



M1 ISIFAR - TPE

FILTRATION STOCHASTIQUE
APPLIQUÉE AUX PROBLÈMES
D'INFORMATIONS PARTIELLES

MÉHEUST WILLIAM

RAGAVAN RANUSHAN

Dans le cadre de l'UE MA1BY010 Mathématiques financières

Sous la direction de Noufel FRIKHA et Zorana GRBAC

Année universitaire 2021-2022

Table des matières

1	Introduction	2
2	Modèle de Markov Caché et Systèmes Gaussiens	3
2.1	Définitions	3
2.1.1	Systèmes Gaussiens	3
2.1.2	Chaînes de Markov	3
2.1.3	Modèle de Markov Caché	4
2.2	Exemple	5
3	Prédictions et Filtres Stochastiques	6
3.1	Filtrage dans le cas discret	6
3.2	Cas Stationnaires	7
3.2.1	ARMA	7
3.3	Cas Non-Stationnaires	10
3.3.1	ARIMA	10
4	Filtre de Kalman	11
4.1	Chaînes de Markov Cachées et Filtre de Kalman	11
4.2	Algorithme de filtration	11
4.2.1	Prédictions	12
4.2.2	Corrections	13
4.3	Lien entre filtre de Kalman et Modèle ARMA	14
5	Application du filtre de Kalman	15
5.1	Codage du filtre de Kalman	15
5.1.1	Fonction d'initialisation	15
5.1.2	Fonction de prédition	15
5.1.3	Fonction de correction	15
5.2	Exemple d'application du filtre de Kalman	17
6	Conclusion	19
7	Bibliographie	20
7.1	Ouvrages et Articles	20
7.2	Liens	20

1 Introduction

Le problème de données manquantes est un problème récurrent en ingénierie, en finance et en science de la vie et les raisons sont nombreuses. Si on considère le cas de mesures issues de capteurs, un des capteurs peut fonctionner de façon intermittente ou alors la transmission des mesures peut subir des perturbations ponctuelles. Quel est l'impact de l'absence de certaines données sur la construction de modèles à partir des données et sur la prise de décision avec ces modèles ? Il s'agit d'un problème, car outre le fait que leurs absences diminuent la puissance de l'étude, elles peuvent être à l'origine de biais importants.

On distingue trois types de données manquantes:

1. Données manquantes de façon complètement aléatoire (**missing completely at random, MCAR**). Dans ce cas, la probabilité d'absence d'une donnée est identique pour toute observation . Dans le cas MCAR, ignorer les observations à données manquantes équivaut à utiliser un échantillon aléatoire des observations. En présence de classes déséquilibrées, cette réduction du nombre d'observations exploitables pour la modélisation peut avoir un impact significatif car les classes les plus rares peuvent être représentées par un nombre très insuffisant d'observations ou bien même disparaître.
2. Données manquantes de façon aléatoire (**missing at random, MAR**). Dans ce cas, la probabilité d'absence de la valeur d'une variable est indépendante de la valeur qu'aurait eu cette variable si elle n'avait pas été absente. Mais cette probabilité dépend des valeurs prises par d'autres variables qui ont été observées. Dans le cas MAR, ignorer les observations à données manquantes introduit un biais dans la modélisation. En effet, dans ce cas la valeur d'une variable manque lorsque d'autres variables, observées, prennent certaines valeurs. Ignorer les observations incomplètes revient à ignorer la plupart des occurrences de ces combinaisons de valeurs pour ces variables observées.
3. Données manquantes de façon non aléatoire (**missing not at random, MNAR**). Dans ce cas, la probabilité d'absence de la valeur d'une variable dépend de variables qui n'ont pas été observées. Dans le cas MNAR, ignorer les observations à données manquantes introduit un biais dans la modélisation car cela revient à éliminer des observations de façon non aléatoire.

Pour cela nous disposons de nombreux outils mathématiques tels que les filtres stochastiques. Dans la suite nous introduirons les modèles de Markov cachés et son lien avec les filtres stochastiques, pour finir nous étudierons différents exemples d'applications.

