Action Selection in Bayesian Reinforcement Learning

Tao Wang
Background Perspective

- Be *Bayesian* about reinforcement learning
- Ideal representation of uncertainty for action selection

Why are Bayesian approaches not prevalent in RL?

- Computational barriers
My Work

- Practical algorithms for approximating Bayes optimal decision making

- Analogy to *game-tree search*
  - on-line lookahead computation
  - global value function approximation
    (but here expecti-max vs. mini-max)

- Two key parts:
  - build a lookahead tree (ICML 2005)
  - approximate leaf values (AAAI 2006)
Sequential Decision Making

How to make an optimal decision?

"Planning"

Requires model $P(r, s'| s, a)$

\[ \max_a \]

This is: finite horizon, finite action, finite reward case

General case: Fixed point equations:

\[ V(s) = \sup_a Q(s, a) \quad Q(s, a) = E_{r, s'| s, a} [r + \gamma V(s')] \]
Reinforcement Learning

Do not have model $P(r, s' | s, a)$
Reinforcement Learning

Cannot Compute $E_{r,s'|s,a}$

Do not have model $P(r,s'|s,a)$
Reinforcement Learning

Standard approach: keep point estimate
e.g. via local Q-value estimates

Do not have model $P(r,s'|s,a)$

How to select action?

Greedy

Problem: greedy does not explore
Reinforcement Learning

How to explore?

Problem: do not account for uncertainty in estimates
Reinforcement Learning

Intuition:
greater uncertainty → greater potential

Interval estimation

How to use uncertainty?

Problem: $\delta$’s computed myopically: doesn’t consider horizon
Bayesian Reinforcement Learning

Prior $P(\theta)$ on model $P(r,s'|sa, \theta)$

Belief state $b = P(\theta)$

meta-level state

Choose action to maximize long term reward

Meta-level MDP

decision

a

outcome

Meta-level Model

$r, s', b'$

$P(r,s'b'|s b,a)$

decision

$a'$

outcome

Meta-level Model

$r', s'', b''$

$P(r',s''b''|s' b',a')$

Have a model for meta-level transitions!

- based on posterior update and expectations over base-level MDPs
Bayesian RL Decision Making

How to make an optimal decision?

Solve planning problem in meta-level MDP:
- Optimal Q,V values

Problem: meta-level MDP much larger than base-level MDP
Impractical
Bayesian RL Decision Making

Current approximation strategies:

Consider current belief state $b$

Greedy approach:

- Draw a base-level MDP
- $b \rightarrow \text{mean base-level MDP model}$
- $\rightarrow \text{point estimate for } Q, V$
- $\rightarrow \text{choose greedy action}$

But doesn’t consider uncertainty
Bayesian RL Decision Making

Current approximation strategies:

Consider current belief state \( b \)

Draw a base-level MDP

Thompson approach:

current \( b \) $\rightarrow$ sample a base-level MDP model
$\rightarrow$ point estimate for \( Q, V \)

(Choose action proportional to probability it is max \( Q \))

 Exploration is based on uncertainty

But still myopic
Part 1
Bayesian Sparse Sampling

Bayesian Sparse Sampling for On-line Reward Optimization
Tao Wang    Daniel Lizotte    Michael Bowling    Dale Schuurmans
ICML 2005
Sparse Sampling

(Kearns, Mansour, Ng 2001)

- Approximate values
- Enumerate action choices
- Subsample action outcomes
- Bound depth
- Back up approx values

+ Chooses approximately optimal action with high probability (if depth, sampling large enough)

- Achieving guarantees too expensive

+ But can control depth, sampling
Bayesian Sparse Sampling
Observation 1

• Do not need to enumerate actions in a Bayesian setting
  – Given random variables $Q_1, \ldots, Q_K$
  – and a prior $P(Q_1, \ldots, Q_K)$
  – Can approximate $\max(Q_1, \ldots, Q_K)$
  – Without observing every variable

(Stop when posterior probability of a significantly better Q-value is small)
Bayesian Sparse Sampling
Observation 2

