EXTRACTED CHAPTER FROM THESIS: BFS3

JOHN ASMUTH

Bayesian Forward Search Sparse Sampling, or **BFS3**, is a method for applying a special kind of tree-search to a BAMDP. This application results in a policy with provably efficient sample complexity.

1. Building blocks

BFS3 is built upon the ideas from a few existing algorithms.

1.1. **Sparse Sampling. Sparse Sampling** (Kearns et al., 1999) works by recursively expanding a full search tree up to a certain depth d. At the root, each of the A actions is chosen a constant number of times C, yielding a set of $A \cdot C$ children. Sparse sampling is then run on each of the children with a recursion depth one less than the root's. Once the tree is fully created, the leaves are each assigned a value of zero. Then, starting at the leaves, the values are backed up and combined via the Bellman equation, giving the parents' values, until the root's value is determined. The total number of nodes visited in this search tree is $(AC)^d$, making the algorithm impractical to run in all but the most trivial of domains.

It is worth noting, however, that **Sparse Sampling** is best known as one of the first reinforcement-learning planning algorithms that can achieve high accuracy with high probability using an amount of computation that is not a function of the size of the state space[1]. Because of this attractive property, it makes sense to select it or one of its variants as the planner for the infinitely large BAMDP. **Sparse Sampling** is the basis for a number of Monte-Carlo Tree Search (MCTS) algorithms, which are considerably faster in practice (Kocsis & Szepesvári, 2006; Walsh et al., 2010; Wang et al., 2005).

Sparse Sampling is discussed in more detail in Chapter **??**.

1.2. **Forward Search Sparse Sampling. Forward Search Sparse Sampling**, or **FSSS**, is a Monte-carlo tree search algorithm used for planning in MDPs. **BFS3**, introduced in Section 2 of this Chapter, uses **FSSS** as a subroutine. It preferentially expands the search tree through the use of rollouts, and is outlined in Algorithm 1. Unlike either **Bayesian Sparse Sampling** (Wang et al., 2005) or **UCT** (Kocsis & Szepesvári, 2006), it retains the attractive guarantees of the original **Sparse Sampling** algorithm. An important property of **FSSS** is that it maintains hard upper and lower bounds on the values for each state and action, and uses those bounds to direct the rollouts; actions are chosen greedily according

[1]This lack of dependence on the number of states assumes that sampling from the model can be done in constant time. In most real situations there is at least a logarithmic dependency on the number of states just for representing any given state.

Input: state s, max depth d, #trajectories t, MDP M
Output: estimated value for state s
for t *times* **do**
  | FSSS-Rollout$(s, d, 0, M)$
$\hat{V}(s) \leftarrow \max_a U_d(s, a)$
return $\hat{V}(s)$

Algorithm 1: FSSS(s, d, t, M)

to the upper bound on the value, and the next state is chosen such that it is the most uncertain of the available candidates (according to the difference in its upper and lower bounds).

FSSS will find the action to take from a given state s_0, which will be the root of its search tree. The tree is expanded by running t trajectories, or rollouts, of length d. There are theoretically justified ways to choose t and d, but in practical applications they are knobs used to balance computational overhead and accuracy. To run a single rollout, the agent will invoke Algorithm 2, FSSS-Rollout$(s_0, d, 0, M)$. The values $U_d(s)$ and $L_d(s)$ are the upper and lower bounds on the value of the node for state s at depth d, respectively. Each time a rollout is performed, the tree will be expanded. After at most $(AC)^d$ rollouts are finished (but often less in practice), **FSSS** will have expanded the tree as much as is possibly useful, and will agree with the action chosen by **Sparse Sampling**.

2. Bayesian Forward Search Sparse Sampling

BFS3 is the application of **FSSS** to a Bayes-adaptive MDP, or BAMDP, and is outlined in Algorithm 3. The BAMDP is defined by the MDP prior $\phi(M)$, and the joint transition and reward function T-R_ϕ is constructed such that

$$P(\langle s', h \cup (s, a, s', r)\rangle, r | \langle s, h\rangle, a) \quad = \quad \int_M P(s', r|s, a, M)\phi(M|h)dM.$$

Here, the BAMDP's state-space is the set of belief-states that include the history of all transitions seen so far. Because of how the BAMDP's transition function is constructed, the set of next-states that an agent can transition to always have a history that includes that transition.