2 Modèle de Markov Caché et Systèmes Gaussiens

2.1 Définitions

2.1.1 Systèmes Gaussiens

La résolution de systèmes est un problème récurrent des mathématiques. Parmi les nombreuses méthodes pour les résoudre, figure le *pivot de Gauss*. Le pivot de Gauss est une méthode de résolution de systèmes en transformant un système en système triangulaire. Cette méthode autorise la permutation des lignes , la multiplication d'une ligne par un coefficient non nul, et l'addition d'un multiple d'une ligne à une autre ligne.

2.1.2 Chaînes de Markov

Les **chaînes de Markov** forment une classe particulière de processus stochastiques dont la particularité est que l'évolution future du processus (l'état suivant de la chaîne) ne dépend que de l'état présent. En particulier, le processus ne garde pas en mémoire la façon dont l'état présent a été atteint, mais seulement l'état qu'il a visité juste avant.

Un exemple typique est celui de la marche aléatoire sur le réseau \mathbf{Z}^2 (ou plus généralement \mathbf{Z}^d pour $d \geq 1$). Le marcheur commence par se placer à l'origine du réseau, puis se déplace en choisissant uniformément un sommet parmi les sommets voisins de sa position actuelle.

Les chaînes de Markov peuvent être décrites simplement à l'aide d'un graphe orienté.

Par exemple :

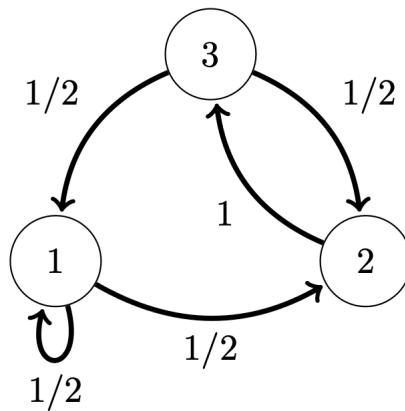


FIG. 1 – *Chaîne de Markov classique*

Ce graphe décrit l'évolution d'une *chaînes de Markov* sur l'espace d'états $\{1,2,3\}$, c'est-à-dire que les valeurs prises par le processus sont 1, 2 ou 3.

Au temps 0, la chaîne X_0 se trouve dans un état inconnu parmi $\{1,2,3\}$. Si la chaîne se trouve en l'état 1, alors au temps suivant elle se déplace en 2 avec probabilité $1/2$ ou bien reste en 1 avec probabilité $1/2$. Si elle se trouve en l'état 2, elle se déplace avec probabilité 1 en 3. Enfin, si elle se trouve en l'état 3, elle saute à la prochaine étape soit en 1 ou bien en 2 avec la probabilité $1/2$.

2.1.3 Modèle de Markov Caché

Un **modèle de Markov caché** est un modèle statistique dans lequel le système modélisé est supposé être un processus markovien de paramètres inconnus. Contrairement à une *chaînes de Markov* classique, où les transitions prises sont inconnues de l'utilisateur mais où les états d'une exécution sont connus, dans un modèle de Markov caché, les états d'une exécution sont inconnus de l'utilisateur.

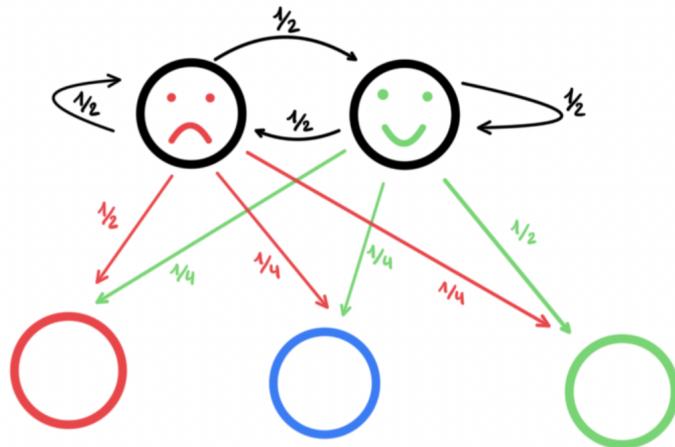


FIG. 2 – Modèle de markov caché

Nous distinguons sur cette figure deux types d'états. Les cercles colorés correspondent aux états observables, tandis que les deux cercles noirs forment une *chaîne de Markov* classique qui est non observable.

