# Probabilistic Decision-Making Under Model Uncertainty

Joelle Pineau

School of Computer Science, McGill University, Canada

Joint work with Mahdi Milani Fard, Peng Sun, Stéphane Ross and Brahim Chaib-draa

# Motivation : A human-robot interaction problem

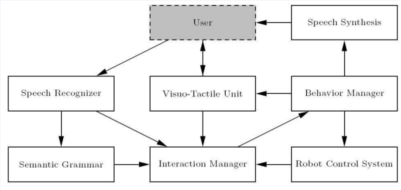
We are currently building robotic systems which must deal with :

- noisy/partial sensing of their environments,
- observations that are discrete/continuous and structured
- poor model of sensors and actuators.

SmartWheeler Platform

[Pineau et al., 2007]

Interaction Architecture




Despite all this, we expect the robot to behave in an engaging and reasonable manner !

# Typical ways of solving such problems :

- Customized solution : Design a script (e.g. finite-state machine) fully describing the possible interactions.
- Supervised Learning : Learn model from data, then plan with the learned model.
- Reinforcement Learning : Learn directly how to act, through trial-and-error interactions with the environment.

# Typical ways of solving such problems :

- Customized solution : Design a script (e.g. finite-state machine) fully describing the possible interactions.
- Supervised Learning : Learn model from data, then plan with the learned model.
- Reinforcement Learning : Learn directly how to act, through trial-and-error interactions with the environment.

# Typical ways of solving such problems :

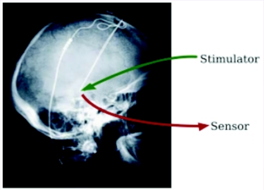
- Customized solution : Design a script (e.g. finite-state machine) fully describing the possible interactions.
- Supervised Learning : Learn model from data, then plan with the learned model.
- Reinforcement Learning : Learn directly how to act, through trial-and-error interactions with the environment.

# Motivation : A treatment design problem

We are also optimizing sequences of medical treatment, which are subject to :

- high-dimensional, noisy input spaces,
- real-time decision-making in diverse environments,
- learning from very small sample sets.

Deep-brain stimulation




[Guez et al., 2008]

Despite all this, we expect the intelligent agent to achieve effective seizure suppresion !

# What do we need for tackling real-world problems ?

- Flexible learning
  - Learning from few data.
  - Online model adaptation.
  - Ability to specify domain knowledge (features, priors, etc.).
- Methods that can deal with :
  - partial state observability,
  - structured representations.
  - complex observations,
- Ability to maximize expected return based on current state of information.

# Partially Observable Markov Decision Processes

## POMDP Model Definition

- $S$ : Set of states (*unobservable by the agent*)
- $A$ : Set of actions
- $T(s, a, s') = \Pr(s'|s, a)$, transition probabilities
- $R(s, a) \in \mathbb{R}$, immediate rewards
- $\gamma$ : discount factor
- $Z$ : **Set of observations**
- $O(s', a, z) = \Pr(z|s', a)$, **the observation probabilities**
- $b_0(s)$ : **Initial state distribution**

Belief monitoring via Bayes rule :

$b_t(s') = \eta O(s', a_{t-1}, z_t) \sum_{s \in S} T(s, a_{t-1}, s') b_{t-1}(s)$

Value function optimization :

$V^*(b) = \max_{a \in A} \left[ R(b, a) + \gamma \sum_{z \in Z} \Pr(z|b, a) V^*(\tau(b, a, z)) \right]$

# Partially Observable Markov Decision Processes

## POMDP Model Definition

- $S$ : Set of states (*unobservable by the agent*)
- $A$ : Set of actions
- $T(s, a, s') = \Pr(s'|s, a)$, transition probabilities
- $R(s, a) \in \mathbb{R}$, immediate rewards
- $\gamma$ : discount factor
- **$Z$ : Set of observations**
- **$O(s', a, z) = \Pr(z|s', a)$, the observation probabilities**
- **$b_0(s)$ : Initial state distribution**

Belief monitoring via Bayes rule :

$$b_t(s') = \eta O(s', a_{t-1}, z_t) \sum_{s \in S} T(s, a_{t-1}, s') b_{t-1}(s)$$

