Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2009-044 September 23, 2009

Efficient POMDP Forward Search by Predicting the Posterior Belief Distribution

Ruijie He and Nicholas Roy

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Efficient POMDP Forward Search by Predicting the Posterior Belief Distribution

Ruijie He
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02142
ruijie@mit.edu

Nicholas Roy
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02142
nickroy@mit.edu

Abstract

Online, forward-search techniques have demonstrated promising results for solving problems in partially observable environments. These techniques depend on the ability to efficiently search and evaluate the set of beliefs reachable from the current belief. However, enumerating or sampling action-observation sequences to compute the reachable beliefs is computationally demanding; coupled with the need to satisfy real-time constraints, existing online solvers can only search to a limited depth. In this paper, we propose that policies can be generated directly from the *distribution* of the agent's posterior belief. When the underlying state distribution is Gaussian, and the observation function is an exponential family distribution, we can calculate this distribution of beliefs without enumerating the possible observations. This property not only enables us to plan in problems with large observation spaces, but also allows us to search deeper by considering policies composed of multi-step action sequences. We present the Posterior Belief Distribution (PBD) algorithm, an efficient forward-search POMDP planner for continuous domains, demonstrating that better policies are generated when we can perform deeper forward search.

1 Introduction

The Partially Observable Markov Decision Process (POMDP) is a general framework for sequential decision making in partially observable environments, when the agent is unable to exactly observe the state of its environment. Traditionally, a POMDP solver generates a policy offline, computing an action for a set of possible beliefs before policy execution. However, for problems with large domains, offline methods can incur significant computation costs. Recently, online forward-search methods have demonstrated promising results in problems with large domains (see [17] for a review), suggesting that POMDP planning can be performed efficiently by only considering the belief states that are reachable from the agent's current belief.

If a POMDP solver is able to search deep enough, it will find the optimal policy for the current belief [6, 14]. Unfortunately, the number of belief states reachable within depth D is $(|A||Z|)^D$, where $|A|$ and $|Z|$ are the sizes of the action and observation sets. Not only does the search quickly become intractable as D increases, but online techniques generally have to meet real-time constraints, which limits the planning time available for each iteration. Existing online, forward search algorithms seek to reduce the number of possible observations that have to be explored by using branch-and-bound [10], Monte Carlo sampling [9] and heuristic search [15, 21] techniques.

Fundamentally, these algorithms still branch on the possible individual actions and observations to determine the set of reachable posterior beliefs. An alternative approach would be to consider shallow policies composed of multi-step action sequences, or macro-actions [20], branching only at the end of each action sequence. However, to plan with multi-step action sequences, an algorithm must have the ability to determine the set of posterior beliefs that could result after the action sequence, since the goal of a POMDP solver is to generate the policy that maximizes the agent's expected discounted reward. This set of beliefs is usually computed by enumerating or sampling from the set of observation sequences, which is itself a costly process and reduces the potential savings of macro-actions. If it were possible to efficiently characterize the *distribution* of posterior beliefs after an action sequence without enumerating the possible observations, forward search POMDP planning could then be done much more efficiently. If the distribution over posterior beliefs can be computed efficiently and is of a low dimension, then sampling from this distribution requires substantially fewer samples and much less computation, allowing much faster search and efficient planning in POMDP problems with large observation spaces.

In this paper, we demonstrate that when the agent's belief and observation models can be represented in parametric form, the distribution of the agent's posterior beliefs can be directly computed for a multi-step action sequence. Parametric representations have previously been proposed [2, 3, 12, 16] as an alternative for compactly representing high-dimensional belief spaces, and are especially valuable for POMDP problems with continuous state spaces. Specifically, we focus on problems where the agent's belief is reasonably represented as a Gaussian distribution over a continuous state space, and where the transition and observation models belong to any member of the exponential family of distributions, such as the linear-Gaussian or multinomial distributions that often characterize POMDP problems. By also constraining the agent's posterior belief to the Gaussian parametric representation, we can directly compute how the sufficient statistics of the agent's belief are expected to evolve over multi-step action sequences. Furthermore, for Gaussian distributions, we will see that the second moment of the belief distribution (i.e., the covariance) can be computed in amortized $\mathcal{O}(1)$ for a given multi-step action sequence, increasing the efficiency of the search process. We present the Posterior Belief Distribution (PBD) algorithm, an efficient, POMDP forward search algorithm that can perform much deeper forward searches.

2 POMDPs

Formally, a POMDP consists of a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, and a set of observations $\mathcal{Z}$. It also includes a state-transition model $p(s'|s,a)$, an observation model $p(z|a,s')$, a reward model $r_S(s,a)$, as well as a discount factor γ and initial belief bel_0. The goal of a POMDP solver is to compute a policy π mapping beliefs to actions $\pi : bel \rightarrow a$ that will maximize the agent's expected total reward over its lifetime. Given a policy π and current belief bel, the agent takes an action $a = \pi(bel)$ and obtains an observation z. It then updates its belief according to

$$bel'(s') = \tau(bel, a, z) = \eta \, p(z|a,s') \int_{s \in \mathcal{S}} p(s'|s,a) bel(s) ds \tag{1}$$

where $\tau(bel, a, z)$ represents the belief update function and η is a normalization constant. Each policy π is also associated with a value function $V_\pi : bel \rightarrow \mathcal{R}$, specifying the expected total reward of executing policy π starting from bel

$$V_\pi(bel) = \max_{a \in \mathcal{A}} \big[r_B(bel, a) + \gamma \sum_{z \in \mathcal{Z}} p(z|bel, a) V_\pi(\tau(bel, a, z)) \big] \tag{2}$$

where the function $r_B(bel, a) = \int_{s \in \mathcal{S}} bel(s) r_S(s,a) ds$ specifies the immediate expected reward of executing action a in belief bel. A POMDP solver seeks to find the optimal policy $\pi^*(bel_0)$ that maximizes $V_{\pi^*}(bel_0)$.

