

Premiers Pas en Filtrage Particulaire

Fabien Campillo



2022

version V.1 — 12 décembre 2022 — © Fabien Campillo

Ce document est mis à disposition selon les termes de la licence Creative Commons “Attribution – Pas d’utilisation commerciale – Partage dans les mêmes conditions 4.0 International”.



Table des matières


Table des matières	i
Avant-propos	iii
Notations	v
1 Quelques rappels de probabilités	1
1.1 Espace d'état fini	2
1.2 Espace d'état continu	5
2 Modèles de Markov cachés à espace d'état fini	17
2.1 MMC à espace d'observation au plus dénombrable	18
2.2 MMC à espace d'observation continu	30
3 Modèles de Markov cachés à espace d'état continu	37
3.1 Formule de Bayes récursive (filtre bayésien)	39
4 Filtre de Kalman et extensions	43
4.1 Système linéaire gaussien	43
4.2 Processus innovation	44
4.3 Filtre de Kalman	45
4.4 Système conditionnellement linéaire/gaussien	47
4.5 Filtre de Kalman étendu	48
4.6 Filtre de Kalman inodore (Unscented Kalman filter)	49
4.7 Filtre de Kalman d'ensemble	52
5 Approximation particulière	55
5.1 Méthode de Monte Carlo	55
5.2 Simulation de variables aléatoires	57
5.3 Échantillonnage d'importance	59
5.4 Filtre SIR (bootstrap)	61
5.5 Redistribution	66

5.6	En pratique	69
6	Applications du filtrage particulière	73
6.1	Modèle numérique de terrain	73
6.2	Géolocalisation par mesure d'atténuation	74
6.3	Poursuite par mesures d'angles avec obstacles	78
6.4	Poursuite dans une séquence d'images	81
	Annexes	85
A	Quelques résultats techniques	87
A.1	Ellipse de confiance	87
A.2	Transformation de Box–Muller	89
A.3	Un peu de complexité algorithmique	90
B	Routines matlab	93
	Bibliographie	97
	Index	103

Avant-propos

Ces notes se fondent notamment sur un cours donné en 2005, 2006 et 2007 au Master Sciences et Technologies de l'université du Sud Toulon/Var ainsi que sur plusieurs formations et écoles.

Il ne s'agit pas de faire ici un exposé même succinct des théories des probabilités, de la statistique et de la statistique bayésienne! Pour la théorie des probabilités, on dispose de nombreuses ressources en ligne. On pourra aussi consulter les monographies [29] pour les probabilités, [10, 11] pour les probabilités appliquées, [54] pour la statistique, [52, 55] pour la statistique bayésienne.

On utilise dans ce cours une notation simplifiée : les réalisations d'une variable aléatoire X sont également notées X et la densité de sa loi est notée $p(X)$. Ainsi $p(X)$ désigne la densité de la loi de X , $p(X, Y)$ la densité de la loi conjointe de (X, Y) etc. $p(X)$ sera indifféremment appelée loi de X , densité de X ou densité de la loi de X . 

Cette notation n'est pas mathématiquement rigoureuse mais elle est très intuitive et permet de se concentrer sur les idées sans trop se préoccuper des problèmes techniques de la théorie de la mesure. Elle montre toutefois rapidement ses limites lorsqu'il s'agit de mener une analyse précise.

Une petite suite logicielle “SMC demos” écrite en Matlab qui reproduit les exemples traités dans ce cours est disponible sur le dépôt git :

<https://gitlab.inria.fr/campillo/smc-demos>

Notations

probabilités

$\mathbb{P}(A)$, probabilité de l'évènement A

$\mathbb{E}(X)$, espérance de (la loi de) X

$\text{var}(X) = \mathbb{E}[(X - \mathbb{E}X)^2]$, variance de (la loi de) X

$\text{cov}(X) = \mathbb{E}[(X - \mathbb{E}X)(X - \mathbb{E}X)^*]$, covariance de (la loi de) X

$X \perp Y$, les variables X et Y sont indépendantes

i.i.d. indépendants et identiquement distribués

$\mathcal{B}(\mathbb{R}^n)$, tribu borélienne de \mathbb{R}^n (engendrée par les “intervalles”)

$\mathcal{C}_b(\mathbb{R}^n)$, ensemble des fonctions continues et bornées de \mathbb{R}^n dans \mathbb{R}

lois de probabilité

$\delta_{x_0}(\mathrm{d}x)$, mesure de Dirac au point x_0 , i.e. $\int \varphi(x) \delta_{x_0}(\mathrm{d}x) = \varphi(x_0)$

$N(\mu, \sigma^2)$, loi gaussienne de moyenne μ et de variance σ^2 ($N(\mu, R)$ loi gaussienne de moyenne μ et de covariance R)

matrices/vecteurs

M^* , matrice transposée de M

I , matrice identité

$|x|$, norme euclidienne du vecteur x , i.e. $|x|^2 = x^* x$

$\text{diag}(x_1, \dots, x_N) \in \mathbb{R}^{N \times N}$ est la matrice diagonale formée des termes x_1, \dots, x_N .

fonctions

$\mathbf{1}_A(x)$, fonction indicatrice de l'ensemble A ($= 1$ si $x \in A$, 0 sinon)

$f(x) \propto g(x)$, $f(x)$ est proportionnelle à $g(x)$, i.e. $f(x)$ est égale à $g(x)$ à une constante multiplicative près

l'argument du maximum d'une fonction f est défini par

$$\operatorname{Argmax}_{x \in \mathcal{X}} f(x) \stackrel{\text{déf}}{=} \{x \in \mathcal{X}; f(x) \geq f(x'), \forall x' \in \mathcal{X}\}$$

ainsi $\hat{x} \in \operatorname{Argmax}_{x \in \mathcal{X}} f(x)$ réalise le maximum de f , le plus souvent on supposera que ce maximum est unique et on notera $\hat{x} = \operatorname{Argmax}_{x \in \mathcal{X}} f(x)$.

divers

$A \stackrel{\text{déf}}{=} B$, A est égal à B par définition

$\lfloor x \rfloor$ partie entière de x , i.e. $\inf\{n \in \mathbb{N} : n \leq x\}$

A^c , complémentaire de l'ensemble A

ssi, si et seulement si

$X_{k:\ell} = (X_k, \dots, X_\ell)$ si $k \leq \ell$ et $X_{k:\ell} = \emptyset$ si $k > \ell$

1

QUELQUES RAPPELS DE PROBABILITÉS

Une expérience est dite *aléatoire* lorsque, répétée dans des conditions identiques, son issue n'est pas prévisible. La liste de tous les résultats possibles de cette expérience est supposée connue.

Une *variable aléatoire* X est un nombre (un entier, un réel...) ou une "étiquette" (oui/non, présent/absent/indéterminé, a/t/g/c...) dont la valeur dépend d'une expérience aléatoire. La *loi de probabilité* de la variable X décrit comment les réalisations de cette variable se répartissent selon les différentes valeurs possibles. L'ensemble \mathcal{X} des valeurs que peut prendre la variable aléatoire X est donné. Il est appelé *espace d'état* ou *espace des réalisations*. On distingue deux cas :

Cas fini : \mathcal{X} est un ensemble de cardinal $|\mathcal{X}|$ fini.

Cas continu : \mathcal{X} est un espace "continu" du type $\mathcal{X} = \mathbb{R}$ ou \mathbb{R}^n .

On ne traitera pas ici le cas dénombrable $\mathcal{X} = \mathbb{N}$ qui se déduit, modulo quelques précautions, du cas fini. Dans la cadre fini on supposera que \mathcal{X} est de cardinal "petit". On ne traitera pas le cas fini de cardinal "grand" (par exemple $|\mathcal{X}| = 10^{10}$). Ce dernier cas, que l'on rencontre par exemple en optimisation combinatoire, requière des techniques spécifiques.

Ce chapitre est consacré à quelques rappels de probabilité. Dans un premier temps nous allons introduire la notion de loi conditionnelle. Ceci nous permettra de résoudre le problème suivant : *on dispose d'un couple (X, Y) de variables aléatoires dont la loi jointe est connue, X désigne l'état d'un système non observé et Y l'observation associée, comment estimer X à partir de l'observation de Y ?* Nous introduirons également la notion de processus de Markov.

1.1 Espace d'état fini

Lorsque $|\mathcal{X}| < \infty$, l'ensemble \mathcal{X} peut être apparenté à un alphabet. Ainsi $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$ est l'alphabet associé au jet d'un dé, $\mathcal{X} = \{a, t, g, c\}$ est l'alphabet des nucléotides utilisé pour l'analyse de séquences d'ADN etc. Les éléments de \mathcal{X} servent à étiqueter les états de la chaîne. En général, il n'y a aucun sens à effectuer des opérations algébriques sur les éléments de \mathcal{X} , comme par exemple de calculer un état moyen.

a. Loi de probabilité

La loi d'une variable aléatoire X à valeurs dans \mathcal{X} est une mesure de probabilité sur \mathcal{X} — c'est-à-dire une collection $[\mathbb{P}(X = x)]_{x \in \mathcal{X}}$ de nombres positifs dont la somme est 1 :

$$\mathbb{P}(X = x) \in [0, 1] \quad \text{et} \quad \sum_{x \in \mathcal{X}} \mathbb{P}(X = x) = 1.$$

Naturellement $\mathbb{P}(X = x)$ est la probabilité que $X = x$. Pour tout $A \subset \mathcal{X}$, la probabilité pour que X appartienne à A est :

$$\mathbb{P}(X \in A) = \sum_{x \in A} \mathbb{P}(X = x) = \sum_{x \in \mathcal{X}} \mathbf{1}_A(x) \mathbb{P}(X = x).$$

On peut voir cette loi comme un vecteur ligne, par exemple dans le cas $\mathcal{X} = \{a, t, g, c\}$:

$$[\mathbb{P}(X = a) \quad \mathbb{P}(X = t) \quad \mathbb{P}(X = g) \quad \mathbb{P}(X = c)].$$

b. Loi de probabilité conditionnelle

Soient X et Y deux variables aléatoires à valeurs dans des ensembles finis \mathcal{X} et \mathcal{Y} , on définit la *loi conditionnelle* de X sachant Y par :

$$\mathbb{P}(X = x | Y = y) \stackrel{\text{déf}}{=} \frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(Y = y)} = \frac{\mathbb{P}(X = x, Y = y)}{\sum_{x' \in \mathcal{X}} \mathbb{P}(X = x', Y = y)} \quad (1.1)$$

pour tout $x \in \mathcal{X}$ et $y \in \mathcal{Y}$ tel que $\mathbb{P}(Y = y) > 0$. On vérifie aisément que pour tout y , $x \mapsto \mathbb{P}(X = x | Y = y)$ est une mesure de probabilité sur \mathcal{X} .

Deux variables aléatoires X et Y sont dites *indépendantes* — noté $X \perp\!\!\!\perp Y$ — lorsque

$$\mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x) \mathbb{P}(Y = y), \quad \forall x \in \mathcal{X}, y \in \mathcal{Y}.$$

Si $X \perp\!\!\!\perp Y$, alors la loi conditionnelle de X sachant Y est simplement la loi de X , en effet :

$$\mathbb{P}(X = x | Y = y) = \frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(Y = y)} = \frac{\mathbb{P}(X = x) \mathbb{P}(Y = y)}{\mathbb{P}(Y = y)} = \mathbb{P}(X = x).$$

c. Matrice de probabilité

Une matrice $Q = [Q(x, x')]_{x \in \mathcal{X}, x' \in \mathcal{X}'}$ indexée par les éléments x et x' d'ensembles finis \mathcal{X} et \mathcal{X}' est dite *de probabilités* si elle vérifie :

$$Q(x, x') \in [0, 1] \quad \text{et} \quad \sum_{x \in \mathcal{X}} Q(x, x') = 1, \quad \forall x' \in \mathcal{X}'.$$

i.e. ces éléments sont compris entre 0 et 1, la somme des éléments d'une même ligne est égal à 1. Ces matrices sont également appelées matrice de Markov ou encore matrice de transition. Un exemple de telle matrice est la loi conditionnelle de X sachant X'

$$Q(x, x') = \mathbb{P}(X' = x' | X = x).$$

d. Formule de Bayes

D'après la définition (1.1) de la loi conditionnelle, la loi jointe $\mathbb{P}(X = x, Y = y)$ se décompose en $\mathbb{P}(X = x | Y = y) \mathbb{P}(Y = y)$ ou bien $\mathbb{P}(Y = y | X = x) \mathbb{P}(X = x)$, d'où la formule de Bayes

$$\begin{aligned} \mathbb{P}(X = x | Y = y) &= \frac{\mathbb{P}(Y = y | X = x) \mathbb{P}(X = x)}{\mathbb{P}(Y = y)} \\ &= \frac{\mathbb{P}(Y = y | X = x) \mathbb{P}(X = x)}{\sum_{x' \in \mathcal{X}} \mathbb{P}(Y = y | X = x') \mathbb{P}(X = x')}. \end{aligned}$$

Ainsi la loi a posteriori est proportionnelle au produit de la vraisemblance et de la loi a priori :

$$\mathbb{P}(X = x | Y = y) \propto \mathbb{P}(Y = y | X = x) \mathbb{P}(X = x).$$

e. Chaîne de Markov

On considère un processus $X = (X_k)_{k \geq 0}$ à valeurs dans \mathcal{X} . La loi de ce processus est la famille de probabilités $\mathbb{P}(X_{0:k} = x_{0:k})$ pour tout $k \geq 0$ et $x_0, \dots, x_k \in \mathcal{X}$. D'une façon générale, on peut factoriser cette loi de la façon suivante :

$$\begin{aligned} \mathbb{P}(X_{0:k} = x_{0:k}) &= \mathbb{P}(X_0 = x_0) \times \mathbb{P}(X_1 = x_1 | X_0 = x_0) \times \\ &\quad \times \mathbb{P}(X_2 = x_2 | X_{0:1} = x_{0:1}) \cdots \mathbb{P}(X_k = x_k | X_{0:k-1} = x_{0:k-1}). \end{aligned}$$

Ce processus est dit markovien lorsque

$$\mathbb{P}(X_k = x_k | X_{0:k-1} = x_{0:k-1}) = \mathbb{P}(X_k = x_k | X_{k-1} = x_{k-1})$$

pour tout $k \geq 0$ et $x_0, \dots, x_k \in \mathcal{X}$. Ainsi la loi d'un tel processus possède une structure particulière :

$$\mathbb{P}(X_{0:k} = x_{0:k}) = \mathbb{P}(X_0 = x_0) \prod_{\ell=1}^k \mathbb{P}(X_\ell = x_\ell | X_{\ell-1} = x_{\ell-1}).$$

On définit la loi initial de ce processus

$$\rho(x) \stackrel{\text{déf}}{=} \mathbb{P}(X_0 = x), \quad \forall x \in \mathcal{X}$$

et la probabilité de transition

$$Q_\ell(x, x') \stackrel{\text{déf}}{=} \mathbb{P}(X_\ell = x' | X_{\ell-1} = x), \quad \forall x, x' \in \mathcal{X}.$$

On suppose que la chaîne de Markov $X = (X_k)_{k \geq 0}$ est homogène, c'est-à-dire que Q_ℓ ne dépend pas de ℓ et sera noté.

Exemple 1.1.1 On considère le système dynamique suivant à valeurs dans \mathcal{X}

$$X_k = f_k(X_{k-1}, W_k), \quad k \geq 1$$

de loi initiale $X_0 \sim \rho$, où $(W_k)_{k \geq 1}$ est un bruit blanc indépendant de X_0 , $f_k : \mathcal{X} \times \mathcal{W} \mapsto \mathcal{X}$. Alos $(X_0)_{k \geq 0}$ est une chaîne de Markov. Notons d'abord que par construction $W_k \perp\!\!\!\perp X_{0:k-1}$, alors

$$\begin{aligned} \mathbb{P}(X_k = x_k | X_{0:k-1} = x_{0:k-1}) &= \mathbb{P}(f_k(X_{k-1}, W_k) = x_k | X_{0:k-1} = x_{0:k-1}) \\ &= \mathbb{P}(f_k(x_{k-1}, W_k) = x_k | X_{0:k-1} = x_{0:k-1}) \\ &= \mathbb{P}(f_k(x_{k-1}, W_k) = x_k) \\ &= \mathbb{P}(f_k(x_{k-1}, W_1) = x_k) \end{aligned}$$

ainsi il s'agit bien d'une chaîne de Markov de matrice de transition

$$Q_k(x, x') = \mathbb{P}(f_k(x, W_1) = x')$$

si $f = f_k$ ne dépend pas de k , alors cette chaîne est homogène.

Remarque 1.1.2 Considérons une chaîne de Markov $(X_k)_{k \geq 0}$ de loi initiale ρ et de matrice de transition Q , alors, conditionnellement à l'événement $\{X_\ell = x\}$, la suite $\{X_p\}_{p > \ell}$ est une chaîne de Markov de loi initiale :

$$\mathbb{P}(X_{\ell+1} = x' | X_\ell = x) = Q(x, x'), \quad x' \in \mathcal{X}$$

et de matrice de transition Q .

1.2 Espace d'état continu

a. Loi de probabilité

On considère une variable aléatoire X à valeurs dans \mathbb{R}^n et de densité $p(X)$, c'est-à-dire :

$$\begin{aligned}\mathbb{E}\phi(X) &= \int_{\mathbb{R}^n} \phi(X) p(X) dX, & \forall \phi \in \mathcal{C}_b(\mathbb{R}^n), \\ \mathbb{P}(X \in A) &= \int_A p(X) dX, & \forall A \in \mathcal{J}(\mathbb{R}^n).\end{aligned}$$

où $\mathcal{C}_b(\mathbb{R}^n)$ est l'ensemble des fonctions continues et bornées définies sur \mathbb{R}^n à valeurs dans \mathbb{R} ; $\mathcal{J}(\mathbb{R}^n)$ est l'ensemble des parties de \mathbb{R}^n de la forme $\prod_{i=1}^N [a_i, b_i]$.

La loi de X est la mesure de probabilité définie sur \mathbb{R}^n par $A \mapsto \mathbb{P}(X \in A)$. Cette loi est entièrement caractérisée par la densité $p(X)$, ou par la donnée de $\mathbb{E}\phi(X)$ pour tout $\phi \in \mathcal{C}_b(\mathbb{R}^n)$, ou bien encore par la donnée de $\mathbb{P}(X \in A)$ pour tout $A \in \mathcal{J}(\mathbb{R}^n)$.

On définit la *moyenne* de (la loi de) X

$$\mathbb{E}X = \int_{\mathbb{R}^n} X p(X) dX \in \mathbb{R}^n$$

ainsi que sa matrice de covariance

$$\text{cov}[X] \stackrel{\text{déf}}{=} \mathbb{E}[(X - \mathbb{E}X)(X - \mathbb{E}X)^*] = \mathbb{E}[XX^*] - [\mathbb{E}X][\mathbb{E}X]^* \in \mathbb{R}^{n \times n}.$$

Cette matrice est symétrique définie non-négative (cf. Annexe A.1). Dans le cas scalaire ($n = 1$), on parle de variance $\text{var}(X) = \mathbb{E}[(X - \mathbb{E}X)^2] \geq 0$. On définit également la covariance croisée

$$\text{cov}(X, Y) \stackrel{\text{déf}}{=} \mathbb{E}[(X - \mathbb{E}X)(Y - \mathbb{E}Y)^*] \in \mathbb{R}^{n \times d}.$$

On a $\text{cov}(X, Y) = \text{cov}(Y, X)^*$.

Soit (X, Y) un vecteur aléatoire à valeurs dans $\mathbb{R}^n \times \mathbb{R}^d$ de loi (de densité) jointe $p(X, Y)$, la *loi marginale* de X est définie par :

$$p(X) \stackrel{\text{déf}}{=} \int_{\mathbb{R}^d} p(X, Y) dY.$$

b. Loi conditionnelle

On considère un vecteur aléatoire (X, Y) à valeurs dans $\mathbb{R}^n \times \mathbb{R}^d$ de densité jointe $p(X, Y)$: X désigne l'état d'un système non observé et Y son observation. Le but est d'estimer X à partir de l'observation y de Y .

Un estimateur est une fonction (mesurable) de l'espace d'observation \mathbb{R}^d à valeurs dans l'espace d'état \mathbb{R}^n . On cherche un *estimateur*, c'est-à-dire une fonction (mesurable) $y \mapsto \hat{\psi}(y)$, qui soit

(i) *sans biais* : c'est-à-dire

$$\mathbb{E}[X - \hat{\psi}(Y)] = 0, \quad (1.2)$$

(ii) *de variance minimum* parmi les estimateurs sans biais : c'est-à-dire

$$\mathbb{E}|X - \hat{\psi}(Y)|^2 = \min_{\substack{\psi(\cdot): \mathbb{R}^d \rightarrow \mathbb{R}^n \\ \text{mesurable} \\ \text{sans biais}}} \mathbb{E}|X - \psi(Y)|^2. \quad (1.3)$$

On cherche donc, parmi les estimateurs sans biais, celui qui minimise l'erreur quadratique.

On définit la *densité conditionnelle de X sachant Y*, notée $p(X|Y)$ et définie par :

$$p(X|Y) \stackrel{\text{déf}}{=} \frac{p(X, Y)}{p(Y)} = \frac{p(X, Y)}{\int_{\mathbb{R}^n} p(X, Y) dX} \quad (1.4)$$

On vérifie aisément qu'à Y fixé $X \mapsto p(X|Y)$ est une densité de probabilité.

Il est important de comprendre que $p(X|Y)$ *représente toute l'information sur X contenue dans Y*.

On utilisera également la notation $p(X|Y = y)$ pour désigner la loi conditionnelle de X sachant que Y prend la valeur y, i.e. $p(X|Y = y) = p(X, Y = y) / p(Y = y)$ où $p(X, Y = y)$ (resp. $p(Y = y)$) est la densité de probabilité de (X, Y) (resp. Y) pris en $Y = y$ fixé.

On définit alors l'*espérance conditionnelle* associée à cette densité conditionnelle

$$\mathbb{E}[\varphi(X)|Y] \stackrel{\text{déf}}{=} \int_{\mathbb{R}^n} \varphi(X) p(X|Y) dX = \frac{\int_{\mathbb{R}^n} \varphi(X) p(X, Y) dX}{\int_{\mathbb{R}^n} p(X, Y) dX} \quad (1.5)$$

en particulier l'espérance conditionnelle de X sachant Y est

$$\mathbb{E}[X|Y] \stackrel{\text{déf}}{=} \int_{\mathbb{R}^n} X p(X|Y) dX.$$

Avant de répondre au problème (1.2)–(1.3) on établit quelques propriétés de l'espérance conditionnelle.

Proposition 1.2.1 *L'espérance conditionnelle est un opérateur linéaire et*

$$\mathbb{E}(\varphi(Y) \mathbb{E}[X|Y]) = \mathbb{E}(\varphi(Y) X), \quad \forall \varphi \in \mathcal{C}_b(\mathbb{R}^n). \quad (1.6)$$

En particulier $\mathbb{E}(\mathbb{E}[X|Y]) = \mathbb{E}(X)$.

Preuve On considère un triplet de variables aléatoires (X_1, X_2, Y) de loi jointe $p(X_1, X_2, Y)$ et deux réels α et β , alors

$$\begin{aligned}
 \mathbb{E}[\alpha X_1 + \beta X_2 | Y] &= \int \int (\alpha X_1 + \beta X_2) p(X_1, X_2 | Y) dX_1 dX_2 \\
 &= \int \int (\alpha X_1 + \beta X_2) \frac{p(X_1, X_2, Y)}{p(Y)} dX_1 dX_2 \\
 &= \alpha \int \int X_1 \frac{p(X_1, X_2, Y)}{p(Y)} dX_1 dX_2 + \beta \int \int X_2 \frac{p(X_1, X_2, Y)}{p(Y)} dX_1 dX_2 \\
 &= \alpha \int X_1 \left(\int \frac{p(X_1, X_2, Y)}{p(Y)} dX_2 \right) dX_1 + \beta \int X_2 \left(\int \frac{p(X_1, X_2, Y)}{p(Y)} dX_1 \right) dX_2 \\
 &= \alpha \int X_1 \frac{p(X_1, Y)}{p(Y)} dX_1 + \beta \int X_2 \frac{p(X_2, Y)}{p(Y)} dX_2 \\
 &= \alpha \int X_1 p(X_1 | Y) dX_1 + \beta \int X_2 p(X_2 | Y) dX_2 \\
 &= \alpha \mathbb{E}[X_1 | Y] + \beta \mathbb{E}[X_2 | Y]
 \end{aligned}$$

ce qui montre que l'espérance conditionnelle est un opérateur linéaire. Par ailleurs

$$\begin{aligned}
 \mathbb{E}(\varphi(Y) \mathbb{E}[X | Y]) &= \int \varphi(Y) \mathbb{E}[X | Y] p(Y) dY \\
 &= \int \int \varphi(Y) X p(X | Y) p(Y) dX dY \\
 &= \int \int \varphi(Y) X p(X, Y) dX dY \\
 &= \mathbb{E}(\varphi(Y) X). \quad \square
 \end{aligned}$$

En fait (1.6) est également une définition de l'espérance conditionnelle : $\mathbb{E}[X | Y]$ est l'unique fonction de Y telle que (1.6).

Proposition 1.2.2 Pour toutes fonctions φ_1, φ_2 continues et bornées

$$\mathbb{E}[\varphi_1(X) \varphi_2(Y) | Y] = \mathbb{E}[\varphi_1(X) | Y] \varphi_2(Y)$$

notamment $\mathbb{E}[\varphi_2(Y) | Y] = \varphi_2(Y)$.

Preuve

$$\begin{aligned}
 \mathbb{E}[\varphi_1(X) \varphi_2(Y) | Y] &= \int \varphi_1(X) \varphi_2(Y) p(X | Y) dX \\
 &= \int \varphi_1(X) p(X | Y) dX \varphi_2(Y). \quad \square
 \end{aligned}$$

On montre maintenant que $\mathbb{E}[X | Y]$ est bien la solution du problème (1.2)–(1.3). Le fait que $\mathbb{E}[X | Y]$ soit sans biais est évident : $\mathbb{E}[X - \mathbb{E}[X | Y]] = E[X - X] = 0$. Pour (1.3) on utilise le résultat suivant

Proposition 1.2.3 Pour toute fonction ψ continue et bornée :

$$\mathbb{E}(|X - \psi(Y)|^2) \geq \mathbb{E}(|X - \mathbb{E}[X | Y]|^2). \quad (1.7)$$

Preuve En développant le carré de la norme euclidienne :

$$\begin{aligned}\mathbb{E}[|X - \psi(Y)|^2 | Y] &= \mathbb{E}[|X - \mathbb{E}[X|Y] + \mathbb{E}[X|Y] - \psi(Y)|^2 | Y] \\ &= \mathbb{E}[|X - \mathbb{E}[X|Y]|^2 | Y] + \mathbb{E}[|\mathbb{E}[X|Y] - \psi(Y)|^2 | Y] \\ &\quad + 2\mathbb{E}[(X - \mathbb{E}[X|Y])^* (\mathbb{E}[X|Y] - \psi(Y)) | Y].\end{aligned}$$

Mais, d'après la Proposition 1.2.2

$$\begin{aligned}\mathbb{E}[(X - \mathbb{E}[X|Y])^* (\mathbb{E}[X|Y] - \psi(Y)) | Y] \\ = \mathbb{E}[X - \mathbb{E}[X|Y] | Y]^* (\mathbb{E}[X|Y] - \psi(Y))\end{aligned}$$

et par linéarité $\mathbb{E}[X - \mathbb{E}[X|Y] | Y] = \mathbb{E}[X|Y] - \mathbb{E}[X|Y] = 0$. Ainsi,

$$\begin{aligned}\mathbb{E}[|X - \psi(Y)|^2 | Y] &= \mathbb{E}[|X - \mathbb{E}[X|Y]|^2 | Y] + \mathbb{E}[|\mathbb{E}[X|Y] - \psi(Y)|^2 | Y] \\ &\geq \mathbb{E}[|X - \mathbb{E}[X|Y]|^2 | Y].\end{aligned}$$

En prenant l'espérance dans les deux membres de cette inégalité et en utilisant la Proposition 1.2.1 on obtient (1.7). \square

La densité conditionnelle vérifie les propriétés suivantes (dont la preuve est laissée en exercice) :

$$p(X, Y) = p(X|Y) p(Y), \quad (1.8)$$

$$p(X, Y, Z) = p(X|Y, Z) p(Y|Z) p(Z), \quad (1.9)$$

$$p(X) = \int_{\mathbb{R}^d} p(X|Y) p(Y) dY, \quad (1.10)$$

$$p(X|Z) = \int_{\mathbb{R}^d} p(X|Y, Z) p(Y|Z) dY. \quad (1.11)$$

c. Indépendance

Soit (X, Y) un vecteur aléatoire à valeurs dans $\mathbb{R}^n \times \mathbb{R}^d$ et de loi $p(X, Y)$. Les variables X et Y sont dites *indépendantes*, noté $X \perp\!\!\!\perp Y$, lorsque

$$\mathbb{P}(X \in A, Y \in B) = \mathbb{P}(X \in A) \times \mathbb{P}(Y \in B) \quad (1.12)$$

pour tout $A \in \mathcal{B}(\mathbb{R}^n)$ et $B \in \mathcal{B}(\mathbb{R}^d)$. Une définition équivalente de l'indépendance est

$$\mathbb{E}(\varphi(X) \psi(Y)) = \mathbb{E}(\varphi(X)) \mathbb{E}(\psi(Y)) \quad (1.13)$$

pour tout $\varphi \in \mathcal{C}_b(\mathbb{R}^n)$ et $\psi \in \mathcal{C}_b(\mathbb{R}^d)$.

Concernant les densités, $X \perp\!\!\!\perp Y$ est équivalent à une des propriétés suivantes :

- (i) $p(X, Y) = p(X) p(Y)$,
- (ii) $p(X|Y) = p(X)$,
- (iii) $p(Y|X) = p(Y)$.

Il est clair que la première de ces propriétés est une simple réécriture de (1.12). L'équivalence avec les deux autres propriétés est évidente.

Proposition 1.2.4 (i) Soit $X \perp\!\!\!\perp Y$, alors

$$\mathbb{E}[X|Y] = \mathbb{E}[X].$$

(ii) Soit (X, Y, Z) tel que $Y \perp\!\!\!\perp (X, Z)$, alors que

$$\mathbb{E}[X|Y, Z] = \mathbb{E}[X|Z].$$

La partie (i) signifie simplement que si $X \perp\!\!\!\perp Y$ alors l'observation de Y n'apporte aucune information supplémentaire concernant X . La partie (ii) signifie qu'en conditionnant par un vecteur, on peut retirer du terme de conditionnement les composantes qui n'apportent aucune information (i.e. indépendantes des autres composantes et de X). La preuve de cette proposition est laissée en exercice.

d. Noyau markovien

Définition 1.2.5 Un noyau markovien sur $\mathcal{X} \times \mathbb{R}^d$ est une famille $\psi = \{\psi(x, \cdot); x \in \mathcal{X}\}$ de fonctions $\mathbb{R}^d \ni x' \mapsto \psi(x, x') \in \mathbb{R}$ telles que :

- (i) $\psi(x, x') \geq 0$ pour tout $x \in \mathcal{X}$ et $x' \in \mathbb{R}^d$.
- (ii) $\int_{\mathbb{R}^d} \psi(x, x') dx' = 1$ pour tout $x \in \mathcal{X}$.

