Optimal Continuous State POMDP Planning with Semantic Observations

Luke Burks and Nisar Ahmed*

***Abstract*— This work develops novel strategies for optimal planning with semantic observations using continuous state Partially Observable Markov Decision Processes (CPOMDPs). We propose two major innovations to Gaussian mixture (GM) CPOMDP policy approximation methods. While these state of the art methods have many theoretically nice properties, they are hampered by the inability to efficiently represent and reason over hybrid continuous-discrete probabilistic models. The first major innovation is the derivation of closed-form variational Bayes (VB) GM approximations of PBVI Bellman policy backups, using softmax models of continuous-discrete semantic observation probabilities. The second major innovation is a new clustering-based technique for mixture condensation that scales well to very large GM policy functions and belief functions. Simulation results for a target search and interception task with binary semantic observations show that the GM policies resulting from these innovations are more effective than those produced by other state of the art GM approximations, but require significantly less modeling overhead and runtime cost.**

I. Introduction

Many applications of planning under uncertainty require autonomous agents to reason over outcomes in continuous dynamical environments using imprecise but readily available semantic observations. For instance, in search and tracking applications, autonomous robots must be able to efficiently reacquire and localize mobile targets that can potentially remain out of view for long periods of time. Planning algorithms must generate vehicle trajectories that optimally exploit 'detection' and 'no detection' data from onboard sensors [3], [9], as well as semantic natural language observations that can be provided by human supervisors [2]. However, for such applications, it remains quite challenging to achieve tight optimal integration of vehicle motion planning with non-linear sensing and non-Gaussian state estimation in large continuous dynamic problem domains.

In recent years, a variety of techniques based on *partially observable Markov decision processes (POMDPs)* have been developed to address these issues. These include methods which (rather than discretizing the continuous state space) preserve the continuous dynamical nature of the problem through suitable function approximations. Of particular interest here are approximations based on *Gaussian mixture (GM)* models, which can flexibly represent complex policy functions and non-Gaussian probability density functions (pdfs) [4], [5]. These techniques theoretically enable efficient closed-form manipulation and recursions for producing compact (yet accurate) optimal POMDP policy approximations. However, these state of the art methods suffer from two major drawbacks when dealing with semantic observations. Firstly, they rely on expensive and non-scalable hybrid probabilistic observation likelihood models for capturing the relationship between observed discrete semantic sensor data and unknown continuous states. Secondly, these methods rely on expensive GM condensation techniques for maintaining computational tractability. These issues greatly increase the modeling and computational effort required for implementation, and thus significantly limit the practical applicability and scalability of GM-based POMDP approximations to continuous state decision-making problems.

This work presents two technical innovations to directly address these issues. The first novel contribution is an efficient variational Bayes (VB) GM POMDP policy approximation method that allows semantic sensor observations to be accurately yet inexpensively modeled by generalized softmax likelihood models (which otherwise lead to intractable policy and pdf updates for continuous POMDPs). The second novel contribution is the development of a fast and scalable two-stage GM condensation technique for large mixtures. Finally, numerical simulation results of the proposed approximation methods are provided on a dynamic target search application, showing favorable comparisons to the existing state-of-the-art approximation methods.

II. Background and Preliminaries

Formally, a POMDP is described by the 7-tuple $(S, A, T, R, \Omega, O, \gamma)$, where: S is a set of states s; A is a set of $|A|$ discrete actions a; T is a discrete time probabilistic transition mapping from state s to state s' given some a; R is the immediate reward mapping over (s, a) pairs; Ω is a set of observations o; O is the likelihood mapping from states to observations; and $\gamma \in [0, 1]$ is a discount factor. An agent whose decision making process is modeled by a POMDP seeks to maximize a utility function defined by the expected future discounted reward: $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$, where $s_t \in S$ is the state at discrete time t, and $a_t \in A$. The expectation operator $\mathbb{E}[\cdot]$ reflects that the agent lacks full knowledge of s_t. It must instead rely on the noisy process model T and observation model O to update a Bayesian belief function $b(s_t) = p(s_t|a_{1:t}, o_{1:t})$, which summarizes all available information for reasoning about present and possible future states. A optimal decision making policy $\pi(b(s_t)) \rightarrow a_t$ must therefore be found for any possible belief $b(s_t)$. Since POMDPs are equivalent to Markov decision processes (MDPs) over beliefs $b(s_t)$, exact policies are impossible to compute for all but the simplest problems.

One well-known family of techniques for computing approximate POMDP policies offline is Point-Based Value

*Authors are with the Smead Aerospace Engineering Sciences Department, University of Colorado Boulder, Boulder CO 80309, USA. E-mail: [luke.burks;nisar.ahmed]@colorado.edu.

Iteration (PBVI) [7]. These methods approximate π at a finite set of 'typical' sample beliefs $\mathcal{B}_0 = \{b_1(s), ..., b_{N_B}(s)\}$, for which explicit finite-horizon Bellman equation recursions can be performed to obtain locally optimal actions in the neighborhood of each $b_i(s)$, $i = 1, ..., N_B$. When S is a set of discrete states with N possible outcomes, then $b(s) \in \mathbb{R}^N$ such that $\sum_{s=1}^{N} b(s) = 1$. In this case, PBVI policies are represented by a set Γ of N_α vectors $\alpha \in \mathbb{R}^N$. The α vectors mathematically represent hyperplanes that encode value functions for taking particular actions at a given belief. The action a recommended by the policy for a given $b(s) \in \mathbb{R}^N$ is found as the action associated with $\arg\max_{\alpha \in \Gamma} < \alpha, b(s) >$, where $< \cdot >$ is the inner product. A number of methods exist for generating typical sample beliefs, e.g. starting with a large set of $b_i(s)$ sampled from the reachable belief space by random simulation [7] (as in this work), or propagating a small initial belief set in between recursive Bellman updates for α vector computations to approximate optimal reachable belief sets [6].

