

Particle Filtering with Progressive Gaussian Approximations to the Optimal Importance Density

Pete Bunch

Signal Processing and Communications Lab.
Cambridge University Engineering Dept.
Cambridge, UK,
Email: pb404@cam.ac.uk

Simon Godsill

Signal Processing and Communications Lab.
Cambridge University Engineering Dept.
Cambridge, UK,
Email: sjg30@cam.ac.uk

Abstract—A new algorithm, the progressive proposal particle filter, is introduced. The performance of a standard particle filter is highly dependent on the choice of importance density used to propagate the particles through time. The conditional posterior state density is the optimal choice, but this can rarely be calculated analytically or sampled from exactly. Practical particle filters rely on forming approximations to the optimal importance density, frequently using Gaussian distributions, but these are not always effective in highly nonlinear models. The progressive proposal method introduces the effect of each observation gradually and incrementally modifies the particle states so as to achieve an improved approximation to the optimal importance distribution.

I. INTRODUCTION

Particle filters are an established class of algorithms for approximating the Bayesian filtering distribution of a latent state in a sequential model (See [1], [2].) Their mechanism is to propagate a set of weighted samples, or “particles”, through time, distributed approximately according to the desired filtering distribution. The particles are typically propagated by importance sampling; a new state is sampled from an importance density and assigned a weight proportional to the ratio of the filtering and importance densities.

The efficacy of a particle filter depends substantially on the choice of importance density. If it is well matched to the filtering density then the resulting particles will have even weights, and the filter works well. On the other hand, if there is a mismatch between the densities, the population is likely to be dominated by a small proportion of particles with high weights, and the filter works poorly.

The optimal choice for the importance density — in the sense of minimising the weight variance — is the conditional posterior density of the state [3]. However, this can rarely be sampled from or evaluated exactly. Instead, it is common to use approximations, for example by choosing some appropriately adapted Gaussian density [3], [4]. For many mildly nonlinear models such an approximation has proven effective, but as the nonlinearity and dimensionality increase, and the modes of the optimal importance density become less and less Gaussian, performance becomes worse.

In this paper, we introduce a new mechanism for carrying out the importance sampling steps in a particle filter. The effect of each new observation is introduced gradually, and a new state is set for each incremental step using a local Gaussian

approximation. Since each final particle state is reached via a series of intermediate steps, we name this algorithm the progressive proposal particle filter (PPPF). By using small steps and adapting the approximations to the current state, a more accurate approximation to the optimal importance density is achieved, and the resulting filter proves more effective than competitors using a single Gaussian as the importance density.

The concept of introducing the likelihood progressively has been used before, for assumed density filtering [5], [6], annealed and progressively corrected particle filters [7]–[10], and for particle flow filters [11]. The new algorithm, whilst sharing this idea, is distinct from all of these.

II. PARTICLE FILTERING

We consider a standard discrete-time HMM in which the transition, observation and prior models have closed-form densities,

$$\begin{aligned}x_t &\sim f(x_t|x_{t-1}) \\ y_t &\sim g(y_t|x_t) \\ x_1 &\sim p(x_1),\end{aligned}$$

where the random variable x_t is the hidden state of a system at time t , and y_t is a partial, noisy observation. We assume here that the transition, observation and prior densities may be evaluated and that the prior and transition densities may be sampled. A particle filter is used to estimate recursively distributions over the path of the state variables, $x_{1:t} = \{x_1, \dots, x_t\}$. Densities are approximated by a sum of weighted probability masses located at a discrete set of states,

$$p(x_{1:t}|y_{1:k}) = \sum_i \bar{w}_t^{(i)} \delta_{x_{1:t}^{(i)}}(x_{1:t}),$$

where $\delta_{x_{1:t}^{(i)}}(x_{1:t})$ denotes a unit probability mass at the point $x_{1:t}^{(i)}$, and the weights sum to 1. The particle filter recursion may be separated into two stages — prediction and update — which produce approximations to the predictive density, $p(x_{1:t}|y_{1:t-1})$, and the filtering density, $p(x_{1:t}|y_{1:t})$, respectively.

