An Analytic Solution to Discrete Bayesian Reinforcement Learning

Pascal Poupart (U of Waterloo)
Nikos Vlassis (U of Amsterdam)
Jesse Hoey (U of Toronto)
Kevin Regan (U of Waterloo)
Motivation

- Automated assistant
  - [Boger et al. IJCAI-05]

- Use RL to adapt to users
  - Learn through user interactions (no simulation)
  - Bear cost of actions
  - Cannot explore too much
  - Real-time response
Model-Based Bayesian RL

- Model-based Bayesian RL:
  - Naturally optimize exploration/exploitation tradeoff
  - Reduce exploration with prior knowledge
  - Mathematically and computationally complex

- Contributions:
  - Optimal value function has simple parameterization
    - i.e., upper envelope of a set of multivariate polynomials
  - **BEETLE: Bayesian Exploration/Exploitation Tradeoff in LEarning**
    - Exploit polynomial parameterization
Outline

- Bayesian reinforcement learning
- Value function parameterization
- BEETLE algorithm
- Experiments
- Conclusion
Reinforcement Learning

- Markov Decision Process:
  - $S$: set of states
  - $A$: set of actions
  - $R$: set of rewards
  - $T(s,a,s') = \Pr(s'|s,a)$: transition function
  - $U(s,a) = r$: reward function

- Bayesian Model-based Reinforcement Learning
- Encode unknown prob. by random variables $\theta$
  - i.e., $\theta_{sas'} = \Pr(s'|s,a)$: random variable in $[0,1]$
  - i.e., $\theta_{sa} = \Pr(\cdot|s,a)$: multinomial distribution
Model Learning

- Assume prior $b(\theta_{sa}) = \Pr(\theta_{sa})$
- Learning: compute posterior given $s, a, s'$
  - $b_{sas'}(\theta_{sa}) = k \Pr(\theta_{sa}) \Pr(s'|s, a, \theta_{sa}) = k b(\theta_{sa}) \theta_{sas'}$

- Conjugate prior:
  - Dirichlet prior $\rightarrow$ Dirichlet posterior
- $b(\theta_{sa}) = \text{Dir}(\theta_{sa}; n_{sa}) = k \prod_{s''} (\theta_{sas''})^{n_{sas''} - 1}$
- $b_{sas'}(\theta_{sa}) = k b(\theta_{sa}) \theta_{sas'}$
  - $= k \prod_{s''} (\theta_{sas''})^{n_{sas''} - 1 + \delta(s', s'')}$
  - $= k \text{Dir}(\theta_{sa}; n_{sa} + \delta(s', s''))$
Prior Knowledge

• Structural priors
  – Tie identical parameters
    • If \( \Pr(\cdot|s,a) = \Pr(\cdot|s',a') \) then \( \theta_{sa} = \theta_{s'a'} \)
  – Factored representation
    • DBN: unknown conditional dist.

• Informative priors
  – No knowledge: uniform Dirichlet
  – If \( (\theta_1, \theta_2) \sim (0.2, 0.8) \)
    then set \( (n_1, n_2) \) to \( (0.2k, 0.8k) \)
    • \( k \) indicates the level of confidence
Policy Optimization

• Classic RL:
  – $V^*(s) = \max_a U(s,a) + \sum_{s'} \Pr(s'|s,a) V^*(s')$
  – Hard to tell what needs to be explored
  – Exploration heuristics: $\varepsilon$-greedy, Boltzmann, etc.

• Bayesian RL:
  – $V^*(s,b) = \max_a U(s,a) + \sum_{s'} \Pr(s'|s,b,a) V^*(s',b_{sas'})$
  – Belief $b$ tells us what parts of the model are not well known and therefore worth exploring
  – Optimal exploration/exploitation tradeoff
  – [Dearden 98,99], [Strens 00], [Duff 02], [Wang 05]
Value Function Parameterization

