Target tracking example.

Step j (j = 1, … , k):

i) Sample $X_j \left| X_{j-1}, Y_j \sim N\left(\theta_j, \Gamma_j\right)\right.$ where

$$\begin{aligned}\theta_j &= GX_{j-1} \\ &\qquad + H\Lambda_j H^T F^T \left(\Sigma_j + FH\Lambda_j H^T F^T\right)^{-1} \left(Y_j - FGX_{j-1}\right) \\ \Gamma_j &= H\Lambda_j H^T \\ &\qquad - H\Lambda_j H^T F^T \left(\Sigma_j + FH\Lambda_j H^T F^T\right)^{-1} FH\Lambda_j H^T\end{aligned}$$

This is gotten by plugging the appropriate matrices into

$$\begin{aligned}\theta_j &= E\left[X_j \left| X_{j-1}\right.\right] \\ &\qquad + \operatorname{Cov}\left(X_j, Y_j \left| X_{j-1}\right.\right) \operatorname{Var}\left(Y_j \left| X_{j-1}\right.\right)^{-1} \left(Y_j - E\left[Y_j \left| X_{j-1}\right.\right]\right) \\ \Gamma_j &= \operatorname{Var}\left(X_j \left| X_{j-1}\right.\right) \\ &\qquad - \operatorname{Cov}\left(X_j, Y_j \left| X_{j-1}\right.\right) \operatorname{Var}\left(Y_j \left| X_{j-1}\right.\right)^{-1} \operatorname{Cov}\left(Y_j, X_j \left| X_{j-1}\right.\right)\end{aligned}$$

$ii)$ Update the weight

$$w_j\left(X_{1:j}\right) = w_{j-1}\left(X_{1:j-1}\right) p\left(Y_j \middle| X_{j-1}\right)$$

since

$$p\left(Y_j \middle| Y_{1:j-1}, X_{1:j-1}\right) = p\left(Y_j \middle| X_{j-1}\right)$$

$p\left(Y_j \middle| X_{j-1}\right)$ is a normal density with

$$\mu_j = FGX_{j-1} = E\left[Y_j \middle| X_{j-1}\right]$$
$$\mathrm{Var}_j = \Sigma_j + FH\Lambda_j H^T F^T = \mathrm{Var}\left(Y_j \middle| X_{j-1}\right)$$

What to do with more complicated models, such as

$$\log s_t = \log s_{t-1} + \delta_{s,t}$$
$$\theta_t = \theta_{t-1} + \delta_{\theta,t}$$
$$x_t = x_{t-1} + \frac{s_t \cos\theta_t + s_{t-1}\cos\theta_{t-1}}{2}$$
$$y_t = y_{t-1} + \frac{s_t \sin\theta_t + s_{t-1}\sin\theta_{t-1}}{2}$$

The optimal proposal distribution still has the form

$$\begin{aligned} q_j^*\left(X_j \middle| Y_{1:j}, X_{1:j-1}\right) &= p\left(X_j \middle| Y_{1:j}, X_{1:j-1}\right) \\ &= p\left(X_j \middle| Y_j, X_{j-1}\right) \\ &\propto p\left(X_j \middle| X_{j-1}\right) p\left(Y_j \middle| X_j\right) \end{aligned}$$

However $p\left(X_j \middle| X_{j-1}\right)$ is no longer normal, though it is based on normal, assuming the random changes in speed and direction are normal.

One approach is to approximate it with a normal matching the mean and variance (approximately). Then the combination of the two pieces is approximately normal.

The normal approximation may be determined by

- Taylor series approximation (Delta rule)
- Numerical quadrature (Scaled unscented transformation)
- ???

For the nonlinear model above, it can also be dealt with by setting the state vector to $X_t^* = [\log S_t \quad \theta_t]$ and having the measurement model for Z_t depend on $X_1^*, \ldots, X_t^*$ in a nonlinear fashion.

The models are exactly the same, just parametrized differently. However the different parametrizations lead to a different normal approximations, and in fact for this example the nonlinear measurement model works better (lower CV for the importance sampling weights and smaller standard errors for the filtered target locations.

Data decompositions

$$X = \{X_1, \ldots, X_k\} \text{ and } Y = \{Y_1, \ldots, Y_k\}$$

Efficiency of SIS depends on how this decomposition is made.

In some problems there may be many ways of doing this decomposition.

For the target tracking example there isn't. The only decomposition that makes sense is to match it with time. (Physical constraints of data collection force this.)

