Bayesian Methods in Reinforcement Learning

Wednesday, June 20th, 2007
ICML-07 tutorial
Corvallis, Oregon, USA

Pascal Poupart (Univ. of Waterloo)
Mohammad Ghavamzadeh (Univ. of Alberta)
Yaakov Engel (Univ. of Alberta)

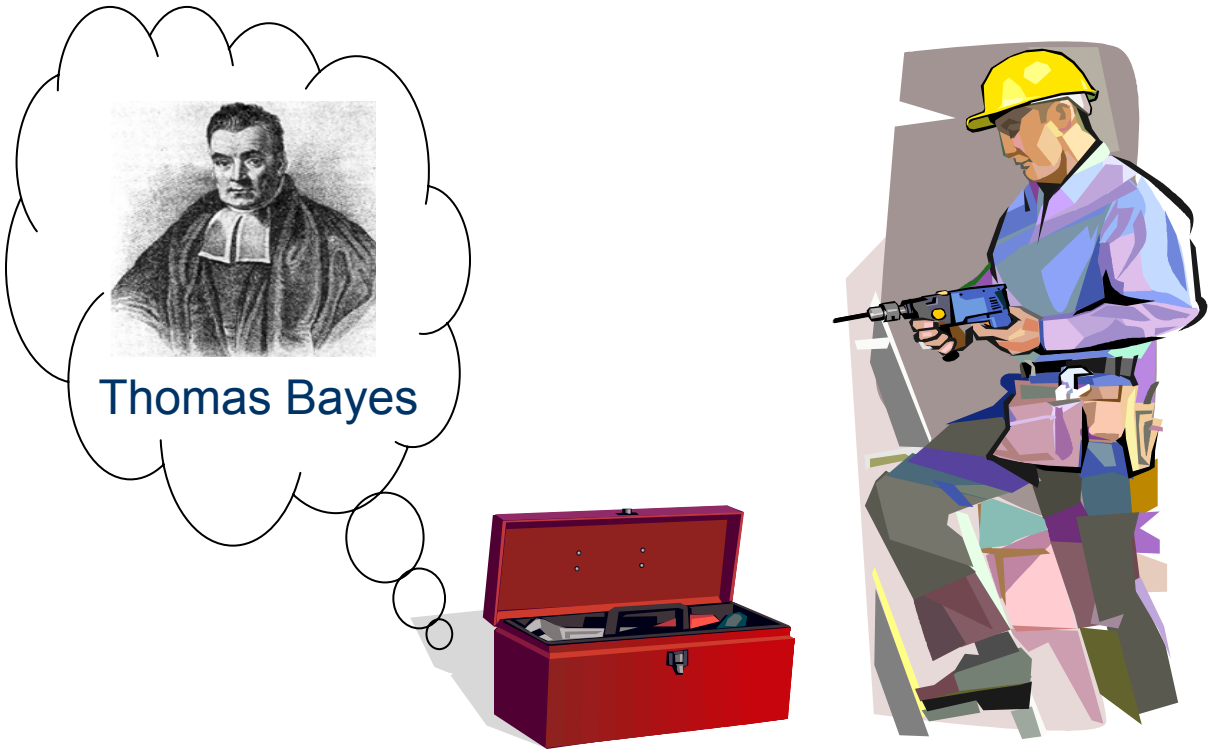
Motivation

- ***Why a tutorial on Bayesian Methods for Reinforcement Learning?***

- Bayesian methods sporadically used in RL
- Bayesian RL can be traced back to the 1950's
- Some advantages:
  - Uncertainty fully captured by probability distribution
  - Natural optimization of the exploration/exploitation tradeoff
  - Unifying framework for plain RL, inverse RL, multi-agent RL, imitation learning, active learning, etc.

Goal

- Add another tool in the toolbox of Reinforcement Learning researchers



Outline

- Intro to RL and Bayesian Learning
- History of Bayesian RL
- Model-based Bayesian RL
  - Prior knowledge, policy optimization, discussion, Bayesian approaches for other RL variants
- Model-free Bayesian RL
  - Gaussian process temporal difference, Gaussian process SARSA, Bayesian policy gradient, Bayesian actor-critique algorithms
- Demo: control of an octopus arm

Outline

- Intro to RL and Bayesian Learning
- **History of Bayesian RL**
- Model-based Bayesian RL
  - Prior knowledge, policy optimization, discussion, Bayesian approaches for other RL variants
- Model-free Bayesian RL
  - Gaussian process temporal difference, Gaussian process SARSA, Bayesian policy gradient, Bayesian actor-critique algorithms
- Demo: control of an octopus arm

Common Belief

- Reinforcement Learning in AI:
  - Formalized in the 1980's by Sutton, Barto and others
  - Traditional RL algorithms are not Bayesian

~~Bayesian RL is a new approach~~

Wrong!