BAMDPs, their construction, and their relation to Bayes-optimality are discussed in detail in Chapter **??**.

Since, with **FSSS**, the next belief-states are only sampled and their likelihoods are never calculated, a simple generative process can be used:

$$M \sim \phi|h \tag{1}$$

$$s', r \sim T_M(s, a), R_M(s, a). \tag{2}$$

This process is used whenever **BFS3** or its subroutine **FSSS** sample a next-state and reward. The algorithm never holds on to an individual MDP after a single transition has

Input: state s, max depth d, current depth l, MDP M
if *Terminal*(s) **then**
  $U_d(s) = L_d(s) = 0$
  return
if $d = l$ **then**
  return
if $\neg$*Visited*$_d(s)$ **then**
  Visited$_d(s) \leftarrow$ true
  foreach $a \in A$ **do**
    $R_d(s,a), \text{Count}_d(s,a,s'), \text{Children}_d(s,a)$
      $\leftarrow 0, 0, \{\}$
    for C *times* **do**
      $s', r \sim T_M(s,a), R_M(s,a)$
      $\text{Count}_d(s,a,s') \leftarrow \text{Count}_d(s,a,s') + 1$
      $\text{Children}_d(s,a) \leftarrow \text{Children}_d(s,a) \cup \{s'\}$
      $R_d(s,a) \leftarrow R_d(s,a) + r/C$
      if $\neg$*Visited*$_{d+1}(s')$ **then**
        $U_{d+1}(s'), L_{d+1}(s') = V_{\max}, V_{\min}$
  Bellman-backup(s,d)
$a \leftarrow \text{argmax}_a\, U_d(s,a)$
$s' \leftarrow \text{argmax}_{s'}(U_{d+1}(s') - L_{d+1}(s')) \cdot \text{Count}_d(s,a,s')$
FSSS-Rollout$(s', d, l+1, M)$
Bellman-backup(s,d)
return

Algorithm 2: FSSS-Rollout(s,d,l,M)

Input: state s, history h, depth d, #trajectories t, MPD prior ϕ
Output: action to take in state s
if $\langle s,h\rangle \in$ *solved-belief-states* **then**
  return $\pi(\langle s,h\rangle)$
foreach $a \in A$ **do**
  for C *times* **do**
    $\langle s',h'\rangle, r \sim T\text{-}R_\phi(\langle s,h\rangle, a)$ $q(a) \leftarrow q(a) + \frac{1}{C}\left[r + \gamma FSSS(\langle s',h'\rangle, d, t, M_\phi)\right]$
solved-belief-states $\leftarrow$ solved-belief-states $\bigcup \{\langle s,h\rangle\}$
$\pi(\langle s,h\rangle) \leftarrow \text{argmax}_a\, q(a)$
return $\pi(\langle s,h\rangle)$

Algorithm 3: BFS3(s,h,d,t,ϕ)

been sampled from it. Also, note that whenever **FSSS** does a Bellman backup, that backup is done for a belief-state (since **FSSS** is acting on the BAMDP).

First, an MDP M is sampled from the posterior $\phi|h$. Sometimes this can be a computationally expensive action, since inference is generally a hard probelm. The **BFS3** algorithm

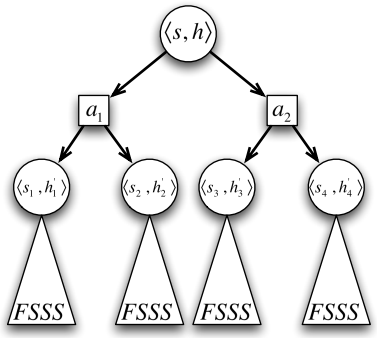

FIGURE 1. **BFS3** samples next-belief-states for each action C times, and then runs **FSSS** on the resulting belief state, using the BAMDP as a generative model. Every node the same distance from the root represents one of the possible worlds that the agent may experience, each with a different history and MDP posterior.

can only work effectively if this step is fast. Once it is sampled, then the next state and reward are sampled from M.

To reconstruct the resulting belief-state, we pack the resulting concrete state s' with the new history made by augmenting h with (s, a, s', r), resulting in a transition from belief-state $\langle s, h\rangle$ to $\langle s', h \cup (s, a, s', r)\rangle$.