Traditionally, policies have been computed offline, and one class of POMDP solvers that has achieved particular success is the point-based methods. For discrete state spaces, point-based approaches such as PBVI [11] and HSVI [18] leverage the piecewise-linear and convex (PWLC) aspects of the value function [19] to obtain lower bounds on the value function (Eqn. 2), performing value updates only at selected belief states. The value function has similarly been shown [12] to be PWLC for continuous state spaces.

3 Forward Search in Parametric Space

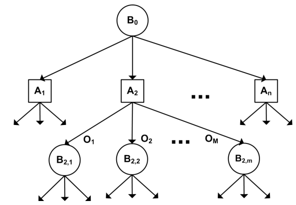

Fig. 1: A forward search tree. An action is chosen at each belief node (OR-node), while all observations must be considered at the action nodes (AND-node)

Rather than computing a policy for every possible belief state, forward search techniques avoid the computational complexity of full policy computation by directing computational effort only towards belief states that are reachable from the current belief under different actions. These techniques alternate between a planning and execution phase, planning online only for the belief at the current timestep. During the planning phase, a forward search algorithm creates an AND-OR tree (Fig. 3) of reachable belief states from the current belief state. The tree is expanded using action-observation pairs that are admissible from the current belief, and the beliefs at the leaf nodes are found using Eqn. 1. By using a value heuristic [17] that estimates the value at the fringe nodes, the expected value of executing a policy from the current belief can be propagated up the tree to the root node (Eqn. 2).

To obtain the set of reachable beliefs, existing forward search algorithms branch on the possible actions and observations at each successive depth. Unfortunately, the branching factor for reasonably large discrete or continuous action and observation sets severely limits the maximum search depth achievable in real-time. Even if we restrict our action space to a set of macro-actions, and compute the expected reward of each macro-action by sampling observation sequences of corresponding length [20], the size of the observation space and sampling complexity will grow exponentially with the length of the action sequence.

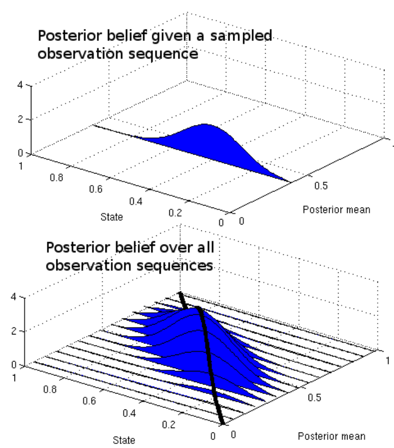

Fig. 2: Distribution of posterior beliefs. a) A Gaussian posterior belief results after incorporating an observation sequence. b) Over all possible observation sequences, the distribution of posterior means is a Gaussian (black line), and for each posterior mean, a Gaussian (blue curve) describes the agent's posterior belief.

For a particular macro-action, the probability of the agent obtaining an observation sequence is equivalent to the probability of obtaining the posterior belief associated with that observation sequence. Seen from another angle, every macro-action generates a distribution over beliefs, or a distribution of distributions. If we are able to calculate the distribution over posterior beliefs for every action sequence, and branch at the end of the action sequence by sampling posterior beliefs within this distribution, the sampling complexity is then independent of the macro-action length. Furthermore, the expected reward of an action sequence can then be computed by finding the expected rewards with respect to that distribution, rather than by sampling the possible observations.

In this paper, we focus on problems where the agent's belief $bel = N(\mu, \Sigma)$ is normally distributed over the state space, and the observations are drawn from an exponential family distribution [1]. Without loss of generality, only the observations are modeled as an exponential family distribution here, though the same analysis could be applied to the state-transition model. These model assumptions imply that the posterior belief is not strictly Gaussian, since the Gaussian distribution is not a conjugate prior for generic exponential family observation models. We nevertheless assume that the agent's posterior belief remains Gaussian, and show in Section 4 that the distribution over posterior beliefs is itself a Gaussian over Gaussian beliefs (Fig. 3). We will show that all posterior beliefs in this distribution have the same covariance, and the posterior means are normally distributed over the continuous state space. Given an action sequence, the posterior distribution over beliefs is therefore a joint distribution over the posterior means and the corresponding distribution over states. We can then evaluate the expected reward of an action sequence by performing Monte Carlo integration over this distribution of distributions.

4 Gaussian Posterior Prediction

Our state-transition and observation models can be represented as follows:

$$s_t = A_t s_{t-1} + B_t u_t + \varepsilon_t, \qquad s_{t-1} \sim N(\mu_{t-1}, \Sigma_{t-1}), \qquad \varepsilon_t \sim N(0, R_t) \tag{3}$$

$$p(z_t|\theta_t) = \exp(z_t^T \theta_t - b_t(\theta_t) + \kappa_t(z_t)) \tag{4}$$

We assume that our state-transition model is linear-Gaussian, and A_t and B_t are the linear transition matrices. θ_t and $b_t(\theta_t)$ are respectively the canonical parameter and normalization factor of the exponential family distribution that generates the observation.

The exponential family encompasses a large set of parametric distributions, including the Gaussian distribution. When the state-transition and observation models are normally distributed and linear functions of the state, the Kalman filter provides a closed-form solution for the posterior belief (μ_t, Σ_t), given a prior belief $(\mu_{t-1}, \Sigma_{t-1})$,

$$\overline{\mu}_t = A_t \mu_{t-1} + B_t u_t \qquad \mu_t = \overline{\mu}_t + K_t(z_t - C_t \overline{\mu}_t) \tag{5}$$

$$\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \qquad \Sigma_t = (C_t^T Q_t^{-1} C_t + \overline{\Sigma}_t^{-1})^{-1}, \tag{6}$$

where C_t is the observation matrix, R_t and Q_t are the covariances of the Gaussian process and measurement noise respectively, and K_t is the Kalman gain. $\overline{\mu}_t$ and $\overline{\Sigma}_t$ are the mean and covariance after an action is taken but before incorporating the measurement.

Eqn. 5 and 6 show that for problems with linear-Gaussian state-transition and observation models, the covariance update is independent of the observation obtained. This is because the Fisher information associated with the observation model, $M_t = C_t^T Q_t^{-1} C_t$, is dependent only on the observation model parameters, rather than the observation obtained [4]. For linear-Gaussian models, M_t is also constant across the entire state space.