A Bit of History

- RL is the problem of controlling a Markov Chain with unknown probabilities.

- While the AI community started working on this problem in the 1980's and called it Reinforcement Learning, **the control of Markov chains with unknown probabilities had already been extensively studied in Operations Research since the 1950's, including Bayesian methods**.

A Bit of History

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

Bayesian RL Work

- Operations Research
  - Theoretical foundation
  - Algorithmic solutions for special cases
    - Bandit problems: Gittins indices
  - Intractable algorithms for the general case
- Artificial Intelligence
  - Algorithmic advances to improve scalability

Artificial Intelligence

- (Non-exhaustive list)
- **Model-based Bayesian RL:** Dearden et al. (1999), Strens (2000), Duff (2002, 2003), Mannor et al. (2004, 2007), Madani et al. (2004), Wang et al. (2005), Jaulmes et al. (2005), Poupart et al. (2006), Delage et al. (2007), Wilson et al. (2007).
- **Model-free Bayesian RL:** Dearden et al. (1998); Engel et al. (2003, 2005); Ghavamzadeh et al. (2006, 2007).

Outline

- Intro to RL and Bayesian Learning
- History of Bayesian RL
- **Model-based Bayesian RL**
  - Prior knowledge, policy optimization, discussion, Bayesian approaches for other RL variants
- Model-free Bayesian RL
  - Gaussian process temporal difference, Gaussian process SARSA, Bayesian policy gradient, Bayesian actor-critique algorithms
- Demo: control of an octopus arm

Model-based Bayesian RL

- 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

Reinforcement Learning

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

Model Learning

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

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

- What is the prior b?
- Could we choose b to be in the same class as $b_{xax'}$?

Outline

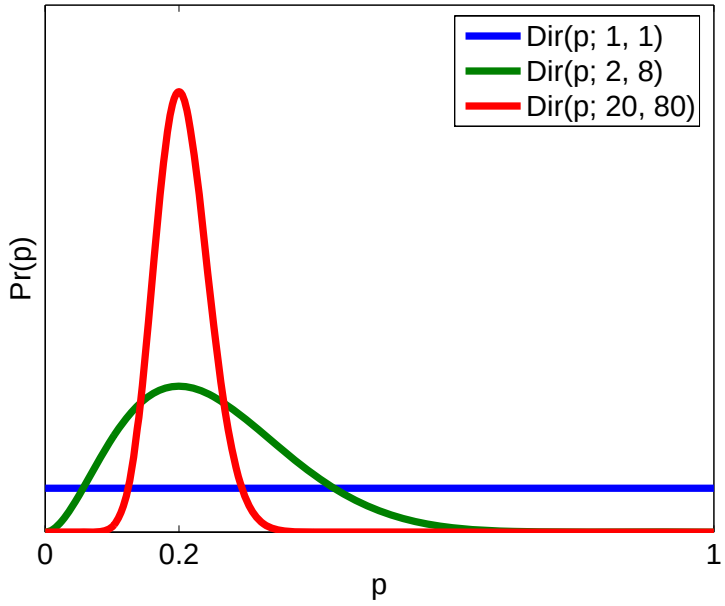
- Intro to RL and Bayesian Learning
- History of Bayesian RL
- Model-based Bayesian RL
  - **Prior knowledge**, policy optimization, discussion, Bayesian approaches for other RL variants
- Model-free Bayesian RL
  - Gaussian process temporal difference, Gaussian process SARSA, Bayesian policy gradient, Bayesian actor-critique algorithms
- Demo: control of an octopus arm

