Multimodal Nonlinear Filtering Using Gauss-Hermite Quadrature

Hannes P. Saal, Nicolas M.O. Heess, and Sethu Vijayakumar

School of Informatics, University of Edinburgh, Edinburgh, Scotland, UK
{hannes.saal,sethu.vijayakumar}@ed.ac.uk, n.m.o.heess@sms.ed.ac.uk

Abstract. In many filtering problems the exact posterior state distribution is not tractable and is therefore approximated using simpler parametric forms, such as single Gaussian distributions. In nonlinear filtering problems the posterior state distribution can, however, take complex shapes and even become multimodal so that single Gaussians are no longer sufficient. A standard solution to this problem is to use a bank of independent filters that individually represent the posterior with a single Gaussian and jointly form a mixture of Gaussians representation. Unfortunately, since the filters are optimized separately and interactions between the components consequently not taken into account, the resulting representation is typically poor. As an alternative we therefore propose to directly optimize the full approximating mixture distribution by minimizing the KL divergence to the true state posterior. For this purpose we describe a deterministic sampling approach that allows us to perform the intractable minimization approximately and at reasonable computational cost. We find that the proposed method models multimodal posterior distributions noticeably better than banks of independent filters even when the latter are allowed many more mixture components. We demonstrate the importance of accurately representing the posterior with a tractable number of components in an active learning scenario where we report faster convergence, both in terms of number of observations processed and in terms of computation time, and more reliable convergence on up to ten-dimensional problems.

1 Introduction

Filtering is a common problem in robotics and other areas where observations become available sequentially. The observations have a stochastic dependence on an unobserved underlying state and the goal is to infer the posterior distribution over the state at a particular time step given the observations up to that time step. In settings where the observation function is nonlinear the posterior state distribution can take on complex shapes and even become multimodal. For example, in visual tracking, observations are often ambiguous due to other moving objects or occlusion. In such cases, the posterior distribution might comprise a relatively small number of reasonably well isolated modes, each of which could be modeled well by a single Gaussian distribution. Representing such multimodal distributions as a whole with a single Gaussian distribution, however, can

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part III, LNAI 6913, pp. 81–96, 2011.

lead to divergence of the filter estimates and generally poor performance. Properly representing posterior state distributions, including their multimodality, is especially important when using active learning methods since the uncertainty captured by the posterior is used directly in the decision of how to query the next observation in order to resolve ambiguities in the state as quickly as possible.

Although filtering approaches that rely on mixtures of Gaussians to represent a skewed or multimodal state distribution have a long history, most of these approaches rely on banks of linear filters, each with a Gaussian state distribution, that are updated independently [2]. While this has the advantage of being computationally very efficient, since interactions between the mixture components are being ignored, the resulting mixture distribution is likely to be a poor fit of the true underlying state distribution. This can lead to poor overall performance unless a large number of mixture components is used.

Yet, in many situations a small number of Gaussian components can be sufficient to capture the essential structure of the posterior distribution if the parameters of the mixture components are chosen appropriately. This leads us to explore new ways of optimizing the parameters of the approximate mixture representation of the posterior distribution: We present a novel approach to the problem of mixture filtering that takes inspiration from variational approaches to approximate inference and combine this with a deterministic sampling approach: We assume that the prior (e.g. the filtering distribution from the previous time step) is given as a mixture of Gaussians (MoG). Due to this MoG prior, and due to a nonlinear observation function the posterior distribution over the state given a new observation can have a complex shape. We therefore approximate the new posterior distribution again as a MoG distribution. We optimize this approximate posterior distribution by approximately minimizing the Kullback-Leibler (KL) divergence between the true updated state distribution and the approximate MoG representation. Exact minimization of the KL divergence is intractable. Our approximate minimization relies on a deterministic sampling approach, Gauss-Hermite Quadrature, which evaluates general integrals by evaluating the integrand at suitably chosen sample points, and we describe a novel way to re-formulate the required integrals so as to optimize the accuracy of the method for the problem at hand.

2 Methods

2.1 Problem Statement

We are interested in filtering problems, but in this paper ignore the time update and instead focus on the measurement update.[1] At each time step we receive a new observation $\mathbf{y}_t$ and use this to update our current estimate of the latent state $\mathbf{x}_t$: $p(\mathbf{x}_t|\mathbf{y}_t, \mathbf{y}_{1\ldots t-1}) \propto \int \mathrm{d}\mathbf{x}_{t-1} p(\mathbf{y}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{1\ldots t-1})$ where

[1] This corresponds to a static target. Including a dynamics model is straightforward and time updates could be done as in other mixture filters by propagating each mixture component independently.

$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \delta(\mathbf{x}_t - \mathbf{x}_{t-1})$. For any time step this problem can be thought of as computing the posterior $p(\mathbf{x}|\mathbf{y})$ using the state distribution from the *previous* time-step as the prior $p(\mathbf{x})$. As discussed above, for most interesting cases the true posterior cannot be computed exactly. This is the case, when the likelihood $p(\mathbf{y}|\mathbf{x})$ is not of the convenient linear-Gaussian form. The focus of this paper is on developing a formulation of the filtering problem that allows for an approximate representation of the state distribution given previous observations that is sufficiently flexible to account for uncertainty in the latent state e.g. when the true posterior is multimodal or skewed. Since in our scenario the posterior computed in step t will be the prior for step $t+1$ we are interested in a representation of the posterior that can be directly used as the prior in the calculations for the next time step. Specifically, we will be representing the prior and the approximate posterior as a MoG distribution. Furthermore, we assume that the observation likelihood is a Gaussian with fixed covariance, but with a mean that depends on $\mathbf{x}$ via a nonlinear function $f(\cdot)$ which we choose to represent as a radial basis function (RBF) network. We choose this form because it allows us to treat terms arising from the likelihood analytically (cf. Sec. 2.2) and at the same time is general enough to approximate any nonlinear function to arbitrary accuracy [15]. Alternative formulations, e.g. using a Gaussian process representation are also conceivable [5][2]. Thus, at each time step we are faced with the following problem: Given