Unfortunately, the linear-Gaussian assumption is highly restrictive, and most POMDP models have observation functions that are non-Gaussian. A more general form of the Kalman filter update exists, which allows for a closed-form solution of the posterior belief for problems with observation models that belong to a larger class of parametric distributions, the exponential family. Building on statistical economics research for time-series analysis of non-Gaussian observations [5], a dynamic generalized linear model [22] has been shown to provide the exponential family equivalent of the Kalman filter (efKF). The key idea is to construct linear-Gaussian models which approximate the non-Gaussian exponential family model in the neighborhood of the conditional mode, $s_t|z_t$. The approximate linear-Gaussian observation model can then be used in a traditional Kalman filter. Since this idea was developed elsewhere, the derivation of the filter is presented as an Appendix, and we present the main equations here.

Constructing the approximate linear-Gaussian model requires computation of the first two moments of the distribution and linearizing about the mean estimate at every timestep. For an exponential family observation model, the first two moments of the distribution [22] are,

$$E(z_t|\theta_t) = \dot{b}_t = \frac{\partial b_t(\theta_t)}{\partial \theta_t}\Big|_{\theta_t = W(\overline{\mu}_t)} \qquad Var(z_t|\theta_t) = \ddot{b}_t = \frac{\partial^2 b_t(\theta_t)}{\partial \theta_t \partial \theta_t^T}\Big|_{\theta_t = W(\overline{\mu}_t)} \qquad \theta_t = W(s_t), \tag{7}$$

where $\dot{b}_t$ and $\ddot{b}_t$ are the derivatives of the exponential family distribution's normalization factor, both linearized about $\overline{\theta}_t = W(\overline{\mu}_t)$. $W(.)$ is the canonical link function, which maps the states to canonical parameter values, and depends on the particular member of the exponential family.

Given an action-observation sequence, the posterior mean of the agent's belief in the efKF can then be updated according to

$$\overline{\mu}_t = A_t \mu_{t-1} + B_t u_t \qquad \mu_t = \overline{\mu}_t + \tilde{K}_t(\tilde{z}_t - W(\overline{\mu}_t)), \tag{8}$$

$$\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \qquad \Sigma_t = (\overline{\Sigma}_t^{-1} + Y_t \ddot{b}_t Y_t^T)^{-1}, \tag{9}$$

where $\tilde{K}_t = \overline{\Sigma}_t Y_t (Y_t \overline{\Sigma}_t Y_t + \ddot{b}_t^{-1})^{-1}$ is the efKF Kalman gain, and $\tilde{z}_t = \overline{\theta}_t - \ddot{b}_t^{-1} \cdot (\dot{b}_t - z_t)$ is the projection of the observation onto the parameter space of the exponential family observation model. $Y_t = \frac{\partial \theta_t}{\partial s_t}\big|_{s_t=\overline{\mu}_t}$ is the gradient of the exponential family distribution's canonical parameter, linearized about $\overline{\mu}_t$.

While our relaxation of the observation model to the exponential family necessarily implies that the Gaussian posterior belief is an approximation of the true posterior, the Gaussian assumption does allow the use of a Kalman filter variant, which in turn allows us to approximate our distribution of posterior beliefs efficiently and in closed-form. In the following subsections, we take advantage of the efKF to compute the distribution of the posterior beliefs after a multi-step action sequence. Since our belief is assumed to be Gaussian, expressing a distribution over the posterior beliefs requires us to have a distribution over the posterior means and covariances.

4.1 Prediction of Posterior Mean Distribution

Eqn. 8 reveals that the posterior mean μ_t directly depends on the observation z_t. Nevertheless, we demonstrate in this section that given a current prior belief after an action, $bel_t \sim N(\overline{\mu}_t, \overline{\Sigma}_t)$, the expected *distribution* of the posterior means $p(\mu_t|\overline{\mu}_t)$ is normally distributed about $\overline{\mu}_t$.

Given the two moments of the exponential family observation model, we can represent the conditional distribution $\tilde{z}_t|s_t$ according to

$$\begin{aligned} \tilde{z}_t|s_t &\sim N(W(s_t), \ddot{b}_t^{-1}) && (10)\\ &\sim N(W(\overline{\mu}_t) + Y_t(s_t - \overline{\mu}_t), \ddot{b}_t^{-1}) && (11) \end{aligned}$$

We can then marginalize out s_t using $p(\tilde{z}_t|\overline{\mu}_t) = \int p(\tilde{z}_t|s_t, \overline{\mu}_t) p(s_t|\overline{\mu}_t) ds_t$ and using linear transformations,

$$\begin{aligned} \tilde{z}_t|\overline{\mu}_t &\sim N(W(\overline{\mu}_t), Y_t \overline{\Sigma}_t Y_t^T + \ddot{b}_t^{-1}) && (12)\\ \overline{\mu}_t + \tilde{K}_t(\tilde{z}_t - W(\overline{\mu}_t))|\overline{\mu}_t &\sim N(\overline{\mu}_t, \tilde{K}_t(Y_t \overline{\Sigma}_t Y_t^T + \ddot{b}_t^{-1})\tilde{K}_t^T) && (13)\\ \mu_t|\overline{\mu}_t &\sim N(\overline{\mu}_t, \overline{\Sigma}_t Y_t \tilde{K}_t^T) && (14) \end{aligned}$$

Eqn. 14 indicates that the posterior mean after a measurement update μ_t is normally distributed about the $\overline{\mu}_t$, with a covariance that depends on the prior covariance $\overline{\Sigma}_t$ and the observation model parameters Y_t and $\ddot{b}_t$. The observation model parameters are linearized about the prior mean $\overline{\mu}_t$; hence, for an action-observation sequence of length 1, the parameters are independent of the observation that will be obtained.