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

- **Proof:** by induction
  - Define value function in terms of $\theta$ instead of $b$
    - i.e. $V^*(s,b) = \int_\theta b(\theta) V_s(\theta) \, d\theta$
  - Bellman’s equation
    - $V_s(\theta) = \max_a U(s,a) + \sum_{s'} \Pr(s'|s,a,\theta) \, V_{s'}(\theta)$
      - $= \max_a k_a + \sum_{s'} \theta_{sas'} \max_i \text{poly}_i(\theta)$
      - $= \max_j \text{poly}_j(\theta)$
BEETLE Algorithm

- Sample a set of reachable belief points $B$
- $V \leftarrow \{0\}$
- Repeat
  - $V' \leftarrow \{\}$
  - For each $b$ in $B$ compute multivariate polynomial
    - $\text{poly}_{as'}(\theta) \leftarrow \arg\max_{\text{poly} \in V} \int \theta \ b_{sas'}(\theta) \ \text{poly}(\theta) \ d\theta$
    - $a^* \leftarrow \arg\max_a \int \theta \ b_{sas'}(\theta) \ R(s,a) + \Sigma_{s'} \ \theta_{sas'} \ \text{poly}_{as'}(\theta) \ d\theta$
    - $\text{poly}(\theta) \leftarrow U(s,a^*) + \Sigma_{s'} \ \theta_{sa^*s'} \ \text{poly}_{a^*s'}(\theta)$
    - $V' \leftarrow V' \cup \{\text{poly}\}$
  - $V \leftarrow V'$
Polynomials

• Computational issue:
  – # of monomials in each polynomial grows by $O(|S|)$ at each iteration

  – $poly(\theta) = U(s,a^*) + \sum_{s'} \theta_{sa^*s'} poly_{a^*s'}(\theta)$
    $= U(s,a^*) + \sum_{s'} \theta_{sas'} \sum_i mono_i(\theta)$
    $= U(s,a^*) + \sum_{i,s'} mono_{i,s'}(\theta)$

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

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

• Find best coefficients $c_i$ by minimizing $L_n$ norm:
  - $\operatorname{Min}_c \int_{\theta} \left|poly(\theta) - \sum_i c_i \phi_i(\theta)\right|^n d\theta$

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

• Which monomials should we use as basis functions?

• Recall that:
  – \( b_{sas'}(\theta) = k \ b(\theta) \ \theta_{sas'} \)
  – \( poly(\theta) \leftarrow U(s,a) + \sum_{s'} \theta_{sas'} \ poly_{as'}(\theta) \)

• Hence we use beliefs as basis functions
Beetle summary

- Offline: optimize policy at sampled belief points
  - Time: minutes to hours
- Online: learn transition model by 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
Empirical Evaluation

- 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
## Empirical Evaluation

| 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                 |


**Informative Priors**

<table>
<thead>
<tr>
<th>Problem</th>
<th>Opt</th>
<th>Informative priors</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>k = 0</td>
<td>k = 10</td>
<td>k = 20</td>
<td>k = 30</td>
</tr>
<tr>
<td>Chain3</td>
<td>3677</td>
<td>1754 ± 42</td>
<td>3453 ± 47</td>
<td>2034 ± 57</td>
<td>3656 ± 32</td>
</tr>
<tr>
<td>Handw2</td>
<td>1153</td>
<td>1082 ± 17</td>
<td>1056 ± 18</td>
<td>1097 ± 17</td>
<td>1106 ± 16</td>
</tr>
<tr>
<td>Handw3</td>
<td>1083</td>
<td>385 ± 10</td>
<td>540 ± 10</td>
<td>1056 ± 12</td>
<td>1056 ± 12</td>
</tr>
</tbody>
</table>
Learning Curves

- Optimal (utopic)
- Beetle (prior 0)
- Beetle (prior 30)
- Exploit
Conclusion

• Motivation
  – Learning by interaction with environment (no simulation)
  – Bear consequence of actions
  – Minimal exploration
  – Real-time execution

• Bayesian RL
  – Optimizes exploration/exploitation tradeoff
  – Can easily encode prior knowledge to reduce exploration

• Contributions
  – Optimal value function parameterization: as the upper envelope of multivariate polynomials
  – BEETLE algorithm
Future work

- Learn user behaviors for assistive technologies
- Consider partially observable domains
- Learn dynamic transition models
- Consider correlated Dirichlet priors