Ici \mathcal{X} peut être un ensemble fini ou continu.

On considère un couple de variables aléatoires (X, Y) de densité jointe $p(X, Y)$, il est clair que :

$$\psi(X, Y) = p(X|Y)$$

est un noyau markovien.

e. Formule de Bayes et vraisemblance

La densité jointe $p(X, Y)$ s'écrit donc de deux façons :

$$p(X, Y) = p(X|Y) p(Y) = p(Y|X) p(X).$$

On en déduit la formule de Bayes :

$$p(X|Y) = \frac{p(Y|X) p(X)}{p(Y)} = \frac{p(Y|X) p(X)}{\int p(Y|X) p(X) dX}. \quad (1.14)$$

Cette formule est plus qu'une simple tautologie. Reprenons notre petit problème d'estimation : on dispose d'un couple état/observation (X, Y) . La loi *a priori* de X , i.e. avant observation de Y , est $p(X)$. Après observation de Y on dispose de la loi *a posteriori*, c'est-à-dire la loi $p(X|Y)$ de X sachant Y .

La formule de Bayes (1.14) dit simplement qu'à Y fixé, la loi *a posteriori* $p(X|Y)$ est égale, à une constante multiplicative près, au produit de la loi *a priori* $p(X)$ et de la vraisemblance $p(Y|X)$,

$$\text{loi a posteriori} = \frac{1}{\text{cte de normalisation}} \times \text{vraisemblance} \times \text{loi a priori}$$

la constante de normalisation est simplement $\int p(Y|X) p(X) dX$.

On utilise souvent la notation :

$$p(X|Y) \propto p(Y|X) p(X) \quad (1.15)$$

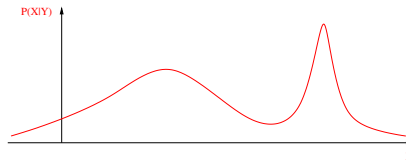
c'est-à-dire

$$\text{loi a posteriori} \propto \text{vraisemblance} \times \text{loi a priori}$$

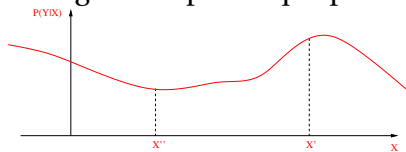
où \propto signifie "proportionnel à" (i.e. comme fonction de X , $p(X|Y)$ est proportionnel à $p(Y|X) p(X)$, la constante de proportionnalité ne dépend que de Y).

Il existe donc deux lectures de la densité conditionnelle :

- à Y fixé, $X \mapsto p(X|Y)$ est une *densité*, i.e. une fonction positive telle que $\int p(X|Y) dX = 1$.



- à Y fixé, $X \mapsto p(Y|X)$ est une *fonction de vraisemblance*, i.e. une fonction positive telle que si $p(Y|X') > p(Y|X'')$ alors l'observation Y a été plus vraisemblablement générée par X' que par X'' .



Il faut maintenant expliquer en quoi (1.15) nous permet d'avancer dans notre problème. On s'intéresse à la loi a posteriori, la formule de Bayes (1.15) nous dit qu'il suffit de connaître la loi a priori et la vraisemblance (oublions la constante de normalisation). Expliquons sur un exemple l'intérêt de la formule.

Exemple 1.2.6 On dispose d'une observation Y partielle et bruitée de X donné par le modèle :

$$Y = h(X) + V$$

où X est de loi a priori $p(X)$ donnée. On suppose que le bruit de mesure V est $\mathcal{N}(0, \sigma^2 I)$ et qu'il est indépendant de X . On suppose également que la fonction d'observation h est connue. Il est clair que la vraisemblance est alors :

$$p(Y|X) \propto \exp\left(-\frac{1}{2\sigma^2}|Y - h(X)|^2\right).$$

Ainsi d'après la formule de Bayes (1.15)

$$p(X|Y) \propto \exp\left(-\frac{1}{2\sigma^2}|Y - h(X)|^2\right) \times p(X).$$

En particulier si $p(X)$ est $\mathcal{N}(\bar{X}, Q)$ alors

$$p(X|Y) \propto \exp\left(-\frac{1}{2\sigma^2}|Y - h(X)|^2\right) \times \exp\left(-\frac{1}{2}(X - \bar{X})^* Q^{-1} (X - \bar{X})\right).$$

Si $h(x)$ n'est pas linéaire, cette loi n'est pas gaussienne.

Ce qu'il faut comprendre, c'est qu'en pratique il est naturel de se donner la loi a priori $p(X)$ et la vraisemblance $p(Y|X)$, la formule de Bayes (1.15) permet alors de calculer la loi a posteriori.

f. Loi gaussienne

Une variable aléatoire à valeurs dans \mathbb{R}^n suit une loi gaussienne centrée et réduite, notée $X \sim N(0, I)$ (où I est la matrice identité), lorsque :

$$p(X) = \frac{1}{\sqrt{(2\pi)^n}} \exp\left(-\frac{1}{2}|X|^2\right).$$

Sa moyenne est $\mathbb{E}X = 0$ et sa covariance $\text{cov}(X) \stackrel{\text{déf}}{=} \mathbb{E}[(X - \mathbb{E}X)(X - \mathbb{E}X)^*] = I$.

Un vecteur aléatoire gaussien X à valeurs dans \mathbb{R}^n est, par définition, la transformée linéaire d'un vecteur aléatoire gaussien centré et réduit, i.e. $X = \mu + C\tilde{X}$ où $\tilde{X} \sim N(0, I)$. La moyenne de X est alors $\mathbb{E}(X) = \mu$, sa covariance est $\text{cov}(X) = CC^*$. On note $X \sim N(\mu, \Gamma)$ où $\Gamma = CC^*$.

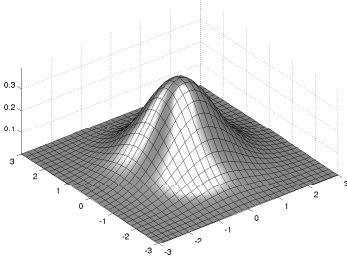


FIGURE 1.1 – *Densité de la normale centrée et réduite en dimension 2.*

Étant donné un vecteur aléatoire gaussien $X \sim N(\mu, \Gamma)$, si la matrice Γ est inversible alors sa densité est

$$p(X) = \frac{1}{\sqrt{(2\pi)^n \det(\Gamma)}} \exp\left(-\frac{1}{2}(X - \mu)^* \Gamma^{-1} (X - \mu)\right).$$

Lorsque Γ n'est pas inversible, la loi de X n'admet pas de densité au sens classique.

De façon générale la loi d'une variable X est gaussienne si et seulement si la fonction caractéristique de cette loi est donnée

$$\mathbb{E} e^{i r^* X} = \exp\left(i r^* \mu - \frac{1}{2} r^* \Gamma r\right), \quad \forall r \in \mathbb{R}^n$$

où $i^2 = -1$.

Revenons au problème initial, on considère un vecteur aléatoire gaussien (X, Y) , la loi conditionnelle de X sachant Y est également gaussienne :

Proposition 1.2.7 Soit $Z = \begin{pmatrix} X \\ Y \end{pmatrix}$ un vecteur aléatoire gaussien à valeurs dans \mathbb{R}^{n+d} de moyenne \bar{Z} et de covariance Q_Z avec :

$$\bar{Z} = \begin{pmatrix} \bar{X} \\ \bar{Y} \end{pmatrix}, \quad Q_Z = \begin{pmatrix} Q_X & Q_{XY} \\ Q_{XY}^* & Q_Y \end{pmatrix},$$

où $Q_Y > 0$ alors $p(X|Y)$ est gaussien $N(\hat{X}(Y), R)$ avec :

$$\hat{X}(Y) = \bar{X} + Q_{XY} Q_Y^{-1} (Y - \bar{Y}), \quad R = Q_X - Q_{XY} Q_Y^{-1} Q_{XY}^*.$$

Preuve Supposons $Q_Z > 0$, alors

$$p(X|Y) = \frac{p(X, Y)}{p(Y)} = \frac{\frac{1}{\sqrt{(2\pi)^{n+d} \det Q_Z}} \exp\left\{-\frac{1}{2}(Z - \bar{Z})^* Q_Z^{-1} (Z - \bar{Z})\right\}}{\frac{1}{\sqrt{(2\pi)^d \det Q_Y}} \exp\left\{-\frac{1}{2}(Y - \bar{Y})^* Q_Y^{-1} (Y - \bar{Y})\right\}}$$

on a

$$\begin{pmatrix} I & -Q_{XY} Q_Y^{-1} \\ 0 & I \end{pmatrix} Q_Z \begin{pmatrix} I & 0 \\ -Q_Y^{-1} Q_{XY}^* & I \end{pmatrix} = \begin{pmatrix} Q_X - Q_{XY} Q_Y^{-1} Q_{XY}^* & 0 \\ 0 & Q_Y \end{pmatrix}$$

donc $1 \times \det Q_Z \times 1 = \det R \times \det Q_Y$ et aussi

$$\begin{pmatrix} I & 0 \\ -Q_Y^{-1} Q_{XY}^* & I \end{pmatrix}^{-1} Q_Z^{-1} \begin{pmatrix} I & -Q_{XY} Q_Y^{-1} \\ 0 & I \end{pmatrix}^{-1} = \begin{pmatrix} R^{-1} & 0 \\ 0 & Q_Y^{-1} \end{pmatrix}$$

i.e. $Q_Z^{-1} = \begin{pmatrix} I & 0 \\ -Q_Y^{-1} Q_{XY}^* & I \end{pmatrix} \begin{pmatrix} R^{-1} & 0 \\ 0 & Q_Y^{-1} \end{pmatrix} \begin{pmatrix} I & -Q_{XY} Q_Y^{-1} \\ 0 & I \end{pmatrix}$

Donc

$$(Z - \bar{Z})^* Q_Z^{-1} (Z - \bar{Z}) = (X - \hat{X}(Y))^* R^{-1} (X - \hat{X}(Y)) + (Y - \bar{Y})^* Q_Y^{-1} (Y - \bar{Y})$$

et

$$p(X | Y) = \frac{1}{\sqrt{(2\pi)^n \det R}} \exp \left\{ -\frac{1}{2} (X - \hat{X}(Y))^* R^{-1} (X - \hat{X}(Y)) \right\}.$$

Si Q_Z n'est pas inversible il est nécessaire de faire appel aux fonctions caractéristiques. \square

Soit (X, Y) un vecteur aléatoire gaussien, on peut vérifier à titre d'exercice que $X \perp\!\!\!\perp Y$ ssi $\text{cov}(X, Y) = Q_{XY} = 0$. Dans le cas gaussien l'indépendance est donc équivalente à la non-corrélation, ce qui est faux dans le cas général.

Les composantes d'un vecteur aléatoire gaussien $X \sim N(\mu, R)$ sont gaussiennes, i.e. les lois marginales d'une gaussienne sont gaussiennes, et elles sont indépendantes ssi la matrice R est diagonale.

Attention, si X et Y sont des variables aléatoires gaussiennes alors (X, Y) n'est pas nécessairement un vecteur aléatoire gaussien. Si X et Y sont des variables aléatoires gaussiennes indépendantes alors (X, Y) est un vecteur aléatoire gaussien. En revanche, si X et Y sont des variables aléatoires gaussiennes non corrélées alors (X, Y) n'est pas nécessairement un vecteur aléatoire gaussien.

Considérons un cas bi-dimensionnel : soient X_1 et Z indépendants et gaussiens. On pose $X_2 = \text{signe}(X_1) \times |Z|$, alors X_1 et X_2 sont gaussiens mais le vecteur (X_1, X_2) ne l'est pas. Pour cela il suffit de considérer $X_1 + X_2$ qui n'est pas gaussien.

Revenons à la Proposition 1.2.7, dans tous les cas $R \leq Q_X$ (cf. Annexe A.1 pour une définition de l'inégalité entre matrices symétriques définies positives). Si X et Y ne sont pas indépendants, i.e. $Q_{XY} \neq 0$ alors $R < Q_X$. Dans ce cas, le fait d'observer Y permet de réellement affiner la connaissance que l'on a sur X , en pratique l'ellipse de confiance sur X est strictement plus petite après observation de Y (cf. Annexe A.1).

On note que, dans la Proposition 1.2.7, la covariance conditionnelle R de X sachant Y ne dépend pas directement de l'observation Y .

g. Loi multinomiale

Dans ce cours on ne sera amené à considérer des variables X prenant un nombre N fini de valeurs ξ^i de \mathbb{R}^n , $1 \leq i \leq N$, avec $\mathbb{P}(X = \xi^i) = \omega^i$. La "densité"

de X est alors ¹

$$p(X) = \sum_{i=1}^N \omega^i \delta_{\xi^i}(X) \quad (1.16)$$

avec $\xi^i \in \mathbb{R}^n$, $\omega^i \geq 0$, $\sum_{i=1}^N \omega^i = 1$ où $\delta_{\xi^i}(X)$ désigne la mesure de Dirac en ξ^i . Ainsi, pour tout $A \subset \mathbb{R}^n$ et toute fonction φ :

$$\mathbb{E}\varphi(X) = \sum_{i=1}^N \omega^i \varphi(\xi^i), \quad \mathbb{P}(X \in A) = \sum_{i=1}^N \omega^i \mathbf{1}_A(\xi^i).$$

Supposons que l'on tire r "particules", c'est-à-dire les points ξ^i , selon la loi (1.16) : on obtient r_1 fois la particule ξ^1 , r_2 fois la particule ξ^2 , ainsi de suite. Ainsi le vecteur $r = r_{1:N}$ suit une loi appelée *multinomiale*, notée $\mathcal{M}(r, \omega^1, \dots, \omega^N)$, telle que

$$\mathbb{P}(r_{1:N} = r_{1:N}) = \begin{cases} r! \prod_{i=1}^N \frac{[\omega^i]^{r_i}}{r_i!} & \text{si } r_1 + \dots + r_N = r, \\ 0 & \text{sinon.} \end{cases}$$

C'est la loi du tirage de r boules avec remise dans une urne contenant N boules numérotées. La proportion de boules i est ω^i . Le nombre de boules i tirées est r_i .

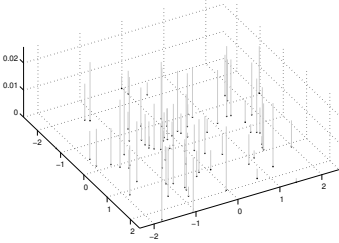


FIGURE 1.2 – Représentation d'une loi discrète $p(X) = \sum_{i=1}^N \omega^i \delta_{\xi^i}(X)$.

h. Chaînes de Markov

Un *processus stochastique*, ou aléatoire, $(X_k)_{k \geq 0}$ à valeurs dans \mathbb{R}^n est une suite de variables aléatoires à valeurs dans \mathbb{R}^n . On utilisera la notation

$$X_{0:k} = [X_0, \dots, X_k].$$

La loi du processus $(X_k)_{k \geq 0}$ est la donnée des lois $p(X_{0:k})$ pour tout k .

Par définition de la loi conditionnelle (cf. Équations (1.8) et (1.9)) :

$$p(X_{0:k}) = p(X_0) p(X_1|X_0) p(X_2|X_{0:1}) \cdots p(X_k|X_{0:k-1}) \quad (1.17)$$

Deux processus $(X_k)_{k \geq 0}$ et $(Y_k)_{k \geq 0}$ sont dits indépendants si $X_{0:k} \perp\!\!\!\perp Y_{0:k}$ pour tout k .

1. C'est un abus de notation, la mesure de Dirac δ_ξ est une mesure de probabilité qui n'admet pas de densité. Par définition $\int_A \delta_\xi(X) dX = 1$ si $\xi \in A$, 0 sinon; de manière équivalente $\int \varphi(X) \delta_\xi(X) dX = \varphi(\xi)$.

Un processus $(X_k)_{k \geq 0}$ à valeurs dans \mathbb{R}^n est appelé *chaîne de Markov* (ou processus de Markov) s'il satisfait à la propriété de Markov suivante :

$$p(X_{k+1}|X_{0:k}) = p(X_{k+1}|X_k) \quad \forall k \geq 0 \quad (1.18)$$

qui peut se lire de la façon suivante : « *le futur conditionné par le passé est le futur conditionné par le présent.* »

La propriété (1.18) utilisée dans (1.17) conduit à :

$$p(X_{0:k}) = p(X_0) \prod_{\ell=1}^k p(X_\ell|X_{\ell-1}).$$

On en déduit que la loi d'un processus de Markov est entièrement caractérisée par sa *loi initiale* $p(X_0)$ et sa *loi de transition* $p(X_\ell|X_{\ell-1})$: 

$$p(X_0) \rightarrow \cdots \rightarrow p(X_{k-1}) \xrightarrow[\text{Chapman-Kolmogorov}]{p(X_k|X_{k-1})} p(X_k)$$

La loi de transition détermine l'évolution de la loi de la variable X_k . En effet, la densité $p(X_k)$ se calcule à partir de $p(X_{k-1})$ à l'aide l'équation de Chapman-Kolmogorov

$$p(X_k) = \int_{\mathbb{R}^n} p(X_k|X_{k-1}) p(X_{k-1}) dX_{k-1} \quad (1.19)$$

qui est exactement la propriété (1.10). On peut itérer cette équation pour calculer $p(X_k)$ à partir de $p(X_{k-\ell})$, on obtient

$$p(X_k) = \int_{\mathbb{R}^n} dX_{k-1} p(X_k|X_{k-1}) \cdots \int_{\mathbb{R}^n} dX_{k-\ell} p(X_{k-\ell+1}|X_{k-\ell}) p(X_{k-\ell}).$$

Exemple 1.2.8 (bruit blanc) Un processus $(X_k)_{k \geq 0}$ sera appelé *bruit blanc* s'il s'agit d'une suite de variables aléatoires indépendantes de moyenne nulle, i.e., pour tout k , les variables X_0, X_1, \dots, X_k sont indépendantes et $\mathbb{E}X_k = 0$. Il s'agit bien d'un cas (très) particulier de processus de Markov puisque par indépendance $p(X_k|X_{0:k-1}) = p(X_k)$. Si de plus ce processus est gaussien, on parlera de *bruit blanc gaussien*.

Exemple 1.2.9 On considère l'équation récursive

$$X_{k+1} = f_k(X_k, W_k)$$

où X_0 et $(W_k)_{k \geq 0}$ sont indépendants, $(W_k)_{k \geq 0}$ est un bruit blanc $W_k \sim q_k(w) dw$.

Pour tout k , $X_{0:k-1}$ est une fonction de X_0 et $W_{0:k-2}$, il est ainsi indépendant de W_{k-1} . Comme $X_k = f_{k-1}(X_{k-1}, W_{k-1})$, donc $p(X_k | X_{0:k-1}) = p(X_k | X_{k-1})$. Il s'agit donc bien d'un processus de Markov. La loi de transition se calcule de la façon suivante :

$$\mathbb{E}[\varphi(X_k) | X_{k-1}] = \mathbb{E}[\varphi(f_{k-1}(X_{k-1}, W_{k-1})) | X_{k-1}]$$

mais comme $X_{k-1} \perp\!\!\!\perp W_{k-1}$ on a $p(W_{k-1} | X_{k-1}) = p(W_{k-1})$ donc :

$$\begin{aligned} \mathbb{E}[\varphi(f_{k-1}(X_{k-1}, W_{k-1})) | X_{k-1}] &= \int \varphi(f_{k-1}(X_{k-1}, W_{k-1})) p(W_{k-1} | X_{k-1}) dW_{k-1} \\ &= \int \varphi(f_{k-1}(X_{k-1}, W_{k-1})) p(W_{k-1}) dW_{k-1} \\ &= \int \varphi(f_{k-1}(X_{k-1}, w)) q_{k-1}(w) dw \end{aligned}$$

et

$$\begin{aligned} \mathbb{E}[\varphi(X_k) | X_{k-1}] &= \int \varphi(f_{k-1}(X_{k-1}, w)) q_{k-1}(w) dw \\ &= \int \int \varphi(\xi) \delta_{f_{k-1}(X_{k-1}, w)}(\xi) d\xi q_{k-1}(w) dw \\ &= \int \varphi(\xi) \int \delta_{f_{k-1}(X_{k-1}, w)}(\xi) q_{k-1}(w) dw d\xi \\ &= \int \varphi(X_k) \int \delta_{f_{k-1}(X_{k-1}, w)}(X_k) q_{k-1}(w) dw dX_k \end{aligned}$$

donc

$$p(X_k | X_{k-1}) = \int \delta_{f_{k-1}(X_{k-1}, w)}(X_k) q_{k-1}(w) dw.$$

Dans le cas d'un bruit additif, $f_{k-1}(x, w) = f_{k-1}(x) + w$, on obtient une formulation explicite de la loi de transition

$$p(X_k | X_{k-1}) = q_{k-1}(X_k - f_{k-1}(X_{k-1})).$$

Si de plus $W_k \sim N(0, \sigma_W^2 I)$ on a

$$p(X_k | X_{k-1}) \propto \exp\left(-\frac{1}{2\sigma_W^2} |X_k - f_{k-1}(X_{k-1})|^2\right).$$

2

MODÈLES DE MARKOV CACHÉS À ESPACE D'ÉTAT FINI

On s'intéresse ici à des variables aléatoires et de processus à valeurs dans un ensemble *fini* \mathcal{X} .

On considère une chaîne de Markov $X = (X_k)_{k \in \mathbb{N}}$ à espace d'états \mathcal{X} fini de loi initiale ρ et de matrice de transition $Q = [Q(x, x')]_{x, x' \in \mathcal{X}}$. La chaîne X n'est pas directement observable, elle est dite *cachée* ou *latente*. On dispose d'observations $Y = \{Y_k\}_{k \in \mathbb{N}}$ à valeurs dans un espace d'observation \mathcal{Y} . Cet espace peut être au plus dénombrable ou continu.

On suppose que les observations sont recueillies au travers d'un canal *sans mémoire*, c'est-à-dire que, conditionnellement aux états X_k , les observations Y_k sont mutuellement indépendantes et que chaque observation ne dépend que de l'état X_k . On notera $\psi(x, y)$ la vraisemblance associée.

Dans le cas dénombrable comme dans le cas continu, les trois problèmes canoniques des modèles de Markov cachés sont :

- P1** Étant donnés les paramètres $\mathbf{M} = (\rho, Q, \psi)$ du modèle, calculer la vraisemblance d'une suite donnée d'observations $Y_{0:k}$. Ce point est résolu par l'algorithme "forward" de Baum.
- P2** Étant donnés les paramètres $\mathbf{M} = (\rho, Q, \psi)$ du modèle et une suite d'observations $Y_{0:k}$, déterminer la suite d'états $X_{0:k}$ qui est le plus vraisemblablement à l'origine ces observations $Y_{0:k}$. On peut considérer trois cas :
 - (i) l'estimation de l'état courant X_k , c'est le problème de *filtrage* résolu par l'algorithme "forward" de Baum;
 - (ii) l'estimation d'un état intermédiaire X_ℓ (pour $\ell < k$), c'est le problème de *lissage* résolu par les algorithmes "forward" et "backward" de Baum;

(iii) l'estimation de la trajectoire complète $X_{0:k}$, ce point est résolu par l'algorithme de Viterbi.

P3 Étant donnée une suite d'observations $Y_{0:k}$, déterminer les paramètres $\mathbf{M} = (\rho, Q, \psi)$ du modèle dont sont le plus vraisemblablement issus ces observations. Ce point est résolu par l'algorithme de Baum-Welch.

Les bases des modèles de Markov cachés ont été publiées dans une série d'articles à la fin des années 1960 et au début des années 1970, autour de Leonard E. Baum [7], [6], [9], [8], et [5].

Ces notes, comme beaucoup d'autres travaux sur les modèles de Markov cachés, s'appuient sur la présentation faite par [50, Chap. 6]. Ce dernier ouvrage concerne le traitement de la parole qui a été l'un des premiers domaines d'application des modèles de Markov cachés.

Il existe également des articles de revue [49, 48], ainsi que des monographies [50, 19, 39, 27]. Pour l'application de ces modèles en génétique, on pourra consulter [18].

Les formules de re-estimation de Baum-Welsh sont un exemple d'algorithme EM (*Expectation-Maximization*). Cet algorithme a été formulé de façon plus ou moins canonique par [16]. Une présentation générale, comprenant des variantes, se trouve dans [43].

2.1 MMC à espace d'observation au plus dénombrable

Nous supposons ici que \mathcal{Y} est un espace au plus dénombrable.

Hypothèse 2.1.1 (Canal d'observation sans mémoire) *Pour tout $k \geq 1$, $x_0, \dots, x_k \in \mathcal{X}$, $y_0, \dots, y_k \in \mathcal{Y}$:*

$$\mathbb{P}(Y_{0:k} = y_{0:k} | X_{0:k} = x_{0:k}) = \prod_{\ell=0}^k \mathbb{P}(Y_{\ell} = y_{\ell} | Y_{\ell} = x_{\ell})$$

Cette hypothèse caractérise la façon dont la loi des observations dépend de celle du processus d'état. Précisons maintenant cette dépendance.

On considère la matrice de Markov :

$$\psi_{\ell}(x, y) \stackrel{\text{déf}}{=} \mathbb{P}(Y_k = y | X_k = x), \quad x \in \mathcal{X}, \quad y \in \mathcal{Y}.$$

On se place dans le cadre homogène où cette matrice ne dépend pas de ℓ et sera notée ψ . La matrice ψ est aussi appelée *matrice d'émission* ou *loi d'émission*.

Proposition 2.1.2 *La loi jointe du couple (X, Y) est entièrement caractérisée par la loi initiale ρ et la matrice de transition Q de la chaîne de Markov X , ainsi que par la matrice d'émission ψ , en effet :*

$$\mathbb{P}(X_{0:k} = x_{0:k}, Y_{0:k} = y_{0:k}) = \rho(x_0) \psi(x_0, y_0) \prod_{\ell=1}^k [Q(x_{\ell-1}, x_\ell) \psi(x_\ell, y_\ell)] \quad (2.1)$$

pour tout $k \in \mathbb{N}$, $x_0, \dots, x_k \in \mathcal{X}$, $y_0, \dots, y_k \in \mathcal{Y}$. En particulier le couple (X, Y) est markovien¹.

Un modèle de Markov caché (X, Y) à espace d'observation dénombrable est donc entièrement caractérisé par :

(i) La loi initiale ρ du processus d'état X :

$$\rho(x) \stackrel{\text{déf}}{=} \mathbb{P}(X_0 = x), \quad x \in \mathcal{X}.$$

(ii) La matrice de transition Q du processus d'état X :

$$Q(x, x') \stackrel{\text{déf}}{=} \mathbb{P}(X_{k+1} = x' | X_k = x), \quad x, x' \in \mathcal{X}.$$

(iii) La matrice d'émission ψ du processus X sachant le processus Y :

$$\psi(x, y) \stackrel{\text{déf}}{=} \mathbb{P}(Y_k = y | X_k = x), \quad x \in \mathcal{X}, \quad y \in \mathcal{Y}.$$

On désigne par $\mathbf{M} = (\rho, Q, \psi)$ les paramètres du modèle de Markov caché.

a. Algorithmes “forward”/“backward” de Baum

On considère un modèle de Markov caché $\mathbf{M} = (\rho, Q, \psi)$ à espace d'état fini \mathcal{X} et à espace d'observation dénombrable ou continu (dans ce dernier cas on supposera que $\mathcal{Y} = \mathbb{R}^d$).

Il s'agit de calculer *efficacement* la distribution de probabilité de la suite d'observations $Y_{0:k}$ en fonction des paramètres du modèle \mathbf{M} , c'est-à-dire la *fonction de vraisemblance*.

D'après (2.1), la distribution de probabilité des observations est donnée par

$$\begin{aligned} \mathbb{P}(Y_{0:k} = y_{0:k}) &= \sum_{x_{0:k} \in \mathcal{X}^{k+1}} \mathbb{P}(X_{0:k} = x_{0:k}, Y_{0:k} = y_{0:k}) \\ &= \sum_{x_{0:k} \in \mathcal{X}^{k+1}} \rho(x_0) \psi(x_0, y_0) \prod_{\ell=1}^k [Q(x_{\ell-1}, x_\ell) \psi(x_\ell, y_\ell)] \end{aligned} \quad (2.2)$$

pour tout $k \in \mathbb{N}$ et $y_0, \dots, y_k \in \mathcal{Y}$. Comme nous le verrons à la Remarque 2.1.4, cette formulation de la loi de $Y_{0:k}$ est trop coûteuse.

1. Le couple (X, Y) ainsi que X sont markoviens mais Y ne l'est pas.

Algorithme “forward” de Baum

On définit la loi jointe de $Y_{0:k}$ et X_k :

$$\alpha_x^k(y_{0:k}) \stackrel{\text{déf}}{=} \mathbb{P}(Y_{0:k} = y_{0:k}, X_k = x) \quad (2.3)$$

pour $k \in \mathbb{N}$, $y_0, \dots, y_k \in \mathcal{Y}$, $x \in \mathcal{X}$. Ce terme est parfois appelée *variable “forward”*. On utilisera également la notation simplifiée :

$$\alpha_x^k = \alpha_x^k(y_{0:k})$$

lorsqu'il n'y a pas d'ambiguïté sur les observations $y_{0:k}$. Enfin on peut présenter ces termes sous forme du vecteur ligne suivant :

$$\alpha^k = \alpha^k(y_{0:k}) \stackrel{\text{déf}}{=} [\alpha_x^k(y_{0:k})]_{x \in \mathcal{X}} \quad (\text{vecteur ligne}).$$

Théorème 2.1.3 (Algorithme “forward” de Baum) *La suite $\{\alpha^k\}_{k \in \mathbb{N}}$ des variables forward est solution de l'équation récurrente suivante :*

$$\alpha_x^{k+1} \leftarrow \psi(x, y_{k+1}) \sum_{x' \in \mathcal{X}} Q(x', x) \alpha_{x'}^k, \quad (2.4a)$$

$$\alpha_x^0 \leftarrow \rho(x) \psi(x, y_0) \quad (2.4b)$$

pour tout $k \geq 0$, $x \in \mathcal{X}$, $y_{0:k} \in \mathcal{Y}^{k+1}$.