Value function optimization :

$$V^*(b) = \max_{a \in A} \left[ R(b, a) + \gamma \sum_{z \in Z} \Pr(z|b, a) V^*(\tau(b, a, z)) \right]$$

# Partially Observable Markov Decision Processes

## POMDP Model Definition

- $S$ : Set of states (*unobservable by the agent*)
- $A$ : Set of actions
- $T(s, a, s') = \Pr(s'|s, a)$, transition probabilities
- $R(s, a) \in \mathbb{R}$, immediate rewards
- $\gamma$ : discount factor
- **$Z$ : Set of observations**
- $O(s', a, z) = \Pr(z|s', a)$**, the observation probabilities**
- $b_0(s)$ **: Initial state distribution**

Belief monitoring via Bayes rule :

$$b_t(s') = \eta O(s', a_{t-1}, z_t) \sum_{s \in S} T(s, a_{t-1}, s') b_{t-1}(s)$$

Value function optimization :

$$V^*(b) = \max_{a \in A} \left[ R(b, a) + \gamma \sum_{z \in Z} \Pr(z|b, a) V^*(\tau(b, a, z)) \right]$$

# Partially Observable Markov Decision Processes

## POMDP Model Definition

- $S$ : Set of states (*unobservable by the agent*)
- $A$ : Set of actions
- $T(s, a, s') = \Pr(s'|s, a)$, transition probabilities
- $R(s, a) \in \mathbb{R}$, immediate rewards
- $\gamma$ : discount factor
- **$Z$ : Set of observations**
- $O(s', a, z) = \Pr(z|s', a)$**, the observation probabilities**
- $b_0(s)$ **: Initial state distribution**

Belief monitoring via Bayes rule :

$$b_t(s') = \eta O(s', a_{t-1}, z_t) \sum_{s \in S} T(s, a_{t-1}, s') b_{t-1}(s)$$

Value function optimization :

$$V^*(b) = \max_{a \in A} \left[ R(b, a) + \gamma \sum_{z \in Z} \Pr(z|b, a) V^*(\tau(b, a, z)) \right]$$

# Motivation

## Given

- A POMDP problem domain with unknown dynamics.
- The ability to sample trajectories from this domain.

## For today's talk, consider two cases :

1. Assume the trajectories have labeled state information, but you can't control the choice of action → **Batch data**
2. Assume you can control the agent during data collection, but the states are only partially observable → **Online data**

# Motivation

## Given

- A POMDP problem domain with unknown dynamics.
- The ability to sample trajectories from this domain.

## For today's talk, consider two cases :

1. Assume the trajectories have labeled state information, but you can't control the choice of action → **Batch data**
2. Assume you can control the agent during data collection, but the states are only partially observable → **Online data**

# Motivation

## Given

- A POMDP problem domain with unknown dynamics.
- The ability to sample trajectories from this domain.

## For today's talk, consider two cases :

1. Assume the trajectories have labeled state information, but you can't control the choice of action → **Batch data**
2. Assume you can control the agent during data collection, but the states are only partially observable → **Online data**

# Let's start with a simple case

## Given

- A POMDP problem domain with unknown dynamics
- Sample trajectories of **two policies** (with labeled state information)

## Ask

- **Which policy is better ?**
- How **confident** are we in this choice ?

# Robot-Human Interaction Example

## Dialogue management problem

- Human operator issues commands such as :
  - *Go to location X.*
  - *Go to location Y.*
- Robot perceives commands through noisy speech recognition output.
- Robot has the option to either **ask for clarification**, or **go to a given location.**

## SmartWheeler

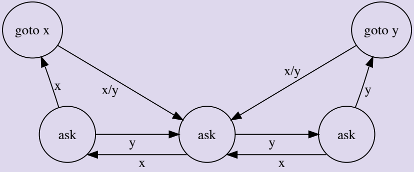
SmartWheeler robotic wheelchair

# Finite State Controller

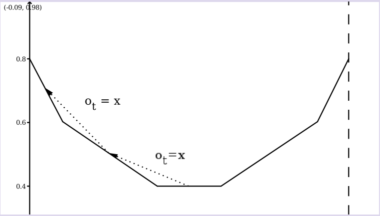
## Quick recap of POMDP methods

- The **policy** is a function mapping belief states to actions.
- The **value** is a function mapping belief states to the expected return of running that policy.
- The value function in the finite horizon is **piecewise linear**.
- A policy can be represented as a **finite state controller**.

