1

Test

A number of possible extensions to the current work are envisaged. Firstly, there are a number of variants of the particle filtering algorithms which would be interesting to investigate. Secondly, there are numerous additional problems in tracking which could be addressed. Finally, the fixed lag particle filter approach could be applied to different applications entirely.

1.1 Ideas for improved particle filters

1.1.1 *Mixed exact-flow filter*

In a traditional particle filter, particles are propagated from one time step to the next through the use of a proposal distribution and a weighting step, with resampling to maintain diversity. In (**?**), the authors introduce a new sort of particle filter, in which particles are propagated as if they were diffusing according a differential equation governed by the observation likelihood. Thus, it is possible to generate an exact, unweighted sample from the desired posterior distribution even for a nonlinear, non-Gaussian model. This method cannot be directly applied to a target tracking problem because such particle flow ideas can only be used for continuous state-spaces. The association variables will need to be handled differently.

It seems intuitive that we may be able to use an exact-flow filter for the kinematic target states, X_t, and an ordinary particle filter for the associations, Λ_t, in a similar manner to the Rao-Blackwellised method where a Kalman filter or filters is used for the kinematic states.

Consider a single target with state x_t and associations λ_t. From the previous processing step we have a particle approximation for the posterior, $\hat{P}(x_{1:t-1}, \lambda_{1:t-1}|Y_{1:t-1})$. The exact-flow method allows us to generate a sample from $P(x_t|x_{t-1}, \lambda_t, Y_t)$. Thus, a new particle set can be generated from the following proposal

$$x_{1:t}^{(m)}, \lambda_{1:t}^{(m)} \sim \hat{P}(x_{1:t-1}, \lambda_{1:t-1}|Y_{1:t-1})q(\lambda_t|X_{t-1}, Y_t)P(x_t|x_{t-1}, \lambda_t, Y_t) \tag{1.1}$$

The same association proposal may be used as for the standard particle filters described in chapter **??**. Note that this proposal implicity includes a resampling step, which could be omited, as always. The importance weights for such a filter will be given by

$$W_t^{(m)} \approx \frac{P(x_{1:t}^{(m)}, \lambda_{1:t}^{(m)}|Y_{1:t})}{P(x_{1:t-1}, \lambda_{1:t-1}|Y_{1:t-1})q(\lambda_t|X_{t-1}, Y_t)P(x_t|Y_{1:t}, \lambda_t)} \tag{1.2}$$

$$= \frac{P(\lambda_t^{(m)})P(Y_t|x_{t-1}^{(m)}, \lambda_t^{(m)})}{P(Y_t|Y_{1:t-1})q(\lambda_t|X_{t-1}, Y_t)} \tag{1.3}$$

Unfortuneately, the $P(Y_t|x_{t-1}^{(m)}, \lambda_t^{(m)})$ term will not be tractable in general. However, a Monte Carlo approximation could be used.

$$P(Y_t|x_{t-1}, \lambda_t) = \int P(Y_t|x_t, \lambda_t)P(x_t|x_{t-1})dx_t \approx \sum_i P(Y_t|x_t^{(i)}, \lambda_t) \tag{1.4}$$

where the summation is over a set of particles sampled from $P(x_t|x_{t-1})$. This of course leads to an algorithm with $O(N^2)$ complexity in the number of particles as a Monte Carlo approximation is needed for each particle in the main filter. However, provided both the transition density and the likelihood have a simple, unimodal form, the inner approximation may require only a small number of particles, rendering the scheme tractable.

1.1.2 SISR proposals for MCMC

The main failure mode of the MCMC-PF tracker occurs when there are two possible convincing routes for a target to take - generally one correct but with some missed detections and one marked by an unfortunate chain of clutter observations. The sampler gets onto the wrong path (is often initialised in it) and fails to jump to the other.

The SISR-PF tracker does not have this problem, because the particles are generated in parallel. However, it only achieves acceptable performance if independence assumptions are introduced to reduce the dimensionality - effectively to run an independent PF on each target.

