Sequential Importance Sampling (SIS)

AKA Particle Filtering, Sequential Imputation

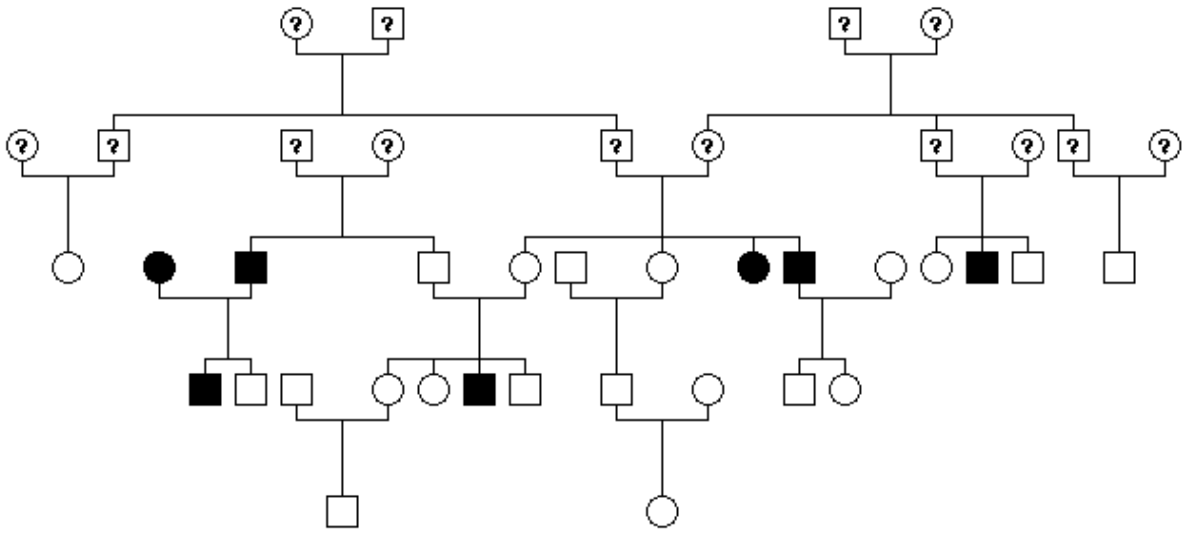
(Kong, Liu, Wong, 1994)

For many problems, sampling directly from the target distribution is difficult or impossible.

One reason possible reason for this is the size of the space that needs to be drawn from

Examples:

1) Linkage Analysis (Irwin, Cox, & Kong, 1994)



- m = 41 members
- n = 27 (nonfounders), f = 14 (founders)
- 8 markers from chromosome 19

- #alleles ranges from 6 to 8
- 14 members in top 2 generation have no marker data

Want to sample joint haplotypes for all pedigree members conditional on the observed marker and disease data

Assume that marker j has n_j possible alleles and the disease locus has two alleles.

Then the number of possible haplotypes for each person is

$$h = 4\prod n_j^2$$

and the maximum number of joint haplotypes possible is

$$H = h^m$$

If n_j = 8 for all markers, $h = 1.1259 \times 10^{15}$ and $H = 1.29268 \times 10^{617}$.

Note that not all possible joint haplotypes included in H have positive probability since they won't be consistent with Mendelian segregration.

In addition the observed data will also reduce the number of possible haplotypes with positive probability.

2) Target tracking (Irwin, Cressie, & Johannesson, 2002)

Movement Model:

Position:

$$x_t = x_{t-1} + \bar{v}_{x,t}$$
$$y_t = y_{t-1} + \bar{v}_{y,t}$$

Velocity:

$$v_{x,t} = v_{x,t-1} + \delta_{x,t}$$
$$v_{y,t} = v_{y,t-1} + \delta_{y,t}$$

$$\bar{v}_{x,t} = \frac{v_{x,t} + v_{x,t-1}}{2} = v_{x,t-1} + \frac{1}{2}\delta_{x,t}$$
$$\bar{v}_{y,t} = \frac{v_{y,t} + v_{y,t-1}}{2} = v_{y,t-1} + \frac{1}{2}\delta_{y,t}$$

where $\delta_{x,t}$ and $\delta_{y,t}$ are the average accelerations in the x and y directions from time $t-1$ to time t.

This gives

$$x_t = x_{t-1} + v_{x,t-1} + \tfrac{1}{2}\delta_{x,t}$$
$$y_t = y_{t-1} + v_{y,t-1} + \tfrac{1}{2}\delta_{x,t}$$

This can be written in matrix format

$$X_t = GX_{t-1} + H\delta_t$$

where

$$X_t^T = \begin{bmatrix} x_t & y_t & v_{x,t} & v_{y,t} \end{bmatrix}$$

$$\delta_t^T = \begin{bmatrix} \delta_{x,t} & \delta_{x,t} \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \; H = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Assume that the model for the average accelerations is

$$\delta_t \sim N_2\left(0, \Lambda_t\right)$$

Measurement model:

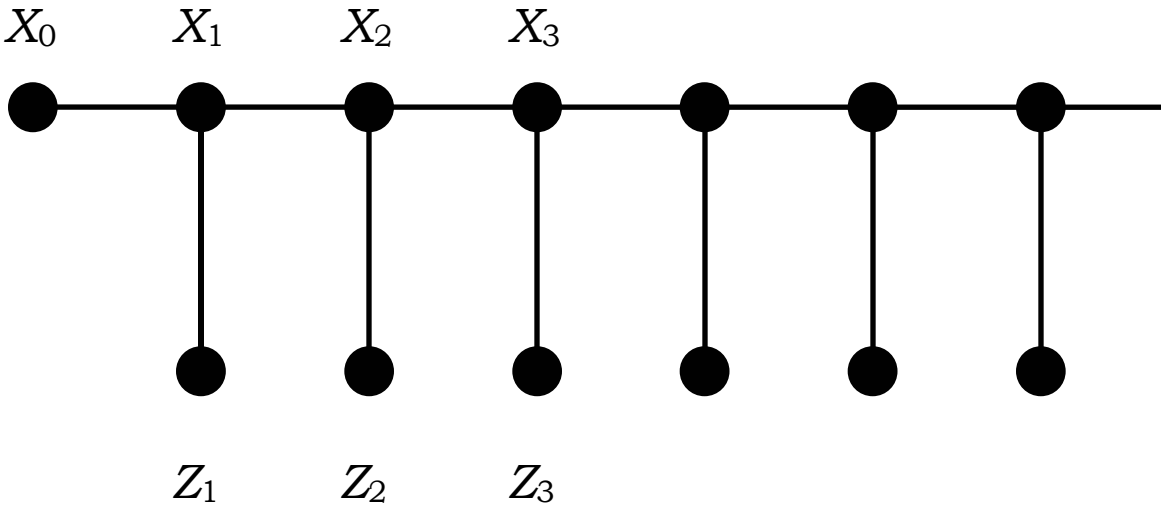
Two radars track the targets position with error

$$Z_t = FX_t + \varepsilon_t;\ \varepsilon_t \sim N\left(0, \Sigma_t\right)$$

where

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The probability structure can be described by the following graph



The model is an example of the hidden Markov model. The state variables X_t are described by a continuous state Markov Chain but are unobserved (hidden). All that is observed are the Z_t, the observed target positions

Problem:

Want to know the distribution of $X_t \left| Z_{1:t} \right.$ for each t ($Z_{1:t} = \{Z_1, \ldots, Z_t\}$).