Pour mieux comprendre la notion de *modèle de markov caché* nous allons faire l'analogie suivante. Une personne a deux humeurs différentes $\{\text{content}, \text{pas content}\}$ et en fonction de son humeur elle va plutôt mettre un t-shirt $\{\text{rouge}, \text{bleu}, \text{vert}\}$. Ainsi l'humeur est l'état caché et la couleur du t-shirt est le paramètre observable. On comprend alors que les couleurs des t-shirts portés précédemment n'influent pas sur la couleur du t-shirt qui sera porté. Il n'y a que les humeurs qui déterminent quel t-shirt sera porté (en fonction de leurs probabilités de passage).

2.2 Exemple

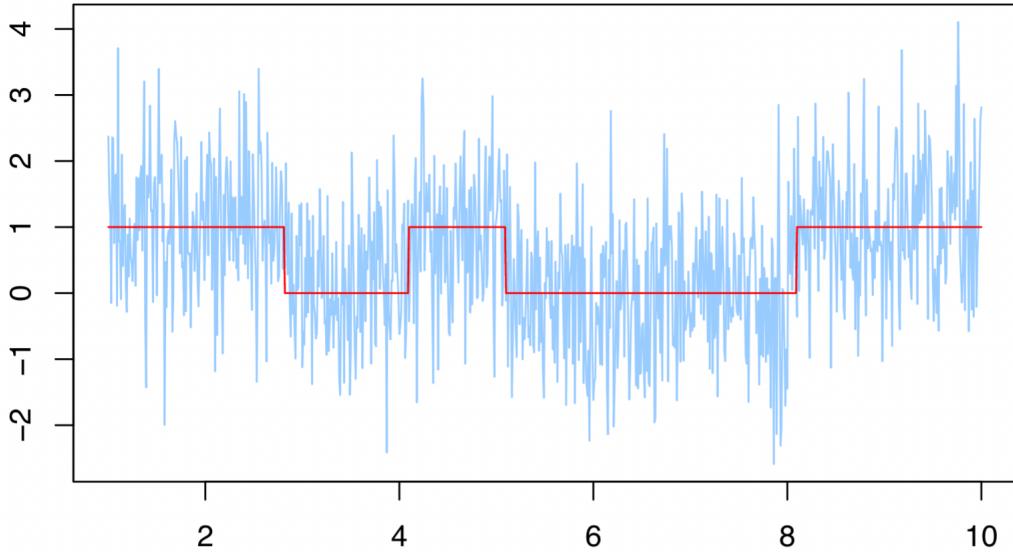


FIG. 3 – Déformation d'un signal par un bruit blanc gaussien

Définition : Un bruit est un parasitage du processus initial par des variables aléatoires qui suivent des lois de probabilités. Ce bruit est dit blanc si les variables aléatoires sont i.i.d.

Dans cet exemple, nous allons donner une première illustration du lien entre un signal et une chaîne de Markov. En rouge figure le signal émis. N'étant généralement pas directement perceptible, il est nécessaire d'utiliser différents types d'outils de mesures tels que les gyroscopes ou accéléromètres, afin de le matérialiser.

Cependant, ces outils possèdent un inconvénient majeur : ils sont sujets aux bruits. Ainsi, le signal relevé par l'outil de mesure ne nous donne pas exactement le signal d'origine. Nous obtenons le signal représenté en bleu qui est équivalent au signal tracé en rouge combiné à un bruit modélisé par des variables aléatoires indépendantes suivant une loi normale centrée réduite (bruit blanc). Notre signal rouge est donc un modèle de Markov caché, de par le fait qu'il n'est pas directement observable mais influe directement le signal observé.

Dans la suite de ce devoir, nous aurons pour objectif de présenter des méthodes utilisant les propriétés des chaînes de Markov et des propriétés de dépendance entre les deux signaux, afin de corriger les déformations du signal observé.

3 Prédictions et Filtres Stochastiques

Dans cette section nous allons introduire les filtres stochastiques qui sont des outils fortement utiles dans la réduction de bruit.

3.1 Filtrage dans le cas discret

Soit X_t et Y_t deux processus discrets de second ordre, X est un processus inconnu tandis que Y correspond aux observations corrélées à X . Cette corrélation nous offre des informations sur le processus X qui nous est inconnu; on pourra faire une analogie entre X et le modèle de Markov caché introduit dans la partie II.

