Projet Cassiopé

Filtrage statistique optimal non supervisé dans une chaîne de Markov couple

Auteurs :

Simon Bussy
Mamery Diarrassouba
Emmanuel Cosnard

Encadrant :

Wojciech Pieczynski
Directeur du département "Communications, Images et Traitement de l'Information"

Soutenance le 3 juin 2013

Table des matières

1 Contexte et notations

Un probléme important dans une large gamme de domaines consiste à pouvoir estimer un ensemble de variables inobservables, dites cachées, à partir d'un ensemble de variables observées, qui peuvent être interprétées comme des versions bruitées des variables cachées.
L'objectif est alors de retrouver les données cachées avec la meilleure précision possible en éliminant l'effet du bruit, ce qui peut être fait avec un filtre de Kalman.
Classiquement, le filtre de Kalman est utilisé dans le cadre d'une chaîne de Markov cachée. On suppose alors dans ce cas que le processus X, dont les réalisations sont les variables inobservables, est une chaîne de Markov.
Notre projet consiste à étudier le problème dans un cadre plus général, et plus récent puisqu'il est étudié depuis une dizaine d'années seulement, celui d'une chaîne de Markov couple qui peut être vu comme la généralisation du modèle précedent.

Notons $\mathrm{Z}_1^2 = (Z_1, Z_2) = (X_1, Y_1, X_2, Y_2)$. Pour simplifier, supposons que toutes les moyennes sont nulles et que toutes les variances valent 1. La distribution Gaussienne de Z_1^2 est alors donnée par la matrice de covariance suivante :

$$\Gamma^{Z_1^2} = \begin{bmatrix} 1 & b & a & d \\ b & 1 & e & c \\ a & e & 1 & b \\ d & c & b & 1 \end{bmatrix} = \begin{bmatrix} \Gamma & D^t \\ D & \Gamma \end{bmatrix} \qquad (1)$$

$$\text{Avec } \Gamma = \mathrm{Cov}(\mathrm{Z}_1) = \begin{bmatrix} 1 & b \\ b & 1 \end{bmatrix} \text{ et D}=\mathrm{Cov}(\mathrm{Z}_1, Z_2) = \begin{bmatrix} a & e \\ d & c \end{bmatrix} \qquad (2)$$

$$\text{En notant Cov}(\mathrm{Z}_1, Z_2) = \begin{bmatrix} cov(X_1, X_2) & cov(X_1, Y_2) \\ cov(Y_1, X_2) & cov(Y_1, Y_2) \end{bmatrix}$$

Ainsi la distribution de Z_1^2 est définie par les cinq covariances a,b,c,d et e ; avec la condition que $\Gamma^{Z_1^2}$ soit définie positive.

Le graphe des dépendances est alors représenté à l'aide de la figure suivante :

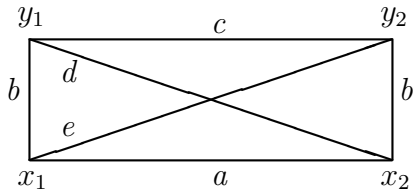


Figure 1.1 Graphe des dépendances

Considérons les matrices suivantes :

$$\text{A}=\begin{bmatrix} A^1 & A^2 \\ A^3 & A^4 \end{bmatrix} =\text{D}\Gamma^{-1} =\begin{bmatrix} a & e \\ d & c \end{bmatrix}\begin{bmatrix} 1 & b \\ b & 1 \end{bmatrix}^{-1} = \frac{1}{1-b^2}\begin{bmatrix} a\text{-}eb & \text{-}ab+e \\ d\text{-}cb & \text{-}db+c \end{bmatrix} \qquad (3)$$

et

$$\text{B}=\begin{bmatrix} B^1 & B^2 \\ B^3 & B^4 \end{bmatrix} \text{ vérifiant } \begin{bmatrix} B^1 & B^2 \\ B^3 & B^4 \end{bmatrix}\begin{bmatrix} B^1 & B^2 \\ B^3 & B^4 \end{bmatrix}^t = \Gamma - D\Gamma^{-1}D^t \qquad (4)$$

On montre alors que la loi de $\text{Z}_2 = (X_2, Y_2)$ conditionnelle à $\text{Z}_1 = (X_1, Y_1) = (x_1, y_1)$ est Gaussienne de moyenne :

$$\begin{bmatrix} A^1 & A^2 \\ A^3 & A^4 \end{bmatrix}\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \qquad (5)$$

et de variance :

$$\begin{bmatrix} B^1 & B^2 \\ B^3 & B^4 \end{bmatrix}\begin{bmatrix} B^1 & B^2 \\ B^3 & B^4 \end{bmatrix}^t = \Gamma - D\Gamma^{-1}D^t \qquad (6)$$

On peut alors écrire

$$\begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} =\begin{bmatrix} A^1 & A^2 \\ A^3 & A^4 \end{bmatrix}\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} +\begin{bmatrix} B^1 & B^2 \\ B^3 & B^4 \end{bmatrix}\begin{bmatrix} U_2 \\ V_2 \end{bmatrix} \qquad (7)$$

avec $\begin{bmatrix} U_2 \\ V_2 \end{bmatrix}$ vecteur indépendant de $\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}$, gaussien, de moyenne nulle et de matrice de covariance unité.

Finalement, le point important est que se donner (1) ou (7), avec A et B définies par (2), (3) et (4), est équivalent.

On considére alors une suite de variables gaussiennes

$\mathrm{Z}_1 = \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}, Z_2 = \begin{bmatrix} X_2 \\ Y_2 \end{bmatrix}, ..., Z_n = \begin{bmatrix} X_n \\ Y_n \end{bmatrix}$ vérifiant :

(i) $\mathrm{Z}_1 = \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}$ de moyenne nulle et de variance $\Gamma = \begin{bmatrix} 1 & b \\ b & 1 \end{bmatrix}$, (8)

(ii) $\forall n \in [\![1, N-1]\!]$, $\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} A^1 & A^2 \\ A^3 & A^4 \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} + \begin{bmatrix} B^1 & B^2 \\ B^3 & B^4 \end{bmatrix} \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix}$, (9)

avec $\begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix}$ indépendants entre eux d'une part, et indépendants de $(\mathrm{Z}_1, ..., Z_n)$ d'autre part, gaussiens, de moyennes nulles, et de matrice de covariance unité ; et N le nombre d'observations.

