Sequential Monte Carlo Methods - A Review

Arnaud Doucet

Engineering Department, Cambridge University, UK

`http://www-sigproc.eng.cam.ac.uk/~ad2/arnaud_doucet.html`

`ad2@eng.cam.ac.uk`

Institut Henri Poincaré - Paris - 2 Décembre 2002.

Outline of the Talk

1. Objectives.
2. Sequential Importance Sampling.
3. Resampling.
4. Sampling while looking ahead.
5. Variance Reduction.
6. Smoothing.
7. Sequential Monte Carlo Samplers (with P. Del Moral).

- Estimate expectations

$$\int f\left(x_{0:n}\right)\pi_n\left(dx_{0:n}\right)$$

with respect to a sequence of probability distributions $\{\pi_n\}_{n\geq 0}$ known *up to a normalizing constant*, i.e.

$$\pi_n\left(dx_{0:n}\right)=\pi_n\left(x_{0:n}\right)dx_{0:n}\text{ where }x_{0:n}=\left(x_0,x_1,\ldots,x_n\right).$$

- **Monte Carlo methods**: Obtain N weighted samples $\left\{X_{0:n}^{(i)},W_n^{(i)}\right\}_{i=1,\ldots,N}$

($W_n^{(i)}>0$, $\sum_{i=1}^N W_n^{(i)}=1$) such that

$$\sum_{i=1}^N W_n^{(i)} f\left(X_{0:n}^{(i)}\right)\underset{N\rightarrow\infty}{\longrightarrow}\int f\left(x_{0:n}\right)\pi_n\left(dx_{0:n}\right)$$

- Sequential Monte Carlo Methods: Having $\left\{X_{0:n-1}^{(i)}, W_{n-1}^{(i)}\right\}_{i=1,\ldots,N}$ from π_{n-1}, obtain $\left\{X_{0:n}^{(i)}, W_{n}^{(i)}\right\}_{i=1,\ldots,N}$ from π_n.

- "Real-time" methods, i.e. non-iterative, good for us engineers.

- Even for batch problems, it can provide better results than MCMC.

- Combination of importance sampling, resampling and MCMC.

⇒ **SMC are not a BLACK-BOX!!!**

1.3– The optimal filtering case

- Let $\{X_n\}_{n\geq 0}$ and $\{Y_n\}_{n\geq 1}$ be $\mathbb{R}^p$ and $\mathbb{R}^q$-valued stochastic processes

$$\textit{Evolution equation:} \quad \Pr\left(X_{n+1} \in A \middle| X_{0:n}, Y_{1:n} \right) = \int_A f\left(x \middle| X_n \right) dx,$$

$$\textit{Observation equation:} \quad \Pr\left(Y_{n+1} \in B \middle| X_{0:n+1}, Y_{1:n} \right) = \int_B g\left(y \middle| X_{n+1} \right) dy.$$

- Sequence of posterior distributions $\Pr\left(X_{0:n} \in dx_{0:n} \middle| Y_{1:n} = y_{1:n} \right) = p\left(x_{0:n} \middle| y_{1:n} \right) dx_{0:n}$

$$p\left(x_{0:n} \middle| y_{1:n} \right) \propto \underbrace{f\left(x_0 \right) \prod_{k=1}^{n} f\left(x_k \middle| x_{k-1} \right)}_{\text{prior}} \underbrace{\prod_{k=1}^{n} g\left(y_k \middle| x_k \right)}_{\text{likelihood}}$$

1.4– Application of Sequential Monte Carlo Methods

- Optimal filtering: the most popular application of SMC.
- Other applications:
  - Polymer design, Protein folding, Quantum physics.
  - Bioinformatics, Genetics, Rare events.
  - Computing eigenmeasure of positive operators.