For brevity here, we consider only the simplest algorithm, in which particle selection is not optional and “auxiliary” methods [12] are not used. At time t , the prediction stage begins with selection (resampling) of a parent from amongst the $t - 1$ particles; an index, a_j , is chosen with probability

$\bar{w}_{t-1}^{(j)}$. Next, a new state $x_t^{(j)}$ is sampled from an importance density, $q(x_t|x_{t-1}^{(a_j)}, y_t)$, and concatenated to the parent path to form the new particle, $x_{1:t}^{(j)} \leftarrow \{x_{1:t-1}^{(a_j)}, x_t^{(j)}\}$. Finally, an importance weight is assigned to the particle to account for the discrepancy between importance and target distributions,

$$\hat{w}_t^{(j)} = \frac{p(x_{1:t}^{(j)}|y_{1:t-1})}{p(x_{1:t-1}^{(a_j)}|y_{1:t-1})q(x_t^{(j)}|x_{t-1}^{(a_j)}, y_t)} \propto \frac{f(x_t^{(j)}|x_{t-1}^{(a_j)})}{q(x_t^{(j)}|x_{t-1}^{(a_j)}, y_t)}.$$

In the update stage, the same set of particles is used to approximate the filtering distribution. Since these are currently distributed according to,

$$\eta(x_{1:t}) = p(x_{1:t-1}|y_{1:t-1})q(x_t|x_{t-1}, y_t),$$

a new importance weight is required to account for the discrepancy,

$$w_t^{(j)} = \frac{p(x_{1:t}^{(j)}|y_{1:t})}{\eta(x_{1:t}^{(j)})} \propto \hat{w}_t^{(j)} \times g(y_t|x_t^{(j)}).$$

Finally, the weights are normalised, $\bar{w}_t = w_t^{(j)} / \sum_i w_t^{(i)}$.

If the transition density is used as the importance density then the resulting algorithm is the “bootstrap filter” of [13]. This often works poorly, since the regions of high observation density may not significantly overlap with the regions of high transition density, particularly if the observation noise variance is low. The optimal importance density (OID) is the conditional state posterior,

$$q(x_t|x_{t-1}^{(a_j)}, y_t) = p(x_t|x_{t-1}^{(a_j)}, y_t). \quad (1)$$

However, this density may rarely be sampled exactly or evaluated analytically. For other models, it is common to use Gaussian approximations of (1) based on either linearisation or the unscented transform [3], [4]. These work well when the OID is unimodal, and the observation nonlinearity is weak, but can otherwise perform poorly.

III. PROGRESSIVE PROPOSALS

The progressive proposal method is a procedure for sampling approximately from the OID by introducing the likelihood progressively and making a series of local Gaussian approximations. In order to achieve this, an auxiliary variable $\lambda \in [0, 1]$ is used. Intuitively, this is a stretch of “pseudo-time” between the predictive and filtering densities, which we link via a continuous sequence of densities,

$$\tilde{\pi}_{t,\lambda}(x_{1:t-1}, x_{t,\lambda}) \propto g(y_t|x_{t,\lambda})^\lambda f(x_{t,\lambda}|x_{t-1})p(x_{1:t-1}|y_{1:t-1}), \quad (2)$$

in which $x_{t,\lambda}$ is defined as the state at time t and pseudo-time λ . This filtering sequence includes the predictive density when $\lambda = 0$ and the desired filtering density when $\lambda = 1$.

State updates are derived by considering a related sequence of optimal importance densities for each particle,

$$\pi_{t,\lambda}(x_{t,\lambda}|x_{t-1}^{(a_j)}) \propto g(y_t|x_{t,\lambda})^\lambda f(x_{t,\lambda}|x_{t-1}^{(a_j)}). \quad (3)$$

This sequence begins with the transition density at $\lambda = 0$ and finishes with the OID at $\lambda = 1$.