To obtain the posterior mean distribution after a multi-step action sequence update, we first combine the process and measurement updates by marginalizing out $\overline{\mu}_t$ using $p(\mu_t|\mu_{t-1}) = \int p(\mu_t|\overline{\mu}_t) p(\overline{\mu}_t|\mu_{t-1}) d\overline{\mu}_t$, obtaining

$$\mu_t|\mu_{t-1} \sim N(A_t \mu_{t-1} + B_t u_t, \overline{\Sigma}_t Y_t \tilde{K}_t^T) \quad (15)$$

We assumed above that for a one-step belief update, μ_{t-1} is a fixed value. For a multi-step update, the mean is a random variable, i.e. $\mu_{t-1} \sim N(m_{t-1}, S_{t-1})$. We can then marginalize out μ_{t-1} to obtain

$$\mu_t \sim N(A_t m_{t-1} + B_t u_t, S_{t-1} + \overline{\Sigma}_t Y_t \tilde{K}_t^T) \quad (16)$$

Equation 16 can now be used to perform a prediction of the posterior mean distribution after a multi-step action sequence. Assuming that the agent is currently at time t and has a particular prior mean $\mu_{t-1} \sim N(\mu_{t-1}, 0)$, the posterior mean after an action sequence of T timesteps is therefore

$$\mu_{t+T} \sim N(f(\mu_{t-1}, A_{t:t+T}, B_{t:t+T}, u_{t:t+T}), \sum_{i=t}^{t+T} \overline{\Sigma}_i Y_i \tilde{K}_i^T) \quad (17)$$

where $f(\mu_{t-1}, A_{t+1:t+T}, B_{t+1:t+T}, u_{t+1:t+T})$ is the deterministic transformation of the means according to $\mu_{t+k} = A_{t+k}\ \mu_{t+k-1} + B_{t+k}\ u_{t+k}$. Since an observation on its own does not shift the mean value m_{t+k} of the distribution of posterior means, m_{t+k} is dependent only on the state-transition model parameters and can be calculated via a recursive update along the action sequence.

4.2 Single-step Prediction of Covariance

Eqn. 9 dictates how the posterior covariance of the agent's belief can be calculated, after an action is taken and an observation is obtained. Given that the Fisher information associated with the observation model $M_t = Y_t \ddot{b}_t Y_t^T$ is independent of the observations, the posterior covariance can be computed in closed form, and is independent of the posterior mean.

For greater efficiency, it has been shown [13] that for a Kalman filter model, the process and measurement updates can be compiled into a single linear transfer function by using a factored representation of the covariance, $\Sigma_{t-1} = B_{t-1}C_{t-1}^{-1}$. Adapting this result to our model, the complete covariance update step can be written as:

$$\Psi_t = \begin{bmatrix} B \\ C \end{bmatrix}_t = \begin{bmatrix} 0 & I \\ I & Y\ddot{b}Y \end{bmatrix}_t \begin{bmatrix} 0 & A^{-T} \\ A & RA^{-T} \end{bmatrix}_t \begin{bmatrix} B \\ C \end{bmatrix}_{t-1}, \tag{18}$$

which enables us to collapse multi-step covariance updates into a single step $\Psi_{t+1:T} = \prod_{i=1}^{T} \Psi_i$, recovering the posterior covariance Σ_{t+T} from $\Sigma_{t+T} = B_{t+T}C_{t+T}^{-1}$. Once $\Psi_{t+1:T}$ is constructed for a given action sequence, it can be reused for the same action sequence at future beliefs.

Calculating the Fisher information associated with the observation model M_t again requires us to perform linearizations about the mean of the agent's prior belief $\overline{\mu}_{t-1}$. This implies that the covariance update after a multi-step action sequence depends on the posterior mean after each timestep. Nevertheless, as shown in Section 4.1, the agent's expectation of the posterior mean after a measurement update μ_{t+1} is a normal distribution around the $\overline{\mu}_t$. Hence, when considering the effect of taking an action sequence from the agent's current belief, we perform our linearizations about $\overline{\mu}_t$, the prior mean at each step along the action sequence.

5 Posterior Belief Distribution (PBD) Algorithm

Section 4 demonstrates that, given the agent's current belief (μ_t, Σ_t), the parameters describing the distribution of posterior beliefs can be computed without enumerating the observations. In particular, the posterior covariance Σ_{t+T} can be predicted via a transfer function (Eqn. 18), while the posterior mean μ_{t+T} is normally distributed according to Eqn. 17. These results enable us to consider a subset of admissible action sequences and obtain the associated distribution of posterior beliefs without enumerating the corresponding observation sequences. We now present an algorithm, the Posterior Distribution Prediction (PBD) algorithm, that takes advantage of the results to perform forward search to much greater depths.

The PBD algorithm, shown in Algs. 1 and 2, builds upon a generic online forward search for POMDPs [17], alternating between a planning and execution phase at every iteration. During the planning phase, the next best action sequence $\hat{a}_{seq}$ is chosen, and the first action a_t of this action sequence is executed. The agent then updates its current belief according to the observation obtained, and the cycle repeats.

During the planning phase, the EXPAND() subroutine is executed recursively in a depth-first search manner. The agent first samples a number of multi-step action sequences that it can execute from its current belief. The choice of these action sequences is domain-specific. For each of these action sequences, the algorithm calculates the parameters of the agent's posterior belief, i.e., the parameters of the posterior mean distribution m_{t+T}, S_{t+T} according to Eqn. 17, as well as the posterior covariance Σ_{t+T} according to Eqn. 18. These three sets of parameters are the sufficient statistics of the distribution of beliefs shown in Fig. 3.