## Policy as a Finite State Controller




## Corresponding Value Function



# Finite State Controller

## Policy Evaluation (matrix form)

$$V = R + \gamma T O \Pi V$$
$$V = (I - \gamma T O \Pi)^{-1} R$$

## Definition

- $V$ : coefficients of piecewise linear value function
- $R$ : coefficients of piecewise linear immediate reward
- $T$ : transition model under given policy
- $O$ : observation model under given policy
- **$\Pi$ : state transitions of the finite state controller**

[Sondik, 1971 ; Hansen, 1998]

# Estimating the Variance in the Value Function

## Model Error

- Most POMDP solvers assume perfect $T$ and $O$ models.
- In practice, models are often imperfect estimates.
  - Designed by experts.
  - Estimated using Expectation-Maximization.
  - **Estimated from recorded trajectories with labeled state information.**

# Model Error

## Frequentist Approach to Estimating the Model

$$\hat{T}_a(i,j) = \frac{N_{ij}^a}{N_i^a}, \quad \hat{O}_a(i,j) = \frac{M_{ij}^a}{M_i^a}$$

## Error Terms

- With finite samples :

$$\hat{T} = T + \tilde{T}, \quad \hat{O} = O + \tilde{O}$$

- Assume error terms are unbiased and independent :

$$E[\tilde{T}] = E[\tilde{O}] = E[\tilde{T}\tilde{O}] = 0$$

- Covariance terms can be estimated from data.

# Variance in Value Function

## Empirical Value Function

$$\hat{V} = (I - \gamma \hat{T}\hat{O}\Pi)^{-1}R$$
$$= (I - \gamma(T + \tilde{T})(O + \tilde{O})\Pi)^{-1}R \qquad \textit{Substitute model error}$$
$$= \sum_{k=0}^{\infty} \gamma^k f_k R \qquad \textit{Taylor expansion}$$

Where

$$f_k = (X(\tilde{T}O\Pi + T\tilde{O}\Pi + \tilde{T}\tilde{O}\Pi))^k X$$
$$X = (I - \gamma TO\Pi)^{-1}$$

**We consider a 2nd order approximation of the Taylor series.**

# Error in Value Function Estimate

## First Moment

$$E[\hat{V}] = V + \gamma^2 E[f_2]R$$

## Second Moment

$$E[\hat{V}\hat{V}^T] = VV^T + \gamma^2(E[f_1 RR^T f_1^T]) + \gamma^2(E[f_0 RR^T f_2^T]) + \gamma^2(E[f_2 RR^T f_0^T])$$

## Covariance

$$E[\hat{V}\hat{V}^T] - E[\hat{V}]E[\hat{V}]^T = \gamma^2(E[f_1 RR^T f_1^T])$$

# Dialogue Manager

## Testing Accuracy of Estimates

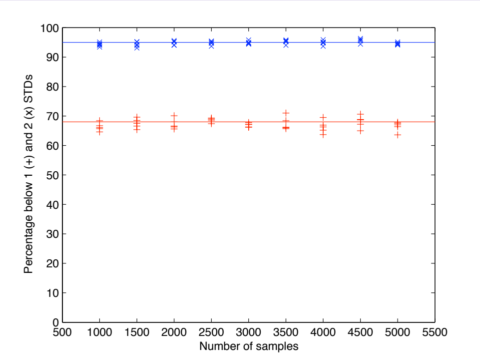
- Fix true models $T$ and $O$
- Generate $N$ test cases
  - each contains fixed number of samples
- For each test case :
  - calculate $\hat{V}(b_0)$.
  - calculate std. dev. over $\hat{V}(b_0)$ using our method.
- Measure how often :
  $|V(b_0) - \hat{V}(b_0)| < 1 * std.dev.$
  $|V(b_0) - \hat{V}(b_0)| < 2 * std.dev.$