When s is a continuous random vector such that $s \in \mathbb{R}^N$ with support $\mathcal{S}(S)$, it is more natural to represent $b(s)$ as a probability density function (pdf), where $\int_{\mathcal{S}(S)} b(s)ds = 1$. In such cases, continuous state POMDPs (CPOMDPs) can be formulated by specifying T, R, O and $\alpha(s)$ as suitable continuous functions over s. Although $b(s)$ can sometimes be represented by simple parametric models such as Gaussian pdfs [1], $b(s)$ is in general analytically intractable for arbitrary T and O models (e.g. nonlinear dynamics, semantic sensor observations). Therefore, $b(s)$ must also be approximated to derive a suitable set Γ of $\alpha(s)$, such that the (approximate) optimal PBVI policy $\pi(b(s))$ is defined by the action associated with $\arg\max < \alpha(s), b(s) >$.

A. Gaussian Mixture CPOMDPs

Finite Gaussian mixture (GM) models provide a very general and flexible way to approximate arbitrary functions $f(s)$ of interest for CPOMDPs, where

$$f(s) = \sum_{m=1}^{M} w_m \phi(s|\mu_m, \Sigma_m) \tag{1}$$

is a GM defined by M weights $w_m \in \mathbb{R}^0$,means $\mu_m \in \mathbb{R}^N$, and symmetric psd covariances $\Sigma_m \in \mathbb{R}^{N \times N}$ for the multivariate normal component pdf ('mixand') $\phi(s|\mu_m, \Sigma_m)$, such that $\sum_{m=1}^{M} w_m = 1$ to ensure normalization when $f(\cdot)$ represents a pdf (this condition need not apply otherwise). Ref. [5] showed that if A describes a discrete action space with $T = p(s'|s, a)$, $O = p(o|s')$, and $R = r_a(s)$ all specified by Gaussian or finte GM functions over s, respectively,

$$T = \phi(s'|s + \Delta(a), \Sigma^a) \tag{2}$$

$$O = \sum_{l} w_l \phi(s'|s_l, \Sigma_l) \tag{3}$$

$$R = \sum_{i} w_i \phi_i(s|\mu_i^a, \Sigma_i^a) \tag{4}$$

then PBVI approximations to $\pi(b(s))$ can be found based on closed-form GM Bellman recursions for a finite set of GM functions $\alpha(s)$ defined over some initial set of GM beliefs $b(s)$. Note that $r_a(s)$ is generally a mixture of *unnormalized* Gaussians, with possibly negative mixture weights and such that $\int_{\mathcal{S}(s)} r_a(s)ds \neq 1$. This allows the CPOMDP to flexibly penalize certain configurations of continuous states with discrete actions, and thus discourage undesirable agent behaviors. However, T must obey the usual constraints for continuous pdfs, such that $\int_{\mathcal{S}(s')} p(s'|s, a)ds' = 1$. The observation likelihood must also obey $\sum_o p(o|s') = 1$ for any given s', where o is assumed to be a discrete random variable describing a semantic observation. As such, $p(o|s')$ can be a strictly positive but unnormalized GM (i.e. with strictly positive weights, but whose components over s' do not individually integrate to unity), in order to model how a discrete conditional probability distribution over o varies with the latent state vectors s'.

If $b(s)$ can always be modeled as a finite GM with J terms,

$$b(s) = \sum_{j}^{J} w_j \phi(s|\mu_j, \Sigma_j),$$

then it is possible to arrive at a set $\Gamma = \{\alpha^1, \alpha^2, ...\alpha^{N_\alpha}\}$, where $N_\alpha \leq N_B$, of $\alpha(s)$ functions for an n-step lookahead decision starting from $b(s)$, such that

$$\alpha_n^i(s) = \sum_{k=1}^{M} w_k^i \phi(s|\mu_k^i, \Sigma_k^i) \quad (\alpha_n^i \in \Gamma_n)$$

and the optimal value function $V_n^*(b(s))$ at $b(s)$ is approximately given by

$$V^*(b(s)) \approx argmax_{\alpha_n^i} < \alpha_n^i, b(s) > \tag{5}$$

$$< \alpha_n^i, b(s) >=$$

$$\int_s \left[\sum_{k}^{M} w_k^i \phi(s|\mu_k^i, \Sigma_k^i)\right] \left[\sum_{j}^{J} w_j \phi(s|\mu_j, \Sigma_j)\right] ds \tag{6}$$

$$= \sum_{k,j}^{M \times J} w_k^i w_j \phi(\mu_j|\mu_k^i, \Sigma_j + \Sigma_k^i) \int_s \phi(s|c_1, c_2)ds \tag{7}$$

$$= \sum_{k,j}^{M \times J} w_k^i w_j \phi(\mu_j|\mu_k^i, \Sigma_j + \Sigma_k^i) \tag{8}$$

$$c_2 = [(\Sigma_k^i)^{-1} + (\Sigma_j)^{-1}]^{-1}$$

$$c_1 = c_2[(\Sigma_k^i)^{-1}\mu_k^i + (\Sigma_j)^{-1}\mu_j]$$

(which follows from the fact that the product of two Gaussian functions is another Gaussian function). The n-step lookahead horizon approximation is commonly used in PBVI approaches, where n is large enough such that the value function V_n^* does not change appreciably (and thus starts converges closely to the infinite horizon V^*).

