Particle Filtering

1) Baye's Recursive Filter

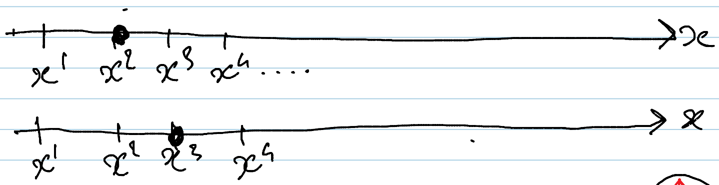
$$\boxed{\begin{aligned}&\text{Update}: \; p(x_k \mid y_k) \propto p(y_k \mid x_k) \cdot p(x_k \mid y_{k-1}) \\ &\text{Predict}: \; p(x_{k+1} \mid y_k) = \int p(x_{k+1} \mid x_k)\, p(x_k \mid y_k)\, dx_k\end{aligned}}$$

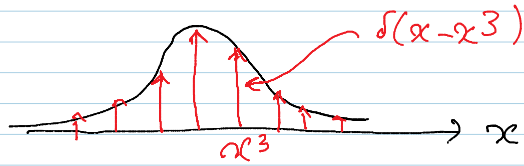
(Rappel)

2) Point Mass Filter

$k=1$

$k=2$





$$(k) \quad p(x) \simeq \sum_{i=1}^{N} q^i \, \delta(x^i - x)$$

$$p(x) \simeq \{ q^1 \; q^2 \; q^3 \ldots\ldots q^N \}$$

$$\sum q^i = 1$$

$$\hat{x} = E[p(x)] = \sum q^i x^i$$

$$\underbrace{\sigma^2 = \operatorname{cov}(x)}_{\text{scalaire}} = \sum q^i (x - x^i) \underbrace{(x - x^i)^T}_{\text{vecteur}}$$

↳ matrice $M \times M$
↳ dim(x)

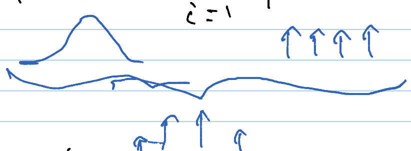
y: mesure
nombre

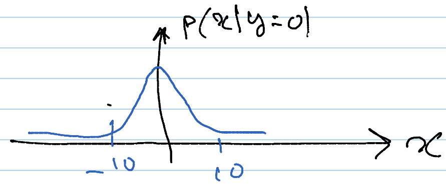
$$1| \; \text{Update} \quad p(x \mid y) \propto p(y \mid x) \cdot p(x)$$

↳ distribution initiale (prior)

↳ modèle de mesure

$$p(x \mid y) \overset{\simeq}{\propto} p(y \mid x) \sum_{i=1}^{N} q^i \, \delta(x - x^i)$$

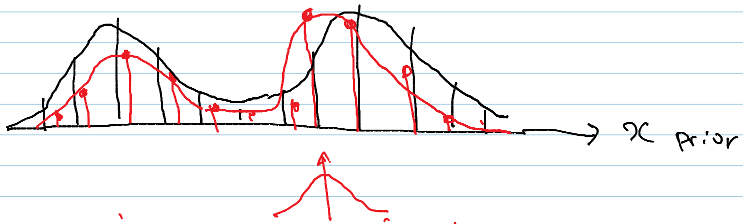




$$\overset{\simeq}{\propto} \sum_{i=1}^{N} p(y \mid x^i)\, q^i \, \delta(x - x^i)$$

$$p(x|y) \sim \{ \underbrace{p(y|x^1)\cdot q^1}_{=}, \; p(y|x^2)\, q^2, \; \ldots \; p(y|x^N)\, q^N \}$$

