Bayesian Reinforcement Learning in Continuous POMDPs

Stéphane Ross\textsuperscript{1}, Brahim Chaib-draa\textsuperscript{2} and Joelle Pineau\textsuperscript{1}

\textsuperscript{1}School of Computer Science, McGill University, Canada
\textsuperscript{2}Department of Computer Science, Laval University, Canada

December 19\textsuperscript{th}, 2007
Motivation

How should robots make decisions when:
- Environment is partially observable and continuous
- Poor model of sensors and actuators
- Parts of the model has to be learnt entirely during execution (e.g. users’ preferences/behavior)

such as to maximize expected long-term rewards?

Typical Examples:

Solution: Bayesian Reinforcement Learning!

[Rottmann]
Partially Observable Markov Decision Processes

POMDP : \((S, A, T, R, Z, O, \gamma, b_0)\)

- **S**: Set of states
- **A**: Set of actions
- **\(T(s, a, s') = \Pr(s'|s, a)\)**, the transition probabilities
- **\(R(s, a) \in \mathbb{R}\)**, the immediate rewards
- **\(Z\)**: Set of observations
- **\(O(s', a, z) = \Pr(z|s', a)\)**, the observation probabilities
- **\(\gamma\)**: discount factor
- **\(b_0\)**: 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:
\[
V^*(b) = \max_{a \in A} \left[ R(b, a) + \gamma \sum_{z \in Z} \Pr(z|b, a) V^*(\tau(b, a, z)) \right]
\]
Bayesian Reinforcement Learning

General Idea:
- Define prior distributions over all unknown parameters.
- Maintain posteriors via Baye’s rule as experience is acquired.
- Plan considering posterior distribution over model.

Allows us to:
- Learn the system at same time we achieve the task efficiently.
- Tradeoff optimally exploration and exploitation.
- Consider model uncertainty during planning.
- Include prior knowledge explicitly.
Bayesian RL in Finite MDPs

In Finite MDPs ($T$ unknown) :

([Dearden 99], [Duff 02], [Poupart 06])

To learn $T$ : Maintain counts $\phi_{ss'}^a$ of number of times $s \xrightarrow{a} s'$ observed, starting from prior $\phi_0$.

Counts define Dirichlet prior/posterior over $T$.

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

- $S'$ : physical state ($s \in S$) + information state ($\phi$)

$$
T'(s, \phi, a, s', \phi') = \Pr(s', \phi'|s, \phi, a) \\
= \Pr(s'|s, \phi, a) \Pr(\phi'|\phi, s, a, s') \\
= \sum_{s'' \in S} \frac{\phi_{ss'}^a}{\phi_{ss''}^a} I(\phi', \phi + \delta_{ss'}^a)
$$

$$
V^*(s, \phi) = \max_{a \in A} \left[ R(s, a) + \gamma \sum_{s' \in S} \frac{\phi_{ss'}^a}{\phi_{ss''}^a} V^*(s', \phi + \delta_{ss'}^a) \right]
$$
Bayesian RL in Finite POMDPs

In Finite POMDPs ($T, O$ unknown): ([Ross 07])

Let:
- $\phi_{ss'}^a$: number of times $s \xrightarrow{a} s'$ observed.
- $\psi_{sz}^a$: number of times $z$ observed in $s$ after doing $a$.

Given action-observation sequence, use Bayes rule to maintain belief over $(s, \phi, \psi)$.

$\Rightarrow$ Decision under partial observability of $(s, \phi, \psi)$ is a POMDP itself:
- $s'$: physical state ($s \in S$) + information state ($\phi, \psi$)

$$P'(s, \phi, \psi, a, s', \phi', \psi', z) = \Pr(s', \phi', \psi', z|s, \phi, \psi, a)$$

$$= \Pr(s'|s, \phi, a) \Pr(z|\psi, s', a) \Pr(\phi'|\phi, s, a, s') \Pr(\psi'|\psi, a, s', z)$$

$$= \frac{\phi_{ss'}^a}{\sum_{s'' \in S} \phi_{ss''}^a} \frac{\psi_{sz}^a}{\sum_{z' \in Z} \psi_{s'z'}^a} I(\phi', \phi + \delta_{ss'}^a) I(\psi', \psi + \delta_{sz}^a)$$
Tiger domain with unknown sensor accuracy:

- Suppose prior $\psi_0 = (5, 3), \ b_0 = (0.5, 0.5)$
- Sequence of action-observation is:
  $\{ \text{Listen}, \text{l} \}, \ \{ \text{Listen}, \text{l} \}, \ \{ \text{Listen}, \text{l} \}, \ \{ \text{Right}, \text{-} \}$

\[
\begin{align*}
  b_0 &: \Pr(\text{L}, < 5, 3 >) = \frac{1}{2} \\
  \ &\quad \Pr(\text{R}, < 5, 3 >) = \frac{1}{2} \\
  b_1 &: \Pr(\text{L}, < 6, 3 >) = \frac{5}{8} \\
  \ &\quad \Pr(\text{R}, < 5, 4 >) = \frac{3}{8} \\
  b_3 &: \Pr(\text{L}, < 8, 3 >) = \frac{7}{9} \\
  \ &\quad \Pr(\text{R}, < 5, 6 >) = \frac{2}{9} \\
  b_4 &: \Pr(\text{L}, < 8, 3 >) = \frac{7}{18} \\
  \ &\quad \Pr(\text{L}, < 5, 6 >) = \frac{2}{18} \\
  \ &\quad \Pr(\text{R}, < 8, 3 >) = \frac{7}{18} \\
  \ &\quad \Pr(\text{R}, < 5, 6 >) = \frac{2}{18}
\end{align*}
\]
In robotics, continuous domains are common (continuous state, continuous action, continuous observations).

Could discretize the problem and apply our current method, but:
- Combinatorial explosion or poor precision
- Can require lots of training data (visit every small cell)

Can we extend Bayesian RL to continuous domains?
Bayesian RL in Continuous Domains?

Can’t use counts (Dirichlet distribution) to learn about the model.

We assume a parametric form for transition and observation model.

For instance, in the Gaussian case:

- \( S \subset \mathbb{R}^m, A \subset \mathbb{R}^n, Z \subset \mathbb{R}^p \)
- \( s_{t+1} = g_T(s_t, a_t, X_t) \)
- \( z_{t+1} = g_O(s_{t+1}, a_t, Y_t) \)

where \( X_t \sim N(\mu_X, \Sigma_X) \), \( Y_t \sim N(\mu_Y, \Sigma_Y) \), and \( g_T, g_O \) are arbitrary functions (possibly non-linear).

We assume \( g_T, g_O \) are known, but that the parameters \( \mu_X, \Sigma_X, \mu_Y, \Sigma_Y \) are unknown.

Relevant statistics depends on the parametric form.
\( \mu, \Sigma \) can be learned by maintaining sample mean \( \hat{\mu} \) and sample covariance \( \hat{\Sigma} \).

These define a Normal-Wishart posterior over \( \mu, \Sigma \):

- \( \mu | \Sigma = R \sim N(\hat{\mu}, \frac{R}{\nu}) \)
- \( \Sigma^{-1} \sim \text{Wishart}(\alpha, \tau^{-1}) \)

where:

- \( \nu \) : number of observations for \( \hat{\mu} \)
- \( \alpha \) : degree of freedom of \( \hat{\Sigma} \)
- \( \tau = \alpha \hat{\Sigma} \)

These can be updated easily after observation \( X = x \):

- \( \hat{\mu}' = \frac{\nu \hat{\mu} + x}{\nu + 1} \)
- \( \nu' = \nu + 1 \)
- \( \alpha' = \alpha + 1 \)
- \( \tau' = \tau + \frac{\nu}{\nu + 1} (\hat{\mu} - x)(\hat{\mu} - x)^T \)
Let’s define

- $\phi = (\hat{\mu}_X, \nu_X, \alpha_X, \tau_X)$: the posterior over $(\mu_X, \Sigma_X)$
- $\psi = (\hat{\mu}_Y, \nu_Y, \alpha_Y, \tau_Y)$: the posterior over $(\mu_Y, \Sigma_Y)$
- $U$: the update function of $\phi, \psi$, i.e. $U(\phi, x) = \phi'$ and $U(\psi, y) = \psi'$