- Action value estimates are not equally important
  - Need better Q value estimates for some actions but not all
  - Preferentially expand tree under actions that might be optimal

**Biased tree growth**
Use Thompson sampling to select actions to expand
Bayesian Sparse Sampling
Observation 3

Correct leaf value estimates to same depth

Use mean MDP Q-value multiplied by remaining depth

effective horizon N=3
Bayesian Sparse Sampling
Observation 4

Include greedy action at decision nodes (if not sampled)
Bayesian Sparse Sampling

Tree growing procedure

1. Sample prior for a model
2. Solve action values
3. Select the optimal action

- Descend sparse tree from root
  - Thompson sample actions
  - Sample outcome
- Until new node added
- Repeat until tree size limit reached

Control computation by controlling tree size
Application: Gaussian process bandits

- General action spaces
  - Continuous actions, multidimensional actions
- Gaussian process prior over reward models
  - Covariance kernel between actions
- Action rewards correlated
- Posterior is a Gaussian process

Robotic application: AIBO walking
Gaussian process experiments

- 2 dimensional continuous action space
- GP priors RBF kernel
- Sampled model from prior
- Run action selection strategies
- Repeat 3000 times
- Average accumulated reward per step
2-dimensional Continuous Gaussian Process

Average Reward per Step vs. Horizon

- eps-Greedy
- Boltzmann
- Interval Est.
- Thompson
- MVPI
2-dimensional Continuous Gaussian Process

Average Reward per Step

Horizon

eps-Greedy
Boltzmann
Interval Est.
Thompson
MVPI
Sparse Samp.
Bayes Samp.
Summary

Bayesian sparse sampling

• Flexible and practical technique for improving action selection

• Reasonably straightforward

• Bandit problems
  – Planning is “easy”
    (at least approximate planning is “easy”)
Question:
How to approximate leaf values?

:\hat{Q}
\hat{V}
Part 2
Approximate value function

Compact, Convex Upper Bound Iteration for Approximate POMDP Planning
Tao Wang  Pascal Poupart  Michael Bowling  Dale Schuurmans
AAAI2006
POMDP model

Observation function
\[ p(o' \mid a, s') \]

Transition function
\[ p(s' \mid s, a) \]

Reward function
\[ R(s, a) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots + \gamma^t r_t \]

Goal: choose actions to maximize long term reward
Belief state

- Probability distribution over underlying states

\[ b = \begin{bmatrix} p(s^1) \\ p(s^2) \\ \vdots \\ p(s^{|S|}) \end{bmatrix} \]

- Sufficient summary of history for decision making

\[ b(\bar{s}_t) = p(\bar{s}_t | b_0 a_1 o_1 a_2 o_2 \cdots a_t o_t) \]
POMDP solving

- Value function is expected total discounted future reward starting from each belief state

Optimal Value function

\[ V^*(b) = \max_a r(b, a) + \gamma \sum_{b'} p(b' | b, a) V^*(b') \]

- Hard to approximate
POMDP approximation approaches

• Value function approximation (Part 2)
  – Hauskrecht 2000
  – Spaan&Vlassis 2005
  – Pineau et al. 2003
  – Parr&Russell 1995

• Policy based optimization
  – Ng&Jordan 00; Poupart & Boutilier 03,04; Amato et al. 06

• Stochastic sampling (Part 1)
  – Kearns et al. 02; Thrun 00
Value function based approaches

- **Optimal value function**
  (satisfies Bellman equation)

\[
V^*(b) = \max_a r(b, a) + \gamma \sum_{b'} p(b'|b, a)V^*(b')
\]

\[
= \max_a r(b, a) + \gamma \sum_{o'} p(o'|b, a)V^*(b'_{(b,a,o')})
\]

- **Difficulty:** belief space is continuous & high dimensional
Optimal 1-step decision

\[ V_1(b) = \max_a b \cdot r_a \]

\[ \Gamma_1 = \{ r_{a_1}, r_{a_2}, r_{a_3} \} = \{ \alpha_1, \alpha_2, \alpha_3 \} \]

