Bayesian Sparse Sampling for On-line Reward Optimization

Tao Wang TRYSI@CS.UALBERTA.CA
Daniel Lizotte DLIZOTTE@CS.UALBERTA.CA
Michael Bowling BOWLING@CS.UALBERTA.CA
Dale Schuurmans DALE@CS.UALBERTA.CA
Department of Computing Science, University of Alberta, Edmonton AB T6G 2E8, Canada

Abstract

We present an efficient "sparse sampling" technique for approximating Bayes optimal decision making in reinforcement learning, addressing the well known exploration versus exploitation tradeoff. Our approach combines sparse sampling with Bayesian exploration to achieve improved decision making while controlling computational cost. The idea is to grow a sparse lookahead tree, intelligently, by exploiting information in a Bayesian posterior—rather than enumerate action branches (standard sparse sampling) or compensate myopically (value of perfect information). The outcome is a flexible, practical technique for improving action selection in simple reinforcement learning scenarios.

1. Introduction

Even though reinforcement learning is a rapidly maturing subject, there remains little convergence on the fundamental question of action selection; that is, how to choose actions during learning. Beyond the standard ϵ-greedy and Boltzmann selection strategies, few techniques have been adopted beyond the papers that originally proposed them. However, there remains a persistent belief that more sophisticated selection strategies can yield improved results (Kaelbling, 1994; Dearden et al., 1999; Strens, 2000; Wyatt, 2001). Possible reasons for the limited use of sophisticated exploration approaches might be the complexity of implementing some proposed methods, or the presumption that the degree of improvement might not always be dramatic. Therefore, beyond the quality of action selection results, it is also important to consider the complexity and computational cost of choosing the actions. In this paper we adopt a Bayesian approach to reinforcement learning and attempt to derive relatively straightforward action selection strategies for a class of problems.

The Bayesian approach to reinforcement learning still appears to be under-researched given the prominent role it has played in other areas of machine learning (Jordan, 1999; Neal, 1996). Flexible Bayesian tools, such as Gaussian process regression (Williams, 1999; Neal, 1996), have had a significant impact on other areas of machine learning research but have only just recently been introduced to reinforcement learning (Engel et al., 2003). Nevertheless, Bayesian approaches seem ideally suited to reinforcement learning as they offer an explicit representation of uncertainty—essential for reasoning about the exploration versus exploitation tradeoff. In fact, Bayesian approaches offer the prospect of *optimal* action selection. Bayesian decision theory solves the exploration versus exploitation tradeoff directly (but implicitly) by stipulating that the optimal action is one which, over the entire time horizon being considered, maximizes the total expected reward (averaged over possible world models). Therefore, any gain in reducing uncertainty is not valued for its own sake, but measured instead in terms of the potential gain in future reward it offers. In this way, explicit reasoning about exploration versus exploitation is subsumed by direct reasoning about rewards obtained over the long term.

Despite the apparent elegance and conceptual simplicity of the Bayesian approach, there remain serious barriers to its application. The most serious drawback is the computational challenge posed by optimal Bayesian decision making, which is known to be intractable in all but trivial decision making contexts (Mundhenk et al., 2000; Lusena et al., 2001). This means that with a Bayesian approach one is forced to consider heuristic approximations. In response, a small body of research has developed on online approximations of optimal Bayesian action selection (Dearden et al., 1999; Duff, 2002; Strens, 2000). Although the number of proposals remains relatively small and no widely adopted approximation strategy has emerged, the potential power of Bayesian modeling makes this approach

Appearing in *Proceedings of the 22^{nd} International Conference on Machine Learning*, Bonn, Germany, 2005. Copyright 2005 by the author(s)/owner(s).

worth investigating.

In this paper we attempt to further develop practical approximations of optimal Bayesian action selection for reinforcement learning. Specifically, we combine the Bayesian approach (Dearden et al., 1999) with the (non-Bayesian) *sparse sampling* technique of (Kearns et al., 2001). Our idea is to exploit information in the posterior to make intelligent action selection decisions during lookahead simulation, rather than simply enumerating actions (Kearns et al., 2001). This approach yields improved action selection quality while controlling computational cost. We also show that our technique improves the myopic value of perfect information approximation strategy of (Dearden et al., 1999) by allowing deeper lookahead. Throughout, we attempt to propose simple strategies that can be easily implemented in a Bayesian framework.

2. Background

The standard reinforcement learning problem involves learning to behave optimally in an unknown Markov decision process.

Markov decision processes A *Markov decision process* (MDP) is defined by a set of actions A; a set of states S; a transition model $p(s_{t+1}|s_t a_t)$, specifying the conditional probability of a successor state s_{t+1} given that the process is in state s_t and action a_t is executed; and a reward model $p(r_t|s_t a_t)$, specifying the conditional distribution over rewards r_t given that action a_t is taken in state s_t. The goal is to choose actions to maximize the reward obtained over the long run, where this can be defined in a few different ways: (1) maximizing the episodic (or finite horizon) reward $r_0 + r_1 + \cdots + r_T$ obtained over a finite episode $t = 0, ..., T$; (2) maximizing infinite horizon (discounted) reward $r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots$ obtained over an infinite run of the system, $0 < \gamma < 1$; or (3) maximizing the asymptotic rate of return. We will focus on *finite horizon* problems in this paper, although all of our techniques easily extend to the infinite horizon discounted case.