Since this is a linear dynamic model, it can easily be solved by the Kalman filter (KF) (Kalman, 1960).

In this case, $X_t \left| Z_{1:t} \right.$ is Gaussian and the means and variances can be determined by the following simple update formulas.

$$\mu_t = E\left[X_t \left| Z_{1:t} \right. \right];\ \mu_{t|t-1} = E\left[X_t \left| Z_{1:t-1} \right. \right]$$

$$P_t = \mathrm{Var}\left(X_t \left| Z_{1:t} \right. \right);\ P_{t|t-1} = \mathrm{Var}\left(X_t \left| Z_{1:t-1} \right. \right)$$

Assuming $E[\delta_t] = E[\varepsilon_t] = 0$, the KF calculations are

$$\begin{aligned}
\mu_{t|t-1} &= G\mu_{t-1} \\
P_{t|t-1} &= GP_tG^T + H\Lambda_t H^T \\
K_t &= P_{t|t-1}F^T\left[\Sigma_t + FP_{t|t-1}F^T\right]^{-1} \\
\mu_t &= \mu_{t|t-1} + K_t\left(Z_t - F\mu_{t|t-1}\right) \\
P_t &= P_{t|t-1} - K_tF_tP_{t|t-1}
\end{aligned}$$

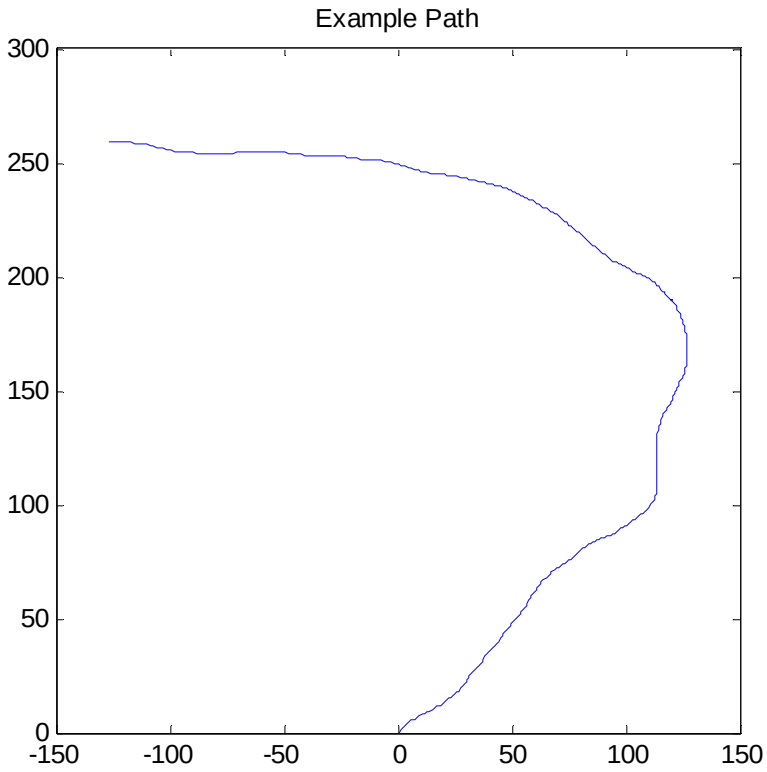
where K_t is known as the Kalman gain.

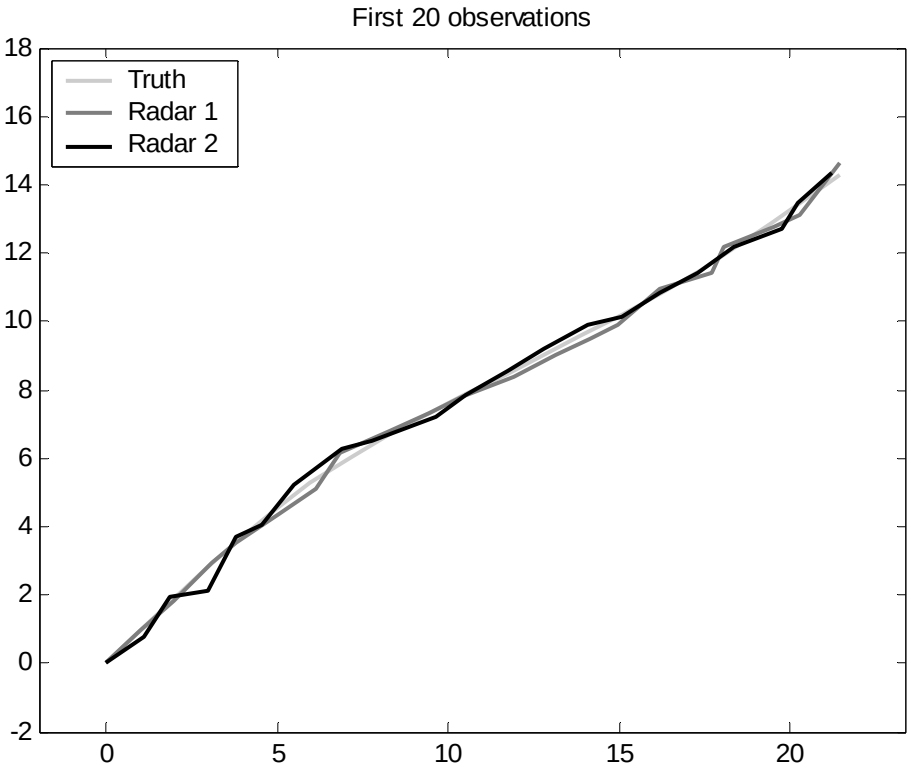
Note that the Kalman filter calculations here reduce to Normal conditional distribution calculations. For example

$$K_t = \text{Cov}(X_t, Z_t)\left(\text{Var}\left(Z_t\right)\right)^{-1},$$

exactly what you need to calculate for a multivariate regression of Z_t on X_t.

P_t reduces to a standard conditional variance calculation.


Example Path
300
250
200
150
100
50
0
-150
-100
-50
0
50
100
150


First 20 observations
Truth
Radar 1
Radar 2
18
16
14
12
10
8
6
4
2
0
-2
0
5
10
15
20

The above data was generated under the model described above with

$$X_0 = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}^T$$

$$\Lambda_t = \begin{bmatrix} 0.003 & 0 \\ 0 & 0.003 \end{bmatrix}; \; \Lambda_t^{0.5} = \begin{bmatrix} 0.0548 & 0 \\ 0 & 0.0548 \end{bmatrix}$$

and

$$\Sigma_t = \begin{bmatrix} 0.03 & 0 & 0 & 0 \\ 0 & 0.03 & 0 & 0 \\ 0 & 0 & 0.04 & 0.008 \\ 0 & 0 & 0.008 & 0.004 \end{bmatrix}$$

However if the movement or measurement models are non linear or contain non-normal random components, and the Kalman filter or its modifications, such as the Extended Kalman filter (EKF) can give poor answers.

The EKF linearizes the system through Taylor series approximations and then runs the standard Kalman filter on this linear system.