On peut alors tenter d'exprimer les X_t en fonction des Y_t . Les mesures étant réalisées aux instants $t_1, t_2, \dots, t_k \in]-\infty; t]$, on cherche une fonction borelienne f de C^k dans C tel que la variable aléatoire $Z = f(Y_{t_1}, \dots, Y_{t_k})$ soit une approximation des X_t que nous cherchons à déterminer.

Nous allons ensuite évaluer la distance entre l'approximation Z et le processus caché X . Pour cela nous passerons par la norme de leur différence dans $L^2(\Omega, \mathcal{A}, P)$. Le problème revient à trouver la projection de X_t sur le sous espace vectoriel fermé des fonctions de $\sigma(Y_{t_1}, \dots, Y_{t_k})$ -mesurables ou $\sigma(Y_s, s \leq t)$ -mesurables. Des résultats de l'espérance conditionnelle nous garantissent l'existence et l'unicité de solutions.

$$Z = E[X_t | Y_{t_1}, \dots, Y_{t_k}] \quad (1)$$

Ce problème se décompose en 2 familles :

- Les problèmes de **prédictions** correspondant aux cas particuliers où $X_t = Y_{t+h}$. Ils reviennent à estimer Y_{t+h} grâce à Y_s pour $s \leq t$.
- Les problèmes de **filtrations**, au coeur de notre problématique, correspondent aux problèmes de la forme $Y_t = X_t + U_t$ où les U_t sont des bruits.

Ainsi, notre Z s'écrit sous la forme :

$$Z = \sum_{j=1}^k a_j Y_{t_j}$$

En pratique, la solution est donnée par l'espérance conditionnelle, cependant elle est difficilement calculable avec les caractéristiques du couple (X, Y) (covariance).

De ce fait, nous sommes contraint de chercher une fonction linéaire et non plus mesurable afin d'estimer X_t . La condition minimale $\|Z - X_t\| L^2$, revient à dire que Z est la projection de X_t sur le sous-espace (de dimension finie donc fermée) de L^2 formé des combinaisons linéaires des Y_{t_j} .

Z est donné par le système d'équations linéaires :

$$E[ZY_{t_j}] = E[X_t Y_{t_j}] \quad j = 1, \dots, k \quad (2)$$

Le système (2) permettra de déterminer les coefficients a_j (qui ne seront uniques que si les Y_{t_j} engendrent un espace de dimension k) en fonction des covariances $K_Y(t_i, t_j)$ et $K_{X,Y}(t, t_j)$.

Il est important de noter que, si le couple X, Y est un processus gaussien alors les variables

$$(Z - X_t, Y_{t_j}) \quad j = 1, \dots, k$$

sont gaussiennes dans leurs ensembles, et l'orthogonalité de $Z - X_t$ et des Y_{t_j} entraînent leurs indépendances de sorte que :

$$Z = E[X_t | Y_{t_1}, \dots, Y_{t_k}]$$

Ainsi, dans le cas gaussien, l'approximation mesurable et l'approximation linéaire coïncident.

La problématique principale issue de la théorie des algorithmes de filtrages est d'obtenir des formules explicites pour la projection de X_t sur le sous-espace vectoriel fermé, engendré par les Y_s , $s \leq t$. (Des systèmes tels que (2) sont pratiquement inutilisables si le nombre d'observations est grand).

3.2 Cas Stationnaires

3.2.1 ARMA

Définition : Soit un processus temporel à valeurs réelles et en temps discret Z_1, Z_2, \dots, Z_t . Il est dit stationnaire au sens fort si pour toute fonction f mesurable :

$$f(Z_1, Z_2, \dots, Z_t) \text{ et } f(Z_{1+k}, Z_{2+k}, \dots, Z_{t+k})$$

ont la même loi.

Définition : On appelle le processus auto-regressif de moyenne mobile d'ordre (p,q) ou processus **ARMA** (p,q) (*Auto Regressive Moving Average*) un processus stationnaire X_n vérifiant :

$$X_n + \sum_{j=1}^p \alpha_j X_{n-j} = \beta_0 W_n + \sum_{k=1}^q \beta_k W_{n-k} \quad \beta_0 \neq 0 \quad (3)$$

Nous supposerons le processus X réel, ainsi que le bruit blanc W_n et les coefficients α_j et β_k .

