Model-Based Bayesian RL

slides adapted from:
Poupart ICML 2007
History

• Reinforcement Learning in AI:
  – Formalized in the 1980’s by Sutton, Barto and others
  – Traditional RL algorithms are not Bayesian
• RL is the problem of controlling a Markov Chain with unknown probabilities.
History

- Reinforcement Learning in AI:
  - Formalized in the 1980’s by Sutton, Barto and others
  - Traditional RL algorithms are not Bayesian
- RL is the problem of controlling a Markov Chain with unknown probabilities.

- Operations Research: Bayesian Reinforcement Learning already studied under the names of
  - Adaptive control processes [Bellman]
  - Dual control [Fel’Dbbaum]
  - Optimal learning
- 1950’s & 1960’s: Bellman, Fel’Dbbaum, Howard and others develop Bayesian techniques to control Markov chains with uncertain probabilities and rewards
History

• Operations Research
  – Theoretical foundation
  – Algorithmic solutions for special cases
    • Bandit problems: Gittins indices
  – Intractable algorithms for the general case

• Machine Learning / Artificial Intelligence
  – Algorithmic advances to improve scalability
Notation

• Markov Decision Process:
  – \( X \): set of states \(<x_s, x_r>\)
    • \( x_s \): physical state component
    • \( x_r \): reward component
  – \( A \): set of actions
  – \( p(x' | x, a) \): transition and reward probabilities – don’t know
Notation

• Markov Decision Process:
  – $X$: set of states $<x_s, x_r>$
    • $x_s$: physical state component
    • $x_r$: reward component
  – $A$: set of actions
  – $p(x' | x, a)$: transition and reward probabilities – don’t know

• Bayesian Model-based Reinforcement Learning
• Encode unknown prob. with random variables $\theta$
  – i.e., $\theta_{xax'} = \Pr(x' | x, a)$: random variable in [0,1]
  – i.e., $\theta_{xa} = \Pr(\bullet | x, a)$: multinomial distribution
Bayesian Inference

• Assume a prior \( b(\theta_{xa}) = \Pr(\theta_{xa}) \)

• Learning: use Bayes theorem to compute posterior \( b_{xax'}(\theta_{xa}) = \Pr(\theta_{xa} | x,a,x') \)

\[
b_{xax'}(\theta_{xa}) = k \Pr(\theta_{xa}) \Pr(x'|x,a,\theta_{xa}) = k b(\theta_{xa}) \theta_{xax'} \]
Bayesian Inference

• Assume a prior $b(\theta_{xa}) = \Pr(\theta_{xa})$
• Learning: use Bayes theorem to compute posterior $b_{xa}(\theta_{xa}) = \Pr(\theta_{xa} | x,a,x')$

$$b_{xa}(\theta_{xa}) = k \Pr(\theta_{xa}) \Pr(x'|x,a,\theta_{xa})$$

$$= k b(\theta_{xa}) \theta_{xa}$$

– What is the prior $b$?

• Could we choose $b$ to be in the same class as $b_{xa'}$?
**Conjugate Prior**

- Suppose $b$ is a monomial in $\theta$
  - i.e. $b(\theta_{xa}) = k \prod_{x''} (\theta_{xax''})^{n_{xax''}} - 1$

- Then $b_{xax'}$ is also a monomial in $\theta$

\[
b_{xax'}(\theta_{xa}) = k \left[ \prod_{x''} (\theta_{xax''})^{n_{xax''}} - 1 \right] \theta_{xax'} \\
= k \prod_{x''} (\theta_{xax''})^{n_{xax''}} - 1 + \delta(x',x'')
\]

- This prior is the Dirichlet distribution and is the **conjugate prior** to multinomials
Structured Priors

• Suppose probability of two transitions is the same
  – Tie identical parameters
  – If $\Pr(\cdot|x,a) = \Pr(\cdot|x',a')$ then $\theta_{xa} = \theta_{x'a'}$
  – Fewer parameters and pool evidence
Structured Priors

• Suppose probability of two transitions is the same
  – Tie identical parameters
  – If \( \Pr(\bullet|x,a) = \Pr(\bullet|x',a') \) then \( \theta_{xa} = \theta_{x'a'} \)
  – Fewer parameters and pool evidence