$$p(\mathbf{x}) = \sum_n \gamma_n p_n(\mathbf{x}) \tag{1}$$

$$p_n(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\nu}_n, \boldsymbol{\Theta}_n) \tag{2}$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{f}(\mathbf{x}), \boldsymbol{\Sigma}_y) \tag{3}$$

$$f(\mathbf{x}) = \sum_j c_j k(\mathbf{x}, \mathbf{m}_j) \tag{4}$$

$$k(\mathbf{x}, \mathbf{m}_j) = \exp\left\{-0.5(\mathbf{x} - \mathbf{m}_j)^T \mathbf{S}^{-1}(\mathbf{x} - \mathbf{m}_j)\right\} \tag{5}$$

our goal is to approximate the state posterior $p(\mathbf{x}|\mathbf{y})$ with a mixture of Gaussians

$$q_{\text{mix}}(\mathbf{x}) = \sum_m \alpha_m q_m(\mathbf{x}) \tag{6}$$

$$q_m(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m). \tag{7}$$

2.2 Fitting a Gaussian Mixture to the Posterior

Given a new observation $\mathbf{y}$, and the current prior $p(\mathbf{x})$ we need to optimize the parameters of $q(\mathbf{x})$ so as to match $p(\mathbf{x}|\mathbf{y})$ as closely as possible. The variational

[2] If the observation function is given in analytical form, expected values can instead be estimated by the using linearization methods from commonly used unimodal filters, like the extended Kalman filter or the unscented filter.

approach requires the Kullback-Leibler (KL) divergence between the approximate posterior $q(\mathbf{x})$ and the true posterior to be minimized

$$\mathrm{KL}\left[q||p\right] = \int \mathrm{d}\mathbf{x}\; q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})}. \tag{8}$$

The choice of $\mathrm{KL}\left[q||p\right]$ can be motivated by the fact that the resulting approximate posterior leads to a lower bound on the log marginal likelihood $\log p(X)$ of a latent variable model $p(X, Z)$:

$$\log p(X) = \log \int \mathrm{d}Z\; p(X, Z) \geq \int \mathrm{d}Z\; q(Z) \log \frac{p(X, Z)}{q(Z)} \tag{9}$$

where the difference between the left-hand and the right-hand side is just the KL divergence between the approximate posterior q and the true posterior $p(Z|X)$. Minimizing this divergence tightens the bound (it becomes tight iff q is equal to the true posterior, in which case the KL divergence is zero). Maximizing this bound with respect to the model parameters allows for maximum likelihood learning in models with intractable posteriors.

In order to be able to capture complex shapes of the true posterior, including multimodality, we choose MoG as our approximate posterior distribution. In our case $\mathrm{KL}\left[q_{\mathrm{mix}}||p\right]$ can be broken down as follows:

$$\begin{aligned} \mathrm{KL}\left[q_{\mathrm{mix}}||p\right] &= \int \mathrm{d}\mathbf{x}\; q_{\mathrm{mix}}(\mathbf{x}) \log \frac{q_{\mathrm{mix}}(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})} && (10) \\ &= -\mathrm{H}\left[q_{\mathrm{mix}}\right] - \sum_m \alpha_m \mathrm{E}_{q_m}\left[\log p(\mathbf{x})\right] \\ &\quad - \sum_m \alpha_m \mathrm{E}_{q_m}\left[\log p(\mathbf{y}|\mathbf{x})\right] + \mathrm{const} && (11) \end{aligned}$$

where $\mathrm{H}\left[q\right] = -\int \mathrm{d}\mathbf{x}\; q(\mathbf{x}) \log q(\mathbf{x})$ is the differential entropy of q and the expectations $\mathrm{E}_{q_m}\left[\cdot\right]$ are taken with respect to the Gaussian components q_m of the mixture posterior.

In order to obtain the approximate posterior this expression needs to be minimized with respect to the parameters of q_{mix}. With the choices made above for prior, likelihood, and approximate posterior (equations 1–6) we can exactly compute the third term in (11) (see Appendix A.1) but the first and second term are not tractable since they involve integrals taken over log-sums. We approximate these intractable integrals in (11) using quadrature methods as described in the next section.

2.3 Gauss-Hermite Quadrature

Gauss-Hermite quadrature [1] approximates d-dimensional integrals by deterministically selecting sample points from a weight function—in this case a Gaussian $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$—and then computing a weighted sum of the function values at those sample points:

$$\int \mathrm{d}\mathbf{x}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)f(\mathbf{x}) = \mathrm{E}_{q_m}\left[f(\mathbf{x})\right] \approx \pi^{-\frac{d}{2}} \sum_{\mathbf{h}} w_{\mathbf{h}} f(\mathbf{z}_{\mathbf{h}}) \tag{12}$$

where $w_{\mathbf{h}} = \prod_d w_{\mathbf{h}(d)}$, i.e. the overall weights are determined as the product of the individual single dimension weights. $\mathbf{z}_{\mathbf{h}}$ are the transformed sample points $\mathbf{z}_{\mathbf{h}} = \mathbf{L}_m \mathbf{x}_{\mathbf{h}}\sqrt{2} + \boldsymbol{\mu}_m$, where $\mathbf{L}_m\mathbf{L}_m^T = \boldsymbol{\Sigma}_m$. In this paper, we set $\mathbf{L}_m$ to be the Cholesky factor, but any triangular decomposition of $\boldsymbol{\Sigma}_m$ could be used (cf. [3]). The sample points and corresponding weights are given by the roots of the Hermite polynomial and can be calculated offline and stored. Derivatives of the resulting approximation are straightforward to calculate. In our setup, the function $f(\cdot)$ is given as an RBF and thus the integral has the following form:

$$\mathrm{E}_{q_m}\left[f\right] = \mathrm{E}_{q_m}\left[\log \sum_n \alpha_n \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)\right] \tag{13}$$