2 Etude du modèle : chaîne de Markov couple

2.1 Markoviannité de la suite

On cherche tout d'abord à montrer que

$$\forall n \in [\![1, N-1]\!], \quad p(z_{n+1}|z_1, ..., z_n) = p(z_{n+1}|z_n)$$

Autrement dit, que la suite $(\mathrm{Z}_n)_{n\in[\![1,N]\!]}$ est markovienne.

Ce résultat est en fait immédiat avec les considérations précédentes. En effet, si on note $\mathrm{W}_n = \begin{bmatrix} U_n \\ V_n \end{bmatrix}$, alors on a :

$$\forall n \in [\![1, N-1]\!], Z_{n+1} = AZ_n + BW_{n+1}$$

Et pour tout n$\in [\![1, N-1]\!], W_{n+1}$ est indépendant des $(\mathrm{Z}_i)_{i\in[\![1,n]\!]}$ par hypothèse.
Donc sachant $(\mathrm{Z}_1, ..., Z_n) = (z_1, ..., z_n)$, Z_{n+1} ne dépend que de $\mathrm{Z}_n = z_n$.

Et $p(\mathrm{Z}_{n+1}|Z_n = z_n) \sim \mathcal{N}(Az_n, BB^t)$

Ainsi, (8) et (9) définissent une loi unique pour le vecteur $(\mathrm{Z}_1, ..., Z_n)$.

2.2 Filtrage de Kalman

Mettons à présent en place le filtrage de Kalman. On cherche donc à exprimer $p(x_{n+1}|y_1, ..., y_n, y_{n+1})$ à partir de $p(x_n|y_1, ..., y_n)$, y_{n+1}, A et B, pour tout n$\in [\![1, N-1]\!]$.

Commençons par rappeler deux propriétés dont nous allons avoir besoin.

Proposition 2.1 *Si* $p(X_1, X_2) \sim \mathcal{N}(\begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix}, \begin{bmatrix}\Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{2,1} & \Sigma_{2,2}\end{bmatrix})$,
alors $p(X_1|X_2) \sim \mathcal{N}(\mu_{1|2}, \Sigma_{1|2})$

$$\text{où} \begin{cases} \mu_{1|2} = \mu_1 + \Sigma_{1,2}.\Sigma_{2,2}^{-1}.(X_2 - \mu_2) \\ \Sigma_{1|2} = \Sigma_{1,1} - \Sigma_{1,2}.\Sigma_{2,2}^{-1}.\Sigma_{2,1} \end{cases}$$

Proposition 2.2 *Si* $p(X_1) \sim \mathcal{N}(\mu_1, \Sigma_1)$ *et* $p(X_2|X_1) \sim \mathcal{N}(AX_1 + b, \Sigma_{2|1})$

$$\text{alors } p(X_1, X_2) \sim \mathcal{N}(\begin{bmatrix}\mu_1 \\ A\mu_1 + b\end{bmatrix}, \begin{bmatrix}\Sigma_1 & \Sigma_1 A^t \\ A\Sigma_1 & \Sigma_{2|1} + A\Sigma_1 A^t\end{bmatrix})$$

On sait que $(\mathrm{X}_n|Y_1, ..., Y_n)$ suit une loi normale, notons alors :

$$p(\mathrm{X}_n|Y_1, ..., Y_n) \sim \mathcal{N}(m_n, \Sigma_n)$$

On oubliera dans la suite le conditionnement par
$(\mathrm{Y}_1 = y_1, ..., Y_n = y_n)$ pour soulager la rédaction, mais on n'oubliera pas qu'il reste présent pour le raisonnement.

En particulier, tout se passe comme si $\mathrm{Y}_n = y_n$ était fixé. On a alors :

$$\mathrm{Z}_{n+1} = \begin{bmatrix}A_1 \\ A_3\end{bmatrix} X_n + \begin{bmatrix}A_2 \\ A_4\end{bmatrix} y_n + BW_n$$

et donc

$$\mathrm{E}[\mathrm{Z}_{n+1}] = A \begin{bmatrix}m_n \\ y_n\end{bmatrix}$$

et

$$\mathrm{Cov}(\mathrm{Z}_{n+1}) = \begin{bmatrix}A_1 \\ A_3\end{bmatrix} \underbrace{Var(X_n)}_{\Sigma_n} \begin{bmatrix}A_1 \\ A_3\end{bmatrix}^t + B \underbrace{Cov(W_{n+1})}_{I_2} B^t$$

Et donc

$$p(\mathrm{Z}_{n+1}) \sim \mathcal{N}(A \begin{bmatrix} m_n \\ y_n \end{bmatrix}, \begin{bmatrix} A_1 \\ A_3 \end{bmatrix} \Sigma_n \begin{bmatrix} A_1 \\ A_3 \end{bmatrix}^t + BB^t)$$

Notons alors

$$p(\mathrm{X}_{n+1}|Y_1, ..., Y_n, Y_{n+1}) \sim \mathcal{N}(m_{n+1}, \Sigma_{n+1})$$

On a alors d'après la proposition 2.1 :

$\mathrm{m}_{n+1} = A^1 m_n + A^2 y_n + (\Sigma_n A^1 A^3 + B^1 B^3 + B^2 B^4) \times (\Sigma_n (A^3)^2 + (B^3)^2 + (B^4)^2)^{-1} \times (y_{n+1} - A^3 m_n - A^4 y_n)$

et

$\Sigma_{n+1} = \frac{\Sigma_n[(A^1B^3 - A^3B^1)^2 + (A^1B^4 - A^3B^2)^2] + (B^1B^4 - B^2B^3)^2}{\Sigma_n(A^3)^2 + (B^3)^2 + (B^4)^2}$

