DENSITY ASSISTED PARTICLE FILTERS FOR STATE AND PARAMETER ESTIMATION

Petar M. Djurić, Mónica F. Bugallo

Dept. of Electrical and Computer Engineering
Stony Brook University
Stony Brook, NY, 11794, USA
{djuric,monica}@ece.sunysb.edu

Joaquín Míguez

Dept. of Electrónica e Sistemas
Universidade da Coruña
Facultade de Informática,
15071 A Coruña, Spain
jmiguez@udc.es

ABSTRACT

In recent years the theory of particle filtering has continued to advance, and it has found increasing use in sequential signal processing. A weakness of particle filtering is that it is inadequate for problems that besides tracking of evolving states require the estimation of constant parameters. In this paper, we propose particle filters that do not have this limitation. We call these filters density assisted particle filters, of which special cases are the recently introduced Gaussian particle filters and Gaussian sum particle filters. An implementation of a density particle filter is shown on a relatively simple but important nonlinear model. Simulations are included that show the performance of this filter.

1. INTRODUCTION

In the past decade, particle filtering has become an important tool for sequential signal processing. Its advantage over other sequential methods is particularly distinctive in situations where the used models are nonlinear and the involved noise processes are non-Gaussian. The underlying idea of particle filtering is the approximation of densities by random measures, which are represented by samples (particles) from the space of the unknowns and weights associated with the particles. An important feature in the implementation of particle filters (PFs) is that the random measure is recursively updated. With the random measure, one can compute various types of estimates with considerable ease.

The theory of PFs has been well established, and its fundamentals and important applications can be found, for example, in [2] and [7]. A more recent review of the theory and a set of applications in wireless communications are presented in [1]. PFs have three important operations, sampling, weight computation, and resampling. With sampling, one generates a set of new particles that represents the support of the random measure, and with weight computation, one calculates the weights of the particles. Resampling is an important operation because without it PFs yield very poor results. With resampling one replicates the particles that have large weights and removes the ones with negligible weights.

This work was supported by the National Science Foundation under Award CCR-0082607.

This work was also supported by Xunta de Galicia, the Ministry of Science and Technology of Spain and the FEDER funds of the European Union (project TIC2001-0751-C04-01)

The resampling entails a problem that is referred to as particle attrition. It is particularly emphasized in cases when the used models have fixed parameters. As the recursions progress with time, unless special steps are undertaken, the size of the particle set of fixed parameters decreases and very quickly is depleted. This deficiency of the PFs has been recognized and addressed in the past, for example in [3] and [8], and more recently in [6]. In these approaches, the idea is to introduce artificial evolution of the particles and thereby treat them in more or less the same way as the dynamic states of the model.

In two recent companion papers, new class of particle filters has been developed. They have been called Gaussian particle filters (GPFs) and Gaussian sum particle filters (GSPFs). The GPFs approximate the predictive and filtering densities of the PF by Gaussian densities whose parameters are estimated from the particles and their weights. Similarly, the GSPFs approximate these densities with mixtures of Gaussians. The approximating densities can be other than Gaussians or mixtures of Gaussian, and therefore we call these filters Density Asissted Particle Filters (DAPFs). They have a very attractive feature in that they do not use resampling in the sense carried out by standard PFs. This entails appealing advantages when considering hardware implementations of PFs. Another important advantage of these filters is that they do not share the limitation of the standard PFs regarding the estimation of constant model parameters.

In this paper, the emphasis is on DAPFs for combined estimation of evolving states and constant parameters. In Section 2, we describe the general setting of the problem and in Section 3, we briefly comment on PFs and their advantages and disadvantages. Then, in Section 4 we present the DAPFs and outline their basic operations. We make our case about using DAPFs for state and parameter estimation in Section 5, where we elaborate on an interesting example by providing details of the filter's implementation. Finally, in Sections 6 and 7, we show simulation results and make concluding remarks, respectively.