- Cannot sample from $\pi_n(x_{0:n})$! $\Rightarrow$ Importance Sampling

- Introduce a pdf $q_n(x_{0:n}) > 0 \Rightarrow \pi_n(x_{0:n}) > 0$ then

$$\pi_n(x_{0:n}) = \frac{w_n(x_{0:n})\, q_n(x_{0:n})}{\int w_n(x_{0:n})\, q_n(x_{0:n})\, dx_{0:n}}$$

where

$$\textit{Importance weight:} \qquad w_n(x_{0:n}) \propto \pi_n(x_{0:n}) / q_n(x_{0:n})$$

- If $X_{0:n}^{(i)} \sim q_n$ then

$$\widehat{\pi}_n^N(dx_{0:n}) = \sum_{i=1}^N W_n^{(i)} \delta_{X_{0:n}^{(i)}}(dx_{0:n}),\ W_n^{(i)} \propto w_n\left(X_{0:n}^{(i)}\right),\ \ \sum_{i=1}^N W_n^{(i)} = 1.$$

- To be sequential ⇒ Weights must be computed sequentially!

$$
\begin{aligned}
w_n\left(x_{0:n}\right) &\propto \pi_n\left(x_{0:n}\right)/q_n\left(x_{0:n}\right) \\
&\propto \frac{\pi_n\left(x_{0:n}\right)}{q_n\left(x_{0:n}\right)}\frac{q_{n-1}\left(x_{0:n-1}\right)}{\pi_{n-1}\left(x_{0:n-1}\right)}w_{n-1}\left(x_{0:n-1}\right).
\end{aligned}
$$

- Only impute x_n at time n

$$
q_n\left(x_{0:n}\right) = q_{n-1}\left(x_{0:n-1}\right) q_n\left(\left. x_n \right| x_{0:n-1}\right),
$$

⇒

$$
w_n\left(x_{0:n}\right) \propto \frac{\pi_n\left(x_{0:n}\right)}{\pi_{n-1}\left(x_{0:n-1}\right) q_n\left(\left. x_n \right| x_{0:n-1}\right)} w_{n-1}\left(x_{0:n-1}\right).
$$

2.3– Sequential Importance Sampling for Optimal Filtering

- One has $\pi_n(x_{0:n}) = p(x_{0:n}|y_{1:n})$ and

$$q_n(x_{0:n}|y_{1:n}) = q_{n-1}(x_{0:n-1}|y_{1:n-1})\, q_n(x_n|x_{0:n-1}, y_{1:n})$$

- The importance weight is

$$\begin{aligned} w_n(x_{0:n}) &\propto \frac{\pi_n(x_{0:n})}{\pi_{n-1}(x_{0:n-1})\, q_n(x_{0:n})} w_{n-1}(x_{0:n-1}) \\ &\propto \frac{f(x_n|x_{n-1})\, g(y_n|x_n)}{q_n(x_n|x_{0:n-1}, y_{1:n})} w_{n-1}(x_{0:n-1}). \end{aligned}$$

- *Sampling Step.* For $i = 1, ..., N$, sample $X_n^{(i)} \sim q_n\left(\cdot | X_{0:n-1}^{(i)}, y_{1:n}\right)$ and update

the importance weights

$$W_n^{(i)} \propto \frac{f\left(X_n^{(i)} \middle| X_{n-1}^{(i)}\right) g\left(y_n | X_n^{(i)}\right)}{q_n\left(X_n^{(i)} \middle| X_{0:n-1}^{(i)}, y_{1:n}\right)} W_{n-1}^{(i)}$$

- The empirical distribution

$$\widehat{p}^N\left(dx_{0:n} | y_{1:n}\right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_{0:n}^{(i)}}\left(dx_{0:n}\right)$$

is an approximation of $p\left(x_{0:n} | y_{1:n}\right)$

2.5– Choice of the Importance Density

- The closer $q_n(x_{0:n})$ is from $\pi_n(x_{0:n})$ the better it works...

- One has

$$w_n(x_{0:n}) \propto \frac{\pi_n(x_n|x_{0:n-1})}{q_n(x_n|x_{0:n-1})} \frac{\pi_n(x_{0:n-1})}{\pi_{n-1}(x_{0:n-1})} w_{n-1}(x_{0:n-1})$$

$\Rightarrow$ Obvious choice $q_n(x_n|x_{0:n-1}) = \pi_n(x_n|x_{0:n-1})$ but requires knowing $\pi_n(x_{0:n-1})$

$\Rightarrow$ Filtering case: $q_n(x_n|x_{0:n-1}, y_{1:n}) \propto g(y_n|x_n) f(x_n|x_{n-1})$

- *Prior distribution.*

$$q_n\left(x_n \mid x_{0:n-1}, y_{1:n}\right) = f\left(x_n \mid x_{n-1}\right), w_n\left(x_{0:n}\right) \propto w_{n-1}\left(x_{0:n-1}\right) g\left(y_n \mid x_n\right).$$

- *"Optimal" distribution.*

$$\begin{aligned} q_n\left(x_n \mid x_{0:n-1}, y_{1:n}\right) &\propto g\left(y_n \mid x_n\right) f\left(x_n \mid x_{n-1}\right), \\ w_n\left(x_{0:n}\right) &\propto w_{n-1}\left(x_{0:n-1}\right) \int g\left(y_n \mid x_n\right) f\left(x_n \mid x_{n-1}\right) dx_n. \end{aligned}$$

- *Alternative suboptimal distributions.* Use EKF, UKF or anything you want to approximate the optimal distribution.