Conjugate Prior

- Suppose b is a monomial in θ
  - i.e. $b(\theta_{xa}) = k \prod_{x''} (\theta_{xax''})^{n_{xax''} - 1}$
- Then $b_{xax'}$ is also a monomial in θ
  - $b_{xax'}(\theta_{xa}) = k [\prod_{x''} (\theta_{xax''})^{n_{xax''} - 1}] \theta_{xax'}$
    $= k \prod_{x''} (\theta_{xax''})^{n_{xax''} - 1 + \delta(x',x'')}$
- Distributions that are closed under Bayesian updates are called **conjugate priors**

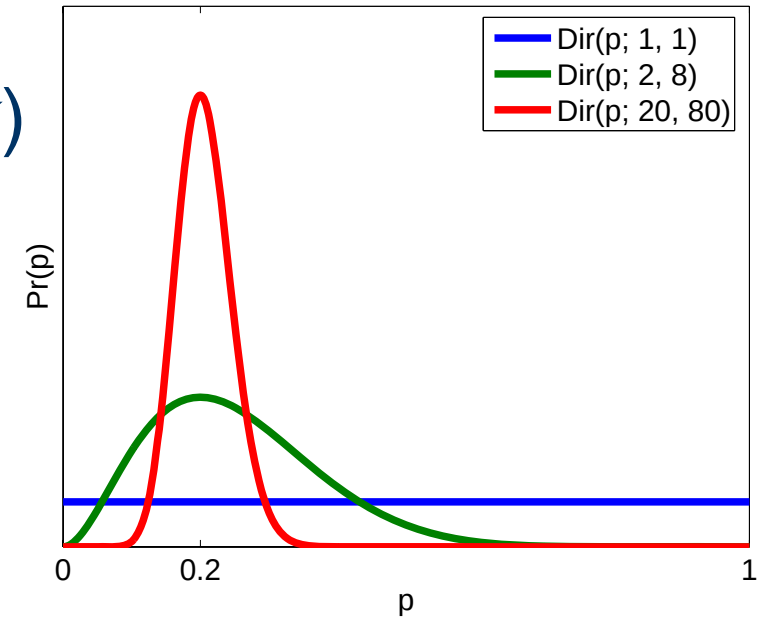
Dirichlet Distributions

- Dirichlets are monomials over discrete random variables:
  - $\mathrm{Dir}(\theta_{xa};n_{xa}) = k \prod_{x''} (\theta_{xax''})^{n_{xax''}-1}$
- Dirichlets are conjugate priors for discrete likelihood distributions



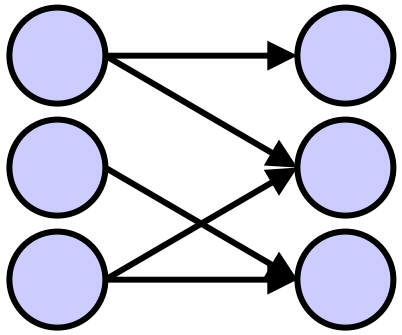
Encoding Prior Knowledge

- No knowledge: uniform distribution
  - E.g., Dir(p; 1, 1)
- I believe p is roughly 0.2, then $(n_1, n_2) \leftarrow (0.2k, 0.8k)$
  - *Dir(p; 0.2k, 0.8k)*
  - *k*: level of confidence



Structural Priors

- Suppose probability of two transitions is the same
  - Tie identical parameters
  - If Pr(•|*x*,*a*) = Pr(•|*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))



Outline

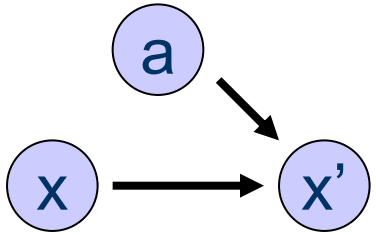
- Intro to RL and Bayesian Learning
- History of Bayesian RL
- Model-based Bayesian RL
  - Prior knowledge, **policy optimization,** discussion, Bayesian approaches for other RL variants
- Model-free Bayesian RL
  - Gaussian process temporal difference, Gaussian process SARSA, Bayesian policy gradient, Bayesian actor-critique algorithms
- Demo: control of an octopus arm