## SmartWheeler

SmartWheeler robotic wheelchair

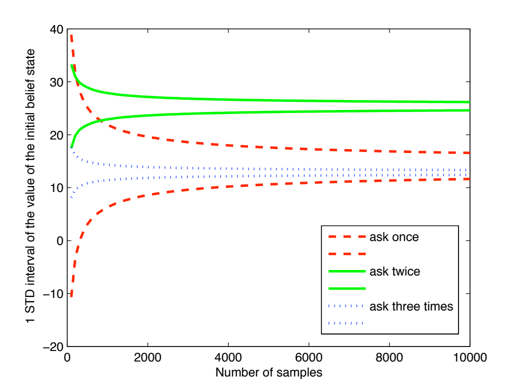
# Dialogue Manager

## Testing Accuracy of Estimates




Percentage of the cases in which $\hat{V}(b_0)$ lies within 1 (+) and 2 ($\times$) approximately calculated standard deviations from $V(b_0)$

# Dialogue Manager

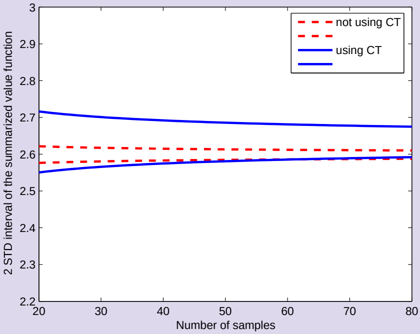



1 standard deviation interval for the calculated value of the initial belief state for different policies

# Comparing treatment strategies for chronic illness

- Goal : optimize treatment design such as to minimize symptom severity.
- Challenges : small data set, different number of samples per treatment.
- Other concern : some treatments may be preferred by some patients.

## Comparing Policies (2 std.dev.)



# Discussion

## Summary

- Using empirical models introduces variance in the calculated value function.
- We provide a way to estimate this variance.
  - Technique presented today is a generalization of earlier work by Mannor et al. (2004) for the MDP case.
- This is useful to quantify performance variation in critical task domains.

## Let's kick it up a notch :

- What if we don't have the state labels ?
- And we have control over how the data is collected ?

# Part 2 :

## Given

- A POMDP problem domain with unknown dynamics.
- The ability to sample trajectories from this domain.

## Let's now consider the second case :

1. Assume the trajectories have labeled state information, but you can't control the choice of action $\rightarrow$ **Batch data**
2. Assume you can control the agent during data collection, but the states are only partially observable $\rightarrow$ **Online data**

# Bayesian Reinforcement Learning

**General Idea** :

- Define prior distributions over all unknown parameters.
- Update posterior via Baye's rule as experience is acquired.
- Optimize action choice w.r.t. posterior distribution over model.

Allows us to :

- Include prior knowledge explicitly.
- Perform learning as necessary to accomplish the task.
- Consider model uncertainty during planning.

# Bayesian Reinforcement Learning

**General Idea** :

- Define prior distributions over all unknown parameters.
- Update posterior via Baye's rule as experience is acquired.
- Optimize action choice w.r.t. posterior distribution over model.

Allows us to :

- Include prior knowledge explicitly.
- Perform learning as necessary to accomplish the task.
- Consider model uncertainty during planning.

# Recall the POMDP model definition

## POMDP model :

- $S$ : Set of states (*unobservable by the agent*)
- $A$ : Set of actions
- $T(s, a, s') = \Pr(s'|s, a)$, transition probabilities
- $R(s, a) \in \mathbb{R}$, immediate rewards
- $\gamma$ : discount factor
- $Z$ : Set of observations
- $O(s', a, z) = \Pr(z|s', a)$, the observation probabilities
- $b_0(s)$ : Initial state distribution

**How should we choose actions if the parameters $T$ and $O$ are uncertain ?**

# Bayesian RL in Finite MDPs

**In Finite MDPs :** ([Dearden et al. 99], [Duff 02], [Poupart et al. 06])

Maintain counts $\phi^{a}_{ss'}$ of number of times the transition $s \xrightarrow{a} s'$ is observed, starting from prior $\phi_0$.

Counts define Dirichlet prior/posterior over $T$.

