Complexity of stochastic branch and bound methods for belief tree search in Bayesian reinforcement learning

Christos Dimitrakakis

December 31, 2009

Abstract

There has been a lot of recent work on Bayesian methods for reinforcement learning exhibiting near-optimal online performance. The main obstacle facing such methods is that in most problems of interest, the optimal solution involves planning in an infinitely large tree. However, it is possible to obtain stochastic lower and upper bounds on the value of each tree node. This enables us to use stochastic branch and bound algorithms to search the tree efficiently. This paper proposes two such algorithms and examines their complexity in this setting.

1 Introduction

Various Bayesian methods for exploration in Markov decision processes (MDPs) and for solving known partially-observable Markov decision processes (POMDPs), were proposed previously (c.f. [Poupart et al., 2006, Duff, 2002, Ross et al., 2008]). However, such methods often suffer from computational tractability problems. Optimal Bayesian exploration requires the creation of an augmented MDP model in the form of a tree [Duff, 2002], where the root node is the current belief-state pair and children are all possible subsequent belief-state pairs. The size of the belief tree increases exponentially with the horizon, while the branching factor is infinite in the case of continuous observations or actions.

In this work, we examine the complexity of efficient algorithms for expanding the tree. In particular, we propose and analyse stochastic search methods similar to the ones proposed in [Bubeck et al., 2008, Norkin et al., 1998]. Related methods have been previously examined experimentally in the context of Bayesian reinforcement learning in [Dimitrakakis, 2008, Wang et al., 2005].

The remainder of this section summarises the Bayesian planning framework. Our main results are presented in Sect. 2. Section 3 concludes with a discussion of related work. Technical proofs and related results are presented in the Appendix.

1.1 Markov Decision Processes

Reinforcement learning [c.f. Puterman [1994,2005]] is discrete-time sequential decision making problem, where we wish to act so as to maximise the expected sum of discounted future rewards $\mathbf{E}\sum_{k=1}^{T}\gamma^k r_{t+k}$, where $r_t \in \mathbb{R}$ is a stochastic reward at time t. We are only interested in rewards from time t to $T > 0$, and $\gamma \in [0,1]$ plays the role of

a discount factor. Typically, we assume that γ and T are known (or have known prior distribution) and that the sequence of rewards arises from a Markov decision process μ:

Definition 1 (MDP) *A Markov decision process is a discrete-time stochastic process with: A state $s_t \in \mathcal{S}$ at time t and a reward $r_t \in \mathbb{R}$, generated by the process μ, and an action $a_t \in \mathcal{A}$, chosen by the decision maker. We denote the distribution over next states s_{t+1}, which only depends on s_t and a_t, by $\mu(s_{t+1}|s_t, a_t)$. Furthermore $\mu(r_{t+1}|s_t, a_t)$ is a reward distribution conditioned on states and actions. Finally, $\mu(r_{t+1}, s_{t+1}|s_t, a_t) = \mu(r_{t+1}|s_t, a_t)\mu(s_{t+1}|s_t, a_t)$.*

In the above, and throughout the text, we usually take $\mu(\cdot)$ to mean $\mathbf{P}_\mu(\cdot)$, the distribution under the process μ, for compactness. Frequently such a notation will imply a marginalisation. For example, we shall write $\mu(s_{t+k}|s_t, a_t)$ to mean:

$$\sum_{s_{t+1},\ldots,s_{t+k-1}} \mu(s_{t+k}, \ldots, s_{t+1}|s_t, a_t).$$

The decision maker takes actions according to a policy π, which defines a distribution $\pi(a_t|s_t)$ over $\mathcal{A}$, conditioned on the state s_t, i.e. a set of probability measures over $\mathcal{A}$ indexed by s_t. A policy π is stationary if $\pi(a_t = a|s_t = s) = \pi(a_{t'} = a|s_{t'} = s)$ for all t, t'. The expected utility of a policy π selecting actions in the MDP μ, from time t to T can be written as the *value function*:

$$V_{t,T}^{\pi,\mu}(s) = \mathbf{E}_{\pi,\mu}\left(\sum_{k=1}^{T} \gamma^k r_{t+k} \Big| s_t\right), \tag{1}$$

where $\mathbf{E}_{\pi,\mu}$ denotes the expectation under the Markov chain arising from acting policy π on the MDP μ. Whenever it is clear from context, superscripts and subscripts shall be omitted for brevity. The *optimal* value function will be denoted by $V^* \triangleq \max_\pi V^\pi$. If the MDP is known, we can evaluate the optimal value function policy in time polynomial to the sizes of the state and action sets [Puterman, 1994,2005] via backwards induction (value iteration).

1.2 Bayesian Reinforcement Learning

If the MDP is unknown, we may use a Bayesian framework to represent our uncertainty [Duff, 2002]. This requires maintaining a belief ξ_t, about which MDP $\mu \in \mathcal{M}$ corresponds to reality. More precisely, we define a measurable space $(\mathcal{M}, \mathfrak{M})$, where $\mathcal{M}$ is a (usually uncountable) set of MDPs, and $\mathfrak{M}$ is a suitable σ-algebra. With an appropriate initial density $\xi_0(\mu)$, we can obtain a sequence of densities $\xi_t(\mu)$, representing our subjective belief at time t, by conditioning $\xi_t(\mu)$ on the latest observations:

$$\xi_{t+1}(\mu) \triangleq \frac{\mu(r_{t+1}, s_{t+1}|s_t, a_t)\xi_t(\mu)}{\int_{\mathcal{M}} \mu'(r_{t+1}, s_{t+1}|s_t, a_t)\,\xi_t(\mu')\,\mathrm{d}\mu'}. \tag{2}$$

In the following, we write $\mathbf{E}_\xi$ to denote expectations with respect to any belief ξ.