$$\sim \{ q^i_{update} \}$$



$$q^i_{update} = \frac{q^i_{update}}{\sum q^i_{update}}$$

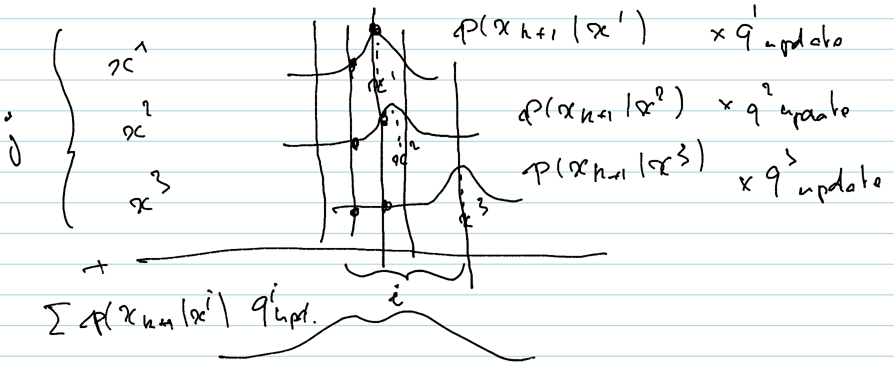
2) Predict

$$p(x_{k+1}|y) = \int p(x_{k+1}|x_k)\, \overbrace{p(x_k)}\, dx_k$$

$$\simeq \int p(x_{k+1}|x_k) \left(\sum q^i_{update}\, \delta(x^i - x_k) \right) dx_k$$

$$\simeq \int \sum_{i=1}^{N} p(x_{k+1}|x^i)\, q^i_{update}\, \overbrace{\delta(x^i - x_k)}^{\int = 1}\, dx_k$$

$$\simeq \sum_{i=1}^{N} p(x_{k+1}|x^i)\, q^i_{update}$$

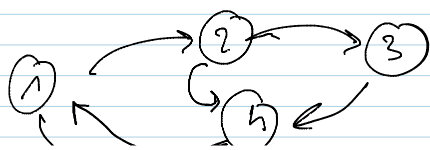


discretisation supplementaire

$$p(x_{k+1}|y_k) \sim q^i_{predict}$$

$$q^i_{predict} = \underbrace{\sum_{j=1}^{N} p(x^i_{k+1}|x^j_k)\, q^j_{update}}_{\text{bcp de calculs}}$$

* Pas adapté pour les grands espaces d'état
* Ok pour petits ______

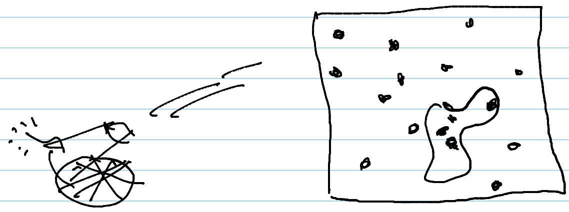


3) Importance sampling

$$I = \int f(x)\,\pi(x)\,dx \quad \text{où} \quad \int \pi(x) = 1$$

$$\simeq \sum f(x^i) \quad \text{où} \quad \underbrace{x^i \sim \pi(x)}$$

x^i tiré aléatoirement selon $\pi(x)$



4) Particle Filtering (general)

Algo General

* choix $\pi(x_{k+1} | x_k)$
* choix nombre de particules N (ex N 100 – 100000)
* modèle $p(x_{k+1} | x_k)$ (evolution)
  $p(y_k | x_k)$ (mesure)
* init x_0 $\qquad x_0^1 \quad x_0^2 \ldots x_0^N$
  $q_0 = \frac{1}{N}$ $\qquad \frac{1}{N} \quad \frac{1}{N} \quad \frac{1}{N}$
* Update : on dispose d'une mesure y_k

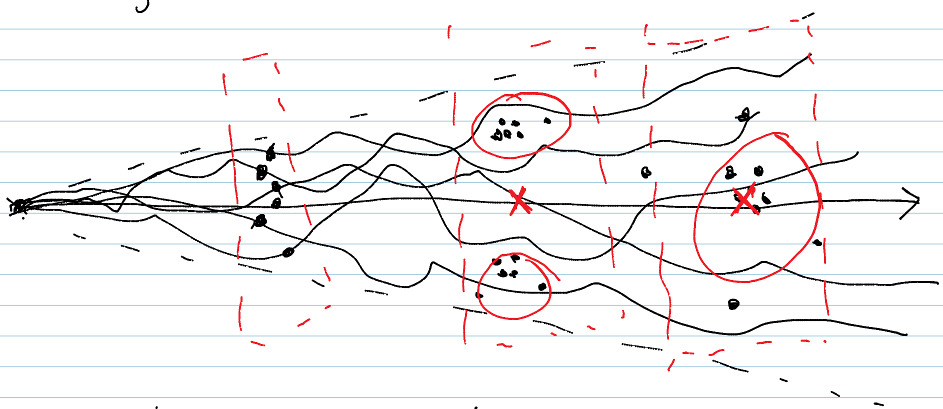
$$\boxed{q_k^i \propto q_k^i \cdot p(y_k | x_k^i)}$$