Planning according to $\phi$ is an MDP problem itself :

- $S'$ : physical state ($s \in S$) + information state ($\phi$)
- $T'$ : describes probability of update $(s, \phi) \xrightarrow{a} (s', \phi')$

# Bayesian RL in Finite POMDPs

**In Finite POMDPs ($T, O$ unknown) :**

Let :

- $\phi^a_{ss'}$ : counts of $s \xrightarrow{a} s'$
- $\psi^a_{sz}$ : counts of seeing $z$ at $s$ after doing $a$.

$\Rightarrow$ Decision problem over $(s, \phi, \psi)$.

# Bayes-Adaptive POMDP

## Bayes-Adaptive POMDP Model ([Ross et al. NIPS'07])

- $S' = S \times \mathbb{N}^{|S|^2|A|} \times \mathbb{N}^{|S||A||Z|}$
- $A' = A$
- $Z' = Z$
- $Pr(s', \phi', \psi' | s, \phi, \psi, a, z) =$
$$\frac{\phi^a_{ss'}}{\sum_{s'' \in S} \phi^a_{ss''}} \frac{\psi^a_{s'z}}{\sum_{z' \in Z} \psi^a_{s'z}} I(\phi', \phi + \delta^a_{ss'}) I(\psi', \psi + \delta^a_{s'z})$$
- $R'(s, \phi, \psi, a) = R(s, a)$

Goal : Maximize return under partial observability of $(s, \phi, \psi)$.

# A few comments

## About the Bayes-Adaptive MDP

- Defines an infinite-state MDP with a known model.
- The state is defined over $(s, \phi)$.
- At every time step, $s$ is observable, and $\phi$ is updated.

## About the Bayes-Adaptive POMDP

- Defines an infinite-state POMDP with a known model.
- The state is defined over $(s, \phi, \psi)$.
- At every time step, $s$ is not observable, so neither are $\phi$ and $\psi$.

# Question

**How can we update counters $\phi$ and $\psi$, if we don't observe $s$?**

(Note : this is the basic problem for classical RL in partially observable environments.)

# Belief in BAPOMDPs

Let

- $b_0$ : initial belief over original state space
- $\phi_0, \psi_0$ : initial counts (prior on $T, O$)

Initial belief of the BAPOMDP :

$$b_0'(s, \phi, \psi) = b_0(s) I(\phi, \phi_0) I(\psi, \psi_0)$$

Monitoring the belief :

- The belief defines a mixture of Dirichlets over $T, O$.
- Allows us to learn the unknown POMDP model.
- Computing $b_t$ exactly is in $O(|S|^{t+1})$ - VERY LARGE !

# Theoretical results

We can bound the error introduced in the value function due to differences in model posteriors.

## Theorem 1 :

$$\sup_{\alpha \in \Gamma_t, s \in S} |V_t^{\alpha}(s, \phi, \psi) - V_t^{\alpha}(s, \phi', \psi')| \leq$$

$$\frac{2\gamma||R||_\infty}{(1-\gamma)^2} \sup_{s, s' \in S, a \in A} \Bigg[ D_S^{sa}(\phi, \phi') + D_Z^{s'a}(\psi, \psi') + \frac{4}{\ln(\gamma^{-e})} \left( \frac{\sum_{s'' \in S} |\phi^a_{ss''} - \phi'^a_{ss''}|}{(\mathcal{N}_\phi^{sa}+1)(\mathcal{N}_{\phi'}^{sa}+1)} + \frac{\sum_{z \in Z} |\psi^a_{s'z} - \psi'^a_{s'z}|}{(\mathcal{N}_\psi^{s'a}+1)(\mathcal{N}_{\psi'}^{s'a}+1)} \right) \Bigg]$$

where :

$$\mathcal{N}_\phi^{sa} = \textstyle\sum_{s' \in S} \phi^a_{ss'}, \qquad \mathcal{N}_\psi^{sa} = \textstyle\sum_{z \in Z} \psi^a_{sz},$$

$$D_S^{sa}(\phi, \phi') = \sum_{s' \in S} \left| \frac{\phi^a_{ss'}}{\mathcal{N}_\phi^{sa}} - \frac{\phi'^a_{ss'}}{\mathcal{N}_{\phi'}^{sa}} \right| \qquad D_Z^{sa}(\psi, \psi') = \sum_{z \in Z} \left| \frac{\psi^a_{sz}}{\mathcal{N}_\psi^{sa}} - \frac{\psi'^a_{sz}}{\mathcal{N}_{\psi'}^{sa}} \right|.$$

**Nice fancy math ! But how good is this value function really ?**

# Theoretical results

We can bound the error introduced in the value function due to differences in model posteriors.