Gauss-Hermite quadrature could be used to approximate this integral directly (see [7] for such an approach), but we found that the estimate can become highly inaccurate if the variances of the individual mixture components differ considerably and only a small number of sample points is used. This can then lead to a divergence of the optimization of the KL-divergence. Instead of increasing the number of sample points, which quickly becomes untenable in high dimensions, we instead rewrite the integral as a sum, which allows us to approximate each term of this sum individually. This should allow for higher accuracy, as we can optimize the sample points for each term separately. Thus, we write $\mathrm{E}_{q_m}\left[f\right]$ as:

$$\mathrm{E}_{q_m}\left[f\right] = \mathrm{E}_{q_m}\left[\log \alpha_1 \mathcal{N}_1\right] + \sum_{n=2}^{N} \mathrm{E}_{q_m}\left[\log\left(1 + \frac{\alpha_n \mathcal{N}_n}{\sum_{k=1}^{n-1} \alpha_k \mathcal{N}_k}\right)\right] \tag{14}$$

where we use the abbreviation $\mathcal{N}_n$ to stand in for the longer $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$. The first component of this sum can be calculated analytically, while the remaining ones have to be approximated. We use different weighting functions for each of these terms. For each integral, we approximate it in the standard way as described above, if the variance of $\mathcal{N}_m$ is smaller than the variance of $\mathcal{N}_n$. Otherwise, we rewrite the integral and choose $\mathcal{N}_n$ instead of $\mathcal{N}_m$ as the weighting function:

$$\underbrace{\mathrm{E}_{q_m}\left[\log\left(1 + \frac{\alpha_n \mathcal{N}_n}{\sum_{k=1}^{n-1} \alpha_k \mathcal{N}_k}\right)\right]}_{\text{Integral over } q_m} = \underbrace{\mathrm{E}_{q_n}\left[\frac{\mathcal{N}_m}{\mathcal{N}_n}\log\left(1 + \frac{\alpha_n \mathcal{N}_n}{\sum_{k=1}^{n-1} \alpha_k \mathcal{N}_k}\right)\right]}_{\text{Integral over } q_n} \tag{15}$$

As illustrated in Fig. 1 this makes it more likely that the sample points will capture the region that is interesting for integration, as the mean and variance of the new weighting function should be closer to the mode and log curvature of the integrand, and thereby improving accuracy [14]. In the multivariate case, we either pick the component with the lowest covariance determinant, or (when restricting ourselves to diagonal covariance matrices) treat each dimension independently.

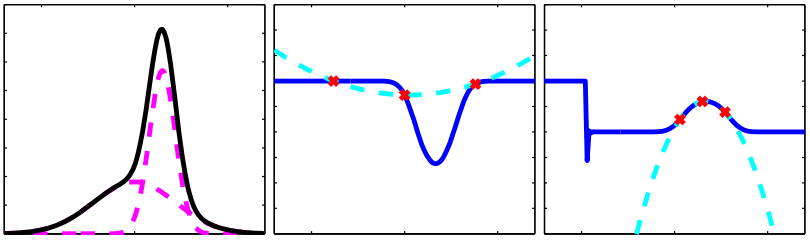

Fig. 1. Illustration of the improved Gauss-Hermite method. Left: Mixture of two Gaussians, q_1 (left) and q_2 (right). Middle: Sample points from q_1 (red), integrand (dark blue) and implicit polynomial fit to integrand by quadrature method (light blue) when integrating according to the left-hand term in Eq. (15). Right: Improved fit in relevant region (around sample points) when integrating according to the right-hand term in Eq. (15), with sample points taken from q_2.

3 Related Work

As explained in the introduction, Gaussian mixture distributions as an approximation to the true state distribution have a long history in the filtering literature. The classic approach to Gaussian mixture filtering uses a weighted sum of extended Kalman filters (EKFs) running in parallel [2]. Newer approaches replace the EKFs with linear filters using deterministic sampling [10,3]. However, in all these cases several unimodal filters run independently of each other in parallel. While this is computationally very efficient, it also leads to inferior representation of the posterior distribution. Different ways to compute the posterior mixture weights have also been proposed, in an attempt to decrease the distance between the true posterior and the mixture approximation [10]. In contrast, in our approach we adapt *all* Gaussian mixture parameters (i.e. means, covariances, and mixture weights) *jointly* so as to fit the true posterior as closely as possible.