1.3 Belief-Augmented MDPs

In order to optimally select actions in this framework, it is necessary to *explicitly* take into account future changes in the belief when planning [Duff, 2002]. The idea is to combine the original MDP's state s_t and our belief state ξ_t into a *hyper-state*.

Definition 2 (BAMDP) *A* Belief-Augmented MDP ν *(BAMPD) is an MDP with a set of hyper-states* $\Omega = \mathcal{S} \times \mathfrak{B}$*, where* $\mathfrak{B}$ *is an appropriate set of probability measures on* $\mathcal{M}$ *and* $\mathcal{S}, \mathcal{A}$ *are the state and action sets of all* $\mu \in \mathcal{M}$*. At time* t*, the agent observes the hyper-state* $\omega_t = (s_t, \xi_t) \in \Omega$ *and takes action* $a_t \in \mathcal{A}$*. We write the transition distribution as* $\nu(\omega_{t+1}|\omega_t, a_t)$ *and the reward distribution as* $\nu(r_t|\omega_t)$*.*

The hyper-state ω_t has the Markov property. This allows us to treat the BAMDP as an infinite-state MDP with transitions $\nu(\omega_{t+1}|\omega_t, a_t)$, and rewards $\nu(r_t|\omega_t)$.[1] When the horizon T is finite, we need only require expand the tree to depth $T-t$. Thus, backwards induction starting from the set of terminal hyper-states Ω_T and proceeding backwards to $T-1, \ldots, t$ provides a solution:

$$V_n^*(\omega) = \max_{a \in \mathcal{A}} \mathbf{E}_\nu(r|\omega) + \gamma \sum_{\omega' \in \Omega_{n+1}} \nu(\omega'|\omega, a) V_{n+1}^*(\omega'), \tag{3}$$

where Ω_n is the set of hyper-states at time n. We can approximately solve infinite-horizon problems if we expand the tree to some finite depth, if we have bounds on the value of leaf nodes.

1.4 Bounds on the Value Function

We shall relate the optimal value function of the BAMDP, $V^*(\omega)$, for some $\omega(s,\xi)$, to the value functions V_μ^π of MDPs $\mu \in \mathcal{M}$ for some π. The optimal policy for μ is denoted as $\pi^*(\mu)$. The *mean* MDP resulting from belief ξ is denoted as $\bar{\mu}_\xi$ and has the properties: $\bar{\mu}_\xi(s_{t+1}|s_t, a_t) = \mathbf{E}_\xi[\mu(s_{t+1}|s_t, a_t)]$, $\bar{\mu}_\xi(r_{t+1}|s_t, a_t) = \mathbf{E}_\xi[\mu(r_{t+1}|s_t, a_t)]$.

Proposition 1 *Dimitrakakis [2008] For any* $\omega = (s, \xi)$*, the BAMDP value function* V^* *obeys:*

$$\int_{\mathcal{M}} V_\mu^{\pi^*(\mu)}(s)\xi(\mu) d\mu \geq V^*(\omega) \geq \int_{\mathcal{M}} V_\mu^{\pi^*(\bar{\mu}_\xi)}(s)\xi(\mu)\, d\mu \tag{4}$$

Proof By definition, $V^*(\omega) \geq V^\pi(\omega)$ for all ω, for any policy π. It is easy to see that the lower bound equals $V^{\pi^*(\bar{\mu}_\xi)}(\omega)$, thus proving the right hand side. The upper bound follows from the fact that for any function f, $\max_x \int f(x,u)\, du \leq \int \max_x f(x,u)\, du$. ∎

If $\mathcal{M}$ is not finite, then we cannot calculate the upper bound of $V(\omega)$ in closed form. However, we can use Monte Carlo sampling: Given a hyper-state $\omega = (s, \xi)$, we draw m MDPs from its belief ξ: $\mu_1, \ldots, \mu_m \sim \xi$,[2] estimate the value function for each μ_k, $\tilde{v}_{U,k}^\omega \triangleq V_{\mu_k}^{\pi^*(\mu_k)}(s)$, and average the samples: $\hat{v}_{U,m}^\omega \triangleq \frac{1}{m}\sum_{k=1}^m \tilde{v}_{U,k}^\omega$. Let $v_U^\omega \triangleq \int_{\mathcal{M}} \xi_\omega(\mu) V_\mu^*(s_\omega)\, d\mu$. Then, $\lim_{m\to\infty}[\hat{v}_{U,m}^\omega] = v_U^\omega$ almost surely and $\mathbf{E}[\hat{v}_{U,m}^\omega] = v^\omega$.

Lower bounds can be calculated via a similar procedure. We begin by calculating the optimal policy $\pi^*(\bar{\mu}_\xi)$ for the mean MDP $\bar{\mu}_\xi$ arising from ξ. We then compute $\tilde{v}_{L,k}^\omega \triangleq V_{\mu_k}^{\pi^*(\bar{\mu}_\xi)}$, the value of that policy for each sample μ_k and estimate $\hat{v}_{L,m}^\omega \triangleq \frac{1}{m}\sum_{k=1}^m \tilde{v}_{L,k}^\omega$.

[1] Because of the way that the BAMDP ν is constructed from beliefs over $\mathcal{M}$, the next reward now depends on the next state rather than the current state and action.

[2] In the discrete case, we sample a multinomial distribution from each of the Dirichlet densities independently for the transitions. For the rewards we draw independent Bernoulli distributions from the Beta of each state-action pair.