Preuve Notons d'abord que

$$\alpha_x^0(y) = \mathbb{P}(X_0 = x, Y_0 = y) = \mathbb{P}(Y_0 = y | X_0 = x) \mathbb{P}(X_0 = x) = \psi(x, y) \rho(x).$$

Par ailleurs,

$$\begin{aligned} \alpha_{x_k}^k(y_{0:k}) &= \mathbb{P}(Y_{0:k} = y_{0:k}, X_k = x_k) \\ &= \sum_{x_{0:k-1}} \mathbb{P}(Y_{0:k} = y_{0:k}, X_{0:k-1} = x_{0:k-1}, X_k = x_k) \\ &= \sum_{x_{0:k-1}} \rho(x_0) Q(x_0, x_1) \cdots Q(x_{k-1}, x_k) \psi(x_0, y_0) \dots \psi(x_k, y_k). \end{aligned}$$

et donc

$$\begin{aligned} \alpha_{x_{k+1}}^{k+1}(y_{0:k+1}) &= \sum_{x_{0:k}} \rho(x_0) Q(x_0, x_1) \cdots Q(x_k, x_{k+1}) \psi(x_0, y_0) \dots \psi(x_{k+1}, y_{k+1}) \\ &= \psi(x_{k+1}, y_{k+1}) \sum_{x_k} Q(x_k, x_{k+1}) \sum_{x_{0:k-1}} \rho(x_0) Q(x_0, x_1) \cdots Q(x_{k-1}, x_k) \times \\ &\quad \times \psi(x_0, y_0) \dots \psi(x_k, y_k) \\ &= \psi(x_{k+1}, y_{k+1}) \sum_{x_k} Q(x_k, x_{k+1}) \alpha_{x_k}^k(y_{0:k}). \end{aligned}$$

□

Les équations (2.4) s'écrivent également sous forme vectorielle. Il suffit d'introduire la matrice diagonale

$$\Psi(y) \stackrel{\text{déf}}{=} \text{diag}[\psi(x, y); x \in \mathcal{X}],$$

on obtient :

$$\begin{aligned} \alpha^{k+1} &\leftarrow \alpha_k Q \Psi(y_{k+1}), \quad k \geq 0, \\ \alpha^0 &\leftarrow \rho \Psi(y_0). \end{aligned}$$

On peut alors calculer la loi des observations $Y_{0:k}$ de la façon suivante :

$$\mathbb{P}(Y_{0:k} = y_{0:k}) = \sum_{x \in \mathcal{X}} \alpha_x^k(y_{0:k}) \quad (2.5)$$

pour tout $k \in \mathbb{N}$ et $y_0, \dots, y_k \in \mathcal{Y}$.

Remarque 2.1.4 La formulation (2.2) de la loi des observations $Y_{0:k}$ est trop coûteuse. Pour chacune des $|\mathcal{X}|^{k+1} = N^{k+1}$ trajectoires $x_{0:k}$ possibles de l'état, il faut en effet effectuer $2k+1$ produits. La complexité de cet algorithme est donc en $O(kN^k)$. Elle est exponentielle en fonction du nombre k d'observations et donc impossible à utiliser en pratique.

Chaque itération en temps de l'algorithme (2.4) fait appel à $N(2N+1)$ opérations élémentaires. Ainsi le calcul (2.7) de la loi des observations $Y_{0:k}$ est en $O(kN^2)$, il est linéaire en fonction du nombre d'observations.

La formulation (2.5) répond donc à la question du problème **P1**, en effet la fonction de vraisemblance associée au modèle **M** est :

$$\mathcal{L}_k(\mathbf{M}) \stackrel{\text{déf}}{=} \mathbb{P}(Y_{0:k} = y_{0:k}) = \sum_{x \in \mathcal{X}} \alpha_x^k(y_{0:k}). \quad (2.6)$$

Filtrage : problème **P2**-(i)

D'après (2.3), la distribution de probabilité conditionnelle de l'état X_k sachant les observations $Y_{0:k}$ est donnée par :

$$\mathbb{P}(X_k = x | Y_{0:k} = y_{0:k}) = \frac{\alpha_x^k(y_{0:k})}{\sum_{x' \in \mathcal{X}} \alpha_{x'}^k(y_{0:k})} \quad (2.7)$$

pour tout $k \in \mathbb{N}$ et $x \in \mathcal{X}$. Ainsi, p_k apparaît comme une distribution de probabilité conditionnelle *non normalisée*.

Algorithme “backward” de Baum

Soit $k \geq 1$ un instant fixé. Pour tout instant $\ell < k$, on introduit la loi conditionnelle des observations $Y_{\ell+1:k}$ sachant que $X_\ell = x$:

$$\beta_x^\ell(y_{\ell+1:k}) \stackrel{\text{déf}}{=} \mathbb{P}(Y_{\ell+1:k} = y_{\ell+1:k} | X_\ell = x) \quad (2.8)$$

pour $x \in \mathcal{X}$, $y_{\ell+1}, \dots, y_k \in \mathcal{Y}$. Ces termes sont également appelés *variables “backward”*. On utilisera également la notation simplifiée :

$$\beta_x^\ell = \beta_x^\ell(y_{\ell+1:k})$$

lorsque les observations sont connues sans ambiguïté. Enfin, on utilise également la représentation sous forme de vecteur colonne :

$$\beta^\ell = \beta^\ell(y_{\ell+1:k}) = [\beta_x^\ell(y_{\ell+1:k})]_{x \in \mathcal{X}} \quad (\text{vecteur colonne}).$$

Il faut noter que β^ℓ dépend de k .

Théorème 2.1.5 (Algorithme “backward” de Baum) *Pour tout $k > 0$, la suite $\beta^{1:k}$ des variables “backward” est solution de l'équation récurrente rétrograde suivante :*

$$\beta_x^\ell \leftarrow \sum_{x' \in \mathcal{X}} Q(x, x') \psi(x', y_{\ell+1}) \beta_{x'}^{\ell+1}, \quad (2.9a)$$

$$\beta_x^k \leftarrow 1 \quad (2.9b)$$

pour tout $0 \leq \ell < k$, $x \in \mathcal{X}$, $y_{0:k} \in \mathcal{Y}^{k+1}$.

Preuve Notons d'abord que pour $\ell = k - 1$, on a :

$$\begin{aligned} \beta_x^{k-1}(y) &= \mathbb{P}(Y_k = y | X_{k-1} = x) \\ &= \sum_{x' \in \mathcal{X}} \mathbb{P}(Y_k = y, X_k = x' | X_{k-1} = x) \\ &= \sum_{x' \in \mathcal{X}} \mathbb{P}(Y_k = y | X_k = x') \mathbb{P}(X_k = x' | X_{k-1} = x) \\ &= \sum_{x' \in \mathcal{X}} \psi(x', y) Q(x, x') \end{aligned}$$

Ainsi en posant (2.9b), on obtient (2.9a) avec $\ell = k - 1$. D'autre part, il résulte de la Remarque 1.1.2 et de la Proposition 2.1.2 que :

$$\begin{aligned} \beta_{x_\ell}^\ell(y_{\ell+1:k}) &= \mathbb{P}(Y_{\ell+1:k} = y_{\ell+1:k} | X_\ell = x_\ell) \\ &= \sum_{x_{\ell+1:k}} \mathbb{P}(Y_{\ell+1:k} = y_{\ell+1:k}, X_{\ell+1:k} = x_{\ell+1:k} | X_\ell = x_\ell) \\ &= \sum_{x_{\ell+1:k}} Q(x_\ell, x_{\ell+1}) \cdots Q(x_{\ell-1}, x_\ell) \psi(x_{\ell+1}, y_{\ell+1}) \cdots \psi(x_k, y_k) \end{aligned}$$

pour tout $x_\ell \in \mathcal{X}$, $y_{\ell+1}, \dots, y_k \in \mathcal{Y}$.

Ainsi

$$\begin{aligned} \beta_{x_\ell}^\ell(y_{\ell+1:k}) &= \sum_{x_{\ell+1:k}} Q(x_\ell, x_{\ell+1}) \cdots Q(x_{\ell-1}, x_\ell) \psi(x_{\ell+1}, y_{\ell+1}) \cdots \psi(x_k, y_k) \\ &= \sum_{x_{\ell+1}} Q(x_\ell, x_{\ell+1}) \psi(x_{\ell+1}, y_{\ell+1}) \sum_{x_{\ell+2:k}} Q(x_{\ell+1}, x_{\ell+2}) \cdots Q(x_{\ell-1}, x_\ell) \\ &\quad \psi(x_{\ell+2}, y_{\ell+2}) \cdots \psi(x_k, y_k) \\ &= \sum_{x_{\ell+1}} Q(x_\ell, x_{\ell+1}) \psi(x_{\ell+1}, y_{\ell+1}) \beta_{x_{\ell+1}}^{\ell+1}(y_{\ell+2:k}) \end{aligned}$$

d'où le résultat. \square

Le système (2.9) admet la représentation vectorielle suivante :

$$\begin{aligned} \beta^\ell &\leftarrow Q\Psi(y_{\ell+1}) \beta^{\ell+1}, \quad 0 \leq \ell < k, \\ \beta^k &\leftarrow \underline{1}. \end{aligned}$$

où $\underline{1}$ est le vecteur colonne composé de N fois 1.

Proposition 2.1.6 *On se fixe $k \in \mathbb{N}$, les équations “forward” (2.4) et “backward” (2.9) sont duales l'une de l'autre :*

$$\sum_{x \in \mathcal{X}} \alpha_x^0 \beta_x^0 = \sum_{x \in \mathcal{X}} \alpha_x^\ell \beta_x^\ell = \sum_{x \in \mathcal{X}} \alpha_x^k$$

pour tout $0 \leq \ell \leq k$. Ces quantités sont égales à la vraisemblance $\mathcal{L}_k(\mathbf{M})$ définie en (2.6) ainsi qu'à la constante de normalisation qui apparaît dans (2.7).

Preuve En utilisant successivement l'équation “backward” (2.9a) et l'équation “forward” (2.4a), on obtient :

$$\begin{aligned} \sum_{x \in \mathcal{X}} \alpha_x^\ell \beta_x^\ell &= \sum_{x \in \mathcal{X}} \alpha_x^\ell \sum_{x' \in \mathcal{X}} Q(x, x') \beta_{x'}^{\ell+1} \psi(x', y_{\ell+1}) \\ &= \sum_{x' \in \mathcal{X}} \psi(x', y_{\ell+1}) \left[\sum_{x \in \mathcal{X}} \alpha_x^\ell Q(x, x') \right] \beta_{x'}^{\ell+1} \\ &= \sum_{x' \in \mathcal{X}} \alpha_{x'}^{\ell+1} \beta_{x'}^{\ell+1}. \end{aligned}$$

\square

Proposition 2.1.7 *Pour tout $\ell < k$, la distribution de probabilité jointe de X_ℓ et des observations $Y_{0:\ell}$ est donnée par :*

$$\mathbb{P}(X_\ell = x, Y_{0:k} = y_{0:k}) = \alpha_x^\ell \beta_x^\ell$$

pour tout $x \in \mathcal{X}$ et $y_0, \dots, y_k \in \mathcal{Y}$.

Preuve

$$\begin{aligned}
& \mathbb{P}(X_\ell = x_\ell, Y_{0:k} = y_{0:k}) \\
&= \sum_{\substack{x_{0:\ell-1} \\ x_{\ell+1:k}}} \mathbb{P}(Y_{0:k} = y_{0:k}, X_{0:k} = x_{0:k}) \\
&= \sum_{\substack{x_{0:\ell-1} \\ x_{\ell+1:k}}} \rho(x_0) Q(x_0, x_1) \cdots Q(x_{k-1}, x_k) \psi(x_0, y_0) \cdots \psi(x_k, y_k) \\
&= \left[\sum_{x_{0:\ell-1}} \rho(x_0) Q(x_0, x_1) \cdots Q(x_{\ell-1}, x_\ell) \psi(x_0, y_0) \cdots \psi(x_\ell, y_\ell) \right] \times \\
&\quad \left[\sum_{x_{\ell+1:k}} Q(x_\ell, x_{\ell+1}) \cdots Q(x_{k-1}, x_k) \psi(x_{\ell+1}, y_{\ell+1}) \cdots \psi(x_k, y_k) \right] \\
&= \alpha_{x_\ell}^\ell \beta_{x_\ell}^\ell.
\end{aligned}$$

□

Lissage : problème P2-(ii)

Pour tout $\ell < k$, la distribution conditionnelle de l'état X_ℓ sachant les observations $Y_{0:k}$ est donnée par :

$$\mathbb{P}(X_\ell = x | Y_{0:k} = y_{0:k}) = \frac{\alpha_x^\ell(y_{0:\ell}) \beta_x^\ell(y_{\ell+1:k})}{\sum_{x' \in \mathcal{X}} \alpha_{x'}^\ell(y_{0:\ell}) \beta_{x'}^\ell(y_{\ell+1:k})} \quad (2.10)$$

pour $x \in \mathcal{X}$, $y_0, \dots, y_k \in \mathcal{Y}$.

La propriété de dualité décrite dans la Proposition 2.1.6 implique que la constante de normalisation dans (2.10) ne dépend pas de k , en effet :

$$\sum_{x' \in \mathcal{X}} \alpha_{x'}^\ell \beta_{x'}^\ell = \sum_{x' \in \mathcal{X}} \alpha_{x'}^k = \mathcal{L}_k(\mathbf{M})$$

pour tout $0 \leq \ell \leq k$. Cette constante est la vraisemblance (2.6) du modèle.

b. Algorithme de Viterbi

La réponse au problème P2-(iii) est fournie par un algorithme de *programmation dynamique*, l'algorithme de Viterbi, qui permet de maximiser la distribution de probabilité conditionnelle de la suite des états $X_{0:k}$.

D'après (2.10) les variables “forward” et “backward” de Baum permettent seulement de calculer la loi conditionnelle de l'état à un instant donné sachant les observations.

Les états $x \in \mathcal{X}$ de la chaîne X sont des “étiquettes” et effectuer des opérations algébriques sur ces valeurs n’a pas de sens. On ne peut donc pas utiliser les lois conditionnelles précédentes afin de calculer des statistiques comme la moyenne conditionnelle. En revanche, il est possible de calculer des estimateurs comme *l’estimateur local du maximum à posteriori*² :

$$X_\ell^{\text{LMAP}} \stackrel{\text{déf}}{=} \underset{x \in \mathcal{X}}{\text{Argmax}} \mathbb{P}(X_\ell = x | Y_{0:k} = y_{0:k}).$$

Cependant, il peut arriver que la suite $X_{0:k}^{\text{LMAP}}$ ainsi générée ne soit pas cohérente avec le modèle : il peut en effet arriver que $X_\ell^{\text{LMAP}} = x$, $X_{\ell+1}^{\text{LMAP}} = x'$ alors que $Q(x, x') = 0$ (i.e. la chaîne X ne peut passer, à l’instant ℓ , de l’état x à l’état x'). Pour cette raison, on fait appel à *l’estimateur global du maximum a posteriori*, défini par :

$$X_{0:k}^{\text{MAP}} \stackrel{\text{déf}}{=} \underset{x_{0:k}}{\text{Argmax}} \mathbb{P}(X_{0:k} = x_{0:k} | Y_{0:k} = y_{0:k}). \quad (2.11)$$

Le calcul efficace de cet estimateur est fourni par un algorithme de programmation dynamique, appelé *algorithme de Viterbi*.

Notons que :

$$\begin{aligned} \underset{x_{0:k} \in \mathcal{X}^{k+1}}{\text{Argmax}} \mathbb{P}(X_{0:k} = x_{0:k} | Y_{0:k} = y_{0:k}) \\ &= \underset{x_{0:k} \in \mathcal{X}^{k+1}}{\text{Argmax}} \mathbb{P}(X_{0:k} = x_{0:k}, Y_{0:k} = y_{0:k}) / \mathbb{P}(Y_{0:k} = y_{0:k}) \\ &= \underset{x_{0:k} \in \mathcal{X}^{k+1}}{\text{Argmax}} \mathbb{P}(X_{0:k} = x_{0:k}, Y_{0:k} = y_{0:k}) \end{aligned}$$

pour tout $k \geq 0$, $y_0, \dots, y_k \in \mathcal{Y}$. Ainsi il est équivalent dans (2.11) de chercher l’argument du maximum de la loi conditionnelle ou de la loi jointe. On introduit :

$$\delta_x^k(y_{0:k}) \stackrel{\text{déf}}{=} \max_{x_{0:k-1} \in \mathcal{X}^k} \mathbb{P}(X_{0:k-1} = x_{0:k-1}, X_k = x, Y_{0:k} = y_{0:k})$$

pour tout $k \geq 0$, $x_0, \dots, x_{k-1}, x \in \mathcal{X}$, $y_0, \dots, y_k \in \mathcal{Y}$. Ces termes sont également appelées fonctions *valeur*. On utilisera également la notation simplifiée :

$$\delta_x^k = \delta_x^k(y_{0:k})$$

lorsqu’il n’y a pas d’ambiguïté sur les observations. On introduira également le vecteur colonne :

$$\delta^k = \delta^k(y_{0:k}) = [\delta_x^k(y_{0:k})]_{x \in \mathcal{X}} \quad (\text{vecteur colonne}).$$

2. En supposant, comme nous le ferons dans toute cette section, que dans chacun des cas le maximum est atteint en un point unique.

Théorème 2.1.8 (Programmation dynamique) La suite $\{\delta^k\}_{k \in \mathbb{N}}$ des fonctions valeur est solution de l'équation non-linéaire récurrente suivante :

$$\delta_x^{k+1} \leftarrow \psi(x, y_{k+1}) \max_{x' \in \mathcal{X}} [Q(x', x) \delta_{x'}^k], \quad n \geq 0, \quad x \in \mathcal{X}, \quad (2.12a)$$

$$\delta_x^0 \leftarrow \rho(x) \psi(x, y_0), \quad x \in \mathcal{X}. \quad (2.12b)$$

pour tout $k > 0, x \in \mathcal{X}, y_0, \dots, y_k \in \mathcal{Y}$.

Preuve D'après la Proposition 2.1.2 :

$$\begin{aligned} \mathbb{P}(X_{0:k+1} = x_{0:k+1}, Y_{0:k+1} = y_{0:k+1}) \\ = \rho(x_0) Q(x_0, x_1) \cdots Q(x_k, x_{k+1}) \psi(x_0, y_0) \cdots \psi(x_{k+1}, y_{k+1}) \\ = \psi(x_{k+1}, y_{k+1}) Q(x_k, x_{k+1}) \mathbb{P}(X_{0:k} = x_{0:k}, Y_{0:k} = y_{0:k}) \end{aligned}$$

pour tout $k \geq 0, x_{0:k+1}, y_{0:k+1}$. On en déduit que :

$$\begin{aligned} \max_{x_{0:k-1}} \mathbb{P}(X_{0:k+1} = x_{0:k+1}, Y_{0:k+1} = y_{0:k+1}) = \\ = \psi(x_{k+1}, y_{k+1}) Q(x_k, x_{k+1}) \delta_{x_k}^k(y_{0:k}). \end{aligned}$$

En maximisant cette dernière expression par rapport à x_k , on obtient :

$$\begin{aligned} \delta_{x_{k+1}}^{k+1}(y_{0:k+1}) &= \max_{x_{0:k}} \mathbb{P}(X_{0:k+1} = x_{0:k+1}, Y_{0:k+1} = y_{0:k+1}) \\ &= \max_{x_k} \max_{x_{0:k-1}} \mathbb{P}(X_{0:k+1} = x_{0:k+1}, Y_{0:k+1} = y_{0:k+1}) \\ &= \psi(x_{k+1}, y_{k+1}) \max_{x_k \in \mathcal{X}} [Q(x_k, x_{k+1}) \delta_{x_k}^k(y_{0:k})] \end{aligned}$$

d'où le résultat. □

Corollaire 2.1.9 (Algorithme de Viterbi) À un instant k fixé, après avoir calculer $\delta^{0:k}$, on obtient l'estimateur $X_{0:k}^{\text{MAP}}$ par récurrence rétrograde :

$$X_k^{\text{MAP}} \leftarrow \text{Argmax}_{x \in \mathcal{X}} \delta_x^k \quad (2.13)$$

puis, pour $\ell = k-1, \dots, 0$:

$$X_\ell^{\text{MAP}} \leftarrow \text{Argmax}_{x \in \mathcal{X}} [Q(x, X_{\ell+1}^{\text{MAP}}) \delta_x^\ell]. \quad (2.14)$$

Preuve Parmi toutes les trajectoires qui aboutissent dans l'état x' à l'instant $\ell + 1$, la trajectoire de plus grande probabilité est passée dans l'état :

$$I_\ell(x') \stackrel{\text{déf}}{=} \text{Argmax}_{x \in \mathcal{X}} [Q(x, x') \delta_x^\ell]$$

à l'instant ℓ précédent (en supposant que ce maximum est atteint en un point unique). En outre, on a nécessairement :

$$Q(I_\ell(x'), x') > 0$$

ce qui garantit que la transition de l'état $I_\ell(x')$ vers l'état x' est possible pour le modèle.

La trajectoire optimale est alors calculée de la façon suivante : on remarque d'abord que

$$\begin{aligned} \max_{x_{0:k}} \mathbb{P}(X_{0:k} = x_{0:k}, Y_{0:k} = y_{0:k}) &= \max_{x_k} \max_{x_{0:k-1}} \mathbb{P}(X_{0:k} = x_{0:k}, Y_{0:k} = y_{0:k}) \\ &= \max_{x_k} \delta_{x_k}^k(y_{0:k}). \end{aligned}$$

on en déduit (2.13) et de proche en proche (2.14). \square

c. Formules de re-estimation de Baum-Welsh

Il s'agit maintenant de proposer une méthode permettant d'ajuster les paramètres $\mathbf{M} = (\rho, Q, \psi)$ à certains critères d'optimisation. Le plus courant est celui de « mieux » correspondre aux observations. Toutefois il n'y a pas de solution générique et *réaliste* à ce problème, on peut néanmoins déterminer \mathbf{M} qui maximise localement la fonction de vraisemblance :

$$\mathbf{M} \mapsto \mathcal{L}_k(\mathbf{M}) \stackrel{\text{déf}}{=} \mathbb{P}_{\mathbf{M}}(Y_{0:k} = y_{0:k})$$

où $\mathbb{P}_{\mathbf{M}}$ est la probabilité sous laquelle le modèle \mathbf{M} est valide.

Il est également possible, si une partie de ces paramètres est connue, de déterminer les autres par la même méthode.

La méthode de re-estimation de Baum-Welsh (qui est une forme de l'algorithme EM) est une procédure de maximisation locale. Afin de déterminer un maximum (global) il est possible de faire appel à des algorithmes de type Metropolis (et de se confronter à toutes les difficultés de mise en œuvre). Il existe de nombreuses alternatives à la méthode de Baum-Welsh, comme les méthodes de gradient.

Considérons la probabilité de transition $x \rightarrow x'$ sachant le modèle \mathbf{M} et l'observation $Y_{0:k} = y_{0:k}$:

$$\xi_{x,x'}^\ell(y_{0:k}) = \mathbb{P}_{\mathbf{M}}(X_\ell = x, X_{\ell+1} = x' | Y_{0:k} = y_{0:k}).$$

Ce dernier terme peut s'écrire en fonction de $\alpha_x^\ell(y_{0:\ell})$ et $\beta_x^\ell(y_{\ell+1:k})$:

$$\begin{aligned}\xi_{x,x'}^\ell(y_{0:k}) &= \frac{\mathbb{P}_{\mathbf{M}}(X_\ell = x, X_{\ell+1} = x', Y_{0:k} = y_{0:k})}{\mathbb{P}_{\mathbf{M}}(Y_{0:k} = y_{0:k})} \\ &= \frac{\mathbb{P}_{\mathbf{M}}(X_\ell = x, X_{\ell+1} = x', Y_{0:k} = y_{0:k})}{\sum_{\bar{x}, \bar{x}' \in \mathcal{X}} \mathbb{P}_{\mathbf{M}}(X_\ell = \bar{x}, X_{\ell+1} = \bar{x}', Y_{0:k} = y_{0:k})} \\ &= \frac{\alpha_x^\ell(y_{0:\ell}) Q(x, x') \psi(x', y_{\ell+1}) \beta_{x'}^{\ell+1}(y_{\ell+2:k})}{\sum_{\bar{x}, \bar{x}' \in \mathcal{X}} \alpha_{\bar{x}}^\ell(y_{0:\ell}) Q(\bar{x}, \bar{x}') \psi(\bar{x}', y_{\ell+1}) \beta_{\bar{x}'}^{\ell+1}(y_{\ell+2:k})}\end{aligned}$$

Posons :

$$\gamma_x^\ell(y_{0:k}) \stackrel{\text{déf}}{=} \sum_{x' \in \mathcal{X}} \xi_{x,x'}^\ell(y_{0:k}),$$

comme $\gamma_x^\ell(y_{0:k}) = \mathbb{P}_{\mathbf{M}}(X_\ell = x | Y_{0:k} = y_{0:k})$, on a déjà vu que :

$$\gamma_x^\ell(y_{0:k}) = \frac{\alpha_x^\ell(y_{0:\ell}) \beta_x^\ell(y_{\ell+1:k})}{\sum_{x \in \mathcal{X}} \alpha_x^\ell(y_{0:\ell}) \beta_x^\ell(y_{\ell+1:k})}$$

Une sommation en $\ell = 0 : k-1$ de cette expression donne :

$$\begin{aligned}\sum_{\ell=0}^k \gamma_x^\ell(y_{0:k}) &= \sum_{\ell=0}^k \mathbb{P}_{\mathbf{M}}(X_\ell = x | Y_{0:k} = y_{0:k}) \\ &= \sum_{\ell=0}^k \mathbb{E}_{\mathbf{M}}(\mathbf{1}_{X_\ell=x} | Y_{0:k} = y_{0:k}) \\ &= \mathbb{E}_{\mathbf{M}}\left(\sum_{\ell=0}^k \mathbf{1}_{X_\ell=x} \middle| Y_{0:k} = y_{0:k}\right) \\ &= \mathbb{E}_{\mathbf{M}}(\#\{\ell : X_\ell = x\} | Y_{0:k} = y_{0:k})\end{aligned}$$

Il s'agit donc de l'estimateur du nombre de transitions issues de x de la suite $X_{0:k}$ au vu des observations $Y_{0:k} = y_{0:k}$.

De même :

$$\begin{aligned}\sum_{\ell=0}^{k-1} \xi_{x,x'}^\ell(y_{0:k}) &= \sum_{\ell=0}^{k-1} \mathbb{P}_{\mathbf{M}}(X_\ell = x, X_{\ell+1} = x' | Y_{0:k} = y_{0:k}) \\ &= \sum_{\ell=0}^{k-1} \mathbb{E}_{\mathbf{M}}(\mathbf{1}_{\{X_\ell=x, X_{\ell+1}=x'\}} | Y_{0:k} = y_{0:k}) \\ &= \mathbb{E}_{\mathbf{M}}\left(\sum_{\ell=0}^{k-1} \mathbf{1}_{\{X_\ell=x, X_{\ell+1}=x'\}} \middle| Y_{0:k} = y_{0:k}\right) \\ &= \mathbb{E}_{\mathbf{M}}(\#\{\ell = 0 : k-1 : X_\ell = x, X_{\ell+1} = x'\} | Y_{0:k} = y_{0:k})\end{aligned}$$

est l'estimateur du nombre de transitions $x \rightarrow x'$ de la suite $X_{0:k}$ au vu des observations $Y_{0:k} = y_{0:k}$.

Ainsi :

$$\sum_{\ell=0}^{k-1} \xi_{x,x'}^{\ell}(y_{0:k}) \qquad \sum_{\ell=0}^{k-1} \gamma_x^{\ell}(y_{0:k})$$

sont des estimateurs respectivement du nombre de transitions $x \rightarrow x'$ et du nombre de transitions issues de x dans la suite $X_{0:k}$, fondés sur l'observation de $Y_{0:k} = y_{0:k}$.

On peut alors proposer, partant du modèle $\mathbf{M} = (\rho, Q, \psi)$, les « re-estimateurs » suivants :

$$\bar{\rho}(x) = \gamma_x^0(y_{0:k}), \quad (2.15)$$

$$\bar{Q}(x, x') = \frac{\sum_{\ell=0}^{k-1} \xi_{x,x'}^{\ell}(y_{0:k})}{\sum_{\ell=0}^{k-1} \gamma_x^{\ell}(y_{0:k})}, \quad (2.16)$$

$$\bar{\psi}(x, y) = \frac{\sum_{\ell=0}^k \mathbf{1}_{y_k=y} \gamma_x^{\ell}(y_{0:k})}{\sum_{\ell=0}^k \gamma_x^{\ell}(y_{0:k})}. \quad (2.17)$$

En effet il est clair que :

$\bar{\rho}(x)$ = estimateur du nombre de d'états x à l'instant initial $\ell = 0$,

$\bar{Q}(x, x')$ = $\frac{\text{estimateur du nombre de transitions } x \rightarrow x'}{\text{estimateur du nombre de transitions issues de } x}$,

$\bar{\psi}(x, y)$ = $\frac{\text{estimateur du nombre de transitions issues de } x \text{ et émettant } y}{\text{estimateur du nombre de transitions issues de } x}$.

Étant donné le modèle courant $\mathbf{M} = (\rho, Q, \psi)$, on peut utiliser les formules (2.15) à (2.17) pour re-estimer le modèle et obtenir $\tilde{\mathbf{M}} = (\bar{\rho}, \bar{Q}, \bar{\psi})$. Baum et ses co-hauteurs ont montré que :

- soit le modèle initial \mathbf{M} est un point critique de la fonction de vraisemblance :

$$\nabla_{\mathbf{M}} \mathcal{L}_k(\mathbf{M}) = \nabla_{\mathbf{M}} \mathbb{P}_{\mathbf{M}}(Y_{0:k} = y_{0:k}) = 0$$

et alors $\tilde{\mathbf{M}} = \mathbf{M}$;

- soit le modèle $\tilde{\mathbf{M}}$ est (strictement) plus vraisemblable au sens où :

$$\mathcal{L}_k(\tilde{\mathbf{M}}) = \mathbb{P}_{\tilde{\mathbf{M}}}(Y_{0:k} = y_{0:k}) > \mathcal{L}_k(\mathbf{M}) = \mathbb{P}_{\mathbf{M}}(Y_{0:k} = y_{0:k}).$$

Itérer cette procédure donne un algorithme EM qui permet de déterminer un maximum local de la fonction de vraisemblance.

2.2 MMC à espace d'observation continu

Nous supposons ici que $\mathcal{Y} \stackrel{\text{déf}}{=} \mathbb{R}^d$.

Hypothèse 2.2.1 (Canal d'observation sans mémoire) *Pour tout $k \geq 1$, $x_1, \dots, x_k \in \mathcal{X}$:*

$$\mathbb{P}(Y_{0:k} \in dy_{0:k} | X_{0:k} = x_{0:k}) = \prod_{\ell=0}^k \mathbb{P}(Y_\ell \in dy_\ell | X_\ell = x_\ell)$$

c'est-à-dire $\text{loi}(Y_{0:k} | X_{0:k} = x_{0:k}) = \text{loi}(Y_0 | X_0 = x_0) \otimes \dots \otimes \text{loi}(Y_k | X_k = x_k)$.

On suppose que la loi conditionnelle de Y_k sachant X_k ne dépend pas de k et admet une densité, on pose :

$$\mathbb{P}(Y_k \in dy | X_k = x) = \psi(x, y) dy.$$

On vérifie aisément que $\psi(x, y)$ est bien un noyau gaussien au sens de la définition 1.2.5. On en déduit le résultat suivant, dont la preuve est laissée en exercice :

Proposition 2.2.2 *La loi du modèle de Markov caché (X, Y) est entièrement caractérisée par la loi initiale ρ et la matrice de transition Q de la chaîne de Markov X , ainsi que par le noyau markovien ψ , en effet :*

$$\mathbb{P}(X_{0:k} = x_{0:k}, Y_{0:k} \in dy_{0:k}) = \rho(x_0) \psi(x_0, y_0) \prod_{\ell=1}^k [Q(x_{\ell-1}, x_\ell) \psi(x_\ell, y_\ell)] dy_{0:k} \quad (2.18)$$

pour tout $k \in \mathbb{N}$, $x_0, \dots, x_k \in \mathcal{X}$.

