

Decision-Making under Uncertainty

Mansur Arief
Stanford Intelligent Systems Lab

POMDP: A systematic framework for decision-making under uncertainty

- POMDP: Partially Observable Markov Decision Process
- Defined by the following:

Variable	Description
S	State space
A	Action space
\mathcal{O}	Observation space
$T(s' s, a)$	Transition function
$R(s, a)$	Reward function
$O(o s')$	Observation function
$\gamma \in [0, 1]$	Discount factor

POMDP Example: Crying baby problem

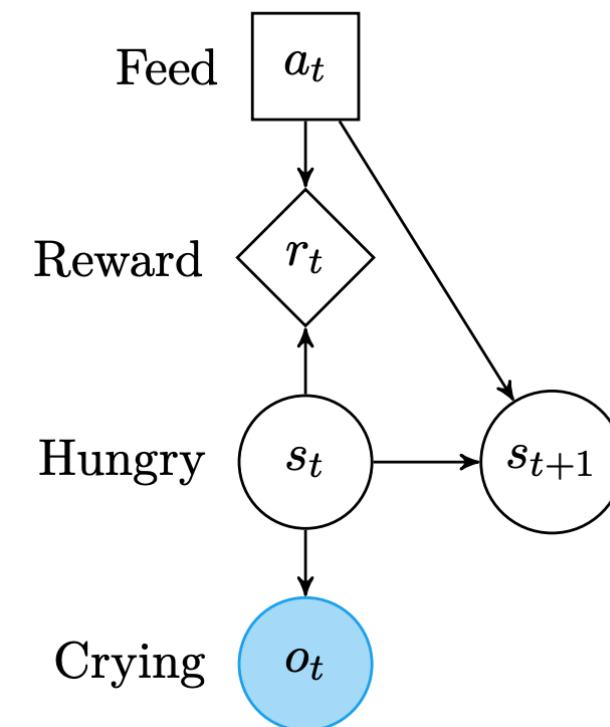
- A simple POMDP with 2 states, 2 actions, and 2 observations

$$\mathcal{S} = \{\text{hungry}, \text{full}\}$$

$$\mathcal{A} = \{\text{feed}, \text{ignore}\}$$

$$\mathcal{O} = \{\text{crying}, \text{quiet}\}$$

- We **cannot** directly tell if the baby is **truly hungry**
- We **can only observe** the **crying** and update our belief about the true state using this information.