As discussed in Section 3, we then sample from the posterior belief distribution to instantiate beliefs for deeper forward search. Given that the covariance can be assumed constant, we perform importance sampling only on the posterior mean distribution, generating samples of posterior beliefs by associating the posterior mean samples $\{n_i\}$ with the posterior covariance Σ_{t+T}. These beliefs are then used to perform an additional layer of depth-first search, and this process repeats for a pre-determined search depth D. At the leaf nodes, a value heuristic is used to provide an estimate of being at the belief associated with the node. These values are then propagated up the tree (Alg. 2, Eqn. 12), and consists of both the instant rewards of executing the action sequence and a

Algorithm 1 Posterior Belief Distribution Algorithm

Require: b_0 : Agent's initial belief,
D : Maximum search depth,
V_h : Value heuristic function

$b_c \leftarrow b_0$
while not EXECUTIONEND() **do**
  while not PLANNINGEND() **do**
    $\hat{a}_{seq} = \text{EXPAND}(b_c, D, V_h)$
  end while
  Execute first action $\hat{a}_t$ of $\hat{a}_{seq}$
  Obtain new observation z_t
  Update current belief
    $b_c \leftarrow \tau(b_c, \hat{a}_t, z_t)$
  $t \leftarrow t + 1$
end while

	FVRS[8,5]		
	Ave rewards	Online time (s)	Offline time (s)
QMDP	12.93	0.0001	9.18
HSVI(150s)	12.89	0.035	150
RTBSS(d1)	12.97	2.13	150
PBD(d1,s100)	12.67	0.19	0

Fig. 3: FVRS results. Algorithm parameters: HSVI(# seconds), RTBSS(search depth), PBD(multi-step search depth, # samples)

Algorithm 2 EXPAND()

Require: b : Belief node to be expanded,
d : Depth of expansion under b,
V_h : Value heuristic function

if $d = 0$ **then**
  $V(b) \leftarrow V_h(b)$
  return $V(b)$
else
  $V(b) \leftarrow -\infty$
  for all $a_{seq,i} \in A_{seq}$ **do**
    $V(b, a_i) \leftarrow R_B(b, a_{seq,i})$
    Compute $m_{t+T}, S_{t+T}, \Sigma_{t+T}$ (Eqn. 17, 18)
    Sample set of posterior means $\{n_i\}$
      according to $N(m_{t+T}, S_{t+T})$
    for all n_i **do**
      $b' \leftarrow N(n_i, \Sigma_{t+T})$
      $V(b, a_{seq,i}) \leftarrow V(b, a_{seq,i}) + \gamma p(n_i | m_{t+T}, S_{t+T})$
        $\times \text{EXPAND}(\tau(b', a_{seq,i}, z), d - 1)$
      if $V(b, a_i) > V(b)$ **then**
        $V(b) \leftarrow V(b, a_{seq,i})$
        $a^\star_{seq} \leftarrow a_{seq,i}$
      end if
    end for
  end for
end if
return $V(b), a^\star_{seq}$

Monte Carlo integration of the value estimates from the sampled posterior beliefs. At the root, the algorithm chooses the action sequence with the highest expected discounted value.

5.1 Errors Induced by Linearizations

Sampling the distribution of posterior means induces an error on the expected rewards. However, error bounds can be computed using Hoeffding's inequality [7],

$$p(V_S(b, a_{seq,i}) - V^*(b, a_{seq,i}) \geq \gamma n \epsilon) \leq \exp\left(-\frac{-2n^2\epsilon^2}{n(V_{max} - V_{min})^2}\right), \tag{19}$$

where $V_S(.)$ and $V^*(.)$ are the sampled and "exact" value estimates respectively, V_{max} and V_{min} are the maximum and minimum rewards that could be obtained, and n is the number of samples. For a desired accuracy ϵ, we can recover the appropriate number of samples.

In addition, for a generic exponential family observation model, calculating the posterior mean and covariance after a single action (Eqn. 14 and 9) requires us to linearize about the prior mean to calculate the Jacobians Y_t and $\ddot{b}_t$. When determining the posterior belief distribution after a multi-step action sequence (Eqn. 17 and 18), we make the assumption that we know the future prior means at each step along the action sequence, in order to perform linearization at every step along that sequence. This assumption is an approximation, since the future mean explicitly depends on the observation sequence. As a result, the Jacobians used to calculate the Kalman gain and covariance are approximate, and the error of the approximation in the Kalman gains and covariances increases with macro-action length. Bounds on approximation errors for the EKF are known to exist [8], and in future, we plan to provide analysis of how these bounds can be used to determine the length of the macro-actions.

6 Empirical Results

We provide initial results demonstrating the performance of the PBD algorithm. Unfortunately, most existing benchmark POMDP problems do not require POMDP solvers to search deeply to generate good policies. For example, the Rocksample problem, originally proposed in [18], and the extension FieldVisionRockSample (FVRS) problem [15] both allow simple approximation techniques to perform well. In the FVRS problem, an agent explores and samples rocks in a grid world. The agent is fully aware of the position of both itself and the rocks. At each timestep, it receives a binary, noisy observation of the value of each rock, and the observation accuracy increases as the agent moves closer to the rock.

The original Rocksample and FVRS problems appear easily solvable with existing POMDP techniques such as HSVI [18] and AEMS [15]. Moving to sample the rock involves the same actions as moving to acquire more information of the rock. For these problems, a greedy policy of finding the shortest path to all the rocks is a good approximation to the optimal policy. Fig. 3 shows that our algorithm is competitive with offline solvers such as HSVI, but suggests that even a naive QMDP solver will perform well. Our PBD algorithm performs slightly poorer since it approximates the discrete state space as continuous, but the error induced, if any, is not statistically significant. For all the experiments reported in this paper, we adapted the HSVI implementation from the ZMDP software package[1], while our RTBSS implementation uses the QMDP and HSVI alpha-vectors as upper and lower bounds on the value function. HSVI ran in C++, while we ran the QMDP, RTBSS, and PBD algorithms in Matlab.

To better test the algorithms, we propose the Information Search RockSample problem (ISRS) (Fig. 4), modifying the FVRS problem in two ways. We introduce a set of beacons (shown as yellow triangles) that each correspond to a rock (shown as grey circles). Rather than the observation accuracy depend on the proximity to each rock, the agent r_t must instead move to the beacon $RB_{i,t}$ to get better information. Second, we modified the rock values $RV_{i,t}$ to each have a continuous value of between 0 and 1, rather than being a binary-valued set. The agent continues to receive a binary, noisy observation of the value of each rock z_t, according to:

$$O: \quad p(z_{i,t}|RV_{i,t}=m, r_t, RB_{i,t}) = \begin{cases} 0.5+(m-0.5)2^{\frac{-4\|r_t-RB_{i,t}\|_2}{D_0}} & z_{i,t}=1 \\ 0.5-(m-0.5)2^{\frac{-4\|r_t-RB_{i,t}\|_2}{D_0}} & z_{i,t}=0 \end{cases} \tag{20}$$

where D_0 is a tuning parameter that controls how quickly the accuracy of the observations decrease with greater distance between the agent and the beacon. The agent's sensor can therefore be modeled as a Bernoulli observation model, and the observations are assumed to be more accurate than an unbiased coin. The agent obtains a large reward if it samples a rock that has a high value, but incurs a large cost for wrongly doing so. Small costs are incurred for moving around the environment.