The $\alpha_n^i \in \Gamma_n$ functions are computed using n-step policy rollouts, starting from N_B different initial GM beliefs $\mathcal{B}_0 = \{b_1(s), ..., b_{N_B}(s)\}$. In each backup step, for each $b_j(s) \in \mathcal{B}_0$, each α_{n-1}^i function's value is updated via the so-called 'Bellman backup' equations, which perform point-wise value iteration to capture the effects of all possible observations and actions on the accumulated expected reward for future

time steps $0, ..., n$. These lead to the recursions

$$\alpha^i_{a,o}(s) = \int_{s'} \alpha^i_{n-1}(s')p(o|s')p(s'|s,a)ds', \quad (9)$$

$$\alpha^i_n(s) = r_a(s) + \gamma \sum_o \arg\max_{\alpha^i_{a,o}}(<\alpha^i_{a,o}, b>), \quad (10)$$

where $\alpha^i_{a,o}(s)$ is an intermediate function corresponding to a value for a given action-observation pair (a, o) at step n, and $\alpha^i_n(s)$ is the discounted marginalization over all observations of the intermediate function that maximizes the belief being backed up, summed with the reward function. The action then associated with each α^i_n is the one which maximized the value marginalized over observations. Due to the choice of Gaussian $p(s'|s,a)$, GM $p(o|s')$ and GM $r_a(s)$ functions, the Bellman backups yield closed-form GM functions for α^i_n. Since the GM function for $r_a(s)$ can have negative weights and values, it follows that each GM function $\alpha^i_n(s)$ can also take on negative weights and values.

A nice property of this GM formulation is that it can theoretically scale well to continuous state spaces where $N \geq 2$, and naturally handles highly non-Gaussian beliefs $b(s)$ stemming from non-linear/non-Gaussian continuous state process and observation models in a deterministic manner. In contrast to approximations that discretize S to transform the CPOMDP into a standard discrete state POMDP (and thus scale badly for large N), the complexity of the CPOMDP policy (i.e. the required number of mixture terms for each $\alpha^i_n(s)$) depends only on the complexity of the dynamics of $b(s)$, rather than the number of continuous states N. Furthermore, since the Bellman backup equations can be performed entirely offline using a set of 'typical' initial beliefs $\mathcal{B}_0$, the resulting policy induced by the final set of $\alpha^i_n(s)$ functions can be quickly and easily computed online: as the agent obtains new beliefs $b(s) \rightarrow b(s')$ over time via the standard Bayes' filter equations,

$$b(s') \propto p(o|s') \int_{S(s)} p(s'|a,s)b(s)ds, \quad (11)$$

the optimal action a to take for $b(s')$ is the one associated with the $\alpha^i_n \in \Gamma_n$ satisfying $\arg\max_{\alpha^i_n} < (b(s'), \alpha^i_n) >$.

B. Limitations for Hybrid Continuous-Discrete Reasoning

If $o \in \Omega$ describes a categorical/discrete-valued semantic observation with $N_o = |\Omega|$ possible values, then the observation likelihood function $O = p(o|s)$ must describe a valid hybrid (continuous-discrete) probability distribution, such that $\sum_o p(o|s) = 1 \ \forall s \in \mathcal{S}(S)$. The current state-of-the-art is to model O by an unnormalized GM for each possible outcome o [5], $p(o|s) \approx \sum_{l_o=1}^{L_o} w_o \phi(s|\mu_{l_o}, \Sigma_{l_c})$, such that $\sum_o p(o|s) \approx 1$ everywhere. Although this preserves the closed-form updates required for PBVI, such models are often very difficult and labor intensive to specify. In particular, for $N \geq 2$, L_o must be very large for each possible o to ensure that the normalization requirement is satisfied for all s and that desired probabilities in $p(o|s)$ are modeled accurately. This effectively turns $p(o|s)$ into a 'soft discretization' model based on GMs and severely restricts the scalability of GM policy approximation.

Another related and more general problem is the fact that the GM multiplication and summation operations in the Bellman recursions defined above lead to a drastic increase in the number of resulting GM components, cf. eq. (8). GM condensation methods are thus needed to control the size of $\alpha^i_n(s)$ between backup steps for offline policy approximation and between Bayes' filter updates for $b(s)$ in online policy evaluation. To this end, refs. [5], [4] propose different general methods for condensing GM functions, although in principle any number of GM merging algorithms developed in the target tracking and data fusion literature could also be applied [8], [11]. However, for large-scale problems such as dynamic target search and tracking, it is not uncommon for offline Bellman backups and online policy evaluations to rapidly produce hundreds or even thousands of new mixands in just one backup step or Bayes' filter prediction/measurement update. As discussed in Sec. III.B, existing GM merging methods tend to be computationally expensive and slow for such large mixtures. The use of dense unnormalized GM models for semantic likelihoods O exacerbates this issue and introduces additional errors in the policy approximation if normalization is not guaranteed for all $s \in \mathcal{S}(S)$. These issues significantly raises the computational cost of offline policy approximation and online policy evaluation.

C. Target Search Example with Semantic Observations

For concreteness, consider an $N = 2$ CPOMDP in which an autonomous robot 'cop' attempts to localize and catch a mobile 'robber', where both are constrained to move along parallel linear paths (see Figure 1). Here, $S = \mathbb{R} \times \mathbb{R}$ consists of two bounded continuous random variables at each discrete time step t, $s = [Cop, Rob]^T$, $Cop \in (0, 5)$, $Rob \in (0, 5)$. The robber executes a Gaussian random walk: $p(Rob_{t+1}) = \phi(Rob_{t+1}|Rob_t, 0.5)$. The cop must choose from among 3 noisy actions $A = \{\text{left}, \text{right}, \text{stay}\}$ to define a movement direction, such that: $p(Cop_{t+1}|Cop_t, \text{left}) = \phi(Cop_{t+1}|Cop_t - 0.5, 0.01)$, $p(Cop_{t+1}|Cop_t, \text{right}) = \phi(Cop_{t+1}|Cop_t + 0.5, 0.01)$, and $p(Cop_{t+1}|Cop_t = Cop_{t+1}, \text{stay}) = 1$. The cop is rewarded for remaining within a set distance of the robber's position, and penalized otherwise,

$$r(|Rob_t - Cop_t| \leq 0.5) = 3,$$
$$r(|Rob_t - Cop_t| > 0.5) = -1.$$

The cop obtains simple binary semantic observations o_t from a noisy sensor (e.g. human supervisor or onboard visual detector), where $o_t \in \{$'robber detected', 'robber not detected'$\}$.