$\Rightarrow$ Advantages of Monte Carlo: still theoretically valid!

• Dimension of the state space increases with $n \Rightarrow$ **importance sampling collapse**.

$\Rightarrow$ Needs for resampling - **KEY element of SMC methods**.

• *Basic idea of resampling.*

• At time n, **multiply** particles $X_{0:n}^{(i)}$ with **high** weights $W_n^{(i)}$ and **discard** ones with **small** weights.

• Give to the resampled particles an equal weight.

• Keep the number of particles fixed.

At time n,

$$\widehat{\pi}_n^N (dx_{0:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_{0:n}^{(i)}} (dx_{0:n})$$

- Sample N times from $\widehat{\pi}_n^N$ to obtain N new samples.

- Equivalent to copy $X_{0:n}^{(i)}$ k_i times where

$$(k_1, \ldots, k_N) \sim \mathcal{M}\left(N; W_n^{(1)}, \ldots, W_n^{(N)}\right).$$

- One has $E[k_i] = NW_n^{(i)}$ but large variance $var[k_i] = NW_n^{(i)}\left(1 - W_n^{(i)}\right)$.

- What you really want

$$\sum_{i=1}^{N} W_n^{(i)} \delta_{X_{0:n}^{(i)}}(dx_{0:n}) \approx \sum_{i=1}^{N} \frac{k_i}{N} \delta_{X_{0:n}^{(i)}}(dx_{0:n}) \text{ where } \sum_{i=1}^{N} k_i = N,\ k_i \in \mathbb{N}.$$

- *Residual resampling* (Baker 1987). Set $k_i = \left\lfloor N W_n^{(i)} \right\rfloor + \widetilde{k}_i$ where $\left(\widetilde{k}_1, \ldots, \widetilde{k}_N\right) \sim \mathcal{M}\left(N; \widetilde{W}_n^{(1)}, \ldots, \widetilde{W}_n^{(N)}\right)$; $\widetilde{W}_n^{(i)} \propto N W_n^{(i)} - \left\lfloor N W_n^{(i)} \right\rfloor$.

- *Stratified/Systematic Sampling* (Wittley 1994, Kitagawa 1996, Carpenter *et al.* 1999): 2 lines of code... very efficient.

- *Mimimum entropy sampling* (Crisan 2001).

• Keep some information about the weights. If k_i copies, give a new weight

$$\widetilde{W}_n^{(i)} \propto \frac{1 + \left(NW_n^{(i)} - \left\lfloor NW_n^{(i)} \right\rfloor \right) / Nk_i}{N}$$

• If a particle of weight $1.7/N$ is copied two times then one substracts $0.3/N$ mass.

$$\widetilde{W}_n^{(i)} \propto \frac{(1 - 0.3/2)}{N}$$

• If a particle of weight $1.7/N$ is copied one time then one adds $0.7/N$ mass.

3.5– SIS + Resampling for Optimal Filtering

- *Sampling Step.* For $i = 1, ..., N$, set $\widetilde{X}_{0:n-1}^{(i)} = X_{0:n-1}^{(i)}$, sample $\widetilde{X}_n^{(i)} \sim q_n \left(\cdot | \, \widetilde{X}_{0:n-1}^{(i)}, y_{1:n} \right)$

$$W_n^{(i)} \propto \frac{f\left(\widetilde{X}_n^{(i)} \middle| \, \widetilde{X}_{n-1}^{(i)} \right) g\left(y_n | \, \widetilde{X}_n^{(i)} \right)}{q_n\left(\widetilde{X}_n^{(i)} \middle| \, \widetilde{X}_{0:n-1}^{(i)}, y_{1:n} \right)}$$

- *Selection Step.* Multiply/Suppress samples $\left\{ \widetilde{X}_{0:n}^{(i)} \right\}$ respectively with high/low weights $\left\{ W_n^{(i)} \right\}$ to obtain $\left\{ X_{0:n}^{(i)} \right\}$.