POMDP Formulation

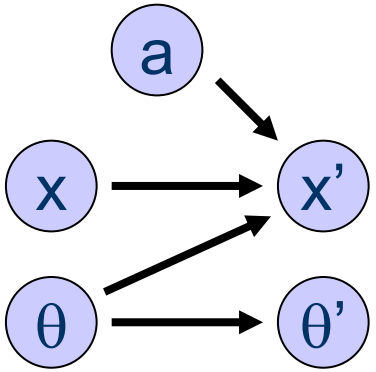
- Traditional RL:
  - X: set of states
  - A: set of actions
  - $p(x'|x,a)$: transition probabilities ← unknown
- Bayesian RL ⟺ POMDP
  - $X \times \theta$: set of states $\langle x, \theta \rangle$
    - x: physical state (observable)
    - θ: model (hidden)
  - A: set of actions
  - $\Pr(x', \theta' | x, \theta, a)$: transition probabilities ← known

Transition Probabilities

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



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



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

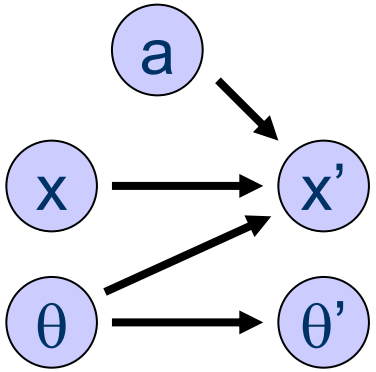
$$\Pr(\theta'|\theta) = \begin{cases} 1 \text{ if } \theta' = \theta \\ 0 \text{ otherwise} \end{cases}$$

Belief MDP Formulation

- Bayesian RL ⟷ POMDP
  - $X \times \theta$: set of states <x,θ>
  - 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>
  - A: set of actions
  - $p(x', b' | x, b, a)$: transition probabilities ⟵ known

Transition Probabilities

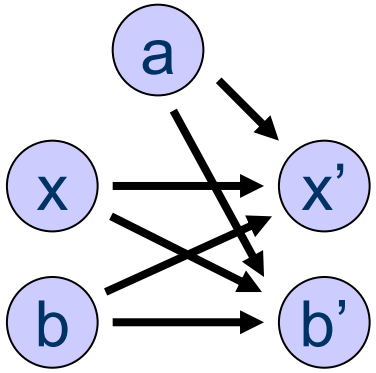
- $\Pr(x', \theta' | x, \theta, a) = \Pr(x' | x, \theta, a) \Pr(\theta' | \theta)$



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

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

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



$$\Pr(x' | x, b, a) = \int_\theta b(\theta) \Pr(x' | x, \theta, a) \, d\theta$$

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

Policy Optimization

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

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

Exploration/Exploitation Tradeoff

- Dilemma:
  - Maximize immediate rewards (exploitation)?
  - Or, maximize information gain (exploration)?
- **Wrong question!**

- Single objective: max expected total rewards
  - $V^{\mu}(x_0) = \Sigma_t \gamma^t E[\, x_{r,t} \,]_{P(x_t|\mu)}$
  - Optimal policy μ^*: $V^{\mu^*}(x) \geq V^{\mu}(x)$ for all x,μ
    - **Optimal exploration/exploitation tradeoff**

Policy Optimization

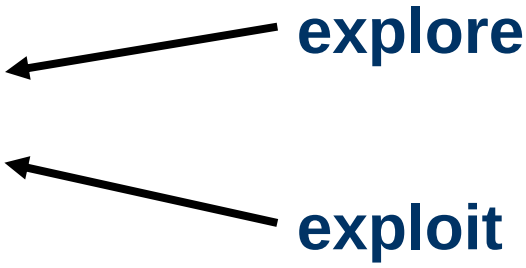
- Use favorite RL/MDP/POMDP algorithm to solve
  - $V^*(x,b) = \max_a \Sigma_{x'} \Pr(x'|x,b,a) [x_r' + \gamma V^*(x', b_{xax'})]$
- Some approaches (non-exhaustive list):
  - Myopic value of information (Dearden et al. 1999)
  - Thompson sampling (Strens 2000)
  - Bayesian Sparse sampling (Wang et al. 2005)
  - Policy gradient (Duff 2002)
  - POMDP discretization (Jaulmes et al. 2005)
  - BEETLE (Poupart et al. 2006)

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
  - $a^* = \text{argmax}_a \underbrace{Q(x,a)}_{\text{exploit}} + \underbrace{MVPI(x,a)}_{\text{explore}}$

Thompson Sampling

- Strens (2000)
- Action selection
  - Sample θ from $b(\theta)$
  - Select best action for θ