Bayes-Adaptive Continuous POMDP : $(S', A', Z', P', R')$

- $S' = S \times \mathbb{R}^{|X|+|X|^2+2} \times \mathbb{R}^{|Y|+|Y|^2+2}$
- $A' = A$
- $Z' = Z$
- $P'(s, \phi, \psi, a, s', \phi', \psi', z) =$
  
  $$I(g_T(s, a, x), s')I(g_O(s', a, y), z)I(\phi', U(\phi, x))I(\psi', U(\psi, y))f_{X|\phi}(x)f_{Y|\psi}(y)$$
- $R'(s, \phi, \psi, a) = R(s, a)$

where $x = (\nu_X + 1)\hat{\mu}'_X - \nu_X \hat{\mu}_X$ and $y = (\nu_Y + 1)\hat{\mu}'_Y - \nu_Y \hat{\mu}_Y$. 
Monte Carlo Belief monitoring: (1 extra assumption: $g_O(s, a, \cdot)$ is 1-1 transformation of $Y$)

1. Sample $(s, \phi, \psi) \sim b_t$
2. Sample $(\mu_X, \Sigma_X) \sim NW(\phi)$
3. Sample $X \sim N(\mu_X, \Sigma_X)$
4. Compute $s' = g_T(s, a_t, X)$
5. Find unique $Y$ s.t. $z_{t+1} = g_O(s', a_t, Y)$
6. Compute $\phi' = U(\phi, X)$, $\psi' = U(\psi, Y)$
7. Sample $(\mu_Y, \Sigma_Y) \sim NW(\psi)$
8. Add $f(Y|\mu_Y, \Sigma_Y)$ to particle $b_{t+1}(s', \phi', \psi')$
9. Repeat until $K$ particles in $b_{t+1}$
Monte Carlo Online Planning (Receding Horizon Control) :
Simple Robot Navigation Task:

- $S : (x, y)$ position
- $A : (v, \theta)$ (velocity $v \in [0, 1]$ and angle $\theta \in [0, 2\pi]$)
- $Z :$ Noisy $(x, y)$ position
- $g_T(s, a, X) = s + v \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} X$
- $g_O(s', a, Y) = s' + Y$
- $R(s, a) = I(||s - s_{GOAL}||_2 < 0.25)$
- $\gamma = 0.85$
We choose exact parameters:

\[
\mu_X = \begin{bmatrix} 0.8 \\ 0.3 \end{bmatrix}, \quad \Sigma_X = \begin{bmatrix} 0.04 & -0.01 \\ -0.01 & 0.01 \end{bmatrix}, \\
\mu_Y = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \Sigma_Y = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}
\]

Starting with prior based on 10 "artificial" samples:

\[
\hat{\mu}_X = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \hat{\Sigma}_X = \begin{bmatrix} 0.04 & -0.01 \\ -0.01 & 0.16 \end{bmatrix}, \\
\hat{\mu}_Y = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \hat{\Sigma}_Y = \begin{bmatrix} 0.16 & 0 \\ 0 & 0.16 \end{bmatrix}
\]

Such that \( \phi_0 = (\hat{\mu}_X, 10, 9, 9\hat{\Sigma}_X), \psi_0 = (\hat{\mu}_Y, 10, 9, 9\hat{\Sigma}_Y) \). Each time robot reaches the goal, a new goal is chosen randomly at 5 distance unit from previous goal.
Robot Navigation Task

Average evolution of the return over time:

![Graph showing the average return over training steps for different models: Prior model, Exact Model, and Learning. The graph plots the average return against training steps.]
Robot Navigation Task

Average accuracy of the model over time:

Model Accuracy is measured as follows:

$$WL1(b) = \sum_{(s, \phi, \psi)} b(s, \phi, \psi) \left[ ||\mu_\phi - \mu_X||_1 + ||\Sigma_\phi - \Sigma_X||_1 + ||\mu_\psi - \mu_Y||_1 + ||\Sigma_\psi - \Sigma_Y||_1 \right]$$
We presented a new framework for planning in partially observable and continuous domains with uncertain model parameters.

Optimal policy maximizes long-term return given the prior over model parameters.

Monte Carlo methods can be used for more tractable approximate belief monitoring and planning.

Interesting future applications human-computer interactions and robotics.