Mixture distributions as approximate posterior distributions have been considered previously in the literature on variational inference [11]. In particular Lawrence and Azzouzi[13], as well as Bouchard and Zoeter [4] consider the use of MoGs to approximate continuous-valued posterior distributions. To our knowledge, these have, however, not been considered in the context of filtering. Compared to previous work, the filtering application leads to an additional intractable term in (11) in the form of the integral over the logarithm of the prior which, in our case, does not have a simple parametric form such as Gaussian, but rather is a MoG itself. Previous work deals with the intractable terms differently: Jaakola and Jordan [11], who consider the case of discrete distributions, employ an advanced upper bound to approximate the expectation of the logarithm arising in the expectation of $\mathrm{H}\left[q_{\mathrm{mix}}\right]$. This bound requires the optimization of additional variational parameters including a set of "smoothing distributions". Lawrence and Azzouzi [13] adapt this approach to the continuous case using MoGs for the posterior. The variational smoothing distributions then

take the form of Gaussians whose parameters need to be optimized alongside the other variational parameters. This happens in an iterative scheme which alternates gradient ascent with respect to the different sets of parameters.

In this paper we are primarily interested in a fast method for estimating the approximate representation of the posterior. Instead of minimizing an upper bound on the KL divergence as in [11,13] we therefore approximate the intractable integrals in (11) using quadrature methods which (a) avoid the need to iteratively optimize additional parameters and are therefore fast, and (b) directly extend to the second intractable term to which the upper bound to the logarithm used for evaluating the entropy is not applicable.

Gauss-Hermite quadrature has been used in unimodal Gaussian filtering before to approximate certain integrals [10,3]. The well-known unscented filter [12] also relies on deterministic sampling, but uses a slightly different approach to selecting the sample points and weights. In all these cases, deterministic sampling is used in order to calculate expectations over the observation function. In our approach, we use deterministic sampling in order to approximate expectations over log-sums, i.e. entropy-like terms and their gradients. While entropies of mixtures of Gaussians have been approximated using deterministic sampling before [7], our way differs in that we treat each component *inside* the log-sum independently, and therefore achieve higher accuracy. Another way that has been proposed would be to derive the Taylor expansion of the log-sum up to a certain degree and then integrate analytically [9].

4 Results

4.1 Problem Setup

In order to test the performance of our proposed variational approach, we examined how well it could fit complex (i.e. multimodal or skewed) state distributions when compared with other approaches. To highlight the importance of such representations, we additionally tested whether an improved posterior representation would help in a localization task with ambiguous observations, while using active learning in order to speed up convergence.

Depending on task difficulty, we compared our approach with a number of other options: first, a linear mixture filter[3] using the same number of components as the variational mixture filter; second, linear filters using a higher number of components (3^d or 7^d)[4]; and finally, a variational filter using just a single component, in order to examine how well a unimodal approach would perform[5].

[3] Our linear mixture filter implementation is a bank of independent filters that are updated independently. Note that because of the particular form of the nonlinear likelihood (RBF, cf. Eqs. (4), (5)) all required expectations can be computed analytically (see Appendix A.1).

[4] d denotes the dimensionality of the state distribution.

[5] The unimodal variational filter still needs to be optimized iteratively, but since the posterior distribution only consists of a single Gaussian, the KL divergence and its gradient can be evaluated analytically, so no numerical quadrature is needed and optimization is usually much quicker.

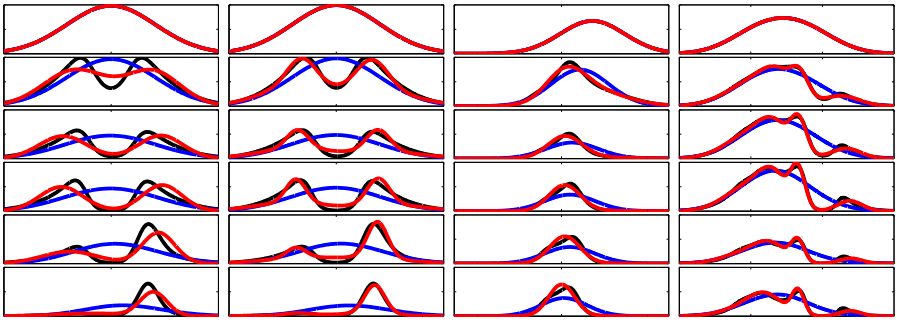

Fig. 2. Comparison of true posterior (black lines), variational (red) and 'bank of filters' (blue) approximations for different observation functions. Each row represents a new observation, with successive time steps ordered from top to bottom, while each of the four columns stands for a different problem: 1. Bimodal posterior, approximated with two components. 2. Same posterior as in (1), approximated with 4 components, 3. Skewed posterior, 2 components, 4. Highly multimodal posterior, 4 components.

The problem was set up as a localization task where the position of a (stationary) target at location $\mathbf{x}^*$ had to be estimated by probing search locations $\boldsymbol{\theta}$ sequentially and receiving observations $\mathbf{y}$ that depended on both the probe and target locations: $\mathbf{y} = f(\boldsymbol{\theta} - \mathbf{x}^*)$. Each run started with a wide Gaussian prior (zero-mean with an isotropic variance of 9), reflecting the fact that the location of the target was unknown. The observation function $f(\cdot)$ was set up as an RBF consisting of a small number of individual squared exponential components. In each new run, the location of these components with respect to the target location was sampled from a uniform distribution over a hyper-cube of length 8 centred at the origin. The number and kernel width varied with the dimensionality of the problem: We used 3 kernels with a width of 1.5 for the 4D problem, 4 kernels with width 1.5 for 6D, 5 kernels with width 3 for 8D, and 5 kernels with width 7 for 10D. We used two different types of observations: Ambiguous and Infomax observations. For ambiguous ones, search locations $\boldsymbol{\theta}$ were fixed such that observations always came from the mode of an individual RBF component, which frequently resulted in the posterior becoming multimodal. This type of observations was used to test how well the different methods were able to model multimodal state distributions (see results in Sec. 4.2). For Infomax observations, the search locations $\boldsymbol{\theta}$ were optimized via active learning such that they resulted in the biggest information gain about the position of the target (see Appendix A.2 for mathematical details and Sec. 4.3 below for results). These observations should allow the posterior to quickly converge onto the correct target. That is, knowing the observation function as well as the current (possibly multimodal) state distribution allows selecting search locations that disambiguate between the different potential target positions effectively.