Figure 2 (a) shows unnormalized GM models for the semantic 'detection' and 'no detection' likelihoods, which are respectively parameterized by 8 and 200 isotropic Gaussian components. These models follow the specification of $O = p(o|s')$ suggested by refs. [4] and [5], and require 624 parameters total. Since it is expected that the cop will gather mostly 'no detection' observations of the robber in a typical scenario, it is clear from eq. (9) that the number of mixing

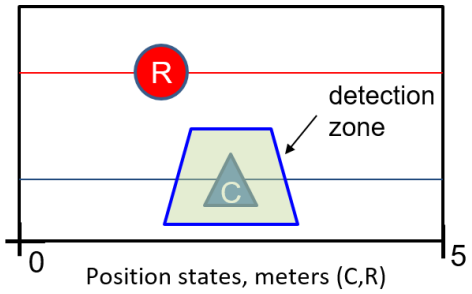
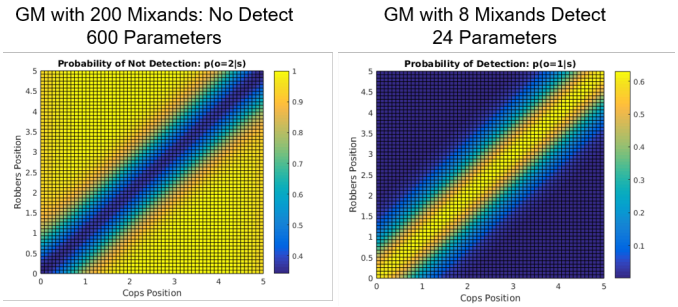
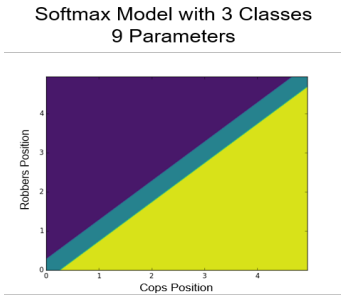


Fig. 1: 'Cop and Robber' target search problem.



(a) Gaussian Mixture Likelihood Function



(b) Softmax likelihood model.

Fig. 2: Semantic 'detection' and 'no detection' likelihood models with parameter counts.

components for $\alpha^i_{a,o}(s)$ will grow by a factor of at least 600 on a majority of the intermediate Bellman backup steps for offline policy approximation. Likewise, eq.(11) implies that the number of mixture components for $b(s')$ will grow by a factor of at least 600 on each update of the Bayes' filter whenever the target is not detected. This example shows that, even for relatively simple and small problems, unnormalized GM likelihood models such as the ones in Fig. 2 (a) are not particularly convenient or conducive to approximating or evaluating optimal policies for continuous state spaces.

III. Variational CPOMDPs and Cluster-based GM Condensation

As discussed in [2] and mentioned in [4], semantic observation likelihoods are ideally modeled by self-normalizing functions like the softmax model,

$$p(o|s) = \frac{\exp(w_o^T s + b_o)}{\sum_{c=1}^{N_o} \exp(w_c^T s + b_c)}$$

where $w_1, ..., w_{N_o} \in \mathbb{R}^N$ and $b_1, ..., b_{N_o}$ are the vector weight parameters and scalar bias parameters for each categorical outcome o given s. In addition to ensuring $\sum_o p(o|s) = 1 \ \forall s \in \mathcal{S}(S)$, softmax functions require relatively few parameters compared to GM likelihoods, and scale well to higher dimensional spaces. Figure 2 (b) shows how the cop's semantic observation likelihood can be easily modeled with a softmax function featuring 3 semantic categorical classes (two of which collectively represent the 'no detect' observation in the blue and yellow regions via the generalized 'multimodal softmax' (MMS) formulation [10]). Unlike the GM likelihood function approxmation, the softmax model only requires 9 parameters.

In general, softmax parameters can be easily synthesized to conform to a priori sensing geometry information and quickly calibrated/tuned with training data [10]. However, since the product of a Gaussian function and softmax function is analytically irreducible, the use of softmax functions for $p(o|s')$ breaks the recursive nature of the α function updates for GM-based PBVI approximations.

This section describes how this issue can be addressed in a novel way using a variational Bayes (VB) inference approximation. The VB approximation allows the product of each Gaussian term within a GM and a softmax likelihood function to be approximated as a GM, thus restoring closed-form recursivity for GM α approximations while keeping the resulting number of mixands in the result to a minimum. Note that this VB approximation is inspired by the use of a very similar technique developed in [2] for approximating eq. (11) for the problem of pure Bayesian filtering when $b(s)$ is a GM pdf and $p(o|s')$ is a softmax model. Hence, the approximate VB inference technique is generalized here to the dual problems of Bayesian filtering and optimal action selection under uncertainty for CPOMDPs.

Since the number of GM mixands for α functions and/or $b(s)$ can still become significantly large over iterations/time even with the VB approximation, this section also describes a novel GM condensation algorithm to help reduce the computational overhead and enable faster policy computation.