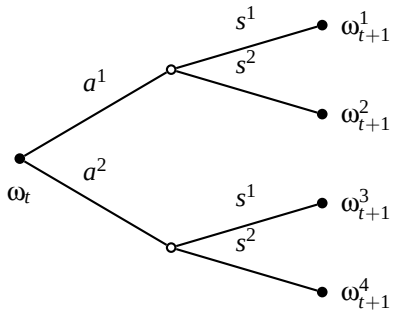

Figure 1: A belief tree, where the rewards are ignored for simplicity, with actions $\mathcal{A} = \{a^1, a^2\}$ and states $\mathcal{S} = \{s^1, s^2\}$.

2 Complexity of belief tree search

We now present our main results. Detailed proofs are given in the appendix. We search trees which arise in the context of planning under uncertainty in MDPs using the BAMDP framework. We can use value function bounds on the leaf nodes of a partially expanded BAMDP tree to obtain bounds for the inner nodes through backwards induction. The bounds can be used both for action selection and for further tree expansion. However, the bounds are estimated via Monte Carlo sampling, something that necessitates the use of stochastic branch and bound technique to expand the tree.

We analyse a set of such algorithms. The first is a search to a fixed depth that employs exact lower bounds. We then show that if only stochastic bounds are available, the complexity of fixed depth search only increases logarithmically. We then present two stochastic branch and bound algorithms, whose complexity is dependent on the number of near-optimal branches. The first of these uses bound samples on leaf nodes only, while the second uses samples obtained in the last half of the parents of leaf nodes, thus using the collected samples more efficiently.

2.1 Assumptions and Notation

We present the main assumptions concerning the tree search, pointing out the relations to Bayesian RL. The symbols V and v have been overloaded to make this correspondence more apparent. The tree that has a branching factor at most ϕ. The branching is due to both action choices and random outcomes (see Fig.1). Thus, the nodes at depth k correspond to the set of hyper-states $\{\omega_{t+k}\}$ in the BAMDP. By abusing notation, we may also refer to the components of each node $\omega = (s, \xi)$ as $s(\omega), \xi(\omega)$.

We define a branch b as a *set* of *policies* (i.e. the set of all policies starting with a particular action). The value of a branch b is $V^b \triangleq \max_{\pi \in b} V^\pi$. The root branch is the set of all policies, with value V^*. A hyper-state ω is b-reachable if $\exists \pi \in b$ s.t $\mathbf{P}_{\pi,\nu}(\omega|\omega_t) > 0$.Any branch b can be partitioned at any b-reachable ω into a set of branches $B(b, \omega)$. A possible partition is any $b_i = \{\pi \in b : i = \arg\max_a \pi(a|\omega)\}$ for any $b_i \in B(b, \omega)$. We simplify this by considering only deterministic policies. We denote the k-horizon value function by $V^b(k) \triangleq \max_{\pi \in b} V^\pi_{t,k}(\omega_t)$. For each tree node $\omega = (s, \xi)$, we define upper and lower bounds $v_U(\omega) \triangleq \mathbf{E}_\xi[V^*_\mu(s)]$, $v_L(\omega) \triangleq \mathbf{E}_\xi[V^{\pi^*(\bar{\mu}_\xi)}(s)]$, from (4). By fully expanding the tree to depth k and performing backwards induction (3), using either v_U or v_L as the value of leaf nodes, we obtain respectively upper and lower

Algorithm 1 Flat oracle search

Expand all branches until depth $k = \log_\gamma \varepsilon/\beta$ or $\hat{\Delta}_L > \beta\gamma^k - \varepsilon$.
Select the root branch $\hat{b}^* = \arg\max_b V_L^b(k)$.

Algorithm 2 Flat stochastic search

FSSEARCH(ω_t, k, m)
Let $\Omega_k = \left\{\omega_{t+k}^i : i = 1, \ldots, \phi^k\right\}$ be the set of all k-step children of ω
for $\omega \in \Omega_k$ **do**
    Draw m samples $\tilde{v}_{L,j}^\omega = V_\mu^\pi$, $\mu \sim \xi(\omega)$
    $\hat{v}_L^\omega = \frac{1}{m}\sum_{j=1}^m \tilde{v}_{L,j}^\omega$,
end for
Calculate $\hat{V}^b$
return $\hat{b}^* = \arg\max \hat{V}^b$.

bounds $V_U^b(k), V_L^b(k)$ on the value of any branch. Finally, we use $\mathcal{C}(\omega)$ for the set of immediate children of a node ω and the short-hand Ω_k for $\mathcal{C}^k(\omega)$, the set of all children of ω at depth k. We assume the following:

Assumption 1 (Uniform linear convergence) *There exists $\gamma \in (0,1)$ and $\beta > 0$ s.t. for any branch b, and depth k, $V^b - V_L^b(k) \leq \beta\gamma^k$, $V_U^b(k) - V^b \leq \beta\gamma^k$.*

Remark 1 *For BAMDPs with $r_t \in [0,1]$ and $\gamma < 1$, Ass. 1 holds, from boundedness and the geometric series, with $\beta = 1/(1-\gamma)$, since $V_L^b(k)$ and $V_U^b(k)$ are the k-horizon value functions with the value of leaf nodes bounded in $1/(1-\gamma)$.*

We analyse algorithms which search the tree and then select an (action) branch $\hat{b}^*$. For each algorithm, we examine the number of leaf node evaluations required to bound the regret $V^* - V^{\hat{b}^*}$.

2.2 Flat Search

With exact bounds, we can expand all branches to a fixed depth and then select the branch $\hat{b}^*$, with the highest lower bound. This is Alg. 1, with complexity given by the following lemma.

Lemma 1 *Alg. 1 on a tree with branching factor ϕ, $\gamma \in (0,1)$, samples $O(\phi^{1+\log_\gamma \varepsilon/\beta})$ times to bound the regret by ε.*

Proof Bound the k-horizon value function error with Ass. 1 and note that there are ϕ^{k+1} leaves. ∎

In our case, we only have a stochastic lower bound on the value of each node. Algorithm 2 expands the tree to a fixed depth and then takes multiple samples from each leaf node.