Under general conditions, for a fully specified MDP there is always a deterministic policy $\pi^* : S \to A$ that gives the optimal action in each state (Bertsekas, 1995). Such a policy can be conveniently characterized by the *action value function* (or Q-function), $Q(s, a)$, which is defined as the supremum of the expected (discounted) reward obtainable by first taking action a in state s and then following an optimal action selection strategy thereafter. The Q-function satisfies the well known Bellman equation (Bertsekas, 1995) $Q(s_t, a_t) = \mathrm{E}\left[r_t|s_t a_t\right] + \gamma \mathrm{E}\left[\max_{a \in A} Q(s_{t+1}, a)|\, s_t a_t\right]$ where we assume the maximum always exists in A. (In the finite horizon case we also assume $\gamma = 1$.) If the Q-function is known for a particular domain, then the optimal action selection strategy can be recovered by $\pi^*(s_t) = \arg\max_{a \in A} Q(s_t, a)$. Classical algorithms for computing π^*, or so called "planning" algorithms, can be based on value iteration, policy iteration, or linear programming (Bertsekas, 1995).

Reinforcement learning Of course, we are interested in the problem of *learning* to behave optimally in an initially *unknown* MDP. Let $p(s_{t+1}|s_t a_t \theta)$ and $p(r_t|s_t a_t \mu)$ denote the transition and reward models, where θ and μ denote the unknown parameters defining these models respectively. Thus, we consider a learning scenario where the transition and reward parameters, θ and μ, are not precisely known, but instead assumed only to belong to a general set, $\theta \in \Theta$ and $\mu \in M$. The three standard approaches to reinforcement learning are value based, policy based and model based learning. Roughly speaking, in the *value based* approach one attempts to estimate the optimal Q-function (or state value function) directly (Sutton & Barto, 1998; Watkins, 1989), from which a greedy policy is recovered. The *policy based* approach tries to estimate a good policy directly (Ng & Jordan, 2000; Strens & Moore, 2002). In *model based* reinforcement learning, one first attempts to estimate the transition and reward models, and then determines a policy by solving the planning problem in the learned model.

Bayesian reinforcement learning The literature on *Bayesian* reinforcement learning by comparison is relatively small. Nevertheless, Bayesian approaches have been considered from the outset (Martin, 1967; Bellman, 1961) and interest has re-emerged in this approach (Engel et al., 2003; Dearden et al., 1999; Strens, 2000; Wyatt, 2001). Much of the research on Bayesian reinforcement learning is *model based*: A *prior* distribution is defined over transition and reward models, $P(\theta, \mu|s_0)$, which is usually assumed to be factored $P(\theta, \mu|s_0) = P(\theta|s_0)P(\mu|s_0) = p_0^\theta(\theta)p_0^\mu(\mu)$. Given experience $s_0 a_0 r_0 s_1 ... s_t a_t r_t s_{t+1}$ one determines the *posterior* distribution $P(\theta, \mu|s_0 a_0 r_0 s_1 ... s_t a_t r_t s_{t+1}) = p_t^\theta(\theta)p_t^\mu(\mu)$; thus learning consists essentially of updating the posterior. Given this model based approach, one of the main difficulties with the Bayesian method (or any model based method) is that *planning* is required for action selection.

Except for the heavy reliance on planning, Bayesian approaches seem ideally suited to reinforcement learning problems. Bayesian modeling is not only a flexible tool that allows prior knowledge about the transition and reward models to be explicitly stated, it also readily allows generalization across actions, states and rewards, through a principled mechanism. Some of the best developed Bayesian modeling tools, such as Gaussian processes (Williams, 1999), are suited specifically for continuous state and action spaces, where classical reinforcement learning methods are not always conveniently applicable. Bayesian approaches also naturally provide an explicit representation of *uncertainty* in the posterior distribution, which is eminently useful for exploration/exploitation decision making.

3. Action selection

We focus on the problem of *on-line* reinforcement learning where action selection decisions involve a tradeoff between exploration and exploitation.[1] Intuitively, achieving a large reward over the long run would seem to involve, early on, taking exploratory actions to allow a good model (or value function or policy) to be estimated, and then later exploiting this model (or value function or policy) to consistently obtain high reward. Of course, the two phases are not necessarily distinct and it is not always advantageous to think of an action as either purely exploratory or exploitive. Classically, action selection in reinforcement learning has not been thought of in Bayesian terms but instead tackled intuitively. The most common action selection strategies have been:

ϵ-greedy With probability $1-\epsilon$, choose the current best estimate $a^* = \arg\max_a \hat{Q}(s,a)$. Otherwise choose a (uniform) random action $a \in A$.[2]

Boltzmann Sample a random action according to $P(a|s) = \exp(\hat{Q}(s,a)/\tau)/Z$ where τ is a temperature parameter and Z is a normalization constant.