For the cases where the linear filter used a higher number of components than the variational one, its prior was initialized to match the original prior as closely

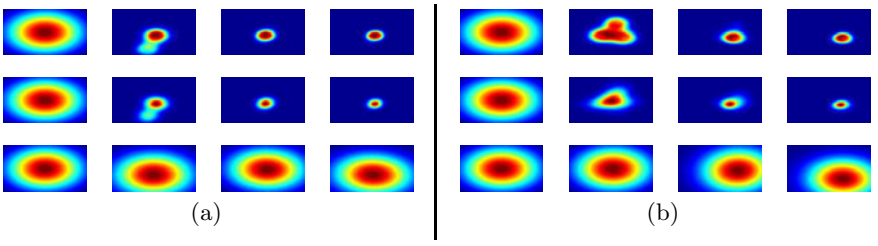

Fig. 3. (a) and (b). Two example runs with time increasing from left to right (4 steps each). Top row: Actual posterior (calculated numerically). Middle row: Variational mixture approximation (3 components). Bottom row: Linear mixture approximations (3 components). All methods start with the same prior and receive the same observations.

as possible by placing components on a grid and adjusting their weights so that the resulting mixture distribution matched the original broad Gaussian prior distribution. We also tried other initializations, e.g. randomly sampling components from the original prior, and found that different initializations did not influence the results much. For the variational approach, we used 3 quadrature points per dimension in all examples, leading to 3^d samples in total. For problems where $d > 2$, we restricted the variational method to diagonal covariance matrices; components in linear filters always maintained full covariance matrices, however. All algorithms were implemented in Matlab, using some functions from the Lightspeed toolbox[6]. For gradient descent we used the scaled conjugate gradient implementation provided by the Netlab toolbox[7].

4.2 Representation of Multimodal Posterior Distributions

First, we tested how well our approach could represent skewed and multimodal posterior state distributions of different dimensionalities. Some examples for different setups in 1D and 2D are shown in Figs. 2 and 3. As can be seen, the variational approximation generally approximates the posterior well and correctly finds and represents the major modes. The quality of approximation evidently improves with the number of mixture components that is used (see second column in Fig. 2). Moreover, skewed distributions can be fitted well, by using several mixture components (see e.g. the third column in Fig. 2). Sometimes the variational approximation covers several posterior modes with a single component, which also happens when there are more posterior modes than mixture components (see right panel in Fig. 3). The 'bank of filters' method on the other hand runs into problems if a unimodal prior splits into a posterior consisting of several components, and often retains an excessively high variance.

[6] `http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/`
[7] `http://www1.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/`

We also ran a more exhaustive test on low-dimensional problems (1–2D), where we kept the observation function constant, but systematically varied the mean and the (co)variance of the prior distribution with respect to the observation function. This introduced a big range of different nonlinearities for the methods to encounter. We ran the algorithms on several different observation functions by varying the number and locations of RBF components. For each individual trial, we (numerically) calculated the KL divergence between the distributions approximated by the mixture methods and the true state posterior. We found that, generally, our algorithm was at least as good as the linear mixture filter but often dramatically better, usually when the posterior state distribution became either considerably skewed or multimodal.

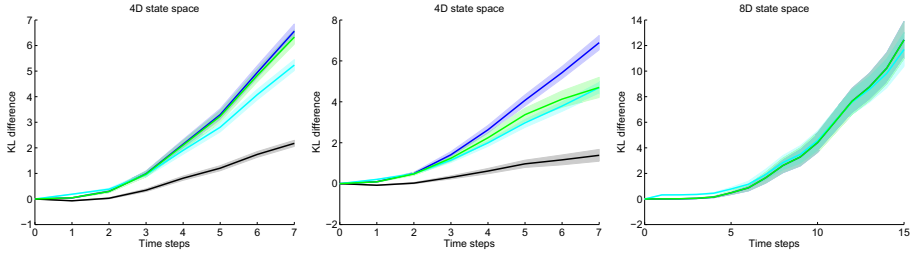

Fig. 4. Differences between the KL divergence between respective posterior approximation and the true posterior, and the KL divergence between the variational filter and the true posterior as a function of the number of observations. Positive values indicate that the variational mixture filter was closer to the true posterior (in terms of the KL divergence), while negative values denote the respective other filter being a better fit. Left: 4D problem with observation function consisting of a single bump. Middle: 4D problem with three observation function modes. Right: 8D problem with 5 modes. Blue line: Average difference between linear and variational mixture filters (both using 4 mixture components). Green line: Difference between unimodal (1 component) and variational filter). Light blue and black lines (4D only): Same for linear filters using 81 and 2401 components, respectively. Shaded regions indicate standard error of the mean.

