Probabilistic planning under partial observability

- Based on stochastic transition systems $\langle S, A, p, R \rangle$ like with full observability
- Computational properties:
  - Belief states are probability distributions on the state space.
  - Belief space is continuous and infinite.
  - *Finite* optimal plans do not always exist.
  - Testing existence of plans with value $\geq c$ undecidable.

Applications

There are many important applications.

- Diagnosis (medical, fault, ...)
- Many applications in economics
- Robotics
- Game playing, problem solving
- Almost everything : -)

Belief states: example

State space $S = \{s_1, s_2\}$.

Belief states:

- everything between $\langle 0, 1 \rangle$ and $\langle 1, 0 \rangle$, e.g. $\langle 0.9, 0.1 \rangle$ and $\langle 0.8, 0.2 \rangle$.
- Contrast to the non-probabilistic case with only 3 (non-empty) belief states $\{s_1\}, \{s_2\}, \{s_1, s_2\}$.

Example: value functions

Actions a_1, a_2 and a_3 do nothing (i.e. $p(s|s, a_i) = 1.0$ for all $i \in \{1, 2, 3\}$ and $s \in S$) and have rewards

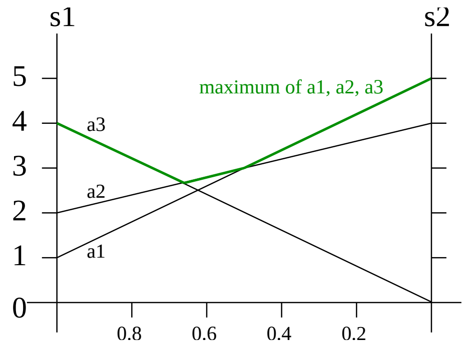
$$\begin{array}{rcl@{\qquad}rcl}
R(a_1, s_1) & = & 1.0 & R(a_1, s_2) & = & 5.0 \\
R(a_2, s_1) & = & 2.0 & R(a_2, s_2) & = & 4.0 \\
R(a_3, s_1) & = & 4.0 & R(a_3, s_2) & = & 0.0
\end{array}$$

Expected reward of a_1 in belief state B s.t. $B(s_1) = 0.7$ and $B(s_2) = 0.3$ is $0.7 \cdot 1.0 + 0.3 \cdot 5.0 = 2.2$.

Form of value functions

- Value functions represented by finite sets of actions/plans are *piecewise linear* and *convex*. *(diagram on the next slide)*
- Optimal value function is convex but not necessarily piecewise linear because it may consist of an infinite number of plans.
- Belief states with high probability on some states have higher value than ones with more even probabilities: *less uncertainty* $\Rightarrow$ *possible to take useful actions* (higher expected rewards).

Example: form of value functions



a_1 is best between $\langle\frac{1}{2},\frac{1}{2}\rangle$ and $\langle 0,1\rangle$.

a_2 is best between $\langle\frac{2}{3},\frac{1}{3}\rangle$ and $\langle\frac{1}{2},\frac{1}{2}\rangle$.

a_3 is best between $\langle 1,0\rangle$ and $\langle\frac{2}{3},\frac{1}{3}\rangle$.

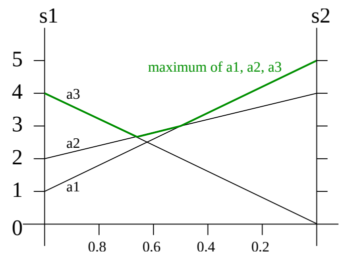
Representation of value functions

A value function V is represented as a set of vectors $\langle v_1,\ldots,v_n\rangle$ that indicate the value of an action/plan in every state $s \in S = \{s_1,\ldots,s_n\}$.

Value of a belief state B (a probability distribution on S) is

$$\max_{\langle v_1,\ldots,v_n\rangle\in V}\left(\sum_{i\in\{1,\ldots,n\}} B(s_i)\cdot v_i\right)$$

Representation of value functions: example

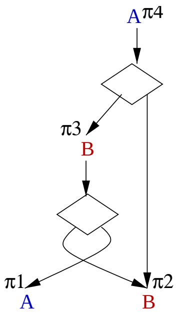


- Value function as *a set of vectors*: $\{\langle 1,5\rangle,\langle 2,4\rangle,\langle 4,0\rangle\}$.
- Each vector indicates the value of a plan in every state.