Interval estimation (Kaelbling, 1994) Choose an action according to $a = \arg\max_a[\hat{Q}(s,a) + U(s,a)]$ where $U(s,a)$ is a $(1-\delta)$ upper confidence interval on the point estimate $\hat{Q}(s,a)$. This approach has been extended by (Wiering, 1999) to general MDPs.

These non-Bayesian action selection strategies are all *myopic*, in that they do not explicitly consider the effects that actions have on future value estimates. Instead, they use uncertainty as a proxy for lookahead. The basic intuition is that the greater the uncertainty in an action's value, the greater the chance that it might actually prove to be optimal, and therefore we should give a greater bonus to exploring this action. One difficulty with this type of intuitive reasoning, however, is that it is hard to quantify. The resulting selection procedures are heuristic, sometimes difficult to justify, and do not perform well in all circumstances.

Bayesian action selection A conceptually more elegant solution to the action selection problem is offered by Bayesian decision theory. A Bayesian approach to learning optimally in a Markov decision process is essentially equivalent to solving a partially observable Markov decision process (POMDP). More precisely, it is equivalent to solving for an optimal action selection strategy in a meta-level Markov decision process defined by the belief states of the problem. This meta-level problem is sometimes referred to as a belief state MDP or a Bayes-adaptive MDP (Duff, 2002). In the meta-level problem, each state b_t is given by a current base-level state s_t and a posterior distribution over the base-level transition and reward models, θ and μ, respectively. That is, $b_t = \langle p_t^\theta p_t^\mu s_t \rangle$ where $p_t^\theta = P(\theta|s_0 a_0 ... s_{t-1} a_{t-1} s_t)$ and $p_t^\mu = P(\mu|s_0 a_0 r_0 ... s_{t-1} a_{t-1} r_{t-1})$. The meta-level reward model is then simply given by the expectation

$$\begin{aligned} P(r_t|p_t^\theta p_t^\mu s_t a_t) &= \int_\mu P(r_t|s_t a_t \mu) p_t^\mu(\mu) d\mu \\ &= P(r_t|s_0 ... s_t a_t) \qquad (1) \end{aligned}$$

and the meta-level transition model is given by

$$\begin{aligned} & P(p_{t+1}^\theta p_{t+1}^\mu s_{t+1}|p_t^\theta p_t^\mu s_t a_t) \\ &= 1_{[p_{t+1}^\theta = P(\theta|s_0 a_0 ... s_{t+1})]} \left[\int_\theta P(s_{t+1}|s_t a_t \theta) p_t^\theta(\theta) d\theta \right] \\ & \quad \int_{r_t} 1_{[p_{t+1}^\mu = P(\mu|s_0 ... s_t a_t r_t)]} \int_\mu P(r_t|s_t a_t \mu) p_t^\mu(\mu) d\mu dr_t \\ &= P(r_t s_{t+1}|s_0 ... s_t a_t) \qquad (2) \end{aligned}$$

where r_t is such that $p_{t+1}^\mu = P(\mu|s_0 ... s_t a_t r_t)$. In fact, the meta-level states $b_t = \langle p_t^\theta p_t^\mu s_t \rangle$ are equivalent to histories $b_t \equiv s_0 a_0 r_0 ... s_{t-1} a_{t-1} r_{t-1} s_t$, and the state transition probability is simply the probability of a particular history extension r_t, s_{t+1} given the current history $s_0 a_0 r_0 ... s_{t-1} a_{t-1} r_{t-1} s_t$ and action a_t.

An optimal action selection strategy for reinforcement learning in this setting is given by a policy that obtains maximum expected reward in the meta-level (belief state) Markov decision process. However, even though this observation nicely characterizes optimal action selection for Bayesian reinforcement learning, there is no efficient way to compute or even approximate this strategy in a guaranteed way (Mundhenk et al., 2000; Lusena et al., 2001). One obvious difficulty is that there are far more meta-level belief states (i.e. base-level histories) than original base-level states. In all but trivial circumstances, there is no hope of

[1] There is an important distinction between *on-line* and and *batch* reinforcement learning. Batch learning distinguishes an initial training phase from a subsequent testing phase. During training, the learning algorithm has no responsibility for obtaining reward and focuses solely on gaining information. During subsequent testing, a non-adaptive policy is executed. Although batch learning is a slightly unnatural model for reinforcement learning, important theoretical results have been obtained which show that near optimal policies can be learned in time polynomial in the size of the state and action spaces (Kearns & Singh, 1998; Brafman & Tennenholtz, 2001). Curiously, these efficient "exploration" algorithms behave by putting artificially high rewards on unknown state-action pairs and then executing exploitive actions.