Car :
$\mathrm{m}_{n+1} = A^1 m_n + A^2 y_n + (\Sigma_n A^1 A^3 + (BB^t)_{1,2}) \times (\Sigma_n (A^3)^2 + (BB^t)_{2,2})^{-1} \times (y_{n+1} - A^3 m_n - A^4 y_n)$

et $\mathrm{BB}^t = \begin{bmatrix} B^1 & B^2 \\ B^3 & B^4 \end{bmatrix} \begin{bmatrix} B^1 & B^3 \\ B^2 & B^4 \end{bmatrix} = \begin{bmatrix} (B^1)^2 + (B^2)^2 & B^1B^3 + B^2B^4 \\ B^1B^3 + B^2B^4 & (B^3)^2 + (B^4)^2 \end{bmatrix}$

puis

$$\Sigma_{n+1} = \Sigma_n(A^1)^2 + (BB^t)_{1,1} - (\Sigma_n A^3 A^1 + (BB^t)_{1,2}) \times (\Sigma_n(A^3)^2 + (BB^t)_{2,2})^{-1} \times (\Sigma_n A^3 A^1 + (BB^t)_{2,1})$$

$$= \Sigma_n(A^1)^2 + (B^1)^2 + (B^2)^2 - \frac{(\Sigma_n A^3 A^1 + B^1 B^3 + B^2 B^4)^2}{\Sigma_n(A^3)^2 + (B^3)^2 + (B^4)^2}$$

$$= \frac{\Sigma_n^2(A^1A^3)^2 + \Sigma_n(A^1B^3)^2 + \Sigma_n(A^1B^4)^2 + \Sigma_n(A^3B^1)^2 + (B^1B^3)^2}{\Sigma_n(A^3)^2 + (B^3)^2 + (B^4)^2}$$
$$+ \frac{(B^1B^4)^2 + \Sigma_n(A^3B^2)^2 + (B^2B^3)^2 + (B^2B^4)^2 - \Sigma_n^2(A^3A^1)^2 - (B^1B^3)^2}{\Sigma_n(A^3)^2 + (B^3)^2 + (B^4)^2}$$
$$- \frac{(B^2B^4)^2 + 2(\Sigma_n A^3A^1B^1B^3 + \Sigma_n A^3A^1B^2B^4 + B^1B^3B^2B^4)}{\Sigma_n(A^3)^2 + (B^3)^2 + (B^4)^2}$$

$$= \frac{\Sigma_n A^1B^3(A^1B^3 - A^3B^1) + \Sigma_n A^1B^4(A^1B^4 - A^3B^2)}{\Sigma_n(A^3)^2 + (B^3)^2 + (B^4)^2}$$
$$+ \frac{\Sigma_n A^3B^1(A^3B^1 - A^1B^3) + \Sigma_n A^3B^2(A^3B^2 - A^1B^4)}{\Sigma_n(A^3)^2 + (B^3)^2 + (B^4)^2}$$
$$+ \frac{B^1B^4(B^1B^4 - B^2B^3) + B^2B^3(B^2B^3 - B^1B^4)}{\Sigma_n(A^3)^2 + (B^3)^2 + (B^4)^2}$$

$$= \frac{\Sigma_n(A^1B^3 - A^3B^1)(A^1B^3 - A^3B^1) + \Sigma_n(A^1B^4 - A^3B^2)(A^1B^4 - A^3B^2)}{\Sigma_n(A^3)^2 + (B^3)^2 + (B^4)^2}$$
$$+ \frac{(B^1B^4 - B^2B^3)(B^1B^4 - B^2B^3)}{\Sigma_n(A^3)^2 + (B^3)^2 + (B^4)^2}$$

$$= \frac{\Sigma_n(A^1B^3 - A^3B^1)^2 + \Sigma_n(A^1B^4 - A^3B^2)^2 + (B^1B^4 - B^2B^3)^2}{\Sigma_n(A^3)^2 + (B^3)^2 + (B^4)^2}$$

Nous avons ainsi pu implémenter sous Matlab la fonction *filtragekalman* à partir de ces résultats théoriques, qui renvoie récurcivement pour k allant de 1 à N une estimation de la moyenne m_k et de la variance σ_k^2 de X_k. On initialisera alors le filtre en prenant comme première estimation la valeur observée y_1.

2.3 Une relation importante

Maintenant que nous avons implémenté notre filtre de Kalman et que nous l'avons testé avec les paramètres vrais du modèle (ceux avec lesquels on a simulé notre chaîne de Markov couple), nous allons chercher à estimer les paramètres du modèle de façon à utiliser le filtre en n'ayant à disposition que les observations $(y_1, ..., y_k)$, ce qui se rapproche des situations rencontrées dans les applications pratiques .
Nous allons alors démontrer une relation qui va être très utile afin d'y parvenir.

$$\text{Montrons que } \forall n \in [\![1, N-1]\!], \forall k \in [\![1, N-n]\!], cov(Z_n, Z_{n+k}) = A^k \Gamma$$

Ecrivons la relation (9) de la façon suivante :

$$\forall n \in [\![1, N-1]\!], Z_{n+1} = AZ_n + BW_{n+1}$$

On a alors par récurrence immédiate :

$$\forall n \in [\![1, N]\!], Z_n = A^n Z_0 + B \sum_{j=0}^{n-1} A^j W_{n-j}$$

puis

$$\forall n \in [\![1, N]\!], \forall k \in [\![1, N-n]\!], Z_{n+k} = A^k Z_n + B \sum_{j=0}^{n-1} A^j W_{n+k-j}$$

$$\text{Notons } \mathcal{U}_n^k = \sum_{j=0}^{n-1} A^j W_{n+k-j}$$

On a alors pour n $\in [\![1, N]\!]$ et pour k $\in [\![1, N-n]\!]$,

$$\begin{aligned} cov(Z_n, Z_{n+k}) &= cov(Z_n, A^k Z_n + B\mathcal{U}_n^k) \\ &= cov(Z_n, A^k Z_n) + cov(Z_n, B\mathcal{U}_n^k) \\ &= A^k cov(Z_n, Z_n) + \underbrace{B\mathrm{cov}(\mathrm{Z}_n, \mathcal{U}_n^k)}_{=0 \ par \ hypothèse} \\ &= A^k var(Z_n) \\ &= A^k \Gamma \end{aligned}$$

3 Estimation des paramètres

On considère donc maintenant le problème de l'estimation des covariances a, b, c, d et e.