Plans: example

Plans are written as $(a, \pi_1, \pi_2, \ldots, \pi_m)$ where a is an action and m is the number of observational classes.

$$\begin{aligned}
\pi_1 &= (A, (), ()) \\
\pi_2 &= (B, (), ()) \\
\pi_3 &= (B, \pi_2, \pi_1) \\
\pi_4 &= (A, \pi_3, \pi_2)
\end{aligned}$$



Value of a plan in a state

$\langle C_1, \ldots, C_m \rangle$ is the partition of S to observational classes.

Values of finite acyclic plans π in states $s \in S$ is defined as

$$\begin{aligned}
v_{(),s} &= 0 \qquad \text{(base case: the empty plan)} \\
v_{(a,\pi_1,\ldots,\pi_m),s} &= \begin{cases} -\infty \text{ if action } a \text{ is not applicable in } s \\ R(s,a) + \\ \lambda(\sum_{s' \in C_1} p(s'|s,a) v_{\pi_1,s'} + \cdots + \sum_{s' \in C_m} p(s'|s,a) v \end{cases}
\end{aligned}$$

Value vector $\langle v_{\pi,s_1}, v_{\pi,s_2}, \ldots, v_{\pi,s_n} \rangle$ for states $S = \{s_1, s_2, \ldots, s_n\}$

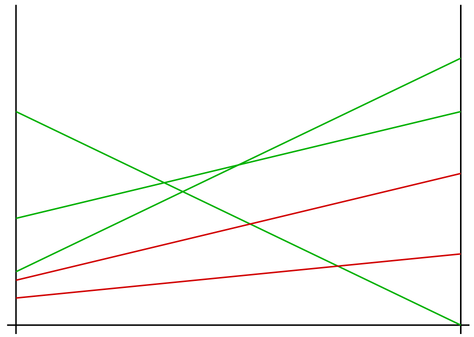
Value of a plan in a state: example

Let s_2 belong to the first observational class and s_3 to the second, let discount factor be 0.96 and let

$$\begin{aligned}
R(s,a) &= 50 & & \\
p(s_2|s,a) &= 0.3 & p(s_3|s,a) &= 0.7 \\
v_{\pi_A,s_2} &= 10 & v_{\pi_B,s_3} &= 20
\end{aligned}$$

Now $v_{(a,\pi_A,\pi_B),s} = 50 + 0.96(0.3 \cdot 10 + 0.7 \cdot 20) = 66.32$.

Dominated plans



Dominated plans: identification by LP

Test whether plan π is for at least one belief state strictly better than any other plan in $\Pi = \{\pi_1, \dots, \pi_n\}$.

Variables are d and p_s for every $s \in S$. Value of d is to be maximized. Constants $v_{\pi,s}$ are values of plans π in states $s \in S$.

$$\begin{aligned} \textstyle\sum_{s\in S} p_s v_{\pi,s} &\geq \textstyle\sum_{s\in S} p_s v_{\pi',s} + d \text{ for all } \pi' \in \Pi\backslash\{\pi\} \\ \textstyle\sum_{s\in S} p_s &= 1 \\ p_s &\geq 0 \text{ for all } s \in S \end{aligned}$$

If the maximum value of d is > 0, then there is a belief state in which the value of π is higher than the value of any other plan.