2. GENERAL PROBLEM SETTING

Many dynamic problems can be represented using the state space representation

$$\begin{aligned}\mathbf{x}_t &= \mathbf{f}_t(\mathbf{x}_{t-1}, \boldsymbol{\theta}, \mathbf{u}_t)\\ \mathbf{y}_t &= \mathbf{g}_t(\mathbf{x}_t, \boldsymbol{\theta}, \mathbf{v}_t)\end{aligned}$$

where $\mathbf{x}_t$ is a state vector, $\mathbf{y}_t$ is a vector of observations, $\mathbf{u}_t$ and $\mathbf{v}_t$ are noise vectors, $\boldsymbol{\theta}$ is a vector of unknown parameters, $\mathbf{f}_t(\cdot)$ is a system transition function, $\mathbf{g}_t(\cdot)$ is a measurement function, and the subscript t denotes time index, where $t \in \mathbb{Z}^+$. These two equations are known as state and observation equations. The analytical forms of the functions and the distributions of the two noise vectors are known. Based on the observations $\mathbf{y}_t$ and the assumptions, the objective is to estimate $\mathbf{x}_t$ *and* $\boldsymbol{\theta}$ recursively. In Bayesian context, this amounts to updating important densities of interest like the filtering density at time $t-1$, $p(\mathbf{x}_{t-1}, \boldsymbol{\theta}|\mathbf{y}_{0:t-1})$, to $p(\mathbf{x}_t, \boldsymbol{\theta}|\mathbf{y}_{0:t})$, where the notation $\mathbf{y}_{0:t}$ signifies $\{\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_t\}$.

3. PARTICLE FILTERING

Particle filtering is a methodology that allows for sequential processing of data by recursive updating of the densities of interest. These densities are the filtering, predictive or smoothing densities of the unknowns. In particle filtering they are all approximated by random measures composed of particles (samples from the state and parameter spaces of the unknowns) and weights. Let $\chi_t = \{\mathbf{x}_t^{(m)}, \boldsymbol{\theta}_t^{(m)}, w_t^{(m)}\}_{m=1}^M$ be the random measure at time t, $\mathbf{x}_t^{(m)}$ and $\boldsymbol{\theta}_t^{(m)}$ the particles of $\mathbf{x}_t$ and $\boldsymbol{\theta}$, respectively, and $w_t^{(m)}$ their associated weights. Note that here we denoted the particles of $\boldsymbol{\theta}$ with a subscript t, which does not mean that these parameters evolve dynamically, but simply that they may represent a different set at t from the one at $t-1$.

In updating the random measures with every new measurement, PFs perform three important operations, (1) sampling, (2) importance computation, and (3) resampling. The sampling operation represents drawing particles from a proposal function, the importance computation assigning weights to the particles, and resampling, replicating particles with dominating weights at the expense of particles that have negligible weights. An important feature of PFs is that they can perform tracking of dynamic state variables with ease as opposed to estimation of fixed parameters. In the latter situations, if the fixed parameters are nuisance parameters, their integration, if possible, is the best solution. In many important cases, however, the integration is analytically intractable. In other problems, the fixed parameters may be of significance, and their estimation of utmost importance.

In problems with constant parameters, one may enforce artificial evolution of the parameters and apply a standard PF [3], use the kernel smoothing procedure from [8], or exploit the auxiliary particle filtering based method from [6]. A common feature of these methods is that they impose artificial evolution of the fixed parameters. Here we propose to use DAPF which can cope with constant parameters more naturally than the above methods.

4. DENSITY ASSISTED PARTICLE FILTERING

Recently, new types of PFs have been proposed, referred to as Gaussian particle filters (GPFs) [4]. These PFs approximate the predictive, $p(\mathbf{x}_t|\mathbf{y}_{0:t-1})$, and filtering, $p(\mathbf{x}_t|\mathbf{y}_{0:t})$, densities by Gaussians whose mean vectors and covariance matrices are computed from the particles. Suppose that at time $t-1$ we approximate the filtering density by $\mathcal{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$. Then, the steps of a simple implementation of the GPF are as follows:

1. Draw particles according to $\mathbf{x}_{t-1}^{(m)} \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$.
2. Draw particles according to $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$.
3. Compute the weights of the particles by
$$\tilde{w}_t^{(m)} = p(\mathbf{y}_t|\mathbf{x}_t^{(m)}).$$
4. Normalize the weights by
$$w_t^{(m)} = \frac{\tilde{w}_t^{(m)}}{\sum_{j=1}^M \tilde{w}_t^{(j)}}.$$
5. Estimate $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ by
$$\begin{aligned} \boldsymbol{\mu}_t &= \sum_{m=1}^M w_t^{(m)} \mathbf{x}_t^{(m)} \\ \boldsymbol{\Sigma}_t &= \sum_{m=1}^M w_t^{(m)} \left(\mathbf{x}_t^{(m)} - \boldsymbol{\mu}_t\right)\left(\mathbf{x}_t^{(m)} - \boldsymbol{\mu}_t\right)^\top \end{aligned}$$
which are the parameters of the Gaussian density that approximates $p(\mathbf{x}_t|\mathbf{y}_{0:t})$.