- $\widehat{p}^N \left(dx_{0:n} | \, y_{1:n} \right) = \frac{1}{N} \sum_{i=1}^N \delta_{X_{0:n}^{(i)}} \left(dx_{0:n} \right)$ is an approximation of $p\left(x_{0:n} | \, y_{1:n} \right)$.

3.6– Using MCMC to prevent degeneracy

- If the distribution of the weights $w_n^{(i)}$ is skewed $\Rightarrow$ A few particles dominate; i.e. poor approximation.

- To prevent degeneracy

  - Use kernel approximation $\widehat{p}^N\left(x_{0:n}\middle| y_{1:n}\right) = \frac{1}{N}\sum_{i=1}^{N} K_h\left(x_{0:n} - X_{0:n}^{(i)}\right)$...

    It modifies the target distribution.

  - Use MCMC step: no perturbation of the target

$$X_{0:n}^{\prime(i)} \sim K_n\left(\cdot\middle| X_{0:n}^{(i)}\right), \text{ where } \pi_n\left(dx'_{0:n}\right) = \int K_n\left(dx'_{0:n}\middle| x_{0:n}\right)\pi_n\left(dx_{0:n}\right).$$

4.1– Look one-step ahead before sampling

- Consider the "optimal" case where $q_n(x_n | x_{0:n-1}) = \pi_n(x_n | x_{0:n-1})$

$\Rightarrow w_n(x_{0:n}) \propto \pi_n(x_{0:n-1}) / \pi_{n-1}(x_{0:n-1})$ independent of x_n!

$\Rightarrow$ Sampling then resampling inefficient.
$\Rightarrow$ Resampling then sampling makes sense.

- In the case of optimal filtering

$$q_n(x_n | x_{0:n-1}, y_{1:n}) \propto g(y_n | x_n) f(x_n | x_{n-1}),$$

$$w_n(x_{0:n}) \propto \int g(y_n | x_n) f(x_n | x_{n-1}) dx_n.$$

i.e. look at what is happening one-step-ahead and sample in this region.

4.2– Look one-step ahead before sampling

• In the general case, one cannot compute $w_n(x_{0:n}) \propto \pi_n(x_{0:n-1})/\pi_{n-1}(x_{0:n-1})$, $\pi_n(x_{0:n-1})$ requires integration!

• *Basic idea.* Approximate it, use say $\left\{\widehat{W}_n^{(i)}\right\}$, and resample $\left\{\widetilde{X}_{0:n-1}^{(i)}\right\}$ according to these weights.

$$\sum_{i=1}^{N} W_{n-1}^{(i)} \delta_{\widetilde{X}_{0:n-1}^{(i)}}(dx_{0:n-1}) \stackrel{\text{Resample}}{\longrightarrow} \sum_{i=1}^{N} \alpha_n^{(i)} \delta_{X_{0:n-1}^{(i)}}(dx_{0:n-1})$$

where

$$\alpha_n^{(i)} \propto W_{n-1}^{(i)}/\widehat{W}_n^{(i)}, \ \sum_{i=1}^{N} \alpha_n^{(i)} = 1.$$

4.3– Auxiliary Particle Filter-Like

- *Selection Step.* Set $\widetilde{X}_{0:n-1}^{(i)} = X_{0:n-1}^{(i)}$, Multiply/Suppress samples $\left\{\widetilde{X}_{0:n-1}^{(i)}\right\}$ respectively with high/low weights $\left\{\widehat{W}_n^{(i)}\right\}$ to obtain $\left\{X_{0:n-1}^{(i)}\right\}$.

- *Sampling Step.* For $i = 1, ..., N$, sample $X_n^{(i)} \sim q\left(\cdot | X_{0:n-1}^{(i)}, y_{1:n}\right)$

$$W_n^{(i)} \propto \frac{W_{n-1}^{(i)}}{\widehat{W}_n^{(i)}} \frac{f\left(X_n^{(i)} \middle| X_{n-1}^{(i)}\right) g\left(y_n | X_n^{(i)}\right)}{q_n\left(X_n^{(i)} \middle| X_{0:n-1}^{(i)}, y_{1:n}\right)}$$

- $\widehat{p}^N\left(dx_{0:n} | y_{1:n}\right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_{0:n}^{(i)}}\left(dx_{0:n}\right)$ is an approximation of $p\left(x_{0:n} | y_{1:n}\right)$.