Dominated plans: Identification, example

For s_1 and s_2 and value vectors $v_{\pi_1} = \langle 1, 5\rangle, v_{\pi_2} = \langle 2, 4\rangle, v_{\pi_3} = \langle 4, 0\rangle$, the following LP tests whether π_1 is somewhere better than π_2 and π_3.

maximize d subject to

$$\begin{aligned} 1p_{s_1} + 5p_{s_2} &\geq 2p_{s_1} + 4p_{s_2} + d \\ 1p_{s_1} + 5p_{s_2} &\geq 4p_{s_1} + 0p_{s_2} + d \\ p_{s_1} + p_{s_2} &= 1 \\ p_{s_1} &\geq 0 \\ p_{s_2} &\geq 0 \end{aligned}$$

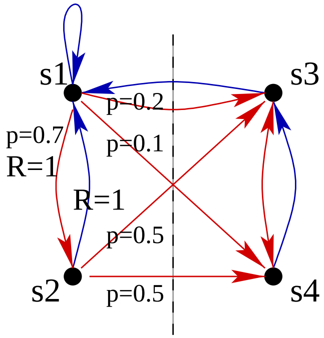
The value iteration algorithm: outline

1. $i := 0$ (*value function for $i = 0$ assigns 0 to all states*.)
2. $i := i + 1$
3. Construct all plans of depth i.
4. Compute the value vectors of the plans.
5. Remove all value vectors dominated by the rest.
6. If the last two value function differ by $> \epsilon$, go to 2.

The value iteration algorithm

1. $i := 0$
2. $\Pi_0 := \{()\}$
3. $i := i + 1$
4. $\Pi_i := \{(a, \pi_1, \dots, \pi_n) | a \in A, \{\pi_1, \dots, \pi_n\} \subseteq \Pi_{i-1}\}$
5. Evaluate the values of plans in Π_i in all states.
6. As long as there is $\pi \in \Pi_i$ that is dominated by $\Pi_i\backslash\{\pi\}$, set $\Pi_i := \Pi_i\backslash\{\pi\}$.
7. If the difference between value functions represented by Π_i and Π_{i-1} is $> \epsilon$ for some belief state, go to 3.

The value iteration algorithm: example



Example: plans of depth 1, value vectors

We use the discount constant $\lambda = 0.5$.

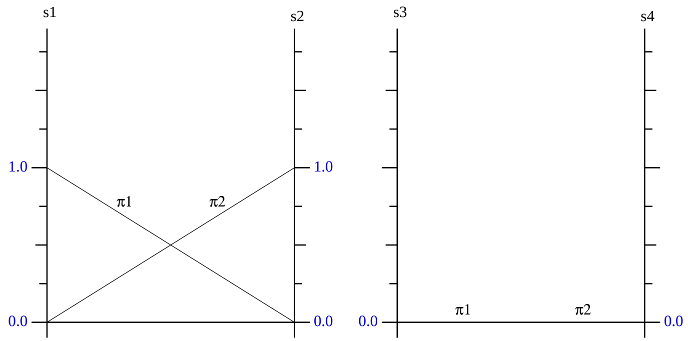
Plans of depth 1 with the corresponding value vectors for all states $S = \{s_1, s_2, s_3, s_4\}$ are the following.

$$\begin{aligned} \pi_1 &= (\text{R}, (), ()) \\ \pi_2 &= (\text{B}, (), ()) \end{aligned}$$

The values of these plans in states s_1, s_2, s_3, s_4 are as follows.

$$\begin{aligned} v_{\pi_1} &= \langle \mathbf{1.0}, \mathbf{0.0}, \mathbf{0.0}, \mathbf{0.0} \rangle \\ v_{\pi_2} &= \langle \mathbf{0.0}, \mathbf{1.0}, \mathbf{0.0}, \mathbf{0.0} \rangle \end{aligned}$$