Here are a couple where it does make a difference

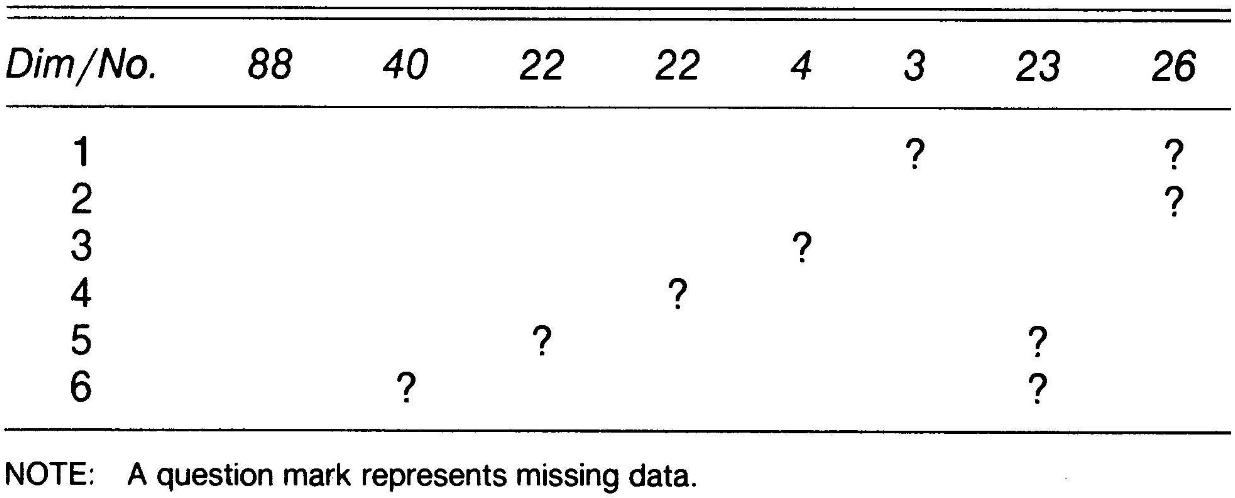
Example 1: Multivariate normal data with missing values (Kong et al, 1994)

Bayesian analysis using the Jeffreys' noninformative prior.

269 observations of a 6 component vector

88 observation complete

181 observations had at least 1 component missing with some missing up to 4



Dim/No.	*88*	*40*	*22*	*22*	*4*	*3*	*23*	*26*
1						?		?
2								?
3					?			
4				?				
5			?				?	
6		?					?	

NOTE: A question mark represents missing data.

They performed simulation in the order given above

1) complete data

2) 1 component missing

3) 2 components missing

etc

Another approach would be to deal with the data in the data collection order (which probably was random)

The importance sampling weights in this second approach will be more variable and thus more imputations will be needed to reach the same precision.

Note that in the analysis in this paper, it was based on simulated data. However it was based on the structure of a real data set from the social sciences.

One potential problem with SIS is that the variance of the importance sampling weights increases over time, which implies that ESS decreases as the sampler proceeds.

Example 2: Linkage Analysis

Similar to the example presented last time, but on a different data set (Irwin et al, 1994)

Want to estimate the disease location of a putative gene for a form of diabetes located on 20q with 8 markers.

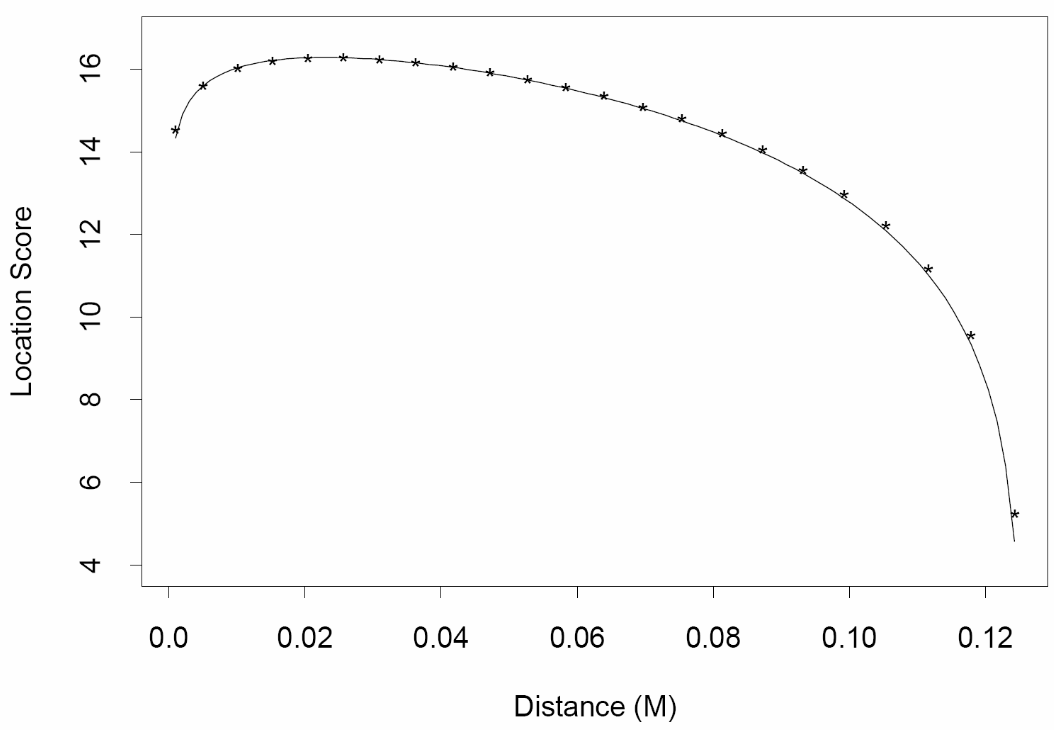
Two approaches:

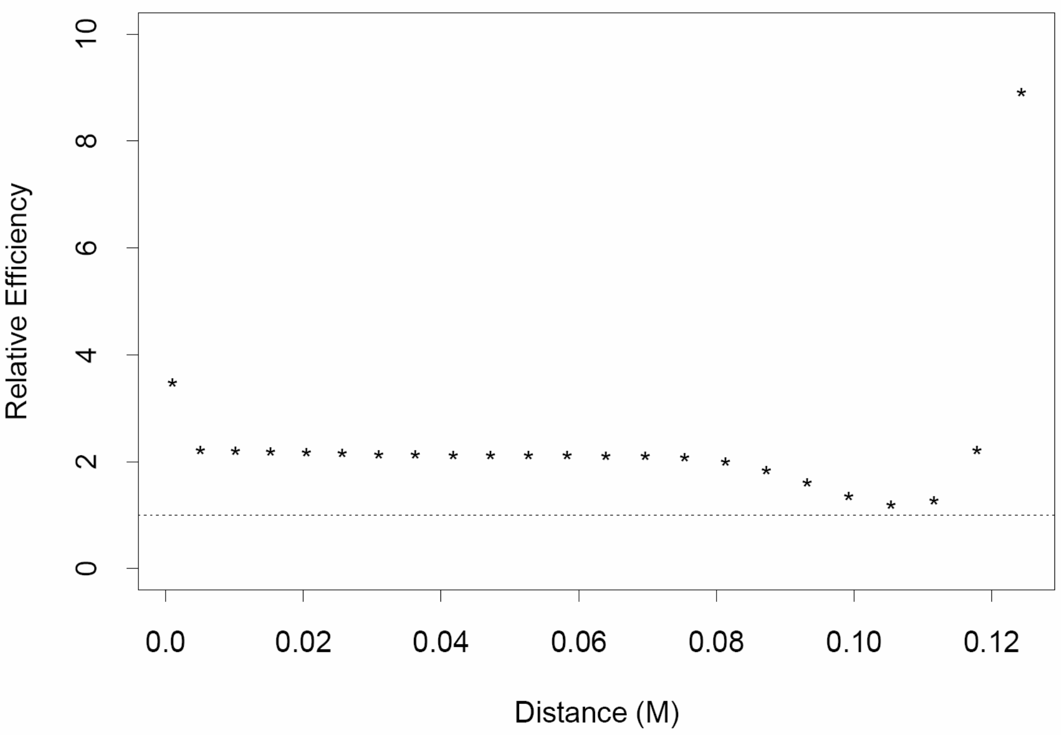
1) Process all marker data first then the disease data

2) Process marker RM292 first, then the disease data, then the other 7 markers

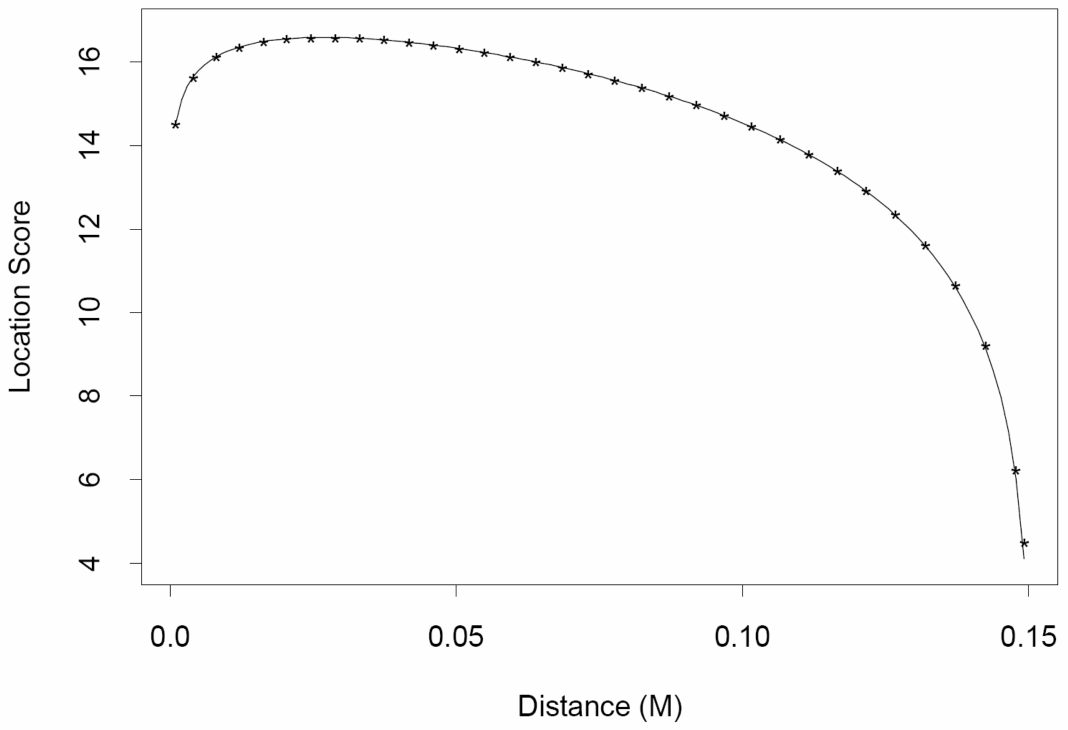
In both approaches the disease was processed in the middle of the marker interval of interest and the likelihood for other points in the interval were determined by reweighting the sample