The progressive proposal procedure begins by sampling a state for each particle from the transition density and assigning a predictive weight, in exactly the same manner as the bootstrap filter. A series of incremental updates are then applied in order to advance each particle state $x_{t,\lambda}^{(j)}$ and its associated weight $w_{t,\lambda}^{(j)}$ through pseudo-time, so that it is correctly distributed according to (2) throughout.

From here on, subscript t is omitted for clarity on variables which vary with λ . Particle superscripts are also omitted where it is unambiguous.

A. State Updates

There is one class of models for which the OID has an analytic form, those which have a linear observation function and Gaussian transition and observation densities; the transition function need not be linear,

$$\begin{aligned} f(x_t|x_{t-1}) &= \mathcal{N}(x_t|\phi(x_{t-1}), Q) \\ g(y_t|x_t) &= \mathcal{N}(y_t|Hx_t, R). \end{aligned}$$

For such models, the OID sequence (3) is,

$$\pi_\lambda(x_\lambda|x_{t-1}) = \mathcal{N}(x_\lambda|\mu_\lambda, \Sigma_\lambda),$$

$$\begin{aligned} \Sigma_\lambda &= [Q^{-1} + \lambda H^T R^{-1} H]^{-1} \\ \mu_\lambda &= \Sigma_\lambda [Q^{-1} \phi(x_{t-1}) + \lambda H^T R^{-1} y_t]. \end{aligned}$$

Since the OID may be sampled and the density evaluated, a progressive proposal is redundant. However, the analytic formulas derived from this case may be used with other classes of models if local Gaussian approximations are made.

A Gaussian random variable may be written as a linear transformation of an underlying standard Gaussian random variable,

$$\begin{aligned} x_\lambda &= \mu_\lambda + \Sigma_\lambda^{\frac{1}{2}} z_\lambda \\ z_\lambda &\sim \mathcal{N}(\cdot|0, I), \end{aligned}$$

where $\Sigma_\lambda^{\frac{1}{2}}$ is the principal square root of the covariance matrix. By assuming that the underlying random variable z_λ is fixed over pseudo-time, we arrive at an update formula for the latent state from λ_0 to λ_1 ,

$$x_{\lambda_1} = \mu_{\lambda_1} + \Sigma_{\lambda_1}^{\frac{1}{2}} \Sigma_{\lambda_0}^{-\frac{1}{2}} (x_{\lambda_0} - \mu_{\lambda_0}). \quad (4)$$

For partially linear-Gaussian models, advancing the state through pseudo-time using (4) generates a particle which is distributed exactly according to the current density in the OID sequence. For other models, such exact analytic updates do not exist. However, in the same manner as for nonlinear extensions of the Kalman filter, we can use the Gaussian update formulas if a local approximation to the OID sequence is first formed,

$$\hat{\pi}_{\lambda|x^*}(x_\lambda|x_{t-1}) = \mathcal{N}\left(x_\lambda \middle| \hat{\mu}_{\lambda|x^*}, \hat{\Sigma}_{\lambda|x^*}\right),$$