4.4– Why only one-step ahead?

- If it is expected that $\pi_{n+L}(x_{0:n-1})$ very different from $\pi_{n-1}(x_{0:n-1})$, multiple step-ahead.

- Boost particles using approximation of $\pi_{n+L}(x_{0:n-1}) / \pi_{n-1}(x_{0:n-1})$.

- Sample particles with approximation of $\pi_{n+L}(x_n | x_{n-1})$.

4.5– Modifying the past of the trajectories

- If large discrepancy between $\pi_{n-1}(x_{0:n-1})$ and $\pi_n(x_{0:n-1})$: all methods are inefficient!

- At time n, we might want to sample new trajectories according to say $q_n(dx'_{0:n}|x_{0:n-1}) \neq \delta_{x_{0:n-1}}(dx'_{0:n-1}) q_n(dx'_n|x_{0:n-1})$.

- *Problem*: If $q_n(dx'_{0:n}|x_{0:n-1})$ is a general kernel then the importance distribution

$$\int q_n(dx'_{0:n}|x_{0:n-1}) \pi_{n-1}(dx_{0:n-1})$$

has not analytical expression.

$\Rightarrow$ Importance sampling on an extended space.

4.6– "Phantom kernel" - More freedom for my particles

- Artificial joint measure

$$\underbrace{\pi_n\left(dx'_{0:n}\right)}_{\text{fixed}}\underbrace{L_n\left(\left. dx_{0:n-1}\right| x'_{0:n}\right)}_{\text{Phantom kernel}}$$

- Importance weights

$$w_n\left(x_{0:n}, x'_{0:n}\right) \propto \frac{\pi_n\left(dx'_{0:n}\right) L_n\left(\left. dx_{0:n-1}\right| x'_{0:n}\right)}{\pi_{n-1}\left(dx_{0:n-1}\right) q_n\left(\left. dx'_{0:n}\right| x_{0:n-1}\right)}$$

- Application to parameter estimation $\pi_n\left(d\theta\right)$ (see SMC samplers)

$$w_n\left(\theta, \theta'\right) \propto \frac{\pi_n\left(d\theta'\right) L_n\left(\left. d\theta\right| \theta'\right)}{\pi_{n-1}\left(d\theta\right) q_n\left(\left. d\theta'\right| \theta\right)}.$$

- *Control Variate.*

- *Antithetic Variables.* $var[A+B] = var[A] + var[B] + 2cov[A,B]$.

- *Quasi Monte Carlo*. Explore more uniformly the space.

- *Rao-Blackwellisation.* Integrate out analytically variables whenever you can.

- *Importance Sampling.* To compute $\int f(x)\,\pi(dx)$, using samples from $\pi(dx)$

is not the best choice.

A Good Monte Carlo is a Dead Monte Carlo - Trotter.

Assume $x_n = (u_n, v_n)$ so

$$\pi_n(x_{0:n}) = \pi_n(u_{0:n}, v_{0:n}) = \pi_n(v_{0:n}|\, u_{0:n}) \underbrace{\pi_n(u_{0:n})}_{\text{known up to a normalizing constant}}.$$

Two possible estimates of $\int f_n(v_{0:n})\, \pi_n(dv_{0:n})$

$$I_1 = \frac{1}{N} \sum_{i=1}^{N} f_n\left(V_{0:n}^{(i)}\right), \qquad I_2 = \frac{1}{N} \sum_{i=1}^{N} E\left[f_n(V_{0:n})|\, U_{0:n}^{(i)}\right]$$

One has

$$var[I_2] \leq var[I_1] \text{ as}$$

$$var[f_n(V_{0:n})] = var[\underbrace{E[f_n(V_{0:n})|\, U_{0:n}]}_{\text{corresponds to } I_2}] + \underbrace{E[var[f_n(V_{0:n})|\, U_{0:n}]]}_{\geq 0}$$

- Conditionally Linear Gaussian State Space Models

$$U_n \text{ Markov process}$$

$$V_n = A(U_n) V_{n-1} + B(U_n) W_n, \; W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$$

$$Y_n = C(U_n) V_n + D(U_n) Z_n, \; Z_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$$