A. Variational PBVI for Softmax Semantic Likelihoods

To use softmax models for $p(o|s)$ in the GM-based PBVI CPOMDP policy approximation described earlier, the local VB approximation for hybrid inference with softmax models developed in [2] is used to approximate the product of a softmax model and a GM as a variational GM,

$$\alpha^i_n \ p(o|s') = \left[\sum_k w^i_k \phi(s'|s^i_k, \Sigma^i_k)\right]\left[\frac{\exp w_o^T s' + b_o}{\sum_{c=1}^{S} \exp w_c^T s' + b_c}\right]$$
$$\approx \sum_{h=1}^{H} w_h \phi(s'|\mu_h, \Sigma_h) \quad (12)$$

Figure 3 shows the key idea behind this VB approximation using a toy 1D problem. The softmax function (blue curve, e.g. representing $p(o|s')$ in (12)) is approximated by a lower bounding variational Gaussian function (black curve). The variational Gaussian is optimized to ensure the product with another Gaussian function (green, e.g. representing α^i_n)

results in a good Gaussian approximation (red dots) to the true non-Gaussian (but unimodal) product of the original softmax function and Gaussian functions (solid magenta).

More formally, the VB update derived in [2] for approximating the product of a normalized Gaussian (mixture) pdf $b(s) = p(s|o)$ and a softmax function $p(o|s)$ can be adapted and generalized for approximating the product of an *unnormalized* Gaussian (mixture) α_n^i (from the intermediate Bellman backup steps) and softmax likelihood. In the first case, consider the posterior Bayesian pdf for a Gaussian prior $p(s)$,

$$p(s|o) = \frac{p(s)p(o|s)}{p(o)} = \frac{1}{C}\phi(s|\mu,\Sigma)\frac{\exp w_o^T s + b_o}{\sum_{c=1}^M \exp w_c^T s + b_c}$$

$$C = \int_{-\infty}^{\infty} \phi(s|\mu,\Sigma)\frac{\exp w_o^T s + b_o}{\sum_{c=1}^M \exp w_c^T s + b_o} ds$$

By approximating the softmax likelihood function as an unnormalized variational Gaussian function $f(o,s)$, the joint pdf and normalization constant C can be approximated as:

$$p(s,o) \approx \hat{p}(s,o) = p(s)f(o,s)$$

$$C \approx \hat{C} = \int_{-\infty}^{\infty} \hat{p}(s,o) ds.$$

The key trick here is that (for any discrete observation category $j \in \Omega$) it is always possible to ensure $f(o=j,s) \leq p(o=j|s)$ by construction, using the variational parameters y_c, α, and ξ_c such that

$$f(o=j,s) = \exp\left\{g_j + h_j^T s - \frac{1}{2}s^T K_j s\right\}$$

$$g_j = \frac{1}{2}[b_j - \sum_{c\neq j} b_c] + \alpha(\frac{m}{2} - 1) + \sum_{c=1}^m \frac{\xi_c}{2} + \lambda(\xi_c)[\xi_c^2 - (b_c - \alpha)^2] - \log(1 + \exp\{\xi_c\})$$

$$h_j = \frac{1}{2}[w_j - \sum_{c\neq j} w_c] + 2\sum_{c=1}^m \lambda(\xi_c)(\alpha - b_c)w_c$$

$$K_j = 2\sum_{c=1}^m \lambda(\xi_c) w_c w_c^T$$

Since $f(o=j,s) \leq p(o=j|s)$ for any choice of the variational parameters, it follows that $\hat{C} \leq C$. As such, the variational parameters which produce the tightest lower bound $\hat{C}$ can be found through an iterative expectation-maximization algorithm, which requires alternately re-estimating $\hat{p}(s|o)$ given new values of the variational parameters, and then recomputing the variational parameters based on new expected values of s from $\hat{p}(s|o)$. Upon convergence of $\hat{C}$ to a global maximum, the product $p(s,o=j) = p(s)p(o=j|s)$ becomes well-approximated by the product $\hat{p}(s,o=j) = p(s)f(o=j|s)$, which is another (unnormalized) Gaussian function,

$$\hat{p}(s,o) = \exp\left\{(g_p + g_J) + (h_p + h_j)s - \frac{1}{2}s^T(K_p + K_j)s\right\}$$

Normalizing this joint distribution gives the posterior Gaussian pdf approximation $\hat{p}(s|o) = \phi(s|\mu_h, \Sigma_h)$.

Now, approximating the product of a Gaussian mixture with a softmax model follows immediately fact that the product is a sum of weighted products of individual Gaussians with the softmax model, where each individual product can be approximated via variational Bayes. To adapt the approximation to the case where the 'prior' GM function is now an unnormalized GM function, the results simply must be multiplied by the normalizing constant $\hat{C}$ (i.e. the approximate joint $\hat{p}(s,o=j)$ is used for each mixture term instead). This allows eq. (9) for the intermediate α function update in the PBVI backup to be approximated as

$$\alpha_{a,o}^i(s) = \int_{s'} \alpha_{n-1}^i(s')p(o|s')p(s'|s,a)ds' \quad (13)$$

$$\approx \int_{s'} \left[\sum_k w_k^i \phi(s'|s_k^i, \Sigma_k^i)\right]\left[\frac{\exp w_o^T s' + b_o}{\sum_{c=1}^S \exp w_c^T s' + b_o}\right] \times [\phi(s'|s + \Delta(a), \Sigma^a)]\, ds', \quad (14)$$

$$\approx \sum_{h=1}^K w_h \phi(s|\hat{\mu}_h - \Delta(a), \hat{\Sigma}_h + \Sigma^a)$$

(where $(\Delta(a), \Sigma^a) =$ known constants for action a).

In practice, the intermediate $\alpha_{a,o}(s)$ functions are often independent of the belief being backed up, and can therefore be calculated once per iteration over all beliefs. Algorithm 1 summarizes how the GM-based PBVI updates developed in Section II are thus modified to use the VB approximation for softmax semantic observation likelihoods.