3.1 Calcul exacte des covariances

Calculons pour k$\in \{1, ..., 5\}$ les covariances $\rho_k = E[Y_n Y_{n+k}]$.

Soit k $\in [\![1, 5]\!]$.

$$\begin{aligned} \rho_k &= E[Y_n Y_{n+k}] \\ &= E[Y_n Y_{n+k}] - \underbrace{E[Y_n]}_{=0}\underbrace{E[Y_{n+k}]}_{=0} \\ &= cov(Y_n, Y_{n+k}) \end{aligned}$$

Et $\mathrm{Cov}(\mathrm{Z}_n, Z_{n+k}) = \begin{bmatrix} cov(X_n, X_{n+k}) & cov(X_n, Y_{n+k}) \\ cov(Y_n, X_{n+k}) & cov(Y_n, Y_{n+k}) \end{bmatrix}$

– <u>k=1</u> :

$$\begin{aligned} Cov(Z_n, Z_{n+1}) &= Cov(Z_1, Z_2) \ \ (par \ stationarité) \\ &= \begin{bmatrix} a & e \\ d & c \end{bmatrix} \end{aligned}$$

D'où $\rho_1 = c$

– k=2 :

$$\begin{aligned} Cov(Z_n, Z_{n+2}) &= A^2\Gamma \\ &= AD \\ &= \frac{1}{1-b^2}\begin{bmatrix} a\text{-}eb & -ab+e \\ d\text{-}cb & -db+c \end{bmatrix}\begin{bmatrix} a & e \\ d & c \end{bmatrix} \\ &= \frac{1}{1-b^2}\begin{bmatrix} a^2\text{-}eab\text{-}dab+ed & ea\text{-}e^2b\text{-}abc+ec \\ ad\text{-}abc\text{-}d^2b+cd & c^2\text{-}cdb\text{-}cbe+ed \end{bmatrix} \end{aligned}$$

D'où $\rho_2 = \frac{c^2\text{-}cdb\text{-}cbe+ed}{1-b^2}$

– k=3 :

On trouve par le calcul

$$\rho_3 = \frac{ead\text{-}e^2bd\text{-}abcd+ecd\text{-}eabc+e^2b^2c+ab^2c^2\text{-}ebc^2\text{-}dbc^2+cb^2d^2+cd^2be\text{-}ed^2b+c^3-c^2db\text{-}c^2de+ced}{(1-b^2)^2}$$

Les expressions de ρ_4 et ρ_5 se complexifient davantage. Ces covariances vont être facilement estimables, comme on le verra dans le paragraphe suivant ; on comprend néanmoins qu'inverser le système d'équations dans le cas général pour exprimer les paramètres du modèle en fonction des covariances s'avère fastidieux. On se limitera donc dans un premier temps au cas classique où on se ramère à la modélisation des chaînes de Markov cachées.

3.2 Estimation des covariances

On utilisera l'estimateur empirique classique des ρ_k suivant :

$$\widehat{\rho_k} = \frac{1}{N-k}\sum_{j=1}^{N-k}(Y_j - \frac{1}{N}\sum_{i=1}^{N}Y_i)(Y_{j+k} - \frac{1}{N-k}\sum_{i=1}^{N-k}Y_{i+k})$$

3.3 Cas du modèle classique

On suppose donc dans ce modèle que certains paramètres sont connus par rapport à notre modélisation initiale :
a et b sont quelconques, mais $e{=}d{=}ab$ et $c{=}ab^2$
On retrouve alors le graphe des dépendances des chaînes de Markov cachées suivant :

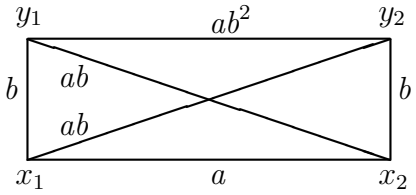


Figure 3.1 Graphe des dépendances dans le modèle classique

dont le graphe minimal est donné par :

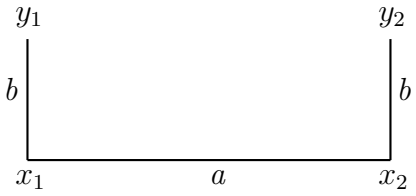


Figure 3.2 Graphe minimal des dépendances dans le modèle classique

On a alors

$$\begin{aligned} A &= \frac{1}{1-b^2}\begin{bmatrix} a\text{-}ab^2 & \text{-}ab+ab \\ ab\text{-}ab^3 & \text{-}ab^2+ab^2 \end{bmatrix} \\ &= \begin{bmatrix} a & 0 \\ ab & 0 \end{bmatrix} \\ &= a\begin{bmatrix} 1 & 0 \\ b & 0 \end{bmatrix} \end{aligned}$$

Et

$$\begin{aligned} Cov(Z_n, Z_{n+2}) &= A\Gamma \\ &= a \begin{bmatrix} 1 & 0 \\ b & 0 \end{bmatrix} \begin{bmatrix} 1 & b \\ b & 1 \end{bmatrix} \\ &= a \begin{bmatrix} 1 & b \\ b & b^2 \end{bmatrix} \end{aligned}$$