On en déduit la définition suivante :

Définition 2.2.3 (Modèle de Markov caché) *Un modèle de Markov caché (X, Y) à espace d'observation continu est entièrement caractérisé par :*

(i) *La loi initiale ρ du processus d'état X :*

$$\rho(x) \stackrel{\text{déf}}{=} \mathbb{P}(X_0 = x), \quad x \in \mathcal{X}.$$

(ii) *La matrice de transition Q du processus d'état X :*

$$Q(x, x') \stackrel{\text{déf}}{=} \mathbb{P}(X_{k+1} = x' | X_k = x), \quad x, x' \in \mathcal{X}.$$

(iii) *Le noyau markovien ψ du processus Y sachant le processus X :*

$$\mathbb{P}(Y_k \in dy | X_k = x) = \psi(x, y) dy.$$

On désigne par $\mathbf{M} = (\rho, Q, \psi)$ les paramètres du modèle de Markov caché.

a. Algorithmes “forward”/“backward” de Baum

On considère un modèle de Markov caché $\mathbf{M} = (\rho, Q, \psi)$ à espace d'état fini $\mathcal{X} = \{1, \dots, N\}$ et à espace d'observation dénombrable ou continu (dans ce dernier cas on supposera que $\mathcal{Y} = \mathbb{R}^d$).

Il s'agit de calculer *efficacement* la distribution de probabilité de la suite d'observations $Y_{0:k}$ en fonction des paramètres du modèle \mathbf{M} , c'est-à-dire la *fonction de vraisemblance*.

On considère l'espace d'observation continu $\mathcal{Y} = \mathbb{R}^d$. D'après (2.18), la loi des observations $Y_{0:k}$ est donnée par

$$\begin{aligned} \mathbb{P}(Y_{0:k} \in dy_{0:k}) &= \\ &= \sum_{x_{0:k}} \mathbb{P}(X_{0:k} = x_{0:k}, Y_{0:k} \in dy_{0:k}) \\ &= \sum_{x_{0:k}} \rho(x_0) \psi(x_0, y_0) \prod_{\ell=1}^k [Q(x_{\ell-1}, x_\ell) \psi(x_\ell, y_\ell)] dy_{0:k} \end{aligned} \quad (2.19)$$

pour tout $k \in \mathbb{N}$.

La Remarque 2.1.4 est aussi valide dans ce contexte. Les preuves des Théorèmes 2.2.4 et 2.2.5, de la Proposition 2.2.6 sont identiques à celles du cas fini (cf. preuves des Théorèmes 2.1.3 et 2.1.5, de la Proposition 2.1.7) et ne seront donc pas reprises dans ce paragraphe.

Algorithme “forward” de Baum

On introduit la loi jointe de $Y_{0:k}$ et X_k :

$$\alpha_x^k(y_{0:k}) dy_{0:k} \stackrel{\text{déf}}{=} \mathbb{P}(Y_0 \in dy_{0:k}, X_k = x) \quad (2.20)$$

pour $k \in \mathbb{N}$, $y_0, \dots, y_k \in \mathcal{Y}$, $x \in \mathcal{X}$. L'expression $\alpha_x^k(y_{0:k})$ est appelée *variable “forward”*. On utilise également la notation simplifiée :

$$\alpha_x^k = \alpha_x^k(y_{0:k})$$

lorsqu'il n'y a pas d'ambiguïté sur les observations. Cette variable “forward” se représente aussi sous forme d'un vecteur ligne :

$$\alpha^k = \alpha^k(y_{0:k}) = [\alpha_x^k(y_{0:k})]_{x \in \mathcal{X}} \quad (\text{vecteur ligne}).$$

Théorème 2.2.4 (Algorithme “forward” de Baum) La suite $\{\alpha_k\}_{k \in \mathbb{N}}$ des variables “forward” est solution de l'équation récurrente suivante :

$$\alpha_x^{k+1} \leftarrow \psi(x, y_{k+1}) \sum_{x' \in \mathcal{X}} Q(x', x) \alpha_{x'}^k, \quad (2.21a)$$

$$\alpha_x^0 \leftarrow \rho(x) \psi(x, y_0) \quad (2.21b)$$

pour tout $k \geq 0$, $x \in \mathcal{X}$, $y_0, \dots, Y_k \in \mathcal{Y}$.

Comme dans le cas discret, le système (2.21) admet une représentation vectorielle. Posons

$$\Psi(y) \stackrel{\text{déf}}{=} \text{diag}[\psi(x, y); x \in \mathcal{X}]$$

alors :

$$\begin{aligned} \alpha^{k+1} &\leftarrow \alpha^k Q \Psi(y_{k+1}), \quad k \geq 0, \\ \alpha^0 &\leftarrow \rho \Psi(y_0). \end{aligned}$$

La loi des observations $Y_{0:k}$ admet la densité suivante :

$$\mathbb{P}(Y_{0:k} \in dy_{0:k}) = \left[\sum_{x \in \mathcal{X}} \alpha_x^k(y_{0:k}) \right] dy_{0:k}.$$

Cette formulation répond donc à la question du problème **P1**, en effet la fonction de vraisemblance associée au modèle **M** est :

$$\mathcal{L}_k(\mathbf{M}) \stackrel{\text{déf}}{=} \sum_{x \in \mathcal{X}} \alpha_x^k(y_{0:k}). \quad (2.22)$$

La Remarque 2.1.4 est aussi valide dans ce cas.

Filtrage : problème **P2**–(i)

La loi conditionnelle de l'état présent X_k sachant les observations $Y_{0:k}$ est donnée par :

$$\mathbb{P}(X_k = x | Y_{0:k} = y_{0:k}) = \frac{\alpha_x^k}{\sum_{x' \in \mathcal{X}} \alpha_{x'}^k}, \quad x \in \mathcal{X}.$$

Ainsi, $x \mapsto \alpha_x^k$ apparaît comme une distribution de probabilité conditionnelle *non normalisée*. La constante de normalisation est la vraisemblance (2.22).

Algorithme “backward” de Baum

Soit $k \geq 1$ un instant fixé. Pour tout instant ℓ strictement antérieur à k , on considère la loi conditionnelle des observations $Y_{\ell+1:k}$ sachant que $X_\ell = x$:

$$\beta_x^\ell(y_{\ell+1:k}) dy_{\ell+1:k} \stackrel{\text{déf}}{=} \mathbb{P}(Y_{\ell+1:k} \in dy_{\ell+1:k} | X_\ell = x) \quad (2.23)$$

pour $x \in \mathcal{X}$, $y_{\ell+1}, \dots, y_k \in \mathcal{Y}$. L'expression $\beta_x^\ell(Y_{\ell+1:k})$ est également *variable “backward”*. On utilise aussi la notation simplifiée :

$$\beta_x^\ell = \beta_x^\ell(y_{\ell+1:k})$$

lorsqu'il n'y a pas d'ambiguïté sur les observations. On introduit la notation vectorielle :

$$\beta^\ell = \beta^\ell(y_{\ell+1:k}) = [\beta_x^\ell(y_{\ell+1:k})]_{x \in \mathcal{X}} \quad (\text{vecteur colonne}).$$

Il faut noter que β^ℓ dépend de k .

Théorème 2.2.5 (Algorithme “backward” de Baum) *On se fixe $k > 0$, la suite $\{\beta^\ell\}_{\ell=0:k}$ des variables “backward” est solution de l'équation récurrente rétrograde suivante :*

$$\beta_x^\ell \leftarrow \sum_{x' \in \mathcal{X}} Q(x, x') \psi(x', y_{\ell+1}) \beta_{x'}^{\ell+1} \quad (2.24a)$$

$$\beta_x^k \leftarrow 1 \quad (2.24b)$$

pour tout $0 \leq \ell < k$, $x \in \mathcal{X}$, $y_0, \dots, y_k \in \mathcal{Y}$.

Le système (2.24) s'écrit également sous forme vectorielle :

$$\begin{aligned} \beta^\ell &\leftarrow Q \Psi(y_{\ell+1}) \beta^{\ell+1}, \quad 0 \leq \ell < k, \\ \beta^k &\leftarrow \underline{1}. \end{aligned}$$

Proposition 2.2.6 *Pour tout $\ell < k$, la loi jointe de X_ℓ et $Y_{0:k}$ est donnée par :*

$$\mathbb{P}(X_\ell = x, Y_{0:k} \in dy_{0:k}) = \alpha_x^\ell(y_{0:\ell}) \beta_x^\ell(y_{\ell+1:k}) dy_{0:k}$$

pour tout $x \in \mathcal{X}$.

Lissage : problème P2-(ii)

Pour tout $\ell < k$, la distribution conditionnelle de l'état X_ℓ sachant les observations $Y_{0:k}$ est donnée par :

$$\mathbb{P}(X_\ell = x | Y_{0:k} = y_{0:k}) = \frac{\alpha_x^\ell(y_{0:\ell}) \beta_x^\ell(y_{\ell+1:k})}{\sum_{x' \in \mathcal{X}} \alpha_{x'}^\ell(y_{0:\ell}) \beta_{x'}^\ell(y_{\ell+1:k})}.$$

La propriété de dualité de la Proposition 2.1.6 est également valide dans ce cas.

b. Algorithme de Viterbi

La réponse au problème P2-(iii) est fournie par un algorithme de *programmation dynamique*, l'algorithme de Viterbi, qui permet de maximiser la distribution de probabilité conditionnelle de la suite des états $X_{0:k}$.

D'après (2.10) les variables “forward” et “backward” de Baum permettent seulement de calculer la loi conditionnelle de l'état à un instant donné sachant les observations.

Les états $\{1, \dots, N\}$ de la chaîne X sont des “étiquettes” et effectuer des opérations algébriques sur ces valeurs n'a ainsi pas de sens. On ne peut donc pas utiliser les lois conditionnelles précédentes afin de calculer des statistiques comme la moyenne conditionnelle. En revanche, il est possible de calculer des estimateurs comme *l'estimateur local du maximum a posteriori*³ :

$$X_\ell^{\text{LMAP}} \stackrel{\text{déf}}{=} \underset{x \in \mathcal{X}}{\text{Argmax}} \mathbb{P}(X_\ell = x | Y_{0:k} = y_{0:k}).$$

Cependant, il peut arriver que la suite $X_{0:k}^{\text{LMAP}}$ ainsi générée ne soit pas cohérente avec le modèle : il peut en effet arriver que $X_\ell^{\text{LMAP}} = x$, $X_{\ell+1}^{\text{LMAP}} = x'$ alors que $Q(x, x') = 0$ (i.e. la chaîne X ne peut passer, à l'instant ℓ , de l'état x à l'état x'). Pour cette raison, on fait appel à *l'estimateur global du maximum a posteriori*, défini par :

$$X_{0:k}^{\text{MAP}} \stackrel{\text{déf}}{=} \underset{x_{0:k}}{\text{Argmax}} \mathbb{P}(X_{0:k} = x_{0:k} | Y_{0:k} = y_{0:k}). \quad (2.25)$$

Le calcul efficace de cet estimateur est fourni par un algorithme de programmation dynamique, appelé *algorithme de Viterbi*.

3. En supposant, comme nous le ferons dans toute cette section, que dans chacun des cas le maximum est atteint en un point unique.

La méthode et certains énoncés de résultats sont identiques à ceux du paragraphe précédent, nous ne contenterons donc d'énoncer les résultats sans les démontrer.

D'après la formule de Bayes, la trajectoire qui maximise la distribution de probabilité conditionnelle de $X_{0:k}$ sachant les observations $Y_{0:k}$ maximise également la distribution jointe de $X_{0:k}$ et des observations $Y_{0:k}$, c'est-à-dire :

$$\begin{aligned} \mathbb{P}(X_{0:k} = x_{0:k} | Y_{0:k} = y_{0:k}) \mathbb{P}(Y_{0:k} \in dy_{0:k}) &= \mathbb{P}(X_{0:k} = x_{0:k}, Y_{0:k} \in dy_{0:k}) \\ &= \rho(x_0) \psi(x_0, y_0) \prod_{\ell=1}^k [Q(x_{\ell-1}, x_\ell) \psi(x_\ell, y_\ell)] dy_{0:k} \end{aligned}$$

Donc maximiser $\mathbb{P}(X_{0:k} = x_{0:k} | Y_{0:k} = y_{0:k})$ en $x_{0:k}$ est équivalent à maximiser $\rho(x_0) \psi(x_0, y_0) \prod_{\ell=1}^k [Q(x_{\ell-1}, x_\ell) \psi(x_\ell, y_\ell)]$. On pose :

$$\delta_{x_k}^k(y_{0:k}) \stackrel{\text{déf}}{=} \max_{x_{0:k-1}} \rho(x_0) \psi(x_0, y_0) \prod_{\ell=1}^k [Q(x_{\ell-1}, x_\ell) \psi(x_\ell, y_\ell)]$$

pour tout $k \geq 0$, $x_k \in \mathcal{X}$, $y_0, \dots, y_k \in \mathcal{Y}$. Cette expression est également appelée fonction *valeur*.

Théorème 2.2.7 (Programmation dynamique) *La suite $\{\delta^k\}_{k \in \mathbb{N}}$ est solution de l'équation récurrente suivante :*

$$\delta_x^{k+1} \leftarrow \psi(x, y_{k+1}) \left[\max_{x' \in \mathcal{X}} Q(x', x) \delta_{x'}^k \right], \quad (2.26a)$$

$$\delta_x^0 \leftarrow \rho(x) \psi(x, y_0) \quad (2.26b)$$

pour tout $k \geq 0$, $x \in \mathcal{X}$, $y_0, \dots, y_k \in \mathcal{Y}$.

On définit :

$$I_k(x') \stackrel{\text{déf}}{=} \text{Argmax}_{x \in \mathcal{X}} [Q(x, x') \delta_x^k], \quad n \geq 0, \quad x' \in \mathcal{X}$$

(en supposant que le maximum est atteint en un point unique).

Corollaire 2.2.8 (Algorithme de Viterbi) *La trajectoire optimale aboutit dans l'état :*

$$X_k^{\text{MAP}} = \text{Argmax}_{x \in \mathcal{X}} \delta_x^k \quad (2.27)$$

(en supposant que le maximum est atteint en un point unique) à l'instant k . La trajectoire optimale est passée dans l'état :

$$X_k^{\text{MAP}} = I_k(X_{k+1}^{\text{MAP}}). \quad (2.28)$$

3

MODÈLES DE MARKOV CACHÉS À ESPACE D'ÉTAT CONTINU

Un système à espace d'état est un couple de processus : d'une part un processus d'état $(X_k)_{k \geq 0}$ et d'autre part un processus d'observation $(Y_k)_{k \geq 1}$. On suppose qu'on ne dispose pas d'observation à l'instant $k = 0$. Le processus d'état est une chaîne de Markov de loi initiale $p(X_0)$ et de loi de transition $p(X_k|X_{k-1})$. Le lien processus d'observation/processus d'état est régi par une hypothèse de canal sans mémoire (cf. Fig. 3.1) : *conditionnellement aux états, les observations sont indépendantes et l'observation X_ℓ ne dépend des états $X_{1:k}$ que par l'intermédiaire de X_ℓ , c'est-à-dire :*

$$p(Y_{1:k}|X_{0:k}) = \prod_{\ell=1}^k p(Y_\ell|X_\ell) \quad (3.1)$$

où $p(Y_\ell|X_\ell)$ désigne la *fonction de vraisemblance locale*.

Sous les hypothèses de Markov (1.18) et de canal sans mémoire (3.1), ce modèle est déterminé par la donnée de la loi initiale $p(X_0)$, de loi de transi-

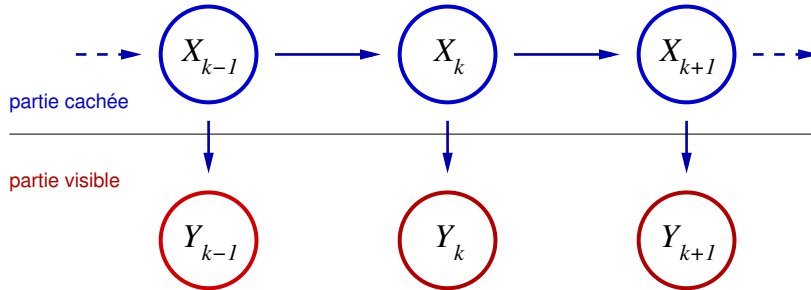


FIGURE 3.1 – *Hypothèse de canal sans mémoire*

tion $p(X_k|X_{k-1})$ et de la fonction de vraisemblance $p(Y_k|X_k)$. En effet la loi du système est :

$$p(X_{0:k}, Y_{1:k}) = p(Y_{1:k}|X_{0:k}) p(X_{0:k}).$$

D'après les propriétés (1.18) et (3.1) :

$$p(X_{0:k}, Y_{1:k}) = p(X_0) \prod_{\ell=1}^k p(X_\ell|X_{\ell-1}) p(Y_\ell|X_\ell).$$

Exemple 3.0.1 (système non linéaire à espace d'état) On considère le système

$$\begin{aligned} X_{k+1} &= f_k(X_k, W_k), \\ Y_k &= h_k(X_k) + V_k \end{aligned}$$

où X_0, W_k, V_k sont indépendants et

$$W_k \sim p(W_k) = q_k^W(W_k), \quad V_k \sim p(V_k) = q_k^V(V_k).$$

On a déjà vu que $(X_k)_{k \geq 0}$ est markovien et on a une expression de la loi de transition. Calculons maintenant la fonction de vraisemblance locale

$$\begin{aligned} \mathbb{E}[\varphi(Y_k)|X_k] &= \mathbb{E}[\varphi(h_k(X_k) + V_k)|X_k] \\ &= \int \varphi(h_k(X_k) + v) q_k^V(v) dv \\ &= \int \varphi(y) q_k^V(y - h_k(X_k)) dy \end{aligned}$$

donc

$$p(Y_k|X_k) = q_k^V(Y_k - h_k(X_k)).$$

Dans le cas particulier $V_k \sim N(0, \sigma_V^2 I)$ on a

$$p(Y_k|X_k) \propto \exp\left(-\frac{1}{2\sigma_V^2} |Y_k - h_k(X_k)|^2\right).$$

Exemple 3.0.2 (système à espace d'état : cas linéaire/gaussien)

$$\begin{aligned} X_{k+1} &= F_k X_k + f_k + G_k W_k, & X_0 &\sim N(\bar{X}_0, Q_0), \\ Y_k &= H_k X_k + h_k + V_k \end{aligned}$$

où X_0, W_k, V_k sont indépendants et $W_k \sim N(0, Q_k^W), V_k \sim N(0, Q_k^V)$, alors

$$\begin{aligned} p(X_k|X_{k-1}) &= N(F_{k-1} X_{k-1} + f_{k-1}, G_{k-1} Q_k^W G_{k-1}^*), \\ p(Y_k|X_k) &= N(H_k X_k + h_k, Q_k^V). \end{aligned}$$

Le processus état/observation est gaussien car, pour tout k , $(X_{0:k}, Y_{1:k})$ est gaussien. Il s'agit en effet de la transformée linéaire du vecteur aléatoire gaussien $(X_0, W_{0:k-1}, V_{1:k})$.

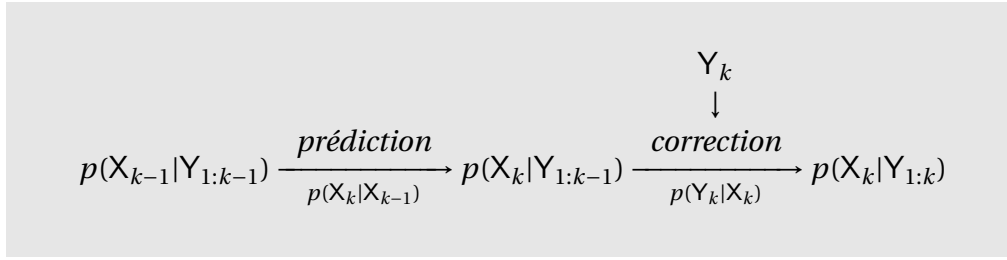


FIGURE 3.2 – Décomposition d'une itération du filtre non linéaire optimal en une étape de prédiction, faisant intervenir la loi de transition, et une étape de correction faisant intervenir la nouvelle observation et la fonction vraisemblance locale associée.

3.1 Formule de Bayes récursive (filtre bayésien)

On considère un système à espace d'état (X_k, Y_k) de loi initiale $p(X_0)$, de loi de transition $p(X_k|X_{k-1})$ et de fonction de vraisemblance locale $p(Y_k|X_k)$. On fait les hypothèses de Markov (1.18) et de canal sans mémoire (3.1).

Le problème est d'estimer X_k en se fondant sur les connaissances a priori et sur les observations $Y_{1:k}$. La plupart des applications sont en temps réel, ce problème d'estimation doit donc être résolu de façon récursive en traitant les observations au fur et à mesure de leur disponibilité.

D'après le chapitre précédent, on sait qu'il s'agit donc de déterminer la densité conditionnelle $p(X_k|Y_{1:k})$, appelée filtre. On sera également amené à déterminer le filtre prédit $p(X_k|Y_{1:k-1})$.

Ce calcul se fait de façon récursive selon le schéma classique en deux étapes :

Proposition 3.1.1 (filtre non linéaire optimal) *Le filtre se calcule de façon récursive en deux étapes (cf. FIG. 3.2). Le filtre prédit $p(X_k|Y_{1:k-1})$ à l'étape k se déduit du filtre $p(X_{k-1}|Y_{1:k-1})$ à l'étape k selon la formule de prédiction*

$$p(X_k|Y_{1:k-1}) = \int_{\mathbb{R}^n} p(X_k|X_{k-1}) p(X_{k-1}|Y_{1:k-1}) dX_{k-1}. \quad (3.2)$$

Le filtre $p(X_k|Y_{1:k})$ à l'étape k se déduit du filtre prédit $p(X_k|Y_{1:k-1})$ à l'étape k selon la formule de correction

$$p(X_k|Y_{1:k}) = \frac{p(Y_k|X_k) p(X_k|Y_{1:k-1})}{\int_{\mathbb{R}^n} p(Y_k|X_k) p(X_k|Y_{1:k-1}) dX_k}. \quad (3.3)$$

Preuve

Étape de prédiction

La première étape est une simple application de l'équation de Chapman–Kolmogorov (1.19)

$$\begin{aligned} p(X_k|Y_{1:k-1}) &= \int p(X_k|X_{k-1}, Y_{1:k-1}) p(X_{k-1}|Y_{1:k-1}) dX_{k-1} \\ &= \int p(X_k|X_{k-1}) p(X_{k-1}|Y_{1:k-1}) dX_{k-1}. \end{aligned}$$

L'étape sera démontrée si l'on vérifie que $p(X_k|X_{k-1}, Y_{1:k-1}) = p(X_k|X_{k-1})$.

$$\begin{aligned} p(X_k|X_{k-1}, Y_{1:k-1}) &= \frac{p(X_k, X_{k-1}, Y_{1:k-1})}{p(X_{k-1}, Y_{1:k-1})} \\ &= \frac{\int dX_0 \cdots \int dX_{k-2} p(X_{0:k}, Y_{1:k-1})}{\int dX_0 \cdots \int dX_{k-2} p(X_{0:k-1}, Y_{1:k-1})} \\ &= \frac{\int dX_0 \cdots \int dX_{k-2} p(X_0) \prod_{\ell=1}^{k-1} p(X_\ell|X_{\ell-1}) p(Y_\ell|X_\ell) p(X_k|X_{k-1})}{\int dX_0 \cdots \int dX_{k-2} p(X_0) \prod_{\ell=1}^{k-1} p(X_\ell|X_{\ell-1}) p(Y_\ell|X_\ell)} \\ &= p(X_k|X_{k-1}). \end{aligned}$$

Étape de correction

D'après la formule de Bayes on a les deux décompositions suivantes

$$\begin{aligned} p(X_{0:k}, Y_{1:k}) &= p(X_{0:k}|Y_{1:k}) p(Y_{1:k}) \\ &= p(X_{0:k}|Y_{1:k}) p(Y_k|Y_{1:k-1}) p(Y_{1:k-1}) \end{aligned}$$

et

$$\begin{aligned} p(X_{0:k}, Y_{1:k}) &= p(Y_k|X_{0:k}, Y_{1:k-1}) p(X_{0:k}, Y_{1:k-1}) \\ &= p(Y_k|X_k) p(X_{0:k}|Y_{1:k-1}) p(Y_{1:k-1}). \end{aligned}$$

Ainsi, après simplification par $p(Y_{1:k-1})$, on obtient :

$$p(X_{0:k}|Y_{1:k}) p(Y_k|Y_{1:k-1}) = p(Y_k|X_k) p(X_{0:k}|Y_{1:k-1})$$

en intégrant chacun des membres de cette égalité en $\int \cdots \int dX_{0:k-1}$, on obtient

$$p(X_k|Y_{1:k}) p(Y_k|Y_{1:k-1}) = p(Y_k|X_k) p(X_k|Y_{1:k-1})$$

donc

$$\begin{aligned}
 p(X_k|Y_{1:k}) &= \frac{p(Y_k|X_k) p(X_k|Y_{1:k-1})}{p(Y_k|Y_{1:k-1})} \\
 &= \frac{p(Y_k|X_k) p(X_k|Y_{1:k-1})}{\int p(Y_k|X_k, Y_{1:k-1}) p(X_k|Y_{1:k-1}) dX_k} \\
 &= \frac{p(Y_k|X_k) p(X_k|Y_{1:k-1})}{\int p(Y_k|X_k) p(X_k|Y_{1:k-1}) dX_k}.
 \end{aligned}$$

□

Maintenant que les équations du filtre non linéaire “optimal” sont établies, que faire en pratique? Deux cas se dégagent selon que l’on puisse ou non résoudre explicitement les équations (3.2)–(3.3).

Ces équations admettent une solution explicite dans le cas linéaire/gaussien. On obtient alors le filtre de Kalman–Bucy.

Dans le cas non linéaire ces équations n’admettent en général pas de solution explicite. Il est donc nécessaire de faire appel à des techniques d’approximation. La première idée développée a été de linéariser le système autour de l’estimée courante : c’est l’idée du filtre de Kalman étendu. C’est une technique qui présente de nombreux avantages mais, d’une part, elle ne donne pas de résultat satisfaisant dans tous les cas et, d’autre part, il n’existe pas dans ce cas de résultat de convergence.

Une autre approche, qui ne sera pas présentée ici, consiste à résoudre l’équation de prédiction à l’aide de schémas d’approximation numérique comme les différences finies. C’est une technique assez lourde qui s’appuie sur une discrétisation contraignante de l’espace d’état.

Enfin, depuis une dizaine d’années, les méthodes d’approximation particulière connaissent un développement important. Ces techniques sont simples à mettre en œuvre et robustes en pratique.

4

FILTRE DE KALMAN ET EXTENSIONS

Rudolph E. Kalman a publié ses travaux sur le filtrage linéaire dans les années 60, voir [35, 36]. De nombreux livres détaillent ses travaux ainsi que le filtre de Kalman étendu, voir par exemple [22], [2] et [42].

4.1 Système linéaire gaussien

On considère un modèle linéaire et gaussien :

$$X_{k+1} = F_k X_k + f_k + G_k W_k, \quad k \geq 0, \quad (4.1)$$

$$Y_k = H_k X_k + h_k + V_k, \quad k \geq 1 \quad (4.2)$$

où $X_0 \sim \mathcal{N}(\bar{X}_0, Q_0)$, $W_k \sim \mathcal{N}(0, Q_k^W)$, $V_k \sim \mathcal{N}(0, Q_k^V)$; X_0 , W_k et V_k sont indépendants. On suppose que $Q_k^V > 0$.

La sortie du système (4.1)–(4.2) est gaussienne, la loi conditionnelle de X_k sachant $Y_{1:k}$ (ou sachant $Y_{1:k-1}$) est donc également gaussienne. On pose $p(X_k|Y_{1:k}) = \mathcal{N}(\hat{X}_k, R_k)$ et $p(X_k|Y_{1:k-1}) = \mathcal{N}(\hat{X}_{k-}, R_{k-})$ avec :

$$\begin{aligned} \hat{X}_k &= \mathbb{E}[X_k|Y_{1:k}], & R_k &= \mathbb{E}[(X_k - \hat{X}_k)(X_k - \hat{X}_k)^*], \\ \hat{X}_{k-} &= \mathbb{E}[X_k|Y_{1:k-1}], & R_{k-} &= \mathbb{E}[(X_k - \hat{X}_{k-})(X_k - \hat{X}_{k-})^*]. \end{aligned}$$

On note (et on vérifiera) que les covariances R_{k-} , R_k sont déterministes.

L'idée est toujours la même. On détermine $p(X_k|Y_{1:k})$ à partir de $p(X_{k-1}|Y_{1:k-1})$ en deux étapes. Pour la prédiction : on calcule $p(X_k|Y_{1:k-1})$ à l'aide de l'équation d'état. Pour la correction : on corrige la prédiction en tenant compte de la nouvelle observation Y_k et de l'équation d'observation.

4.2 Processus innovation

Un point important est de déterminer ce que la nouvelle observation Y_k apporte de nouveau par rapport aux observations passées $Y_{1:k-1}$? La réponse est donnée par le processus innovation (cf. Fig. 4.1) :

$$v_k \stackrel{\text{déf}}{=} Y_k - \hat{Y}_{k-} \quad \text{où} \quad \hat{Y}_{k-} \stackrel{\text{déf}}{=} \mathbb{E}[Y_k | Y_{1:k-1}] \quad (4.3)$$

Proposition 4.2.1 *L'innovation s'exprime de la façon suivante :*

$$v_k = Y_k - (H_k \hat{X}_{k-} + h_k). \quad (4.4)$$

Elle forme une suite de variables aléatoires indépendantes de loi $v_k \sim \mathcal{N}(0, Q_k^v)$ avec

$$Q_k^v = H_k R_k^- H_k^* + Q_k^V \quad (4.5)$$

et $v_k \perp\!\!\!\perp Y_{1:k-1}$. De plus :

$$\mathbb{E}[(X_k - \hat{X}_{k-}) v_k^*] = R_k^- H_k^*. \quad (4.6)$$

Preuve Par définition

$$\begin{aligned} v_k &= Y_k - \mathbb{E}[Y_k | Y_{1:k-1}] \\ &= Y_k - (H_k \mathbb{E}[X_k | Y_{1:k-1}] + h_k + \mathbb{E}[V_k | Y_{1:k-1}]). \end{aligned}$$

Par indépendance, $\mathbb{E}[V_k | Y_{1:k-1}] = \mathbb{E}[V_k] = 0$, d'où (4.4).

Pour vérifier les autres points il suffit de noter que

$$v_k = Y_k - (H_k \hat{X}_{k-} + h_k) = H_k (X_k - \hat{X}_{k-}) + V_k.$$

□

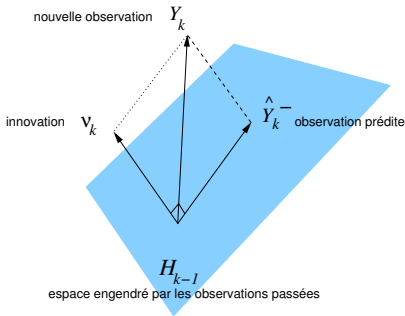


FIGURE 4.1 – Le processus innovation : l'information “fraîche” contenue dans Y_k .