VB-POMDP Backup
Input : $b \in B_0$, Γ_{n-1}
for $\forall \alpha_{n-1} \in \Gamma_{n-1}$,$\forall a \in A$, $\forall o \in \Omega$:
$\alpha_{a,o}(s) \leftarrow \sum_h w_h \phi(s|\mu_h - \Delta(a), \Sigma^a + \Sigma_h)$
$\alpha_n(s) = r_a(s) + \gamma \sum_o \arg\max_{\alpha_{a,o}} (< \alpha_{a,o}, b >)$
return $\alpha_n(s)$

Algorithm 1: VB-POMDP Backup

As per [2], recursive semantic observation updates to GM $b(s)$ pdfs can also be carried out online during execution of these policies using softmax likelihoods with the VB approximation, as shown in Fig. 4,

$$b(s') \propto p(o|s') \int_s p(s'|s,a)b(s)ds$$

$$= \left(\frac{\exp w_r^T s'}{\sum_{c=1}^S \exp w_c^T s'}\right)\left[\sum_j w_j \phi(s'|\mu_j + \Delta(a), \Sigma^a + \Sigma_j)\right]$$

$$\approx \sum_{z=1}^Z w_z \phi(s'|\mu_z, \Sigma_z).$$

In this example, the resulting posterior GM pdf for the 'no detection' update has only 4 components [1], thus demonstrating that parametrically simpler softmax models can drasti-

[1] the 2 prior components in this example are evaluated against separate categories for 'no detection left' and 'no detection right', which together make up a non-convex 'no detection' semantic observation class

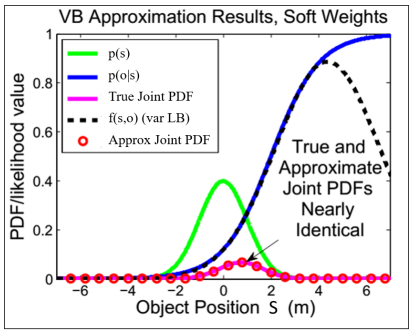


Fig. 3: 1D illustration of VB approximation.

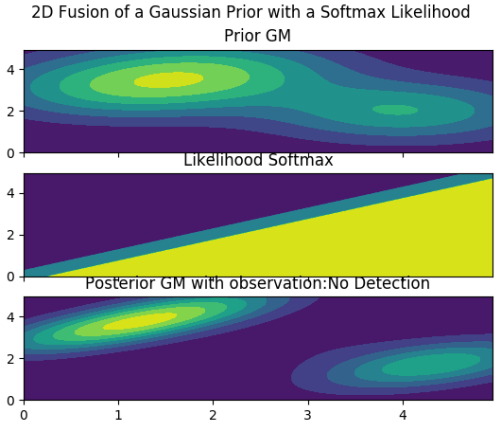


Fig. 4: GM belief update with softmax function observations

cally reduce the resulting complexity of inference compared to unnormalized GM likelihood functions.

B. Clustering-based GM Condensation

To remain computationally tractable, the GMs representing each α function must also be condensed such that,

$$\alpha_n^i = \sum_{k=1}^{M} w_k \phi(s|\mu_k, \Sigma_k) \approx \hat{\alpha_n^i} = \sum_{k=1}^{M'} w_k \phi(s|\mu_k, \Sigma_k),$$
$$\text{where } M' < M.$$

(mixture terms in $b(s)$ must also be compressed following dynamics prediction and Bayesian observation updates).

Existing GM condensation algorithms perform myopic pairwise merging of the M components in α_n^i, such that the resulting M' components in $\hat{\alpha_n^i}$ minimize some information loss metric [5], [4]. Naïve pairwise merging tends to be very expensive and slow when $M \geq 100$ (often the case for long horizon Bellman recursions with $N \geq 2$).

To improve the speed of condensation, we employ a 'divide and conquer' strategy which first pre-classifies the mixture indices into local clusters (submixtures) using K-means, and then condenses each cluster to some pre-determined number of components via pairwise merging, before recombining the results to a condensed mixture with the desired size $M' < M$. For merging within submixture clusters, we use the Runnall's algorithm [8], which uses an upper bound on the KL divergence between the pre-merge and post-merge submixture to select the least dissimilar component pairs merging.

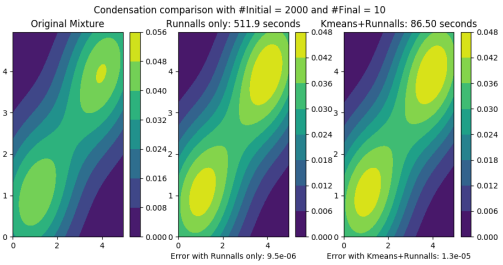


Fig. 5: Condensation Comparison of Runnall's method to K-means hybrid method.

Preliminary results indicate that this new hybrid method achieves approximately the same accuracy (measured by the integral square difference metric for GMs [11]) for condensation performance as classical full scale pairwise merging, although the hybrid method is considerably cheaper and faster (e.g. 512 secs vs. 86.5 secs for $M = 2000 \rightarrow M' = 10$ with $N = 2$).

Figure 5 shows a comparison of the classical full-mixture Runnall's condensation method to our hybrid cluster-then-condense method for a GM with $M = 2000$ components. The Integral Square Difference metric proposed by [11] is used to assess the accuracy of each method,

$$ISD(G_h, G_r) = J_{hh} - 2J_{hr} + J_{rr},$$
$$J_{hh} = \sum_{i}^{N_h} \sum_{j}^{N_h} w_i w_j \phi(\mu_i|\mu_j, \Sigma_i + \Sigma_j),$$
$$J_{hr} = \sum_{i}^{N_h} \sum_{j}^{N_r} w_i w_j \phi(\mu_i|\mu_j, \Sigma_i + \Sigma_j),$$
$$J_{rr} = \sum_{i}^{N_r} \sum_{j}^{N_r} w_i w_j \phi(\mu_i|\mu_j, \Sigma_i + \Sigma_j).$$

The results show that both methods achieve approximately the same ISD accuracy, although the hybrid method is considerably faster.