**Theorem 1 :**

$$\sup_{\alpha \in \Gamma_t, s \in S} |V_t^\alpha(s, \phi, \psi) - V_t^\alpha(s, \phi', \psi')| \leq$$

$$\frac{2\gamma||R||_\infty}{(1-\gamma)^2} \sup_{s,s' \in S, a \in A} \Bigg[ D_S^{sa}(\phi, \phi') + D_Z^{s'a}(\psi, \psi') + \frac{4}{\ln(\gamma^{-e})} \left( \frac{\sum_{s'' \in S} \left|\phi_{ss''}^a - \phi_{ss''}^{\prime a}\right|}{(\mathcal{N}_\phi^{sa}+1)(\mathcal{N}_{\phi'}^{sa}+1)} + \frac{\sum_{z \in Z} \left|\psi_{s'z}^a - \psi_{s'z}^{\prime a}\right|}{(\mathcal{N}_\psi^{s'a}+1)(\mathcal{N}_{\psi'}^{s'a}+1)} \right) \Bigg]$$

where :

$\mathcal{N}_\phi^{sa} = \sum_{s' \in S} \phi_{ss'}^a$, $\mathcal{N}_\psi^{sa} = \sum_{z \in Z} \psi_{sz}^a$,

$D_S^{sa}(\phi, \phi') = \sum_{s' \in S} \left| \frac{\phi_{ss'}^a}{\mathcal{N}_\phi^{sa}} - \frac{\phi_{ss'}^{\prime a}}{\mathcal{N}_{\phi'}^{sa}} \right|$ $D_Z^{sa}(\psi, \psi') = \sum_{z \in Z} \left| \frac{\psi_{sz}^a}{\mathcal{N}_\psi^{sa}} - \frac{\psi_{sz}^{\prime a}}{\mathcal{N}_{\psi'}^{sa}} \right|$.

**Nice fancy math ! But how good is this value function really ?**

# Finite POMDP Approximation

We can bound the error introduced in the value function when we approximate the BAPOMDP by thresholding count vectors.

**Theorem 2 :**

To achieve $|\tilde{V}_t^{\alpha}(\mathcal{P}_\epsilon(s,\phi,\psi)) - V_t^{\alpha}(s,\phi,\psi)| < \frac{\epsilon}{1-\gamma}$,

where $\tilde{\alpha}_t$ is computed from $M_\epsilon$ and $\alpha_t$ is computed from $M$,

define $\epsilon' = \frac{\epsilon(1-\gamma)^2}{8\gamma||R||_\infty}$, $\epsilon'' = \frac{\epsilon(1-\gamma)^2 \ln(\gamma^{-e})}{32\gamma||R||_\infty}$,

$N_S^\epsilon = \max\left(\frac{|S|(1+\epsilon')}{\epsilon'}, \frac{1}{\epsilon''} - 1\right)$, $N_Z^\epsilon = \max\left(\frac{|Z|(1+\epsilon')}{\epsilon'}, \frac{1}{\epsilon''} - 1\right)$.

**Ok ! We know how many samples we need to get an $\epsilon$-optimal solution. But is this practical ?**

# Finite POMDP Approximation

We can bound the error introduced in the value function when we approximate the BAPOMDP by thresholding count vectors.

**Theorem 2 :**

To achieve $|\tilde{V}_t^{\alpha}(\mathcal{P}_\epsilon(s,\phi,\psi)) - V_t^{\alpha}(s,\phi,\psi)| < \frac{\epsilon}{1-\gamma}$,

where $\tilde{\alpha}_t$ is computed from $M_\epsilon$ and $\alpha_t$ is computed from $M$,

define $\epsilon' = \frac{\epsilon(1-\gamma)^2}{8\gamma||R||_\infty}$, $\quad \epsilon'' = \frac{\epsilon(1-\gamma)^2 \ln(\gamma^{-e})}{32\gamma||R||_\infty}$,

$N_S^\epsilon = \max\left(\frac{|S|(1+\epsilon')}{\epsilon'}, \frac{1}{\epsilon''} - 1\right)$, $\quad N_Z^\epsilon = \max\left(\frac{|Z|(1+\epsilon')}{\epsilon'}, \frac{1}{\epsilon''} - 1\right)$.