• Suppose transition dynamics are factored
  – E.g., transition probabilities can be encoded with a dynamic Bayesian network
  – Exponentially fewer parameters
  – E.g. \( \theta_{x,pa(X)} = \Pr(X=x|pa(X)) \)
POMDP formulation

- Traditional RL:
  - $\mathcal{X}$: set of states
  - $\mathcal{A}$: set of actions
  - $p(x' | x, a)$: transition probabilities (UNKNOWN)

- Bayesian RL
  - $\mathcal{X} \times \theta$: set of states $\langle x, \theta \rangle$
    - $x$: physical state (observable)
    - $\theta$: model (hidden)
  - $\mathcal{A}$: set of actions
  - $\Pr(x', \theta' | x, \theta, a)$: transition probabilities (KNOWN)
**POMDP formulation**

- **Traditional RL:**
  - \( X \): set of states
  - \( A \): set of actions
  - \( p(x'|x,a) \): transition probabilities \((UNKNOWN)\)

- **Bayesian RL -> POMDP (state unknown)**
  - \( X \times \theta \): set of states \(<x,\theta>\)
    - \( x \): physical state (observable)
    - \( \theta \): model (hidden)
  - \( A \): set of actions
  - \( Pr(x',\theta'|x,\theta,a) \): transition probabilities \((KNOWN)\)
POMDP formulation

- $\Pr(x'|x,a) = ?$
POMDP formulation

- $\Pr(x'|x,a) = ?$

- $\Pr(x',\theta'|x,\theta,a) = \Pr(x'|x,\theta,a) \Pr(\theta'|\theta)$

- $\Pr(x'|x,\theta,a) = \theta_{x,a,x'}$

- $\Pr(\theta'|\theta) = \begin{cases} 1 & \text{if } \theta' = \theta \\ 0 & \text{otherwise} \end{cases}$
Belief MDP

• Bayesian RL -> POMDP
  – $X \times \Theta$: set of states $<x,\theta>$
  – $A$: set of actions
  – $\Pr(x',\theta' | x,\theta,a)$: transition probabilities (KNOWN)

• Bayesian RL -> Belief MDP
  – $X \times B$: set of states $<x,b(\theta)>$
  – $b$ is a distribution (belief) over the unknown $\theta$
  – $A$: set of actions
  – $p(x',b' | x,b,a)$: transition probabilities (KNOWN)
Transition Probabilities

- \( \Pr(x', \theta'|x, \theta, a) = \Pr(x'|x, \theta, a) \Pr(\theta'|\theta) \)

- \( \Pr(x'|x, \theta, a) = \theta_{x,a,x'} \)

- \( \Pr(\theta'|\theta) = \begin{cases} 1 & \text{if } \theta' = \theta \\ 0 & \text{otherwise} \end{cases} \)

- \( \Pr(x', b'|x, b, a) = \Pr(x'|x, b, a) \Pr(b'|x, b, a, x') \)

- \( \Pr(x'|x, b, a) = \int_{\theta} b(\theta) \Pr(x'|x, \theta, a) \, d\theta \)

- \( \Pr(b'|x, b, a, x') = \begin{cases} 1 & \text{if } b' = b_{xax'} \\ 0 & \text{otherwise} \end{cases} \)
Policy Optimisation

- Classic RL:
  - \( V^*(x) = \max_a \sum_{x'} \Pr(x' | x, a) [r_{x'} + \gamma V^*(x')] \)
  - Hard to tell what needs to be explored
  - Exploration heuristics: \( \varepsilon \)-greedy, Boltzmann, etc.

- Bayesian RL:
  - \( V^*(x, b) = \max_a \sum_{x'} \Pr(x' | x, b, a) [r_{x'} + \gamma V^*(x', b)] \)
  - Belief \( b \) tells us what parts of the model are not well known and therefore worth exploring
  - Exploration/exploitation? Naturally taken care of
  - Single objective: max expected total rewards
Policy Optimisation

• Classic RL:
  – \( V^*(x) = \max_a \sum_x \Pr(x' | x, a) [x_r' + \gamma V^*(x')] \)
  – Hard to tell what needs to be explored
  – Exploration heuristics: \( \epsilon \)-greedy, Boltzmann, etc.