- Yields an exploration heuristic

Empirical Comparison

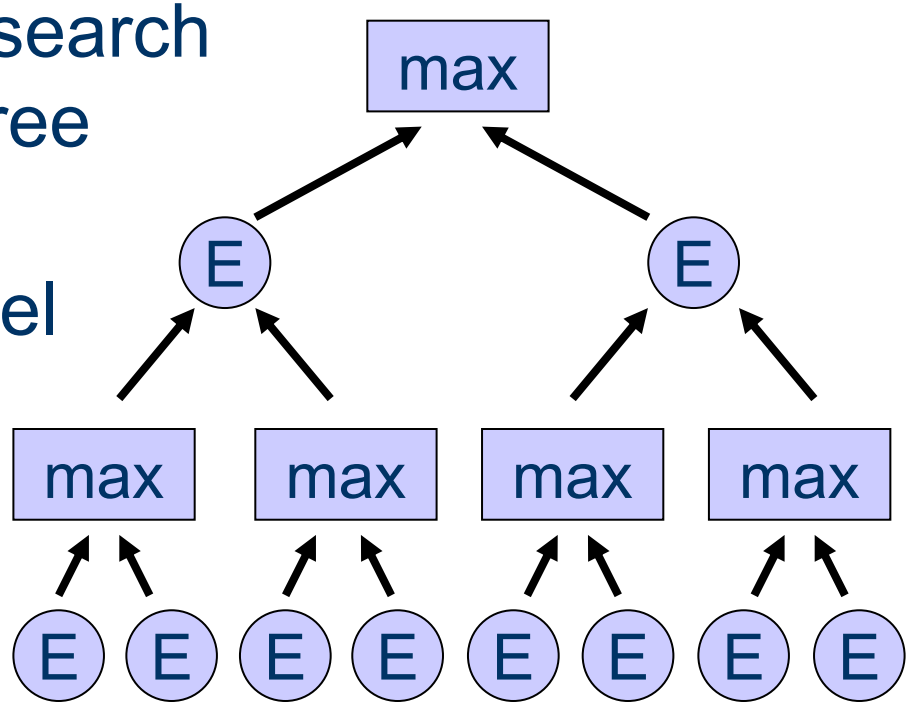
From Strens (2000)

	CHAIN		LOOP	
Method	Phase 1	Phase 2	Phase 1	Phase 2
QL semi-uniform	1594 ± 2	1597 ± 2	337 ± 2	392 ± 1
QL Boltzmann	1606 ± 26	1623 ± 22	186 ± 1	200 ± 1
IEQL+	**2344 ± 78**	**2557 ± 90**	**264 ± 1**	**293 ± 1**
Bayes VPI+MIX	**1697 ± 112**	**2417 ± 217**	**326 ± 31**	**340 ± 31**
Heuristic DP	2855 ± 29	3450 ± 21	314 ± 3	376 ± 2
Bayesian DP	**3158 ± 31**	**3611 ± 27**	**377 ± 1**	**397.5 ± 0.1**

Bayesian Sparse Sampling

- Wang, Lizotte, Bowling & Schuurmans (2005)

- Perform lookahead search by growing sparse tree of reachable beliefs
- Evaluate mean model at the leaves



Policy Gradient

- Duff (2002)

- Policy: stochastic finite-state controller
  - Action selection: Pr(a|n)
  - Node transition: Pr(n'|n,o)
- Estimate gradient by Monte-Carlo sampling
- Policy improvement small steps in gradient direction

POMDP Discretization

- Jaulmes, Pineau and Precup (2005)

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

Policy Optimization

- Bayesian RL:
  - $V^*(x,b) = \max_a \Sigma_{x'} \Pr(x'|x,b,a) [x_r' + \gamma V^*(x', b_{xax'})]$
- Difficulty:
  - b (and θ) 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

Value Function Parameterization

- **Theorem:** V^* is the upper envelope of a set of multivariate polynomials ($V_x(\theta) = \max_i poly_i(\theta)$)
- **Proof:** by induction
  - Define value function in terms of θ instead of b
    - i.e. $V^*(x,b) = \int_\theta b(\theta)\, V_x(\theta)\, d\theta$
  - Bellman's equation
    - $V_x(\theta) = \max_a \Sigma_{x'} \underbrace{\Pr(x'|x,a,\theta)}_{\theta_{xax'}} [\underbrace{x_r'}_{k} + \gamma \underbrace{V_{x'}(\theta)}_{\max_i poly_i(\theta)}]$