Lemma 2 *Calling Alg. 2 with $k = \lceil \log_\gamma \varepsilon/2\beta \rceil$, $m = 2\lceil \log_\gamma(\varepsilon/2\beta) \rceil \cdot \log\phi$, we bound the regret by ε using $O\left(\phi^{1+\log_\gamma \varepsilon/2\beta} \log_\gamma(\varepsilon/2\beta) \cdot \log\phi\right)$ samples.*

Algorithm 3 Stochastic branch and bound 1

Let $\mathcal{L}_0$ be the root.
for $n = 1, 2, \ldots$ **do**
  for $\omega \in \mathcal{L}_n$ **do**
    $m_\omega{+}{+}$, $\mu \sim \xi(\omega)$, $\tilde{v}^\omega_{m_\omega} = V^*_\mu(s(\omega))$.
    $\hat{v}^\omega_U = \frac{1}{m_\omega} \sum_{i=1}^{m_\omega} \tilde{v}^\omega_i$
  end for
  $\hat{\omega}^*_n = \arg\max_\omega \hat{v}^\omega_U$.
  $\mathcal{L}_{n+1} = \mathcal{C}(\hat{\omega}^*_n) \cup \mathcal{L}_n \backslash \hat{\omega}^*_n$
end for

Proof The regret now is due to both limited depth and stochasticity. We bound each by $\varepsilon/2$, the first via Lem. 1 and the second via Hoeffding's inequality. ∎

Thus, stochasticity mainly adds a logarithmic factor to the oracle search. We now consider two algorithms which do not search to a fixed depth, but select branches to deepen adaptively.

2.3 Stochastic Branch and Bound 1

A stochastic branch and bound algorithm similar to those examined here was originally developed by Norkin et al. [1998] for optimisation problems. At each stage, it takes an additional sample at each leaf node, to improve their upper bound estimates, then expands the node with the highest mean upper bound. Algorithm 3 uses the same basic idea, averaging the value function samples at every leaf node.

In order to bound complexity, we need to bound the time required until we discover a nearly optimal branch. We calculate the number of times a suboptimal branch is expanded before its suboptimality is discovered. Similarly, we calculate the number of times we shall sample the optimal node until its mean upper bound becomes dominant. These two results cover the time spent sampling upper bounds of nodes in the optimal branch without expanding them and the time spent expanding nodes in a sub-optimal branch.

Lemma 3 *If N is the (random) number of samples $\tilde{v}_i$ from random variable $V \in [0, \beta]$ we must take until its empirical mean $\hat{V}_k \triangleq \sum_{i=1}^k \tilde{v}_i > \mathbf{E}V - \Delta$, then:*

$$\mathbf{E}[N] \leq 1 + \beta^2 \Delta^{-2} \tag{5}$$

$$\mathbf{P}[N > n] \leq \exp\left(-2\beta^{-2} n^2 \Delta^2\right). \tag{6}$$

Proof The first inequality follows from the Hoeffding inequality and an integral bound on the resulting sum, while the second inequality is proven directly via a Hoeffding bound. ∎

By setting Δ to be the difference between the optimal and second optimal branch, we can use the above lemma to bound the number of times N the leaf nodes in the optimal branch will be sampled without being expanded. The converse problem is bounding the number of times that a suboptimal branch will be expanded.

Algorithm 4 Stochastic branch and bound 2

for $\omega \in \mathcal{L}_n$ **do**
  $\hat{V}_U^\omega = \frac{1}{\sum_{\omega' \in \mathcal{C}(\omega)} m_{\omega'}} \sum_{\omega' \in \mathcal{C}(\omega)} \sum_{i=1}^{m'_\omega} \tilde{v}_i^{\omega'}$
end for
Use (3) to obtain $\hat{V}_U$ for all nodes.
Set ω_0 to root.
for $d = 1, \ldots$ **do**
  $a_d^* = \arg\max_a \sum_{\omega \in \Omega_d} \omega_{d-1}(j|a) \hat{V}_U(\omega)$
  $\omega_d \sim \omega_{d-1}(j|a_d^*)$
  if $\omega_d \in \mathcal{L}_n$ **then**
    $\mathcal{L}_{n+1} = \mathcal{C}(\omega_d) \cup \mathcal{L}_n \backslash \omega_d$
    Break
  end if
end for

Lemma 4 *If b is a branch with $V^b = V^* - \Delta$, then it will be expanded at least to depth $k_0 = \log_\gamma \Delta/\beta$. Subsequently,*

$$\mathbf{P}(K > k) < \mathcal{O}\left(\exp\left\{-2\beta^{-2}\left[(k-k_0)\Delta^2\right]\right\}\right). \tag{7}$$

Proof In the worst case, the branch is degenerate and only one leaf has non-zero probability. We then apply a Hoeffding bound to obtain the desired result. ∎

2.4 Stochastic Branch and Bound 2

The degeneracy is the main problem of Alg. 3. Alg. 4 not only propagates upper bounds from multiple leaf nodes to the root, but also re-uses upper bound samples from inner nodes, in order to handle the degenerate case where only one path has non-zero probability. (Nevertheless, Lemma 3 applies without modification to Alg. 4). Because we are no longer operating on leaf nodes, we can take advantage of the upper bound samples collected along a given trajectory. However, if we use all of the upper bounds along a branch, then the early samples may bias our estimates a lot. For this reason, if a leaf is at depth k, we only average the upper bounds along the branch to depth $k/2$. The complexity of this approach is given by the following lemma:

Lemma 5 *If b is s.t. $V^b = V^* - \Delta$, it will be expanded to depth $k_0 > \log_\gamma \Delta/\beta$ and*

$$\mathbf{P}(K > k) \lessapprox \exp\left(-2(k-k_0)^2(1-\gamma^2)\right), \qquad k > k_0$$

Proof There is a degenerate case where only one sub-branch has non-zero probability. However we now re-use the samples that were obtained at previous expansions, thus allowing us to upper bound the bias by $\frac{\Delta(1-\gamma^{k+1})}{(k-k_0)(1-\gamma)}$. This allows to use a tighter Hoeffding bound and so obtain the desired outcome. ∎

This bound decreases faster with k. Furthermore, there is no dependence on Δ after the initial transitory period, which may however be very long. The gain is due to the fact that we are re-using the upper bounds previously obtained in inner nodes. Thus, this algorithm should be particularly suitable for stochastic problems.