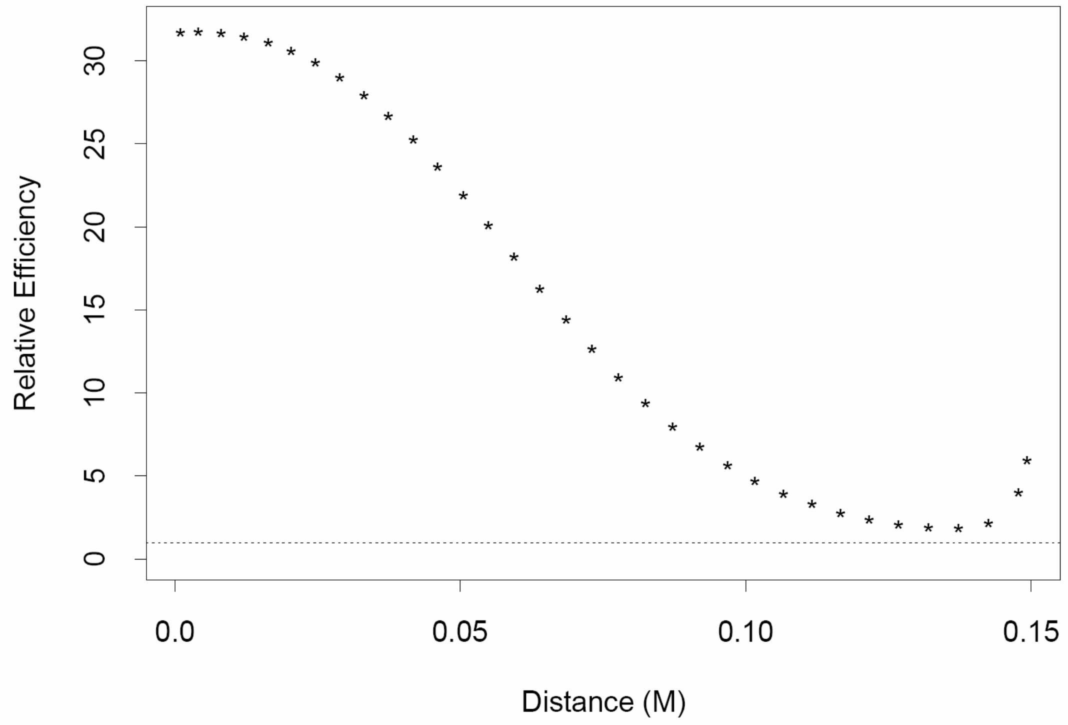
CEPH Distances:





MCEM Distances





In all cases processing the disease early gave more precise estimates, in some cases by a factor over 30.

When possible, you want include as much information in Y_1.

Want to sample with trial distribution based on

$$p\left(X\middle|Y\right) = p\left(X_1\middle|Y\right) p\left(X_2\middle|X_1,Y\right)\times\ldots \times p\left(X_k\middle|X_{1:k-1},Y\right)$$

instead of

$$q\left(X\middle|Y\right) = q\left(X_1\middle|Y_1\right) q\left(X_2\middle|X_1,Y_{1:2}\right)\times\ldots \times q\left(X_k\middle|X_{1:k-1},Y_{1:k}\right)$$

The first case will have importance sampling weights = 1 (assuming that you don't need to use importance sampling for any of the components $p\left(X_j\middle|X_{1:j-1},Y\right)$).

Thus careful thought can help alleviate the problem I talked about last time, the increasing variance of the importance sampling weights as the sampler progresses.