A distinctive feature of GPFs is that they do not require resampling, which is important in hardware implementation because the resampling operation complicates hardware architectures. The resampling in GPFs is replaced by sampling from a Gaussian, which as a procedure is much simpler. In a companion paper, [5], the approximating densities are modeled as mixture Gaussians that provide more flexibility in capturing the shapes of the predictive and filtering densities.

In general, as noted in [4], the approximating densities can be any appropriate parametric densities. We refer to PFs that exploit predefined approximating densities as to density assisted particle filters. If we denote the approximating density of $p(\mathbf{x}_t|\mathbf{y}_{0:t-1})$ by $p_f(\boldsymbol{\phi}_t)$, where $\boldsymbol{\phi}_t$ are the parameters of this density, the steps of the DAPF are the following:

1. Draw particles according to $\mathbf{x}_{t-1}^{(m)} \sim p_f(\boldsymbol{\phi}_{t-1})$.
2. Draw particles according to $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$.
3. Compute the weights of the particles by
$$\tilde{w}_t^{(m)} = p(\mathbf{y}_t|\mathbf{x}_t^{(m)}).$$
4. Normalize the weights by
$$w_t^{(m)} = \frac{\tilde{w}_t^{(m)}}{\sum_{j=1}^M \tilde{w}_t^{(j)}}.$$
5. Estimate the parameters $\boldsymbol{\phi}_t$ from $\mathbf{x}_t^{(m)}$ and $w_t^{(m)}$, $m = 1, 2, \cdots, M$.

We point out here that DAPFs do not share the problem of standard PFs regarding constant parameters. This is clear because in the first step when the particles are drawn from $p_f(\mathbf{x}_{t-1}, \boldsymbol{\theta}_{t-1}|\mathbf{y}_{0:t-1})$ there is a natural evolution of the particles of $\boldsymbol{\theta}_{t-1}$. We present the implementation details of DAPF on models that have fixed parameters by way of an example.

5. EXAMPLE

We investigate a very specific and rather simple but important problem. It can be described as follows:

$$x_t = ax_{t-1} + u_t \quad (1)$$
$$y_t = bx_t + v_t \quad (2)$$

where $0 < a < 1$, $b, x_t \in \mathbb{R}$, and u_t and v_t are independent white Gausian noises with zero mean and variances σ_u^2 and σ_v^2, respectively. The unknowns of interest are the evolving state x_t and the parameters a and b. Thus, in our previous notation, $\boldsymbol{\theta} = [a\ b]^\top$.

The density of interest is the posterior

$$p(a_t, b_t, x_t|y_{0:t}) = p(x_t, b_t|a_t, y_{0:t})p(a_t|y_{0:t}).$$

We approximate it by using

$$p(a_t, b_t, x_t|y_{0:t}) \simeq \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\mathcal{B}e(\alpha_t, \beta_t) \quad (3)$$

where the first density on the right hand is bivariate Gaussian and the second is *beta* density. The predictive density $p(a_t, b_t, x_t|y_{0:t-1})$ is approximated by

$$p(a_t, b_t, x_t|y_{0:t-1}) \simeq 1/M \sum_{m=1}^{M} p(x_t|x_{t-1}^{(m)})\delta(b_t - b_{t-1}^{(m)})\delta(a_t - a_{t-1}^{(m)})$$

where $\delta(\cdot)$ denotes the Dirac delta function. A straightforward application of the DAPF has the following steps:

1. Draw $a_{t-1}^{(m)} \sim \mathcal{B}e(\alpha_{t-1}, \beta_{t-1})$.
2. Draw $(x_{t-1}^{(m)}, b_{t-1}^{(m)}) \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$.
3. Draw $x_t^{(m)} \sim p(x_t|x_{t-1}^{(m)}, a_{t-1}^{(m)})$.
4. Let $a_t^{(m)} = a_{t-1}^{(m)}$ and $b_t^{(m)} = b_{t-1}^{(m)}$.
5. Compute the weights of the generated particles by
$$\tilde{w}_t^{(m)} = p(y_t|x^{(m)}).$$
6. Normalize the weights $\tilde{w}_t^{(m)}$.
7. Compute the parameters of the approximating densities.