An example with a nonlinear component is given with the movement model

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

In this setting, simulating realizations of X_t will give a better approximation to $\mu_t = E\left[X_t \middle| Z_{1:t}\right]$ and $P_t = \mathrm{Var}\left(X_t \middle| Z_{1:t}\right)$, (or any other functional of X_t).

In this case the distribution of X_t can be extremely difficult to deal with directly, but is fairly easy to deal with conditional of the earlier parts of the path (drawing X_t given X_{t-1} and Z_t is tractable)

For both examples (linkage analysis and target tracking), sequential importance sampling is a useful technique for sampling from the desired posterior distributions.

Let $X = \{X_1, X_2, \ldots, X_k\}$ be some decomposition of the random variable you wish to sample from and $Y = \{Y_1, Y_2, \ldots, Y_k\}$ be the corresponding decomposition of the data you wish to condition on.

Want to sample from

$$p(X|Y) = \frac{p(X)\,p(Y|X)}{p(Y)}$$

which is assumed to be difficult to do.

Want to find a distribution $q(X|Y)$ that is easy to sample from and use importance sampling.

SIS

1) Sample $X_1 \sim q_1(X_1|Y_1)$ and calculate

$$w_1(X_1) = \frac{p(X_1|Y_1)}{q_1(X_1|Y_1)}$$

2) Then for $j = 2, \ldots, k$

Sample $X_j \sim q_j(X_j|Y_{1:j}, X_{1:j-1})$ and calculate

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

The factor

$$\frac{p\left(X_{1:j}\middle|Y_{1:j}\right)}{q_j\left(X_j\middle|Y_{1:j}, X_{1:j-1}\right) p\left(X_{1:j-1}\middle|Y_{1:j-1}\right)}$$

is often easy to calculate.

The resulting sample $X = X_{1:k}$ is a weighted sample from $p\left(X\middle|Y\right)$ with unnormalized importance sampling weight

$$w_k\left(X_{1:k}\right) = \frac{p\left(X_{1:k}\middle|Y_{1:k}\right)}{q\left(X_{1:k}\middle|Y_{1:k}\right)}$$

where

$$q\left(X_{1:k}\middle|Y_{1:k}\right) = q_1\left(X_1\middle|Y_1\right)\prod_{j=2}^{k} q_j\left(X_j\middle|Y_{1:j}, X_{1:j-1}\right)$$

The components of the proposal need to be chosen so that that they are easy to sample from.

Two popular choices are

$$q_j^*\left(X_j \middle| Y_{1:j}, X_{1:j-1}\right) = p\left(X_j \middle| Y_{1:j}, X_{1:j-1}\right)$$

and

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

The first choice is optimal in that is minimizes the variance of the importance sampling weights (which will increase the ESS).

The second choice is often easy, such as with the target tracking example. However by ignoring the data, it can significantly increase the importance sampling weight variance.

Optimal proposal properties

For the optimal proposal

$$w\left(X_{1:k}\right) = p\left(Y_1\right)\prod_{j=2}^{k} p\left(Y_j \middle| Y_{1:j-1}, X_{1:j-1}\right)$$

which implies (Kong et al, 1994, Irwin et al, 1994)

$$q^*\left(X_{1:k}\middle|Y_{1:k}\right)=\frac{p\left(Y_{1:k}\right)p\left(X_{1:k}\middle|Y_{1:k}\right)}{w\left(X_{1:k}\right)}$$

or

$$w\left(X_{1:k}\right)=\frac{p\left(Y_{1:k}\right)p\left(X_{1:k}\middle|Y_{1:k}\right)}{q^*\left(X_{1:k}\middle|Y_{1:k}\right)}$$

so

$$\begin{aligned}E_q\left[w\left(X_{1:k}\right)\right]&=\int w\left(X\right)q^*\left(X_{1:k}\middle|Y_{1:k}\right)dX_{1:k}\\&=\int w\left(X\right)\frac{p\left(Y_{1:k}\right)p\left(X_{1:k}\middle|Y_{1:k}\right)}{w\left(X_{1:k}\right)}dX_{1:k}\\&=p\left(Y_{1:k}\right)\int p\left(X_{1:k}\middle|Y_{1:k}\right)dX_{1:k}\\&=p\left(Y_{1:k}\right)\end{aligned}$$

Thus the likelihood of the data can be estimated with the average of the unnormalized importance sampling weights.

Implementing SIS for the target tracking example.

Since the movement is described by a Markov chain and the observations are assumed to be independent

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

So the optimal proposal is tractable here.

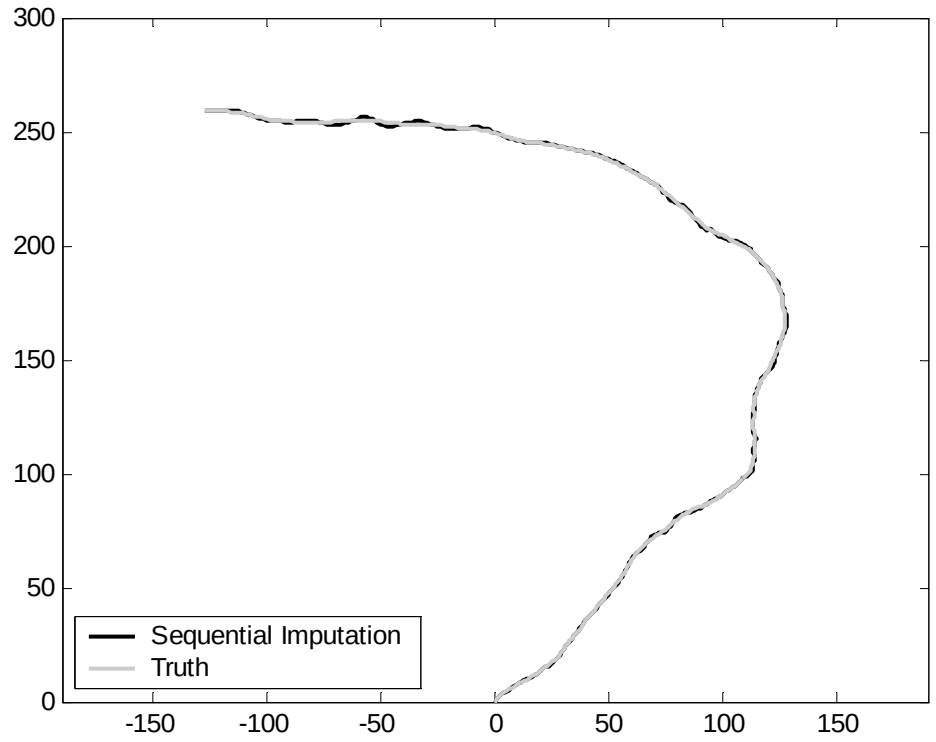
In fact, $X_j \left| X_{j-1}, Y_j \right. \sim N\left(\theta_j, \Gamma_j\right)$ where