**Ok ! We know how many samples we need to get an $\epsilon$-optimal solution. But is this practical ?**

# Approximate Belief Monitoring

**Problem** : Computing $b_t$ exactly in a BAPOMDP is in $O(|S|^{t+1})$.

Use particle filters for efficient approximation of the belief :

- **Monte Carlo** : Perform belief update by sampling $K$ particles and state transitions.
- **K Most Likely** : After each belief update, keep only the $K$ particles with highest probability.
- **Weighted Distance Metric** : After each belief update, use a greedy algorithm to pick the $K$ particles which best fit the posterior (using the distance metric in Theorem 1).

# Approximate Belief Monitoring

**Problem** : Computing $b_t$ exactly in a BAPOMDP is in $O(|S|^{t+1})$.

Use particle filters for efficient approximation of the belief :

- **Monte Carlo** : Perform belief update by sampling $K$ particles and state transitions.
- **K Most Likely** : After each belief update, keep only the $K$ particles with highest probability.
- **Weighted Distance Metric** : After each belief update, use a greedy algorithm to pick the $K$ particles which best fit the posterior (using the distance metric in Theorem 1).

# Approximate Belief Monitoring

**Problem** : Computing $b_t$ exactly in a BAPOMDP is in $O(|S|^{t+1})$.

Use particle filters for efficient approximation of the belief :

- **Monte Carlo** : Perform belief update by sampling $K$ particles and state transitions.
- **K Most Likely** : After each belief update, keep only the $K$ particles with highest probability.
- **Weighted Distance Metric** : After each belief update, use a greedy algorithm to pick the $K$ particles which best fit the posterior (using the distance metric in Theorem 1).

# Approximate Belief Monitoring

**Problem** : Computing $b_t$ exactly in a BAPOMDP is in $O(|S|^{t+1})$.

Use particle filters for efficient approximation of the belief :

- **Monte Carlo** : Perform belief update by sampling $K$ particles and state transitions.
- **K Most Likely** : After each belief update, keep only the $K$ particles with highest probability.
- **Weighted Distance Metric** : After each belief update, use a greedy algorithm to pick the $K$ particles which best fit the posterior (using the distance metric in Theorem 1).

# Approximation Planning in BAPOMDPs

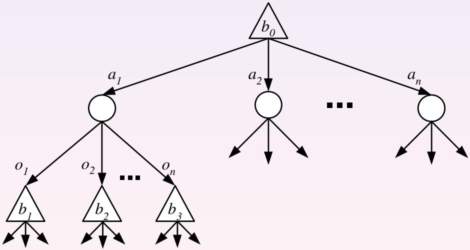
We still need to **optimize a policy** :

$$Pr(s, \phi, \psi | \phi_0, \psi_0, a_1, z_1, ..., a_{t-1}, z_{t-1}) \rightarrow a$$

This involves solving an infinite-state POMDP !

- It can be solved exactly for finite horizons given prior $(\phi_0, \psi_0)$.

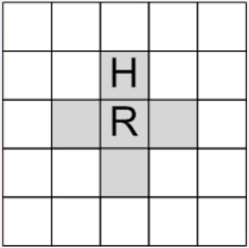
Monte Carlo Online Planning (Receding Horizon Control) :



# Experimental Results

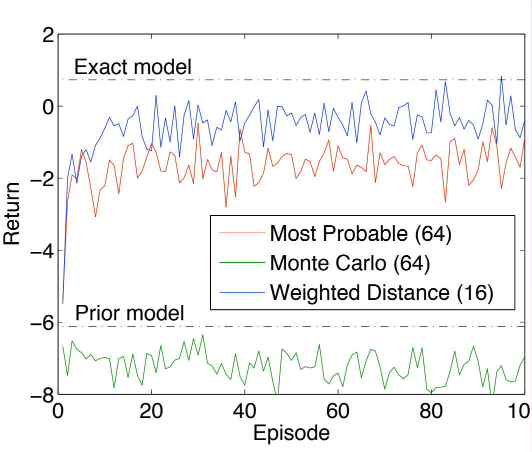
## Follow :

- A robot has to follow an individual within a known environment.
- There are 2 possible individuals with different motion behaviors. The behaviors are unknown a priori.
- The individual changes at the beginning of each trajectory, and can only be identified by observations of the behavior.