      $= \max_a \underbrace{\Sigma_{x'} \theta_{xax'} [k + \gamma \max_i poly_i(\theta)]}_{poly_j(\theta)}$

      $= \max_j poly_j(\theta)$

Partially Observable domains

- 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
    - $poly_{ax'}(\theta) \leftarrow \text{argmax}_{poly \in V} \int_\theta b_{xax'}(\theta)\, poly(\theta)\, d\theta$
    - $a^* \leftarrow \text{argmax}_a \int_\theta b_{sas'}(\theta) \Sigma_{x'} \theta_{xax'} [x_r' + \gamma\, poly_{ax'}(\theta)]\, d\theta$
    - $poly(\theta) \leftarrow \Sigma_{x'} \theta_{xa^*x'} [x_r' + \gamma\, poly_{a^*x'}(\theta)]$
    - $V' \leftarrow V' \cup \{poly\}$
  - $V \leftarrow V'$

Polynomials

- Computational issue:
  - # of monomials in each polynomial grows by $O(|S|)$ at each iteration
  - $poly(\theta) = \Sigma_{x'} \theta_{xa*x'} [x_r' + \gamma\ poly_{a*x'}(\theta)]$
    $= \Sigma_{x'} \theta_{xax'} [x_r' + \gamma \Sigma_i\ mono_i(\theta)]$
    $= x_r' + \gamma \Sigma_{i,x'}\ mono_{i,x'}(\theta)$
- 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. $poly(\theta) \approx \Sigma_i\, c_i\, \phi_i(\theta)$
- Find best coefficients c_i by minimizing L_n norm:
  - $Min_c \int_\theta |poly(\theta) - \Sigma_i\, c_i\, \phi_i(\theta)|^n\, d\theta$
- For the Euclidean norm (L_2), this can be done by solving a system of linear equations $Ax = b$ such that
  - $A_{ij} = \int_\theta \phi_i(\theta)\, \phi_j(\theta)\, d\theta$
  - $b_i = \int_\theta poly(\theta)\, \phi_j(\theta)\, d\theta$
  - $x_i = c_i$

Basis functions

- Which monomials should we use as basis functions?
- Recall that:
  - $b_{xax'}(\theta) = k\, b(\theta)\, \theta_{xax'}$
  - $poly(\theta) \leftarrow \Sigma_{x'}\, \theta_{xax'}\, [x_r' + \gamma\, poly_{ax'}(\theta)]$
- 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

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 θ
  - 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

Problem	Opt	Informative priors			
		k = 0	k = 10	k = 20	k = 30
Chain3	3677	1754 ± 42	3453 ± 47	2034 ± 57	3656 ± 32
Handw2	1153	1082 ± 17	1056 ± 18	1097 ± 17	1106 ± 16
Handw3	1083	385 ± 10	540 ± 10	1056 ± 12	1056 ± 12

Outline

- Intro to RL and Bayesian Learning
- History of Bayesian RL
- Model-based Bayesian RL
  - Prior knowledge, policy optimization, **discussion,** Bayesian approaches for other RL variants
- Model-free Bayesian RL
  - Gaussian process temporal difference, Gaussian process SARSA, Bayesian policy gradient, Bayesian actor-critique algorithms
- Demo: control of an octopus arm

Discussion

- Priors
- Online learning
- Active learning

Misconceptions

- Wouldn't it be better to learn everything from scratch without having to specify any prior?

- **No!**
- There is no such thing as RL without any prior.
- Every learning algorithm has a learning bias
  - Bayesian RL: bias explicit in the prior
  - Other RL techniques: bias implicit but always present
    - Policy search: parameterization of the policy space
    - Value function approximation: type of function approximator

Generalization Assumption

- Consider RL with continuous states
- Approximate V(x) with your favorite approximator
  - polynomial, neural network, radial-basisfunction, etc.
- Common problem: divergence
- Possible cause: Implicit (inaccurate) assumption regarding the generalization across states
- Bayesian RL forces an explicit encoding of the assumptions made
  - Easier to verify that the assumptions are reasonable