where $\hat{\mu}_{\lambda|x^*}$ and $\hat{\Sigma}_{\lambda|x^*}$ are the mean and covariance of the Gaussian approximation formed at the point x^* , which are themselves functions of λ . Such approximations may be

```

1: for  $t = 1, 2, \dots$  do
2:   for  $i = 1, \dots, N_F$  do
3:     if  $t > 1$  then
4:       Select an ancestor,  $a_i = j$ , with probability  $\bar{w}_{t-1}^{(j)}$ 
5:       Initialise state by sampling from the transition density,  $x_{t,0}^{(i)} \sim f(x_t | x_{t-1}^{(a_i)})$ .
6:     else
7:       Initialise state by sampling from the prior density,  $x_{t,0}^{(i)} \sim p(x_t)$ .
8:     end if
9:     Initialise pseudo-time,  $\lambda = 0$ .
10:    Initialise weight,  $w_{t,0}^{(i)} = 1$ .
11:    while  $\lambda < 1$  do
12:      Increment pseudo-time,  $\lambda \leftarrow \lambda + \delta\lambda$  (see text).
13:      Update state  $x_{t,\lambda}^{(i)}$  using (4).
14:      Update weight  $w_{t,\lambda}^{(i)}$  using (??).
15:    end while
16:    Finalise,  $x_t^{(i)} = x_{t,1}^{(i)}$ ,  $w_t^{(i)} = w_{t,1}^{(i)}$ .
17:  end for
18:  Normalise weights,  $\bar{w}_t = w_t^{(i)} / \sum_j w_t^{(j)}$ .
19: end for

```

TABLE I. PROGRESSIVE PROPOSAL PARTICLE FILTER

formed by linearisation or Taylor series truncation of the log-density, as in for example [3], [12]. A new approximation is formed for each incremental update step.

The length of pseudo-time increments may be set at a fixed step size or they may vary according to some pre-determined scheme. Due to the geometric nature of the density sequences, states tend to move more rapidly for λ close to 0, so the latter scheme, with smaller step sizes early on, is preferable. It is possible to vary step sizes adaptively according to the quality of the Gaussian approximation at each point.

B. Weight Updates

Suppose we have a particle $\{x_{1:t-1}, x_{\lambda_0}\}$ with weight w_{λ_0} distributed as $\tilde{\pi}_{\lambda_0}$. If a new state x_{λ_1} is generated using (4) in combination with a Gaussian approximation of π_{λ_0} , and the old state x_{λ_0} discarded, then the distribution of the unweighted particle may be updated using the standard change of variable approach for a probability density,

$$\eta_{\lambda_1}(x_{1:t-1}, x_{\lambda_1}) = \eta_{\lambda_0}(x_{1:t-1}, x_{\lambda_0}) \times \left| \frac{\partial x_{\lambda_0}}{\partial x_{\lambda_1}} \right|.$$

Hence, the weight update is,

$$w_{\lambda_1} = \frac{\tilde{\pi}_{\lambda_1}(x_{1:t-1}, x_{\lambda_1})}{\eta_{\lambda_1}(x_{1:t-1}, x_{\lambda_1})} \propto w_{\lambda_0} \times \frac{g(y_t | x_{\lambda_1})^{\lambda_1} f(x_{\lambda_1} | x_{t-1})}{g(y_t | x_{\lambda_0})^{\lambda_0} f(x_{\lambda_0} | x_{t-1})} \times \sqrt{\frac{\hat{\Sigma}_{\lambda_1 | x_{\lambda_0}}}{\hat{\Sigma}_{\lambda_0 | x_{\lambda_0}}}}. \quad (5)$$

C. Algorithm Summary

The PPPF is summarised in table ??.

The complexity of the PPPF at time t is $\mathcal{O}(N_F M)$, where N_F is the number of particles and M the number of update steps.

IV. SIMULATIONS

Numerical testing using simulated data are presented to demonstrate the efficacy of the PPPF. Algorithm performance is assessed through two measures. The accuracy of each

filter is measured by a root-mean-square error (RMSE) value, calculated using the weighted average of the particle states as a point estimate. The quality of the particle population is measured using the average effective sample size (ESS),

$$N_{E,t} = \left[\sum_i \bar{w}_t^{(i)2} \right]^{-1},$$

which takes a value between 1 and N_F .

The PPPF is compared with the following particle filters and importance densities:

- A bootstrap filter (BF), using the transition density.
- An extended particle filter (EPF), using a Gaussian density chosen by linearisation about the predictive mean, in the style of an extended Kalman filter.
- An unscented particle filter (UPF), using a Gaussian density chosen using the unscented transform, in the style of an unscented Kalman filter.
- An optimal Gaussian importance particle filter (OGIPF), using a Gaussian density chosen by truncation of the Taylor series of the log of the unnormalised OID around a local maximum [3]. Gradient ascent is used to locate the maximum.