- One has

$$p(u_{0:n}, v_{0:n} | y_{1:n}) = \underbrace{p(v_{0:n} | y_{1:n}, u_{0:n})}_{\text{Gaussian distribution}} \underbrace{p(u_{0:n} | y_{1:n})}_{\text{known up to a normalizing constant}},$$

$$p(u_{0:n} | y_{1:n}) \propto \underbrace{p(y_{1:n} | u_{0:n})}_{\text{likelihood - Kalman filter}} p(u_{0:n}).$$

- Sample from $p(u_{0:n} | y_{1:n})$ and not from $p(u_{0:n}, v_{0:n} | y_{1:n})$ gain can be enormous!

- Standard Estimate: If $\left(U_{0:n}^{(i)}, V_{0:n}^{(i)}\right) \sim p\left(u_{0:n}, v_{0:n} \middle| y_{1:n}\right)$

$$\widehat{E}_1\left[V_n \middle| y_{1:n}\right] = \frac{1}{N}\sum_{i=1}^{N} V_n^{(i)}.$$

- Raoblackwell Estimate: If $U_{0:n}^{(i)} \sim p\left(u_{0:n} \middle| y_{1:n}\right)$

$$\widehat{E}_2\left[V_n \middle| y_{1:n}\right] = \frac{1}{N}\sum_{i=1}^{N} \underbrace{\widehat{E}\left[V_n \middle| y_{1:n}, U_{0:n}^{(i)}\right]}_{\text{Kalman}}.$$

- One has

$$var\left[\widehat{E}_2\left[V_n \middle| y_{1:n}\right]\right] \leq var\left[\widehat{E}_1\left[V_n \middle| y_{1:n}\right]\right]$$

and the gain will be high if the average variance of

$$E\left[var\left[V_n \middle| y_{1:n}, U_n\right]\right] \text{ large.}$$

- *Smoothing problems*: Sample from $\pi_n(dx_{0:n})$ / Estimate $\pi_n(dx_k)$.

- SMC methods yield

$$\widehat{\pi}_n^N(dx_{0:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_{0:n}^{(i)}}(dx_{0:n})$$

so

  - $X_{0:n} \sim \widehat{\pi}_n^N$ is an approximate sample from π_n.
  - $\widehat{\pi}_n^N(dx_k) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_k^{(i)}}(dx_k)$.

- **Problem**: As n is large, approximation of the joint distribution is BAD!!!.

6.2– Smoothing using Marginal Distributions

- **Aim**: Sampling from $p\left(\left.dx_{0:n}\right| y_{1:n}\right)$.

- *Forward Backward Sampling* from $p\left(\left.x_{0:n}\right| y_{1:n}\right)$ based on

$$p\left(\left.x_{0:n}\right| y_{1:n}\right) \propto p\left(\left.x_{n}\right| y_{1:n}\right) \prod_{k=0}^{n-1} \frac{f\left(\left.x_{k+1}\right| x_{k}\right) p\left(\left.x_{k}\right| y_{1:k}\right)}{p\left(\left.x_{k+1}\right| y_{1:k}\right)}$$

- Sample $X_n^* \sim \widehat{p}^N\left(\left.dx_n\right| y_{1:n}\right)$, then Sample for $k = n-1, \ldots, 0$

$$X_k^* \sim \frac{f\left(\left.X_{k+1}^*\right| x_k\right) \widehat{p}^N\left(\left.dx_k\right| y_{1:k}\right)}{\widehat{p}^N\left(\left.X_{k+1}^*\right| y_{1:k}\right)} = \frac{\sum_{i=1}^N W_k^{(i)} f\left(\left.X_{k+1}^*\right| X_{k+1}^{(i)}\right) \delta_{X_k^{(i)}}\left(dx_k\right)}{\sum_{i=1}^N W_k^{(i)} f\left(\left.X_{k+1}^*\right| X_{k+1}^{(i)}\right)}$$

6.3– Smoothing using Marginal Distributions

- **Aim**: Estimating $\pi_n(dx_k)$.
- *Forward Filtering Backward Smoothing*

$$p(x_k|\, y_{1:n}) = p(x_k|\, y_{1:k}) \int \frac{f(x_{k+1}|\, x_k)\, p(x_{k+1}|\, y_{1:n})}{p(x_{k+1}|\, y_{1:k})} dx_{k+1}$$