• Bayesian RL:
  – \( V^*(x,b) = \max_a \sum_x \Pr(x' | x, b, a) [x_r' + \gamma V^*(x',b_{xax'})] \)
  – Belief \( b \) tells us what parts of the model are not well known and therefore worth exploring
Policy Optimisation

• Classic RL:
  – \( V^*(x) = \max_a \sum_x \Pr(x' | x, a) [r_{x'} + \gamma V^*(x')] \)
  – Hard to tell what needs to be explored
  – Exploration heuristics: \( \varepsilon \)-greedy, Boltzmann, etc.

• Bayesian RL:
  – \( V^*(x, b) = \max_a \sum_x \Pr(x' | x, b, a) [r_{x'} + \gamma V^*(x', b_{xax'})] \)
  – Belief \( b \) tells us what parts of the model are not well known and therefore worth exploring

• Exploration/exploitation? Naturally taken care of
  – single objective: max expected total rewards
    \[ V^\mu(x_0) = \sum_t \gamma^t \mathbb{E}[ r_{x,t} \mid x_t, \mu] \]

Optimal policy \( \mu^*: V^{\mu^*}(x) \geq V^\mu(x) \) for all \( x, \mu \)
Policy Optimisation

• Bayesian RL, need to solve:
  \[ V^*(x,b) = \max_a \sum_x \Pr(x'|x,b,a) [x_r' + \gamma V^*(x',b_{xax'})] \]

• Some approximations:
  – Myopic
  – Sampling
  – Discretization
Myopic Value of Information

• Dearden, Friedman, Andre (1999)

• Myopic value of information:
  – Expected gain from the observation of a transition

• Myopic value of perfect information MVPI(x,a):
  – Upper bound on myopic value of information
  – Expected gain from learning the true value of a in x

• Action selection
  – \( a^* = \arg\max_a Q(x,a) + \text{MVPI}(x,a) \)
Thompson Sampling

- Strens (2000)

- Thomson Sampling
  - Sample $\theta$ from $b(\theta)$ (EXPLORE)
  - Select best action for $\theta$ Policy Gradient (EXPLOIT)

- Provides an exploration heuristic
# Empirical Comparison

From Strens (2000)

<table>
<thead>
<tr>
<th>Method</th>
<th>CHAIN</th>
<th>LOOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Phase 1</td>
<td>Phase 2</td>
</tr>
<tr>
<td>QL semi-uniform</td>
<td>1594 ± 2</td>
<td>1597 ± 2</td>
</tr>
<tr>
<td>QL Boltzmann</td>
<td>1606 ± 26</td>
<td>1623 ± 22</td>
</tr>
<tr>
<td>IEQL+</td>
<td>2344 ± 78</td>
<td>2557 ± 90</td>
</tr>
<tr>
<td>Bayes VPI+MIX</td>
<td>1697 ± 112</td>
<td>2417 ± 217</td>
</tr>
<tr>
<td>Heuristic DP</td>
<td>2855 ± 29</td>
<td>3450 ± 21</td>
</tr>
<tr>
<td>Bayesian DP</td>
<td>3158 ± 31</td>
<td>3611 ± 27</td>
</tr>
</tbody>
</table>
POMDP Discretization

- Jaulmes, Pineau and Precup (2005)

- Idea: discretize $\theta$ with a grid.
- Use your favourite POMDP algorithm
- Problem: state space grows exponentially with the number of $\theta_{xax'}$ parameters
Policy Optimisation

• Bayesian RL:

\[ V^*(x, b) = \max_a \Sigma_{x'} \Pr(x'|x, b, a) [x'_r + \gamma V^*(x', b_{x'x})] \]

• Difficulty:
  – \( b \) (and \( \theta \)) are continuous
  – What is the form/parameterization of \( V^* \)?
Policy Optimisation

- Bayesian RL:
  \[ V^*(x,b) = \max_a \sum_{x'} \Pr(x'|x,b,a) [x_r' + \gamma V^*(x',b_{ax'})] \]

- Difficulty:
  - \( b \) (and \( \theta \)) are continuous
  - What is the form/parameterization of \( V^* \)?