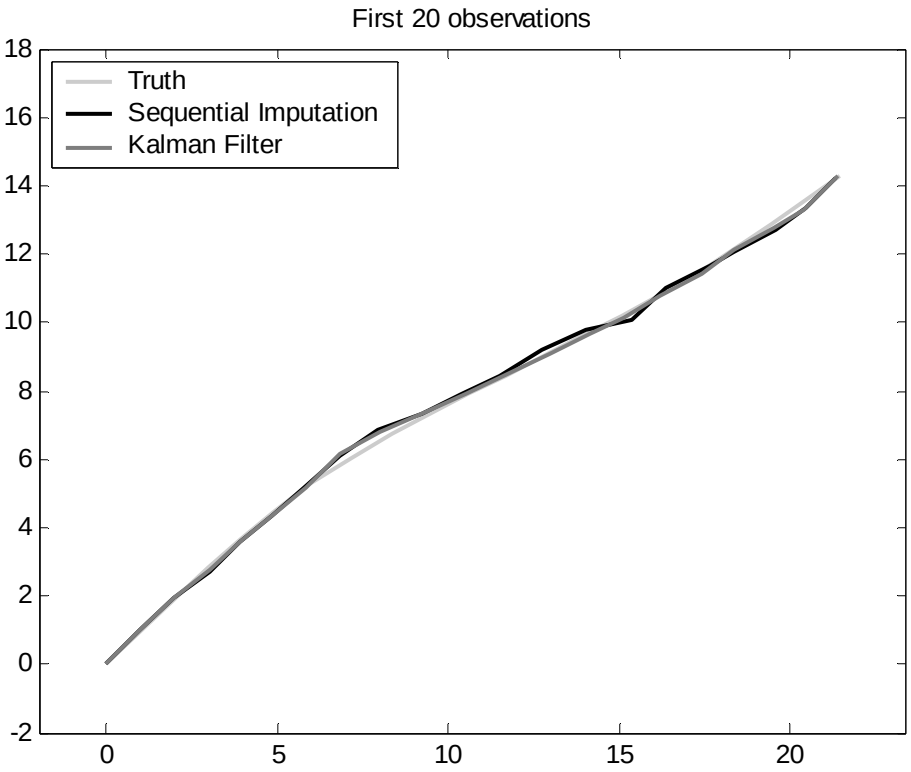
$$\begin{aligned} \theta_j &= GX_{j-1} \\ &\quad + H\Lambda_j H^T F^T \left(\Sigma_j + FH\Lambda_j H^T F^T\right)^{-1} \left(Y_j - FGX_{j-1}\right) \\ \Gamma_j &= H\Lambda_j H^T \\ &\quad - H\Lambda_j H^T F^T \left(\Sigma_j + FH\Lambda_j H^T F^T\right)^{-1} FH\Lambda_j H^T \end{aligned}$$

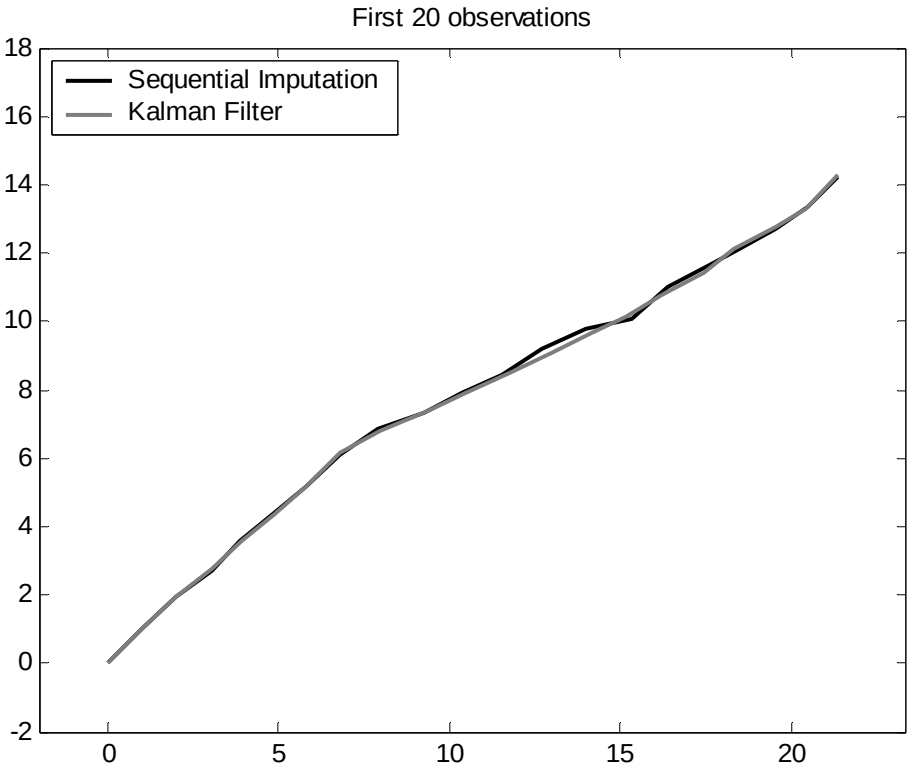
In addition, the multiplier for the weight is

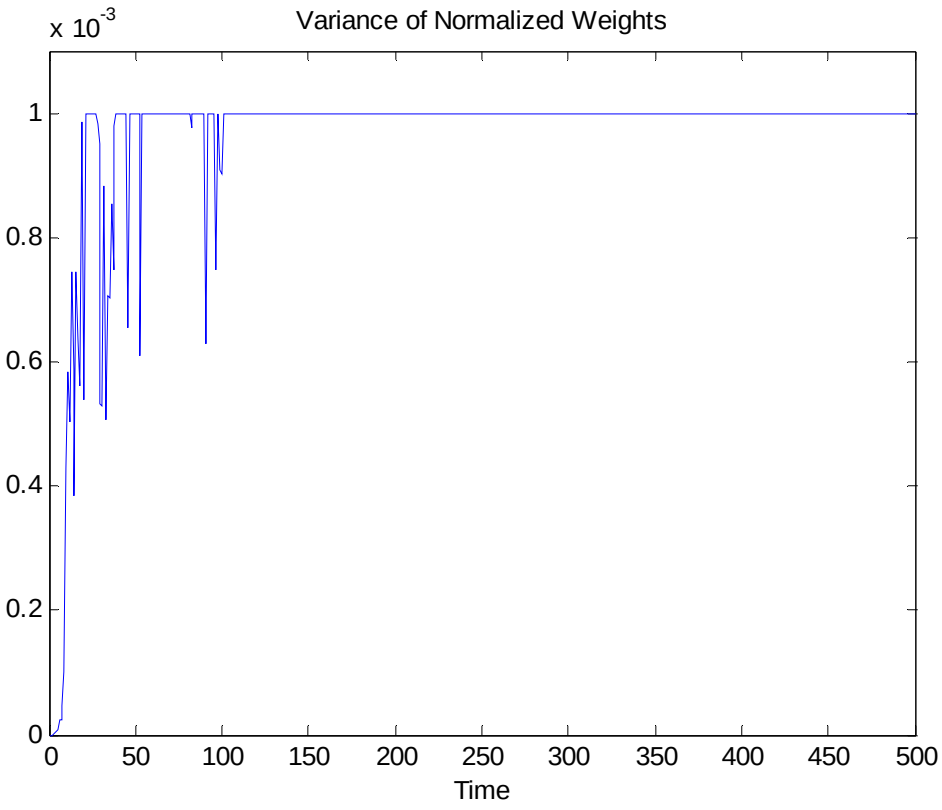
$$p\left(Y_j \middle| Y_{1:j-1}, X_{1:j-1}\right) = p\left(Y_j \middle| X_{j-1}\right)$$