- *Two-Filter Formula* - Requires additional assumption.

$$p(x_k|\, y_{1:n}) \propto \underbrace{p(x_k|\, y_k)}_{\text{Filter}}\ \underbrace{p(y_{k+1:n}|\, x_k)}_{\text{Information Filter}}\ .$$

7.1– Sampling from a probability distribution

- **Aim:** Sample from a sequence $\pi_n(dx)$ (space of fixed dimension)
- **Solution**: Define

$$\pi_n(dx_{0:n}) \triangleq \pi_n(dx_n) \prod_{k=1}^{n} L_k(dx_{k-1}|\, x_k)$$

where

- $\pi_n(dx) = \pi(dx)$, $\pi_n(dx) \propto [\pi_0(x)]^{\gamma_n} [\pi(x)]^{1-\gamma_n} dx$ $(\gamma_n \to 1)$,

  $\pi_n(dx) \propto \pi^{\gamma_n}(x)\, dx$ $(\gamma_n \to \infty)$, $\pi_n(dx) = p(dx|\, y_{1:n})$.

- $\{L_n\}$ sequence of "phantom" kernels - free parameters.

7.2– Sequential Monte Carlo Sampler

- For $i = 1, ..., N$, set $\widetilde{X}_{0:n-1}^{(i)} = X_{0:n-1}^{(i)}$ and sample $\widetilde{X}_n^{(i)} \sim M_n\left(\widetilde{X}_{n-1}^{(i)}, \cdot\right)$ and

$$W_n^{(i)} \propto \frac{\pi_n\left(d\widetilde{X}_n^{(i)}\right) L_n\left(\widetilde{X}_n^{(i)}, d\widetilde{X}_{n-1}^{(i)}\right)}{\pi_{n-1}\left(d\widetilde{X}_{n-1}^{(i)}\right) M_n\left(\widetilde{X}_{n-1}^{(i)}, d\widetilde{X}_n^{(i)}\right)}, \; \sum_{i=1}^{N} W_n^{(i)} = 1.$$

- Multiply/Discard particles $\left\{\widetilde{X}_{0:n}^{(i)}\right\}$ with respect to high/low weights $\left\{W_n^{(i)}\right\}$ to obtain N particles $\left\{X_{0:n}^{(i)}\right\}$.

* P. Del Moral & A. Doucet, *On a Class of Genealogical and Interacting Metropolis Models*, to appear Lecture Notes in Maths, 2003.
* P. Del Moral & A. Doucet, *Sequential Monte Carlo Samplers*, TR Cambridge Univ., CUED-F-INFENG no. 444, 2002.

• *Annealed Importance Sampling* (Neal, 1998): No resampling, M_n MCMC kernel of invariant dist. π_n and

$$L_n(x, dx') = M_n(x', dx) \frac{\pi_n(dx')}{\pi_n(dx)} \Rightarrow W_n^{(i)} \propto \frac{\pi_k\left(d\widetilde{X}_{n-1}^{(i)}\right)}{\pi_{k-1}\left(d\widetilde{X}_{n-1}^{(i)}\right)}.$$

• *Population Monte Carlo* (TR. Cappé & al. 2002): Homogeneous case $M_n = M$, $L_n = L$ and $\pi_n = \pi$, M MCMC kernel of invariant dist. π

$$L(x, dx') = \pi(dx') \Rightarrow W_n^{(i)} \propto \frac{\pi\left(d\widetilde{X}_n^{(i)}\right)}{M\left(\widetilde{X}_{n-1}^{(i)}, d\widetilde{X}_n^{(i)}\right)}.$$

7.4– A Few Comments

- All SMC algorithms can be reused.
- Basic element of more complex algorithms.
- Many degrees of freedom $\Rightarrow$ Optimization of L given M...
- Many convergence results available; see Del Moral & Miclo's papers.

8.1– References

- SMC Website: `http://www-sigproc.eng.cam.ac.uk/smc/index.html`

- Algorithms: *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag, 2001.

- Theoretical results: P. Del Moral & L. Miclo `http://www.lsp.ups-tlse.fr/Delmoral/`

- Special Issue *Applied Signal Processing*, Deadline April 30, 2003.

- Many potential algorithmical and theoretical developments / Numerous applications.