An Analytic Solution to Discrete Bayesian Reinforcement Learning

Authors: Pascal Poupart, Nikos Vlassis, Jesse Hoey, Kevin Regan. ICML 2006

Presented by Yanqi Gu, Nov 15, 2019

Contact: yanqig1@uci.edu

Summary

- Use POMDP formulation of Bayesian RL as problem framework
- Use Point-based value iteration to represent POMDP value function
- Prove $\alpha - functions$ in Bayesian RL are multivariate polynomials
- Parameterize optimal value function by sets of multivariate polynomials

Outline

- POMDP formulation of Bayesian RL
- Offline Approximate Policy Optimization
- The Beetle Algorithm
- Experiments

Motivation

- Problem: Find an optimal policy for an MDP with a partially or completely unknown transition function.
- Author's approach: Analytically derive a simple parameterization of the optimal value function for Bayesian Model-Based approach
- Quick Recap of Bayesian Learning:
- Pick a prior distribution encoding the learner's initial belief about the possible values of each unknown parameter(transition dynamics).
- Then, whenever a sampled realization of the unknown parameter is observed, update the belief to reflect the observed data.
- Unknow transition parameter $T(s', a, s)$: unknown parameter $\theta_a^{s,s'}$

Outline

- **POMDP formulation of Bayesian RL**
- Offline Approximate Policy Optimization
- The Beetle Algorithm
- Experiments

POMDP formulation of Bayesian RL

- Formulate Model-Based, Bayesian RL as a Partially Observable Markov Decision Process: $\langle \mathcal{S}_P, \mathcal{A}_P, \mathcal{O}_P, T_P, Z_P, R_P \rangle$
- $S_P = S \times \{\theta_a^{s,s'}\}$, $O_P = S$: observable MDP state space
- $T_P(s, \theta, a, s', \theta') = \Pr(s', \theta' | s, \theta, a)$ factored in $\theta_a^{s,s'}$ and $\Pr(\theta|\theta') = \delta_\theta(\theta')$
- $\delta_\theta(\theta')$ = 1 when $\theta = \theta'$ and 0 otherwise, which reflects assumption that unknow parameters are stationary.
- $Z_P(s', \theta', a, o) = \Pr(o|s', \theta', a) = \delta_{s'}(o)$
- $R_P(s, \theta, a, s', \theta') = R(s, a, s')$

Dirichlet

- Dirichlets are conjugate priors of multinomials.
- A Dirichlet distribution over a multinomial *p* is parameterized by positive numbers n_i, such that $n_i - 1$ can be interpreted as the number of times that the p_i-probability event has been observed

$$\mathcal{D}(p; n) = k \Pi_i p_i^{n_i - 1}$$

Learn transition model θ by belief monitoring

Using Bayes theorem, at each step, $b(\theta) = \Pr(\theta)$over all unknown parameters $\theta_a^{s,s'}$ is updated observing transitions s, a, s'

Here we assume b is a product of Dirichlets

$$
\begin{aligned}
b_a^{s,s'}(\theta) &= k b(\theta) \Pr(s'|\theta, s, a) && (1) \\
&= k b(\theta) \theta_a^{s,s'} && (2)
\end{aligned}
$$

The unknown transition model is made up of one unknown distribution θ_a^s per s,a pair. Use Dirichlet distribution $D(p; n) = k \prod_i p_i^{n_i - 1}$ so $b(\theta) = \prod_{s,a} D(\theta_a^s; n_a^s)$, n_a^s is a vector of hyperparameters $n_a^{s,s'}$

The posterior obtained after transition $\hat{s}, \hat{a}, \widehat{s'}$ is

k: normalizing constant

$$
\begin{aligned}
b_a^{s,s'}(\theta) &= k \theta_a^{s,s'} \Pi_{s,a} \mathcal{D}(\theta_a^s; n_a^s) && (3) \\
&= \Pi_{s,a} \mathcal{D}(\theta_a^s; n_a^s + \delta_{\hat{s},\hat{a},\hat{s}'}(s, a, s')) && (4)
\end{aligned}
$$

In practice, belief monitoring is as simple as incrementing the hyperparameter corresponding to the observed transition

Outline

- POMDP formulation of Bayesian RL
- **Offline Approximate Policy Optimization**
- The Beetle Algorithm
- Experiments