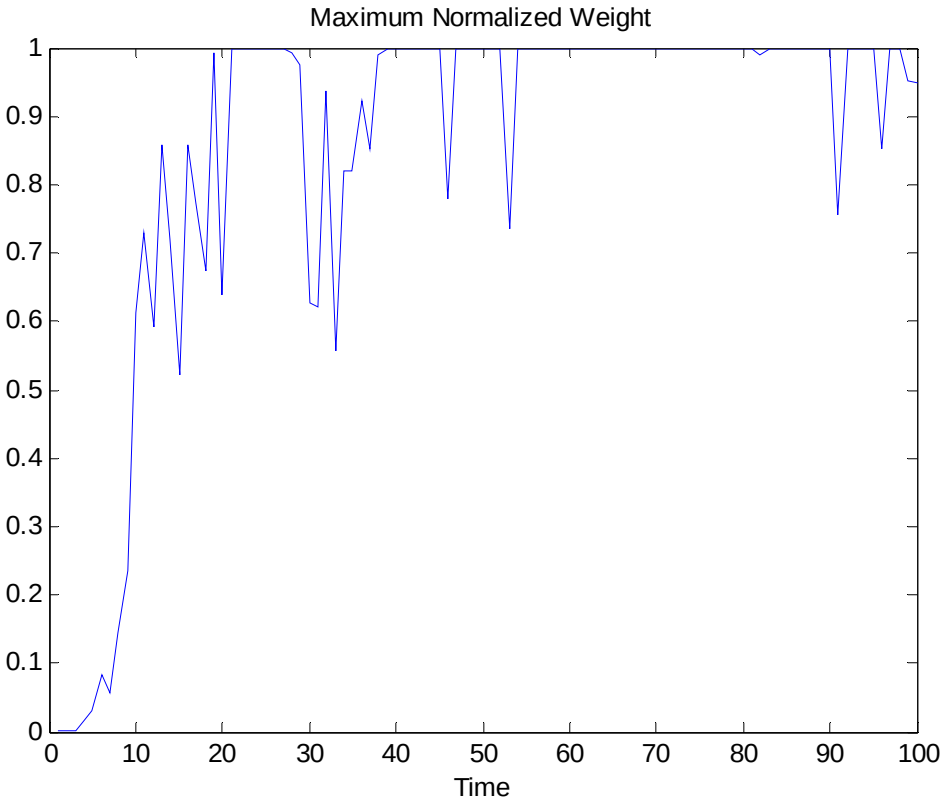
which is the density of a
$N\left(FGX_{j-1}, \Sigma_j + FH\Lambda_j H^T F^T\right)$ random variable.




First 20 observations
Truth
Sequential Imputation
Kalman Filter
18
16
14
12
10
8
6
4
2
0
-2
0
5
10
15
20


First 20 observations
Sequential Imputation
Kalman Filter
18
16
14
12
10
8
6
4
2
0
-2
0
5
10
15
20


Variance of Normalized Weights
x 10^-3
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
350
400
450
500
Time


Maximum Normalized Weight
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90
100
Time

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.

Thus the estimates of the mean are less precise, the further into the sampler we go.

Solution: Resampling.

Target tracking example.

Step j ($j = 1, \ldots, k$):

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

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

$$\Gamma_j = H\Lambda_j H^T - H\Lambda_j H^T F^T \left(\Sigma_j + FH\Lambda_j H^T F^T\right)^{-1} FH\Lambda_j H^T$$

This is gotten by plugging the appropriate matrices into

$$\theta_j = E\left[X_j \left| X_{j-1}\right.\right] + \mathrm{Cov}\left(X_j, Y_j \left| X_{j-1}\right.\right) \mathrm{Var}\left(Y_j \left| X_{j-1}\right.\right)^{-1} \left(Y_j - E\left[Y_j \left| X_{j-1}\right.\right]\right)$$

$$\Gamma_j = \mathrm{Var}\left(X_j \left| X_{j-1}\right.\right) - \mathrm{Cov}\left(X_j, Y_j \left| X_{j-1}\right.\right) \mathrm{Var}\left(Y_j \left| X_{j-1}\right.\right)^{-1} \mathrm{Cov}\left(Y_j, X_j \left| X_{j-1}\right.\right)$$

$ii)$ 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 $X_t^* = \begin{bmatrix} \log S_t & \theta_t \end{bmatrix}$ 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.

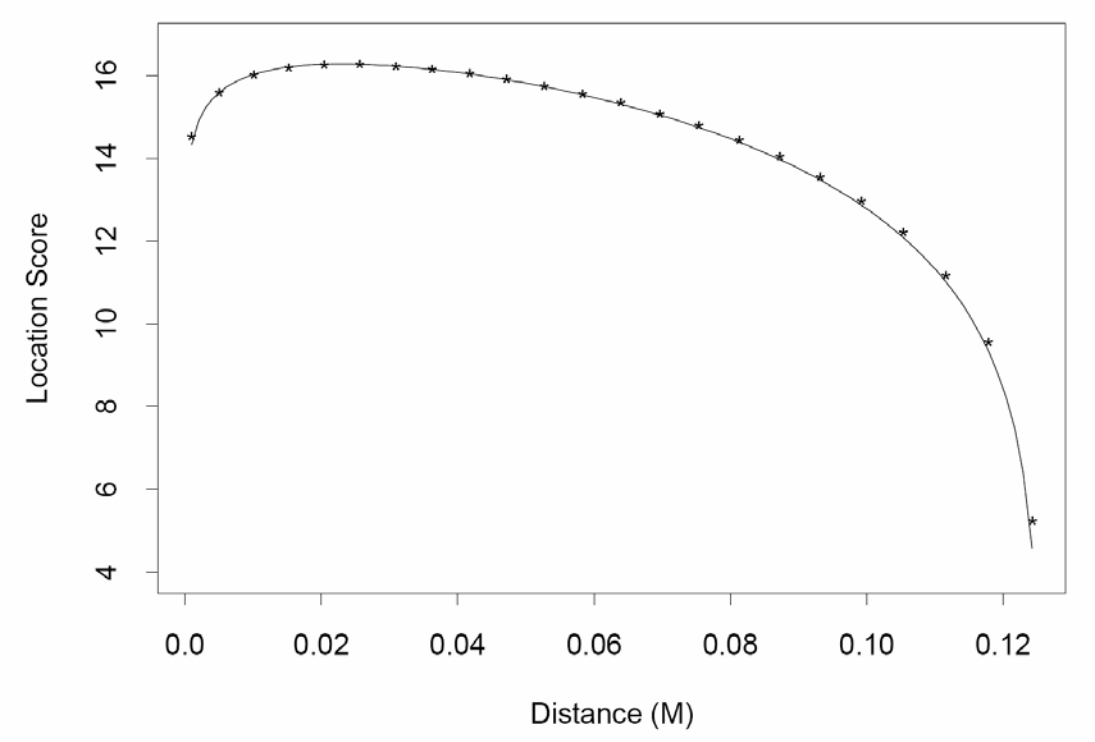
Two approaches:

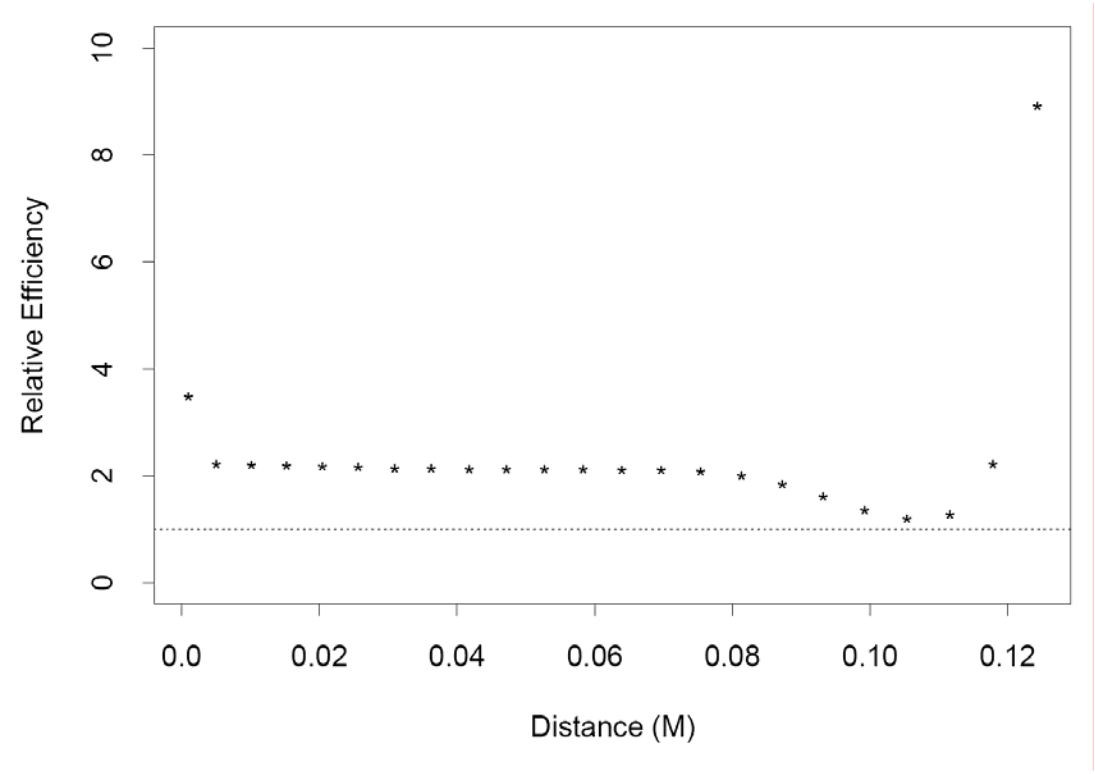
1) Process all marker data first then the disease data

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

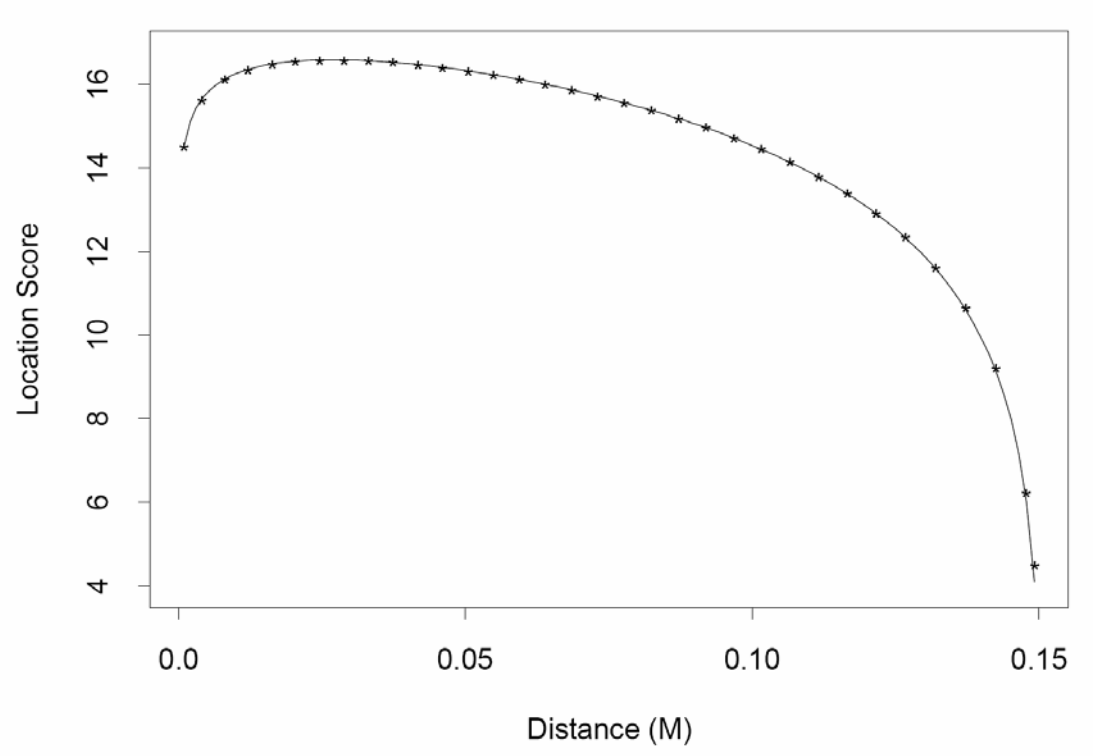
In both approaches the disease was processed in the middle of the marker interval of interest and the likelihood for other points in the interval were determined by reweighting the sample

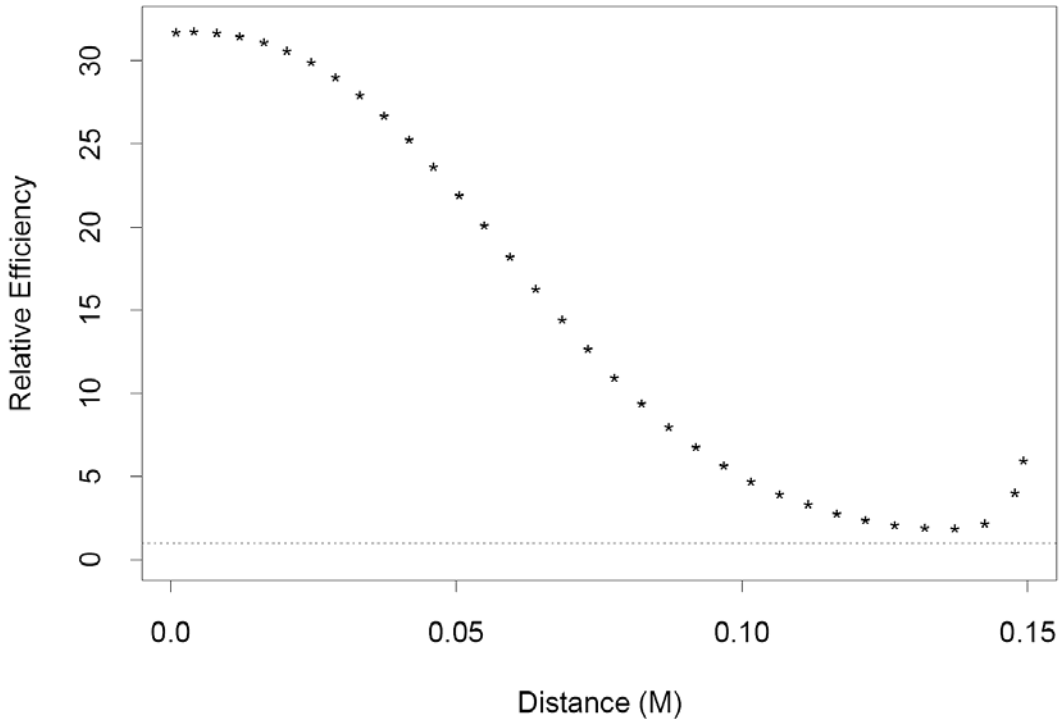
CEPH Distances:





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

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

instead of

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

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

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: $[X|Y]$

Data: $[Z_x, Z_y | X, Y] = [Z_x | X][Z_y | Y]$

Want to sample X and Y from $[X, Y | Z_x, Z_y]$.