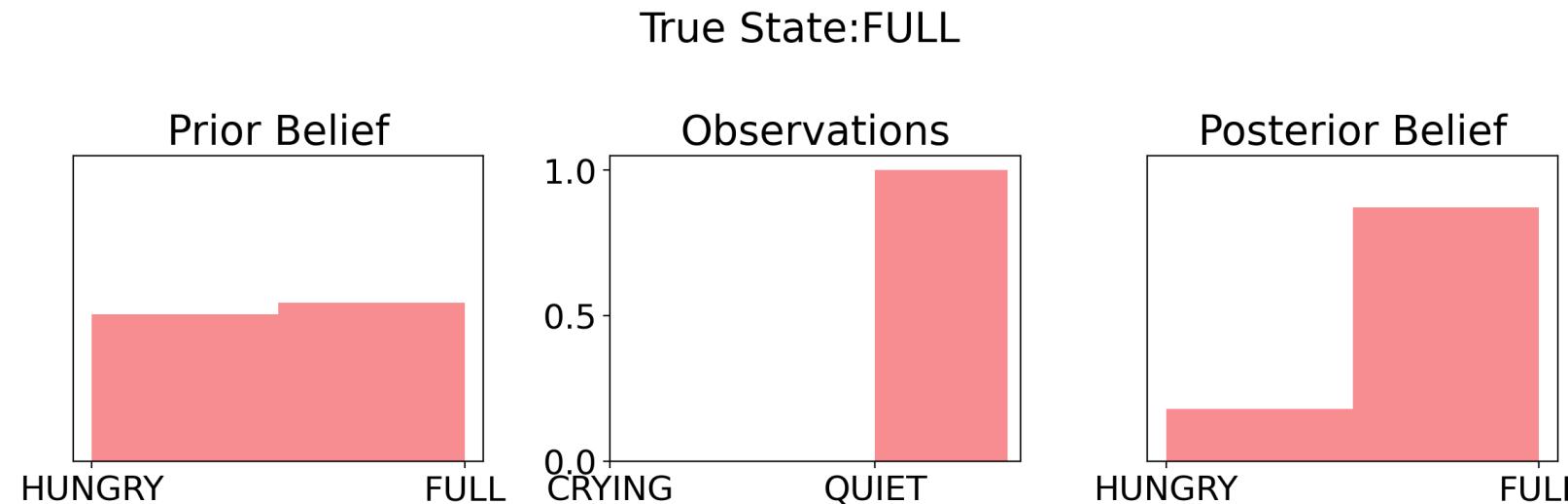
POMDP Example: Crying baby problem

- Suppose we have the following state uncertainty

$$\begin{aligned}O(\text{crying} \mid \text{hungry}) &= 80\% \\O(\text{quiet} \mid \text{hungry}) &= 20\%\end{aligned}$$

$$\begin{aligned}O(\text{crying} \mid \text{full}) &= 10\% \\O(\text{quiet} \mid \text{full}) &= 90\%\end{aligned}$$

- We can start with some prior belief and update it as we observe data



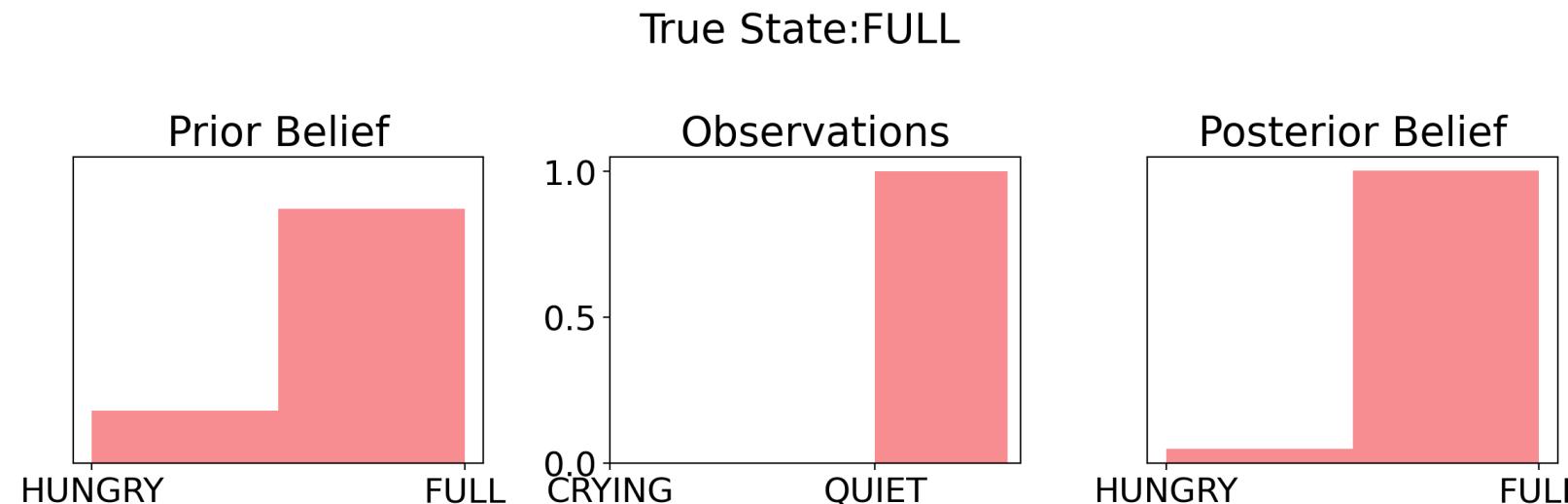
POMDP Example: Crying baby problem

- Suppose we have the following state uncertainty

$$\begin{aligned}O(\text{crying} \mid \text{hungry}) &= 80\% \\O(\text{quiet} \mid \text{hungry}) &= 20\%\end{aligned}$$

$$\begin{aligned}O(\text{crying} \mid \text{full}) &= 10\% \\O(\text{quiet} \mid \text{full}) &= 90\%\end{aligned}$$