For example, suppose you have a process that you want to model with the following hierarchical structure

Process level 1: $[Y]$

Process level 2: $\left[X \middle| Y\right]$

Data: $\left[Z_x, Z_y \middle| X, Y\right] = \left[Z_x \middle| X\right]\left[Z_y \middle| Y\right]$

Want to sample *X* and *Y* from $\left[X, Y \middle| Z_x, Z_y\right]$.
One possible scheme is to use the following SIS scheme

1) Sample *X* from $\left[X \middle| Z_x\right]$ by SIS giving weights $w_x(X)$.

2) Sample *Y* from $\left[Y \middle| X, Z_x, Z_y\right]$. Given the probability structure above

   $$\left[Y \middle| X, Z_x, Z_y\right] = \left[Y \middle| X, Z_y\right]$$

   If it is possible to sample directly from $\left[Y \middle| X, Z_y\right]$,

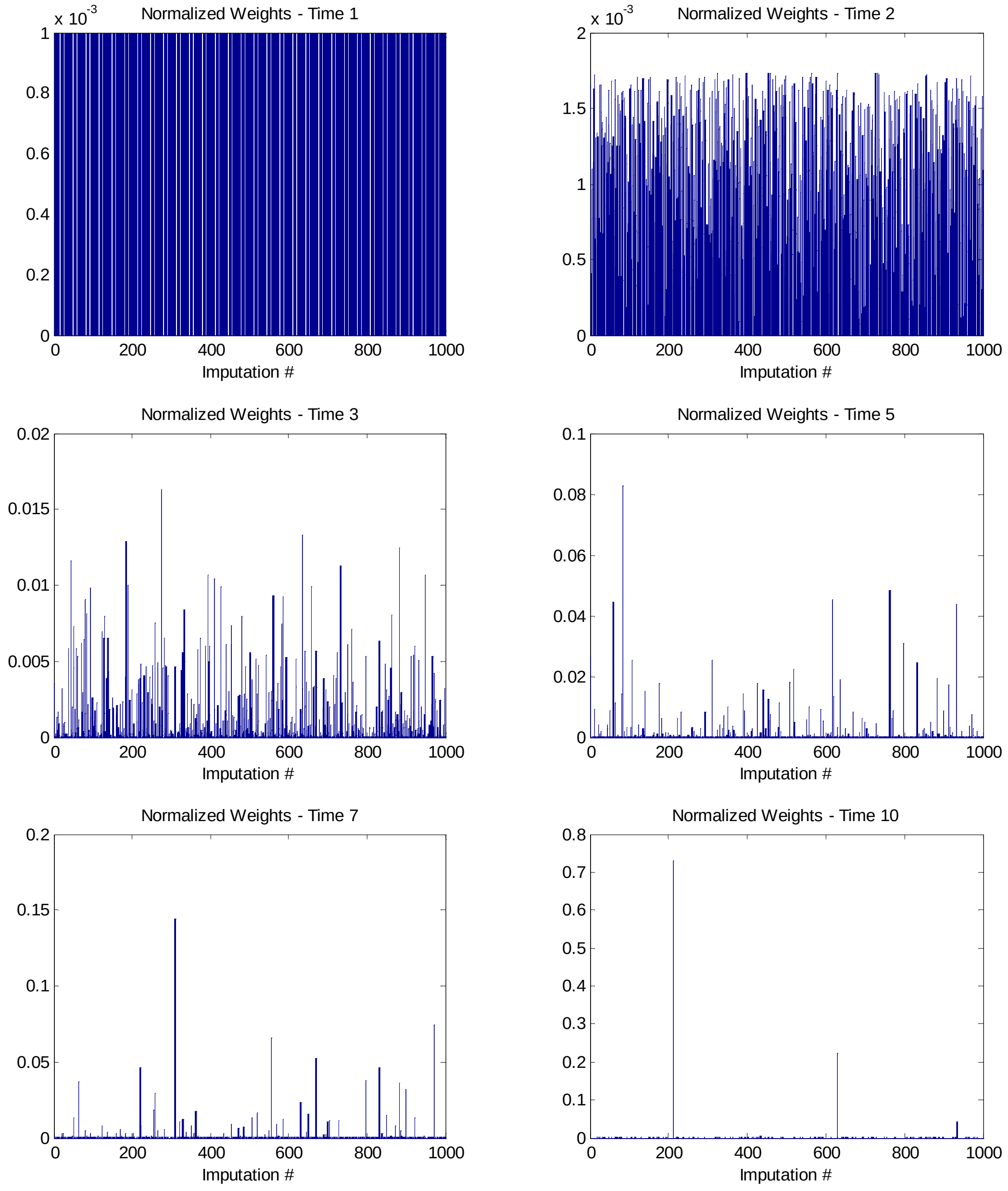
   $$w_{x,y}(X,Y) = w_x(X)$$

i.e. the simulation of Y in this case won't increase the variance of the importance sampling weights.