normalisation : $c = \sum_i q_k^i \qquad q_k^i = \frac{1}{c} q_k^i$

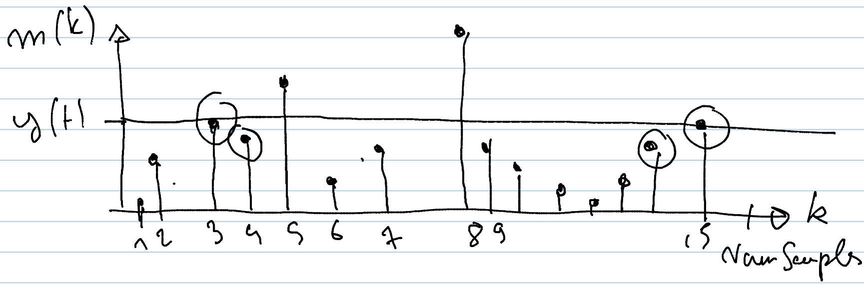
* Predict : on génère les particules $\boxed{x_{k+1}^i \text{ selon } \pi(x_{k+1} | x_k)}$
  on adapte les poids selon $\boxed{q_{k+1}^i = q_k^i \frac{p(x_{k+1}^i | x_k^i)}{\pi(x_{k+1}^i | x_k^i)}}$

Problemes : choix de π ? Bootstrap filter $\pi(x_{k+1}|x_k) = p(x_{k+1}|x_k)$
dégénérescence

dégénérescence

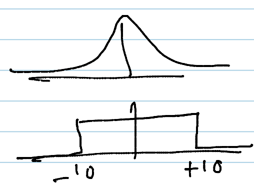


Update en pratique : à l'instant t



$\xi = m - y(t)$

probabilité d'avoir la mesure $y(t)$: $\frac{1}{\sqrt{2\pi..}} \cdot \exp\left[-\frac{1}{2R}(y-m)^2\right]$



$$\boxed{q = q .* \underbrace{\exp\left(-\frac{1}{2R}\left(y(t)-m\right)^2\right)}_{\text{modèle de capteur}}}$$

5) Resampling

Critère déclenchement resampling : quand le nombre de particules efficaces est trop faible : $N_{eff} = \frac{1}{\sum q_i^2}$ ex : tous les $q_i = \frac{1}{N}$ $N_{eff} = N$

——— 0 sauf $q_i = 1$: $N_{eff} = 1$

Algo resampling (1)

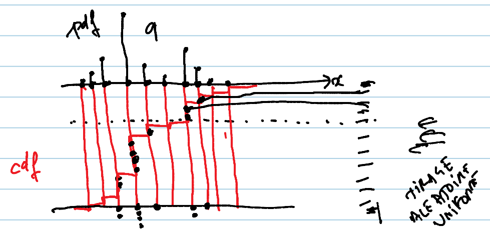
```matlab
function [ indx ] = resampleMultinomial( w )

M = length(w);
Q = cumsum(w);
Q(M)=1; % Just in case...

i=1;
while (i<=M),
    sampl = rand(1,1);  % (0,1]
    j=1;
    while (Q(j)<sampl),
        j=j+1;
    end;
    indx(i)=j;
    i=i+1;
end
```

https://fr.mathworks.com/matlabcentral/fileexchange/24968-resampling-methods-for-particle-filtering?focused=5136865&tab=functionb

Algo resampling (2)

```matlab
qc = [0 cumsum(q)];
[N, I] = histc(sort(rand(numSamples,1)), qc);
xu = xa(:, I);
```

$$q = \frac{1}{numSamples} .ones(1, numSamples);$$



I = [3 3 4 4 4 4 5 7 7 8]