Inaccurate priors

- **What if the prior is wrong?**
  - This is the same as asking: *what if the learning bias is wrong?*
- **All RL algorithms use a learning bias that may be wrong. You just have to live with this!**

Inaccurate priors

- **Ok, but I still want to know what will happen if my prior is wrong…**
- A prior is wrong when the probability it assigns to each hypothesis is different from the underlying distribution
- Consequences:
  - Learning may take longer
  - May not converge true hypothesis

Convergence

- Bayesian learning converges to the hypothesis with highest likelihood
  - If the true hypothesis has a non-zero prior probability, Bayesian learning will converge to it (in the limit).
  - If the true hypothesis has zero prior probability, Bayesian learning converges to hypotheses that have highest likelihood of generating the data.
- For n independent pieces of evidence:
  - $\Pr(h|e) = k \Pr(h) \Pr(e_1|h) \Pr(e_2|h) \ldots \Pr(e_n|h)$

Benefits of Explicit Priors

- Facilitates encoding of domain knowledge
- Assumptions made can be easily verified
- Prior information simplifies learning
  - Faster training (assuming good prior)

Online Learning

- Online learning
  - Must bear reward/cost of each action
  - Exploration/exploitation tradeoff
  - Data samples often limited due to interaction with environment
- Bayesian RL
  - Naturally balance exploration and exploitation
  - Facilitates prior knowledge inclusion
    - reduces need for data samples

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)

Outline

- Intro to RL and Bayesian Learning
- History of Bayesian RL
- Model-based Bayesian RL
  - Prior knowledge, policy optimization, discussion, **Bayesian approaches for other RL variants**
- Model-free Bayesian RL
  - Gaussian process temporal difference, Gaussian process SARSA, Bayesian policy gradient, Bayesian actor-critique algorithms
- Demo: control of an octopus arm

Other variants of RL

- Bayesian methods can also be used for several variants of reinforcement learning:
  - **Bayesian Inverse RL [Ramachandran et al., 2007]**
  - **Bayesian Imitation learning [Price et al., 2003]**
  - **Bayesian coordination [Chalkiadakis et al., 2003]**
  - Bayesian coalition formation [Chalkiadaki et al., 2004]
  - **Bayesian partially observable stochastic games [Gmytrasiewicz & Doshi, 2005]**
  - Bayesian Multi-Task Reinforcement Learning [Wilson et al., 2007]

Bayesian Inverse RL

- Ramachandran and Amir (2007)

- Bayesian inverse RL: $\langle X,A,p,\mu^* \rangle$
- Unknown: R

- Prior: Pr(R)
- Likelihood: $Pr(x,a|R) = ke^{\alpha Q^*(x,a,R)}$
- Posterior: Pr(R|x,a)

Bayesian Inverse RL

- Reward learning
  - $R^* = \text{argmax}_R \Pr(R|x,a)$
- Apprenticeship learning
  - Let $\overline{R} = \Sigma_R \Pr(R|x,a)\, R$
  - μ^* = best policy for $\langle X,A,p,\overline{R}\rangle$
- Advantages:
  - Natural encoding of uncertainty about R
  - Facilitate inclusion of prior knowledge
  - Mentor does not need to be infallible
  - Mentor policy may be only partially known

Bayesian Imitation Learning

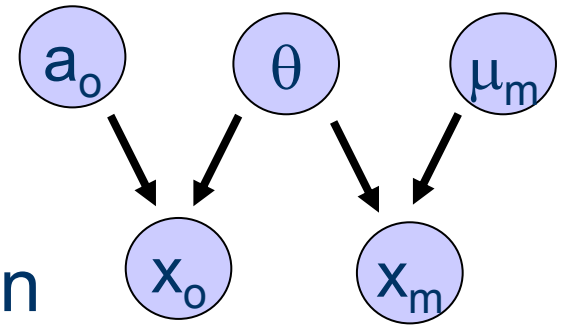
- Price and Boutilier (2003)

- Two agents: learner and mentor
- They share:
  - Same state space
  - Same action space
- Learner observes mentor states, but not mentor actions
- Mentor executes a fix policy (not necessarily optimal), which is unknown to the learner