It may be possible to combine the two schemes. First, taking the particle distribution from the previous processing step, a set of independent SISR-PFs can be run. The resulting marginal distributions may then be used as proposal distributions for moves in a joint-target MCMC-PF. Such a scheme may benefit from the advantages of the two different algorithms, maintaining a complete joint representation of the target space while minimising track loss. Furthermore, if the right set of values from the independent filters are stored in memory there is very little additional calculation required for the joint-target MCMC-PF stage.

Such an algorithm is likely to fail when many targets are close together. In such a case, many of the proposals will result in association contradictions, and the MH acceptance rate will fall. Such invalid proposals can be rejected very quickly as we only need to check target associations rather than calculate transition or likelihood probabilities. Thus, we can mitigate this problem by simply lengthening the chain.

1.1.3 MCMC importance sampling

The weakness of the MCMC-PF can be viewed in an another way. When the probability of the correct path is low, the SISR-PF is able to maintain it in the particle distribution with a number of low-weight particles until its probability rises again. In an MCMC scheme, all particles have equal weights, and such low-probability modes may get very few or no particles. An alternative way to combine the advantages of the SISR and MCMC methods would be to construct a Markov chain which generated a weighted particle distribution.

In an SISR scheme, a set of weighted particles $\{x_{1:t}^{(m)}, W_t^{(m)}\}$ are used to approximate the posterior distibution, $P(x_{1:t}|Y_{1:t})$. The unweighted particles may be considered to be samples from some underlying distribution, $\mu(x_{1:t}|Y_{1:t})$ (determined by the proposal distribution and previous resampling steps). Potentially, a Markov chain could be targetted on $\mu(x_{1:t}|Y_{1:t})$ instead of $P(x_{1:t}|Y_{1:t})$.

Suppose at the m^{th} step in the chain we have the state $x_{1:t}^{(m)}$, with unnormalised weight $W_t^{(m)}$. A new state, $x_{1:t}^{*}$ is sampled from a MH proposal distribution $q(.|x_{1:t}^{(m)}, Y_{1:t})$. The standard acceptance probability formula will be

$$\alpha = \min\left\{1, \frac{\mu(x_{1:t}^{*}|Y_{1:t})q(x_{1:t}^{(m)}|x_{1:t}^{*}, Y_{1:t})}{\mu(x_{1:t}^{(m)}|Y_{1:t})q(x_{1:t}^{*}|x_{1:t}^{(m)}, Y_{1:t})}\right\} \tag{1.5}$$

For any given particle, we know that $P(x_{1:t}^{(m)}|Y_{1:t}) = W_t^{(m)}\mu(x_{1:t}^{(m)}|Y_{1:t})$ by definition, so

$$\alpha = \min\left\{1, \frac{W_t^{(m)}}{W_t^{*}} \times \frac{P(x_{1:t}^{*}|Y_{1:t})q(x_{1:t}^{(m)}|x_{1:t}^{*}, Y_{1:t})}{P(x_{1:t}^{(m)}|Y_{1:t})q(x_{1:t}^{*}|x_{1:t}^{(m)}, Y_{1:t})}\right\} \tag{1.6}$$

But with this formulation W_t^{*} is unknown. In fact, it could be set to anything: $\mu(x_{1:t}|Y_{1:t})$

is defined by $P(x_{1:t}|Y_{1:t})$ and the choice of weights rather than vice versa. An obvious choice of weight is to select

$$W_t^* = W_t^{(m)} \times \frac{P(x_{1:t}^*|Y_{1:t})q(x_{1:t}^{(m)}|x_{1:t}^*, Y_{1:t})}{P(x_{1:t}^{(m)}|Y_{1:t})q(x_{1:t}^*|x_{1:t}^{(m)}, Y_{1:t})} \tag{1.7}$$

which results in an acceptance probability of 1 for all moves.