2.5 Lower Bounds for Bayesian RL

We can reduce the branching factor ϕ, (which is $|\mathcal{A} \times \mathcal{S} \times \mathcal{R}|$ for a full search) by employing sparse sampling methods [Kearns et al., 1999] to $O\{|\mathcal{A}| \exp[1/(1-\gamma)]\}$. This was essentially the approach employed by [Wang et al., 2005]. However, our main focus here is to reduce the depth to which each branch is searched.

The main problem with the above algorithms is the fact that we must reach $k_0 = \lceil \log_\gamma \Delta \rceil$ to discard Δ-optimal branches. However, since the hyper-state ω_t arises from a Bayesian belief, we can use an additional smoothness property:

Lemma 6 *The Dirichlet parameter sequence ψ_t/n_t, with $n_t \triangleq \sum_{i=1}^K \psi_t^i$, is a c-Lipschitz martingale with $c_t = 1/2(n_t+1)$.*

Proof Simple calculations show that, no matter what is observed, $\mathbf{E}_{\xi_t}(\psi_{t+1}/n_{t+1}) = \psi_t/n_t$. Then, we bound the difference $|\psi_{t+k}/n_{t+k} - \psi_t/n_t|$ by two different bounds, which we equate to obtain c_t. ∎

Lemma 7 *If $\mu, \hat{\mu}$ are such that $\|\mathcal{T} - \hat{\mathcal{T}}\|_\infty \leq \varepsilon$ and $\|r - \hat{r}\|_\infty \leq \varepsilon$, for some $\varepsilon > 0$, then $\left\|V^\pi - \hat{V}^\pi\right\|_\infty \leq \frac{\varepsilon}{(1-\gamma)^2}$, for any policy π.*

Proof By subtracting the Bellman equations for $V, \hat{V}$ and taking the norm, we can repeatedly apply Cauchy-Schwarz and triangle inequalities to obtain the desired result. ∎

The above results help us obtain better lower bounds in two ways. First we note that initially $1/k$ converges faster than γ^k, for large γ, thus we should be able to expand less deeply. Later, n_t is large so we can sample even more sparely.

If we search to depth k, and the rewards are in $[0,1]$, then, naively, our error is bounded by $\sum_{n=k}^\infty \gamma^n = \gamma^k/(1-\gamma)$. However, the mean MDPs for $n > k$ are close to the mean MDP at k due to Lem. 6. This means that β can be significantly smaller than $1/(1-\gamma)$. In fact, the total error is bounded by $\sum_{n=k}^\infty \gamma^n (n-k)/n$. For undiscounted problems, our error is bounded by $T-k$ in the original case and by $T-k[1+\log(T/k)]$ when taking into account the smoothness.

3 Conclusions and related work

Much recent work on Bayesian RL focused on myopic estimates or full expansion of the belief tree up to a certain depth. Exceptions include [Poupart et al., 2006], which uses an analytical bound based on sampling a small set of beliefs and [Wang et al., 2005], which uses Kearn's sparse sampling algorithm [Kearns et al., 1999] to expand the tree. Both methods have complexity exponential in the horizon, something which we improve via the use of smoothness properties induced by the Bayesian updating.

There are also connections with work on POMDPs problems [Ross et al., 2008]. However this setting, though equivalent in an abstract sense, is not sufficiently close to the one we consider. Results on bandit problems, employing the same value function bounds used herein were reported in [Dimitrakakis, 2008], which experimentally compared algorithms operating on leaf nodes only.

Related results on the online sample complexity of Bayesian RL were developed by [Kolter and Ng, 2009], who employs a different upper bound to ours and [Asmuth et al., 2009], who employs MDP samples to plan in an augmented MDP space, similarly to

Auer et al. [2008] (who consider the set of plausible MDPs) and uses Bayesian concentration of measure results [Zhang, 2006] to prove mistake bounds on the online performance of the algorithm.

Interestingly, Alg. 4 resembles HOO [Bubeck et al., 2008] in the way that it traverses the tree, with two major differences. (a) The search is adapted to *stochastic* trees. (b) We use means of samples of upper bounds, rather than upper bounds on sample means. For these reasons, we are unable to simply restate the arguments in [Bubeck et al., 2008].

We presented complexity results and counting arguments for a number of tree search algorithms on trees where stochastic upper and lower bounds satisfying a smoothness property exist. These are the first results of this type and partially extend the results of [Norkin et al., 1998], which provided an asymptotic convergence proof, under similar smoothness conditions, for a stochastic branch and bound algorithm. In addition, we introduce a mechanism to utilise samples obtained at inner nodes when calculating mean upper bounds at leaf nodes. Finally, we relate our complexity results to those of [Kearns et al., 1999], for whose lower bound we provide a small improvement. We plan to address the online sample complexity of the proposed algorithms, as well as their practical performance, in future work.

Acknowledgements

This work was part of the ICIS project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. I would also like to thank the anonymous reviewers for their detailed reviews of earlier versions of this paper; Peter Auer, Peter Grünwald, Ronald Ortner and Remi Munos for extensive discussions; and finally Shimon Whiteson and Frans Groen for further comments and corrections.