Optimal value function is \textcolor{red}{\textbf{piecewise linear convex}}
Optimal n+1-step decision

Value function representation

\[ V_n(b) = \max_{\alpha_{\pi'} \in \Gamma_n} b \cdot \alpha_{\pi'} \]

\[
\begin{bmatrix}
    p(s_1) \\
p(s_2) \\
    \vdots \\
p(s_{|s|})
\end{bmatrix}
\]

\[
\begin{bmatrix}
    v_{\pi'}(s_1) \\
v_{\pi'}(s_2) \\
    \vdots \\
v_{\pi'}(s_{|s|})
\end{bmatrix}
\]

\[ \Gamma_n = \{ \alpha_{\pi'} : \pi' \in \Pi_n \} \]

Value function iteration

\[ V_{n+1}(b) = \max_a r(b, a) + \gamma \sum_{o'} p(o' | b, a) V_n\left(b'_{(b, a, o')}\right) \]

\[ = \max_{a, \{o' \rightarrow \pi'\}} b \cdot \alpha_{a, \{o' \rightarrow \pi'\}} \]
Current approximation strategies

• Grid based approach
  – Gorden 95
  – Hauskrecht 00
  – Zhou & Hansen 01
  – Bonet 02

• Belief point approach
  – Pineau et al. 03
  – Smith & Simmons 05
  – Spaan & Vlassis 05

value function representation
\(\alpha\)-vectors
Our idea

Approximate $V^*(b)$ with a **convex quadratic** upper bound

- Maintain compact (bounded size) representation of value approximation
- Can still model multiple $\alpha$-vectors
- Can be optimized easily
Quadratic approximation

Value function representation
\[ \hat{V}(b) = b^\top W b + w^\top b + \omega \]

Would like to enforce
\[ \hat{V}_{n+1}(b) \geq \max_a \hat{q}_a(b) \]

Need action-value backup for each action
\[ \hat{q}_a(b) = r(b, a) + \gamma \sum_{o'} p(o' | b, a) \hat{V}_n(b'_{(b,a,o')}) \]
Quadratic approximation

Combine with belief update

\[ b'_{(b,a,o')} = \frac{M_{a,o'}b}{e^\top M_{a,o'}b} \]

\[ M_{a,o'}(s', s) = p(o'|a, s')p(s'|s, a) \]

Get action-value

\[ q_a(b) = r(b, a) + \gamma \sum_{o'} \frac{b^\top M_{a,o'}^\top W M_{a,o'}b}{e^\top M_{a,o'}b} + (w + \omega e)^\top M_{a,o'}b \]

**Theorem 1** \( q_a(b) \) is convex in \( b \).

**Corollary 1** \( \max_a q_a(b) \) is convex in \( b \).
Algorithm

\[ \hat{V}_n \quad \hat{q}_a(b) \quad \max_a \hat{q}_a(b) \quad \hat{V}_{n+1} \]

Maintain tight upper bound of the maximum of the action-values
Mathematically

Optimization problem

$$\min_{W,w,\omega} \int_b \left( b^\top W b + w^\top b + \omega \right) \mu(b) \, db$$

subject to

$$b^\top W b + w^\top b + \omega \geq q_a(b), \quad \forall \, a, b$$

$$W \succeq 0 \quad \text{(positive semi-definite)}$$

a measure over space of possible beliefs

ensure upper bound

ensure convexity
Two difficulties

- Integral in objective
- Infinite number of linear constraints

But it is a convex optimization problem (SDP plus infinitely many linear constraints)
Integral

Objective

\[ \int_b (b^T W b + w^T b + \omega) \mu(b) db \]

is equal to

\[ \langle W, E[bb^T] \rangle + w^T E[b] + \omega \]  \hspace{1cm} \text{(linear)}

