) \\
&\quad \cdot P_S(s'' \mid s', a)b_t(s')V_{n+1}^*(b_o^a)] \\
&= V_n^*(b_t)
\end{aligned}
$$

Value Iteration on Belief States

Value Functions on Belief States

Compute sequence of value functions $(V_n)_{0 \leq n \leq T_F}$ defined on belief states, $V_n : B \longrightarrow \mathbb{R}$

Value Iteration on Belief States

Value Functions on Belief States

Compute sequence of value functions $(V_n)_{0 \leq n \leq T_F}$ defined on belief states, $V_n : B \longrightarrow \mathbb{R}$

Algorithm

1. Initialization

$$V_{T_F}(b) = \sum_{s \in S} b(s) r(s, \cdot)$$

Value Iteration on Belief States

Value Functions on Belief States

Compute sequence of value functions $(V_n)_{0 \leq n \leq T_F}$ defined on belief states, $V_n : B \to \mathbb{R}$

Algorithm

1. Initialization

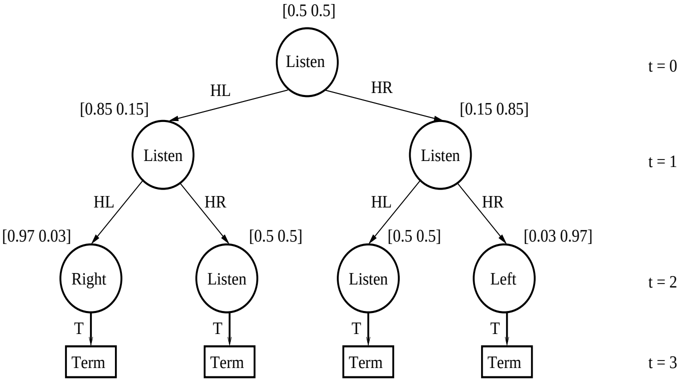
$$V_{T_F}(b) = \sum_{s \in S} b(s) r(s, \cdot)$$

2. Bellman Equation

$$V_n^*(b) := \max_{a \in A} [\sum_{s \in S} b(s) r(s, a) + \beta \sum_{o \in O} \sum_{s' \in s, s'' \in S} P_O(o \mid s'', a) \cdot P_S(s'' \mid s', a) b(s') V_{n+1}^*(b_o^a)]$$

Optimal Policy for $T_F = 3$

Optimal Policy Tree



General Value Functions for POMDPs

Question What about non-uniform initial beliefs over states?

General Value Functions for POMDPs

Question What about non-uniform initial beliefs over states?

Short Answer The optimal policy tree may change!

General Value Functions for POMDPs

Question What about non-uniform initial beliefs over states?

Short Answer The optimal policy tree may change!

Long Answer There may be several regions B_r partitioning the belief space B such that a policy tree P_r is optimal within region B_r

General Value Functions for POMDPs

Question What about non-uniform initial beliefs over states?

Short Answer The optimal policy tree may change!

Long Answer There may be several regions B_r partitioning the belief space B such that a policy tree P_r is optimal within region B_r

General Value Functions

Let V_r be the optimal value function for policy tree P_r

$$V^*(b) = \max_{B_r} V_r(b)$$

Representing Value Functions by Vector Sets

Lemma

Let b_s be the belief state assigning probability $p = 1$ to state $s \in S$. Thus, it holds that

$$V(b) = \sum_{s \in S} b(s) V(b_s)$$

Representing Value Functions by Vector Sets

Lemma

Let b_s be the belief state assigning probability $p = 1$ to state $s \in S$. Thus, it holds that

$$V(b) = \sum_{s \in S} b(s) V(b_s)$$

Vector Representation of V^*

$$V^*(b) = \max_{B_r} V_r(b)$$

Representing Value Functions by Vector Sets

Lemma

Let b_s be the belief state assigning probability $p = 1$ to state $s \in S$. Thus, it holds that

$$V(b) = \sum_{s \in S} b(s) V(b_s)$$

Vector Representation of V^*

$$\begin{aligned} V^*(b) &= \max_{B_r} V_r(b) \\ &= \max_{B_r} \sum_{s \in S} b(s) V_r(b_s) \end{aligned}$$