[2] For infinite action spaces we assume the range of possible actions is bounded.

exactly following an optimal action selection strategy.[3]

For the most part, work on approximating Bayes optimal action selection has followed two approaches: pre-compilation and on-line computation. In the *pre-compilation* approach, one attempts to derive a compact approximation to the optimal value function (Bertsekas & Tsitsiklis, 1996; Boyan & Moore, 1996) or the optimal policy (Ng & Jordan, 2000; Strens & Moore, 2002) for the meta-level belief state MDP. In fact, any approximation strategy for general POMDPs is applicable in this case, although a few interesting specializations have been attempted for belief state MDPs (Duff, 2002). One potential shortcoming of the pre-compilation approach is that once an action selection strategy has been fixed, it is hard to adapt it to the belief states that are actually encountered during learning. Moreover, this approach necessarily cannot obtain a uniformly accurate approximation over the entire state and action space, and there is no guarantee that the approximation holds over the belief states that are encountered in a particular learning episode. Pre-compilation might nevertheless be the only viable approach if actions need to be selected in real time (Ng & Jordan, 2000).

In contrast, the *on-line* approach to approximating Bayes optimal action selection attends only to the particular belief states encountered during learning, which would seem to relax the burden on the approximation strategy and offer the prospect of higher quality decisions. The drawback is that instead of extensive pre-compilation (allowing fast on-line decisions), these techniques can require nontrivial computation for each action selection decision.

The simplest on-line strategies are pure myopic strategies. In fact, Bayesian variants of the ϵ-greedy and Boltzmann action selection strategies are easy to develop. In this case one uses the *expected* Q-function $\bar{Q}_t(s,a) = E_{\theta\sim p_t^\theta,\mu\sim p_t^\mu}[Q_{\theta\mu}(s,a)]$ defined by the current belief state. Since Bayesian approaches are generally *model based*, in that the belief state keeps a distribution over transition and reward models, the mean Q-value function has to be computed by *planning* in the underlying mean Markov decision process defined by the belief state distributions p_t^θ and p_t^μ (Dearden et al., 1999).[4] The fact that Bayesian on-line action selection strategies require (even limited) replanning for every belief state they encounter is probably the single greatest barrier to their routine use. Nevertheless, replanning is still viable in a range of interesting scenarios, which we will exploit below. For example, planning is trivial in bandit problems, and remains feasible in many episodic problems. Dearden et al. (1999) also show how importance sampling and prioritized sweeping can reduce the cost of replanning to just a few sampled models while maintaining reasonable estimates of $\bar{Q}_t(s,a)$.

One of the most interesting myopic action selection strategies in the Bayesian setting is in fact one of the first action selection strategies to have ever been proposed (Thompson, 1933; Berry & Fristedt, 1985).

Thompson sampling Given a current belief state $b_t = \langle p_t^\theta p_t^\mu s_t \rangle$, sample a transition and reward model, θ and μ, from the belief state distributions p_t^θ and p_t^μ, solve for the optimal Q-function $Q_{\theta\mu}(s,a)$ for this model, then select the optimal action $a_t = \arg\max_{a\in A} Q_{\theta\mu}(s_t,a)$.

This technique was originally proposed by (Thompson, 1933) for bandit problems, and has recently reemerged in the reinforcement learning literature (Strens, 2000). Thompson sampling selects actions according to the *probability* that they are optimal in models drawn randomly from the current belief state. Although old, this remains an elegant and effective action selection strategy that often outperforms modern proposals (Berry & Fristedt, 1985). Thompson sampling is not Bayes optimal however, as it is still myopic. In our experiments we find that it tends to over-explore (which is obviously true at the horizon).

A more recent action selection strategy of significance is that of (Dearden et al., 1999), which attempts to take the effects of exploration explicitly into account (see also (Wyatt, 2001)). This approach is based on considering the value that is gained by improving a Q-value estimate.

Value of perfect information Given the distribution over action value functions $Q_{\theta\mu}(s_t,a)$, defined by the current belief state, $\theta \sim p_t^\theta$, $\mu \sim p_t^\mu$, for each action $a \in A$, consider the value of learning the exact value $Q^*(s_t,a)$ under the true model. Let a_1 and a_2 be the actions with the largest and second largest expected Q-values respectively. The *gain* in value of learning $Q^*(s_t,a)$ for an action a is given by $Gain(Q^*(s_t,a_1)) = \big(\bar{Q}(s_t,a_2) - Q^*(s_t,a_1)\big)_+$ if $a = a_1$, and $Gain(Q^*(s_t,a)) = \big(Q^*(s_t,a) - \bar{Q}(s_t,a_1)\big)_+$ otherwise, where the mean Q-values are taken with respect to $\theta \sim p_t^\theta$, $\mu \sim p_t^\mu$. (That is, value is gained only if a new action becomes the best, but not otherwise.) The value of learning the exact Q-value of an action in the current belief state is then simply given by the expected gain $VPI_t(a) = E_{\theta\mu} Gain(Q_{\theta\mu}(s_t,a))$, which provides an upper bound on the myopic value of information of executing

[3] Perhaps the only well known exception to this is the result of (Gittins, 1989) which shows that in the special case where there are finitely many actions, each with their own independent (finite) state spaces (i.e., bandit problems), then optimal action decisions can be made in polynomial time to maximize the expected sum of infinite horizon discounted rewards. However, the restrictiveness of the independence assumption has prevented this approach from being widely applied in reinforcement learning problems. Beyond (Salganicoff & Ungar, 1995; Duff, 2002) very few successes have been reported in this direction.