Figure 1 illustrates the **BFS3**'s process for each belief-state visited by the agent. In the future, the agent may find itself in one of the reachable belief-states in the search tree.

In many cases, the history h can be summarized by a more compact sufficient statistic, and next-state posteriors can be sampled efficiently (for example, the FDM prior detailed in Chapter **??**).

The sufficient statistic for FDM (and for any discrete state and discrete action MDP) is the set of histograms indicating how many times each state s' has resulted from a particular state-action pair. Due to conjugacy, next-states can be sampled using FDM without sampling an intermediate MDP. The next-state distribution is a multinomial with weights proportional to the corresponding histogram added to the prior vector α.

2.0.1. *Theorem statement.* If:

(1) **FSSS**, given the true MDP m_0 and some bound on computational resources, can provide accurate value estimates with high probability, and
(2) the posterior next-state distribution for some state-action pair, given N examples, will be an accurate estimate of m_0's next-state distribution with high probability,

then: with high probability, **BFS3** will behave Bayes-optimally for all but a polynomial number of steps.

2.1. **Proof sketch.** First, we will show that there is a BAMDP, constructed from the prior ϕ, whose optimal policy is the Bayes-optimal policy for $m_0 \sim \phi$. Then, we will show that **BFS3**, acting in the BAMDP, will satisfy the three criteria required for PAC-MDP behavior (Kakade, 2003; Li, 2009) in that BAMDP[2]. These criteria are: (1) accuracy, (2) bounded discoveries, and (3) optimism.

First, because of Condition 2 in our theorem statement, we know that once we have received N examples of transitions from a state-action pair (s, a), our estimate of the next-state distribution for that pair will be accurate. (This condition need not hold for degenerate priors, but it appears to hold quite broadly.)

Second, since we forget all additional transitions from state-action pairs for which we have seen N examples, the number of possible state-histories that an agent can observe is bounded. Specifically, each time a transition from some state-action (s, a) is observed, either no change will be made to the state-action's histogram (it already sums to N), or exactly one entry in the histogram will be incremented by 1. Since the histogram can be changed at most N times, the total number of histories possible for an agent over the course of a single experiment is $N \cdot S \cdot A$ (N histories for each state-action pair).

A discovery event, or one that potentially changes the MDP posterior, is an event that results in a change to the history. There are $N \cdot S \cdot A$ discoveries possible, since other transitions will be forgotten.

Third, $\mathbf{FSSS}(s', d, t, M_\phi)$ is guaranteed to have an optimistic value estimate for belief-state s' as t (the number of trajectories), our bounded resource, grows smaller. We also know that, from Condition 1 of the theorem, t is sufficient to find accurate estimates of s' if all states in s''s subtree have converged next-state posteriors. Simply put, if s''s subtree has no unknown state-action pairs, then **FSSS**'s estimate of that state's value will be accurate. As a result, if **FSSS**'s estimate of a state's value is inaccurate, there must be something to learn about in s''s subtree. **FSSS** guarantees that this inaccuracy will be optimistic.

Also possible is that the value estimate of s' is accurate *and* there are unknown states in its subtree. In this case, the agent can decide whether or not to visit that state fully informed of its value, and can take a Bayes-optimal action.

The PAC-MDP criteria direct the agent to areas of either high value or high uncertainty, managing the exploration/exploitation tradeoff. Because the agent will only go to areas of high uncertainty over areas of high reward a bounded number of times that grows linearly with the number of possible discovery events, we bound the number of sub-optimal steps taken over the lifetime of the agent.

3. Proof

This section presents the detailed argument that **BFS3** is near Bayes-optimal.

3.1. Terms.

- N is the maximum number of transitions that will be observed from any particular state-action pair. All subsequent transitions will not be recorded.

[2]PAC-MDP behavior in the BAMDP implies near Bayes-optimal behavior in the learning setting.