D'où $\rho_1 = ab^2$

Puis

$$\begin{aligned} Cov(Z_n, Z_{n+1}) &= A^2\Gamma \\ &= a^2 \begin{bmatrix} 1 & 0 \\ b & 0 \end{bmatrix} \begin{bmatrix} 1 & b \\ b & 1 \end{bmatrix} \\ &= a^2 \begin{bmatrix} 1 & b \\ b & b^2 \end{bmatrix} \end{aligned}$$

D'où $\rho_2 = a^2b^2$

De la même façon, on trouve :
$\rho_3 = a^3b^2$
$\rho_4 = a^4b^2$
$\rho_5 = a^5b^2$

Les paramètres du modèle s'expriment donc ici simplement en fonction des covariances et sont donc faciles à estimer :
On a $a{=}\frac{\rho_2}{\rho_1}$, $b = \frac{\rho_1}{\sqrt{\rho_2}}$, $c = \rho_1$, et $d{=}e{=}\sqrt{\rho_2}$

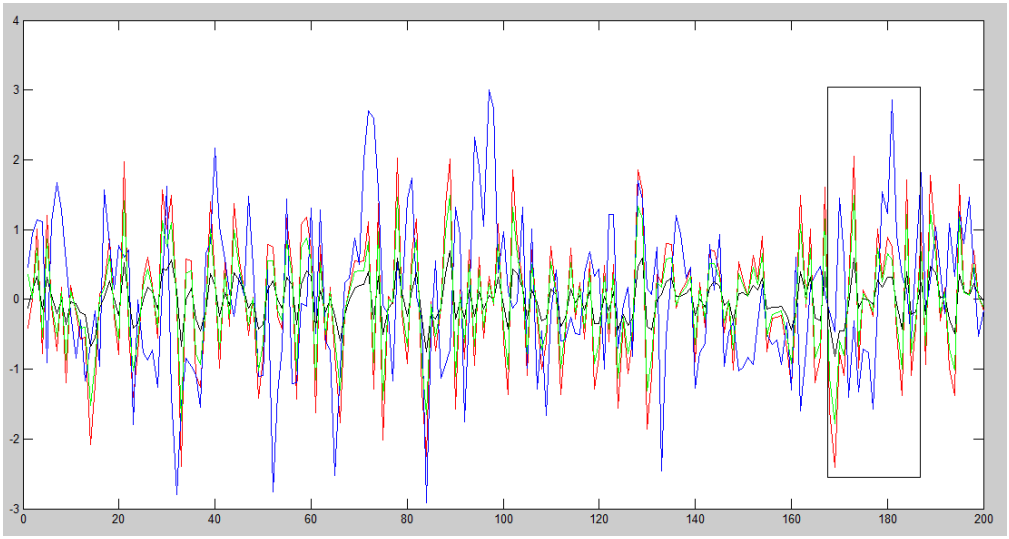
4 Simulations dans le cas classique

4.1 Filtrage supervisé

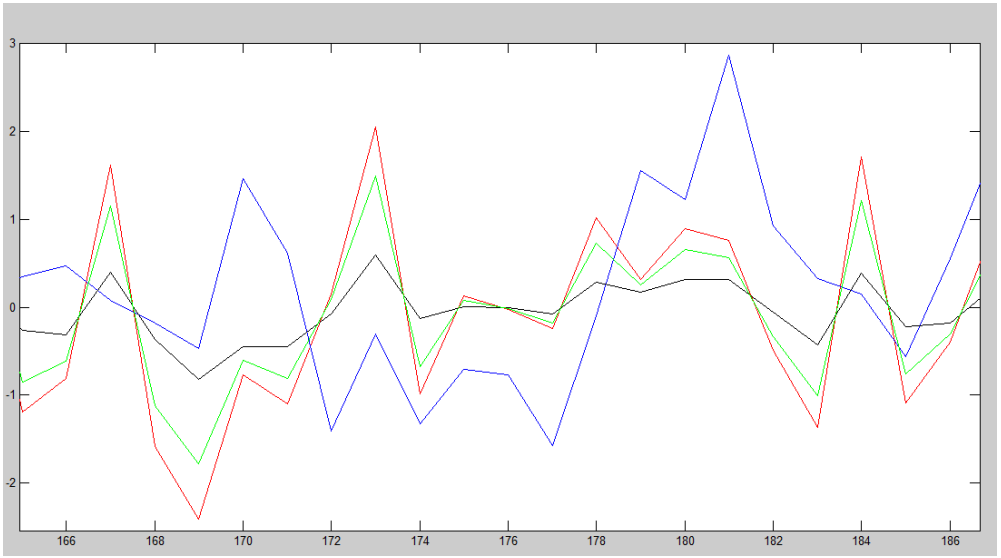
Dans le cas de la classification supervisée, on dispose d'un échantillon d'apprentissage $(z_1, z_2, ..., z_N)$ où pour i $\in [\![1, N]\!]$, $z_i = (x_i, y_i)$. Les paramètres peuvent alors être estimés par les estimateurs classiques et on procède ensuite à l'estimation Bayésienne des variables cachées à partir des variables observées en utilisant les paramètres estimés.
Nous n'avons pas fait de simulations dans ce cas pour se concentrer d'avantage sur le filtrage non supervisé qui est plus intéressant.