[4] Note however that planning in a base-level MDP is much easier than planning in the meta-level MDP.

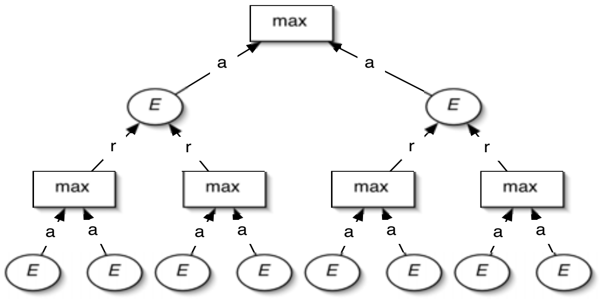

Figure 1. Illustration of a lookahead tree, showing decision (max) nodes and outcome (expectation) nodes. Once built, optimal value and action estimates are backed up to the root.

action a. Finally, one chooses the action that maximizes $\bar{Q}_t(s_t, a) + VPI_t(a)$.

Although these myopic strategies are interesting, a shortcoming of all such strategies is that they cannot explicitly account for the effects that actions have on future belief states, and therefore can only supply proxy summaries for the future rewards that might indirectly accrue as the result of a current action.

```
GrowSparseTree (node, branchfactor, horizon)

        If node.depth = horizon; return

        If node.type = “decision”
           For each $a \in A$
                child = (“outcome”, depth, node.belstate, $a$)
                GrowSparseTree (child, branchfactor, horizon)

        If node.type = “outcome”
           For $i = 1$...branchfactor
                [rew,obs] = sample(node.belstate, node.act)
                post = posterior(node.belstate, obs)
                child = (“decision”, depth+1, post, [rew,obs])
                GrowSparseTree (child, branchfactor, horizon)

EvaluateSubTree (node, horizon)

        If node.children = empty
           immed = MaxExpectedValue(node.belstate)
           return immed * (horizon - node.depth)

        If node.type = “decision”
           return max(EvaluateSubTree(node.children))

        If node.type = “outcome”
           values = EvaluateSubTree(node.children)
           return avg(node.rewards + values)
```

Figure 2. Sketch of sparse sampling algorithm. Grows a balanced lookahead tree, enumerating actions at decision nodes and sampling at outcome nodes. Sufficient values of branchfactor and horizon yield approximation guarantees.

4. Bayesian sparse sampling

The gap between Bayes optimal and myopic action selection strategies appears to be intuitively large. For example, for a finite horizon problem the Bayes optimal action is determined by dynamic programming: first solve for the optimal actions and values at the horizon, and then progressively back these up to earlier belief states (see Figure 1). Bayes optimal action selection essentially involves enumerating possible futures, averaging according to their realization probabilities, and choosing the best action. It is no surprise therefore that the only guaranteed way to approximate Bayes optimal action selection at a given belief state is to simulate the belief state MDP to the effective horizon.

The *sparse sampling* approach of (Kearns et al., 2001) replaces myopic estimates of the value of exploration with explicit lookahead to the effective horizon. This approach yields a general strategy for approximating optimal action selection in Markov decision processes, including the meta-level belief state MDPs we consider. A generic outline of the sparse sampling algorithm for finite horizon problems is given in Figure 2.

Note that sparse sampling requires a generative model, but this is conveniently exactly what a model based Bayesian approach provides, as shown in Equations (1) and (2). In this approach, lookahead is performed only by simulation in the meta-level belief state MDP which is maintained internally, not by actually taking actions in the world. That is, sparse sampling is an action selection strategy where, upon entering a belief state, extensive computation is exploited to determine an action that would yield near optimal reward over the long run (i.e. to the horizon) in the meta-level belief state MDP. Once chosen, the action is executed, and a new belief state is entered. To the extent that the Bayesian posterior concentrates on the true underlying model, this next belief state would have been influential in the previous computation.

Ignoring the obviously massive computation it requires to select each action, sparse sampling has some advantages. First, as (Kearns et al., 2001) show, it is guaranteed to produce a near optimal action for *any* belief state encountered, not just a restricted class of belief states. Second, sparse sampling can be easily applied to *infinite* state spaces. Of course, the theoretical procedure is too expensive to be applied in any real problem. But as Figure 2 shows, this procedure can be parameterized so that the computational cost can be controlled, by making the outcome branching factor and lookahead depth inputs to the procedure. Doing so requires us to forgo any theoretical guarantees of near optimality, but of course, one should not be surprised, since guaranteed approximation in this case is still provably intractable (Mundhenk et al., 2000; Lusena et al., 2001).