Bellman Equation

- In POMDPs, $\pi(b) = a$. V^{π} is the policy value measured by the discounted sum of the rewards earned while executing it.
$$V^{\pi}(b) = \sum_{t=0}^{\infty} \gamma^t R(b_t, \pi(b_t), b_{t+1})$$
- The optimal value function satisfies Bellman's equation, piecewise linear and convex $$V^*(b) = \max_a \sum_o \Pr(o|b,a)[R(b,a,b_a^o) + \gamma V^*(b_a^o)]. \quad (5)$$
- Duff(2002): $$V_s^*(b) = \max_a \sum_o \Pr(o|s,b,a)[R(s,b,a,s',b_a^o) + \gamma V_{s'}^*(b_a^o)]. \quad (6)$$
- Simplified: $$V_s^*(b) = \max_a \sum_{s'} \Pr(s'|s,b,a)[R(s,a,s') + \gamma V_{s'}^*(b_a^{s,s'})]. \quad (7)$$

Exploration/Exploitation Tradeoff

$$V_s^*(b) = \max_a \sum_{s'} \Pr(s'|s,b,a)[R(s,a,s') + \gamma V_{s'}^*(b_a^{s,s'})]. \quad (7)$$

$$V_s^*(b) = \max_a \sum_{s'} \Pr(s'|s,b,a)[R(s,a,s') + \gamma V_{s'}^*(b)] \quad (8)$$

Pure exploit

(8) choose actions that maximize total rewards based on b(exploitation) only

(7) use information gained by observing outcome of chosen action, use conditional planning to hypothesize future action outcomes(exploration)

Conclusion: an optimal policy learned by (7) optimizes the exploration/exploitation tradeoff

Optimal Value Function Parameterization

- In Bayesian RL, the optimal value function corresponds to the upper envelope of a set Γ of linear segments called α-functions due to the continuous nature of θ: $V_s^*(b) = max_{\alpha\in\Gamma}\alpha_s(b))$, α can be defined as a linear function of b or θ subscripted by s
- Existing algorithms are computational intensive or make drastic approximations
- Theorem 1: α-functions in Bayesian RL are multivariate polynomials. Proved by induction
- Multivariate polynomials form a closed representation for α-functions under Bellman backups.
- Bellman backup and updating α-functions :
- 1.Suppose optimal value function: $V_s^k(b) = max_{\alpha\in\Gamma^k}\alpha_s(b)$ for k steps-to-go
- 2.Use Bellman's equation to compute best set Γ^{k+1} for optimal value function $V_s^{k+1}(b)$
- Rewrite value function without V^k :

$$V_s^{k+1}(b) = \max_a \sum_{s'} \Pr(s'|s,b,a)[R(s,a,s')+\gamma \max_{\alpha\in\Gamma^k} \alpha_{s'}(b_a^{s,s'})]$$

states. The belief is a sufficient statistic for a given history:

$$b_t := Pr(s_t \mid b_0, a_0, o_1, ..., o_{t-1}, a_{t-1}, o_t) \tag{1}$$

and is updated at each time-step to incorporate the latest action, observation pair:

$$b_t(s') := \eta\Omega(o, s', a) \sum_{s \in S} T(s, a, s') b_{t-1}(s) \tag{2}$$

where η is the normalizing constant.

The goal of POMDP planning is to find a sequence of actions $\{a_0, \ldots, a_t\}$ maximizing the expected sum of rewards $E[\sum_t \gamma^t R(s_t, a_t)]$. Given that the state is not necessarily fully observable, the goal is to maximize expected reward for each belief. The value function can be formulated as:

$$V(b) = \max_{a \in A} \left[R(b, a) + \gamma \sum_{b' \in B} T(b, a, b') V(b') \right] \tag{3}$$

When optimized exactly, this value function is always piecewise linear and convex in the belief [Sondik, 1971] (see Fig. 1, left side). After n consecutive iterations, the solution consists of a set of α-vectors: $V_n = \{\alpha_0, \alpha_1, ..., \alpha_m\}$. Each α-vector represents an $|S|$-dimensional hyper-plane, and defines the value function over a bounded region of the belief: $V_n(b) = \max_{\alpha \in V_n} \sum_{s \in S} \alpha(s) b(s)$. In addition, each α-vector is associated with an action, defining the best immediate policy assuming optimal behavior for the following $(n-1)$ steps (as defined respectively by the sets $\{V_{n-1}, ..., V_0\}$).

