The simplest PF and how bad / good it is.

Setting: Model : $x_k = f(x_{k-1}) + v_k, \quad v_k \sim N(0, Q)$

obs. : $y_k = h(x_k) + \eta_k, \quad \eta_k \sim N(0, R)$

target : $p(x_{0:k} | y_{1:k}) \propto p(x_{0:k-1} | y_{1:k-1}) p(x_k | x_{k-1}) p(y_k | x_k)$

proposal : $q(x_{0:k} | y_{1:k}) \propto q(x_0) \prod_{j=1}^{k} q(x_j | x_{j-1}, y_j)$

We still need to pick a $q(x_j | x_{j-1}, y_j)$

This is one choice (the original SIR, by now forgotten upon!)

$$q(x_j | x_{j-1}, y_j) = p(x_j | x_{j-1})$$

↝ "forecast ensemble"

↝ use model, without most recent obs, to propose states.

weights:

$$w_k \propto w_{k-1} \frac{p(x_k | x_{k-1}) p(y_k | x_k)}{p(x_k | x_{k-1})} \propto w_{k-1} p(y_k | x_k)$$

In SIR setting:

$\{x_{k-1}^j\}$ an ensemble.

$x_k^j \sim p(x_k | x_{k-1}^j)$ ↝ "Run the stochastic model"

$w^j \propto p(y_k | x_k^j) \propto \exp\left(-\frac{1}{2}(y_k - h(x_k^j))^T R^{-1} (y_k - h(x_k^j))\right)$

$\Rightarrow \{x_k^j, w^j\}$

Resample: $\{x_k^j\} \sim p(x_k | y_{1:k})$ ✓

Pros: · Easy to implement
(no optimization, just simulation + computing weights)
· flexible : not many assumptions about x_0, η_t, f, h

Question: How well does this work?

Simplifying assumptions:

(1) Recall: $N_{eff} = N_e / \delta$, $\delta = \frac{E[\omega^2]}{E[\omega]^2}$

$= \frac{var(\omega)}{E[\omega]^2} + 1$

→ the larger variance of weights, the worse the algorithm.

→ Keep of faith: if $var(-\log(\omega))$ is large, then $var(\omega)$ is even larger!

$var(-\log \omega) = C$

$\rightsquigarrow var(\omega) \propto \exp(C)$

(we don't get into this in detail)

(2) Suppose that: $y_t = Hx_t + \eta_t$, $\eta \sim N(0, R)$

$y_t - Hx_t \sim N(0, R)$

$R^{-1/2}(y_t - Hx_t) \sim N(0, I)$ ← l-dimensional, l = # of obs!

$$\text{var}(-\log w) = \text{var}\left(\tfrac{1}{2}\underbrace{(y_k - Hx_k)^T R^{-1}(y_k - Hx_k)}_{S^T S,\ \text{where } S = R^{-1/2}(y - Hx) \sim N(0, I)} \right)$$

$$= \text{var}\left(\tfrac{1}{2} S^T S \right) = \text{var}\left(\tfrac{1}{2} \sum_{j=1}^{\ell} S_j^2 \right)$$

($\uparrow$ components of S)

$$= \frac{1}{4} \sum_{j=1}^{\ell} \text{var}(S_j^2)$$

($2 S_j \sim N(0,1)$)

$$\text{var}(S_j^2) = E[S_j^4] - E(S_j^2)^2 = 3 - 1 = 2$$

$$\Rightarrow \text{var}(-\log w) = \frac{1}{4} \sum_{j=1}^{\ell} 2 = \underline{\frac{1}{2}\ell}$$

$\Rightarrow$ $\text{var}(-\log w)$ scales linearly with ℓ, the # of obs.

$\text{var}(w)$ scales exponentially with ℓ, the # of obs.

$\rho = \frac{\text{var}(w)}{E[w]^2} + 1$ scales exponentially with ℓ.

$$\Rightarrow \boxed{N_e \propto \exp(\ell)}$$

"not surprisingly, we already know that sampling in high-D is exponentially hard."

↳ Required ensemble size scales exponentially with ℓ.

↳ not a good algo. if # of obs is large.

↳ has been successful in applications with low-dimension.