Before we explain the generation of particles in steps 1 and 2, we note that the *beta* density is defined as

$$p(a) = \frac{1}{B(\alpha, \beta)} a^{\alpha-1}(1-a)^{\beta-1} I_{(0,1)}(a)$$

where $\alpha > 0$ and $\beta > 0$. The mean and variance of a are given by

$$\mu_a = \frac{\alpha}{\alpha + \beta}$$

$$\sigma_a^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}.$$

If we know the mean and the variance of a, we can estimate α and β from

$$\alpha = \frac{\mu_a^2(1-\mu_a)}{\sigma_a^2} - \mu_a \quad (4)$$
$$\beta = \left(\frac{\mu_a(1-\mu_a)}{\sigma_a^2} - 1\right)(1-\mu_a). \quad (5)$$

This suggests that it is straightforward to generate particles from a *beta* density whose first two moments are predefined.

The generation of the particles $a_{t-1}^{(m)}, b_{t-1}^{(m)}$, and $x_{t-1}^{(m)}$, in steps 1 and 2 proceeds as follows. When at time $t-1$ the computation of particle weights in step 6 is completed, we estimate the mean vector and the covariance matrix of the filtering density given by

$$\widetilde{\boldsymbol{\mu}}_{t-1} = \begin{bmatrix} \mu_{t-1,x} \\ \mu_{t-1,b} \\ \mu_{t-1,a} \end{bmatrix}$$

and

$$\widetilde{\boldsymbol{\Sigma}}_{t-1} = \begin{bmatrix} \sigma_{t-1,xx}^2 & \rho_{t-1,xb} & \rho_{t-1,xa} \\ \rho_{t-1,xb} & \sigma_{t-1,bb}^2 & \rho_{t-1,ba} \\ \rho_{t-1,xa} & \rho_{t-1,ba} & \sigma_{t-1,aa}^2 \end{bmatrix}.$$

First we draw $a_{t-1}^{(m)}$ from $\mathcal{B}e(\alpha_{t-1}, \beta_{t-1})$ whose parameters are estimated from (4) and (5), where $\mu = \mu_{t-1,a}$ and $\sigma^2 = \sigma_{t-1,aa}^2$. The particles $x_{t-1}^{(m)}$ and $b_t^{(m)}$ are obtained from

$$x_{t-1}^{(m)} = A_{t-1,1}z_{t-1,1}^{(m)} + B_{t-1,1}z_{t-1,2}^{(m)} + C_{t-1,1}a_{t-1}^{(m)} + D_{t-1,1}$$

$$b_{t-1}^{(m)} = A_{t-1,2}z_{t-1,1}^{(m)} + B_{t-1,2}z_{t-1,2}^{(m)} + C_{t-1,2}a_{t-1}^{(m)} + D_{t-1,2}$$

where $z_{t-1,1}$ and $z_{t-1,2}$ are independent standard Gaussian random variables, which are also independent from $a_{t-1}^{(m)}$, and the coefficients $A_{t-1,i}, B_{t-1,i}, C_{t-1,i}, D_{t-1,i}, i = 1, 2$ are chosen so that the mean and covariance matrix of $[x_{t-1} \ \ b_{t-1} \ \ a_{t-1}]^\top$ are preserved. It can be shown that the following set of coefficients can be used:

$$\begin{aligned}
A_{t-1,1} &= \pm\sqrt{\sigma_{t-1,xx}^2 - C_{t-1,1}^2\sigma_{t-1,aa}^2} \\
A_{t-1,2} &= \frac{\rho_{t-1,xb} - C_{t-1,1}C_{t-1,2}\sigma_{t,aa}^2}{A_{t-1,1}} \\
B_{t-1,1} &= 0 \\
B_{t-1,2} &= \pm\sqrt{\sigma_{t-1,bb}^2 - C_{t-1,2}^2\sigma_{t-1,aa}^2 - A_{t-1,2}^2} \\
C_{t-1,1} &= \frac{\rho_{t-1,xa}}{\sigma_{t-1,aa}^2} \\
C_{t-1,2} &= \frac{\rho_{t,ba}}{c_{t-1,aa}} \\
D_{t-1,1} &= \mu_{t-1,x} - \mu_{t-1,a}C_{t-1,1} \\
D_{t-1,2} &= \mu_{t-1,b} - \mu_{t-1,a}C_{t-1,2}.
\end{aligned}$$

In summary one first generates $a_{t-1}^{(m)}$, followed by drawing $z_{t-1,1}$ and $z_{t-1,2}$ from the standard normal distribution, followed by computing $x_{t-1}^{(m)}$ and $b_{t-1}^{(m)}$ using the above transformation.

6. SIMULATION RESULTS

In this section we provide simulation results obtained for the problem described in the previous section. The parameters of the model were $a = 0.9$ and $b = 2$. The noise variances σ_u^2 and σ_v^2 were set to one. There were 100 observations available, and the number of particles was $M = 200$. The initial particles were drawn as follows: $x_0^{(m)} \sim \mathcal{N}(x_1, 3)$, $b_0^{(m)} \sim \mathcal{N}(1, 2)$, and $a_0^{(m)} \sim \mathcal{U}(0, 1)$.