I have been able to do this with some genetics example, where X are the haplotypes, and Y is the inheritance vector.

However this idea won't work in the target tracking example.

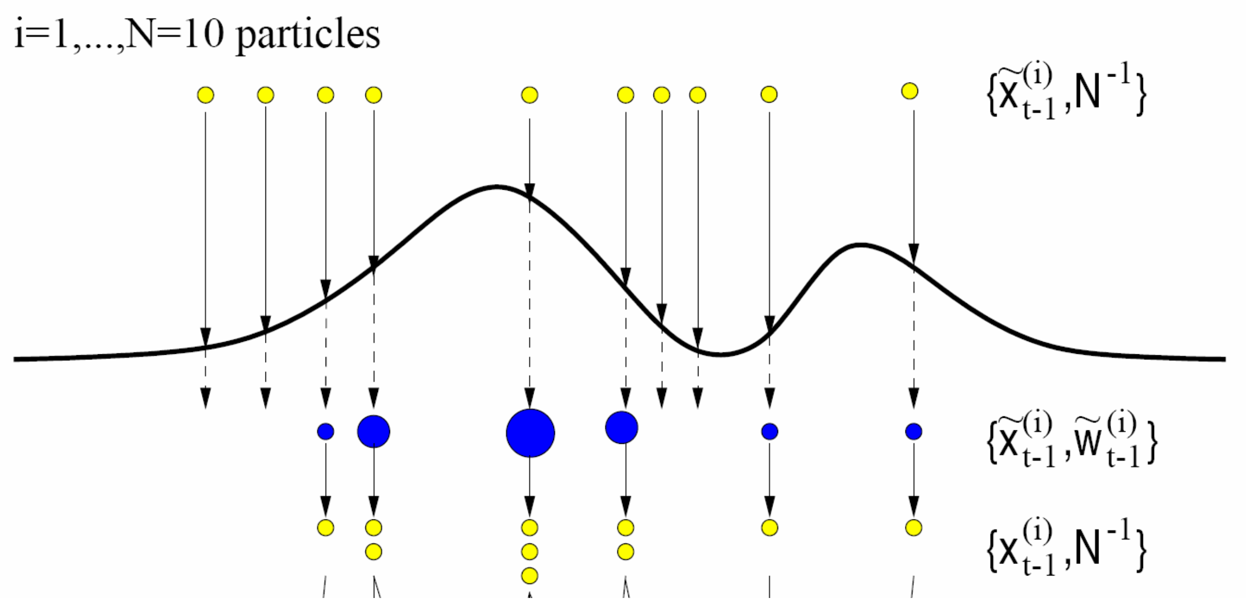
Lets look at how the normalized importance sampling weights can evolve over time in the target tracking example.


Normalized Weights - Time 1
x 10-3
1
0.8
0.6
0.4
0.2
0
0
200
400
600
800
1000
Imputation #
Normalized Weights - Time 2
x 10-3
2
1.5
1
0.5
0
0
200
400
600
800
1000
Imputation #
Normalized Weights - Time 3
0.02
0.015
0.01
0.005
0
0
200
400
600
800
1000
Imputation #
Normalized Weights - Time 5
0.1
0.08
0.06
0.04
0.02
0
0
200
400
600
800
1000
Imputation #
Normalized Weights - Time 7
0.2
0.15
0.1
0.05
0
0
200
400
600
800
1000
Imputation #
Normalized Weights - Time 10
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
200
400
600
800
1000
Imputation #

One way to think about importance sampling weights is in terms of how many samples would you expect to see if you sampled from the target distribution instead of the trial distribution you actually sampled from (if weights normalize to have mean 1).

So if $w(X)$ = 2, you would expect to see about twice as many copies of X if you sampled directly from the target distribution.

$w(X)$ = 0.5 implies you would expect half as many


i=1,...,N=10 particles
$\{\tilde{x}_{t-1}^{(i)}, N^{-1}\}$
$\{\tilde{x}_{t-1}^{(i)}, \tilde{w}_{t-1}^{(i)}\}$
$\{x_{t-1}^{(i)}, N^{-1}\}$

(From: van der Merwe et al, 2000, The Unscented Particle Filter)

Resampling:

Sample realizations from the set $\left\{X_{1:j}^{1},\ldots,X_{1:j}^{n}\right\}$ with probabilities proportional to the weights $w\left(X_{1:j}^{1}\right),\ldots,w\left(X_{1:j}^{n}\right)$.

Treat this new sample as an equally weighted from the target distribution.