The number of particles used by each algorithm was selected so that the running times were roughly equal. The PPPF used $M = 10$ steps per time frame, with smaller step sizes close to $\lambda = 0$.

A. The Model

We consider tracking a small aircraft over a mapped landscape. Time of flight and Doppler measurements from a radio transmitter on the aircraft provide accurate measurements of range r_t , and range rate s_t to a reference station, but only a low resolution measurement of bearing θ_t . In addition, accurate measurements are made of the height above the ground h_t . The profile of the terrain (i.e. the height of the ground above a datum at each point) has been mapped.

At t , the latent state for our model is,

$$x_t = [p_t^T \quad v_t^T]^T,$$

where p_t and v_t are the 3-dimensional position and velocity of the aircraft respectively, and the observation is,

$$y_t = [\theta_t \quad r_t \quad h_t \quad s_t]^T.$$

The observation function is comprised of,

$$\begin{aligned} \theta_t &= \arctan\left(\frac{p_{t,1}}{p_{t,2}}\right) & r_t &= \sqrt{p_{t,1}^2 + p_{t,2}^2 + p_{t,3}^2} \\ h_t &= p_{t,3} - T(p_{t,1}, p_{t,2}) & s_t &= \frac{p_t \cdot v_t}{r_t}, \end{aligned}$$

where $T(p_{t,1}, p_{t,2})$ is the terrain height at the corresponding horizontal coordinates. The four measurements are independent and the respective variances are $(\frac{\pi}{9})^2$, 0.1^2 , 0.1^2 , 0.1^2 .

A linear transition model is used, based on the near-constant velocity model with Gaussian innovations,

$$f(x_t | x_{t-1}) = \mathcal{N}(x_t | Fx_{t-1}, Q),$$

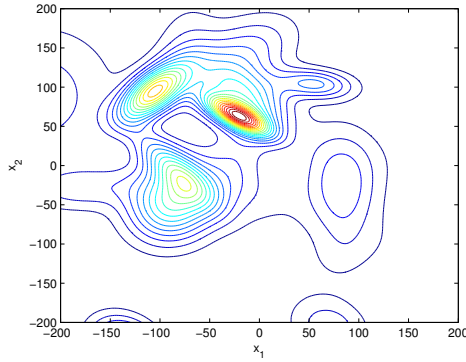


Fig. 1. Contour plot of an example simulated terrain map.

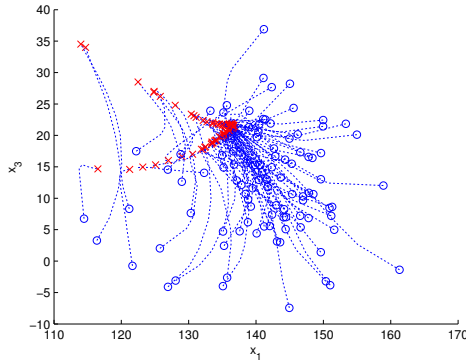


Fig. 2. An example of the PPPF particle motion running on the terrain tracking model, showing one horizontal and the vertical state component. Prior states are shown with circles and posterior states with crosses.

$$F = \begin{bmatrix} I & I \\ 0 & I \end{bmatrix} \quad Q = 10 \begin{bmatrix} \frac{1}{3}I & \frac{1}{2}I \\ \frac{1}{2}I & I \end{bmatrix}.$$

For the simulations presented here, the terrain profile was modelled as a mixture of randomly-generated Gaussian blobs. An example is shown in figure 1.