4.2 Filtrage non supervisé

Dans le cas de la classification non supervisée, on ne dispose que d'un échantillon $(y_1, y_2, ..., y_N)$ sans échantillon d'apprentissage. On doit alors estimer tous les paramètres à partir de cet échantillon.
Le chapitre précédent présente les méthodes que nous avons utilisé pour y parvenir dans le cas classique.
Nous avons alors pu faire de nombreuse simulations dont nous allons présenter les plus parlantes.
Commençons par présenter une simulation où le bruit entre les variables cachées et les variables observées est relativement important, on prendra par exemple a=0.3 et b=0.3.
On obtient le graphe suivant pour une taille d'échantillon de 200 :

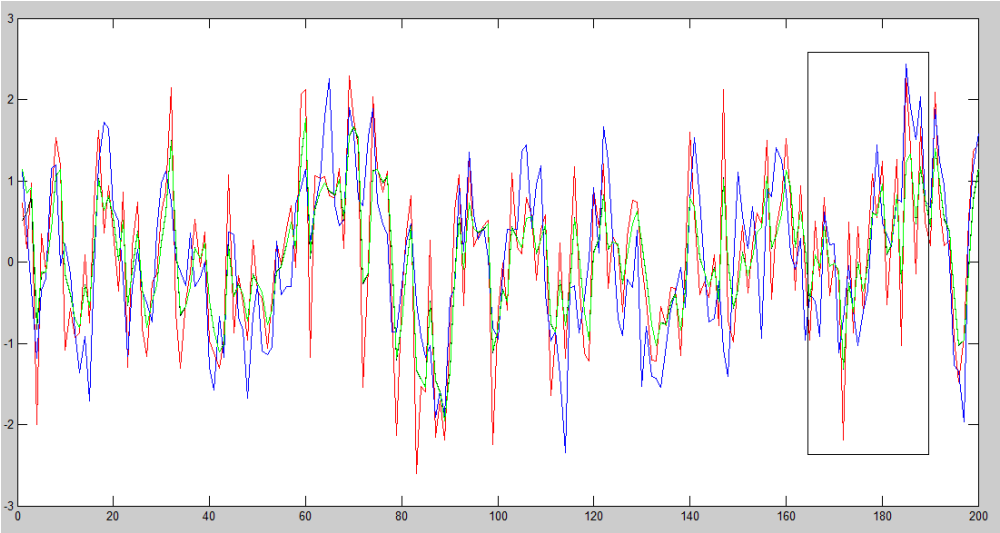


Signal observé (bruité) Signal caché Signal filtré avec les vrais paramètres Signal filtré en non supervisé

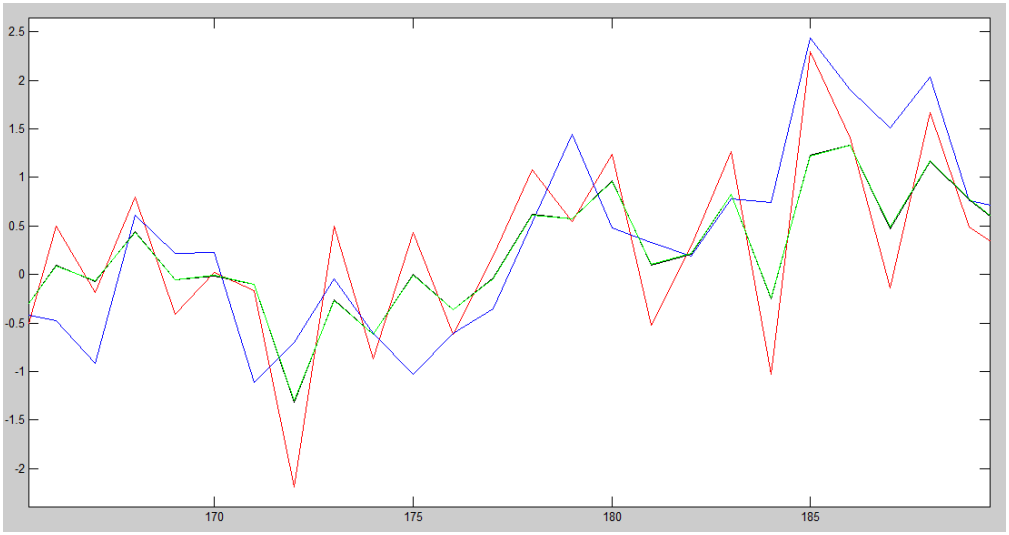


On se rend donc compte qu'en moyenne, la courbe noire est entre la courbe rouge et la courbe bleue, ce qui signifie que le filtrage de Kalman est efficace ; et que la courbe verte est entre la courbe noire et la courbe bleue, ce qui est normal car le filtrage représenté par la courbe noire est meilleur

que celui représenté par la courbe verte car ce dernier est réalisé avec les paramètres estimés du modèle alors que la courbe noire est obtenue avec les vrais paramètres.
Si on réduit maintenant le bruit, on peut faire les mêmes observations à ceci près que la qualité du filtrage non supervisé s'améliore : la courbe verte est quasimment confondue avec la courbe noire.
Par exemple en prenant a=0.7 et b=0.7, on obtient les tracés suivants :



Puis en zoomant sur le rectangle noir :


2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
170
175
180
185

5 Conclusion

Nous sommes donc parvenus à effectuer un filtrage non supervisé dans le cas classique et les résultats que nous obtenons sont très satisfaisants et encourageants. Les perspectives en vue sont tout d'abord d'étendre ce travail au cas général en utilisant un logiciel capable d'inverser des systèmes linéaires complexes, puis d'introduire une troisième variable, discréte, que l'on appellera variable de saut, qui modélisera les instants de changement de paramètres du modèle.
Nous serons donc en mesure d'approcher n'importe quel système non linéaire par une succession de filtrages linéaires tels que ceux mis en place dans ce projet.

Il existe déjà des traitements possibles pour les systèmes non linéaires, comme le filtrage particulaire, mais la complexité de calcul explose rapidement lorsque le nombre de particules augmente. Il sera alors intérressant de confronter les résultats de la suite de nos travaux avec ceux du filtrage particulaire, et qui, en toute vraissemblance, devraient être de meilleure qualité.

Il est important de préciser que nos travaux trouveront de nombreuses applications dans divers domaines technologiques.

À titre d'exemple :

- Poursuite de cible à partir d'un signal radar : position et vitesse exactes sont cachées, leurs versions bruitées sont observées
- En finance : prévision de la volatilité stochastique et du rapport cours-bénéfices
- Ou encore en météorologie, océanographie, trafic routier, communications numériques, ...