Sequential Imputation with Resampling

For i = 1, … , n

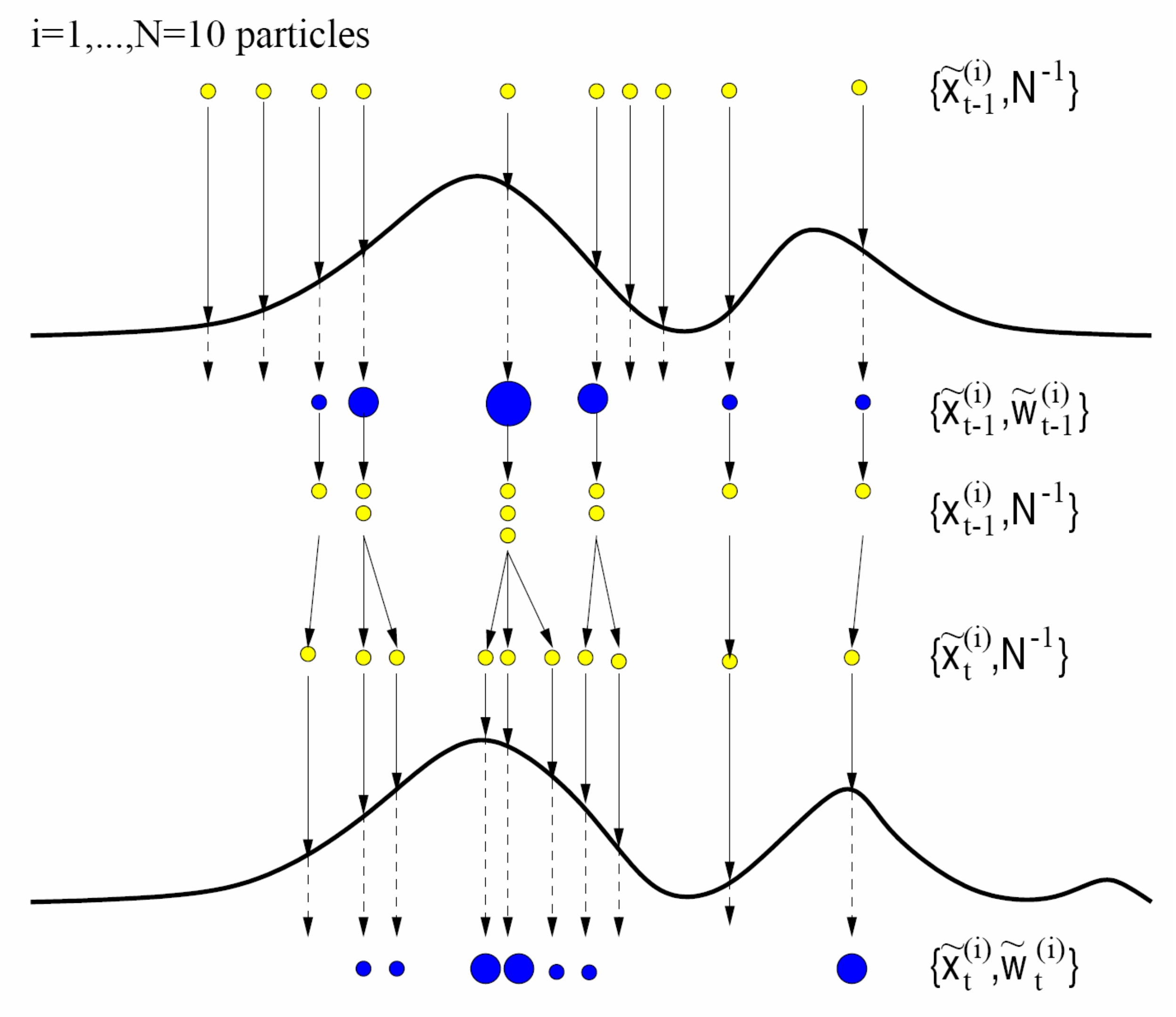
1) Sample $X_j^i \sim q_j\left(X_j \middle| Y_{1:j}, X_{1:j-1}^i\right)$

2) Update weight

$$w_j\left(X_{1:j}^i\right) = w_{j-1}\left(X_{1:j-1}^i\right) \times \frac{p\left(X_{1:j}^i \middle| Y_{1:j}\right)}{q_j\left(X_j^i \middle| Y_{1:j}, X_{1:j-1}^i\right) p\left(X_{1:j-1}^i \middle| Y_{1:j-1}\right)}$$

3) If appropriate, resample n realizations from$\left\{X_{1:j}^{1},\ldots,X_{1:j}^{n}\right\}$ with probabilities proportional to $w\left(X_{1:j}^{1}\right),\ldots,w\left(X_{1:j}^{n}\right)$.