Bayesian Imitation Learning

- Idea: learner can learn faster by observing mentor's state trajectories

- Two unknowns:
  - θ: model (same for both agents)
  - μ_m: policy of the mentor
- Prior: $Pr(\theta,\mu_m)$
- Posterior: $Pr(\theta,\mu_m|a_o,x_o,x_m)$
- Belief MDP algorithm based on approximate value of information



Bayes. Multiagent Coordination

- Chalkiadakis & Boutilier (2003)

- Multiagent RL: Stochastic Game
- Problem: Multiple equilibria
- Coordination
  - Necessary to converge to the same equilibrium
  - Induces an exploration/exploitation tradeoff
- Bayesian coordination optimizes this tradeoff

Bayes. Multiagent Coordination

- Stochastic Game: $<\alpha, \{A_i\}_{i\in\alpha}, X, p, \{R_i\}_{i\in\alpha}>$
- Unknowns:
  - $\theta = <p, \{R_i\}_{i\in\alpha}>$: model (game)
  - μ_{-i}: other agents' policy
  - H: relevant aspects of game history used by μ_{-i}
- Prior: $Pr(\theta, \mu_{-i}, H)$
- Posterior: $Pr(\theta, \mu_{-i}, H|x,a,r,x')$
- Belief MDP algorithm based on approximate value of information

Partially Observable Stochastic Games (POSGs)

- Gmytrasiewicz and Doshi (2005)
- Interactive-POMDPs: $\langle IS_i, A, p_i, O_i, \Omega_i, R_i \rangle$
  - hierarchical Bayesian formulation of POSGs
  - IS_i: interactive state
  - Ω_i: set of observations
  - $O_i : A, X_i, \Omega_i \rightarrow [0,1]$: observation function
- Nested beliefs: $is_{i,l} = \langle x_i, \theta_{i,l-1} \rangle$
  s.t. $\theta_{i,l-1} = \langle b(is_{-i,l-1}), A, p_i, O_i, \Omega_i, R_i \rangle$

Partially Observable Stochastic Games (POSGs)

- Bayesian POSGs
  - Natural model
  - No assumption of common knowledge among agents
  - Facilitate encoding of prior knowledge

Summary

- History of Bayesian RL
- Formulation of model-based Bayesian RL
- Priors
  - Dirichlets (conjugate priors for multinomials)
  - Inclusion of structure and parameter knowledge
- Natural balance of exploration and exploitation
- Optimal value function
  - Can use favorite RL/MDP/POMDP algorithm
  - Closed form: upper envelope of polynomials
- Bayesian approaches for several variants of RL

Open Problems

- Prior:
  - What are common types of domain knowledge in RL?
  - How to encode this knowledge in a prior?
  - Hierarchical priors for Bayesian RL?
- Belief inference:
  - Non-parametric Bayesian techniques?
  - Monte Carlo techniques?
- Policy optimization
  - Closed-form value functions for continuous domains?
  - Scalable, yet non-myopic approaches?

Bayesian RL Related Surveys

- R. Bellman (1961) Adaptive Control Processes: A Guided Tour, Princeton University Press
- A. Fel'dhaum (1965) Optimal Control Systems, Academic Press, NY
- J.J. Martin (1967) Bayesian Decision Problems and Markov Chains, Wiley & Sons
- D.A. Berry & B. Fristedt (1985) Bandit Problems: Sequential Allocation of Experiments, Chapman & Hall
- P.R. Kumar & P. Varaiya (1986) Stochastic Systems: Estimation, Identification and Adaptive Control, Prentice-Hall
- M.O. Duff (2002) Optimal Learning: Computational Procedures for Bayes-Adaptive Markov Decision Processes, PhD Thesis, University of Massachusetts, Amherst

ICML-07 Papers Related to Bayesian RL

- E. Delage, S. Mannor (2007) Percentile Optimization in Uncertain MDPs with Application to Efficient Exploration, ICML.
- M. Ghavamzadeh, Y. Engel (2007) Bayesian Actor-Critic, ICML.
- A. Krause and C. Guestrin (2007) Nonmyopic Active Learning of Gaussian Processes: an Exploration—Exploitation Approach, ICML.
- S. Pandey, D. Chakrabarti, D. Agarwal (2007) Multi-armed Bandit Problems with Dependent Arms, ICML.
- A. Wilson, A. Fern, S. Ray, P. Tadepalli (2007) Multi-Task Reinforcement Learning: A Hierarchical Bayesian Approach, ICML.