- We can start with some prior belief and update it as we observe data



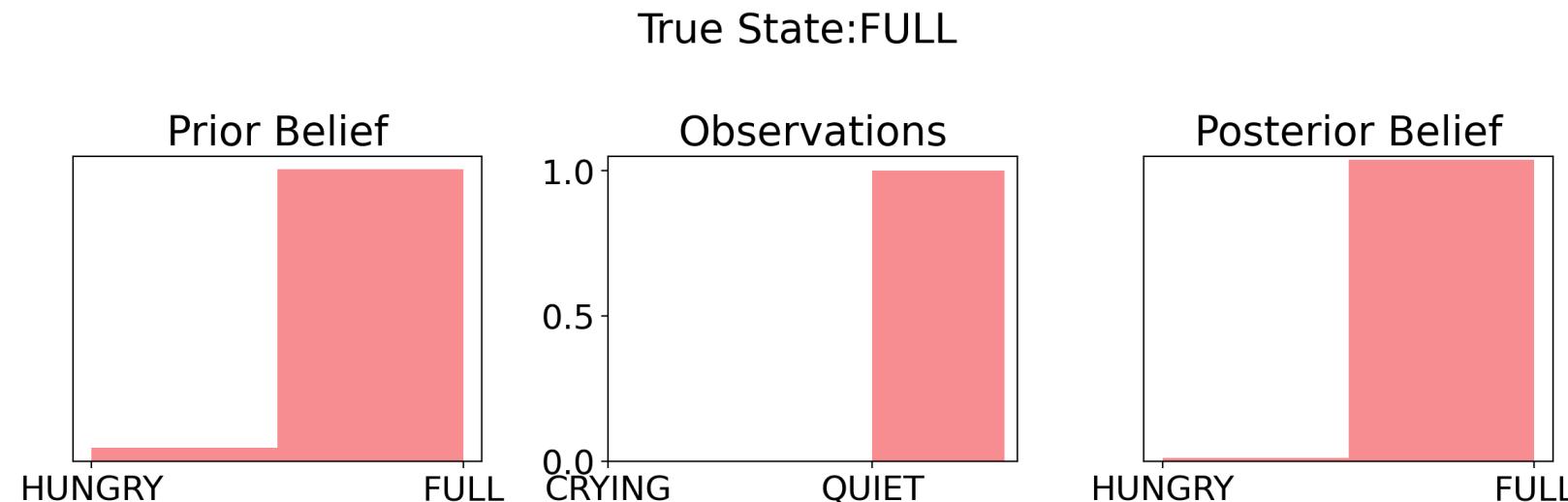
POMDP Example: Crying baby problem

- Suppose we have the following state uncertainty

$$\begin{aligned}O(\text{crying} \mid \text{hungry}) &= 80\% \\O(\text{quiet} \mid \text{hungry}) &= 20\%\end{aligned}$$

$$\begin{aligned}O(\text{crying} \mid \text{full}) &= 10\% \\O(\text{quiet} \mid \text{full}) &= 90\%\end{aligned}$$

- We can start with some prior belief and update it as we observe data



Solving a POMDP model

```
using POMDPs, POMDPModelTools, QuickPOMDPs

@enum State hungry full
@enum Action feed ignore
@enum Observation crying quiet

pomdp = QuickPOMDP(
    states      = [hungry, full], # S
    actions     = [feed, ignore], # A
    observations = [crying, quiet], # O
    initialstate = [full], # Deterministic
    discount     = 0.9, # γ

    transition = function T(s, a)
        if a == feed
            return SparseCat([hungry, full], [0, 1])
        elseif s == hungry && a == ignore
            return SparseCat([hungry, full], [1, 0])
        elseif s == full && a == ignore
            return SparseCat([hungry, full], [0.1, 0.9])
        end
    end,

    observation = function O(s, a, s')
        if s' == hungry
            return SparseCat([crying, quiet], [0.8, 0.2])
        elseif s' == full
            return SparseCat([crying, quiet], [0.1, 0.9])
        end
    end,
    reward = (s,a)→(s == hungry ? -10 : 0) + (a == feed ? -5 : 0)
)
```