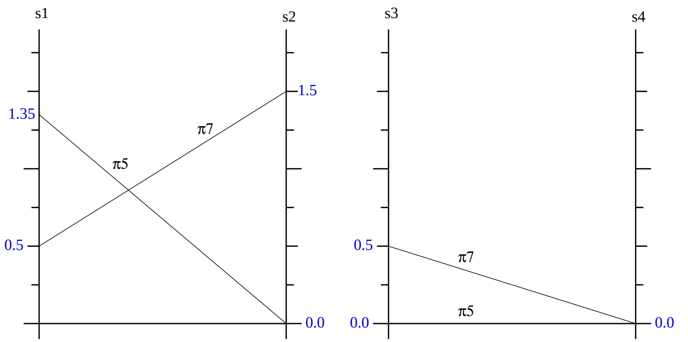
Example: value function at iteration 1



Example: values of plans of depth 2

$$\begin{aligned}
\pi_3 &= (\text{R}, \pi_1, \pi_1) & v_{\pi_3} &= \langle 1.0, 0.0, 0.0, 0.0 \rangle \\
\pi_4 &= (\text{R}, \pi_1, \pi_2) & v_{\pi_4} &= \langle 1.0, 0.0, 0.0, 0.0 \rangle \\
\pi_5 &= (\text{R}, \pi_2, \pi_1) & v_{\pi_5} &= \langle \mathbf{1.35}, \mathbf{0.0}, \mathbf{0.0}, \mathbf{0.0} \rangle \\
\pi_6 &= (\text{R}, \pi_2, \pi_2) & v_{\pi_6} &= \langle 1.35, 0.0, 0.0, 0.0 \rangle \\
\pi_7 &= (\text{B}, \pi_1, \pi_1) & v_{\pi_7} &= \langle \mathbf{0.5}, \mathbf{1.5}, \mathbf{0.5}, \mathbf{0.0} \rangle \\
\pi_8 &= (\text{B}, \pi_1, \pi_2) & v_{\pi_8} &= \langle 0.5, 1.5, 0.5, 0.0 \rangle \\
\pi_9 &= (\text{B}, \pi_2, \pi_1) & v_{\pi_9} &= \langle 0.0, 1.0, 0.0, 0.0 \rangle \\
\pi_{10} &= (\text{B}, \pi_2, \pi_2) & v_{\pi_{10}} &= \langle 0.0, 1.0, 0.0, 0.0 \rangle
\end{aligned}$$

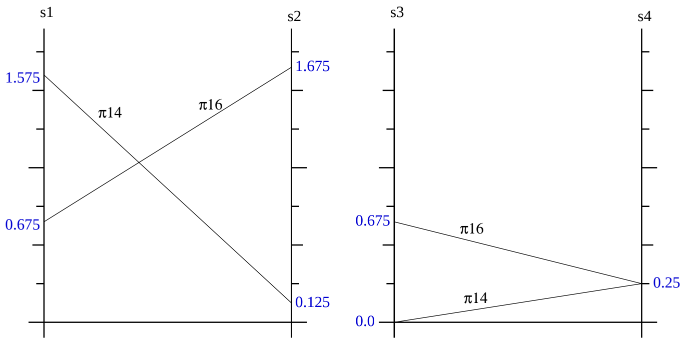
Example: value function at iteration 2



Example: values of plans of depth 3

$$
\begin{array}{lcllcl}
\pi_{11} & = & (\mathsf{R}, \pi_5, \pi_5) & v_{\pi_{11}} & = & \langle 1.0, 0.0, 0.0, 0.0 \rangle \\
\pi_{12} & = & (\mathsf{R}, \pi_5, \pi_7) & v_{\pi_{12}} & = & \langle 1.05, 0.125, 0.0, 0.25 \rangle \\
\pi_{13} & = & (\mathsf{R}, \pi_7, \pi_5) & v_{\pi_{13}} & = & \langle 1.525, 0.0, 0.0, 0.0 \rangle \\
\pi_{14} & = & (\mathsf{R}, \pi_7, \pi_7) & v_{\pi_{14}} & = & \langle \mathbf{1.575}, \mathbf{0.125}, \mathbf{0.0}, \mathbf{0.25} \rangle \\
\pi_{15} & = & (\mathsf{B}, \pi_5, \pi_5) & v_{\pi_{15}} & = & \langle 0.675, 1.675, 0.675, 0.0 \rangle \\
\pi_{16} & = & (\mathsf{B}, \pi_5, \pi_7) & v_{\pi_{16}} & = & \langle \mathbf{0.675}, \mathbf{1.675}, \mathbf{0.675}, \mathbf{0.25} \rangle \\
\pi_{17} & = & (\mathsf{B}, \pi_7, \pi_5) & v_{\pi_{17}} & = & \langle 0.25, 1.25, 0.25, 0.0 \rangle \\
\pi_{18} & = & (\mathsf{B}, \pi_7, \pi_7) & v_{\pi_{18}} & = & \langle 0.25, 1.25, 0.25, 0.25 \rangle
\end{array}
$$