Reset weights $w_j\left(X_{1:j}^i\right) = \frac{1}{n}$


i=1,...,N=10 particles
$\{\tilde{x}^{(i)}_{t-1}, N^{-1}\}$
$\{\tilde{x}^{(i)}_{t-1}, \tilde{w}^{(i)}_{t-1}\}$
$\{x^{(i)}_{t-1}, N^{-1}\}$
$\{\tilde{x}^{(i)}_{t}, N^{-1}\}$
$\{\tilde{x}^{(i)}_{t}, \tilde{w}^{(i)}_{t}\}$

Note that the resampling step does not have to be done through each pass. Two approaches are

1) resample every m times through (j = m, 2m, …)

2) monitor the weights and resample when the behaviour starts to get poor (e.g. when CV > C)

With resampling some realizations will get replicated and some will drop out.

There are a number of ways of doing the sampling.

Let

$$\tilde{w}\left(X^i\right)=\frac{w\left(X^i\right)}{\sum w\left(X^j\right)}$$

be the normalized weights

1) Multinomial sampling (Gordon, 1994)

   Sample

   $$l_1,\ldots,l_n \sim \text{Multi}\left(n,\left\{\tilde{w}\left(X^j\right)\right\}\right)$$

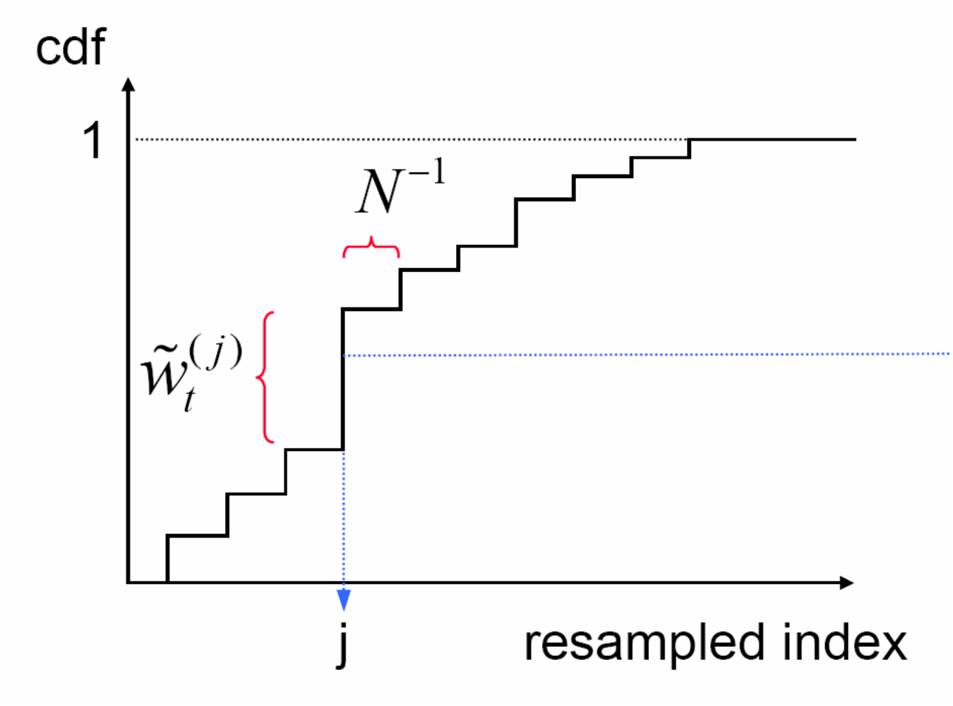
   where l_j is the number of copies of X^j in the new sample

This is equivalent to

Draw $U_i \sim \mathrm{Unif}(0,1)$, $i = 1, \ldots, n$

Set $\tilde{X}^i = X^j$ if

$$\sum_{l=1}^{j-1} \tilde{w}\left(X^l\right) \leq U_i < \sum_{l=1}^{j} \tilde{w}\left(X^l\right)$$



2) Residual sampling (Higuchi (1997), Liu and Chen (1998))

3) Minimum variance sampling (Kitagawa (1996), Crisan (2001))

Sample $U_1 \sim \text{Unif}\left(0, \tfrac{1}{\text{n}}\right)$

Let $U_j = U_1 + \dfrac{j-1}{n}$ for j = 2, … , n

$$\frac{j-1}{n} \le U_j < \frac{j}{n}$$

Set $\tilde{X}^i = X^j$ if

$$\sum_{l=1}^{j-1} \tilde{w}\left(X^l\right) \le U_i < \sum_{l=1}^{j} \tilde{w}\left(X^l\right)$$

This procedure has the property that X^j will occur either $\left\lfloor n\tilde{w}\left(X^j\right)\right\rfloor$ or $\left\lfloor n\tilde{w}\left(X^j\right)\right\rfloor + 1$ times in the new sample.

This implies that samples with high weights must be included in the new sample and that lowly weighted samples can’t get in very often.

This will minimize the variances on $\left\{l_j\right\}$