A Proofs of the main results

Proposition 1 By definition, $V^*(\omega) \geq V^\pi(\omega)$ for all ω, for any policy π. The lower bound follows trivially, since

$$V^{\pi^*(\hat{\mu}_\omega)}(\omega) \triangleq \int V_\mu^{\pi^*(\hat{\mu}_\omega)}(s_\omega)\xi_\omega(\mu)\,d\mu. \tag{8}$$

The upper bound is derived as follows. First note that for any function f, $\max_x \int f(x,u)\,du \leq \int \max_x f(x,u)\,du$. Then, we remark that:

$$V^*(\omega) = \max_\pi \int V_\mu^\pi(s_\omega)\xi_\omega(\mu)\,d\mu \tag{9a}$$

$$\leq \int \max_\pi V_\mu^\pi(s_\omega)\xi_\omega(\mu)\,d\mu \tag{9b}$$

$$= \int V_\mu^{\pi^*(\mu)}(s_\omega)\xi_\omega(\mu)\,d\mu. \tag{9c}$$

∎

Lemma 1 For any b' with $V_L^{b'} < V_L^b$, we have: $V^{b'} \leq V_L^{b'} + \beta\gamma^k < V_L^b + \beta\gamma^k \leq V^b + \beta\gamma^k$. This holds for $b = \hat{b}^*$. Thus, in the worst case, the regret that we suffer if there exists some $b' : V^{b'} > V^{\hat{b}^*}$ is $\varepsilon = V^{b'} - V^{\hat{b}^*} < \beta\gamma^k$. To reach depth k in all branches we need $n = \sum_{t=1}^k \phi^k < \phi^{k+1}$ expansions. Thus, we require $k = \frac{\log(\varepsilon/\beta)}{\log\gamma}$ and $n \leq \phi^{1+\log_\gamma(\varepsilon/\beta)}$. ∎

Lemma 2 The total number of samples is km, the number of leaf nodes times the number of samples at each leaf node. The search is until depth

$$k = \left\lceil \log_\gamma \varepsilon/2\beta \right\rceil \leq 1 + \log_\gamma \varepsilon/2\beta \tag{10}$$

and the number of samples is

$$m = 2\log_\gamma(\varepsilon/2\beta)\log\phi. \tag{11}$$

The complexity follows trivially. Now we must prove that this bounds the expected regret with ε. Note that $\beta\gamma^k < \varepsilon/2$, so for all branches b:

$$\hat{V}_L^b - V^b < \varepsilon/2. \tag{12}$$

The expected regret can now be written as

$$\mathbf{E}R \leq \frac{\varepsilon}{2} + \mathbf{E}[R|\hat{V}_L^{\hat{b}^*} < \hat{V}_L^{b^*} + \varepsilon/4]\,\mathbf{P}(\hat{V}_L^{\hat{b}^*} < \hat{V}_L^{b^*} + \varepsilon/4) \tag{13}$$

$$+ \mathbf{E}[R|\hat{V}_L^{\hat{b}^*} \geq \hat{V}_L^{b^*} + \varepsilon/4]\,\mathbf{P}(\hat{V}_L^{\hat{b}^*} \geq \hat{V}_L^{b^*} + \varepsilon/4). \tag{14}$$

From the Hoeffding bound (21)

$$\mathbf{P}(\hat{V}_L - V_L > \varepsilon/4) < \exp\left(-\frac{1}{8}m\beta^{-2}\gamma^{-2k}\varepsilon^2\right)$$

and with a union bound the total error probability is bounded by $\phi^k \exp\left(-\frac{1}{8}m\beta^{-2}\gamma^{-2k}\varepsilon^2\right)$. If our estimates are within $\varepsilon/4$ then the sample regret is bounded by $\varepsilon/4$, while the other terms are trivially bounded by 1, to obtain

$$\mathbf{E}R \leq \frac{\varepsilon}{2} + \left\{\phi^k \exp\left(-\frac{1}{8}m\beta^{-2}\gamma^{-2k}\varepsilon^2\right) + \frac{\varepsilon}{4}.\right\} \tag{15}$$

Substituting m and k, we obtain the stated result. ∎

Lemma 3

$$\mathbf{E}[N] = \sum_{n=1}^{\infty} n \prod_{j=1}^{n-1} \mathbf{P}(\hat{V}(j) \geq V + \varepsilon)\,\mathbf{P}(\hat{V}(n) < V + \varepsilon) \tag{16}$$

$$\leq \sum_{n=1}^{\infty} n \exp\left(-2\beta^{-2}\varepsilon^2 \sum_{j=1}^{n-1} j\right) = \sum_{n=1}^{\infty} n \exp\left(-\beta^{-2}\varepsilon^2 n(n+1)\right) \tag{17}$$

Let us now set $\rho = \exp(-\beta^{-2}\varepsilon^2)$. Observe that $n\rho^{n(n+1)} < n\rho^{n^2}$, since $\rho < 1$. Then, note that $\int n\rho^{n^2}\,dn = O\left(\frac{\rho^{n^2}}{2\log\rho}\right)$. So we can bound the sum by

$$\sum_{n=1}^{\infty} n\rho^{n(n+1)} < 1 + \left[\frac{\rho^{n^2}}{2\log\rho}\right]_1^{\infty} 1 + \frac{\exp(-\beta^{-2}\varepsilon^2)}{2\beta^{-2}\varepsilon^2} < 1 + \left(\frac{\beta}{\varepsilon}\right)^2. \tag{18}$$