Assume measure \( \mu(b) \) is Dirichlet distribution on \( b \)

then \( E[bb^T] \) and \( E[b] \) have \textbf{closed form}
Infinite constraints

Have $b^\top W b + w^\top b + \omega \geq q_a(b), \quad \forall \ a, \ b$

ininitely many linear constraints on $Ww\omega$

**Optimal constraint generation:** most violated constraint

$$\min_b b^\top W b + w^\top b + \omega - q_a(b)$$

subject to $b \geq 0$, $\sum_s b(s) = 1$

*Unfortunately, not necessarily a convex minimization problem in $b$*
Experimental results

- Benchmark problems
- Mean discounted reward & Run time
  - 10 runs
  - 1000 trajectories
- Competitors
  - Perseus (Pineau et al. 2003)
  - PBVI (Spaan & Vlassis 2005)
## Problem characteristics

| Problems     | $|S|$ | $|A|$ | $|O|$ |
|--------------|-----|-----|-----|
| Maze         | 20  | 6   | 8   |
| Tiger-grid   | 33  | 5   | 17  |
| Hallway      | 57  | 5   | 21  |
| Hallway2     | 89  | 5   | 17  |
| Aircraft     | 100 | 10  | 31  |
Mean discounted reward

<table>
<thead>
<tr>
<th>Avg. Reward</th>
<th>CQUB</th>
<th>Perseus</th>
<th>PBVI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hallway</td>
<td>0.58 ±0.14</td>
<td>0.51 ±0.06</td>
<td>0.53 ±0.03</td>
</tr>
<tr>
<td>Hallway2</td>
<td>0.43 ±0.25</td>
<td>0.34 ±0.16</td>
<td>0.35 ±0.03</td>
</tr>
<tr>
<td>Tiger-grid</td>
<td>2.16 ±0.02</td>
<td>2.34 ±0.02</td>
<td>2.25 ±0.06</td>
</tr>
<tr>
<td>Maze</td>
<td>45.35 ±3.28</td>
<td>30.49 ±0.75</td>
<td>46.70 ±2.00</td>
</tr>
<tr>
<td>Aircraft</td>
<td>16.70 ±0.58</td>
<td>12.73 ±4.63</td>
<td>16.37 ±0.42</td>
</tr>
</tbody>
</table>
Compact representation

<table>
<thead>
<tr>
<th>Size</th>
<th>CQUB</th>
<th>Perseus</th>
<th>PBVI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maze</td>
<td>231</td>
<td>460</td>
<td>1160</td>
</tr>
<tr>
<td>Tiger-grid</td>
<td>595</td>
<td>4422</td>
<td>15510</td>
</tr>
<tr>
<td>Hallway</td>
<td>1711</td>
<td>3135</td>
<td>4902</td>
</tr>
<tr>
<td>Hallway2</td>
<td>4095</td>
<td>4984</td>
<td>8455</td>
</tr>
<tr>
<td>Aircraft</td>
<td>5151</td>
<td>10665</td>
<td>47000</td>
</tr>
</tbody>
</table>
Summary

New approximation algorithm

- Quadratic value function approximator
- Compact representation
- Competitive approximation quality
- Provable upper bound on the optimal values
- Computational cost independent of iteration number
Contributions

• Use game-tree search ideas
  ✓ Bayesian sparse sampling
  ✓ Approximate POMDP planning
  ➢ To combine them to solve meta-level MDPs

• Exploit Bayesian modelling tools
  – E.g. Gaussian processes
  – Robotic & game applications
Future work on Bayesian sparse sampling

AIBO dog walking
Opponent modeling (Kuhn poker)
Vendor-bot (Pioneer)

Improve tree search?
Theoretical guarantees?
Cheaper re-planning?
Future work on approximate planning

• Set of quadratics
• Use belief state compression & factored models
• Combine with sampling
• Interpretation: $2^{nd}$ order Taylor expansion
Research questions

• How to choose actions during reinforcement learning?

• How to scale up solutions for realistic RL problems?

• How to combine Part 1 and Part 2?
  – Challenge: infinite dimensionality
  – Scale up Part 2 (run time)
Acknowledgments

Dale Schuurmans
Mike Bowling

Rich Sutton
Paul Messinger
Robert Holte

Pascal Poupart
Daniel Lizotte
Adam Milstein