- Poupart et al. (2006)
  - Optimal value function: \( V_x^*(\theta) = \max_i \text{poly}_i(\theta) \)
  - BEETLE algorithm (Bayesian Exploration Exploitation Tradeoff in LEarning)
Value Function Parameterisation

- **Theorem:** $V^*$ is the upper envelope of a set of multivariate polynomials ($V_x(\theta) = \max_i \text{poly}_i(\theta)$)
Value Function Parameterisation

• **Theorem:** $V^*$ is the upper envelope of a set of multivariate polynomials ($V_x(\theta) = \max_i \operatorname{poly}_i(\theta)$)

• **Proof:** by induction
  - Define value function in terms of $\theta$ instead of $b$
    • i.e. $V^*(x,b) = \int_\theta b(\theta) \ V_x(\theta) \ d\theta$
  - Bellman’s equation

  \[
  V_x(\theta) = \max_a \sum_{x'} \Pr(x'|x,a,\theta) \left[ x_{r'} + \gamma \ V_{x'}(\theta) \right]
  = \max_a \sum_{x'} \theta_{xax'} \left[ k + \gamma \ \max_i \operatorname{poly}_i(\theta) \right]
  = \max_j \ \operatorname{poly}_j(\theta)
  \]
Partially Observable Domain

• State \( (x) \) unobserved
• Beliefs: mixtures of Dirichlets

• Theorem also holds for partially observable domains:
  \[ V_x(\theta) = \max_i \text{polynomials}_i(\theta) \]
BEETLE Algorithm

- Sample a set of reachable belief points $B$
- $V \leftarrow \{0\}$
- Repeat
  - $V' \leftarrow \{\}$
  - For each $b \in B$ compute multivariate polynomial
    - $poly_{ax}(\theta) \leftarrow \arg\max_{poly \in V} \int_\theta b_{xax}(\theta) poly(\theta) \, d\theta$
BEETLE Algorithm

- Sample a set of reachable belief points \( B \)
- \( V \leftarrow \{0\} \)
- Repeat
  - \( V' \leftarrow \{\} \)
  - For each \( b \in B \) compute multivariate polynomial
    - \( \text{poly}_{ax}(\theta) \leftarrow \text{argmax}_{\text{poly} \in V} \int_\theta \text{poly}_x(\theta) \text{poly}(\theta) \, d\theta \)
    - \( a^* \leftarrow \text{argmax}_a \int_\theta b_{sas}(\theta) \sum_{x'} \theta_{xax'} [x' + \gamma \text{poly}_{ax}(\theta)] \, d\theta \)
BEETLE Algorithm

- Sample a set of reachable belief points $B$
- $V \leftarrow \{0\}$
- Repeat
  - $V' \leftarrow \{}$
  - For each $b \in B$ compute multivariate polynomial
    - $\text{poly}_{ax}(\theta) \leftarrow \text{argmax}_{poly \in V} \int_{\theta} b_{xax}(\theta) \text{poly}(\theta) \ d\theta$
    - $a^* \leftarrow \text{argmax}_a \int_{\theta} b_{sas}(\theta) \sum_{x'} \theta_{xax'} [x'_r + \gamma \text{poly}_{ax}(\theta)] \ d\theta$
    - $\text{poly}(\theta) \leftarrow \sum_{x'} \theta_{xa^*x'} [x'_r + \gamma \text{poly}_{a^*x'}(\theta)]$
    - $V' \leftarrow V' \cup \{\text{poly}\}$
  - $V \leftarrow V'$
Computational Complexity

• **Computational issue:**
  - # of monomials in each polynomial grows by $O(|X|)$ at each iteration

$$
poly(\theta) = \sum_x \theta_{xa^*x'} [x'_r + \gamma poly_{a^*x'}(\theta)]
= \sum_x \theta_{xax'} [x'_r + \gamma \sum_i mono_i(\theta)]
= x'_r + \gamma \sum_{i,x'} mono_{i,x}(\theta)
$$

• **After $n$ iterations:** polynomials have $O(|X|^n)$ monomials!
Projection Scheme

- Approximate polynomials by a linear combination of a fixed set of monomial basis functions $\phi_i(\theta)$:
  - i.e. $poly(\theta) \approx \sum_i c_i \phi_i(\theta)$
Projection Scheme