6.1 Comparison with other algorithms

The ISRS problem was used to compare the PBD algorithm against a fast upper bound (QMDP), a point-based offline value iteration technique (HSVI) and an online forward search algorithm (RTBSS). Macro-actions were generated for the PBD algorithm by sampling robot poses in the grid world and computing the sequence of actions necessary to reach the sampled pose. Fig. 4 and 6 illustrate the fundamentally different policies generated by our PBD algorithm, relative to existing techniques. In Fig. 4, the RTBSS solver obtains an observation suggesting that rock 1 is valuable. Unfortunately, its inability to search beyond depth 1 causes it to fail to realize that good information about rock 1 can be obtained by making a short detour. Instead, it heads towards rock 1, but because subsequent observations are noisy, due to distance from the beacon, the solver is unable to commit to sampling the rock and is stuck at a local minimum. In contrast, the PBD solver (Fig. 6) is able to evaluate the value of making a detour when seeking information about rock 5. Subsequently, the agent moves to the region with multiple beacons, concludes that there is little value left to sample, and exits the problem.

Fig. 5 reports the performance of the different algorithms tested on the ISRS problem. The problem was initialized with different hidden values of the rocks, and 20 trials were conducted for each simulation. The different algorithms were also initialized with different offline processing time (HSVI),

[1] ZMDP Software for POMDP and MDP Planning. http://www.cs.cmu.edu/∼trey/zmdp/

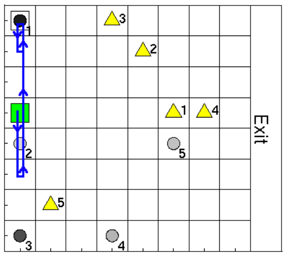

Fig. 4: ISRS world. Blue line indicates example RTBSS policy

	ISRS[8,5] (2304s, 5a, 32o)		
	Ave rewards	Online time(s)	Offline time(s)
QMDP	9.43	0.0001	10.4
HSVI(150s)	8.06	0.016	150
HSVI(500s)	5.31	0.040	500
HSVI(4000s)	10.46	0.17	4000
RTBSS(d1)	28.08	2.88	150
RTBSS(d2)	30.55	125.2	50
PBD(d1,s300)	45.47	0.94	0
PBD(d2,s30)	45.38	19.53	0

Fig. 5: Performance of POMDP solvers in ISRS problem

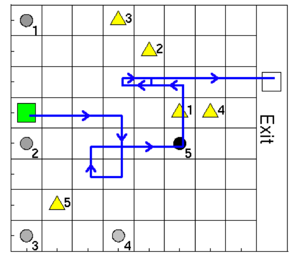

Fig. 6: Example PBD policy

different search depth (RTBSS), and different posterior mean samples (PBD). The results demonstrate that the PBD algorithm is able to perform significantly better than other existing algorithms, thereby demonstrating the value of searching deeper.

7 Conclusion and Discussion

We have demonstrated the ability to perform belief updates for multi-step action sequences in closed form for models that have specific parametric representations. When the models are linear-Gaussian, our expression for the posterior belief distribution after a multi-step action sequence is exact. For observation models that are members of the exponential family, the belief update can be approximated by using a linearized variant of the Kalman filter. By being able to compute the agent's distribution of posterior beliefs after multi-step action sequences, we developed an algorithm that can perform deeper forward search when planning under uncertainty. While we may sacrifice an exhaustive search over all action sequences up of a particular length, our algorithm enables us to search to an equivalent length of multi-step action sequences. This effectively allows us to search deeper, allowing us to discover policies that would otherwise not have been found with a shallow search depth.

In contrast to most POMDP solvers, our algorithm assumes that the agent's belief and observation models are representable as particular classes of parametric distributions. This necessary implies that our algorithm is not a generic POMDP solver. Nevertheless, we believe that not only do these classes of probability distributions represent a wide variety of belief distributions, but also that continuous, parametric representations are our only solution for avoiding the curse of dimensionality [2], especially as we seek to solve larger POMDP problems in future. In addition, the notion of predicting the distribution of posterior beliefs should be extendable to a broader class of belief representations. Exponential family distributions that are conjugate priors to exponential family observation models are attractive candidates for representing the belief space, since the posterior belief after a belief update belongs to the same parametric class as the prior belief.

References

[1] O.E. Barndorff-Nielsen. Information and exponential families in statistical theory. *Bull. Amer. Math. Soc. 1 (1979), 667-668.*, 273(0979), 1979.

[2] A. Brooks, A. Makarenko, S. Williams, and H. Durrant-Whyte. Parametric POMDPs for planning in continuous state spaces. *Robotics and Autonomous Systems*, 54(11):887–897, 2006.

[3] E. Brunskill, L. Kaelbling, T. Lozano-Perez, and N. Roy. Continuous-state POMDPs with hybrid dynamics. In *Symposium on Artificial Intelligence and Mathematics*, 2008.

[4] T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley-Interscience, 2006.

[5] J. Durbin and SJ Koopman. Time series analysis of non-Gaussian observations based on state space models from both classical and Bayesian perspectives. *Journal of the Royal Statistical Society: Series B (Methodological)*, 62(1):3–56, 2000.