Ainsi les informations contenues dans $\{Y_{1:k-1} Y_k\}$ et dans $\{Y_{1:k-1} v_k\}$ sont identiques. De plus $v_k \perp\!\!\!\perp Y_{1:k-1}$, cela signifie donc que v_k est l'information “fraîche” contenue dans Y_k , i.e. l'information concernant X_k contenue dans Y_k qui n'est pas déjà contenue dans $Y_{1:k-1}$.

4.3 Filtre de Kalman

Proposition 4.3.1 *La loi conditionnelle de X_k sachant $Y_{1:k}$ (resp. de X_k sachant $Y_{1:k-1}$) est gaussienne $\mathcal{N}(\hat{X}_k, R_k)$ (resp. $\mathcal{N}(\hat{X}_{k-}, R_{k-})$) où les espérances/covariances conditionnelles sont calculées récursivement à l'aide du filtrage de Kalman :*

initialisation

$$\hat{X}_0 = \bar{X}_0 = \mathbb{E}[X_0], \quad R_0 = Q_0 = \text{cov}(X_0), \quad (4.7)$$

prédiction

$$\hat{X}_{k-} = F_{k-1} \hat{X}_{k-1} + f_{k-1}, \quad (4.8)$$

$$R_{k-} = F_{k-1} R_{k-1} F_{k-1}^* + G_{k-1} Q_{k-1}^W G_{k-1}^*, \quad (4.9)$$

correction

$$K_k = R_{k-} H_k^* [H_k R_{k-} H_k^* + Q_k^V]^{-1}, \quad (4.10)$$

$$\hat{X}_k = \hat{X}_{k-} + K_k [Y_k - (H_k \hat{X}_{k-} + h_k)], \quad (4.11)$$

$$R_k = [I - K_k H_k] R_{k-}. \quad (4.12)$$

Le terme K_k est appelé gain.

Pour assurer que ce filtre soit bien défini on suppose que $Q_k^V > 0$, ainsi

$$H_k R_{k-} H_k^* + Q_k^V \geq Q_k^V > 0$$

et le système linéaire apparaissant dans (4.10) est bien posé. On peut noter que les covariances ne dépendent pas des observations. Enfin le filtre de Kalman est à la fois un théorème et un algorithme.

Preuve L'initialisation (4.7) est évidente. On prouve l'étape de prédiction. Le vecteur $(X_k, Y_{1:k-1})$ est gaussien, la loi conditionnelle de X_k sachant $Y_{1:k-1}$ est donc gaussienne. Notons \hat{X}_{k-} sa moyenne et R_{k-} sa covariance. On a

$$\begin{aligned} \hat{X}_{k-} &= \mathbb{E}[X_k | Y_{1:k-1}] \\ &= F_{k-1} \mathbb{E}[X_{k-1} | Y_{1:k-1}] + f_{k-1} + G_{k-1} \mathbb{E}[W_{k-1} | Y_{1:k-1}]. \end{aligned}$$

Comme $\mathbb{E}[X_{k-1}|Y_{1:k-1}] = \hat{X}_{k-1}$ et $\mathbb{E}[W_{k-1}|Y_{1:k-1}] = \mathbb{E}[W_{k-1}] = 0$ on en déduit (4.8) d'une part et

$$X_k - \hat{X}_{k-} = F_{k-1} (X_{k-1} - \hat{X}_{k-1}) + G_{k-1} W_{k-1}$$

d'autre part, d'où

$$\begin{aligned} R_{k-} &= \mathbb{E}[(X_k - \hat{X}_{k-})(X_k - \hat{X}_{k-})^* | Y_{1:k-1}] \\ &= \mathbb{E}[(F_{k-1}(X_{k-1} - \hat{X}_{k-1}) + G_{k-1}W_{k-1})(F_{k-1}(X_{k-1} - \hat{X}_{k-1}) + G_{k-1}W_{k-1})^* | Y_{1:k-1}] \\ &= F_{k-1} R_{k-1} F_{k-1}^* + G_{k-1} Q_{k-1}^W G_{k-1}^* \end{aligned}$$

(on note que cette expression ne dépend pas des observations). En effet $(X_k - \hat{X}_k) \perp\!\!\!\perp W_k$ donc $\mathbb{E}[(X_{k-1} - \hat{X}_{k-1}) W_{k-1}^*] = 0$. Ce qui prouve (4.9).

On démontre les équations de l'étape de correction. Le vecteur $(X_k, Y_{1:k})$ est gaussien, la loi conditionnelle de X_k sachant $Y_{1:k}$ l'est donc également. Notons \hat{X}_k sa moyenne et R_k sa covariance. On a

$$\hat{X}_k = \mathbb{E}[X_k | Y_{1:k}] = \hat{X}_{k-} + \mathbb{E}[X_k - \hat{X}_{k-} | Y_{1:k}].$$

D'après la définition de l'innovation (cf. § 4.2)

$$\mathbb{E}[X_k - \hat{X}_{k-} | Y_{1:k}] = \mathbb{E}[X_k - \hat{X}_{k-} | Y_{1:k-1}, v_k]$$

en utilisant la Proposition 1.2.4 (ii), comme $Y_{1:k-1} \perp\!\!\!\perp (X_k - \hat{X}_{k-}, v_k)$ on a

$$\mathbb{E}[X_k - \hat{X}_{k-} | Y_{1:k-1}, v_k] = \mathbb{E}[X_k - \hat{X}_{k-} | v_k].$$

Ainsi

$$\hat{X}_k = \hat{X}_{k-} + \mathbb{E}[X_k - \hat{X}_{k-} | v_k]. \quad (4.13)$$

On calcule la loi conditionnelle de $X_k - \hat{X}_{k-}$ sachant v_k . D'après la Proposition 4.2.1, le vecteur $(X_k - \hat{X}_{k-}, v_k)$ est gaussien, centré de covariance

$$\begin{pmatrix} R_{k-} & R_{k-} H_k^* \\ H_k R_{k-} & H_k R_{k-} H_k^* + Q_k^v \end{pmatrix}.$$

D'après la Proposition 1.2.7, la loi conditionnelle de $X_k - \hat{X}_{k-}$ sachant v_k est gaussienne de moyenne et covariance explicite. Avec (4.13) on en déduit (4.10), (4.11) et (4.12). \square

```

% initialisation
 $\hat{X}_0 \leftarrow \bar{X}_0$ 
 $R_0 \leftarrow Q_0$ 

% itérations
pour  $k = 1, 2, 3 \dots$  faire

    % prédiction
     $\hat{X}_{k^-} \leftarrow F_{k-1} \hat{X}_{k-1} + f_{k-1}$ 
     $R_{k^-} \leftarrow F_{k-1} R_{k-1} F_{k-1}^* + G_{k-1} Q_{k-1}^W G_{k-1}^*$ 

    % correction
     $K_k \leftarrow R_{k^-} H_k^* [H_k R_{k^-} H_k^* + Q_k^V]^{-1}$  % gain
     $\hat{X}_k \leftarrow \hat{X}_{k^-} + K_k [Y_k - (H_k \hat{X}_{k^-} + h_k)]$ 
     $R_k \leftarrow [I - K_k H_k] R_{k^-}$ 

fin pour

```

ALG. 1: *Filtre de Kalman pour le système linéaire/gaussien (4.1)–(4.2).*

4.4 Système conditionnellement linéaire/gaussien

On considère un modèle conditionnellement linéaire/gaussien

$$X_{k+1} = F_k(Y) X_k + f_k(Y) + G_k(Y) W_k, \quad k \geq 0, \quad (4.14)$$

$$Y_k = H_k(Y) X_k + h_k(Y) + V_k, \quad k \geq 1 \quad (4.15)$$

où $X_0 \sim \mathcal{N}(\bar{X}_0, Q_0)$, $W_k \sim \mathcal{N}(0, Q_k^W)$, $V_k \sim \mathcal{N}(0, Q_k^V)$; X_0 , W_k et V_k sont indépendants. On suppose que $Q_k^V > 0$.

Ici on suppose que les coefficients $F_k(Y)$, $f_k(Y)$, $G_k(Y)$ dépendent de $Y_{1:k}$, et les coefficients $H_k(Y)$, $h_k(Y)$ dépendent de $Y_{1:k-1}$. La covariance Q_k^W (resp. Q_k^V) peut éventuellement dépendre de $Y_{1:k}$ (resp. $Y_{1:k-1}$).

La sortie (X_k, Y_k) du système n'est pas gaussienne, mais conditionnellement à $(Y_k)_{k \geq 1}$ le processus $(X_k)_{k \geq 0}$ est gaussien. Ainsi on a toujours $p(X_k | Y_{1:k}) = \mathcal{N}(\hat{X}_k, R_k)$ et $p(X_k | Y_{1:k-1}) = \mathcal{N}(\hat{X}_{k^-}, R_{k^-})$ avec :

$$\begin{aligned} \hat{X}_k &= \mathbb{E}[X_k | Y_{1:k}], & R_k &= \mathbb{E}[(X_k - \hat{X}_k)(X_k - \hat{X}_k)^* | Y_{1:k}], \\ \hat{X}_{k^-} &= \mathbb{E}[X_k | Y_{1:k-1}], & R_{k^-} &= \mathbb{E}[(X_k - \hat{X}_{k^-})(X_k - \hat{X}_{k^-})^* | Y_{1:k-1}]. \end{aligned}$$

Les matrices de covariance R_{k^-} , R_k dépendent maintenant des observations.

En utilisant les mêmes arguments que ceux de la section précédente on obtient le même type d'algorithme (cf. ALG. 2).

```

% initialisation
 $\hat{X}_0 \leftarrow \bar{X}_0$ 
 $R_0 \leftarrow Q_0$ 

% itérations
pour  $k = 1, 2, 3 \dots$  faire

    % prédiction
     $\hat{X}_{k-} \leftarrow F_{k-1}(Y) \hat{X}_{k-1} + f_{k-1}(Y)$ 
     $R_{k-} \leftarrow F_{k-1}(Y) R_{k-1} F_{k-1}(Y)^* + G_{k-1}(Y) Q_{k-1}^W G_{k-1}(Y)^*$ 

    % correction
     $K_k \leftarrow R_{k-} H_k(Y)^* [H_k(Y) R_{k-} H_k(Y)^* + Q_k^V]^{-1}$  % gain
     $\hat{X}_k \leftarrow \hat{X}_{k-} + K_k [Y_k - (H_k(Y) \hat{X}_{k-} + h_k(Y))]$ 
     $R_k \leftarrow [I - K_k H_k(Y)] R_{k-}$ 

fin pour

```

ALG. 2: *Filtre de Kalman dans le cas du système conditionnellement linéaire/gaussien (4.14)–(4.15).*

4.5 Filtre de Kalman étendu

Il existe de nombreuses approximations adaptations des filtres du type Kalman étendu, voir [22, 2, 42].

Considérons un système non linéaire

$$X_{k+1} = f(X_k) + g(X_k) W_k, \quad (4.16)$$

$$Y_k = h(X_k) + V_k. \quad (4.17)$$

On suppose que $\mathbb{E}[W_k] = 0$, $\text{cov}[W_k] = Q_k^W$, $\mathbb{E}[V_k] = 0$, $\text{cov}[V_k] = Q_k^V$, et que $(W_k)_{k \geq 0}$, $(V_k)_{k \geq 1}$, X_0 sont indépendants.

Le filtre optimal donné à la proposition 3.1.1 ne peut pas se résoudre explicitement comme dans le cas linéaire/gaussien. L'idée du filtre de Kalman étendu est de linéariser le système (4.16)–(4.17) autour de l'estimée courante et d'appliquer la technique du filtre de Kalman.

Pour l'initialisation on pose

$$\hat{X}_0 = \mathbb{E}[X_0], \quad R_0 = \text{cov}(X_0). \quad (4.18)$$

Notons que X_0 n'est pas nécessairement gaussien.

Pour la phase de prédiction, on linéarise l'équation d'état autour de \hat{X}_{k-1} :

$$\begin{aligned} X_k &= f(X_{k-1}) + g(X_{k-1}) W_{k-1} \\ &\simeq f(\hat{X}_{k-1}) + F_{k-1} (X_{k-1} - \hat{X}_{k-1}) + G_{k-1} W_{k-1} \end{aligned}$$

où

$$F_{k-1} \stackrel{\text{déf}}{=} \nabla f(\hat{X}_{k-1}), \quad G_{k-1} \stackrel{\text{déf}}{=} g(\hat{X}_{k-1}).$$

Ainsi

$$\begin{aligned} \hat{X}_{k-} &= \mathbb{E}[X_k | Y_{1:k-1}] \\ &\simeq \mathbb{E}[f(\hat{X}_{k-1}) | Y_{1:k-1}] + F_{k-1} \mathbb{E}[X_{k-1} - \hat{X}_{k-1} | Y_{1:k-1}] \\ &\quad + G_{k-1} \mathbb{E}[W_{k-1} | Y_{1:k-1}] \\ &= f(\hat{X}_{k-1}). \end{aligned}$$

La covariance prédite est donnée par

$$\begin{aligned} R_{k-} &= \mathbb{E}[(X_k - \hat{X}_{k-})(X_k - \hat{X}_{k-})^*] \\ &\simeq F_{k-1}^* R_{k-1} F_{k-1} + G_{k-1} Q_{k-1}^W G_{k-1}^*. \end{aligned}$$

Notons que les matrices de covariances dépendent des observations.

Pour la phase de correction, on linéarise l'équation d'observation autour de l'estimée courante, c'est-à-dire \hat{X}_{k-} , on obtient

$$\begin{aligned} Y_k &= h(X_k) + V_k \\ &\simeq h(\hat{X}_{k-}) + H_k (X_k - \hat{X}_{k-}) + V_k \end{aligned}$$

où $H_k \stackrel{\text{déf}}{=} \nabla h(\hat{X}_{k-})$. En appliquant le filtre de Kalman on obtient

$$\begin{aligned} K_k &= R_{k-} H_k^* [H_k R_{k-} H_k^* + Q_k^V]^{-1}, \\ \hat{X}_k &= \hat{X}_{k-} + K_k [Y_k - h(\hat{X}_{k-})], \\ R_k &= [I - K_k H_k] R_{k-}. \end{aligned}$$

En résumé on obtient l'algorithme 3.

4.6 Filtre de Kalman inodore (Unscented Kalman filter)

Le but ici est de se passer des calculs de gradient qui apparaissent dans le filtre de Kalman étendue. On fait pour cela appel à des formules de quadrature. Cette idée a été introduite par Julier & Uhlmann vers la fin des années 90, voir [31, 32, 34, 33].

```

% initialisation
 $\hat{X}_0 \leftarrow \bar{X}_0$ 
 $R_0 \leftarrow Q_0$ 

% itérations
pour  $k = 1, 2, 3 \dots$  faire

    % prédiction
     $F_{k-1} \leftarrow \nabla f(\hat{X}_{k-1})$ 
     $\hat{X}_{k-} \leftarrow f(\hat{X}_{k-1})$ 
     $R_{k-} \leftarrow F_{k-1} R_{k-1} F_{k-1}^* + G_{k-1} Q_{k-1}^W G_{k-1}^*$ 

    % correction
     $H_k \leftarrow \nabla h(\hat{X}_{k-})$ 
     $K_k \leftarrow R_{k-} H_k^* [H_k R_{k-} H_k^* + Q_k^V]^{-1}$  % gain
     $\hat{X}_k \leftarrow \hat{X}_{k-} + K_k [Y_k - h(\hat{X}_{k-})]$ 
     $R_k \leftarrow [I - K_k H_k] R_{k-}$ 

fin pour

```

ALG. 3: *Filtre de Kalman étendu pour le système non-linéaire (4.16)–(4.17).*

On définit une transformée “inodore” (“unscented transform”) : supposons que l’on dispose d’une formule de quadrature au second ordre $(\omega^i, x^i)_{i=1:N}$ pour X , c’est-à-dire qu’il existe des poids positifs $\omega^{1:N}$ et des points $x^{1:N}$ de \mathbb{R}^n – appelés σ -points – tels que :

$$\sum_{i=1}^N \omega^i x^i = \mathbb{E}X \quad \text{et} \quad \sum_{i=1}^N \omega^i (x^i - \mathbb{E}X)(x^i - \mathbb{E}X)^* = \text{cov}(X) \quad (4.19)$$

alors $(\omega^i, y^i)_{i=1:N}$ avec $y^i = f(x^i)$ sont une formule de quadrature pour $Y = f(X)$, i.e.

$$\sum_{i=1}^N \omega^i y^i = \mathbb{E}Y \quad \text{et} \quad \sum_{i=1}^N \omega^i (y^i - \mathbb{E}Y)(y^i - \mathbb{E}Y)^* = \text{cov}(Y).$$

Attention, ici les poids $\omega^i \in \mathbb{R}$ peuvent être négatif, la seule condition est que $\sum_i \omega^i = 1$.

Le problème est de choisir les σ -points. Les auteurs proposent le choix

suivant :

$$\begin{aligned}
 x^i &\leftarrow \mathbb{E}(X) + [\sqrt{(n+\lambda) \text{cov}(X)}]_i & \omega^i &\leftarrow \frac{1}{2(d+\lambda)} & i = 1 \cdots n \\
 x^{i+d} &\leftarrow \mathbb{E}(X) - [\sqrt{(n+\lambda) \text{cov}(X)}]_i & \omega^{i+n} &\leftarrow \frac{1}{2(n+\lambda)} & i = 1 \cdots n \\
 x^{2n+1} &\leftarrow \mathbb{E}(X) & \omega^{2n+1} &\leftarrow \frac{\lambda}{n+\lambda}
 \end{aligned}$$

où $N = 2n + 1$ et $\lambda \in \mathbb{R}$ est un paramètre d'échelle que l'on peut librement choisir. En pratique, le choix $n + \lambda = 3$ correspondant au cas gaussien semble être judicieux.

L'itération $k - 1 \rightarrow k$ devient :

$$\begin{aligned}
 \sigma_{k-1} &= (\omega_{k-1}^i, x_{k-1}^i) \text{ } \sigma\text{-points pour } \mathcal{N}(\hat{X}_{k-1}, R_{k-1}) \\
 \hat{X}_{k-} &= \sum_{\sigma_{k-1}} \omega_{k-1}^i f_k(x_{k-1}^i) \\
 R_{k-} &= \sum_{\sigma_{k-1}} \omega_{k-1}^i [f_k(x_{k-1}^i) - \hat{X}_{k-}] [f_k(x_{k-1}^i) - \hat{X}_{k-}]^* + Q_k^W \\
 \sigma_{k-} &= (\omega_{k-}^i, x_{k-}^i) \text{ } \sigma\text{-points pour } \mathcal{N}(\hat{X}_{k-}, R_{k-}) \\
 \hat{Y}_{k-} &= \sum_{\sigma_{k-}} \omega_{k-}^i h_k(x_{k-}^i) \\
 S_{k-} &= \sum_{\sigma_{k-}} \omega_{k-}^i [h_k(x_{k-}^i) - \hat{Y}_{k-}] [h_k(x_{k-}^i) - \hat{Y}_{k-}]^* + Q_k^V \\
 U_{k-} &= \sum_{\sigma_{k-}} \omega_{k-}^i [x_{k-}^i - \hat{X}_{k-}] [h_k(x_{k-}^i) - \hat{Y}_{k-}]^* \\
 \hat{X}_k &= \hat{X}_{k-} + U_{k-} [S_{k-}]^{-1} [Y_k - \hat{Y}_{k-}] \\
 R_k &= R_{k-} - U_{k-} [S_{k-}]^{-1} [U_{k-}]^*
 \end{aligned}$$

4.7 Filtre de Kalman d'ensemble

Une des extensions les plus intéressantes du filtre de Kalman est le filtre de Kalman d'ensemble ("ensemble Kalman filter") proposé par [20] dans le cadre de l'assimilation de données. Dans ce cadre la dimension n de l'espace d'état est très grande, trop grande pour stocker les matrices de covariance en mémoire et pour inverser la matrice $H_k P_k - H_k^* + Q_k^V$. L'auteur propose alors une méthode originale pour résoudre ce problème à l'aide de méthodes de quadrature de type Monte Carlo (voir la section §5.1). Voir [23, 47, 13, 21] pour plus de détails.

On considère le système non linéaire :

$$X_{k+1} = f(X_k) + W_k, \quad (4.20)$$

$$Y_k = H X_k + V_k. \quad (4.21)$$

On suppose que $\mathbb{E}[W_k] = 0$, $\text{cov}[W_k] = Q_k^W$, $\mathbb{E}[V_k] = 0$, $\text{cov}[V_k] = Q_k^V$, et que W_k , V_k , X_0 sont indépendants. On suppose également que la loi de la condition initiale X_0 est $\mathcal{N}(\bar{X}_0, P_0)$.

On considère la moyenne et la covariance de la loi conditionnelle de X_k sachant $Y_{1:k}$:

$$\hat{X}_k^a \stackrel{\text{déf}}{=} \mathbb{E}[X_k | Y_{1:k}], \quad R_k^a \stackrel{\text{déf}}{=} \mathbb{E}[(X_k - \hat{X}_k^a)(X_k - \hat{X}_k^a)^* | Y_{1:k}]$$

ainsi que la moyenne et la covariance de la loi conditionnelle de X_k sachant $Y_{1:k-1}$:

$$\hat{X}_k^f \stackrel{\text{déf}}{=} \mathbb{E}[X_k | Y_{1:k-1}], \quad R_k^f \stackrel{\text{déf}}{=} \mathbb{E}[(X_k - \hat{X}_k^f)(X_k - \hat{X}_k^f)^* | Y_{1:k-1}]$$

Les indices "a" (analyse) et "f" (forecast) proviennent des techniques de prévision météorologique : entre deux instants d'observation on effectue une prévision (forecast) qui est ensuite corrigée (analyse) lorsque une observation est disponible. L'idée de la technique d'ensemble est d'approcher les moments précédents par des moments empiriques calculés à partir de N particules :

$$\hat{X}_k^a \simeq \tilde{X}_k^a \stackrel{\text{déf}}{=} \frac{1}{N} \sum_{i=1}^N \xi_k^{a,i}, \quad \hat{X}_k^f \simeq \tilde{X}_k^f \stackrel{\text{déf}}{=} \frac{1}{N} \sum_{i=1}^N \xi_k^{f,i}$$

(idem pour les covariances). Entre deux observations les particules évoluent indépendamment en "singant" l'équation d'état (4.20). L'observation (4.21) est ensuite utilisée afin de "corriger" la position de ces particules. Voir l'algorithme 4.

“The EnKF was designed to resolve two major problems related to the use of the EKF with nonlinear dynamics in large state spaces. The first problem relates to the use of an approximate closure scheme in the EKF, and the other to the huge computational requirements associated with the storage and forward integration of the error covariance matrix.”

```
% initialisation
 $\xi_0^{a,i} \sim \mathcal{N}(\tilde{X}_0, P_0), i = 1 : N$ 

% itérations
pour  $k = 1, 2, 3 \dots$  faire

    % prédiction
     $w_k^i \sim \mathcal{N}(0, Q_k^W), i = 1 : N$ 
     $\xi_k^{f,i} \leftarrow f(\xi_{k-1}^{a,i}) + w_k^i, i = 1 : N$ 
     $\tilde{X}_k^f \leftarrow \frac{1}{N} \sum_{i=1}^N \xi_k^{f,i}$ 
     $\tilde{R}_k^f \leftarrow \frac{1}{N-1} \sum_{i=1}^N [\xi_k^{f,i} - \tilde{X}_k^f] [\xi_k^{f,i} - \tilde{X}_k^f]^*$ 

    % correction
     $v_k^i \sim \mathcal{N}(0, Q_k^V), i = 1 : N$ 
     $\tilde{Q}_k^V \leftarrow \frac{1}{N-1} \sum_{i=1}^N v_k^i (v_k^i)^*$ 
     $K_k \leftarrow \tilde{R}_k^f H^* [H \tilde{R}_k^f H^* + \tilde{Q}_k^V]^{-1}$ 
     $\xi_k^{a,i} \leftarrow \xi_k^{f,i} + K_k [Y_k - H \xi_k^{f,i} - v_k^i], i = 1 : N$ 

fin pour
```

ALG. 4: Filtre de Kalman d'ensemble pour le système non-linéaire (4.20)–(4.21).

5

APPROXIMATION PARTICULAIRE

Dans le cas général les équations du filtre optimal (3.2)–(3.3) n’admettent pas de solution explicite. Dans ce chapitre on se propose d’introduire les bases d’une technique d’approximation dite particulaire. Cette technique fait appel à la méthode de Monte Carlo que nous présentons dans un premier temps.

[3] est un article de survol. [17] fait le point sur les connaissances dans le domaine. [4] est un cours qui traite du sujet .

5.1 Méthode de Monte Carlo

On se contente ici de présenter quelques résultats. On peut trouver sur le réseau des cours de qualité comme celui d’Annie Millet [44] ou de Bernard Delyon [15].

On considère la densité de probabilité $p(x)$ d’une variable aléatoire X à valeurs dans \mathbb{R}^n , il s’agit d’approcher numériquement l’intégrale

$$\mathcal{J} = \int_{\mathbb{R}^n} \varphi(x) p(x) dx \quad (5.1)$$

pour toute fonction $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$ donnée.

L’idée est de donner une *interprétation probabiliste* à l’intégrale \mathcal{J} . En l’occurrence cette intégrale peut s’écrire sous la forme $\mathcal{J} = \mathbb{E}\varphi(X)$. On considère un échantillon de taille N de la loi de densité $p(x)$, i.e. N variables aléa-

toires $\xi^1 \dots \xi^N$ i.i.d. de loi de densité $p(x)$, on note¹

$$\xi^{1:N} \stackrel{\text{iid}}{\sim} p(x).$$

L'intégrale \mathcal{J} est alors approchée par

$$\mathcal{J} \simeq \mathcal{J}^N \stackrel{\text{déf}}{=} \frac{1}{N} \sum_{i=1}^N \varphi(\xi^i). \quad (5.2)$$

D'après la loi des grands nombres, cette approximation est convergente²

$$\mathcal{J}^N \xrightarrow[N \uparrow \infty]{} \mathcal{J} \text{ p.s.} \quad (5.3)$$

La vitesse de convergence est donnée par le théorème de la limite centrale³

$$\frac{\sqrt{N}}{\sigma} (\mathcal{J}^N - \mathcal{J}) \xrightarrow[N \uparrow \infty]{\text{loi}} \mathcal{N}(0, 1) \quad (5.4)$$

où $\sigma^2 = \text{var}(\varphi(X))$. Ce dernier résultat est très important en pratique, il permet par exemple d'écrire

$$\mathcal{J} \simeq \mathcal{J}^N \pm \sigma^N \frac{1.96}{\sqrt{N}} \quad \text{avec probabilité} \simeq 0.95$$

où $(\sigma^N)^2 \stackrel{\text{déf}}{=} \frac{1}{N} \sum_{i=1}^N \varphi^2(\xi^i) - \left(\frac{1}{N} \sum_{i=1}^N \varphi(\xi^i) \right)^2$.

La vitesse de convergence $1/\sqrt{N}$ donnée par (5.4) est faible : un gain en précision de facteur 2 nécessite 4 fois plus de simulations. Un point positif est que cette vitesse ne dépend pas de la dimension n de l'espace. Cette vitesse étant donnée, on peut toutefois améliorer cette méthode en simulant de façon efficace le N -échantillon, c'est le problème de simulation. Par ailleurs on peut se demander s'il est possible de diminuer σ^2 dans (5.4), c'est le problème de réduction de variance.

1. On note $\xi \sim p(x)$ le fait que la variable aléatoire ξ suit la loi de densité $p(x)$ ainsi que le fait que le nombre ξ est simulé selon la loi de densité $p(x)$. Donc $\xi^{1:N} \stackrel{\text{iid}}{\sim} p(x)$ signifie également que $\xi^{1:N}$ sont N nombres indépendamment simulés selon la loi de densité $p(x)$. Enfin $\xi^{1:N} \stackrel{i}{\sim} p_1(x_1) \dots p_N(x_N)$ signifie que $\xi^{1:N}$ sont N variables aléatoires indépendantes telles que $\xi^i \sim p_i(x)$ pour tout i .

2. Convergence presque sûre : $X_N \xrightarrow{\text{p.s.}} X$ signifie que $\mathbb{P}(X_N \rightarrow X) = 1$.

3. Convergence en loi : $X_N \xrightarrow{\text{loi}} \mathcal{N}(0, 1)$ signifie que $\mathbb{E}\varphi(X_N) \rightarrow \mathbb{E}\varphi(X)$ pour toute fonction φ continue, bornée, où $X \sim \mathcal{N}(0, 1)$.

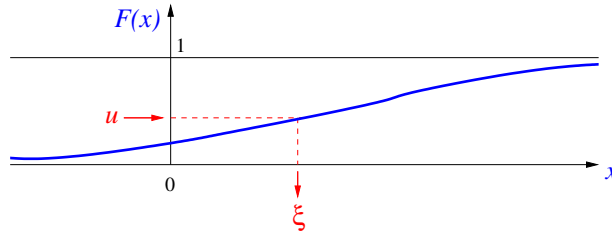


FIGURE 5.1 – Méthode de la fonction de répartition.

5.2 Simulation de variables aléatoires

Comment générer une suite de nombres $\xi^1 \dots \xi^N$, dits pseudo-aléatoires, qui “ressemblent” à un échantillon de taille N de la loi de densité $p(x)$? Ce problème est complexe. On se contente ici de présenter quelques méthodes. Il faut garder à l’esprit qu’en pratique avant d’utiliser une telle méthode il est nécessaire de la soumettre à quelques tests.

Tous les ordinateurs proposent des simulateurs selon la loi uniforme sur un ensemble fini $\{1, \dots, M\}$. Les simulateurs selon la loi uniforme $\mathcal{U}[0, 1]$ sur l’intervalle $[0, 1]$ sont également très répandus (sinon il est simple de “bricoler” un simulateur $\mathcal{U}[0, 1]$ à partir d’un simulateur de loi uniforme sur $\{1, \dots, M\}$).

Changement de variable

À l’aide de changement de variables certains générateurs se déduisent de générateurs de lois uniformes. Il est par exemple évident que le générateur de la loi uniforme $\mathcal{U}[a, b]$ sur $[a, b]$ se déduit de celui de la loi $\mathcal{U}[0, 1]$.

On peut également citer la transformation de Box–Muller : si U_1 et U_2 sont deux réalisations indépendantes de $\mathcal{U}[0, 1]$ alors X_1 et X_2 définis par

$$\begin{aligned} X_1 &\leftarrow \sqrt{-2 \log(U_1)} \cos(2\pi U_2), \\ X_2 &\leftarrow \sqrt{-2 \log(U_1)} \sin(2\pi U_2) \end{aligned}$$

sont deux réalisations indépendantes de $\mathcal{N}(0, 1)$ (cf. Annexe A.2). Ainsi en générant deux à deux des nombres pseudo-aléatoires de la loi uniforme $\mathcal{U}[0, 1]$ on en déduit des nombres pseudo-aléatoires de la loi gaussienne $\mathcal{N}(0, 1)$.