- Approximate polynomials by a linear combination of a fixed set of monomial basis functions $\phi_i(\theta)$:
  - i.e. $\text{poly}(\theta) \approx \sum_i c_i \phi_i(\theta)$

- Find best coefficients $c_i$ by minimizing $L_n$ norm:
  - $\min_c \int_\theta |\text{poly}(\theta) - \sum_i c_i \phi_i(\theta)|^n \, d\theta$

- For the Euclidean norm ($L_2$), this can be done by solving a system of linear equations $Ax = b$ such that
  - $A_{ij} = \int_\theta \phi_i(\theta) \phi_j(\theta) \, d\theta$
  - $b_i = \int_\theta \text{poly}(\theta) \phi_j(\theta) \, d\theta$
  - $x_i = c_i$
Basis Functions

• Which monomials should we use as basis functions?

• Recall that:
  - \( b_{xax'}(\theta) = k \ b(\theta) \ \theta_{xax'} \)
  - \( poly(\theta) \leftarrow \sum_{x'} \theta_{xax'} [x'_r + \gamma \ poly_{ax'}(\theta)] \)

• Hence we use beliefs as basis functions
BEETLE Properties

• Offline: optimize policy at sampled belief points
  – Time: minutes to hours
• Online: learn transition model (belief monitoring)
  – Time: fraction of a second

• Advantages:
  – Fast enough for online learning
  – Optimizes exploration/exploitation tradeoff
  – Easy to encode prior knowledge in initial belief

• Disadvantage:
  – Policy may not be good for all belief points
Experiments

• Comparison with two heuristics:
  • **Exploit**: pure exploitation strategy
    – Greedily select best action of the mean model at each time step
    – Slow execution: must solve an MDP at each time step
  • **Discrete POMDP**: discretize $\theta$
    – Discretization leads to an exponential number of states
    – Intractable for medium to large problems
# Experiments

| Problem  | |S| | |A| | Free params | Opt | Discrete POMDP | Exploit | Beetle | Beetle time (minutes) |
|----------|---|---|---|---|---|---|---|---|---|---|---|
| Chain1   | 5 | 2 | 1 | 3677 | 3661 ± 27 | 3642 ± 43 | 3650 ± 41 | 1.9 |
| Chain2   | 5 | 2 | 2 | 3677 | 3651 ± 32 | 3257 ± 124 | 3648 ± 41 | 2.6 |
| Chain3   | 5 | 2 | 40 | 3677 | na-m | 3078 ± 49 | 1754 ± 42 | 32.8 |
| Handw1   | 9 | 2 | 4 | 1153 | 1149 ± 12 | 1133 ± 12 | 1146 ± 12 | 14.0 |
| Handw2   | 9 | 2 | 8 | 1153 | 990 ± 8 | 991 ± 31 | 1082 ± 17 | 55.7 |
| Handw3   | 9 | 6 | 270 | 1083 | na-m | 297 ± 10 | 385 ± 10 | 133.6 |
Incorporating Prior Knowledge

<table>
<thead>
<tr>
<th>Problem</th>
<th>Opt</th>
<th>Informative priors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>k = 0</td>
</tr>
<tr>
<td>Chain3</td>
<td>3677</td>
<td>1754 ± 42</td>
</tr>
<tr>
<td>Handw2</td>
<td>1153</td>
<td>1082 ± 17</td>
</tr>
<tr>
<td>Handw3</td>
<td>1083</td>
<td>385 ± 10</td>
</tr>
</tbody>
</table>
Active Learning

• Active learning: learner chooses training data

• In RL:
  – learner chooses actions, which influence future states
  – How can we choose actions that reveal the most information at the least cost?
  – Same problem as the exploration/exploitation tradeoff
  – Bayesian RL provides a solution (in principle)
Other RL Variants for which Bayesian approaches have been used

- Inverse RL
  - know transition probabilities, and optimal policy, want to infer the reward function
- Imitation Learning
  - two agents: learner and mentor
  - learner observers mentor states, but not policy
- Multiagent Coordination
  - stochastic game
  - many equilibriums, necessary to converge to the same one, preferably the best one; induces exploration/exploitation trade off
- Partially Observable Stochastic Games
  - Multiagent system, with partial observable interactive states
  - Difficulty: nested beliefs