The n-th horizon value function can be built from the previous solution V_{n-1} using the *Backup* operator, H. We use notation $V = HV'$ to denote an exact value backup:

$$\begin{aligned} V(b) \quad = \quad & \max_{a \in A} \Bigg[\sum_{s \in S} R(s, a) b(s) + \qquad (4) \\ & \gamma \sum_{o \in O} \max_{\alpha' \in V'} \sum_{s \in S} \sum_{s' \in S} T(s, a, s') \Omega(o, s', a) \alpha'(s') b(s) \Bigg] \end{aligned}$$

Optimal Value Function Parameterization

- Decompose Bellman's equation in 3 steps:
- Find maximal α function for each a and s'
- Find the best action a
- Perform actual Bellman backup

$$\alpha_{b,a}^{s,s'} = \underset{\alpha \in \Gamma^k}{\operatorname{argmax}}\, \alpha_{s'}(b_a^{s,s'}) \quad (9)$$

$$a_b^s = \underset{a}{\operatorname{argmax}} \sum_{s'} \Pr(s'|s,b,a)[R(s,a,s') + \gamma \alpha_{b,a}^{s,s'}(b_a^{s,s'})] \quad (10)$$

$$V_s^{k+1}(b) = \sum_{s'} \Pr(s'|s,b,a_b^s)[R(s,a_b^s,s') + \gamma \alpha_{b,a_b^s}^{s,s'}(b_{a_b^s}^{s,s'})] \quad (11)$$

- Also could rewrite equation (11) using α functions w.r.t. θ

$$\sum_{s'} \Pr(s'|s,b,a_b^s)[R(s,a_b^s,s') + \gamma \int_\theta b_{a_b^s}^{s,s'}(\theta)\alpha_{b,a_b^s}^{s,s'}(\theta)d\theta] \quad (12)$$

$$= \sum_{s'} \int_\theta b(\theta) \Pr(s'|s,\theta,a_b^s)[R(s,a_b^s,s') + \gamma \alpha_{b,a_b^s}^{s,s'}(\theta)d\theta] \quad (13)$$

$$= \int_\theta b(\theta) \Big[\sum_{s'} \Pr(s'|s,\theta,a_b^s)[R(s,a_b^s,s') + \gamma \alpha_{b,a_b^s}^{s,s'}(\theta)] \Big] d\theta \quad (14)$$

- For each b, define an α function and together they form a set Γ^{k+1}

$$\alpha_{b,s}(\theta) = \sum_{s'} \Pr(s'|s,\theta,a_b^s)[R(s,a_b^s,s') + \gamma \alpha_{b,a_b^s}^{s,s'}(\theta)]. \quad (15)$$

- Since each $\alpha_{b,s}$ is defined by using optimal action and α functions in Γ^k, then it's optimal.

$$\begin{aligned} V_s^{k+1}(b) &= \int_\theta b(\theta)\alpha_{b,s}(\theta)d\theta && (16) \\ &= \alpha_{b,s}(b) && (17) \\ &= \max_{\alpha \in \Gamma^{k+1}} \alpha_s(b) && (18) \end{aligned}$$

Outline

- POMDP formulation of Bayesian RL
- Offline Approximate Policy Optimization
- **The Beetle Algorithm**
- Experiments

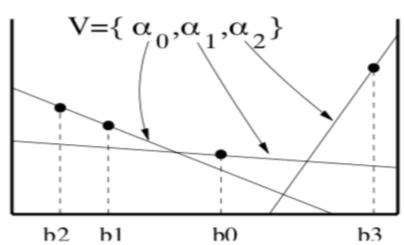
Policy-based value iteration

- Multivariate polynomials form a closed representation for α-functions under Bellman backups
- BEETLE (Bayesian Exploration Exploitation Tradeoff in LEarning)
- A simple and efficient point-based value iteration algorithm
- A set of reachable s, b pairs sampled by simulating several runs of a default or random policy
- For a given s, b pair, the best α-function for each a and s' is computed by (9)
- The optimal action is computed according to (10)
- A new α-function is constructed by (15) and represented by the non-negative powers λ of its monomial terms
- But it suffers from intractability: At each backup, the number of terms of the multivariate polynomial of the α- function grows significantly

3 Point-based value iteration

It is a well understood fact that most POMDP problems, even given arbitrary action and observation sequences of infinite length, are unlikely to reach most of the points in the belief simplex. Thus it seems unnecessary to plan equally for all beliefs, as exact algorithms do, and preferable to concentrate planning on most probable beliefs.

The *point-based value iteration* (PBVI) algorithm solves a POMDP for a finite set of belief points $B = \{b_0, b_1, ..., b_q\}$. It initializes a separate α-vector for each selected point, and repeatedly updates (via value backups) the value of that α-vector. As shown in Figure 1, by maintaining a full α-vector for each belief point, PBVI preserves the piece-wise linearity and convexity of the value function, and defines a value function over the entire belief simplex. This is in contrast to grid-based approaches [Lovejoy, 1991; Brafman, 1997; Hauskrecht, 2000; Zhou and Hansen, 2001; Bonet, 2002], which update only the value at each belief grid point.