POMDP solvers

Package	State Spaces	Actions Spaces	Observation Spaces
QMDP.jl	Discrete	Discrete	Discrete
FIB.jl	Discrete	Discrete	Discrete
BeliefGridViewIteration.jl	Discrete	Discrete	Discrete
SARSOP.jl	Discrete	Discrete	Discrete
BasicPOMCP.jl	Continuous	Discrete	Discrete
ARDESPOT.jl	Continuous	Discrete	Discrete
MCVI.jl	Continuous	Discrete	Continuous
POMDPSolve.jl	Discrete	Discrete	Discrete
IncrementalPruning.jl	Discrete	Discrete	Discrete
POMCPOW.jl	Continuous	Continuous	Continuous
AEMS.jl	Discrete	Discrete	Discrete
PointBasedValueIteration.jl	Discrete	Discrete	Discrete

Solving a POMDP model

```
using POMDPs, POMDPModelTools, QuickPOMDPs

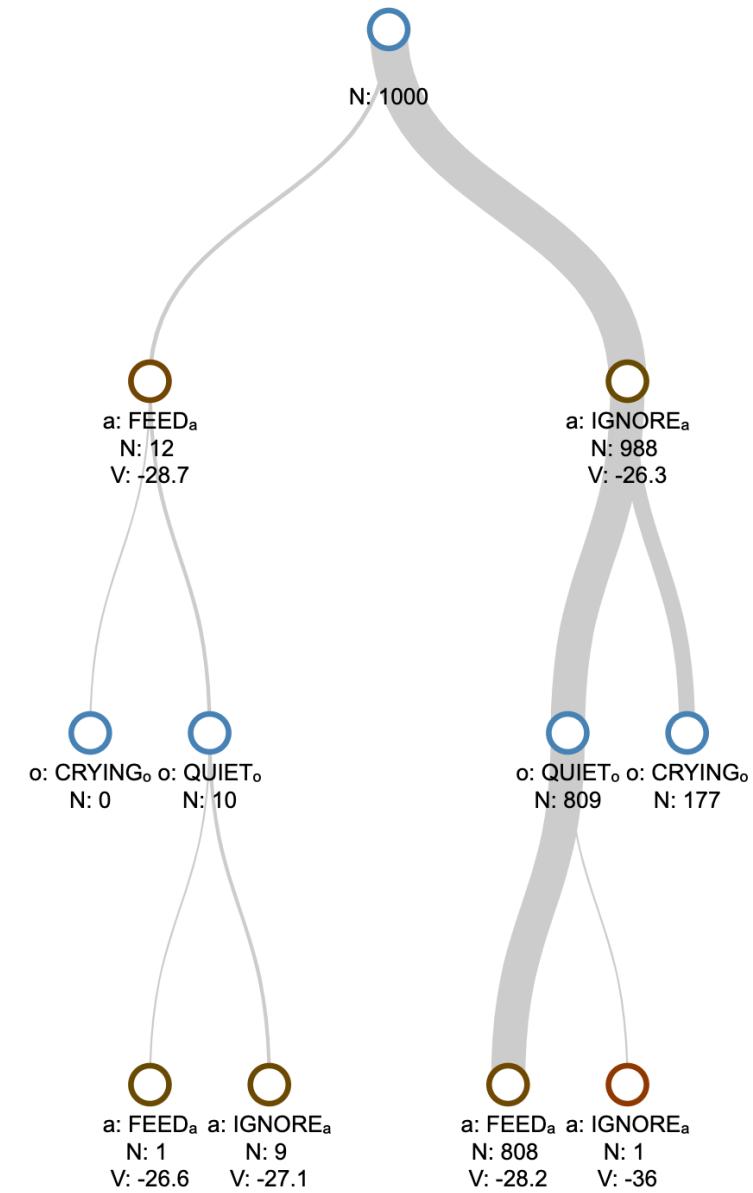
@enum State hungry full
@enum Action feed ignore
@enum Observation crying quiet

pomdp = QuickPOMDP(
    states      = [hungry, full], # S
    actions     = [feed, ignore], # A
    observations = [crying, quiet], # O
    initialstate = [full], # Deterministic
    discount     = 0.9, # γ

    transition = function T(s, a)
        if a == feed
            return SparseCat([hungry, full], [0, 1])
        elseif s == hungry && a == ignore
            return SparseCat([hungry, full], [1, 0])
        elseif s == full && a == ignore
            return SparseCat([hungry, full], [0.1, 0.9])
        end
    end,

    observation = function O(s, a, s')
        if s' == hungry
            return SparseCat([crying, quiet], [0.8, 0.2])
        elseif s' == full
            return SparseCat([crying, quiet], [0.1, 0.9])
        end
    end,

    reward = (s,a)→(s == hungry ? -10 : 0) + (a == feed ? -5 : 0)
)
```