In a further set of tests, we examined how well the different mixture methods were able to capture high-dimensional complex posterior distributions. In order to quantify the fit of the different representations, we calculated differences in the KL divergences between the different mixture approximations and the true posterior distributions over time: we used Monte-Carlo integration in order to arrive at an estimate of the KL divergence (up to a constant). In these tests we only presented ambiguous samples as we were interested in a complex posterior shape. Results for tasks in 4D (with either a single or three observation function modes) and 8D (using 5 modes) can be seen in Fig. 4. We ran this task for 100 (4D) or 25 (8D) random configurations of the observation function and target. We noticed several interesting effects. First, a linear filter using the same

number of components as the variational filter consistently performs worse, independent of whether the posterior state distribution is skewed (left panel) or becomes multimodal (middle panel). Second, adding more components to the linear filter improves the difference in KL divergence. However, even with a very large number of components (2401), performance is generally much worse than the variational mixture filter. Also, the number of components that would be needed quickly becomes infeasible in higher dimensions ($d > 4$). For example, using 7 mixture components per dimension would have required more than 5 million components in 8 dimensions, which exceeds the memory limitations of our setup. Finally, a unimodal variational filter cannot represent the complex posterior shapes properly. We noticed that most of the time, the unimodal version tends to cover all of the posterior modes, although in some cases it could "fall" into a single posterior mode, leaving other ones uncovered.

4.3 Convergence When Using Active Learning

In another set of simulations, we used active learning in order to quickly resolve the uncertainty introduced by ambiguous observations. For this, we optimized successive search locations with respect to their informativeness about the target location (see Appendix A.2 for mathematical details). This was to highlight the importance of multimodal representations in a practical scenario. In these tests we first presented a number of ambiguous samples (3–5), causing the posterior distribution to acquire a complex shape and possibly become multimodal. We then iterated between optimizing the next search location and updating the state distribution after receiving an observation from that location. This optimization crucially depended on the prior state distribution at the current time step. Using active learning should quickly resolve any ambiguities in the state distributions and lead to quick convergence of the posterior distribution to the actual target. Methods better at representing multimodal posteriors should converge more quickly as they should be better at correctly representing the uncertainty in the state space. In this part of the analysis, we did not examine linear filters with a bigger number of components than the variational filter, as the runtime of our active learning framework is quadratic in the number of Gaussian mixture components (see Appendix A.2), and therefore prohibitively slow to use with a lot of individual components.

We examined how well the different algorithms converged onto the target location for both 6D and 10D problems. Fig. 5 shows both the mean squared error (MSE) as well as the root covariance determinant over time[8]. Additionally we plot the log likelihood of the actual target over time. Convergence is indicated by both decreasing error and root covariance determinant, while the log likelihood of the target should increase over time. We found that the variational mixture filter converged well towards the actual target over time, while both the linear and unimodal filters seemed to stall. This means that the active learning component could exploit the multimodal representation of the variational approach

[8] The MSE and mean root covariance determinant were calculated with respect to the overall mean and covariance of the approximate mixture distribution.

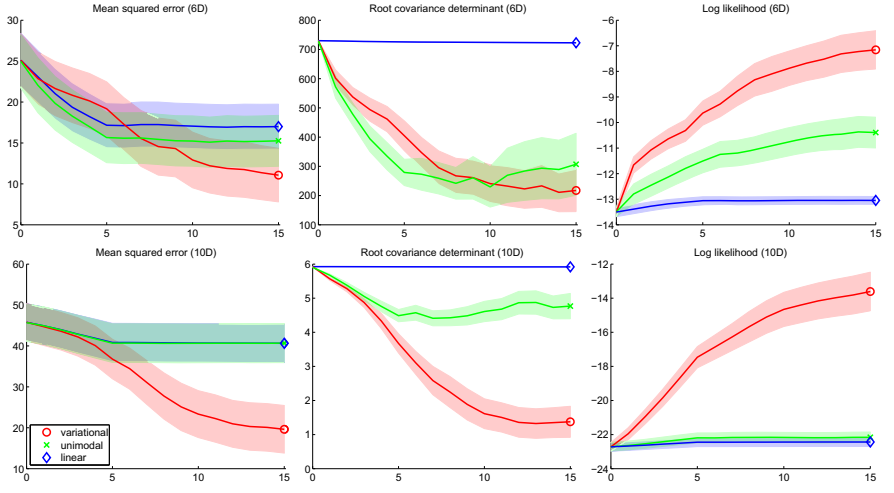

Fig. 5. Convergence results for different algorithms on a 6D (top row of panels) and 10D (bottom row) problem, respectively, with the linear and variational algorithm using 3 components each. Left panels: Average mean squared error over 15 or 20 time steps, respectively, for 25 runs of the variational (red), linear (blue), and unimodal variational (green) algorithms. Shaded regions indicate standard error of the mean. Both multimodal approaches used 4 mixture components. Middle panels: Average covariance determinant over time. Right panels: Log likelihood of the actual target over time.

and resolve the ambiguity about the target location. The representation of the state distribution by the other methods, however, was not sufficient to allow for effective target localization.