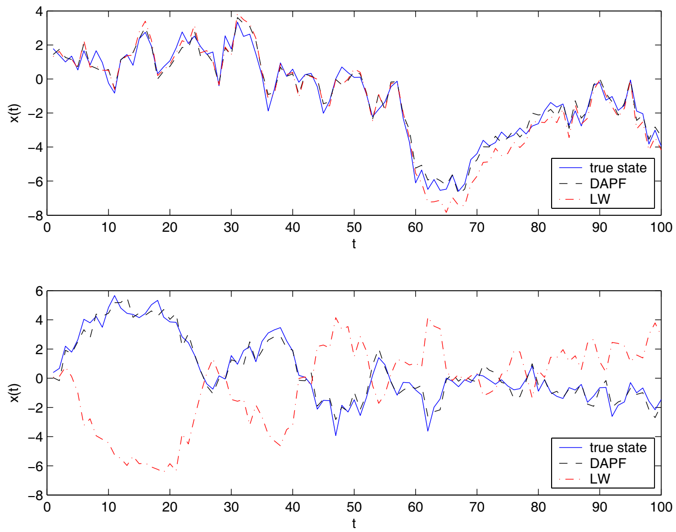

Fig. 1. Tracking plots of two different realizations. In the second plot, the LW method switched to estimating the mirror state trajectory.

We compared the proposed method with the one of Liu and West described in [6]. For that method, we used the same priors for generation of initial particles. It should be noted that this problem is inherently ambiguous due to the product bx_t in the observation equation. For some realizations, the methods can switch to tracking the mirror trajectory of x_t, $-x_t$, rather than x_t, and estimate $-b$ instead of b. We point out that this problem was much more rarely observed with the proposed method than with that of Liu and West. Sample trackings of the methods are presented in Figure 1.

In Figure 2, we observe the mean square errors (MSEs) of the estimated states and parameters of the two methods as functions of time. These estimates were obtained from 50 realizations.

7. CONCLUSIONS

In this paper we described a special class of particle filters which we call density assisted particle filters. Their main feature is that they approximate the filtering density with a predefined parametric density. We showed an important advantage of DAPFs over standard PFs in problems where the addressed models have constant parameters. With a judicious choice of approximating densities, one can develop DAPFs that handle the problem of fixed parameters with ease and that yield excellent performance.

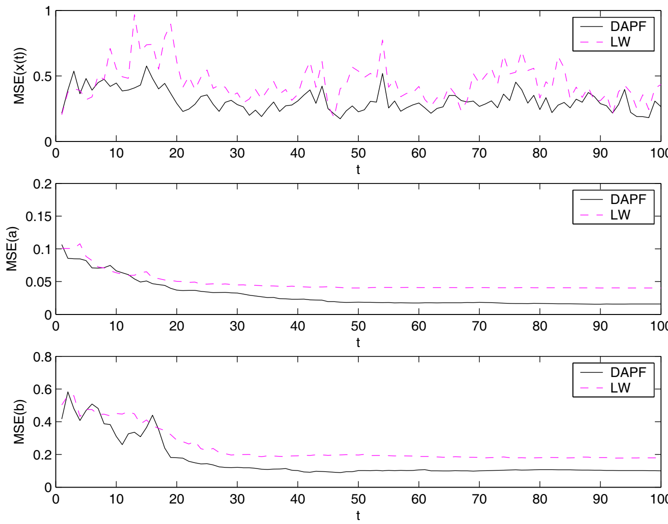

Fig. 2. Comparison of the MSEs of the states and the parameters between DAPF and the LW method.

8. REFERENCES

[1] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, "Particle filtering: A review of the theory and how it can be used for solving problems in wireless communications," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[2] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer, New York, 2001.

[3] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings-F*, vol. 140, no. 2, pp. 107–113, 1993.

[4] J. Kotecha and P. M. Djurić, "Gaussian particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, 2003.

[5] J. Kotecha and P. M. Djurić, "Gaussian sum particle filtering," *IEEE Transactions on Signal Processing*, vol. 10, no. 10, pp. 2602–2612, 2003.

[6] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., pp. 197–223. Springer, 2001.

[7] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, New York, 2001.

[8] M. West, "Mixture models, Monte Carlo, Bayesian updating and dynamic models," *Computer Science and Statistics*, vol. 24, pp. 325–333, 1993.