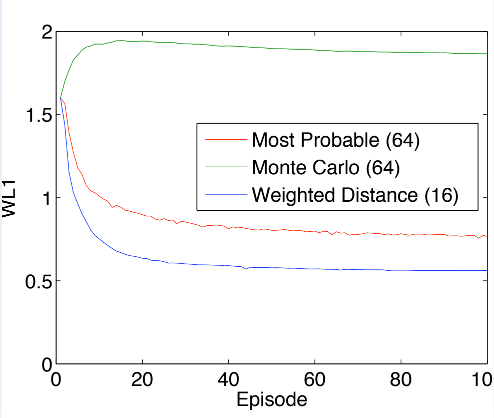
# Experimental Results

## Expected return :



# Experimental Results

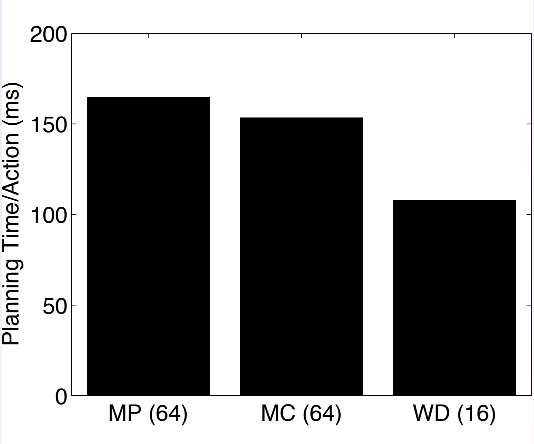
## Model Accuracy :




$$WL1(b) =$$
$$\sum_{(s,\phi,\psi)\in S_b'} b(s,\phi,\psi) \sum_{a\in A} \sum_{s'\in S} \left[ \sum_{s\in S} |T_\phi^{sas'} - T^{sas'}| + \sum_{z\in Z} |O_\psi^{s'az} - O^{s'az}| \right]$$

# Experimental Results

**Planning time :**



# Summary

We extended the model-based bayesian RL framework to handle partially observable domains.

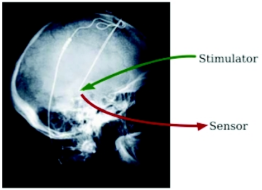
Optimal policy maximizes long-term return (given the prior), simultaneously :

- Exploring to learn the model.
- Identifying the system's state.
- Gathering rewards.

Monte Carlo methods can be used to achieve tractable (approximate) belief monitoring and planning.

# Recent work

**Problem :** Most real-world domains are represented using many state features. Will this scale to such large domains ? What if there are dependencies between state variables ?




Recent work has extended the bayesian RL framework to continuous domains (*Ross et al., ICRA'08*) and structured domains (*Ross et al., UAI'08*).

# Conclusion

Donald Rumsfeld once said :

*As we know*
*There are known knowns.*
*There are things we know we know.*

*We also know there are known unknowns.*
*That is to say*
*We know there are some things*
*We do not know.*

*But there are also unknown unknowns,*
*The ones we don't know*
*We don't know.*

**My talk today is really about turning those unknown unknowns into known unknows.**

# Acknowledgments

This work was done in collaboration with **Mahdi Milani Fard** and **Peng Sun** (first half), as well as **Stephane Ross** and **Brahim Chaib-draa** (second half).

Many thanks go to the STAR*D team for providing the data for the medical treatment design example (in particular Dr. John Rush, UTSouthwestern).

Funding provided by :

- National Institute of Mental Health (NIMH)
- Natural Sciences and Engineering Research Council of Canada (NSERC)
- Fonds Quebecois de Recherche sur la Nature et les Technologies (FQRNT)
- Canadian Foundation for Innovation (CFI).