Theoretically, the cluster-then-merge approach is natural to consider, since any GM can be generally viewed a 'mixture of local submixtures'. From this standpoint, mixture components belonging to different local submixtures are unlikely to be directly merged in a pairwise global condensation algorithm, whereas those belonging to the same submixture are much more likely to be merged. As such, the global merging operation can be broken up into several smaller parallel merging operations within each submixture. In the approach used here, the submixtures are identified using a simple fast k-means clustering heuristic on the component means. Additional work is required to verify the robustness of the K-means hybrid method in for general problem settings, although other techniques for identifying submixture groups could also be used (e.g. to also account for mixand weights, covariances, etc.).

IV. Simulation Results

A. Colinear Robots Simulation Results

Fig. 6 compares the resulting average final rewards achieved over a 100 step simulation for 100 simulation runs, using policy approximations for the 1D cop-robot search problem presented earlier. The second column shows the average final rewards the proposed VB-POMDP method (with the softmax likelihood model shown in Fig. 2b), while the first column shows the average final rewards obtained for the GM-POMDP policy approximation of [5] (using the GM observation models shown in Fig. 2a). For reference on the optimality of both methods, results for a third greedy one-step implementation of the latter approximation is also shown in the third column. Statistically, the VB-POMDP policy approximation average performance could not be differentiated from the baseline GM-POMDP policy, with $p > 0.01$. However, both policies achieved a higher average accumulated reward than the comparison greedy approach, with $p < 0.01$.

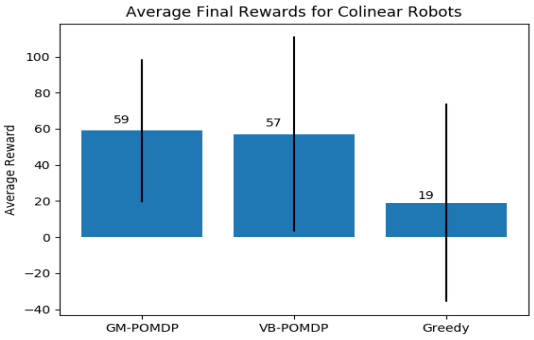


Fig. 6: Rewards achieved on simple target search problem with standard deviations over 100 simulation runs shown

B. 2D Movement Target Search Results

In a more complex extension of the previous example, another CPOMDP was developed in which an autonomous robot 'cop' attempts to localize and catch a mobile 'robber', where both are allowed to move within a bounded 2D space. Here, $S = \mathbb{R} \times \mathbb{R}$ consists of two bounded continuous random variables at each discrete time step t, but now $s = [\Delta X, \Delta Y] = [Cop_x - Rob_x, Cop_y - Rob_y]$. The robber again executes a Gaussian random walk, and the cop can now choose from among 5 noisy actions $A = \{East, West, North, South, Stay\}$.

The cop obtains semantic observations chosen from $\Omega = \{East, West, North, South, Near\}$; the softmax likelihood model for this is shown in Fig. 7. Rewards are dispensed to the cop by evaluating a GM for a particular action at the state at each time step. Reward GMs each consist of a single weighted Gaussian located at the point $[0 - \Delta(a)_x, 0 - \Delta(a)_y]$, where $\Delta(a)$ is the expected displacement of the cop resulting from a given action. Both GM-POMDP and VB-POMDP solvers were given the same amount of time to find a policy and again compared to a simple greedy online policy. Figs. 8 and 9 show that the VB-POMDP method in this case significantly outperforms both the GM-POMDP and greedy approximations with $p < 0.01$.

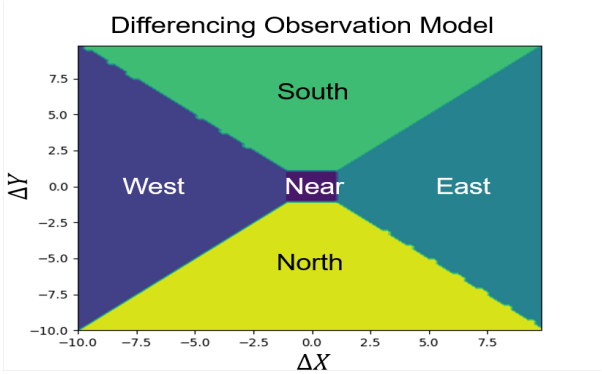


Fig. 7: Softmax observation model for the 2D target motion problem

.

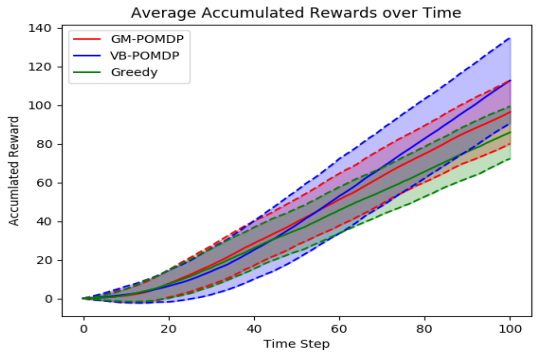


Fig. 8: Rewards vs. time for 2D search problem over 100 simulation runs (solid lines = mean value; shaded = 2 standard deviations).

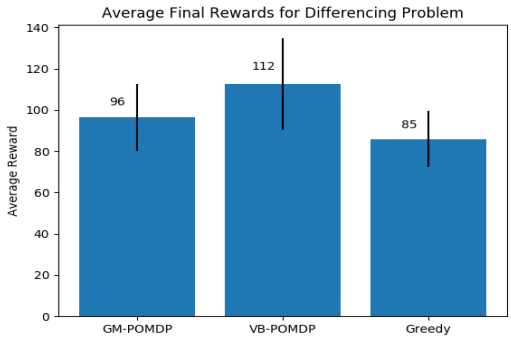


Fig. 9: Average Final Rewards for the Differencing Problem with standard deviations over 100 simulation runs shown

C. Discussion

The results indicate that for a simple problem the VB-POMDP method achieves near parity with the existing method, and that both methods surpass the greedy approach. This is as expected, as both the Gaussian mixture and

softmax observation models were constructed to approximate the same semantic model with all else held equal. Importantly, the approximations involved with VB seem to not significantly impact the final result.