Even though sparse sampling can be parameterized to render a controllable lookahead strategy, it is still not a so-

```
GrowSparseBayesianTree (node, budget, $\rho$, horizon)
    While # nodes $<$ budget
        branchnode = BayesDescend(root, $\rho$)
        If branchnode.type = "decision"
            Add outcome then leaf node below branchnode
        If branchnode.type = "outcome"
            Add leaf decision node below branchnode
    return EvaluateTree(root)

BayesDescend (node, $\rho$)
    If node.type = "decision"
      $a$ = ThompsonSample(node.belstate)
      If $a \notin$ node.children     % new branch
            return [node,$a$]
      Else     % follow
            return BayesDescendTree(node.child($a$))
    If node.type = "outcome"
      If possible to branch, with probability $\rho$ % new branch
            return node
      Else     % follow
            [rew,obs] = sample(node.belstate,node.act)
            return BayesDescend(node.child([rew,obs]))
```

Figure 3. A Bayesian sparse sampling algorithm that adaptively grows a lookahead tree.

phisticated action value estimator, and can be easily improved by addressing some shortcomings. Given a belief state, our goal is to use lookahead search to estimate the long term value of possible actions. This situation is not unlike game tree search where one wants to expand the lookahead tree (here an expecti-max tree) intelligently so that search effort is not wasted and important branches are explored. The first idea we pursue is not to build a fully balanced lookahead tree, but instead attempt to grow the tree adaptively. The intuition is that one need only investigate actions in detail that are potentially optimal, and not waste computational resources on proving that unpromising actions are, indeed, suboptimal. That is, uniformly accurate estimates are not required at every decision node in the lookahead tree. Our second idea is to use an effective myopic action selection strategy—specifically Thompson sampling—to preferentially expand the tree below actions that at least appear to be locally promising. Finally, to reduce the variance of the estimates at outcome nodes, we also exploit the fact that unbiased reward expectations, locally, can be obtained by sampling them from the mean model, rather than first sampling a model and then sampling rewards from a random model. These ideas lead to the algorithm shown in Figure 3.

Once grown, the sparse lookahead tree must be evaluated to choose an action at the root. There are a few subtleties in doing so effectively. Clearly, values are backed up from the leaves; averaging at outcome nodes, and maximizing at decision nodes, as shown in Figure 1. However, when evaluating leaf nodes (which are always decision nodes in this approach) it is important to account for differing depths. Therefore, at each leaf, the mean posterior reward for each action is first multiplied by the number of decisions remaining to the horizon, thus correcting the leaf values to the same absolute depth. Another important issue is to consider *every* action at each decision node, even if some were not sampled during the tree growing phase.[5] That is, actions that have not been explored at a decision node are still evaluated by multiplying their posterior mean reward by the number of decisions remaining to the horizon.

Note that in this overall approach, myopic strategies are only used to decide where to look ahead in the simulation, not make any real action selection decisions. Real decisions are left to the full lookahead search. The procedure exploits the fact that there is a lot of latitude, during lookahead, to make heuristic action choices at the internal decision nodes (i.e. max nodes). In fact, the Bayesian sparse sampling procedure can be easily applied to *infinite* action spaces, whereas standard sparse sampling is inapplicable if actions cannot be enumerated. The Bayesian approach also has an advantage in that it allows one to approximate the maximum of a set of random variables without enumeration: Given a prior and sampled values, a posterior is determined over the distribution of the remaining variables. Thus, it is possible to stop whenever the expected posterior maximum value is no larger than the current maximum value, plus ϵ. In this way, it appears as though one can derive sparser sampling bounds in the Bayesian setting that are applicable to infinite action spaces.

5. Experimental results

To investigate the effectiveness of this sampling approach we conducted experiments on a number of simple domains where the planning problem is not difficult. These include bandit problems, but also episodic reinforcement learning problems. Our goal in this paper is not to focus on MDP planning, but rather to demonstrate action selection improvements, which is already a challenge even in simple reinforcement learning scenarios. (However, subject to coping with MDP planning challenges (Dearden et al., 1999) our approach can be applied to richer domains.)

We compare Bayesian sparse sampling (BayesSamp) with standard sparse sampling (SparseSamp) and standard myopic action selection strategies. These included Bayesian ϵ-greedy with $\epsilon = 0.1$ (eps-Greedy), Boltzmann exploration with temperature $\tau = 0.1$ (Boltzmann), and interval estimation (IE) with a range of two standard deviations, all using the expected Q-values given the current belief state.

[5] In the continuous action case we did not consider actions beyond those explicitly sampled, although additional local sampling could be used to ensure that a reasonable number of actions are considered at each decision node.