Solving a POMDP model

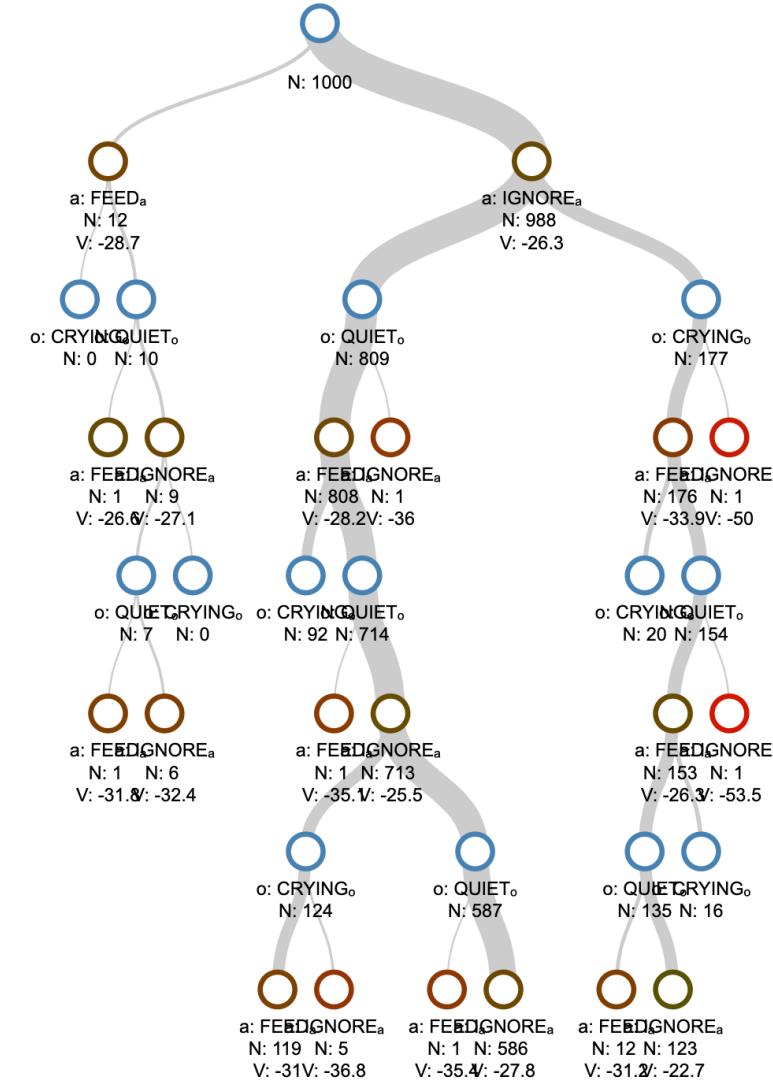
```
using POMDPs, POMDPModelTools, QuickPOMDPs

@enum State hungry full
@enum Action feed ignore
@enum Observation crying quiet

pomdp = QuickPOMDP(
    states      = [hungry, full], # S
    actions     = [feed, ignore], # A
    observations = [crying, quiet], # O
    initialstate = [full], # Deterministic
    discount     = 0.9, # γ

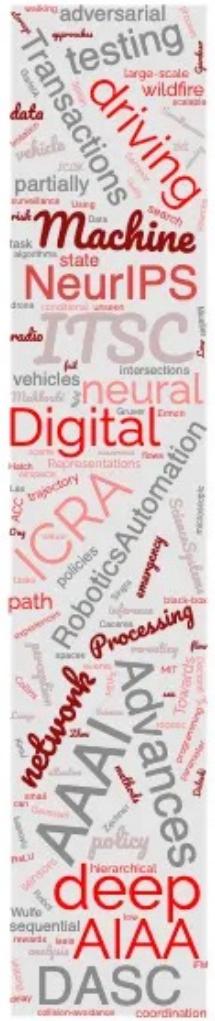
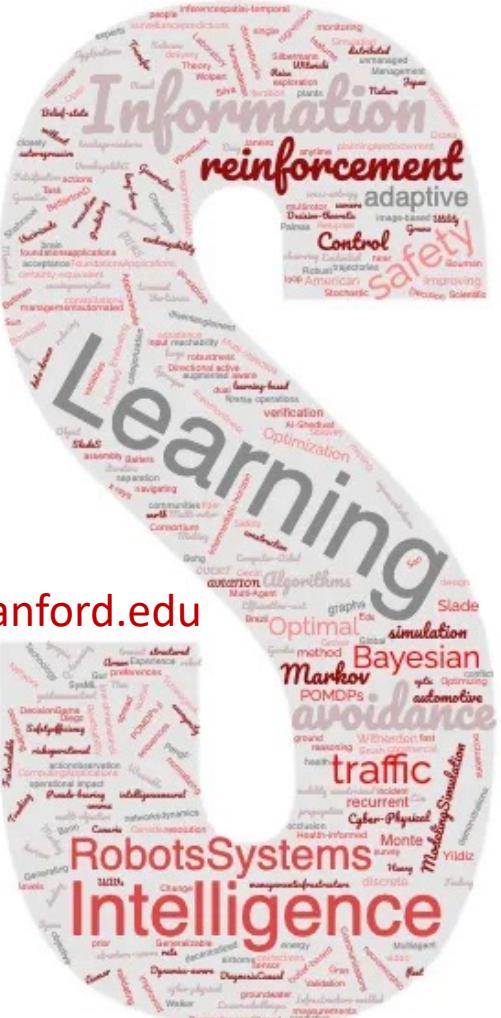
    transition = function T(s, a)
        if a == feed
            return SparseCat([hungry, full], [0, 1])
        elseif s == hungry && a == ignore
            return SparseCat([hungry, full], [1, 0])
        elseif s == full && a == ignore
            return SparseCat([hungry, full], [0.1, 0.9])
        end
    end,

    observation = function O(s, a, s')
        if s' == hungry
            return SparseCat([crying, quiet], [0.8, 0.2])
        elseif s' == full
            return SparseCat([crying, quiet], [0.1, 0.9])
        end
    end,
    reward = (s,a)→(s == hungry ? -10 : 0) + (a == feed ? -5 : 0)
)
```



About Stanford Intelligent Systems Lab (SISL)

Web: sisl.stanford.edu



People:
5 postdocs,
14 PhD students,
a few grad students, &
undergrad researchers

A group photo of approximately 18 people, mostly young adults, posing outdoors in front of a building with trees in the background. They are dressed in casual to semi-casual attire, including various shirts, jackets, and jeans. The group is arranged in three rows, with some people sitting in the front and others standing behind them.

Thanks for your time!

SISL Group Photo, <https://sisl.stanford.edu/>