As the variational mixture filter is computationally more involved, it is slower in updating the posterior state distribution after receiving an observation due to the numerical optimization involved. We asked whether these delays would have any effects on performance. We therefore set a fixed time span (30–90 seconds), during which each of the methods would iteratively determine the next sample point using active learning, then receive an observation and update its posterior state distribution. The time needed for both optimization of the next search location (i.e. the active learning part) and updating the posterior distributions counted towards each method's time budget. Thus, the faster an algorithm updated its state representation and the lower its number of mixture components, the more observations it was able to request. Fig. 6 plots the number of observations that was used by each method against the log likelihood at the end of the time span for a 3D (left) and a 6D problem (right). In this plot, a marker in the left, upper corner indicates that the respective method was only able to request a small number of observations, but achieving a high log likelihood. Markers in the right, lower corner, however, would indicate that an algorithm requested a high number of observations but failed to increase the log likelihood considerably. We found that the variational mixture filter does

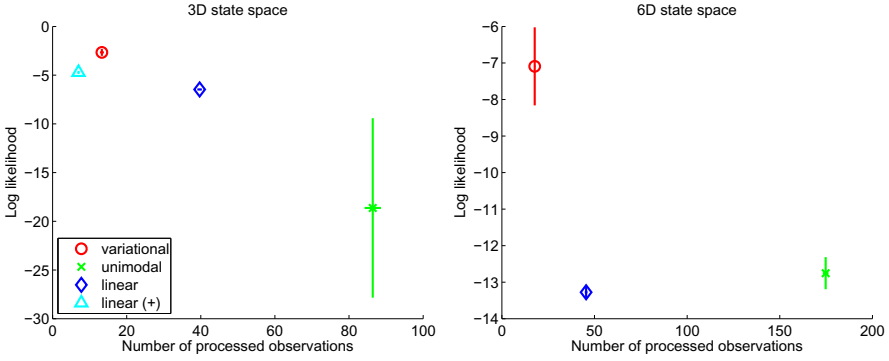

Fig. 6. Log likelihood of target after updates plotted against number of processed observations. Markers denote mean (over 25 runs each) and vertical lines indicate standard errors of the mean. Left: 3D state space. Right: 6D state space. A fixed time budget of 30 and 90 seconds, respectively, was imposed per algorithm. Light blue marker (left plot only): Linear filter with 27 components. Red: Variational mixture filter (3 components). Blue: Linear filter (3 components). Green: Unimodal variational filter.

well in both tasks, by increasing the log likelihood more than other methods, despite being relatively slow and therefore only processing a small number of observations. The linear filter with 27 components also does well in the 3D example, but performs even slower due to increased computational demands in the active learning stage. The variational unimodal filter is generally the fastest, but does not perform well. The extremely high standard error observed is due to its mode-seeking behavior, which caused it to model only a single posterior mode, which often turned out to be the "wrong" one.

5 Discussion

In this paper we have proposed a novel approach to filtering in which the approximate posterior distribution over the state is maintained as a mixture of Gaussians. Using a MoG to approximately represent the posterior makes it possible to capture complex shapes of the true state distribution such as skewedness or multimodality which often arise when the observation function is nonlinear. Unlike previous approaches to mixture filtering we do not maintain a set of independent Gaussian components but take interactions between the mixture components into account when optimizing the approximate representation of the posterior distribution given a new observation. This requires the calculation of expectations over log-sums, which cannot be done analytically, and we propose to approximate these terms using quadrature methods. We find that optimizing the mixture representation directly captures the true shape of the posterior much better than a bank of independent linear filters, even when allowing many more components. We demonstrate the impact of this improved

representation in a task where active learning is used in order to directly resolve the uncertainty in the posterior distribution resulting from ambiguous samples: The proposed approach converges considerably faster and more reliably than the alternative filtering approaches. Importantly, faster convergence is achieved not just when measured as a function of observations but also in terms of overall compute time, despite the fact that our filtering approach is computationally more expensive than the alternatives: The additional computational complexity in processing individual observations by the filter is more than compensated for by the noticeably faster convergence per observation as demonstrated in the experiments with limited overall run-time.

In this paper we have focused on a mixture of Gaussians representation in order to capture multimodal or skewed posterior distributions. Sampling methods like the particle filter have been proposed as an alternative to traditional filtering methods when dealing with nonlinear observation functions and the resulting multimodal state distributions. They work by maintaining a large collection of weighted samples (particles) that provide a sample based representation of the posterior distribution. While such a sample based representation can in principle approximate distributions of any shape (including multi-modal distributions), the number of particles required increases exponentially so that they become impractical in high-dimensional spaces. Furthermore, for many applications a compact representation of the posterior e.g. in terms of a small number of mixture components is crucial: For instance, in the context of the application discussed in this paper, active learning, the sample-based methods that have been proposed so far are very slow even with a relatively small number of particles, which renders them feasible in very low-dimensional spaces only [8,16].

There are several interesting directions for future work. Firstly, our method uses an observation function that is represented as an RBF network. A relatively straightforward extension would be to use a Gaussian Process representation instead [5]. In cases where the observation function is given directly in analytic form, it might not be possible to calculate expectations over this function. In such cases, expectations can be estimated using methods from unimodal Gaussian filters, such as in the extended Kalman filter or in the unscented filter [12]. Secondly, an important practical consideration is the number of mixture components used to represent the posterior. In the experiments described above this number was fixed. Using too many components does not negatively impact the quality of the posterior representation, but it does slow down the algorithm. Using as few components as possible is therefore desirable in order to achieve fast convergence in terms of compute time. On the other hand, choosing the number of components too small will leave some part of the state space unrepresented or requires a single component to cover multiple posterior modes, which will result in a worse representation and in the extreme case can lead to similar problems as for a unimodal filtering approach. One interesting approach would be to dynamically adjust the number of components, adding new components in each observation step, and then merging the most similar ones.