On comprend dès lors l'importance que peuvent prendre les résultats futurs.

6 Annexes

6.1 Script des fonctions Matlab implémentées

```
clear all
close all
clc

% Initialisation :

N=200 ; % Taille de l'échantillon
% Paramètres choisis (cas classique) :
a=0.7 ;
b=0.7 ;
c=a*b^2;
d=a*b ;
e=a*b ;
% Matrices des covariances :
G=[1 b ; b 1] ;
D=[a e ; d c] ;
GZ2=[G D' ; D G] ;
A=D/G ;
C=G-(D/G)*D' ; % C=BB'
B=chol(C) ;


% construction de la chaîne de Markov couple :

% version 1 :

   % Initialisation :

   x1 = randn(1);
   y1 = b * x1 + sqrt(1 - b^2) * randn(1);
   Z = zeros(2, N);
   Z(:, 1) = [x1; y1];
```

```
   for i = 2 : N
      Z(2,i) = A(2,:) * Z(:,i-1) + sqrt(C(2,2)) * randn(1);
      Z(1,i) = A(1,:) * Z(:,i-1) + C(1,2) * (Z(2,i) - A(2,:) * Z(:
,i-1))/C(2,2) + sqrt(C(1,1) - C(1,2)^2/C(2,2)) * randn(1);
   end

% version 2 :

   %for i = 2 : N
   %W = randn(2,1);
   %Z(:,i) = A * Z(:,i-1) + B * W;
   %end

% version 3 :
   %R = chol(G);
   %Z = (randn(N,2) * R)';


% version 4 :

   %Zn = Z(:,1);
   %Znsuiv = ProbaTransit(Zn, A, B)


% Filtrage de Kalman avec les vrais paramètres :

   %Initialisation :

   m_1 = x1;
   sigma_1 = 0;
   y_1 = y1;
   m_n = m_1;
   sigma_n = sigma_1;
   cov_n_n_1 = D;
   rho_1 = cov_n_n_1(2,2);

   Xest = zeros(1,N);
   Xest(1) = b * y_1;
```

```
for n = 2 : N
[m_n sigma_n] = filtrage kalman(A, C, Z(2, n), Z(2, n-1), m_n, sigma_n);
    Xest(n) = m_n;

end
```

```
% Tracé des courbes :

plot(Z(2, :),'-r') ; % Signal observé (bruité) en rouge
hold on
plot(Z(1, :),'-b') ; % Signal caché en bleu
hold on
plot(Xest,'-k') ; % Signal filtré par Kalman à partir des vrais paramètres en
noir


% Calcul exact des rho_k :

%   rho_k = zeros(1, 5);

%   for k = 1 : 5
%       cov_n_n_k = calc_cov(A, G, k);
%       rho_k(k) = cov_n_n_k(2, 2);
%   end


% Estimation des rho_k :

   M = 100; %On simule M fois de suite la chaîne de Markov et on calcule la
     %moyenne des M estimations des rho_k pour améliorer la précision
   rho_k_est = zeros(M, 5);

   for f = 1 : M

      %Simulation de la chaîne de Markov :
```

```
%initialisation :

x_1 = randn(1);
y_1 = b * x1 + sqrt(1 - b^2) * randn(1);
R = zeros(2, N);
R(:, 1) = [x_1; y_1];

for i = 2 : N
    R(2, i) = A(2, :) * R(:, i - 1) + sqrt(C(2, 2)) * randn(1);
    R(1, i) = A(1, :) * R(:, i - 1) + C(1, 2) * (R(2, i) - A(2, :) * R(:
, i - 1))/C(2, 2) + sqrt(C(1, 1) - C(1, 2)^2/C(2, 2)) * randn(1);
end

%Estimationdesrho_k :

temp = zeros(1, 5);

for k = 1 : 5

    for j = 1 : N - k

  temp(j) = ((R(2, j+k) - sum(R(2, :))/N) * (R(2, j)) - sum(R(2, :))/N);
    temp(1);
end

    rho_k_est(f, k) = (1/(N - k)) * sum(temp);

end

end

moyenne_rho_kest = mean(rho_k_est, 1);


% Estimation des paramètres à partir des estimations des rho_k

a_est=moyenne_rho_k_est(2)/moyenne_rho_k_est(1) ;
b_est=moyenne_rho_k_est(1)/sqrt(moyenne_rho_k_est(2)) ;
```

```
c_est=moyenne_rho_k_est(1) ;
d_est=sqrt(moyenne_rho_k_est(2)) ;
e_est=sqrt(moyenne_rho_k_est(2)) ;


% Comparaison des paramètres vrais et estimés :

disp('Comparaison des paramètres vrais et estimés :') ;
disp(' a : b : c : d : e :') ;
disp([a b c d e]) ;
disp(' a_est : b_est : c_est : d_est : e_est :');
disp([a_est b_est c_est d_est e_est]);


% Filtrage de Kalman non supervisé :

    %Initialisation :
    x1_est = randn(1);
    y1_est = b_est * x1_est + sqrt(1 - b_est^2) * randn(1);
    Z_1 = zeros(2, N);
    Z_1(:, 1) = [x1_est; y1_est];

    G_est = [1 b_est; b_est 1];
    D_est = [a_est e_est; d_est c_est];
    GZ2_est = [G_est D_est'; D_est G_est];
    A_est = D_est/G_est;
    C_est = G_est - (D_est/G_est) * D_est';
    B_est = chol(C_est);

    X_est_2 = zeros(1, N);
    X_est_2(1) = b_est * y1_est;

    m_n_2 = x1_est;
    sigma_n_2 = 0;


        for n = 2 : N
```

$[m\_n\_2 sigma\_n\_2] =$
$filtragekalman(A\_est, C\_est, Z(2,n), Z(2,n-1), m\_n\_2, sigma\_n\_2);$
$X\_est\_2(n) = m\_n\_2;$

end

% Ajout du tracé du filtrage non supervisé, en vert

hold on
plot(X_est_2,'-g') ;