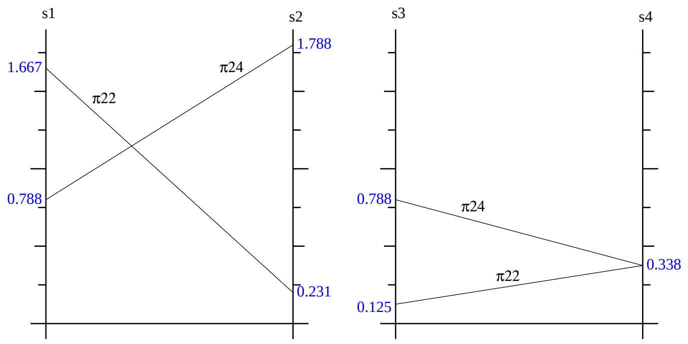
Example: value function at iteration 3



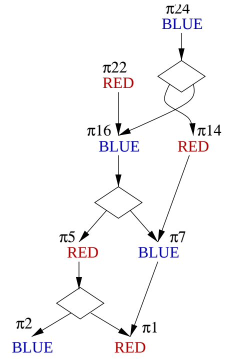
Example: values of plans of depth 4

$$
\begin{array}{lcllcl}
\pi_{19} & = & (\mathsf{R}, \pi_{14}, \pi_{14}) & v_{\pi_{19}} & = & \langle 1.05625, 0.0625, 0.125, 0.0 \rangle \\
\pi_{20} & = & (\mathsf{R}, \pi_{14}, \pi_{16}) & v_{\pi_{20}} & = & \langle 1.12375, 0.23125, 0.125, 0.3375 \rangle \\
\pi_{21} & = & (\mathsf{R}, \pi_{16}, \pi_{14}) & v_{\pi_{21}} & = & \langle 1.59875, 0.0625, 0.125, 0.0 \rangle \\
\pi_{22} & = & (\mathsf{R}, \pi_{16}, \pi_{16}) & v_{\pi_{22}} & = & \langle \mathbf{1.66625}, \mathbf{0.23125}, \mathbf{0.125}, \mathbf{0.3375} \rangle \\
\pi_{23} & = & (\mathsf{B}, \pi_{14}, \pi_{14}) & v_{\pi_{23}} & = & \langle 0.7875, 1.7875, 0.7875, 0.0 \rangle \\
\pi_{24} & = & (\mathsf{B}, \pi_{14}, \pi_{16}) & v_{\pi_{24}} & = & \langle \mathbf{0.7875}, \mathbf{1.7875}, \mathbf{0.7875}, \mathbf{0.3375} \rangle \\
\pi_{25} & = & (\mathsf{B}, \pi_{16}, \pi_{14}) & v_{\pi_{25}} & = & \langle 0.3375, 1.3375, 0.3375, 0.0 \rangle \\
\pi_{26} & = & (\mathsf{B}, \pi_{16}, \pi_{16}) & v_{\pi_{26}} & = & \langle 0.3375, 1.3375, 0.3375, 0.3375 \rangle
\end{array}
$$

Example: value function at iteration 4



Example: plan for horizon length 4



Use of plans like in the FO case:

1. Choose action by executing the plan from the beginning.
2. Compute the new belief state by using the observation and the probabilities.
3. Continue from 1. (This is known as *receding-horizon control*)

- When horizon really is finite, execute the plan in the normal way.

Comments on the algorithm

- The algorithm we described can easily be extended with *sensory uncertainty.*
- There are many improvements to the generation and pruning of the value vectors.
- Algorithms for planning with partial observability is an active research topic: how to scale up to big state spaces?