Comparing the results from section IV.A and IV.B suggests that, as the complexity of the problem increases, the VB-POMDP approximation outperforms the standard GM-POMDP approach. A primary contributing factor to this disparity is that VB-POMDP requires less time per backup step than GM-POMDP for this problem (e.g. 10 secs vs. 92 secs running on a 2.6 GHz processor running Linux with 16 GB RAM). This is largely due to the number of mixands generated by each method. In a single backup step, the GM-POMDP method produces alpha-functions of size $|\alpha_n| = \sum^{|\Omega|} |\alpha_{n-1}||p(o|s)|$, while VB-POMDP produces alpha-functions of size $|\alpha_n| = \sum^{|\Omega|} |\alpha_{n-1}|$. The additional time needed for VB to converge is more than offset by the condensation time savings from having fewer mixands. Being able to accomplish more backups should allow to the solver to more closely approximate the ideal policy. This would suggest that solvers given an infinite amount of time (and therefore an infinite number of backups) to find a policy would yield the same results for both approaches. However, in practical time-limited situations, VB-POMDP holds a distinct advantage.

V. Conclusion

This paper presented VB-POMDP, a new method for solving CPOMDP policies using semantic sensor statement observations that are modeled by softmax models. Softmax models are ideal for modeling semantic sensor observation likelihoods for CPOMDPs, since they are both cheaper and simpler to construct and evaluate compared to unnormalized GM functions that have been proposed for the same purpose. To overcome the analytical intractability of using softmax models in standard GM-based PBVI policy approximations for CPOMDPs, a variational Gaussian inference approximation was described and introduced to maintain the closed-form recursive nature of the GM PBVI approximation. Thanks to the use of compact softmax likelihood models, this approach also tends to produce far fewer mixands in the intermediate PBVI recursion steps.

A novel approach to Gaussian mixture condensation was also described and demonstrated, using K-means to pre-cluster mixands into sub-mixtures that are then condensed in parallel. This approach was shown to be considerably faster than an existing state-of-the-art global condensation technique, while maintaining a similar level of accuracy.

The VB-POMDP method was shown in a baseline target search problem to achieve near parity with a state of the art GM-based CPOMDP policy approximation method, and was shown to be significantly more effective on a more complicated search problem. As such, this work has many interesting implications for developing sophisticated planning and control algorithms for autonomous reasoning in hybrid probabilistic domains with continuous state spaces and discrete semantic observations.

Future work will explore various relaxations of simplifying modeling assumptions made in this work. For instance, non-linear switching mode dynamics models, as developed in [4] will be studied. These models will allow for more complex probabilistic state transition functions $p(s'|s, a)$, and can themselves be modeled with softmax functions. Additionally, future work will examine problems with larger spaces of actions and observations than those considered here. The various approximations made in this work will also analyzed in closer detail. In particular, it is desirable to obtain bounds on the accuracy of the K-means condensation method, as well as possible lower bounds on the value function via the VB inference approximation. Finally, bulding on previous work in [2], [10], the CPOMDP framework developed here will be leveraged for cooperative human-robot target search and tracking applications, so that semantic 'human sensor data' from natural language inputs can be combined with optimal robotic sensing and motion planning for tightly integrated teaming.

References

[1] P Abbeel. Reinforcement learning for nonlinear dynamical systems and Gaussian belief space planning. *Reinforcement Learning and Decision Making*, page 13, 2013.

[2] Nisar R. Ahmed, Eric M. Sample, and Mark Campbell. Bayesian multicategorical soft data fusion for human-robot collaboration. *IEEE Transactions on Robotics*, 29(1):189–206, 2013.

[3] Frederic Bourgault, Tomonari Furukawa, and Hugh F Durrant-Whyte. Coordinated decentralized search for a lost target in a bayesian world. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 48–53. IEEE, 2003.

[4] Emma Brunskill, Leslie Pack Kaelbling, Tomas Lozano-Perez, and Nicholas Roy. Planning in partially-observable switching-mode continuous domains. *Annals of Mathematics and Artificial Intelligence*, 58(3):185–216, 2010.

[5] M Spaan P Poupart JM Porta, N Vlassis. Point-based value iteration for continuous POMDPs. *IJCAI International Joint Conference on Artificial Intelligence*, 7:1968–1974, 2011.

[6] H Kurniawati, D Hsu, and W S Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. *Proc. Robotics: Science and Systems*, 2008.

[7] Joelle Pineau, Geoff Gordon, and Sebastian Thrun. Point-based value iteration: An anytime algorithm for POMDPs. *IJCAI International Joint Conference on Artificial Intelligence*, pages 1025–1030, 2003.

[8] Andrew R. Runnalls. Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):989–999, 2007.

[9] Allison Ryan and J. Karl Hedrick. Particle filter based information-theoretic active sensing. *Robotics and Autonomous Systems*, 58(5):574–584, 2010.

[10] Nicholas Sweet and Nisar Ahmed. Structured synthesis and compression of semantic human sensor models for Bayesian estimation. *Proceedings of the American Control Conference*, 2016-July(2):5479–5485, 2016.

[11] Jason L. Williams and Peter S. Maybeck. Cost-function-based Gaussian mixture reduction for target tracking. *Proceedings of the 6th International Conference on Information Fusion, FUSION 2003*, 2:1047–1054, 2003.