Méthode de l'inverse de la fonction de répartition

On définit la fonction de répartition de la loi de X

$$F(x) = \int_{-\infty}^x p(x') dx' = \mathbb{P}(X \leq x)$$

et son inverse (également appelée fonction quantile)

$$F^{-1}(u) \stackrel{\text{déf}}{=} \inf\{x \in \mathbb{R}; F(x) \geq u\}.$$

On se donne un N -échantillon de la loi uniforme $u^{1:N} \stackrel{\text{iid}}{\sim} \mathcal{U}[0, 1]$ et on pose $\xi^i = F^{-1}(u^i)$ alors $\xi^{1:N} \stackrel{\text{iid}}{\sim} p(x)$ (cf. Fig. 5.1).

```

 $u \sim \mathcal{U}[0, 1]$ 
 $j \leftarrow 1$ 
tant que  $\omega^1 + \dots + \omega^j < u$  faire
   $j \leftarrow j + 1$ 
fin tant que
 $\xi \leftarrow x^j$ 
rendre  $\xi$ 

```

ALG. 5: Générateur de la loi discrète $p(x) = \sum_{i=1}^N \omega^i \delta_{x^i}(x)$ par la méthode de l'inverse de la fonction de répartition.

Cette méthode s'applique par exemple à la loi exponentielle ou à la loi de Weibull. On peut également l'appliquer à la loi discrète : $\mathbb{P}(X = x^i) = \omega^i$ avec $\omega^i \geq 0$ et $\sum_{i=1}^N \omega^i = 1$ i.e. $p(x) = \sum_{i=1}^N \omega^i \delta_{x^i}(x)$. On obtient alors l'algorithme 7 page 64.

Méthode d'acceptation/rejet

```

répéter
   $u \sim \mathcal{U}[0, 1]$ 
   $\xi \sim \tilde{p}(x)$ 
jusqu'à  $u \leq p(\xi) / \kappa \tilde{p}(\xi)$ 
rendre  $\xi$ 

```

ALG. 6: Générateur d'une loi $p(x)$ par méthode d'acceptation/rejet.

Supposons maintenant que l'on s'intéresse à une loi de densité $p(x)$: d'une part on ne sait pas simuler directement selon elle, d'autre part on ne sait pas

inverser sa fonction de répartition. En revanche on suppose que l'expression de la densité $p(x)$ est connue et que l'on sait simuler selon une autre densité $\tilde{p}(x)$ telle que $\frac{p(x)}{\tilde{p}(x)} \leq \kappa < \infty$.

La méthode de rejet donne l'algorithme 6.

D'autres méthodes

Il existe de nombreuses autres méthodes itératives de génération de nombres pseudo-aléatoires selon une loi $p(x)$, notamment les méthodes MCMC (Monte Carlo par Chaîne de Markov), les notes des cours d'Annie Millet [44] et de Bernard Delyon [15] détaillent ces approches.

5.3 Échantillonnage d'importance

Avant d'introduire l'approximation particulière du filtre non linéaire, essayons de comprendre ce que l'on entend par "approximation particulière" d'une loi $p(x)$.

L'approximation (5.2) de (5.1) est valide pour toute fonction φ . Il s'agit donc d'une approximation de la loi $p(x)$, en effet

$$p(x) \simeq p^N(x) \stackrel{\text{déf}}{=} \frac{1}{N} \sum_{i=1}^N \delta_{\xi^i}(x) \quad \text{où} \quad \xi^{1:N} \stackrel{\text{iid}}{\sim} p(x). \quad (5.5)$$

Ainsi $p^N(x)$ est une *approximation particulière* de la loi $p(x)$. Plus généralement, une approximation particulière sera de la forme

$$p^N(x) = \sum_{i=1}^N \omega^i \delta_{\xi^i}(x). \quad (5.6)$$

avec $\omega^i \geq 0$ et $\sum_{i=1}^N \omega^i = 1$. L'avantage de ce genre d'approximation réside dans le fait que les calculs d'intégrales sous cette loi sont très simples, en effet

$$\mathbb{E}\varphi(X) = \int \varphi(x) p(x) dx \simeq \int \varphi(x) p^N(x) dx = \sum_{i=1}^N \omega^i \varphi(\xi^i), \quad (5.7)$$

$$\mathbb{P}(X \in A) = \int_A p(x) dx \simeq \int_A p^N(x) dx = \frac{\#\{i = 1 : N; \xi^i \in A\}}{N}. \quad (5.8)$$

On cherche une approximation de la forme (5.6) d'une loi $p(x)$ donnée. Si l'on sait échantillonner selon $p(x)$ on peut utiliser l'approximation (5.5). Si

on ne sait pas (ou ne veut pas) échantillonner selon $p(x)$, on peut faire appel à une méthode dite d'échantillonnage d'importance qui s'appuie sur la remarque suivante : supposons que l'on dispose d'une autre loi $\tilde{p}(x)$ telle que

$$p(x) = 0 \Rightarrow \tilde{p}(x) = 0 \quad (5.9)$$

(i.e. $p(x)$ est absolument continue par rapport à $\tilde{p}(x)$), alors

$$\mathbb{E}\varphi(X) = \int \varphi(x) p(x) dx = \int \varphi(x) \frac{p(x)}{\tilde{p}(x)} \tilde{p}(x) dx.$$

Si on dispose d'un N -échantillon $\xi^{1:N}$ de la loi $\tilde{p}(x)$ alors

$$\mathbb{E}\varphi(X) \simeq \sum_{i=1}^N \omega^i \varphi(\xi^i) \quad \text{où} \quad \begin{cases} \xi^{1:N} \stackrel{\text{iid}}{\sim} \tilde{p}(x), \\ \omega^i \stackrel{\text{déf}}{=} p(\xi^i) / \tilde{p}(\xi^i). \end{cases} \quad (5.10)$$

Ce n'est pas une méthode d'acceptation/rejet : ici toutes les simulations sont utilisées et on n'a pas à majorer $p(x)/\tilde{p}(x)$. Toutefois si $p(x)/\tilde{p}(x)$ n'est pas borné, la valeur des poids ω^i dans (5.10) risquent alors de beaucoup varier et les particules correspondant à un poids faible contribuent peu à l'approximation de $\mathbb{E}\varphi(X)$.

L'utilisation pratique de (5.10) est importante : supposons que l'on ne sache pas simuler selon $p(x)$ (la loi cible), ou que cela soit difficile, mais que l'on sache simuler selon une loi $\tilde{p}(x)$ (la loi d'importance) telle que (5.9). Il suffit alors de calculer un échantillon de $\tilde{p}(x)$ et de calculer les poids d'importance ω^i définis par (5.10).

Une question importante, non traitée ici, est de savoir comment choisir la loi d'importance.

Application

On souhaite par exemple approcher $E[\varphi(X)|Y]$ alors que l'on ne sait/veut pas échantillonner selon la loi conditionnelle $p(X|Y)$ mais selon $p(X)$. En faisant appel à la formule de Bayes, on a

$$\mathbb{E}[\varphi(X)|Y] = \int \varphi(X) p(X|Y) dX = \int \varphi(X) \frac{p(Y|X)}{p(Y)} p(X) dX$$

alors

$$\mathbb{E}[\varphi(X)|Y] \simeq \sum_{i=1}^N \omega^i \varphi(\xi^i) \quad \text{où} \quad \xi^{1:N} \stackrel{\text{iid}}{\sim} p(X), \quad \omega^i \propto p(Y|X = \xi^i).$$



On peut relire ce résultat de la façon suivante : on dispose, avant toute

observation, d'un N -échantillon $\xi^{1:N}$ de la loi a priori $p(X)$. Cet échantillon nous permet d'écrire une approximation de la loi a priori

$$p(X) \simeq p^N(X) \stackrel{\text{déf}}{=} \frac{1}{N} \sum_{i=1}^N \delta_{\xi^i}(X) \quad \text{où} \quad \xi^{1:N} \stackrel{\text{iid}}{\sim} p(X).$$

Après observation de Y , on peut calculer la vraisemblance $p(Y|X = \xi^i)$ associée à chacune des positions ξ^i . Ces vraisemblances permettent de *repondérer* les particules ξ^i , on obtient une approximation de la loi a posteriori

$$p(X|Y) \simeq p^N(X|Y) \stackrel{\text{déf}}{=} \sum_{i=1}^N \omega^i \delta_{\xi^i}(X) \quad \text{où} \quad \omega^i \stackrel{\text{déf}}{=} \frac{p(Y|X = \xi^i)}{\sum_{j=1}^N p(Y|X = \xi^j)}.$$

5.4 Filtre SIR (bootstrap)

L'idée de base du filtrage particulaire consiste donc à déterminer des approximations du filtre et du filtre prédit sous la forme

$$p(X_k|Y_{1:k-1}) \simeq p^N(X_k|Y_{1:k-1}) = \sum_{i=1}^N \omega_{k-}^i \delta_{\xi_{k-}^i}(X_k),$$

$$p(X_k|Y_{1:k}) \simeq p^N(X_k|Y_{1:k}) = \sum_{i=1}^N \omega_k^i \delta_{\xi_k^i}(X_k).$$

à chaque instant k , il s'agit donc de déterminer un nombre fini de paramètres $(\omega_{k-}^{1:N}, \xi_{k-}^{1:N})$ et $(\omega_k^{1:N}, \xi_k^{1:N})$.

Étape de prédiction

Supposons que

$$p(X_{k-1}|Y_{1:k-1}) = \sum_{i=1}^N \omega_{k-1}^i \delta_{\xi_{k-1}^i}(X_{k-1}).$$

On détermine $p(X_k|Y_{1:k-1})$ à l'aide l'équation (3.2)

$$\begin{aligned} p(X_k|Y_{1:k-1}) &= \int p(X_k|X_{k-1}) p(X_{k-1}|Y_{1:k-1}) dX_{k-1} \\ &= \sum_{i=1}^N \omega_{k-1}^i \int p(X_k|X_{k-1}) \delta_{\xi_{k-1}^i}(X_{k-1}) dX_{k-1} \\ &= \sum_{i=1}^N \omega_{k-1}^i p(X_k|X_{k-1} = \xi_{k-1}^i). \end{aligned}$$

On obtient donc un mélange des lois $p(X_k|X_{k-1}=\xi_{k-1}^i)$ (qui n'est pas sous forme particulière). Pour obtenir une approximation de type particulière on peut échantillonner selon cette loi. On choisit une autre possibilité qui consiste à utiliser

$$p(X_k|Y_{1:k-1}) = \sum_{i=1}^N \omega_{k-1}^i \delta_{\xi_{k-1}^i}(X_k) \quad \text{où} \quad \xi_{k-1}^i \sim p(X_k|X_{k-1}=\xi_{k-1}^i).$$

Étape de correction

Supposons maintenant que

$$p(X_k|Y_{1:k-1}) = \sum_{i=1}^N \omega_{k-1}^i \delta_{\xi_{k-1}^i}(X_k).$$

On calcule $p(X_k|Y_{1:k})$ à l'aide de l'équation de correction (3.3)

$$\begin{aligned} p(X_k|Y_{1:k}) &= \frac{p(Y_k|X_k) p(X_k|Y_{1:k-1})}{\int p(Y_k|X_k) p(X_k|Y_{1:k-1}) dX_k} \\ &= \sum_{i=1}^N \frac{\omega_{k-1}^i p(Y_k|X_k)}{\sum_{j=1}^N \int \omega_{k-1}^j p(Y_k|X_k) \delta_{\xi_{k-1}^j}(X_k) dX_k} \delta_{\xi_{k-1}^i}(X_k) \\ &= \sum_{i=1}^N \frac{\omega_{k-1}^i p(Y_k|X_k=\xi_{k-1}^i)}{\sum_{j=1}^N \omega_{k-1}^j p(Y_k|X_k=\xi_{k-1}^j)} \delta_{\xi_{k-1}^i}(X_k) \end{aligned}$$

ainsi

$$p(X_k|Y_{1:k}) = \sum_{i=1}^N \omega_k^i \delta_{\xi_k^i}(X_k)$$

avec

$$\omega_k^i = \frac{\omega_{k-1}^i p(Y_k|X_k=\xi_{k-1}^i)}{\sum_{j=1}^N \omega_{k-1}^j p(Y_k|X_k=\xi_{k-1}^j)}.$$

L'étape de correction se fait donc de façon explicite.

Un premier (mauvais) filtre

En résumé, on obtient l'approximation suivante :

— prédiction

$$p^N(X_k|Y_{1:k-1}) = \sum_{i=1}^N \omega_{k-1}^i \delta_{\xi_{k-1}^i}(X_k) \quad \text{où} \quad \begin{cases} \omega_{k-1}^i = \omega_{k-2}^i \\ \xi_{k-1}^i \sim p^N(X_k|X_{k-1}=\xi_{k-2}^i) \end{cases} \quad (5.11)$$

— correction

$$p^N(\mathbf{X}_k | \mathbf{Y}_{1:k}) = \sum_{i=1}^N \omega_k^i \delta_{\xi_k^i}(\mathbf{X}_k) \quad \text{où} \quad \begin{cases} \omega_k^i = \frac{\omega_{k-}^i p^N(\mathbf{Y}_k | \mathbf{X}_k = \xi_{k-}^i)}{\sum_{j=1}^N \omega_{k-}^j p^N(\mathbf{Y}_k | \mathbf{X}_k = \xi_{k-}^j)} \\ \xi_k^i = \xi_{k-}^i \end{cases} \quad (5.12)$$

Dégénérescence des poids

L'algorithme (5.11)–(5.12) présente un grave défaut : en quelques itérations k , presque tous les poids ω_k^i sont nuls. Idéalement les poids doivent tous rester proches de $\frac{1}{N}$, i.e. les particules sont d'égale importance dans l'approximation.

On peut considérer le critère suivant :

$$N_k^{\text{eff}} = \frac{1}{\sum_{i=1}^N (\omega_k^i)^2} \in [1, N] \quad (5.13)$$

qui représente le nombre efficace de particules. Lorsque N_k^{eff} est proche de N alors les particules sont d'égale importance. Il y a dégénérescence des poids lorsque N_k^{eff} est proche de 1, c'est le cas lorsque l'on utilise l'algorithme (5.11)–(5.12). Dans la section suivante on présente une méthode destinée à se prémunir de cette dégénérescence en “redistribuant” les particules.

Il existe également un critère entropique

$$\mathcal{H}_k = - \sum_{i=1}^N \omega_k^i \log(\omega_k^i) \in [0, \log(N)]$$

avec la convention $0 \log 0 = 0$. Bien que ce critère soit plus fondé, N_k^{eff} lui sera préféré car beaucoup moins coûteux à calculer.

Rééchantillonnage

Si N^{eff} est petit alors la plupart des particules a un poids proche de 0 et a donc une contribution négligeable dans l'approximation. L'idée est donc de “défavoriser” ces particules au bénéfice des particules “importantes” qui ont un poids non négligeable. On “favorise” ces dernières en les dupliquant au détriment de celles que l'on souhaite “défavoriser”. Dans l'approche utilisée ici le nombre de particules reste constant. Il s'agit donc d'une étape de *rééchantillonnage* ou de *redistribution* des particules.

Une première possibilité consiste à utiliser un rééchantillonnage multinomial. On tire N particules parmi $\{\xi_k^1, \dots, \xi_k^N\}$ avec les probabilités $\{\omega_k^1, \dots, \omega_k^N\}$:

on obtient ainsi j_1 fois la particule ξ_k^1 , j_2 fois la particule ξ_k^2 , ainsi de suite. Le vecteur (j_1, \dots, j_N) suit une loi multinomiale $\mathcal{M}(N, \omega_k^1, \dots, \omega_k^N)$. Cela revient à simuler un N -échantillon tiré selon la loi discrète

$$\sum_{i=1}^N \omega_k^i \delta_{\xi_k^i}(\mathbf{X})$$

i.e. on choisit au hasard des points de $\{\xi_k^1, \dots, \xi_k^N\}$ selon la loi $\{\omega_k^1, \dots, \omega_k^N\}$, on obtient de nouveaux points de poids $\frac{1}{N}$

$$\xi_k^{1:N} \leftarrow \text{resample}(\xi_k^{1:N}, \omega_k^{1:N})$$

où `resample` est défini par l'algorithme 7 (qui consiste à utiliser N fois l'algorithme 5). On verra au § 5.5 que cette procédure est trop lente. Il faudra en pratique faire appel à des méthodes plus rapides.

```

procédure resample( $\omega^{(1:N)}, x^{(1:N)}$ )
  pour  $i = 1 : N$  faire
     $u \sim \mathcal{U}[0, 1]$ 
     $j \leftarrow 1$ 
    tant que  $\omega^1 + \dots + \omega^j < u$  faire
       $j \leftarrow j + 1$ 
    fin tant que
     $\xi^i \leftarrow x^j$ 
  fin pour
  rendre  $\xi^{1:N}$ 

```

ALG. 7: *Redistribution par échantillonnage multinomial (cf. ALG. 5) : produit un N -échantillon $\xi^{1:N}$ de loi discrète $(\omega^i, x^i)_{i=1:N}$.*

Filtre SIR (bootstrap)

On obtient un des premiers algorithmes proposé en 93 par Gordon, Salmond, Smith [24]. Ce filtre dit bootstrap est également connu sous le nom de filtre SIR (sampling importance resampling).

Il est possible de redistribuer les particules

- lorsque par exemple $N^{\text{eff}} \leq 0.75$ (cf. ALG. 8),
- systématiquement, i.e. à chaque itération (cf. ALG. 9),
- toutes les M itérations.

Ces algorithmes sont également de type génétique : une population de particules est soumise à un mécanisme de *mutation*, correspondant à l'étape de prédiction, puis à un mécanisme de *sélection*, correspondant à l'étape de correction.


```

 $\xi^{1:N} \stackrel{\text{iid}}{\sim} p(X_0)$ 
 $\omega^{1:N} \leftarrow 1/N$ 
pour  $k = 1, 2, 3 \dots$  faire
   $\tilde{\xi}^i \sim p(X_k | X_{k-1} = \xi^i)$  pour  $i = 1 : N$  % mutation
   $\tilde{\omega}^i \leftarrow \omega^i p(Y_k | X_k = \tilde{\xi}^i)$  pour  $i = 1 : N$  % vraisemblance
   $\tilde{\omega}^i \leftarrow \tilde{\omega}^i / \text{sum}(\tilde{\omega}^{1:N})$  pour  $i = 1 : N$  % normalisation
   $N^{\text{eff}} \leftarrow (\sum_{i=1}^N (\tilde{\omega}^i)^2)^{-1}$ 
  si  $N^{\text{eff}}/N \leq 0.75$  alors
     $\xi^{1:N} \leftarrow \text{resample}(\tilde{\omega}^{1:N}, \tilde{\xi}^{1:N})$  % sélection
     $\omega^{1:N} \leftarrow 1/N$ 
  sinon
     $\xi^{1:N} \leftarrow \tilde{\xi}^{1:N}$ 
     $\omega^{1:N} \leftarrow \tilde{\omega}^{1:N}$ 
  fin si
  sortie  $(\xi^{1:N}, \omega^{1:N})$ 
fin pour

```

ALG. 8: Filtre SIR (sampling important resampling) avec test sur le nombre efficace de particules.

```

 $\xi^{1:N} \stackrel{\text{iid}}{\sim} p(X_0)$  % initialisation
pour  $k = 1, 2, 3 \dots$  faire
   $\xi^i \sim p(X_k | X_{k-1} = \xi^i)$  pour  $i = 1 : N$  % mutation
   $\omega^i \leftarrow p(Y_k | X_k = \xi^i)$  pour  $i = 1 : N$  % vraisemblance
   $\omega^i \leftarrow \omega^i / \text{sum}(\omega^{1:N})$  pour  $i = 1 : N$  % normalisation
   $\xi^{1:N} \leftarrow \text{resample}(\omega^{1:N}, \xi^{1:N})$  % sélection
  sortie  $\xi^{1:N}$ 
fin pour

```

ALG. 9: Filtre SIR (sampling important resampling) avec rééchantillonnage systématique. On peut également rééchantillonner toutes les M itérations.

5.5 Redistribution

En pratique la procédure $\text{resample}(\omega^{1:N}, \xi^{1:N})$ de l'algorithme 7) est trop lente. Elle doit être repensée.

Il s'agit donc de proposer une procédure

$$\xi^{1:N} \leftarrow \text{resample}(\omega^{1:N}, \xi^{1:N})$$

qui permet de dupliquer les particules à fort poids au détriment de celles à faible poids.

Redistribution multinomiale

Reprenons l'algorithme 7 de redistribution par échantillonnage multinomiale (cf. FIG. 5.2). La complexité de cet algorithme est en $O(N \log(N))$ (quelques rappels sur la complexité algorithmique sont donnés au § A.3).

On peut améliorer cette procédure en triant au préalable les poids $\omega^{1:N}$ par ordre décroissant : $\omega^{s[1]} > \omega^{s[2]} > \dots > \omega^{s[N]}$ (cf. FIG. 5.3). Cet algorithme est encore en $O(N \log(N))$, il est néanmoins plus rapide car il fait appel à moins de tests.

On peut à nouveau améliorer cette procédure en faisant appel aux statistiques d'ordre (cf. FIG. 5.3). On obtient alors un algorithme en $O(N)$ à condition de savoir simuler efficacement des statistiques d'ordre.

```

procédure  $\text{resample}(\omega^{(1:N)}, x^{(1:N)})$ 
   $u^1 < u^2 < \dots < u^N \stackrel{\text{iid}}{\sim} \mathcal{U}[0, 1]$ 
   $(\omega^{1:N}, s[1:N]) \leftarrow \text{sort}(\omega^{1:N})$ 
   $j \leftarrow 1$ 
  pour  $i = 1 : N$  faire
    tant que  $\omega^1 + \dots + \omega^j < u$  faire
       $j \leftarrow j + 1$ 
    fin tant que
     $\xi^i \leftarrow x^{s[j]}$ 
  fin pour
  rendre  $\xi^{1:N}$ 

```

ALG. 10: *Redistribution par échantillonnage multinomiale avec statistique d'ordre.*

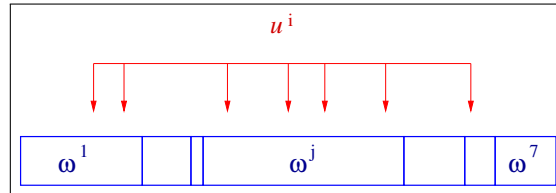


FIGURE 5.2 – *Rééchantillonnage multinomial* : les nombres $u^{i:N} \stackrel{\text{iid}}{\sim} \mathcal{U}[0,1]$ “tombent” dans la partition de $[0,1]$ en intervalles de largeur ω^i donnant l’indice des particules redistribuées. Cette procédure est en $O(N \log(N))$.

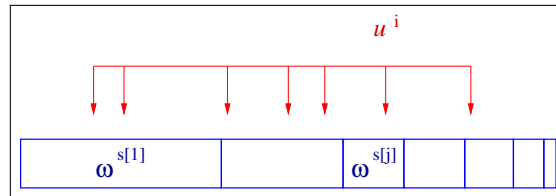


FIGURE 5.3 – *Première amélioration* : on trie les poids $\omega^{1:N}$ par ordre décroissant, l’algorithme est toujours en $O(N \log(N))$ mais il est plus rapide car il nécessite moins de tests.

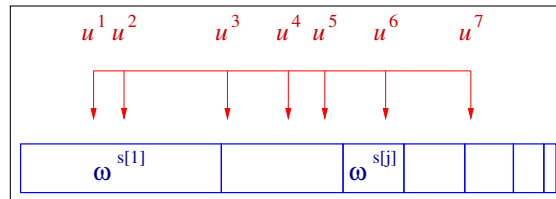
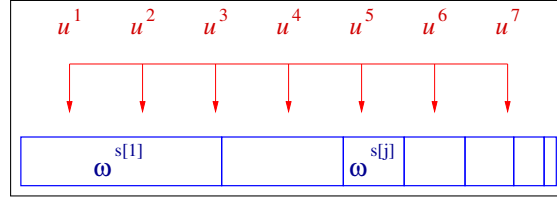


FIGURE 5.4 – *Deuxième amélioration* : on fait appel à des statistiques d’ordre $u^1 < u^2 < \dots < u^N \stackrel{\text{iid}}{\sim} \mathcal{U}[0,1]$. l’algorithme est toujours en $O(N)$ à condition de savoir simuler efficacement des statistiques d’ordre.

FIGURE 5.5 – *Redistribution de Kitagawa, en $O(N)$.*

Redistribution de Kitagawa

Tous les algorithmes précédents seront toujours coûteux dans la mesure où ils font appel à N nombres pseudo-aléatoires. On peut imaginer de faire moins appel à ces générateurs. À la limite, au lieu de simuler des statistiques d'ordre, on simule uniquement u^1 et assigne de façon déterministe les $N - 1$ autres u^i de la manière suivante

$$u_1 \sim \mathcal{U}[0, \frac{1}{N}] \quad u^i = u^1 + \frac{i}{N}, \quad i = 2 : N.$$

Cet algorithme est rapide, en $O(N)$, mais ne marche pas toujours!

Redistribution des résidus

De façon intermédiaire : on utilise l'idée de Kitagawa pour générer un certain nombre de particules puis on fait appel à la méthode de redistribution multinomiale pour déterminer les particules restantes.

Dans un premier temps, pour chaque i , on sélectionne la particule ξ^i de façon déterministe $\lfloor \omega^i / N \rfloor$ fois ATTENTION c'est $\lfloor \omega^i \times N \rfloor$ (cf. FIG. 5.6). Dans un deuxième temps, il reste donc

$$\tilde{N} \leftarrow \sum_{i=1}^N \lfloor \frac{\omega^i}{N} \rfloor$$

particules à sélectionner. On applique la méthode de redistribution multinomiale (cf. ALG. 7) à la loi de probabilité correspondant aux poids restants

$$\tilde{\omega}^i \leftarrow \omega^i - N \lfloor \frac{\omega^i}{N} \rfloor \quad \text{et} \quad \tilde{\omega}^i \leftarrow \tilde{\omega}^i / \sum_{j=1}^N \tilde{\omega}^j.$$

Pour cela on utilise la méthode de redistribution multinomiale (cf. ALG. 7).

Cette approche n'a d'intérêt que s'il ne reste que peu de particules \tilde{N} à générer dans la deuxième étape. C'est le cas lorsque $N^{\text{eff}} = 1 / \sum_{i=1}^N (\omega^i)^2$ est petit et c'est justement dans ce cas qu'il est nécessaire de redistribuer.

Cet algorithme, notamment proposé par Liu & Chen [38], donne de bons résultats en pratique. Son implémentation `matlab` est donnée à l'annexe B (cf. ALG. 13).

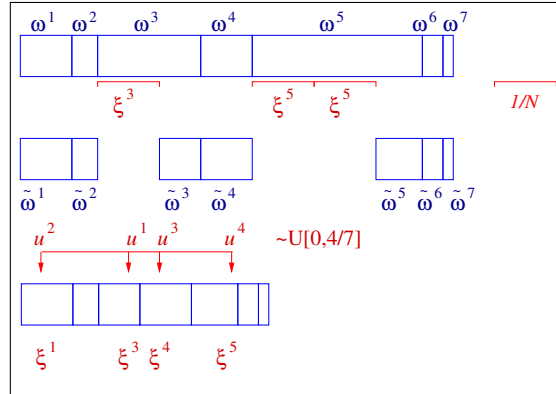


FIGURE 5.6 – *Redistribution des résidus : dans un premier temps on sélectionne la particule ξ^i de façon déterministe $\lfloor \omega^i / N \rfloor$ fois (pour tout i). Les poids restants $\tilde{\omega}^i \leftarrow \omega^i - N \lfloor \omega^i / N \rfloor$ sont normalisé pour en faire une loi de probabilité, les particules manquantes sont tirés selon cette loi à l'aide d'une méthode de redistribution multinomiale (cf. ALG. 7).*

5.6 En pratique

Pour mettre en œuvre cette méthode de filtrage particulière il est nécessaire de savoir

- (i) simuler selon $p(X_0)$,
- (ii) simuler selon $p(X_{k+1}|X_k = \xi)$ quelque soit ξ ,
- (iii) calculer la vraisemblance $p(Y_k|X_k)$ pour tout X_k (à Y_k fixé)

et de disposer d'une routine de redistribution.

Cet algorithme présente de nombreuses variantes et extensions :

- Il est possible de facilement traiter des contraintes de la forme $X_k \in D_k$ (voir ci-dessous).
- On peut tirer parti du fait que certaines composantes du vecteur d'état sont solution d'un système linéaire/gaussien lorsque les autres composantes sont fixées. On obtient alors une résolution hybride particulière/filtre de Kalman. Cette technique est appelée "rao-blackwellisation".
- On peut considérer des modèles à espaces d'état hybrides (continus/discrets/ finis).
- Si on ne sait pas, ou ne veut pas, échantillonner selon $p(X_{k+1}|X_k)$ on peut échantillonner selon une loi d'importance $q(X_{k+1})$, il faudra alors

modifier la fonction de vraisemblance par $q(Y_k|X_k)$ (cf. échantillonnage d'importance § 5.3).

Prise en compte des contraintes

Supposons que l'on sache qu'à chaque instant k l'état est soumis à la contrainte $X_k \in D_k$. La prise en compte de cette contrainte dans le filtre SIR est très simple (cf. ALG. 11).

Cette idée de contrainte s'étend à des cas plus sophistiqués, voir l'exemple § 6.3.

```

 $\xi^{1:N} \stackrel{\text{iid}}{\sim} p(X_0)$  % initialisation
pour  $k = 1, 2, 3 \dots$  faire
   $\xi^i \sim p(X_k|X_{k-1} = \xi^i)$  pour  $i = 1 : N$  % mutation
   $\omega^i \leftarrow \mathbf{1}_{D_k}(\xi^i) p(Y_k|X_k = \xi^i)$  pour  $i = 1 : N$  % vraisem-
  blance
   $\omega^i \leftarrow \omega^i / \text{sum}(\omega^{1:N})$  pour  $i = 1 : N$  % normalisation
   $\xi^{1:N} \leftarrow \text{resample}(\omega^{1:N}, \xi^{1:N})$  % sélection
  sortie  $\xi^{1:N}$ 
fin pour

```

ALG. 11: *Filtre SIR (sampling important resampling) avec rééchantillonnage systématique et prise en compte d'une contrainte.*

Lorsque le filtre perd la piste...

Un seul problème peut survenir mais il est de taille. Imaginons qu'à un instant k , toutes les vraisemblances locales sont nulles, i.e.

$$p(Y_k|X_k = \xi^i) = 0 \quad \forall i = 1 \dots N$$

i.e. les particules ne correspondent plus du tout à l'observation. Dans ce cas les algorithmes 8 et 9 sont en échec : il est en effet impossible de normaliser les poids. Il est alors nécessaire de réinitialiser le filtre : supposons qu'à tous les instants on dispose d'une information, même grossière, sur la position de X_k sous forme d'une loi $\tilde{p}(X_k)$. Lorsque les vraisemblances locales sont toutes nulles on peut échantillonner selon $\tilde{p}(X_k)$.