We also compared to Thompson sampling (Thompson) and the myopic value of perfect information (MVPI), using the same number of samples as a full lookahead tree of depth one to estimate the Q-value distributions. Finally, we compared to a lookahead strategy for action selection in MDPs proposed by (Péret & Garcia, 2004). This technique can be applied to Bayesian reinforcement learning simply by treating the problem as acting in a belief-state MDP. The Péret & Garcia strategy uses fixed length lookahead trajectories sampled from a current state; employing Boltzmann selection to choose the actions along each trajectory. Independent trajectories to a fixed horizon H are generated (set to $H = 5$ in our experiments) and the action with the best overall trajectory reward on average is selected at the root.

For each problem domain, we set a finite horizon time T and measure the rewards accumulated by each action selection strategy, averaged over 1000 to 10,000 repeats to estimate the expected total reward achieved as a function of horizon time. The lookahead strategies were set up to give a controlled comparison with each other. First, standard sparse sampling was run with a given lookahead depth (1 or 2) and fixed decision and outcome branching factors, yielding a balanced tree. Then the total number of nodes expanded in the balanced tree generated by sparse sampling was set as a maximum node budget for both Bayesian sparse sampling and Péret & Garcia sampling. Figures 4 to 7 show the results obtained.

The first domain is a simple bandit problem with five actions, each yielding $\{0, 1\}$ rewards according to independent Bernoulli distributions with payoff probabilities distributed according to a Beta prior. Here we see that lookahead strategies outperform the myopic strategies, even MVPI which uses comparable computation (Figure 4). Nevertheless this simple problem does not show much advantage for Bayesian over standard sparse sampling. Similar results were obtained for a related five action bandit problem where instead each action yields a reward according to an independent Gaussian distribution with means distributed according to a Gaussian prior (Figure 5).

More interesting results are obtained on complex domains where the action rewards are correlated. Here we conducted experiments in a scenario that involved *continuous* action spaces. Specifically, we considered problems where the reward distribution over actions is defined by a *Gaussian process* prior over the action space (Williams, 1999). This creates an interesting exploration problem where rewards are correlated between actions, and the acions themselves are not restricted to a trivial finite set.

Figures 6 and 7 show the results of the two continuous problems we considered. The first involved a 1-dimensional action space and the second a 2-dimensional action space. In each case, a Gaussian process prior over

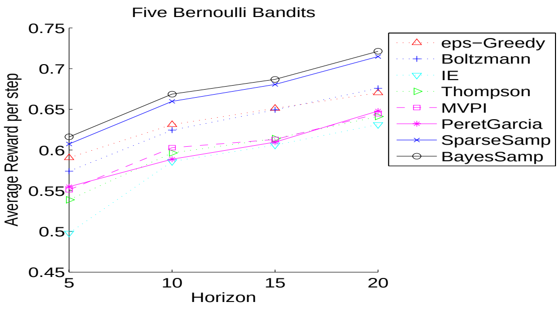

Figure 4. Bernoulli bandits

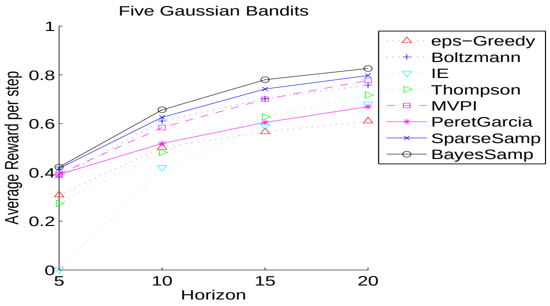

Figure 5. Gaussian bandits

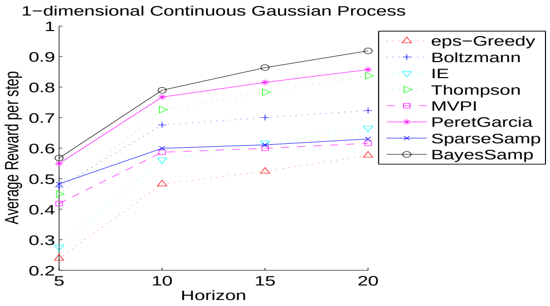

Figure 6. 1-dimensional continuous action Gaussian process

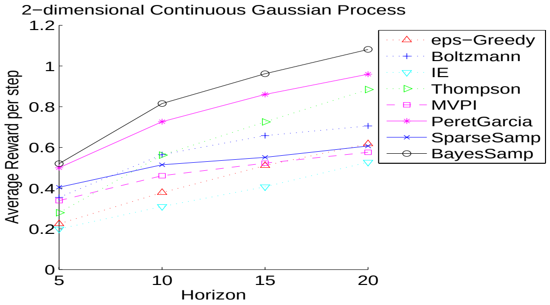

Figure 7. 2-dimensional continuous action Gaussian process

rewards was defined by an RBF kernel on actions, specifying the covariance between action rewards. (We used a Gaussian RBF kernel with width parameter 1. The noise standard deviation was set to $\sigma = 0.5$.) Technically standard sparse sampling and MPI are unable to cope with continuous action spaces, so we sampled actions for them to consider according to a uniform distribution. Figures 6 and 7 show a clear advantage for Bayesian sparse sampling over standard sparse sampling and the myopic approaches—using the same number of lookahead nodes as standard sparse sampling and Péret & Garcia sampling, and similar computation to MVPI. Surprisingly, Péret & Garcia sampling performed nearly as well in this case, even though it exhibits weaker performance in the bandit problems.