One possible scheme is to use the following SIS scheme

1) Sample X from $[X|Z_x]$ by SIS giving weights $w_x(X)$.

2) Sample Y from $[Y|X, Z_x, Z_y]$. Given the probability structure above

$$[Y|X, Z_x, Z_y] = [Y|X, Z_y]$$

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

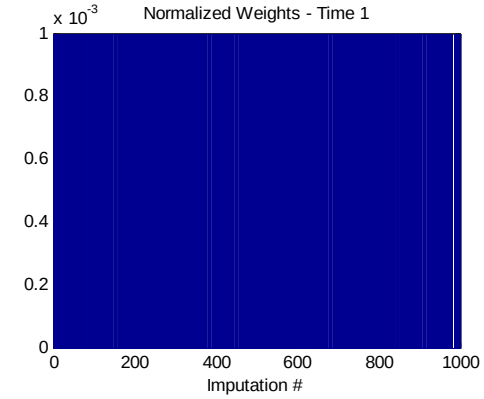
$$w_{x,y}(X, Y) = w_x(X)$$

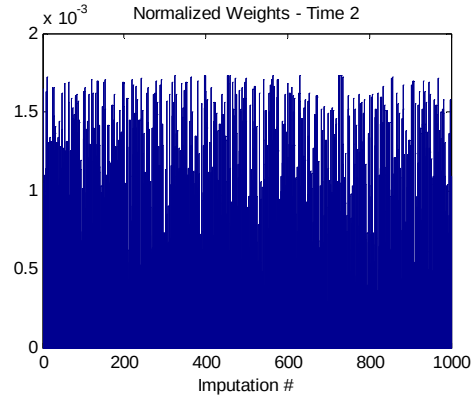
i.e. the simulation of Y in this case won't increase the variance of the importance sampling weights.

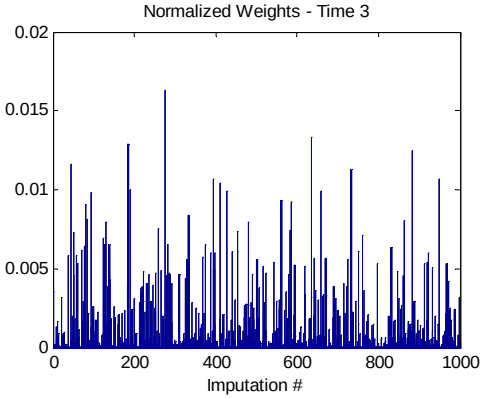
I have been able to do this with some genetics example, where X are the haplotypes, and Y is the inheritance vector.

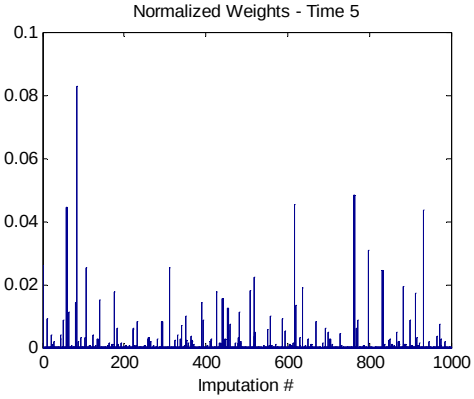
However this idea won't work in the target tracking example.

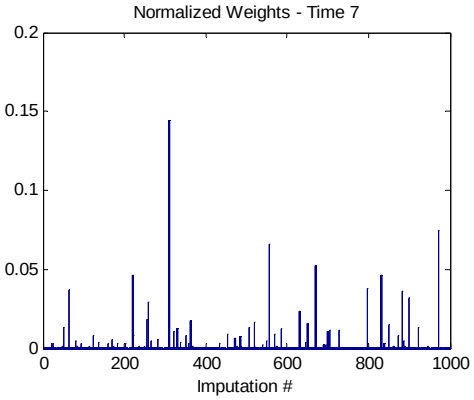
Lets look at how the normalized importance sampling weights can evolve over time in the target tracking example.

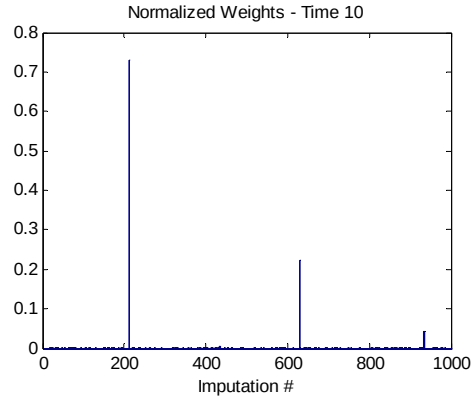

x 10^-3
Normalized Weights - Time 1
1
0.8
0.6
0.4
0.2
0
0
200
400
600
800
1000
Imputation #


x 10^-3
Normalized Weights - Time 2
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 #

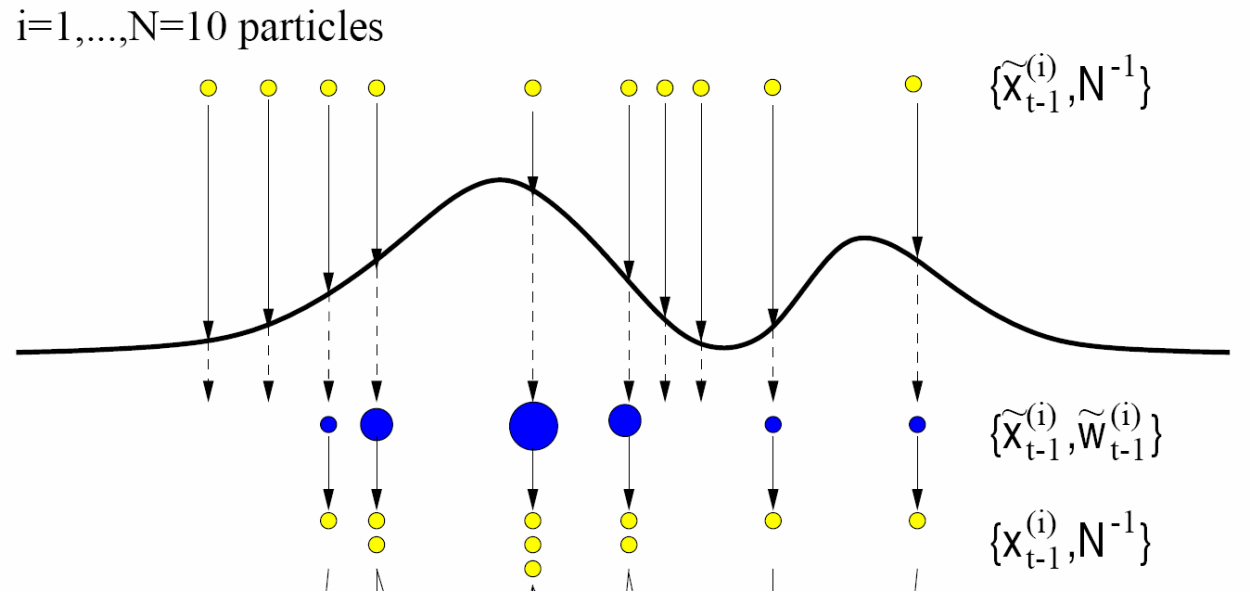

Normalized Weights - Time 7
0.2
0.15
0.1
0.05
0
0
200
400
600
800
1000
Imputation #