This proves the first inequality. For the second inequality, we have:

$$\mathbf{P}(N > n) = \mathbf{P}\left(\bigwedge_{k=1}^{n} \hat{V}(k) > V + \varepsilon\right) < \prod_{k=1}^{n} \exp\left(-2k\beta^{-2}\varepsilon^2\right) \tag{19}$$

$$= \exp\left(-\beta^{-2}\varepsilon^2 n(n+1)\right) < \exp\left(-n^2\beta^{-2}\varepsilon^2\right). \tag{20}$$

This completes the proof for the first case. The second case ise symmetric. ∎

Lemma 4 In order to stop expanding a sub-optimal branch b, at depth k, we must have $V_U^b(k) < V^*$, since in the worst case $V_U^*(k) = V^*$ for all k. Since $V^b = V^* - \Delta$, this only happens when k is greater than $k_0 \triangleq \left\lceil \log_\gamma \Delta/\beta \right\rceil$, which is the minimum depth we must expand to. Subsequently, we shall note that the probability of stopping is $\mathbf{P}(\hat{V}_U^b(k) > \Delta - \beta\gamma^k) < \exp(-2(\Delta - \beta\gamma^k)^2\beta^{-2})$. We can not do better due to the degenerate case where only one leaf node of the branch has non-zero probability.

The probability of not stopping at depth k is bounded by:

$$\begin{aligned}
\mathbf{P}(K > k) &\leq \prod_{j=k_0}^{k} \exp(-2(\Delta - \beta\gamma^j)^2\beta^{-2}) \leq \exp\left(-2\beta^{-2}\sum_{j=k_0}^{k}(\Delta - \beta\gamma^j)^2\right) \\
&\leq \exp\left[-\frac{2}{\beta^2}\left((k-k_0)\Delta^2 + \frac{\beta h}{1-\gamma^2}\right)\right], \\
h &= \beta(\gamma^{2k_0} - \gamma^{2(k+1)} - 2\Delta(\gamma^{k_0} - \gamma^{k+1})(1+\gamma) \\
&= \beta(\Delta^2 - \gamma^{2(k+1)} - 2\Delta(\Delta - \gamma^{k+1})(1+\gamma).
\end{aligned}$$

∎

Lemma 5 Similarly to the previous lemma, there is a degenerate case where only one sub-branch has non-zero probability. However this algorithm re-uses the samples that were obtained at previous expansions. When at depth k, we average the bounds from $\lceil k/2 \rceil$ to k. Since, in the worst case, we cannot stop until $k > k_0 = \left\lceil \log_\gamma \Delta/\beta \right\rceil$, we shall bound the probability that we stop at some depth $K > 2k_0$. Then the mean upper bound bias is at most:

$$h_k \triangleq \frac{1}{k-k_0}\sum_{n=k_0}^{k}\beta\gamma^n = \frac{\beta\gamma^{k_0}}{k-k_0}\frac{1-\gamma^{k+1}}{1-\gamma} < \frac{\Delta}{k-k_0}\frac{1-\gamma^{k+1}}{1-\gamma}.$$

The procedure continues only if the sampling error exceeds $\Delta - h_k$, so it suffices to bound $\mathbf{P}(\hat{X}_k > \bar{X}_k + \varepsilon)$, where $\hat{X}_k = \sum_{n=\lceil k/2 \rceil}^{k}\hat{V}_U(k)$ and $\bar{X}_k = V + h_k$ for $\varepsilon = \Delta(1 - \frac{1-\gamma^k}{(k-k_0)(1-\gamma)})$: $\mathbf{P}(\hat{X}_k > \bar{X}_k + \varepsilon) < \exp\left(-\frac{2(k-k_0)^2\varepsilon^2}{\sum_{n=k_0}^{k}(\beta\gamma^n)^2}\right)$. Since $\sum_{n=k_0}^{k}(\beta\gamma^n)^2 = \Delta^2\frac{1-\gamma^{2(k+1)}}{1-\gamma^2}$: $\mathbf{P}(\hat{X}_k > \bar{X}_k + \varepsilon) < \exp\left(-\frac{2(k-k_0)^2(1-\gamma^2)\varepsilon^2}{\Delta^2(1-\gamma^{2(k+1)})}\right)$. By setting $\varepsilon = \Delta - h_k$ we can bound this by

$$\exp\left(-\frac{2(k-k_0)^2(1-\gamma^2)}{(1-\gamma^{2(k+1)})}\cdot\left(1 - \frac{1-\gamma^{k+1}}{(k-k_0)(1-\gamma)}\right)^2\right).$$

For large k, this is approximately $\mathcal{O}(\exp(-k^2))$. ∎

Lemma 6 It is easy to see that $\mathbf{E}(\psi_{t+1}/n_{t+1}|\xi_t) = \psi_t/n_t$. This follows trivially when no observations are made since $\psi_{t+1} = \psi_t$. When one observation is made, $n_{t+1} = 1 + n_t$. Then $\mathbf{E}(\psi_{t+1}/n_{t+1}|\xi_t) = [\psi_t + \xi_t(\psi)]/n_{t+1} = [\psi_t + \psi_t/n_t]/(1+n_t) = \psi_t/n_t$. Thus, the matrix $\mathcal{T}_{\xi_t}$ is a martingale. We shall now prove the Lipschitz property. For all $k > 0$, $\psi_t > 0$:

$$\psi_t^i/(n_t+k) \leq \psi_{t+k}^i/n_{t+k} \leq (\psi^i + k)/n_{t+k}.$$

Note that $\left|\frac{\psi_{t+k}^i}{n_{t+k}} - \frac{\psi_t^i}{n_t}\right|$ is upper bounded by $\frac{k(n_t - \psi_t^i)}{n_t(n_t+k)}$ and $\frac{k\psi_t^i}{n_t(n_t+k)}$ and thus by $\frac{k\min\{\psi_t^i, n_t - \psi_k^i\}}{n_t(n_t+k)}$.