Un autre idée pour éviter que les particules se perdent est d'affiner l'étape de prédiction, par exemple simuler en tenant compte de X_k et Y_{k+1} .

```

 $\xi^{1:N} \stackrel{\text{iid}}{\sim} p(X_0)$  % initialisation
pour  $k = 1, 2, 3 \dots$  faire
   $\xi^i \sim p(X_k | X_{k-1} = \xi^i)$  pour  $i = 1 : N$  % mutation
   $\omega^i \leftarrow p(Y_k | X_k = \xi^i)$  pour  $i = 1 : N$  % vraisemblance
  si  $\text{sum}(\omega^{1:N}) \neq 0$  alors
     $\omega^i \leftarrow \omega^i / \text{sum}(\omega^{1:N})$  pour  $i = 1 : N$  % normalisation
     $\xi^{1:N} \leftarrow \text{resample}(\omega^{1:N}, \xi^{1:N})$  % sélection
  sinon
     $\xi^{1:N} \stackrel{\text{iid}}{\sim} \tilde{p}(X_k)$  % reset
  fin si
  sortie  $\xi^{1:N}$ 
fin pour

```

ALG. 12: *Filtre SIR (sampling important resampling) avec redistribution systématique et test de réinitialisation.*

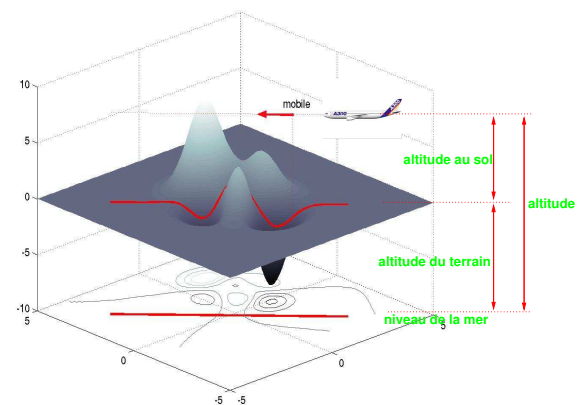
6

APPLICATIONS DU FILTRAGE PARTICULAIRE

Un des premiers domaines d'application du filtrage particulaire a été la poursuite de mobiles, voir [26] et [51]. Un des domaines les plus récents est la robotique, voir [56]. On présente ici quelques exemples permettant d'illustrer les capacités du filtrage particulaire.

6.1 Modèle numérique de terrain

En se déplaçant, un avion mesure l'altitude du terrain au point situé à sa verticale. Ces mesures d'altitude sont bruitées. La trajectoire de l'avion dans le plan est également connue modulo des erreurs dues aux différentes perturbations. Enfin, on dispose d'une carte numérique de la zone géographique donnant l'altitude au sol de chaque pixel (cf. FIG. 6.1). On cherche à déterminer la position en (x, y) de l'avion sur la carte. Il s'agit d'une version (très) simplifiée de techniques de recalage altimétrique de centrales de navigation inertielle par filtrage particulaire, voir [45], [37], [14].



Équation d'état

Pour le déplacement de l'avion on utilise un modèle à cap et vitesse "bruits blancs". Le vecteur d'état

FIGURE 6.1 – En se déplaçant l'avion relève l'altitude du terrain situé à sa verticale, il obtient ainsi un "profile" qu'il s'agit de positionner sur une carte numérique.

est

$$X_k = \begin{cases} r_k^1 & \text{coordonnée en } x^1, \\ r_k^2 & \text{coordonnée en } x^2, \\ v_k & \text{vitesse,} \\ c_k & \text{cap.} \end{cases}$$

L'équation d'état est

$$\begin{cases} r_k^1 = r_{k-1}^1 + \Delta t v_{k-1} \cos(c_{k-1}), \\ r_k^2 = r_{k-1}^2 + \Delta t v_{k-1} \sin(c_{k-1}), \\ v_k = v_{k-1} + \Delta t \sigma_v W_{k-1}^1, \\ c_k = c_{k-1} + \Delta t \sigma_c W_{k-1}^2 \end{cases} \quad (6.1)$$

avec $\Delta t = 1$. On suppose que W_k^1 et W_k^2 sont des bruits blancs gaussiens $\mathcal{N}(0, 1)$ indépendants.

Équation d'observation

L'équation d'observation est

$$Y_k = h(X_k) + \sigma V_k$$

où $h(X_k)$ est l'altitude du terrain à la position (r_k^1, r_k^2) . En fait, on associe au point (r_k^1, r_k^2) les coordonnées du pixel sur la carte numérique, $h(X_k)$ est alors le "niveau de gris" (l'altitude) de ce pixel.

Résultats

Comme illustré dans la figure 6.2, le filtre "utilise" le relief pour sélectionner les particules. Le plus souvent les particules se concentrent après quelques itérations sur la bonne position, comme dans la figure, mais il arrive parfois que le filtre perde la piste du mobile, il est alors nécessaire de réinitialiser le filtre (cf. ALG. 12 page 71).

6.2 Géolocalisation par mesure d'atténuation

Il s'agit de positionner un mobile, par exemple un piéton, munit d'un téléphone portable dans un réseau cellulaire urbain. À chaque instant, le mobile émet un signal reçu par plusieurs stations de base (cf. FIG. 6.3). On dispose de différents types de mesures : les angles d'arrivée du signal, les instants d'arrivée du signal ou bien la puissance du signal reçu. C'est ce dernier type de mesures que nous considérons ici. Voir [1], [25].

Carte d'atténuation

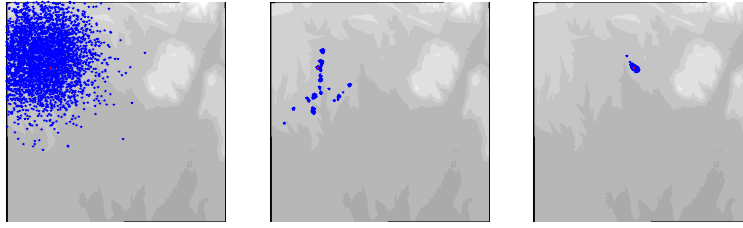


FIGURE 6.2 – Coordonnées x^1 et x^2 de la position réelle (point rouge) et des particules (points bleus) à l'initialisation et à deux autres instants. À l'instant intermédiaire le filtre utilise le relief pour commencer à sélectionner les particules. Les ambiguïtés sont levées après quelques itérations.

La station de base reçoit le signal émis par le mobile, on peut donc en mesurer l'atténuation. L'idée est de mettre en correspondance cette puissance mesurée avec une carte d'atténuation de référence. Cette carte associe à chaque position de la zone géographique la puissance du signal d'un mobile supposé être dans cette position. Il s'agit d'une carte numérique : la zone géographique est divisée en petites cellules, correspondant aux pixels de la carte, une atténuation de référence est associée à chacun de ces cellules. Ces cartes peuvent être établies de façon empirique à l'aide de mesures de terrain, elles peuvent être également simulées.

Ici on simule une telle carte dans le cas d'une géométrie simple de type "manhattan" (cf. FIG. 6.4 et FIG. 6.7). Dans ce cas on dispose de formules explicites pour calculer cette atténuation.

La transmission du signal en zone urbaine est essentiellement conditionnée par la géométrie des rues et des bâtiments. Si le mobile est dans la ligne de vue du mobile (cf. FIG. 6.5), la puissance reçue se calcule en fonction de la puissance du signal transmis de la façon suivante

$$P_{\text{reçu}} = P_{\text{transmis}} - 10\alpha \log\left(\frac{d_1}{K}\right)$$

où d_1 est la distance mobile/station (K et α sont des paramètres donnés). Lorsque la station n'est pas en vue directe du mobile, le signal "parcourt" plusieurs segments pour atteindre la station, dans le cas de L segments la puissance reçue est

$$P_{\text{reçu}} = P_{\text{transmis}} - 10\alpha \log\left(\frac{d_L}{K}\right)$$

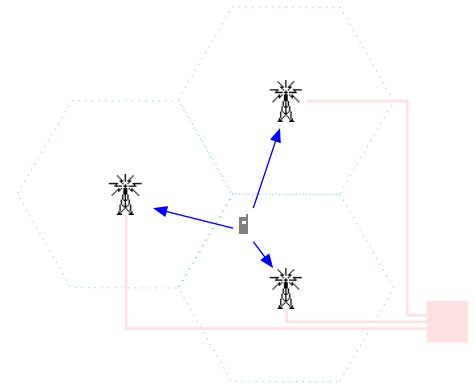


FIGURE 6.3 – Géolocalisation d'un mobile dans un réseau cellulaire urbain. Le mobile communique simultanément avec plusieurs stations.

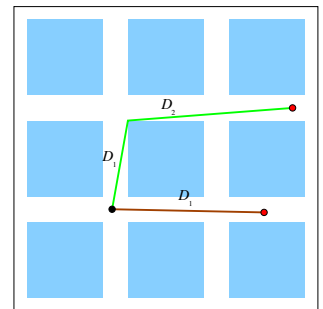
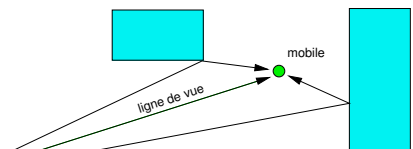


FIGURE 6.4 – carte d'atténuation simplifiée.



où

$$d_\ell = d_{\ell-1} + C_\ell + \beta_\ell D_\ell, \ell = 1 \cdots L, d_0 = 0$$

et D_ℓ est la longueur du segment ℓ (les coefficients C_ℓ et β_ℓ sont donnés).

Enfin pour tenir compte des imperfections on “bruite” le modèle

$$P_{\text{reçu}} = P_{\text{transmis}} - 10 \alpha \log\left(\frac{d_\ell}{K}\right) + \nu \quad (6.2)$$

où $\nu \sim \mathcal{N}(0, \sigma^2)$. Lorsque le mobile est à l’intérieur d’un bâtiment on suppose que la puissance du signal décroît de façon significative, après seuillage on suppose que cette puissance est nulle.

On construit donc pour chaque station une carte à l’aide du modèle (6.2). On obtient par exemple la carte présentée à la figure FIG. 6.7.

Modèle de déplacement du mobile

Le vecteur d’état est la position du mobile à l’instant k

$$\mathbf{X}_k = \begin{pmatrix} X_k^1 \\ X_k^2 \end{pmatrix}.$$

On suppose que le mobile est un piéton qui se déplace au hasard. On choisit l’équation d’état

$$\begin{aligned} X_{k+1}^1 &= X_k^1 + \sigma_W \sqrt{\Delta t} W_k^1, \\ X_{k+1}^2 &= X_k^2 + \sigma_W \sqrt{\Delta t} W_k^2 \end{aligned}$$

où W_k^1 et W_k^2 des bruits blancs gaussiens $\mathcal{N}(0, 1)$ indépendants. Δt est l’intervalle de temps entre deux observations. Naturellement ce mode de déplacement n’est pas réaliste (cf. FIG. 6.6) mais il est très simple et permet au filtre particulaire de bien explorer l’espace d’état. Le paramètre σ_W est choisit afin de correspondre à la vitesse de déplacement d’un piéton.

Équation d’observation

On suppose que le mobile est constamment en relation avec S stations de base. Pour chacune des stations $s = 1 \cdots S$ on dispose d’une mesure de la forme

$$Y_k^s = h_s(\mathbf{X}_k) + \sigma_V V_k$$

où $(V_k)_{k \geq 0}$ est un bruit blanc gaussien $\mathcal{N}(0, 1)$, la fonction d’observation $h_s(\mathbf{X}_k)$ est la puissance sur la carte d’atténuation de la station s pour un mobile en position \mathbf{X}_k .

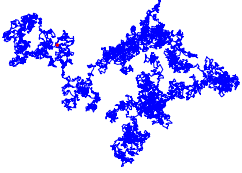


FIGURE 6.6 – Modèle de mouvement pour le piéton.

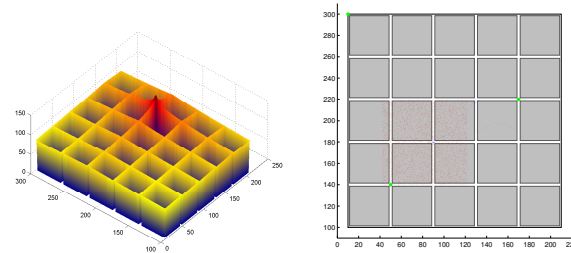


FIGURE 6.7 – Dans cet exemple on a disposé 3 stations (points verts). À gauche : un exemple de carte d'atténuation simulée correspondant à une station. À droite : les particules initiales.

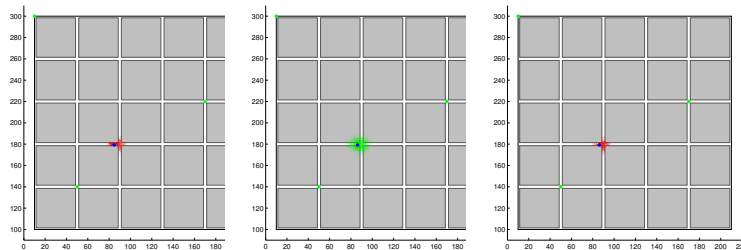


FIGURE 6.8 – Lorsque le mobile est dans la rue les particules sont également dans la rue : au niveau de la prédiction (image du centre), les particules diffusent dans toutes les directions, au niveau de la correction (image de droite) on élimine les particules qui sont à l'intérieur des b,timents.

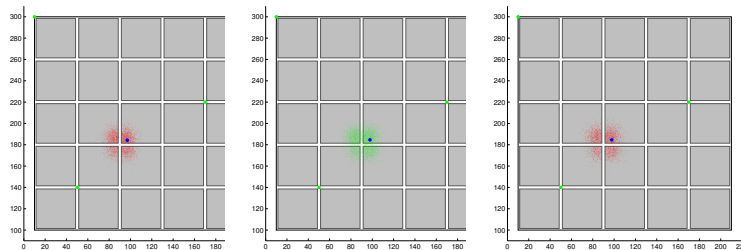


FIGURE 6.9 – Lorsque le mobile est dans un b,timent les particules sont également dans les bâtiments : au niveau de la prédiction (image du centre), les particules diffusent dans toutes les directions, au niveau de la correction (image de droite) on élimine les particules qui sont dans les rues.

Résultats

Dans l'exemple présenté dans les figures 6.7 à 6.9 on dispose de trois stations, trois cartes sont en effet nécessaires pour complètement lever les ambiguïtés sur la position du mobile.

On peut en particulier noter que lorsque le mobile est dans la rue, après correction les particules associées au filtre $p^N(X_k|Y_{1:k})$ sont uniquement dans les rues (cf. FIG. 6.7). Inversement, lorsque le mobile pénètre dans un bâtiment ces particules sont uniquement dans les bâtiments (cf. FIG. 6.7).

6.3 Poursuite par mesures d'angles avec obstacles

Cet exemple, dû à Simon Maskell, illustre les capacités du filtrage particulaire à tenir compte de contraintes plus complexes que celles introduites dans l'algorithme 11.

Un mobile se déplace dans le plan. Plusieurs stations effectuent des mesures bruitées d'azimut. Chaque station "sait" exactement si le mobile est masqué par un obstacle ou non (cf. FIG. 6.10). Ces obstacles sont des segments de droite. Les coordonnées des stations et des segments sont connues.

Le vecteur d'état est la position du mobile dans le plan à l'instant k

$$X_k = \begin{pmatrix} X_k^1 \\ X_k^2 \end{pmatrix}.$$

L'équation d'état est

$$\begin{aligned} X_{k+1}^1 &= X_k^1 + \sigma_W W_k^1, \\ X_{k+1}^2 &= X_k^2 + \sigma_W W_k^2 \end{aligned}$$

où W^1 et W^2 sont des bruits blancs gaussiens $\mathcal{N}(0, 1)$ et $X_0 \sim \mathcal{N}(\bar{x}_0, \sigma_0^2 I)$.

La loi de transition du processus de Markov X_k est

$$p(X_k|X_{k-1} = \xi) = \mathcal{N}(X_{k-1}, \sigma_w^2 I).$$

Pour simuler selon la loi $\tilde{\xi} \sim p(X_k|X_{k-1} = \xi)$, il suffit de poser

$$\tilde{\xi} \leftarrow \xi + \sigma_w^2 \begin{bmatrix} w^1 \\ w^2 \end{bmatrix}$$

où w^1 et w^2 sont tirées indépendamment selon la loi $\mathcal{N}(0, 1)$.

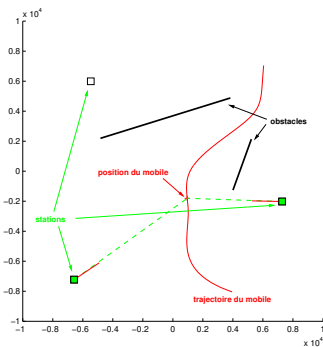


FIGURE 6.10 – Poursuite par mesures d'angles avec obstacles. Chaque station dispose de mesures bruitées de l'azimut au mobile. Chaque station sait exactement si le mobile est masqué par un obstacle ou non : lorsque le mobile n'est pas masqué la station prend des mesures

On dispose de S stations indicées par $s = 1 \dots S$. Ces stations travaillent indépendamment les unes des autres. À l'instant k la station s peut "surveiller" un domaine D_k^s . On pose

$$\delta_k^s \stackrel{\text{déf}}{=} \mathbf{1}_{D_k^s}(X_k).$$

À chaque instant k et pour toute station s on sait si $\delta_k^s = 0$ ou 1.

Donc à l'instant k , il y a deux cas :

— Si $X_k \in D_k^s$ alors on dispose de la mesure :

$$Y_k^s = h_s(X_k) + \sigma_V V_k^s$$

où $h_s(X_k) \stackrel{\text{déf}}{=} \text{atan2}(X_k^1 - x^{1,s}, X_k^2 - x^{2,s})$, $(x^{1,s}, x^{2,s})$ sont les coordonnées de la station s et V_k^s est un bruit blanc gaussien. Les processus $(V_k^s)_{k \geq 1}$ pour $s = 1, \dots, S$, $(W_k^1)_{k \geq 1}$, $(W_k^2)_{k \geq 1}$ et la loi position initiale X_0 sont supposées indépendantes.

— Si $X_k \notin D_k^s$ on ne dispose alors pas de mesure de la station s .

Initialisation

$$\xi_0^i \leftarrow \bar{x}_0 + \sigma_0 \begin{bmatrix} w_1^i \\ w_2^i \end{bmatrix} \quad i = 1 : N$$

où les w_j^i sont tous tirés indépendamment selon la loi $\mathcal{N}(0, 1)$.

Prédiction

On dispose des particules $(\xi_{k-1}^i)_{i=1:N}$, on suppose que l'on redistribue à chaque itération, ces particules sont donc équiprobables (i.e. de poids $1/N$).

$$\xi_{k-}^i \leftarrow \xi_{k-1}^i + \sigma_W \begin{bmatrix} w_1^i \\ w_2^i \end{bmatrix} \quad i = 1 : N$$

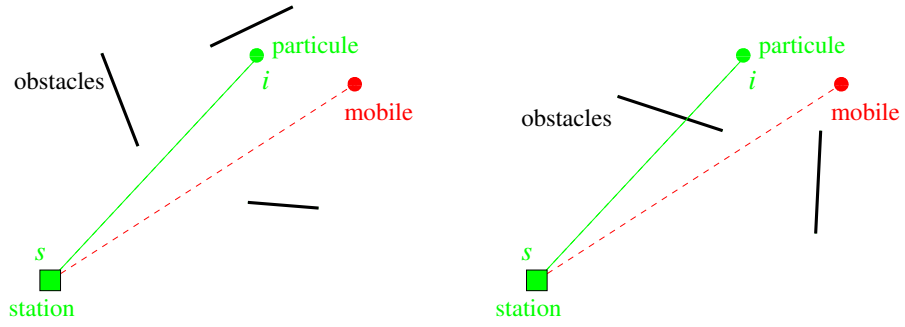
où les w_j^i sont tous tirés indépendamment selon la loi $\mathcal{N}(0, 1)$.

Correction

On dispose des particules prédites $(\xi_{k-}^i)_{i=1:N}$. Dans un premier temps on va calculer la vraisemblance locale ω_k^i associée à chacune de ces particules prédites.

On se fixe une station s on va calculer la vraisemblance locale $\omega_k^{i,s}$ relative à cette station (i.e. en supposant que seule cette station est opérationnelle).

- (i) Cas 1 : la station voit le mobile, i.e. $X_k \in D_k^s$. Il y a alors deux sous-cas selon que la station voit ou non la particule ξ_{k-}^i :



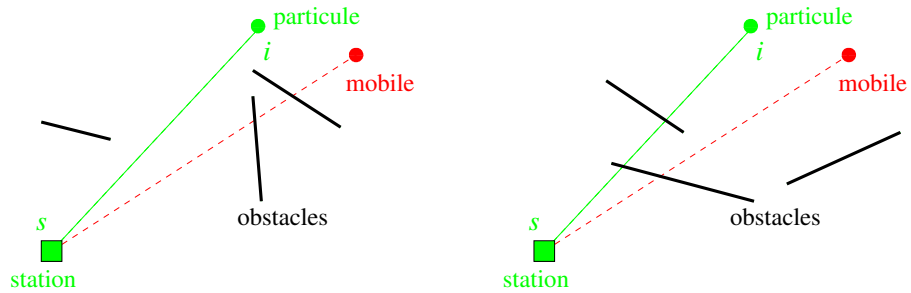
(i)-1 la station voit la particule (i)-2 la station ne voit pas la particule

Dans le deuxième cas la particule n'est pas vraisemblable du tout, i.e. sa vraisemblance locale est nulle. Dans le premier cas on se retrouve dans la situation classique sans obstacle. On peut donc écrire la vraisemblance locale de la particule sous la forme

$$\omega_k^{i,s} = \mathbf{1}_{D_k^s}(\xi_{k-}^i) \times \exp\left(-\frac{|Y_k^s - h_s(\xi_{k-}^i)|^2}{2\sigma_V^2}\right)$$

pour $i = 1 \cdots N$.

- (ii) Cas 2 : la station ne voit pas le mobile, i.e. $X_k \notin D_k^s$. Il y a alors deux sous-cas selon que la station voit ou non la particule ξ_{k-}^i :



(ii)-1 la station voit la particule (ii)-2 la station ne voit pas la particule

Dans le premier cas la particule n'est pas vraisemblable du tout, i.e. sa vraisemblance locale est nulle. Dans le deuxième cas, le mobile ainsi que la particule sont masqués, la vraisemblance locale de la particule est donc maximale (i.e. égale à 1) :

$$\omega_k^{i,s} = \mathbf{1}_{[D_k^s]^c}(\xi_{k-}^i) = 1 - \mathbf{1}_{D_k^s}(\xi_{k-}^i)$$

pour $i = 1 \cdots N$.

En conclusion on obtient l'expression

$$\omega_k^{i,s} = \delta_k^s \delta_k^{i,s} \exp\left(-\frac{|Y_k^s - h_s(\xi_{k-}^i)|^2}{2\sigma_V^2}\right) + (1 - \delta_k^s)(1 - \delta_k^{i,s}) \quad (6.3)$$

où

$$\delta_k^s \stackrel{\text{déf}}{=} \mathbf{1}_{D_k^s}(X_k) \quad \text{et} \quad \delta_k^{i,s} \stackrel{\text{déf}}{=} \mathbf{1}_{D_k^s}(\xi_{k-}^i)$$

sont donnés (i.e. pour tout k, s, i on sait si $X_k \in D_k^s$ ou non et si $\xi_{k-}^i \in D_k^s$ ou non). Lorsque la station s ne dispose pas de mesure, c'est-à-dire lorsque $\delta_k^s = 0$, on pose $Y_k^s = 0$ dans (6.3) (ou tout autre valeur, cela n'a pas d'importance).

Comme les mesures provenant des stations sont indépendantes, la vraisemblance de la particule i à l'instant k est

$$\omega_k^i = \prod_{s=1}^S \omega_k^{i,s}. \quad (6.4)$$

Si $\omega_k^i = 0$ pour tout i cela signifie que le filtre a “perdu la piste du mobile”, il est alors nécessaire de ré-initialiser le filtre. Dans le cas contraire on applique alors un algorithme de redistribution à $(\omega_k^i, \xi_{k-}^i)_{i=1:N}$ pour déterminer $(\xi_k^i)_{i=1:N}$.

Supposons que l'on sait qu'une zone D de la carte ne peut pas être parcourue par le mobile (imaginez que l'on suive une voiture et que D soit un lac). Cette contrainte est très facilement prise en compte en remplaçant (6.4) par

$$\omega_k^i = \mathbf{1}_{D^c}(\xi_{k-}^i) \prod_{s=1}^S \omega_k^{i,s}.$$

6.4 Poursuite dans une séquence d'images

On observe une séquence images $(Y_k)_{k \geq 1}$ représentant un mobile en mouvement dans du bruit. La position $r_k = (r_k^1, r_k^2)$ du mobile dans le plan affecte un pixel donné et les pixels voisins selon un modèle décrit ci-dessous. La trace

du mobile dans une seule image est noyée dans le bruit, il est nécessaire de traiter les images en séquence afin de poursuivre le mobile. C'est un exemple de problème de poursuite avant détection ("track before detect"), voir [53, 41]. On trouvera dans [28] et [46] des exemples réalistes de poursuites dans des séquences d'images par filtrage particulaire.

a. Modèle d'observation

Chaque image Y_k se décompose en

$$Y_k(s) = \mathcal{I}_{r_k}(s) + B_k(s)$$

où $s \in \mathcal{S}$ est l'indice de pixel (cf. Fig. 6.11), il s'agit de la somme d'une image bruit B_k et de l'image \mathcal{I}_{r_k} associé au mobile qui se trouve en position $r_k = (r_k^1, r_k^2)$. position du mobile

Pour l'image du mobile, on utilise un modèle de fonction d'étalement ("point spread function", PSF), c'est-à-dire :

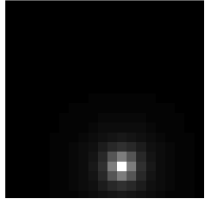
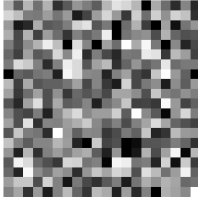


image du mobile



bruit

où

$$\mathcal{I}_r(s) = \frac{\delta^2 c}{2\pi\sigma^2} \exp\left(-\frac{|s\delta - r|^2}{2\sigma^2}\right) \mathbf{1}_{C(r)}(s)$$

$$\mathbf{1}_{C(r)}(s) \stackrel{\text{déf}}{=} \mathbf{1}_{(|s^1\delta - r^1| < 3)} \mathbf{1}_{(|s^2\delta - r^2| < 3)},$$

$$|s\delta - r|^2 \stackrel{\text{déf}}{=} (s^1\delta - r^1)^2 + (s^2\delta - r^2)^2$$

FIGURE 6.11 – L'image observée Y_k et la somme de l'image \mathcal{I}_{r_k} (à gauche) — contenant l'information sur la position r_k du mobile — et d'une image de bruit B_k (à droite).

et $s = (s^1, s^2) \in \mathcal{S}$.

L'image bruit est un bruit blanc

$$B_k(s) \sim \mathcal{N}(0, \sigma_B^2)$$

et $\{B_k(s)\}_{k \geq 1, s \in \mathcal{S}}$ est une famille de variables indépendantes (ce qui est une hypothèse peut réaliste en pratique). Par définition le rapport signal/bruit est $10 \times \log_{10} \frac{1}{\sigma_B}$.

À $Y = \{Y(s)\}_{s \in \mathcal{S}}$ fixé, on définit la fonction de vraisemblance $r \mapsto \mathcal{L}(Y|r)$ définie par :

$$\mathcal{L}(Y|r) \stackrel{\text{déf}}{=} \exp\left\{-\frac{1}{2\sigma_B^2} \sum_{s \in \mathcal{S}} [\mathcal{I}^r(s) - Y(s)]^2\right\}$$

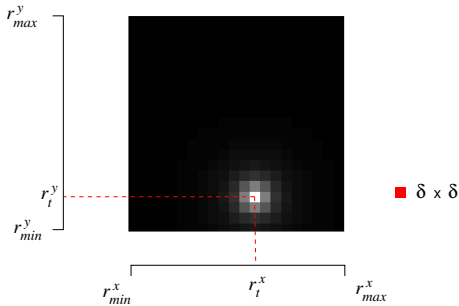


FIGURE 6.12 – Image $\mathcal{I}_r(s)$ du mobile en position r .

$$\begin{aligned}
\log L(Y|r) &= -\frac{1}{2\sigma_B^2} \sum_{s \in \mathcal{S}} \mathcal{J}_r(s) [\mathcal{J}_r(s) - 2Y(s)] - \frac{1}{2\sigma_B^2} \sum_{s \in \mathcal{S}} [Y(s)]^2 \\
&= -\frac{1}{2\sigma_B^2} \sum_{s \in C(r)} \mathcal{J}_r(s) [\mathcal{J}_r(s) - 2Y(s)] - \frac{1}{2\sigma_B^2} \sum_{s \in \mathcal{S}} [Y(s)]^2 \\
&= \text{cte} - \frac{1}{2\sigma_B^2} \sum_{s \in C(r)} \mathcal{J}_r(s) [\mathcal{J}_r(s) - 2Y(s)]
\end{aligned}$$

On note que

$$M_{C(r)}(\mathcal{J}_r, Y) \stackrel{\text{déf}}{=} \sum_{s \in C(r)} \mathcal{J}_r(s) [\mathcal{J}_r(s) - 2Y(s)]$$

est simple à calculer puisqu'il s'agit d'une sommation portant uniquement sur les pixels de $C(r)$.

La fonction de vraisemblance se réduit donc à :

$$\tilde{\mathcal{L}}(Y|r) = \exp \left\{ -\frac{1}{2\sigma_B^2} M_{C(r)}(\mathcal{J}_r, Y) \right\}.$$

b. Modèle d'état

Le vecteur d'état du mouvement du mobile est :

$$\mathbf{X}_k = \begin{pmatrix} \mathbf{X}_k^1 \\ \mathbf{X}_k^2 \end{pmatrix} = \begin{pmatrix} a_k^1 \\ v_k^1 \\ r_k^1 \\ a_k^2 \\ v_k^2 \\ r_k^2 \end{pmatrix} \begin{matrix} \rightarrow \text{utilisé dans l'observation} \\ \rightarrow \text{utilisé dans l'observation} \end{matrix}$$

L'équation d'état est :

$$\mathbf{X}_{k+1}^j = F \mathbf{X}_k^j + \sigma_W G W_k^j, \quad k \geq 1, \quad j = 1, 2$$

avec

$$F = \begin{pmatrix} 1 & 0 & 0 \\ \Delta & 1 & 0 \\ \frac{\Delta^2}{2} & \Delta & 1 \end{pmatrix} \quad G G^* = \begin{pmatrix} \Delta & \Delta^2/2 & \Delta^3/6 \\ \Delta^2/2 & \Delta^3/3 & \Delta^4/8 \\ \Delta^3/6 & \Delta^4/8 & \Delta^5/20 \end{pmatrix} \quad Q = \sigma_W^2 G G^*$$

c. Rao-Blackwellisation

Dans cette application, et pour un rapport signal/bruit très faible, l'approche classique par filtrage "bootstrap" montre rapidement ses limites. Nous faisons appel ici à une technique de Rao-Blackwellisation. Pour cela faisons deux remarques :

- seule la composante r de l'état apparaît dans le calcul de la vraisemblance, i.e. l'observation ne dépend de l'état que par la composante r , en effet :

$$\mathcal{L}(Y_k | \xi_{k-}^i) = \mathcal{L}(Y_k | r_{k-}^i).$$

- conditionnellement à la composante r l'équation d'état est linéaire/gaussienne.