[6] M. Hauskrecht. Value-function approximations for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, 13(2000):33–94, 2000.
[7] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, pages 13–30, 1963.
[8] B.F. La Scala, R.R. Bitmead, and M.R. James. Conditions for stability of the extended Kalman filter and their application to the frequency tracking problem. *Mathematics of Control, Signals, and Systems (MCSS)*, 8(1):1–26, 1995.
[9] D. McAllester and S. Singh. Approximate planning for factored POMDPs using belief state simplification. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 409–416, 1999.
[10] S. Paquet, L. Tobin, and B. Chaib-draa. An online POMDP algorithm for complex multiagent environments. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 970–977. ACM New York, NY, USA, 2005.
[11] J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *International Joint Conference on Artificial Intelligence*, volume 18, pages 1025–1032, 2003.
[12] J.M. Porta, N. Vlassis, M.T.J. Spaan, and P. Poupart. Point-based value iteration for continuous POMDPs. *The Journal of Machine Learning Research*, 7:2329–2367, 2006.
[13] S. Prentice and N. Roy. The Belief Roadmap: Efficient Planning in Linear POMDPs by Factoring the Covariance. In *Proceedings of the 13th International Symposium of Robotics Research (ISRR)*, 2007.
[14] M.L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, Inc. New York, NY, USA, 1994.
[15] S. Ross and B. Chaib-draa. AEMS: An Anytime Online Search Algorithm for Approximate Policy Refinement in Large POMDPs. In *Proceedings of The 20th Joint Conference in Artificial Intelligence (IJCAI 2007), Hyderabad, India*, 2007.
[16] S. Ross, B. Chaib-draa, and J. Pineau. Bayesian reinforcement learning in continuous POMDPs with application to robot navigation. In *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*, pages 2845–2851, 2008.
[17] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa. Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research (JAIR)*, 32:663–704, 2008.
[18] T. Smith and R. Simmons. Point-based POMDP algorithms: Improved analysis and implementation. In *Proc. Uncertainty in Artificial Intelligence*, 2005.
[19] E.J. Sondik. The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. *Operations Research*, 26(2):282–304, 1978.
[20] G. Theocharous and L.P. Kaelbling. Approximate planning in POMDPs with macro-actions. *Advances in Neural Processing Information Systems*, 17, 2003.
[21] R. Washington. BI-POMDP: Bounded, incremental partially-observable Markovmodel planning. In *Proceedings of the 4th European Conference on Planning (ECP)*, pages 440–451. Springer, 1997.
[22] M. West, P.J. Harrison, and H.S. Migon. Dynamic generalized linear models and Bayesian forecasting. *Journal of the American Statistical Association*, pages 73–83, 1985.

Appendix A: Exponential Family Kalman Filter

Building on statistical economics research for time-series analysis of non-Gaussian observations [5], we present the Kalman filter equivalent for systems with linear-Gaussian state-transitions and observation models that belong to the exponential family of distributions.

The state-transition and observation models can be represented as follow:

$$s_t = A_t s_{t-1} + B_t u_t + \varepsilon_t, \qquad s_{t-1} \sim N(\mu_{t-1}, \Sigma_{t-1}), \qquad \varepsilon_t \sim N(0, R_t) \tag{21}$$

$$p(z_t|\theta_t) = \exp(z_t^T \theta_t - b_t(\theta_t) + \kappa_t(z_t)), \qquad \theta_t = W(s_t) \tag{22}$$

For the state-transition model, s_t is the system's hidden state, u_t is the control actions, A_t and B_t are the linear transition matrices, and ϵ_t is the state-transition Gaussian noise with covariance R_t.

The observation model belongs to the exponential family of distributions. θ_t and $b_t(\theta_t)$ are the canonical parameter and normalization factor of the distribution, and $W(.)$ maps the states to canonical parameter values. $W(.)$ depends on the particular member of the exponential family. For ease of notation, we let

$$\beta_t(z_t|\theta_t) = -\log p(z_t|\theta_t) = -z_t^T \theta_t + b_t(\theta_t) + \kappa_t(z_t) \tag{23}$$

Following the traditional Kalman filter, the process update can be written as

$$\overline{\mu}_t = A_t \mu_{t-1} + B_t u_t, \qquad \overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \tag{24}$$

where $\overline{\mu}_t$ and $\overline{\Sigma}_t$ are the mean and covariances of the posterior belief after the process update but before the measurement udpate. For the measurement update, we seek to find the conditional mode

$$\begin{aligned}
\mu_t &= \arg\max_{s_t} p(s_t|z_t) && (25)\\
&= \arg\max_{s_t} p(z_t|s_t)\overline{bel}(s_t) \qquad \text{(Bayes rule)} && (26)\\
&= \arg\max_{s_t} p(z_t|\theta_t)\overline{bel}(s_t) && (27)\\
&= \arg\max_{s_t} \exp(-J_t), \qquad \text{where } J_t = -\log p(z_t|\theta_t) + \frac{1}{2}(s_t - \overline{\mu}_t)^T \overline{\Sigma}_t^{-1}(s_t - \overline{\mu}_t) && (28)
\end{aligned}$$

$$\Rightarrow \qquad 0 = \frac{\partial J_t}{\partial s_t}\Big|_{s_t=\mu_t} = \frac{\partial \beta_t(z_t, \theta_t)}{\partial \theta_t}\frac{\partial \theta_t}{\partial s_t} + \overline{\Sigma}_t^{-1}(\mu_t - \overline{\mu}_t), \tag{29}$$

Taking the derivative of $\theta_t = W(s_t)$ about the prior mean $\overline{\mu}_t$, we let

$$Y_t = \frac{\partial W(s_t)}{\partial s_t}\Big|_{s_t=\overline{\mu}_t} \tag{30}$$