6. Conclusion

We have proposed a simple approach to improving action selection quality in model based Bayesian reinforcement learning. The main advantage is that the approach yields improved exploration/exploitation decision making whenever Bayesian posteriors can be conveniently calculated. The main drawback of our approach is shared by all model based Bayesian approaches to reinforcement learning: the need to repeatedly solve an MDP planning problem. Nevertheless, there are many interesting domains where this is not a significant barrier, and promising approaches have been developed for mitigating this expense (Dearden et al., 1999). Another area for future research is to compare on-line action selection strategies with pre-compilation approaches (Boyan & Moore, 1996) to verify that the perceived advantages of the on-line approach are real. It is also interesting to contemplate the prospect of hybrid action selection strategies that combine pre-compilation with on-line computation, perhaps by allowing a pre-compiled value function approximation to guide lookahead simulation without the need for on-line MDP planning.

Acknowledgments

Research supported by the Alberta Ingenuity Centre for Machine Learning, CRC, NSERC, MITACS and CFI.

References

Bellman, R. (1961). *Adaptive control processes*. Princeton.

Berry, D., & Fristedt, B. (1985). *Bandit problems*. Chapman Hall.

Bertsekas, D. (1995). *Dynamic programming and optimal control*, vol. 2. Athena Scientific.

Bertsekas, D., & Tsitsiklis, J. (1996). *Neuro-dynamic programming*. Athena Scientific.

Boyan, J., & Moore, A. (1996). Learning evaluation functions for large acyclic domains. *Proceedings ICML*.

Brafman, R., & Tennenholtz, M. (2001). R-MAX - a general polynomial time algorithm for near-optimal reinforcement learning. *Proceedings IJCAI*.

Dearden, R., Friedman, N., & Andre, D. (1999). Model based Bayesian exploration. *Proceedings UAI*.

Duff, M. (2002). *Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes*. Doctoral dissertation, U. Mass.

Engel, Y., Mannor, S., & Meir, R. (2003). Bayes meets Bellman: The Gaussian process approach to temporal difference learning. *Proceedings ICML*.

Gittins, J. (1989). *Multi-armed bandit allocation indices*. Wiley.

Jordan, M. (Ed.). (1999). *Learning in graphical models*. MIT Press.

Kaelbling, L. P. (1994). Associative reinforcement learning: Functions in k-DNF. *Machine Learning*, *15*, 279–298.

Kearns, M., Mansour, Y., & Ng, A. (2001). A sparse sampling algorithm for near-optimal planning in large markov decision processes. *JMLR*, 1324–1331.

Kearns, M., & Singh, S. (1998). Near-optimal reinforcement learning in polynomial time. *Proceedings ICML*.

Lusena, C., Goldsmith, J., & Mundhenk, M. (2001). Nonapproximability results for partially observable Markov decision processes. *JAIR*, *14*, 83–103.

Martin, J. (1967). *Bayesian decision problems and Markov chains*. Wiley.

Mundhenk, M., Goldsmith, J., Lusena, C., & Allender, E. (2000). Complexity of finite-horizon Markov decision processes. *JACM*, *47*, 681–720.

Neal, R. (Ed.). (1996). *Bayesian learning for neural networks*. Springer.

Ng, A., & Jordan, M. (2000). Pegasus: A policy search method for large MDPs and POMDPs. *Proceedings UAI*.

Péret, L., & Garcia, F. (2004). On-line search for solving Markov decision processes via heuristic sampling. *Proceedings ECAI*.

Salganicoff, M., & Ungar, L. (1995). Active exploration and learning in real-valued spaces using multi-armed bandit allocation indices. *Proceedings ICML*.

Strens, M. (2000). A Bayesian framework for reinforcement learning. *Proceedings ICML*.

Strens, M., & Moore, A. (2002). Policy search using paired comparisons. *JMLR*, *3*, 921–950.

Sutton, R., & Barto, A. (1998). *Reinforcement learning: An introduction*. MIT Press.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, *25*, 285–294.

Watkins, C. (1989). *Learning from delayed rewards*. Doctoral dissertation, King's College Cambridge.

Wiering, M. (1999). *Explorations in efficient reinforcement learning*. Doctoral dissertation, Univ. Amsterdam.

Williams, C. (1999). Prediction with Gaussian processes. In *Learning in graphical models*. MIT Press.

Wyatt, J. (2001). Exploration control in reinforcement learning using optimistic model selection. *Proc. ICML*.