% Calcul des erreurs quadratiques :

disp('E_Q_1 : Erreur Quadratique entre les données cachées et les données observées') ;
disp('E_Q_2 : Erreur Quadratique entre les données cachées et les données restaurées par Kalman (supervisé)') ;
disp('E_Q_3 : Erreur Quadratique entre les données cachées et les données restaurées par Kalman (non supervisé)') ;
disp('E_Q_4 : Erreur Quadratique entre les données restaurées par filtrage supervisé et non supervisé') ;
disp(' ') ;
disp(' E_Q_1 : E_Q_2 :') ;

disp([sqrt((Z(1, :) - Z(2, :))*(Z(1, :) - Z(2, :))') sqrt((Z(1, :) - Xest)*(Z(1, :) - Xest)')]) ;

disp(' E_Q_3 : E_Q_4 :') ;
disp([sqrt((Z(1, :) - X_est_2)*(Z(1, :) - X_est_2)') sqrt((Xest - X_est_2)*(Xest - X_est_2)')]) ;

Puis la fonction *filtragekalman* :
`function`
$[m_k sigma_k] = filtragekalman(A, C, y\_k, y\_kprec, m\_kprec, sigma\_kprec)$

m_k=A(1,1)*m_kprec+A(1,2)*y_kprec+((sigma_kprec*A(1,1)*A(2,1)+C(1,2))*(y_k-A(2,1)*m_kprec-A(2,2)*y_kprec))/(sigma_kprec*A(2,1)^2 + $C(2,2)$);

sigma_k=sigma_kprec*A(1,1)^2 + $C(1,1)$ − (($sigma\_kprec * A(2,1) * A(1,1) + C(1,2))^2$)/($sigma\_kprec * A(2,1)^2 + C(2,2)$);

end

Et enfin la fonction *calc_cov quicalculelescovariances* :

```
function [ cov_n_n_k ] = calc_cov(A, G, k)
```

$cov\_n\_n\_k = A^k * G$;

end

6.2 Poster

TELECOM SudParis
Institut Mines-Télécom

Filtrage statistique optimal non supervisé dans une chaîne de Markov couple

Equipe 80 Recherche

Auteurs

BUSSY Simon
COSNARD Emmanuel
DIARRASSOUBA Mamery

Encadrant

PIECZYNSKI Wojciech

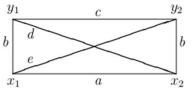
Enjeux du projet

- Un problème important dans une large gamme de domaines consiste à pouvoir estimer un ensemble de variables inobservables, dites cachées, à partir d'un ensemble de variables observées qui peuvent être interprétées comme des versions bruitées des variables cachées.
- L'objectif est alors de retrouver les données cachées avec la meilleure précision possible en éliminant l'effet du bruit, ce qui peut être fait avec un filtre de Kalman (FK).
- Classiquement, le FK est utilisé dans le cadre d'une chaîne de Markov cachée.
- Notre projet consiste à étudier le problème dans le cadre d'un modèle récent plus général, celui d'une chaine de Markov couple:

$$\forall n \in \mathbb{N},\ \begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} A^1 & A^2 \\ A^3 & A^4 \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} + \begin{bmatrix} B^1 & B^2 \\ B^3 & B^4 \end{bmatrix} \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix}$$

$$\Gamma^{Z_1^2} = \begin{bmatrix} 1 & b & a & d \\ b & 1 & e & c \\ a & e & 1 & b \\ d & c & b & 1 \end{bmatrix} = \begin{bmatrix} \Gamma & D^t \\ D & \Gamma \end{bmatrix}$$

$$\text{Avec } \Gamma = \text{Cov}(Z_1) = \begin{bmatrix} 1 & b \\ b & 1 \end{bmatrix} \text{ et } D = \begin{bmatrix} a & c \\ d & c \end{bmatrix}$$



Matrice de covariance de (X1, Y1, X2, Y2) et graphe des dépendances, cas stationnaire

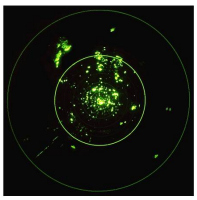
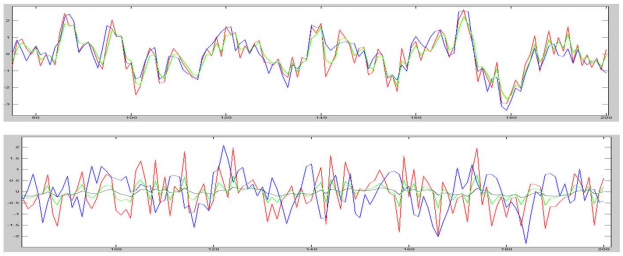


Photo (a)

Contributions réalisées

- Etude du filtre dans le cadre du modèle couple
- Introduction et programmation des estimateurs des paramètres du modèle à partir des seules observations
- Proposition d'une méthode originale de filtrage non supervisé et son étude numérique (simulations en Matlab)



Signal observé (bruité) Signal caché Signal filtré (FK) Signal filtré non supervisé (FK)
Sur le premier graphe, la qualité du filtrage est bonne, elle l'est beaucoup moins sur le second où nous avons augmenté le bruit.

Les applications

Nos travaux trouveront de nombreuses applications dans divers domaines technologiques. A titre d'exemple:

- Poursuite de cible à partir d'un signal radar (a) : position et vitesse exactes sont cachées, leurs versions bruitées sont observées
- En finance: prévision de la volatilité stochastique et du rapport cours-bénéfices
- Ou encore en météorologie (b), océanographie, trafic routier, communications numériques, …

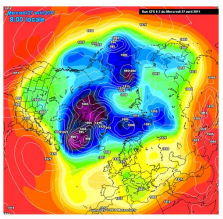


Photo (b)

Contacts: simon.bussy@telecom-sudparis.eu http://www.telecom-sudparis.eu