- The history $h \in H$ is the collection of all observed transitions $(s, a) \to s'$.
- ϕ is the MDP prior. $\phi|h$ is the MDP posterior.
- m_0 is the unknown true MDP.
- $P_m(s'|s, a)$ is the probability of going from s to s' when performing action a in some MDP m.
- $P_{\phi|h}(s'|s, a) = \int_m \phi(m|h)P_m(s'|s, a)dm$ is the posterior transition likelihood, integrating out all possible MDPs.
- $V_m(s)$ is the true value of state s according to MDP m.
- $\text{FSSS}_m(s, B)$ is the value of state s estimated by **FSSS** when using s as a root, m as an oracle (that samples next-states given a state-action pair), and with computational resource bounds represented by B. In this context, B is the number of trajectories and the search depth that **FSSS** uses to perform roll-outs.

3.2. **Prerequisites.**

(1) The **planning assumption**: For any s, w.p. $1 - \delta_p$, $|\text{FSSS}_{m_0}(s, B) - V_{m_0}(s)| \leq \epsilon_p$. In other words, **FSSS** will estimate values accurately if it is given the true model and at least B computation time.

(2) The **accuracy assumption**: For some state-action pair (s, a), if h includes N samples from $P_{m_0}(s'|s, a)$, then w.p. $1 - \delta_c/(SA)$,

$$\forall_{s'} \ |P_{\phi|h}(s'|s, a) - P_{m_0}(s'|s, a)| \leq \alpha_c.$$

As a result, if N examples of each of the $S \cdot A$ state-action pairs are observed, w.p. $1 - \delta_c$, all transition probability deviations are bounded by α_c.

Note: this condition removes the need to say $m_0 \sim \phi$. All of our closeness assumptions are already met.

(3) The **MCTS simulation conjecture**, described in Section 3.3.

3.3. **MCTS simulation conjecture.** Let an MCTS algorithm $\mathcal{A}_m(s, B)$ be an estimator of the value of state s in MDP m, bounded by computational resources B, whose sole source of stochasticity comes from using m as a generator of $s' \sim s, a|m$. **FSSS** and **UCT** (Kocsis & Szepesvári, 2006) both fall into this category; their resources are the number of roll-outs and maximum search depth.

Intuitively, we conjecture that if $\mathcal{A}$ can reliably find accurate values for states in one MDP, then it can also reliably find accurate values for the MDP when using an approximate second MDP as its next-state generator.

If,

(1) for all s, a, s', $|P_{m_a}(s'|s, a) - P_{m_b}(s'|s, a)| \leq \alpha_c$,
(2) for all s, w.p. $1 - \delta_p/(CA)$, $|\mathcal{A}_{m_a}(s, B) - V_{m_a}(s)| \leq \epsilon_p$,

then

- for all s, w.p. $1 - \delta_{p'}/(CA)$, $|\mathcal{A}_{m_b}(s, B) - V_{m_a}(s)| \leq \epsilon_{p'}$,

where $1/\delta_{p'} = \text{poly}(1/(1-\alpha_c), 1/\delta_p, 1/\epsilon_p, m_a, B)$ and $\epsilon_{p'} = \text{poly}(1/(1-\alpha_c), 1/\delta_p, 1/\epsilon_p, m_a, B)$.

3.3.1. *Note on $C \cdot A$.* The probability $1-\delta_p/(CA)$ is used because, for each step, **BFS3** will invoke **FSSS** $C \cdot A$ times, or C times for each action. This choice was made to ensure that all next states get enough resources to ensure accuracy when their dynamics are known. We use the union bound to say, w.p. $1 - \delta_p$, *all* such next states have accurate estimates.

3.4. **PAC-MDP behavior in the BAMDP.** Let

- the BAMDP be $m_{\phi|h}$,
- the set of all possible histories be H,
- the state space of the BAMDP be the set of belief-states $\mathbb{S} = S \times H$, and
- the action space of the BAMDP be the same as the regular MDP.

We shall limit H to contain only histories with at most N examples of transitions from any state-action pair. That is, when the agent observes the N_{th} transition from state s and action a, it shall never again update its history when making a transition from that state with that action. This forgetfulness causes the set H to be finite.

In a discrete state and action MDP, the history $h \in H$ may be summarized by a set of histograms, one for each state-action pair. Since we ignore all transitions from any state-action pair after the N_{th} transition from that state-action pair, we know that the sum of all entries in all histograms cannot exceed $S \cdot A \cdot N$.

Since, whenever the history *is* updated, a single entry for a single histogram will be incremented by 1, we can infer that at most $S \cdot A \cdot N$ unique histories can possibly be experienced by an agent over the course of a single experiment.