Equating the two terms, we obtain $\left|\frac{\psi_{t+k}^i}{n_{t+k}} - \frac{\psi_t^i}{n_t}\right| \leq \frac{k}{2(n_t+k)}$. ∎

Lemma 7 The transitions $P, \hat{P}$ induced by any policy obey $\|P - \hat{P}\|_\infty < \varepsilon$. By repeated use of Cauchy-Schwarz and triangle inequalities:

$$\begin{aligned}
\|V - \hat{V}\|_\infty &= \left\|r - \hat{r} + \gamma\left(PV - \hat{P}\hat{V}\right)\right\|_\infty \\
&\leq \|r - \hat{r}\|_\infty + \gamma\left\|PV - \hat{P}\hat{V}\right\|_\infty \\
&\leq \varepsilon + \gamma\left\|PV - (P - \tilde{P})\hat{V}\right\|_\infty \\
&\leq \varepsilon + \gamma\left(\left\|P(V - \hat{V})\right\|_\infty + \left\|\tilde{P}\hat{V}\right\|_\infty\right) \\
&\leq \varepsilon + \gamma\left(\|P\|_\infty \cdot \|V - \hat{V}\|_\infty + \|\tilde{P}\|_\infty \cdot \|\hat{V}\|_\infty\right) \\
&\leq \varepsilon + \gamma\left(\left\|V - \hat{V}\right\|_\infty + \varepsilon \cdot \frac{1}{1-\gamma}\right)
\end{aligned}$$

where $\tilde{P} = P - \hat{P}$, for which of course holds $\|\tilde{P}\|_\infty < \varepsilon$. Solving gives us the required result. ∎

B Hoeffding bounds for weighted averages

Hoeffding bounds can also be derived for weighted averages. Let us first recall the standard Hoeffding inequality:

Lemma 8 (Hoeffding inequality) *If $\hat{x}_n \triangleq \frac{1}{n}\sum_{i=1}^n x_i$, with $x_i \in [b_i, b_i + h_i]$ drawn from some arbitrary distribution f_i and $\overline{x_n} \triangleq \frac{1}{n}\sum_i \mathbf{E}[x_i]$, then, for all $\varepsilon \geq 0$:*

$$\mathbf{P}\left(\hat{x}_n \geq \overline{x_n} + \varepsilon\right) \leq \exp\left(-\frac{2n^2\varepsilon^2}{\sum_{i=1}^n h_i^2}\right). \tag{21}$$

We have a weighted sum, $\hat{x}'_n \triangleq \sum_{i=1}^n w_i x'_i$, $\sum_{i=1}^n w_i = 1$. If we set $v_i \triangleq n w_i$, then we can write the above as $\frac{1}{n}\sum_{i=1}^n v_i x'_i$. So, if we let $x_i = v_i x'_i$ and assume that $x'_i \in [b, b+h]$, then $x_i \in [v_i b + v_i(b+h)]$. Substituting into (21) results in

$$\mathbf{P}\left(\hat{x}_n \geq \overline{x} + \varepsilon\right) \leq \exp\left(-\frac{2\varepsilon^2}{h^2\sum_{i=1}^n w_i^2}\right). \tag{22}$$

Furthermore, note that

$$\mathbf{P}\left(\hat{x}_n \geq \overline{x} + \varepsilon\right) < \exp\left(-\frac{2\varepsilon^2}{h^2}\right), \tag{23}$$

since $w_i^2 \leq w_i$ for all i, as $w_i \in [0,1]$. Thus $\sum_i w_i^2 \leq \sum_i w_i = 1$. Note that $\sum_i w_i^2 = 1$ iff $w_j = 1$ for some j.

References

J. Asmuth, L. Li, M. L. Littman, A. Nouri, and D. Wingate. A Bayesian sampling approach to exploration in reinforcement learning. In *UAI 2009*, 2009.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. In *Proceedings of NIPS 2008*, 2008.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. Online optimization in X-armed bandits. In *NIPS*, pages 201–208, 2008.

Christos Dimitrakakis. Tree exploration for Bayesian RL exploration. In *Computational Intelligence for Modelling, Control and Automation, International Conference on*, pages 1029–1034, Los Alamitos, CA, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3514-2. doi: http://doi.ieeecomputersociety.org/10.1109/CIMCA.2008.32.

Michael O'Gordon Duff. *Optimal Learning Computational Procedures for Bayes-adaptive Markov Decision Processes*. PhD thesis, University of Massachusetts at Amherst, 2002.

Michael J. Kearns, Yishay Mansour, and Andrew Y. Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. In Thomas Dean, editor, *IJCAI*, pages 1324–1231. Morgan Kaufmann, 1999. ISBN 1-55860-613-0.

J. Zico Kolter and Andrew Y. Ng. Near-Bayesian exploration in polynomial time. In *ICML 2009*, 2009.

Vladimir I. Norkin, Georg Ch. Pflug, and Andrzej Ruszczyski. A branch and bound method for stochastic global optimization. *Mathematical Programming*, 83(1):425–450, January 1998. doi: 10.1007/BF02680569.

P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete Bayesian reinforcement learning. In *ICML 2006*, pages 697–704. ACM Press New York, NY, USA, 2006.

Marting L. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New Jersey, US, 1994,2005.

Stéphane Ross, Joelle Pineau, Sébastien Paquet, and Brahim Chaib-draa. Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Resesarch*, 32:663–704, July 2008.

Tao Wang, Daniel Lizotte, Michael Bowling, and Dale Schuurmans. Bayesian sparse sampling for on-line reward optimization. In *ICML '05*, pages 956–963, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: http://doi.acm.org/10.1145/1102351.1102472.

Tong Zhang. From ε-entropy to KL-entropy: Analysis of minimum information complexity density estimation. *Annals of Statistics*, 34(5):2180–2210, 2006.