Representing Value Functions by Vector Sets

Lemma

Let b_s be the belief state assigning probability $p = 1$ to state $s \in S$. Thus, it holds that

$$V(b) = \sum_{s \in S} b(s) V(b_s)$$

Vector Representation of V^*

$$\begin{aligned} V^*(b) &= \max_{B_r} V_r(b) \\ &= \max_{B_r} \sum_{s \in S} b(s) V_r(b_s) \\ &= \max_{B_r} b * \alpha_r \text{ [with } \alpha_r(s) := V_r(b_s)] \end{aligned}$$

General Value Iteration for POMDPs

Theorem

Given a POMDP $(T, S, A, O, P_S, P_O, r)$ *and a finite horizon* T_F*, each value function from the sequence of optimal value functions* $(V_n^*)_{(0 \leq n < T_F)}$ *can be represented by a finite set of vectors* Γ_n*.*

General Value Iteration for POMDPs

Theorem

Given a POMDP $(T, S, A, O, P_S, P_O, r)$ and a finite horizon T_F, each value function from the sequence of optimal value functions $(V_n^*)_{(0 \le n < T_F)}$ can be represented by a finite set of vectors Γ_n.

$$\Gamma_n := \{\sum_{o \in O} \alpha_{f(o)}^{o,a} \mid f \in f(O, \Gamma_{n+1}),\ a \in A\}$$

The symbol $f(O, \Gamma_{n+1})$ denotes the set of possible mappings from observations to vectors from Γ_{n+1}

General Value Iteration for POMDPs

Theorem

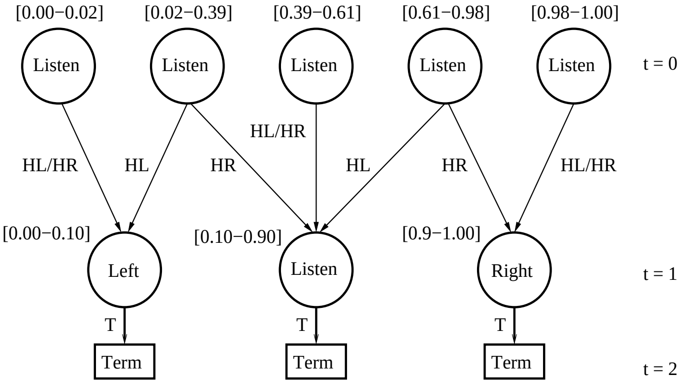
Given a POMDP $(T, S, A, O, P_S, P_O, r)$ and a finite horizon T_F, each value function from the sequence of optimal value functions $(V_n^*)_{(0 \leq n < T_F)}$ can be represented by a finite set of vectors Γ_n.

$$\Gamma_n := \{\sum_{o \in O} \alpha_{f(o)}^{o,a} \mid f \in f(O, \Gamma_{n+1}),\ a \in A\}$$

The symbol $f(O, \Gamma_{n+1})$ denotes the set of possible mappings from observations to vectors from Γ_{n+1}

$$\forall \gamma \in \Gamma_{n+1} : \alpha_\gamma^{o,a}(s') := \frac{r(s', a)}{|O|} + \beta \sum_{s \in S} P_O(o \mid s, a) P_S(s \mid s', a) \gamma(s)$$

General Optimal Policy for $T_F = 2$



Computational Complexity

Theorem

There exists a family of POMDPs such that, for every m, $|S| = 2m$, $|A| = 1$, $|O| = m$, *it exists* $|\Gamma_n| = 2$ *and* $|\Gamma_{n-1}| = 2^m$

Computational Complexity

Theorem

There exists a family of POMDPs such that, for every m, $|S| = 2m$, $|A| = 1$, $|O| = m$, *it exists* $|\Gamma_n| = 2$ *and* $|\Gamma_{n-1}| = 2^m$

$\Rightarrow$ Solving POMDPs exactly is fundamentally inefficient!