The agent can also visit at most S true states (all of them). Since there are S true states and $S \cdot A \cdot N$ possible histories, the number of belief-states that an agent can visit, in a single experiment, is $M = S^2 \cdot A \cdot N$.

Our strategy will be to say that, for each of these at most M possible belief-states visited by the agent, the agent will either choose an action that is approximately optimal in the BAMDP (and therefore approximately Bayes-optimal in the true MDP), or exploratory in the BAMDP (with no guarantees about how good this action is in the true MDP).

We say that an agent is near Bayes-optimal if we can limit the number of exploratory actions taken to a polynomial of the domain parameters.

To show PAC-MDP behavior in the BAMDP (and, therefore, near Bayes-optimal behavior), we need to show that three conditions hold (Li, 2009):

(1) bounded discoveries,
(2) accuracy, and
(3) optimism.

To understand these criteria and how they connect PAC-MDP behavior in the BAMDP to near Bayes-optimal behavior in the true MDP, we need to define the concept of *known* and *unknown states* in m_0 and *known* and *unknown belief-states* in m_ϕ.

A *state* s is considered known if the history h contains N examples of transitions from s for each action $a \in A$.

A *belief-state* $\langle s, h \rangle$ is considered known if **FSSS**'s estimate of its value is accurate.

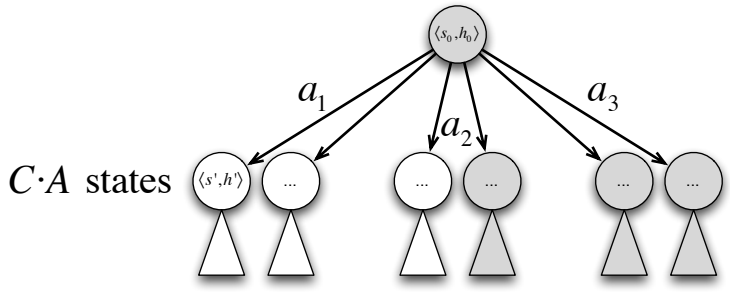

FIGURE 2. **BFS3** will run **FSSS** on each of the $C \cdot A$ next belief-states, $\langle s', h' \rangle$ and its peers. Triangles below belief-states represent search trees explored by **FSSS**. A greyed search tree indicates that some of the true states visited in that search tree are unknown, and the value estimate may be inaccurate. Because we are using **FSSS**, an inaccurate estimate is guaranteed to be optimistic, and can bubble up to the root.

Figure 2 illustrates how unknown states in a belief-state's **FSSS** subtree may cause a belief-state's estimate to be inaccurate. Because of Prerequisite 1, the planning assumption, we know that if there are no unknown states in a belief-state's subtree, then the belief-state's estimate will be accurate, and we can consider the belief-state to be known.

It is possible for a belief-state to be known if there are unknown states in its subtree; in the limit of computation, **FSSS** will be accurate for all belief-states. But, it is not possible for all states in the belief-state's subtree to be known if the belief-state is not known.

As a result, an unknown belief-state indicates an unknown state that is at most D steps away, where D is the search depth given to **FSSS**. Since unknown belief-states always have optimistic estimates when they are evaluated with **FSSS**, the agent is pulled towards unknown states.

3.4.1. *Satisfying the PAC-MDP criteria.* **BFS3** is PAC-MDP for the belief-state MDP because it satisfies the 3 criteria.

Condition 1 holds because if the agent reaches an unknown belief state (one with an inaccurate, optimistic value), it must be because there is an unknown state-action pair beneath it in the tree *and* this node will be reached by the current policy with high probability. (Quantifying this claim involves an argument that closely parallels the "explore or exploit" lemma from PAC-MDP theory.) Note that when an unknown state-action pair is reached, it moves closer to being known. In fact, an agent can only do this update $S \cdot A \cdot N$ times, so, the number of discoveries is bounded.

Condition 2 holds because of Prerequisite 2 (the accuracy assumption). Once we have N examples of a state-action pair, we have an accurate estimate of the next-state distribution for that pair. Once we have N examples of all state-action pairs, we have an accurate estimate of the entire MDP.

Condition 3 holds because of our definition of known and unknown belief-states and the behavior of **FSSS**. Known belief-states have accurate values, and unknown belief-states have optimistic values. No belief-state ever has a pessimistic value.