References

1. Abramowitz, M., Stegun, I.A.: Handbook of mathematical function with formulas, graphs, and mathematical tables. U.S. Dept. of Commerce (1972)
2. Alspach, D., Sorenson, H.: Nonlinear bayesian estimation using Gaussian sum approximations. IEEE Trans. Autom. Control 17(4), 439–448 (1972)
3. Arasaratnam, I., Haykin, S., Elliott, R.J.: Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature. Proc. IEEE 95(5), 953–977 (2007)
4. Bouchard, G., Zoeter, O.: Split variational inference. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, pp. 57–64. ACM, New York (2009)
5. Deisenroth, M.P., Huber, M.F., Hanebeck, U.D.: Analytic moment-based Gaussian process filtering. In: ICML (2009)
6. Girard, A., Rasmussen, C.E., Candela, J.Q., Murray-Smith, R.: Gaussian process priors with uncertain inputs—applications to multiple-step ahead time series forecasting. In: NIPS (2003)
7. Goldberger, J., Gordon, S., Greenspan, H.: An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures. In: Proc. IEEE Int. Conf. on Computer Vision (ICCV), vol. 1, pp. 487–493 (2003)
8. Hoffmann, G., Waslander, S., Tomlin, C.: Mutual information methods with particle filters for mobile sensor network control. In: Proc. IEEE Conf. on Decision and Control, pp. 1019–1024 (2006)
9. Huber, M.F., Bailey, T., Durrant-Whyte, H., Hanebeck, U.D.: On entropy approximation for Gaussian mixture random vectors. In: Proc. IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems, pp. 181–188 (2008)
10. Ito, K., Xiong, K.: Gaussian filters for nonlinear filtering problems. IEEE Trans. Autom. Control. 45(5), 910–927 (2000)
11. Jaakola, T.S., Jordan, M.I.: Improving the mean field approximation via the use of mixture distributions. In: Jordan, M.I. (ed.) Learning in Graphical Models, pp. 163–173. Kluwer Academic Publishers, Dordrecht (1998)
12. Julier, S., Uhlmann, J.: Unscented filtering and nonlinear estimation. Proc. IEEE 92(3), 401–422 (2004)
13. Lawrence, N.D., Azzouzi, M.: A variational Bayesian committee of neural networks. Technical report, University of Cambridge, UK (1999)
14. Liu, Q., Pierce, D.A.: A note on Gauss-Hermite quadrature. Biometrika 81(3), 624–629 (1994)
15. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Comput. 3, 246–257 (1991)
16. Saal, H.P., Ting, J., Vijayakumar, S.: Active sequential learning with tactile feedback. In: AISTATS (2010)
17. Torkkola, K.: Feature extraction by non-parametric mutual information maximization. J. Mach. Learn. Res. 3, 1415–1438 (2003)

A Appendix

A.1 Additional Calculations

Usually, expectations over the observation function f have to be approximated, however if f is represented as a RBF as in our case, or as a Gaussian Process [5], then these expectations can be calculated analytically (e.g. [6]), as given below:

$$\mathrm{E}_{q_{mix}}\left[f(\mathbf{x})\right] = \sum_m \sum_j \mathrm{E}_{q_m}\left[c_j k(\mathbf{x}, \mathbf{m}_j)\right] \tag{16}$$

$$\mathrm{E}_{q_m}\left[c_j k(\mathbf{x}, \mathbf{m}_j)\right] = c_j |\mathbf{S}^{-1}\boldsymbol{\Sigma}_m + \mathbf{I}|^{-\frac{1}{2}} \cdot \tag{17}$$

$$\exp\left\{-0.5(\boldsymbol{\mu}_m - \mathbf{m}_j)^T(\boldsymbol{\Sigma}_m + \mathbf{S})^{-1}(\boldsymbol{\mu}_m - \mathbf{m}_j)\right\} \tag{18}$$

$$\mathrm{E}_{q_{mix}}\left[f(\mathbf{x})^2\right] = \sum_m \sum_i \sum_j \mathrm{E}_{q_m}\left[c_i k(\mathbf{x}, \mathbf{m}_i) c_j k(\mathbf{x}, \mathbf{m}_j)\right] \tag{19}$$

$$\mathrm{E}_{q_m}\left[c_i k(\mathbf{x}, \mathbf{m}_i) c_j k(\mathbf{x}, \mathbf{m}_j)\right] = c_i c_j |2\mathbf{S}^{-1}\boldsymbol{\Sigma}_m + \mathbf{I}|^{-\frac{1}{2}} \tag{20}$$

$$\exp\left\{-0.5(\boldsymbol{\mu}_m - \hat{\mathbf{m}}_{ij})^T(\boldsymbol{\Sigma}_m + 0.5\mathbf{S})^{-1}(\boldsymbol{\mu}_m - \hat{\mathbf{m}}_{ij})\right\} \tag{21}$$

A.2 Active Learning

In the active learning scenario, our aim is to pick a search location $\boldsymbol{\theta}$, which maximizes the mutual information between the current state distribution and the expected observation. As the mutual information cannot be calculated analytically when distributions are represented as mixtures of Gaussians, we instead optimize a surrogate measure, called 'quadratic mutual information' (QMI) that has been originally proposed for clustering [17].

$$I_{QMI}(X; Y|\Theta) = \iint \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}\big(p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) - p(\mathbf{x})p(\mathbf{y}|\boldsymbol{\theta})\big)^2 \tag{22}$$

$$= \iint \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}\ p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})^2 + \iint \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}\ p(\mathbf{x})^2 p(\mathbf{y}|\boldsymbol{\theta})^2 - \tag{23}$$

$$2\iint \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}\ p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) p(\mathbf{x}) p(\mathbf{y}|\boldsymbol{\theta}) \tag{24}$$

Each of the integrals involved can now be calculated analytically in a similar fashion as described in Appendix A.1. At each step, I is optimized by gradient ascent with respect to the new search location $\boldsymbol{\theta}$. The computational complexity of this approach is quadratic in the number of mixture components.