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



(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 $i = 1, \ldots, n$

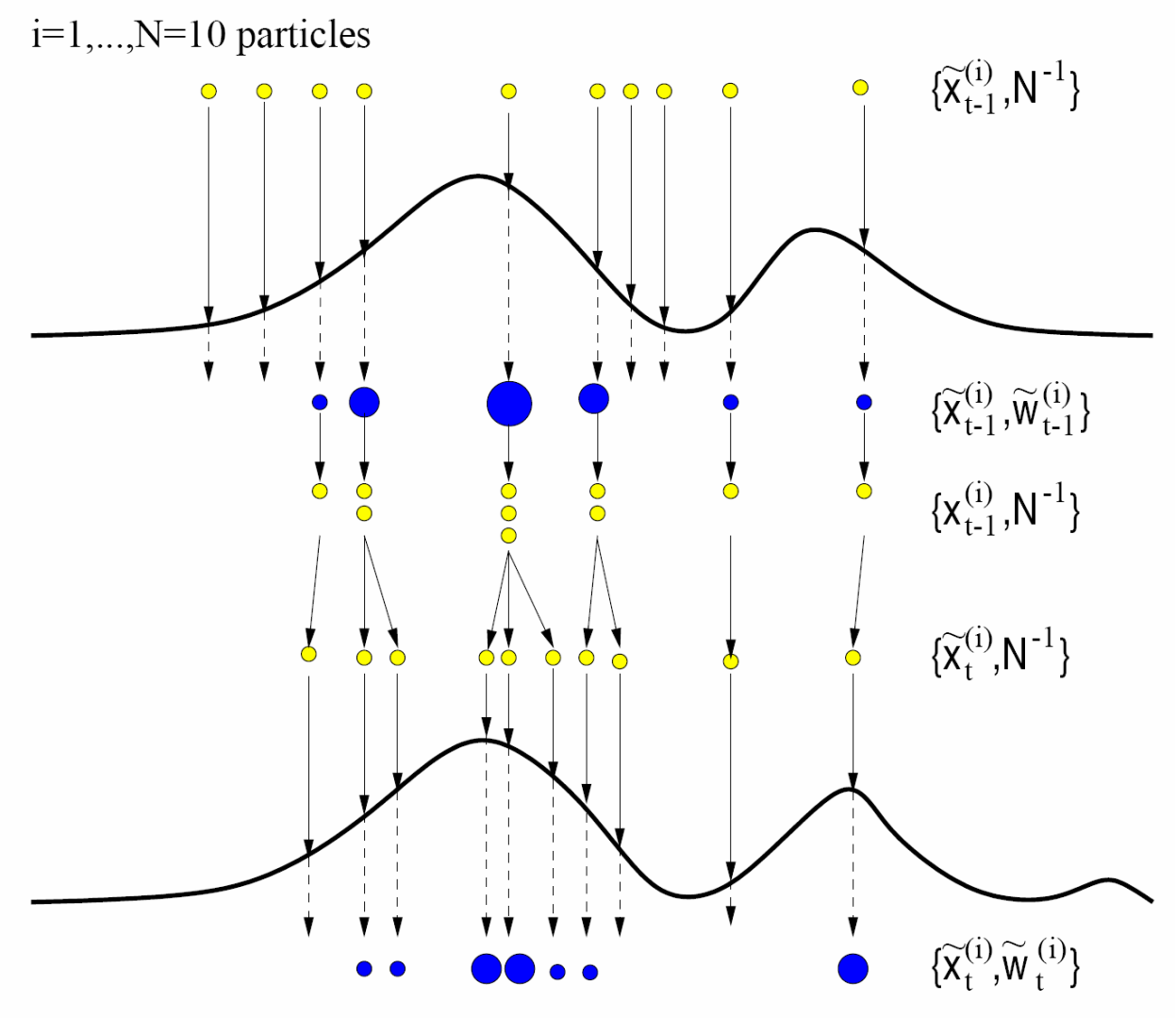
1) Sample $X_j^i \sim q_j\left(X_j \middle| Y_{1:j}, X_{1:j-1}^i\right)$

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



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

1) resample every m times through ($j = m, 2m, \ldots$)

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

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

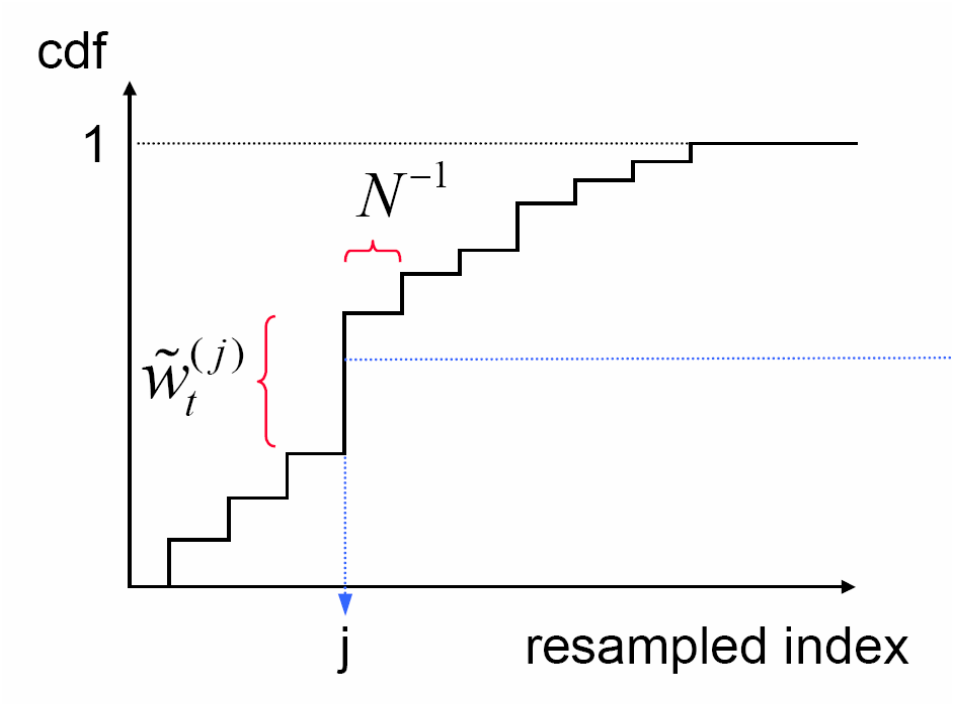
   where l_j is the number of copies of X^j in the new sample

This is equivalent to

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

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

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



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

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

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

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

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

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

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

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 $\left\{l_j\right\}$