B. Results

The accurate measurements of range, range rate and height constrain the region of high posterior probability to lie on a 3 dimensional subspace, which can take some unusual shapes (see figure 2). This means that particle filters using simple Gaussian importance densities do not perform well — the EPF diverges completely and produces no useful results. Furthermore, the optimal Gaussian importance density method performs poorly as the maximisation procedure struggles with the narrow mode.

Figure 2 shows the motion of the particles from the PPPF on a typical frame, and the awkward shape of the posterior mode. Table II shows the average ESSs and RMSEs for each algorithm over 100 simulated data sets, each of 100 time steps.

Repeating the tests with an equal number of particles for each algorithm, the PPPF still yielded by far the highest average ESS.

TABLE II. ALGORITHM PERFORMANCE RESULTS ON THE TERRAIN TRACKING MODEL.

| Algorithm | N_F | ESS | RMSE |
|-----------|-------|------|------|
| BF | 6000 | 1.0 | 78.6 |
| UPF | 460 | 2.4 | 70.2 |
| OGIPF | 10 | 3.1 | 62.9 |
| PPPF | 180 | 56.4 | 22.3 |

V. CONCLUSIONS AND EXTENSIONS

The PPPF outperforms other particle filters on complex nonlinear problems, achieving higher effective sample sizes and lower estimation errors. This improvement is achieved by using a multi-stage approximation to the optimal importance density, and comes at a higher cost in computational load.

Numerous extensions to the algorithm have been developed, including a generalisation of the algorithm to allow stochastic evolution of the state through pseudo-time, methods for selecting step sizes adaptively, the inclusion of Markov Chain Monte Carlo steps to improve the quality of the particle population, and modifications for Gaussian mixture models.

REFERENCES

- [1] O. Cappé, S. Godsill, and E. Moulines, “An overview of existing methods and recent advances in sequential Monte Carlo,” *Proceedings of the IEEE*, vol. 95, pp. 899–924, 2007.
- [2] A. Doucet and A. M. Johansen, “A tutorial on particle filtering and smoothing: Fifteen years later,” in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovsky, Eds. Oxford University Press, 2009.
- [3] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, vol. 10, pp. 197–208, 2000.
- [4] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan, “The unscented particle filter,” *Advances in Neural Information Processing Systems*, vol. 13, pp. 584–590, 2000.
- [5] U. Hanebeck, K. Briechle, and A. Rauh, “Progressive Bayes: A new framework for nonlinear state estimation,” in *Proc. SPIE*, vol. 5099, 2003, pp. 256–267.
- [6] J. Hagmar, M. Jirstrand, L. Svensson, and M. Morelande, “Optimal parameterization of posterior densities using homotopy,” in *Proc. 14th Int. Conf. Information Fusion*, 2011, pp. 1–8.
- [7] S. Godsill and T. Clapp, “Improvement strategies for monte carlo particle filters,” in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. Springer, 2001, ch. 7, pp. 139–158.
- [8] J. Gall, J. Potthoff, C. Schnörr, B. Rosenhahn, and H.-P. Seidel, “Interacting and annealing particle filters: Mathematics and a recipe for applications,” *Journal of Mathematical Imaging and Vision*, vol. 28, no. 1, pp. 1–18, 2007.
- [9] J. Deutscher, A. Blake, and I. Reid, “Articulated body motion capture by annealed particle filtering,” in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, 2000, pp. 126–133.
- [10] N. Oudjane and C. Musso, “Progressive correction for regularized particle filters,” in *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, vol. 2, 2000.
- [11] F. Daum and J. Huang, “Particle degeneracy: root cause and solution,” in *Proc. SPIE*, vol. 8050. SPIE, 2011.
- [12] M. K. Pitt and N. Shephard, “Filtering via simulation: Auxiliary particle filters,” *Journal of the American Statistical Association*, vol. 94, pp. 590–599, 1999.
- [13] N. Gordon, D. Salmond, and A. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *Radar and Signal Processing, IEE Proceedings F*, vol. 140, pp. 107–113, 1993.