Décomposons le vecteur d'état en $\alpha = (a, v)$ et $\beta = r$. L'équation d'état est de la forme :

$$\begin{pmatrix} \alpha_{k+1} \\ \beta_{k+1} \end{pmatrix} = \begin{pmatrix} F^{\alpha\alpha} & F^{\alpha\beta} \\ F^{\beta\alpha} & F^{\beta\beta} \end{pmatrix} \begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} + W_k$$

où

$$\text{cov}(W_k) = \begin{pmatrix} Q^{\alpha\alpha} & Q^{\alpha\beta} \\ Q^{\beta\alpha} & Q^{\beta\beta} \end{pmatrix}$$

Le structure du filtre est la suivante :

```

 $\xi_{1-}^i \sim \mu_{1-}$     $i = 1 : N$  % initialisation
pour  $k \geq 1$  faire
   $\omega_k^i \leftarrow \tilde{\mathcal{L}}(Y_k | r_{k-}^i)$ ,  $i = 1 : N$  % vraisemblance
   $r_k^i \sim \sum_j \omega_k^j \delta(r_{k-}^j)$ ,  $i = 1 : N$  % sélection
   $r_{k+1-}^i \sim \text{loi}(r_{k+1} | r_k = r_k^i)$ ,  $i = 1 : N$  % mutation
  sortie  $\xi_{1:N}$ 
fin pour

```

où $\text{loi}(r_{k+1} | r_k = r_k^i)$ est gaussienne. Il s'agit donc de simuler selon la loi $r_{k+1}^i \sim \text{loi}(\beta_{k+1} | \beta_k = r_k^i)$. Cela se fait selon l'itération du filtre de Kalman $k \rightarrow k+1$ pour la composante α où $\beta = r_k^i$ est traité comme une observation :

- prédiction de la composante α , $\text{loi}(\alpha_{k+1} | \beta_k = r_k^i) = \mathcal{N}(\hat{\alpha}_{k+1-}^i, R_{k+1-})$ avec :

$$\begin{aligned} \hat{\alpha}_{k+1-}^i &= F^{\alpha\alpha} \hat{\alpha}_k^i + F^{\alpha\beta} r_k^i \\ R_{k+1-} &= F^{\alpha\alpha} R_k (F^{\alpha\alpha})^* + Q^{\alpha\alpha} \end{aligned}$$

- prédiction de l'observation $\text{loi}(\beta_{k+1} | \beta_k = r_k^i) = \mathcal{N}(\hat{r}_{k+1-}^i, \Xi_{k+1})$ avec :

$$\begin{aligned} \hat{r}_{k+1-}^i &= F^{\beta\alpha} \hat{\alpha}_k^i + F^{\beta\beta} r_k^i \\ \Xi_{k+1} &= F^{\beta\alpha} R_k (F^{\beta\alpha})^* + Q^{\beta\beta} \end{aligned}$$

- correction de la composante α $\text{loi}(\alpha_{k+1} | \beta_{k+1} = r_{k+1-}^i) = \mathcal{N}(\hat{\alpha}_{k+1}^i, R_{k+1})$ avec :

$$\begin{aligned} S_{k+1} &= F^{\alpha\alpha} R_k (F^{\beta\alpha})^* + Q^{\alpha\beta} \\ \hat{\alpha}_{k+1}^i &= \hat{\alpha}_{k+1-}^i + S_{k+1} \Xi_{k+1}^{-1} (r_{k+1}^i - \hat{r}_{k+1-}^i) \\ R_{k+1} &= R_{k+1-} - S_{k+1} \Xi_{k+1}^{-1} S_{k+1}^* \end{aligned}$$

Notons que les covariances ne dépendent pas de l'indice de particule. Au cours de cette itération on simule la particule prédite (étape de mutation) :

$$r_{k+1-}^i \sim \text{loi}(\beta_{k+1} | \beta_k = r_k^i) = \mathcal{N}(\hat{r}_{k+1-}^i, \Xi_{k+1}).$$

Dans le cadre de Rao-Blackwell, un filtre de Kalman est associé à chaque particule. Il s'agit donc d'une méthode plus coûteuse que le filtre bootstrap. L'avantage est qu'une partie des composantes est traitée sans approximation. Donc, bien que plus lourde, cette approche est plus performante.

A

QUELQUES RÉSULTATS TECHNIQUES

A.1 Ellipse de confiance

Matrice de covariance

Une matrice symétrique R de taille $n \times n$ est dite définie non-négative, noté $R \geq 0$, lorsque $x^* R x \geq 0$ pour tout $x \in \mathbb{R}^n$. Elle est dite définie positive, noté $R > 0$, lorsque $x^* R x > 0$ pour tout $x \in \mathbb{R}^n$ non nul. Soient R et Q deux matrices symétriques de taille $n \times n$, on dit que $R \geq Q$ (resp. $R > Q$) lorsque $R - Q \geq 0$ (resp. $R - Q > 0$).

Soit $X \sim N(\mu, \Sigma)$ avec $\Sigma > 0$. La matrice de covariance Σ admet une décomposition de Cholesky $\Sigma = LL^*$ où L est une matrice diagonale inférieure dont les éléments diagonaux sont positifs alors $\tilde{X} = L^{-1}X \sim \mathcal{N}(0, I)$. La fonction `chol` de MatLab fournit une décomposition $\Sigma = U^*U$ où U est triangulaire supérieure.

On note $L = \Sigma^{1/2}$, ainsi $X = \mu + \Sigma^{1/2}\tilde{X}$ où $\tilde{X} \sim \mathcal{N}(0, I)$, en effet $\text{cov}(X) = \Sigma^{1/2} \text{cov}(\tilde{X}) (\Sigma^{1/2})^* = I$. Cette matrice “racine carrée” n’est pas unique, il existe différentes méthodes pour la calculer (cf. la section ci-dessous).

Ellipse de confiance

L’ellipse de confiance généralise la notion d’intervalle de confiance au cas des vecteurs gaussiens en dimension 2. On considère un vecteur aléatoire gaussien $X \sim N(\mu, \Sigma)$ de dimension 2 avec $\Sigma > 0$. Il s’agit de déterminer les courbes de niveau de la densité gaussienne, i.e. les points x de l’ellipse

$$(x - \mu)^* \Sigma^{-1} (x - \mu) = r.$$

régles de calcul

	espace discret	espace continue
loi	$\mathbb{P}(X = x)$ loi de X $\mathbb{P}(X \in A) = \sum_{x \in A} \mathbb{P}(X = x)$ $\mathbb{E}(\phi(X)) = \sum_x \phi(x) \mathbb{P}(X = x)$	$p(X)$ densité de la loi de X $\mathbb{P}(X \in A) = \int_A p(X) dX$ $\mathbb{E}(\phi(X)) = \int \phi(X) p(X) dX$
loi jointe loi marginale	$\mathbb{P}(X = x, Y = y)$ loi jointe de (X, Y) $\mathbb{P}(X = x) = \sum_y \mathbb{P}(X = x, Y = y)$	$p(X, Y)$ densité de la loi de (X, Y) $p(X) = \int p(X, Y) dY$
loi conditionnelle propriétés	$\mathbb{P}(X = x Y = y) \stackrel{\text{def}}{=} \frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(Y = y)} = \frac{\mathbb{P}(X = x, Y = y)}{\sum_{x'} \mathbb{P}(X = x', Y = y)}$ $\mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x Y = y) \mathbb{P}(Y = y)$ $\mathbb{P}(X_{1:k} = x_{1:k}) = \mathbb{P}(X_1 = x_1) \prod_{\ell=2}^k \mathbb{P}(X_\ell = x_\ell X_{1:\ell-1} = x_{1:\ell-1})$ $\mathbb{P}(X = x) = \sum_y \mathbb{P}(X = x Y = y) \mathbb{P}(Y = y)$ $\mathbb{P}(X = x Z = z) = \sum_y \mathbb{P}(X = x Y = y, Z = z) \mathbb{P}(Y = y Z = z)$	$p(X Y) \stackrel{\text{def}}{=} \frac{p(X, Y)}{p(Y)} = \frac{p(X, Y)}{\int p(X, Y) dY}$ $p(X, Y) = p(X Y) p(Y)$ $p(X_{1:k}) = p(X_1 = x_1) \prod_{\ell=2}^k p(X_\ell X_{1:\ell-1})$ $p(X) = \int p(X Y) p(Y) dY$ $p(X Z) = \int p(X Y, Z) p(Y Z) dY$
indépendance	$X \perp\!\!\!\perp Y$ $\Leftrightarrow \mathbb{P}(X \in A, Y \in B) = \mathbb{P}(X \in A) \times \mathbb{P}(Y \in B), \forall A, B$ $\Leftrightarrow \mathbb{E}[\phi(X) \times \psi(Y)] = \mathbb{E}\phi(X) \times \mathbb{E}\psi(Y), \forall \phi, \psi$ $\Leftrightarrow \mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x) \mathbb{P}(Y = y)$ $\Leftrightarrow \mathbb{P}(X = x Y = y) = \mathbb{P}(X = x)$ si $Z \perp\!\!\!\perp (X, Y) : \mathbb{P}(X = x Y = y, Z = z) = \mathbb{P}(X = x Y = y)$	$X \perp\!\!\!\perp Y$ $\Leftrightarrow \mathbb{P}(X \in A, Y \in B) = \mathbb{P}(X \in A) \times \mathbb{P}(Y \in B), \forall A, B$ $\Leftrightarrow \mathbb{E}[\phi(X) \times \psi(Y)] = \mathbb{E}\phi(X) \times \mathbb{E}\psi(Y), \forall \phi, \psi$ $\Leftrightarrow p(X, Y) = p(X) p(Y)$ $\Leftrightarrow p(X Y) = p(X)$ si $Z \perp\!\!\!\perp (X, Y) : p(X Y, Z) = p(X Y)$
formule de Bayes loi a priori loi a posteriori vraisemblance	$\mathbb{P}(X = x Y = y) = \frac{\mathbb{P}(Y = y X = x) \mathbb{P}(X = x)}{\mathbb{P}(Y = y)} = \frac{\mathbb{P}(Y = y X = x) \mathbb{P}(X = x)}{\sum_{x'} \mathbb{P}(Y = y X = x') \mathbb{P}(X = x')}$ $\mathbb{P}(X = x)$ $\mathbb{P}(X = x Y = y)$ $\mathbb{P}(Y = y X = x)$	$p(X Y) = \frac{p(Y X) p(X)}{p(Y)} = \frac{p(Y X) p(X)}{\int p(Y X) p(X) dX}$ $p(X)$ $p(X Y)$ $p(Y X)$

En tant que matrice symétrique définie positive Σ admet la décomposition spectrale $\Sigma = PDP^*$ où P est la matrice orthonormale des vecteurs propres (dans un ordre quelconque) et D est la matrice diagonale des valeurs propres (dans le même ordre). Pour chaque x on définit le point \tilde{x} par $\tilde{x} = D^{-1/2} P x$, alors $|\tilde{x}|^2 = r$, i.e. l'ensemble des \tilde{x} est le cercle centré de rayon r : $\tilde{x}(\rho) = r [\cos(\rho) \sin(\rho)]^*$ pour $0 \leq \rho < 2\pi$. L'équation paramétrique de l'ellipse est donc

$$x(\rho) = \mu + r P^* D^{1/2} \begin{bmatrix} \cos(\rho) \\ \sin(\rho) \end{bmatrix} \quad (0 \leq \rho < 2\pi).$$

La norme au carré du vecteur aléatoire $\tilde{X} = D^{-1/2} P X$ suit une loi du chi-2 à 2 degrés de liberté. En prenant $r = 4.61$ alors

$$\mathbb{P}\left(X \in \{x \in \mathbb{R}^2 : (x - \mu)^* \Sigma^{-1} (x - \mu) \leq r\}\right) = 0.9.$$

Voir la routine `matlab` ALG. 16.

A.2 Transformation de Box–Muller

Cette transformation — proposée par [12] — se vérifie à l'aide d'un changement de variable dans une intégrale sur \mathbb{R}^2 (voir ci-dessous). On pose :

$$\begin{aligned} u_1 &= f_1(x_1, x_2) = e^{-(x_1^2 + x_2^2)/2}, \\ u_2 &= f_2(x_1, x_2) = \frac{1}{2\pi} \tan^{-1}(x_2/x_1). \end{aligned}$$

Le jacobien de f est

$$J = \det \begin{bmatrix} \partial_{x_1} f_1 & \partial_{x_2} f_1 \\ \partial_{x_1} f_2 & \partial_{x_2} f_2 \end{bmatrix} = -\frac{1}{\sqrt{2\pi}} e^{-x_1^2/2} \frac{1}{\sqrt{2\pi}} e^{-x_2^2/2}.$$

Ainsi, si X_1 et X_2 sont indépendants et gaussiens $N(0, 1)$ alors

$$\begin{aligned} \mathbb{E}\varphi(X_1, X_2) &= \int \int_{\mathbb{R}^2} \varphi(x_1, x_2) q(x_1) q(x_2) dx_1 dx_2 \quad \text{avec } q(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \\ &= \int \int_{[0,1]^2} \varphi(\sqrt{-2 \log u_1} \cos u_2, \sqrt{-2 \log u_1} \sin u_2) du_1 du_2 \\ &= \mathbb{E}\varphi(\sqrt{-2 \log u_1} \cos u_2, \sqrt{-2 \log u_1} \sin u_2) \end{aligned}$$

où u_1 et u_2 sont indépendants et de loi uniforme $U[0, 1]$.

Formule de changement de variable dans une intégrale : On considère une bijection $\tau : C \mapsto D$ où C et D sont des ouverts de \mathbb{R}^n telle que τ et τ^{-1} soient continûment différentiables. Le *jacobien* de τ au point $u = (u_1, \dots, u_n)$ est défini par

$$J(u) \stackrel{\text{déf}}{=} \det \left[\frac{\partial \tau_i}{\partial u_j}(u) \right]_{1 \leq i, j \leq n}.$$

Si $f : D \mapsto \mathbb{R}$ est borélienne, positive ou intégrable par rapport à la mesure de Lebesgue, alors

$$\int_D f(x) dx = \int_C f(\tau(u)) |J(u)| du.$$

A.3 Un peu de complexité algorithmique

La complexité d'un algorithme est le nombre d'opérations élémentaires (accès mémoire, comparaisons, opérations arithmétiques) effectuées par cet algorithme et s'exprime en fonction de la taille N des données.

On dit que la complexité de l'algorithme est $O(f(N))$ où $f(N)$ est une combinaison de polynômes, de logarithmes ou d'exponentielles. Il s'agit de la notation mathématique classique : dire qu'un algorithme est en $O(f(N))$ signifie que le nombre d'opérations effectuées est borné par $C f(N)$, où C est une constante, lorsque N tend vers l'infini.

On a les règles classiques de calcul :

- $O(3N + 5) = O(N)$,
- $O(\log N) + O(\log N) = O(\log N)$,
- $O(N^2) + O(N) = O(N^2)$,
- $O(N^2) O(N \log(N)) = O(N^3 \log(N))$ etc.

On peut distinguer par ordre croissant de complexité :

- $O(\log(N))$ logarithmique (sous-linéaire),
- $O(N)$ linéaire,
- $O(N \log(N))$,
- $O(N^p)$ polynomiale,
- $O(p^N)$ exponentielle.

Par exemple la recherche d'un élément dans un ensemble ordonné fini de cardinal N est en $O(\log(N))$; les algorithmes optimaux de tri sont en $O(N \log(N))$; la multiplication des matrices ou le parcours dans les graphes sont polynomiaux $O(N^p)$ avec $p \leq 3$. En général, les algorithmes polynomiaux en $O(N^k)$ avec $k > 3$ sont considérés comme très lents. Les algorithmes exponentiels sont impraticables dès que la taille des données est supérieure à quelques dizaines d'unités.

Pour résoudre un problème on cherche l'algorithme ayant la plus faible complexité, mais il faut faire attention à la constante multiplicative C : passer de $\frac{1}{2}N^2$ à $10^{10} N \log(N)$ n'est pas une amélioration!

La complexité n'est pas tout. L'occupation mémoire est également importante (même si c'est de moins en moins le cas) sans parler de la lisibilité de l'algorithme.

B

ROUTINES MATLAB

On donne maintenant les sources de 2 fonctions `matlab` de redistribution introduites au § 5.5 page 66

ALG. 13 `f_resample_residual` — redistribution des résidus.

ALG. 14 `f_resample_comb_randshift` — redistribution de Kitagawa.

En sortie de ces 2 dernières fonctions on obtient le tableau `offsprings(1:N)` du nombre `offsprings(i)` de descendants pour chacune des particules i . Il est encore nécessaire de calculer le tableau `indices(1:N)` des indices des particules sélectionnées. C'est l'objet de la routine :

ALG. 15 `f_resample_multi`

Par ailleurs on propose une fonction pour le calcul d'ellipse de confiance pour une loi gaussienne en dimension 2 :

ALG. 16 `f_plot_ellipse`

Important : *Ces fonctions `matlab` peuvent être utilisées dans toutes applications non-commerciales (enseignement et recherche universitaire). Contacter l'auteur pour tout autre utilisation.*

```

function offsprings=f_resample_residual(weights)
%-----
% function offsprings=f_resample_residual(weights)
% RESIDUAL RESAMPLING PROCEDURE for particle filter algorithms
% INPUT  weights : probability measure
%         (it is not checked if the sum of the weights is one)
% OUTPUT offsprings : drawn offsprings
%
% offsprings(k)=j means that point #k is resampled j times
% resampling according to the probability measure weights(k) k=1:np
% with np=length(weights)
% 1) one calculates the integer parts of np*weights
% 2) one draws the residual particles according to the residual law:
%     weights(k)-floor(weights(k)*np)/np    k=1:np
%-----
% AUTHOR: Fabien Campillo - INRIA
%-----

np = length(weights);

% --- one computes the np first points

offsprings = floor(np*weights);
weights    = weights - (1/np)*offsprings;

% ---- there remain np-sum(offsprings) particles to be drawn

nnp      = np - sum(offsprings);
weights  = weights/sum(weights);
weights  = cumsum(weights);

for k=1:nnp
    threshold = rand;
    i = 1;
    while weights(i)<threshold, i = i+1; end
    offsprings(i) = offsprings(i)+1;
end

```

ALG. 13: *Redistribution des résidus.*

```

function offsprings=f_resample_comb_randshift(weights)
%-----
% function offsprings=f_resample_comb(weights)
%-----
% function offsprings=f_resample_residual(weights)
% KITAGAWA RESAMPLING PROCEDURE for particle filter algorithms
% INPUT  weights : probability measure
%        (it is not checked if the sum of the weights is one)
% OUTPUT offsprings : drawn offsprings
%
% offsprings(k)=j means that point #k is resampled j times
% resampling according to the probability measure weights(k) k=1:np
% with np=length(weights)
%-----
% AUTHOR: Fabien Campillo - INRIA
%-----
n_weights      = length(weights);
weights         = cumsum(weights)+rand/n_weights;
weights         = weights/weights(end);
weights         = floor(weights*n_weights);
offsprings      = [weights(1) weights(2:n_weights)-weights(1:n_weights-1)];

```

ALG. 14: *Redistribution de Kitagawa.*

```

function indices = f_resample_indices(ftable)
%-----
% function indices = f_resample_indices(ftable)
% to be used after a resampling function
% INPUT a table [i1 i2 ... in]
%        ij : means that point #j has been drawn ij times
% OUTPUT [1*ones(1,i1) 2*ones(1,i2) ... n*ones(1,in)]
% ex.    mult([0  2  0  2  1  1])
%        gives [2  2  4  4  5  6]
%-----
% AUTHOR: Fabien Campillo - Qinghua Zhang - INRIA
%-----
fftable        = cumsum(ftable+1);
a              = zeros(1,fftable(end));
a(fftable)     = 1;
aa            = cumsum(a);
indices        = aa(~a)+1;

```

ALG. 15: *Calcul des indices (à utiliser avec les fonctions de redistribution).*

```

function f_plot_ellipse(fmean,fcov,npoint,fcolor,R)
%-----
% function f_plot_ellipse(fmean,fcov,npoint,fcolor)
% compute and plot a confidence ellipse of level 90%
% corresponding to the two-dimensional normal law N(fmean, fcov).
% parameters: npoint : number of points for the layout of the ellipse
%              color [character] color of the ellipse
% One decomposes the covariance matrix  $R = P D P'$ 
% (P orthonormal D diagonal) then  $Z = D^{-1/2} P (x-m)$ 
% where M is the average, then  $|Z|^2 = R$  the parametric equation
% of the ellipse is
%  $y(s) = m + P' \sqrt{r D} [\cos(s) \sin(s)]'$ 
%  $|Z|^2$  follows a law of the chi2 with two degrees of freedom
% R=4.61 gives an confidence ellipse of 90%
%-----
% AUTHOR: Fabien Campillo - INRIA
%-----
if nargin<=4, R=4.61; end      % default R=4.61 gives a confidence ellipse
                               % of level 90%
if nargin<=3, fcolor='k'; end % default fcolor is 'black'
if nargin<=2, npoint=45; end  % default 45 points

[vec,val] = eig(fcov);
t         = linspace(0,2*pi,npoint);
ellipse   = vec * sqrt(val*R) * [cos(t);sin(t)];
ellipse   = ellipse + repmat([fmean(1);fmean(2)],1,npoint);
plot(ellipse(1,:),ellipse(2,:),fcolor)

```

ALG. 16: *Calcul et tracé d'une ellipse de confiance pour une loi gaussienne en dimension 2 (cf. Annexe § A.1).*

BIBLIOGRAPHIE

- [1] Christer Åkerblom. *Tracking Mobile Phones in Urban Areas*. Licentiate Thesis, Departement of Mathematical Statistics, Chalmers University of Technology, Göteborg, 2000.
- [2] Brian D. O. Anderson and John B. Moore. *Optimal Filtering*. Prentice-Hall, 1978.
- [3] M. Sanjeev Arulampalam, Simon Maksell, Neil J. Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, SP-50(2 (Special issue on Monte Carlo Methods for Statistical Signal Processing)) :174–188, February 2002.
- [4] Nathalie Bartoli and Pierre Del Moral. *Simulation et algorithmes stochastiques*. Cépaduès, Toulouse, 2001.
- [5] Leonard E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In Oved Shisha, editor, *Inequalities III, Los Angeles 1969*, pages 1–8. Academic Press, New York, 1971.
- [6] Leonard E. Baum and J.A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the AMS*, 73 :360–363, 1967.
- [7] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, 37(6) :1554–1563, December 1966.
- [8] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1) :164–171, February 1970.
- [9] Leonard E. Baum and George R. Sell. Growth transformations for functions on manifolds. *Pacific Journal of Mathematics*, 27(2) :211–227, 1968.

- [10] Nicolas Bouleau. *Probabilités de l'ingénieur : variables aléatoires et simulation*. Hermann, Paris, 1986.
- [11] Nicolas Bouleau. *Processus stochastiques et applications*. Hermann, Paris, 1988.
- [12] G.P.E. Box and M.E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29 :610–611, 1958.
- [13] Gerrit Burgers, Peter Jan van Leeuwen, and Geir Evensen. On the analysis scheme in the Ensemble Kalman Filter. *Monthly Weather Review*, 126 :1719–1724, 1998.
- [14] Karim Dahia. *Nouvelles méthodes en filtrage particulière – Application au recalage de navigation inertielle par mesures altimétriques*. Thèse, LMC/IMAG, Université Joseph Fourier, Grenoble I, January 2005.
- [15] Bernard Delyon. *Simulation et modélisation*. Université Rennes 1, 2006.
- [16] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39 :1–38, 1977.
- [17] Arnaud Doucet, Nando de Freitas, and Neil J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.
- [18] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, 1998.
- [19] Robert J. Elliott, Lakhdar Aggoun, and John B. Moore. *Hidden Markov Models : Estimation and Control*, volume 29 of *Applications of Mathematics*. Springer-Verlag, New York, 1995.
- [20] Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5) :10,143–10,162, 1994.
- [21] Geir Evensen. The ensemble Kalman filter : theoretical formulation and practical implementation. *Ocean Dynamics*, 53 :343–367, 2003.
- [22] Arthur Gelb, editor. *Applied Optimal Estimation*. MIT Press, 1974.
- [23] Michael Ghil. The essence of data assimilation, or Why combine data with models? In *Proc. 3rd WMO Int'l Symp. Assimilation of Observations in Meteorology & Oceanography*, pages 1–4, Québec, Canada, 1999, 2000.

- [24] Neil J. Gordon, David J. Salmond, and Adrian F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings, Part F*, 140(2) :107–113, April 1993.
- [25] Fredrik Gunnarsson, Per-Johan Nordlund, and Fredrik Gustafsson. Particle filters for positioning in wireless networks. In *Proc. of EUSIPCO*, Toulouse, 2002.
- [26] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forsell, Jonas Jansson, Rikard Karlsson, and Per-Johan Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 50(2), 2002.
- [27] Peter Guttorp. *Stochastic modeling of scientific data*. Chapman & Hall, London, 1995.
- [28] Michael Isard and Andrew Blake. CONDENSATION — Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1) :5–28, August 1998.
- [29] Jean Jacod and Philip Protter. *L'essentiel en théorie des probabilités*. Cassini, 2003.
- [30] Andrew H. Jazwinski. *Stochastic processes and filtering theory*. Academic Press, 1970.
- [31] Simon Julier and Jeffrey K. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, RRG, Dept. of Engineering Science, University of Oxford, 1996.
- [32] Simon J. Julier and Jeffrey K. Uhlmann. A consistent, debiased method for converting between polar and cartesian coordinate systems, 1997. citeseer.nj.nec.com/julier97consistent.html.
- [33] Simon J. Julier and Jeffrey K. Uhlmann. Unscented filtering and nonlinear estimation. *IEEE Review*, 92(3) :401–422, 2004.
- [34] Simon J. Julier, Jeffrey K. Uhlmann, and Hugh F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45 :477–482, March 2000.
- [35] Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82, Series D :35–45, 1960.

- [36] Rudolf E. Kalman and Richard S. Bucy. New results in linear filtering and prediction theory. *Transactions of the ASME–Journal of Basic Engineering*, 83, Series D :95–108, 1961.
- [37] Rickard Karlsson, Fredrik Gustafsson, and Tobias Karlsson. Particle filter and Cramer–Rao lower bound for underwater navigation. In *Proceedings of ICASSP*, Hongkong, 2003.
- [38] Jun S. Liu, Rong Chen, and Wing-Hung Wong. Rejection control and sequential importance sampling. *Journal of the American Statistical Association*, 93(443) :1022–1031, September 1997.
- [39] Iain L. MacDonald and Walter Zucchini. *Hidden Markov and Other Models for Discrete-Valued Time Series*. Chapman & Hall, London, 1997.
- [40] Andrei Andreevich Markov. Primer statisticheskogo issledovaniya nad tekstom “Evgeniya Onegina”, illyustriruyuschij svyazí ispytaniy v cepí (an example of statistical study on the text of “Eugene Onegin” illustrating the linking of events to a chain). In *Izvestija Imp. Akademii nauk, serija VI*, volume 3, pages 153–162, 1913.
- [41] Simon Maskell, M. Rollason, Neil J. Gordon, and D. Salmond. Efficient particle filtering for multiple target tracking with application to tracking in structured images. *Image and Vision Computing*, 21(10) :931–939, 2003.
- [42] Peter S. Maybeck. *Stochastic Models, Estimation and Control*. Academic Press, 1979. 3 volumes (from 1979 to 1982).
- [43] Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, 1997.
- [44] Annie Millet. Méthodes de Monte Carlo. Université Paris 6, 2006.
- [45] Christian Musso and Nadia Oudjane. Recent particle filter applied to terrain navigation. In *Proceedings of the 3rd International Conference on Information Fusion, Paris 2000*. ISIF, July 2000. Paper WeB5-4.
- [46] Patrick Pérez, Jaco Vermaak, and Andrew Blake. Data fusion for visual tracking with particles. *Proc. IEEE*, 92(3), 2004.
- [47] Dinh-Tuan Pham. Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Monthly Weather Review*, 129(5) :1194–1207, 2001.
- [48] A.B. Poritz. Hidden Markov models : a guided tour. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1988*, pages 7–13. IEEE–SPS, 1988.

- [49] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286, February 1989.
- [50] Lawrence R. Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, 1993.
- [51] Branko Ristic, M. Sanjeev Arulampalam, and Niel J. Gordon. *Beyond the Kalman Filter : Particle Filters for Tracking Applications*. Artech House, 2004.
- [52] Christian P. Robert. *L'Analyse Statistique Bayésienne*. Economica, Paris, 1992.
- [53] D.J. Salmond and H. Birch. A particle filter for track-before-detect. In *Proceedings of the 2001 American Control Conference, Arlington, VA, USA*, volume 5, pages 3755–3760, 2001.
- [54] Gilbert Saporta. *Probabilités, analyse des données et statistique*. Technip, 1990.
- [55] Gilbert Saporta, Jean-Jacques Dreesbeke, and Fine Jeanne, editors. *Méthodes bayésiennes en statistique*. Technip, 2002.
- [56] Sebastian Thrun. Particle filters in robotics. In *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, 2002.

INDEX

δ_ξ , mesure de Dirac, 14
 $X \perp\!\!\!\perp Y$, voir indépendance

$\mathcal{M}(r, \omega^1, \dots, \omega^N)$, voir loi
multinomiale

$\xi^{1:N} \stackrel{\text{iid}}{\sim} p(x)$, 56

$\xi \sim p(x)$, 56

$\xi^{1:N} \stackrel{i}{\sim} p_1(x_1) \cdots p_N(x_n)$, 56

algorithme EM, 27

Baum

algorithme backward, 17

cas continu, 33, 33–34

cas fini, 22, 22–24

algorithme forward, 17

cas continu, 32, 31–32

cas fini, 20, 20–21

Baum-Welch

algorithme de, 18

Bayes, formule de, 9

Box–Muller

transformation de, 57, 89

bruit blanc, 15

bruit blanc gaussien, 15

canal sans mémoire, 17

chaîne de Markov, 15

cachée, 17

loi de transition d’une, 15

loi initiale d’une, 15

densité conditionnelle, 6

dualité des équations de Baum
forward/backward, 23

échantillonnage d’importance, 59

ellipse de confiance, 87

équation de

Chapman–Kolmogorov, 15

espérance conditionnelle, 6

estimateur, 6

global du maximum a

posteriori, 25, 34

local du maximum a

posteriori, 25, 34

filtrage

problème de, 17

fonction

valeur, 25

fonction de vraisemblance, 10

fonction valeur, 35

indépendance, 8

innovation, 44

lissage

problème de, 17

LMAP, voir estimateur local du

maximum a posteriori,

voir estimateur local du

maximum a posteriori

loi

conditionnelle, voir densité

conditionnelle

- gaussienne, 11
- marginale, 5
- multinomiale, 13, 14
- loi d'importance, 60
- loi d'émission, 18
- MAP, *voir* estimateur global du
maximum a posteriori,
voir estimateur global du
maximum a posteriori
- matrice d'émission, 18
- méthode d'acceptation/rejet, 58
- noyau markovien, 9
- poids d'importance, 60
- processus stochastique, 14
- programmation dynamique, 24,
26, 34, 35
- variable backward
 - cas continu, 33
 - cas fini, 22
- variable forward
 - cas continu, 31
 - cas fini, 20
- Viterbi
 - algorithme de, 18, 24, 25, 34
 - cas continu, 35
 - cas fini, 26

Ce document est mis à disposition selon les termes de la licence [Creative Commons](#) “Attribution – Pas d’utilisation commerciale – Partage dans les mêmes conditions 4.0 International”.