One possible choice of proposal would be to select a state history, $x_{1:t-1}$ from the previous particle distribution, $\hat{P}(x_{1:t-1}|Y_{1:t-1})$, and a current state dependent on this. This would result in a recursive cancellation in the particle weights, leading to (assuming $W_t^{(0)} = 0$):

$$W_t^* = \frac{P(Y_t|x_t^*)P(x_t^*|x_{t-1}^*)}{q(x_t^*|x_{1:t-1}^*, Y_t)} \tag{1.8}$$

which is just the normal weight update formula for a simple SISR particle filter! However, we need not necessarily propose an entire new state for every particle, but can change one component (target) at a time. The result will be an SISR-like algorithm in which particles and weights are not generated independently but dependent on the previous particle and weight. In addition, we can choose different forms for the weight update equation. At the other end of the spectrum, we could set $W_t^* = 1$ which gives us a standard MCMC-PF. Other choices could give some sort of intermediate filter which would allow some variation in weight but not as much as the SISR version.

The hope is that such a scheme should allow us to maintain a complete joint-target posterior distribution and yet not lose track of targets whose path is temporarily of low probability. However, it may simply suffer from all the problems of a joint-target SISR-PF. Specifically, it may discard particles with good estimates of one target because their estimates of another target are poor. In any case, it seems like an interesting new way to view the equivalence of SISR and MCMC.

1.2 Additional tracking problems

1.2.1 Additions to the algorithms

There are various possible extensions to the fixed-lag particle filter trackers. Firstly, the target detection mechanism currently only works with the SISR-PF and requires an independence assumption. It would be desirable to implement target detection for an MCMC-PF. It would also be interesting to try the algorithms on some more nonlinear models with non-Gaussian noise. In such cases the full particle filters would be expected to perform better than their

Rao-Blackwellised equivalents. Models for manoeuvring targets would also be of interest, such as those of (?) or the intrinsic coordinate models of (?).

Furthermore, it would be interesting to compare the algorithms with an MC-JPDAF and a proper MHT, to see how they compare in terms of tracking performance and computation time.

1.2.2 Target correspondence

In all the studies conducted so far there has been only one sensor observing the scene. Simple situations with multiple sensors can be handled by introducing additional association variables, as in (?). However, another problem of interest is that of matching the targets viewed by one sensor with those of another when the time periods of observation may not be overlapping, the characteristics of the sensors may be very different, including different biases, and the fields of view may not overlap. In particular, if one object in a group is of particular interest and has been identified by one sensor, we want to be able to identify it with the other sensor, based on the its location in the group. Such a problem will require the detection and tracking of objects as well as simultaneous estimation of correspondence.

1.2.3 Other applications

So far the work conducted has been entirely theoretical, using simulated data with no specific application in mind. It would be interesting to test the fixed-lag particle filter trackers on some real data. Suggested potential applications include tracking groups of people or animals in video sequences, or cells in microscopy data. Extensions for additional latent variables would also be interesting, such as the group structure variables used in (?).

1.3 Fixed lag particle filtering

There are numerous other situations in which fixed-lag particle filters may be of use. For example, the music transcription algorithms of (?) could well be improved by revising particle estimates once further data has been received. Note onsets and offsets would certainly be more easily detected.

Another area in which the fixed-lag particle filters could be applied is in change-point detection, for example in the jump-diffusion models of (?) and (?). Here a particle filter is used to estimate a latent series of jump times in a signal. Intuitively, it should be easier to see whether a jump has occured after further data has been received, so a fixed-lag particle filter may improve the estimate of jump times and thus the predictive power of the algorithm.

Task	Time estimate (months)
Application to more nonlinear models	1/2
Mixed exact-flow PF for tracking	3
Independent SISR proposals for an MCMC-PF tracker	1
MCMC importance sampling	3
Cursor intentionality tracking	1/2
Fixed-lag PFs for jump-diffusion models	2
Target correspondence and other high-level modelling	6
Application of trackers to real data	3
Fixed-lag PFs for music transcription	3

Table 1.1:

1.4 A plan for future research

As anyone working in signal processing will be aware, prediction is a difficult task, prone to exponential growth in uncertainty. Practically speaking, any attempt to plan out a PhD with a horizon longer than a few months is unlikely to bare much resemblence to the course actually followed. So in table **??** we simply list a number of the aforementioned interesting possibilities in a rough order of priority with an estimate of the time required.

The time estimates in table **??** sum to 22 months, leaving 3 months before the 3 year deadline (or 6 months before funding ends) for writing up. Realistically, it is far more likely that one of the tasks provides numerous interesting opportunities and takes over the entire course of research.