Similarly, performing Taylor expansion on $\frac{\partial \beta_t(z_t|\theta_t)}{\partial \theta_t}$ about $\overline{\theta}_t = W(\overline{\mu}_t)$,

$$\begin{aligned}
\frac{\partial \beta_t(z_t|\theta_t)}{\partial \theta_t} &= \frac{\partial \beta_t(z_t|\theta_t)}{\partial \theta_t}\Big|_{\theta_t=\overline{\theta}_t} + \frac{\partial^2 \beta_t(z_t|\theta_t)}{\partial \theta_t \partial \theta_t^T}\Big|_{\theta_t=\overline{\theta}_t}(\theta_t - \overline{\theta}_t) && (31)\\
\frac{\partial \beta_t(z_t|\theta_t)}{\partial \theta_t} &= \dot{\beta}_t + \ddot{\beta}_t(\theta_t - \overline{\theta}_t) && (32)\\
\text{where} \qquad \dot{\beta}_t &= \frac{\partial}{\partial \theta_t}(-z_t^T \theta_t + b_t(\theta_t) - \kappa_t(z_t))\Big|_{\theta_t=\overline{\theta}_t}, \qquad \text{(Eqn. 23)} && (33)\\
&= \frac{\partial b_t(\theta_t)}{\partial \theta_t}\Big|_{\theta_t=\overline{\theta}_t} - z_t && (34)\\
\dot{\beta}_t &= \dot{b}_t - z_t && (35)\\
\text{and} \qquad \ddot{\beta}_t &= \frac{\partial^2 \beta_t(z_t|\theta_t)}{\partial \theta_t \partial \theta_t^T}\Big|_{\theta_t=\overline{\theta}_t}(\theta_t - \overline{\theta}_t) && (36)\\
\ddot{\beta}_t &= \ddot{b}_t && (37)
\end{aligned}$$

Plugging Equations 35 and 37 into Equation 32, and then into Equation 29,

$$Y_t\ (\dot{b}_t - z_t + \ddot{b}_t(\theta_t - \overline{\theta}_t)) = -\ \overline{\Sigma}_t^{-1}(\mu_t - \overline{\mu}_t) \tag{38}$$

$$Y_t\ \ddot{b}_t(\ddot{b}_t^{-1}(\dot{b}_t - z_t) - \overline{\theta}_t + \theta_t) = -\ \overline{\Sigma}_t^{-1}(\mu_t - \overline{\mu}_t) \tag{39}$$

$$Y_t\ \ddot{b}_t((\overline{\theta}_t - \ddot{b}_t^{-1}(\dot{b}_t - z_t)) - \theta_t) = \overline{\Sigma}_t^{-1}(\mu_t - \overline{\mu}_t) \tag{40}$$

$$Y_t\ \ddot{b}_t(\tilde{z}_t - W(s_t)) = \overline{\Sigma}_t^{-1}(\mu_t - \overline{\mu}_t) \tag{41}$$

where $\tilde{z}_t = (\overline{\theta}_t - \ddot{b}_t^{-1}(\dot{b}_t - z_t))$ is the projection of the observation onto the parameter space of the exponential family distribution, and is independent of s_t. In Equation 41 we substituted θ_t using Equation 22.

Mean Update

Using Equation 41 and substituting μ_t for s_t,

$$\overline{\Sigma}_t^{-1}(\mu_t - \overline{\mu}_t) = Y_t\ \ddot{b}_t(\tilde{z}_t - W(\mu_t)) \tag{42}$$

$$= Y_t\ \ddot{b}_t(\tilde{z}_t - W(\mu_t)) + W(\overline{\mu}_t) - W(\overline{\mu}_t) \tag{43}$$

$$= Y_t\ \ddot{b}_t(\tilde{z}_t - W(\overline{\mu}_t)) - Y_t\ \ddot{b}_t(W(\mu_t) - W(\overline{\mu}_t)) \tag{44}$$

Linearizing $W(s_t)$ about $\overline{\mu}_t$,

$$W(s_t) = W(\overline{\mu}_t) + W'(s_t)_{s_t = \overline{\mu}_t}(s_t - \overline{\mu}_t) \tag{45}$$

$$= W(\overline{\mu}_t) + Y_t\ (\mu_t - \overline{\mu}_t) \tag{46}$$

$$\Rightarrow \qquad \overline{\Sigma}_t^{-1}(\mu_t - \overline{\mu}_t) = Y_t\ \ddot{b}_t(\tilde{z}_t - W(\overline{\mu}_t)) - Y_t\ \ddot{b}_t Y_t\ (\mu_t - \overline{\mu}_t) \tag{47}$$

$$Y_t\ \ddot{b}_t(\tilde{z}_t - W(\overline{\mu}_t)) = (\overline{\Sigma}_t^{-1} + Y_t\ \ddot{b}_t Y_t\ )(\mu_t - \overline{\mu}_t) \tag{48}$$

$$= \Sigma_t^{-1}(\mu_t - \overline{\mu}_t) \tag{49}$$

$$\Rightarrow \qquad \mu_t - \overline{\mu}_t = \Sigma_t Y_t\ \ddot{b}_t(\tilde{z}_t - W(\overline{\mu}_t)) \tag{50}$$

where $\Sigma_t Y_t\ \ddot{b}_t = \tilde{K}_t$ is the Kalman gain for non-Gaussian exponential family distributions. Via a standard transformation, the Kalman gain can be written in terms of covariances other than Σ_t,

$$\tilde{K}_t = \overline{\Sigma}_t Y_t\ (Y_t \overline{\Sigma}_t Y_t + \ddot{b}_t^{-1})^{-1} \tag{51}$$

$$\text{and} \qquad \mu_t = \overline{\mu}_t + \tilde{K}_t(\tilde{z}_t - W(\overline{\mu}_t)) \tag{52}$$

Covariance Update

Given a Gaussian posterior belief, $\frac{\partial^2 J}{\partial s_t^2}$ is the inverse of the covariance of the agent's belief

$$\Sigma_t^{-1} = \frac{\partial^2 J}{\partial s_t^2} \tag{53}$$

$$= \frac{\partial}{\partial x}(\overline{\Sigma}_t^{-1}(s_t - \overline{\mu}_t) - Y_t\ \ddot{b}_t(\tilde{z}_t - W(s_t))) \tag{54}$$

$$= \overline{\Sigma}_t^{-1} + Y_t\ \ddot{b}_t Y_t \tag{55}$$

$$\Rightarrow \Sigma_t = (\overline{\Sigma}_t^{-1} + Y_t\ \ddot{b}_t Y_t)^{-1} \tag{56}$$