3.5. **How accurate, and how likely?** We have shown that **BFS3** is near Bayes-optimal, meaning with high probability, it will be accurate for all but a small number of steps. This section details how accurate **BFS3** is and with what probability it achieves this accuracy.

3.5.1. *Bayes-optimal behavior for a converged posterior.* We declare the posterior to be converged if, for each s, a, our history h includes N examples of a transition from s, a.

Let m_0 and $\phi|h$ take the place of m_a and m_b in the MCTS simulation conjecture. By Prerequisite 2, we have Condition 1 for the MCTS simulation conjecture. By Prerequisite 1, we have Condition 2 for the MCTS simulation conjecture. For each step in the environment, **BFS3** runs **FSSS** C times for each action, resulting in $C \cdot A$ executions of **FSSS**. We can put a lower bound on the likelihood that they are all $\epsilon_{p'}$-accurate of $1 - \delta_{p'}$.

The estimate for the value of taking action a from the root state s_0 is calculated as

$$\hat{Q}(s_0, a) = R(s_0, a) + \gamma \frac{1}{C} \sum_{i=1}^{C} \mathrm{FSSS}_{\phi|h}(s_i, B). \tag{3}$$

Let the Monte Carlo estimate of $Q(s_0, a)$, using the true value for all next states, be

$$\tilde{Q}(s_0, a) = R(s_0, a) + \gamma \frac{1}{C} \sum_{i=1}^{C} V^*(s_i). \tag{4}$$

From the Bellman equation, we know that

$$Q^*(s_0, a) = R(s_0, a) + \gamma \sum_{s_i} P_{m_0}(s_i|s, a) V^*(s_i). \tag{5}$$

First, let's bound the difference between $\hat{Q}(s_0, a)$ and $\tilde{Q}(s_0, a)$.

$$\begin{aligned}
|\hat{Q}(s_0, a) - \tilde{Q}(s_0, a)| &= \left| \gamma \frac{1}{C} \sum_{i=1}^{C} \mathrm{FSSS}_{\phi|h}(s_i, B) - \gamma \frac{1}{C} \sum_{i=1}^{C} V^*(s_i) \right| \\
&= \left| \gamma \frac{1}{C} \sum_{i=1}^{C} \left(\mathrm{FSSS}_{\phi|h}(s_i, B) - V^*(s_i) \right) \right| \\
&\leq \gamma \frac{1}{C} \sum_{i=1}^{C} \left| \mathrm{FSSS}_{\phi|h}(s_i, B) - V^*(s_i) \right| \\
&\leq \gamma \frac{1}{C} \sum_{i=1}^{C} \epsilon_{p'} \\
|\hat{Q}(s_0, a) - \tilde{Q}(s_0, a)| &\leq \epsilon_{p'}. \qquad (6)
\end{aligned}$$

Second, let's bound the difference between $\tilde{Q}(s_0, a)$ and $Q^*(s_0, a)$.

$$\begin{aligned} |Q^*(s_0,a) - \tilde{Q}(s_0,a)| &= \left| \gamma \sum_{s_i} P_{m_0}(s_i|s,a) V^*(s_i) - \gamma \frac{1}{C} \sum_{i=1}^{C} V^*(s_i) \right| \\ &= \gamma \left| \sum_{s_i} P_{m_0}(s_i|s,a) V^*(s_i) - \frac{1}{C} \sum_{i=1}^{C} V^*(s_i) \right| \end{aligned}$$

$$|Q^*(s_0,a) - \tilde{Q}(s_0,a)| \leq \lambda, \tag{7}$$

w.p. $1 - e^{-\lambda^2 C / V_{\max}^2}$ (Kearns et al., 1999).

This bounds the error in the estimated Q-values:

$$\begin{aligned} |\hat{Q}(s_0,a) - Q^*(s_0,a)| &= |\hat{Q}(s_0,a) - \tilde{Q}(s_0,a) + \tilde{Q}(s_0,a) - Q^*(s_0,a)| \\ &\leq |\hat{Q}(s_0,a) - \tilde{Q}(s_0,a)| + |\tilde{Q}(s_0,a) - Q^*(s_0,a)| \end{aligned}$$

$$|\hat{Q}(s_0,a) - Q^*(s_0,a)| \leq \epsilon_{p'} + \lambda, \tag{8}$$

w.p. $1 - \delta_{p'} - e^{-\lambda^2 C / V_{\max}^2}$.