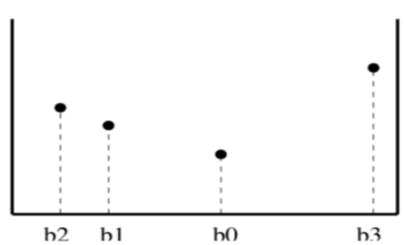


Figure 1: POMDP value function representation using PBVI (on the left) and a grid (on the right).

α function projection

when projecting an α-function onto a linear combination of monomial basis functions (i.e., $\sum_i c_i\phi_i(\theta)$), minimizing an L_n norm yields:

$$\min_{\{c_i\}} \int_\theta |\alpha(\theta) - \sum_i c_i\phi_i(\theta)|^n d\theta \qquad (22)$$

- To mitigate the exponential growth in the number of monomials
- Project each new α-function onto a multivariate polynomial with a smaller number of monomials after each Bellman backup -> minimizes the error at each θ. Optimization
- Pick basis functions as close as possible to the monomials of α-functions
- ▪ In both equations for belief monitoring (4) and backing up α-functions (11), powers are incremented with each s, a, s' transition -> they are made up of similar monomials
- ▪ Use the set of reachable belief states generated at the beginning of the BEETLE algorithm as the fixed basis set
- A fixed basis set allows precomputation of the projection of each backed-up component:
- α-functions can be presented by a column vector (i.e., coefficients of the fixed basis functions)
- A projected transition function in matrix form for each s, a, s' can be pre-computed
- The projection of the reward function can be pre-computed and basis coefficients can be stored
- Point-based backups can be performed by simple matrix operations. Then (15) becomes

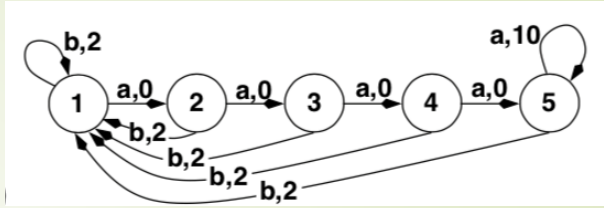
$$\tilde{\alpha}_{b,s} = \sum_{s'} \tilde{T}_a^{s,s'} [\tilde{R}_a^{s,s'} + \gamma \tilde{\alpha}_{b,a}^{s,s'}]. \qquad (23)$$

BEETLES: **effective** online learning

- Effective: focus on time to execute the policy, instead of the offline optimization time
- Actions should be selected in less than a second for realtime execution
- BEETLE: online (belief monitoring and action selection) fast
- Offline: policy optimization(consists of a mapping from state-belief pairs to actions):slow
- The belief states change with each state transition

- **Drawback** of offline policy optimization: the precomputed policy should prescribe an optimal action for every belief state, but this is usually intractable.
- point-based value iteration concentrates its effort on finding good actions at a sample of reachable states

Experiment



The "chain" problem (Strens 2000; Dearden et al., 1998)
The agent has 2 actions a, b that cause transitions between 5 states
At each time step, the agent "slips"
and performs the opposite action with probability $pslip$ = 0.2
Three types of priors:

- Tied: state and action independent
- Semi-tied: action dependent
- Full: extreme (rare) case when dynamics are completed unknown

problem	$\|\mathcal{S}\|$	$\|\mathcal{A}\|$	free params	optimal (utopic)	discrete POMDP	exploit	Beetle	Beetle time (minutes) precomputation	optimization
chain_tied	5	2	1	3677	3661 ± 27	3642 ± 43	3650 ± 41	0.4	1.5
chain_semi	5	2	2	3677	3651 ± 32	3257 ± 124	3648 ± 41	1.3	1.3
chain_full	5	2	40	3677	na-m	3078 ± 49	1754 ± 42	14.8	18.0

Conclusion: Near optimal in tied and semi, but poor in full

Summary

- Using POMDP formulation of Bayesian RL as problem framework
- Using Point-based value iteration to represent POMDP value function
- Prove $\alpha - functions$ in Bayesian RL are multivariate polynomials
- Parameterize optimal value function by sets of multivariate polynomials
- Efficient: explore only truly unknown dynamics; precompute offline a policy and only do action selection and belief monitoring online

Thank you

- Reference:
- An analytical solution to discrete Bayesian RL. ICML 2006
- Point-based value iteration: An anytime algorithm for POMDPs. IJCAI 03
- A Bayesian Sampling Approach to Exploration in Reinforcement Learning. ICML 2009