Let $\epsilon = \epsilon_{p'} + \lambda$, and $\delta = \delta_c + S\left(\delta_{p'} + e^{-\lambda^2 C / V_{\max}^2}\right)$.

Once the model has converged, every time the agent takes a step from some state s_0 it remembers what action it took, and will take that action in the future. The likelihood of it choosing an ϵ-optimal action for each of the S possible states is no less than $1 - \delta$.

3.5.2. *Bayes-optimal behavior for an unconverged posterior.* Since the agent will not update the history h on a transition from s, a when transitions s, a have been observed N times in the past, we know there are only $S \cdot A \cdot N$ different histories possible over the course of a single experiment.

Given a particular history h and state s that has been experienced before, **BFS3** will choose the same action as last time, limiting the total amount of computation possible, and also the total number of times **BFS3** has to succeed is limited at $M = S^2 \cdot A \cdot N$, or the number of states times the number of possible histories.

In Sections 3.5.3 and 3.5.4, we set conditions for **BFS3** to choose either an exploratory action or an optimal action for each of these M possible events.

On a given step, **BFS3** will run **FSSS** on each of A possible actions and, for each action, C possible next-states.

3.5.3. *In the limit of infinite computation.* Each of the next-state possibilities will be fully explored and FSSS will agree with Sparse Sampling, giving an error no more than

$$\epsilon_\infty = \frac{\lambda(1 - \gamma^{D+1})}{1 - \gamma} + \gamma^D V_{\max},$$

w.p. at least $1 - (C \cdot A)^D e^{-\lambda^2 C / V_{\max}^2}$, where D is the maximum search depth. Let $\delta_\infty / M = (C \cdot A)^D e^{-\lambda^2 C / V_{\max}^2}$; the agent will then be able to pick an ϵ_∞-optimal action for each of the M events w.p. at least $1 - \delta_\infty$.

3.5.4. *Limited computation.* In cases where there is limited computation and all states in the subtree have converged according to Prerequisite 2, we are guaranteed to act ϵ_f-optimal w.p. at least $1 - \delta_f/M$, according to Section 3.5.1, where

$$\epsilon_f = \epsilon_{p'} + \lambda$$

and

$$\delta_f/M = \delta_{p'} + e^{-\lambda^2 C/V_{\max}^2}.$$

In cases with limited computation and when there are states in the subtree that are not converged, we are guaranteed to have an ϵ_f-*optimistic* estimate w.p. at least $1 - \delta_f/M$. This is true because of the guarantees provided by **FSSS**.

An action a's estimate is ϵ-optimistic if

$$\hat{Q}(s, a) \geq Q^*(s, a) - \epsilon.$$

Note that an ϵ-optimal estimate is also ϵ-optimistic.

Since for each of the M events we have an ϵ_f-optimistic estimate w.p. $1 - \delta_f/M$, we will have an ϵ_f-optimistic estimate for each event simultaneously w.p. $1 - \delta_f$.

References

Kakade, S. M. (2003). *On the sample complexity of reinforcement learning.* Doctoral dissertation, Gatsby Computational Neuroscience Unit, University College London.

Kearns, M., Mansour, Y., & Ng, A. Y. (1999). A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)* (pp. 1324–1331).

Kocsis, L., & Szepesvári, C. (2006). Bandit based Monte-Carlo planning. *Proceedings of the 17th European Conference on Machine Learning (ECML-06)* (pp. 282–293). Springer Berlin / Heidelberg.

Li, L. (2009). *A unifying framework for computational reinforcement learning theory (pp 78-79).* Doctoral dissertation, Rutgers University.

Walsh, T., Goschin, S., & Littman, M. (2010). Integrating sample-based planning and model-based reinforcement learning. *Proceedings of the Association for the Advancement of Artificial Intelligence.*

Wang, T., Lizotte, D., Bowling, M., & Schuurmans, D. (2005). Bayesian sparse sampling for on-line reward optimization. *ICML '05: Proceedings of the 22nd International Conference on Machine Learning* (pp. 956–963). New York, NY, USA: ACM.