PROPOSITION :

Si les polynômes

$$\alpha(z) = 1 + \sum_{j=1}^p \alpha_j z^j \quad \text{et} \quad \beta(z) = \sum_{k=0}^q \beta_k z^k$$

ont leurs zéros de module > 1.

On a :

$$W_n = \sum_{j=0}^{\infty} a_j X_{n-j} \quad (4)$$

et

$$X_n = \sum_{k=0}^{\infty} b_k W_{n-k} \quad (5)$$

où

$$A(z) = \sum_{j=0}^{\infty} a_j z^j \quad \text{et} \quad B(z) = \sum_{k=0}^{\infty} b_k z^k$$

sont données par

$$A(z) = \frac{\alpha(z)}{\beta(z)} \quad B(z) = \frac{\beta(z)}{\alpha(z)}$$

des fonctions holomorphes dans une boule ouverte contenant $\{|z| \leq 1\}$ donc telles que :

$$\sum_{i=0}^{\infty} |a_i| < \infty \quad \text{et} \quad \sum_{ki=0}^{\infty} |b_k| < \infty$$

de sorte que les séries (4) et (5) sont normalement convergentes dans $L^2(\Omega \mathcal{A}, P)$.

Il est clair que $\mathcal{L}_n(X) = \mathcal{L}_n(W)$ par (4) et (5). Cette propriété rend particulièrement simple les problèmes de prédiction et de filtrage pour les processus ARMA.

PRÉDICTION :

On veut calculer la projection \hat{X}_{n+h} de X_{n+h} sur $\mathcal{L}(X)$.

Comme $\hat{X}_{n+h} \in \mathcal{L}_n(W)$ et que $(W_m, m \leq n)$ est une base orthonormée de $\mathcal{L}_n(W)$, il existe un unique $(d_k) \in l^2(N)$ tel que :

$$\hat{X}_{n+h} = \sum_{k=0}^{\infty} d_k W_{n-k} \quad (6)$$

d'où, en exprimant W_{n-k} par (2),

$$\hat{X}_{n+h} = \sum_{l=0}^{\infty} c_l X_{n-l} \quad (7)$$

avec

$$c_l = \sum_{k+j=l} \sum_{k \geq 0, j \geq 0} d_k a_j$$

On a $c \in l^2(N)$. En exprimant X_{n-l} à son tour par (5) et par l'unicité de (6) :

$$d_m = \sum_{l+k=m} \sum_{l \geq 0, k \geq 0} c_l b_k$$

Autrement dit, si on pose

$$B(z) = \sum_0^{\infty} b_n z^n, \quad C(z) = \sum_0^{\infty} c_n z^n \quad \text{et} \quad D(z) = \sum_0^{\infty} d_n z^n$$

on a :

$$D(z) = B(z)C(z)$$

Il faut maintenant exprimer \hat{X}_{n+h} comme la projection de X_{n+h}

$$E(X_{n+h} - \hat{X}_{n+h})^2 = E\left(\sum_{k=0}^{\infty} b_k W_{n+h-k} - \sum_{q=0}^{\infty} d_q W_{n-q}\right)^2 = \sum_{k=0}^{k-1} b_k^2 + \sum_{q=0}^{\infty} (b_{k+q} - d_q)^2 \quad (8)$$

expression qui est minimale pour

$$d_q = b_{h+q}$$

donc lorsque

$$D(z) = \sum_{q=0}^{\infty} b_{h+q} z^q.$$

Si pour une série

$$S(z) = \sum_{-\infty}^{+\infty} S_n z^n$$

on pose

$$[S(z)]_+ = \sum_0^{\infty} S_n z^n$$

on a donc d'après (8)

$$D(z) = [B(z)z^{-h}]_+ = B(z)C(z)$$

d'où

$$C(z) = \frac{[B(z)z^{-h}]_+}{B(z)}$$

et d'après (8), l'erreur de prédiction est

$$\sum_{k=0}^{h-1} |b_x|^2$$

Filtrage

Soit X toujours un processus ARMA vérifiant (4) et (5) et soit Y un processus stationnaire, stationnairement corrélé avec X . On pose :

$$g_{XY}(j) = E[Y_n X_{n-j}] \quad G_{XY}(z) = \sum_{-\infty}^{+\infty} g_{XY}(j) z^j \quad (9)$$

$$srw(k) = E[Y_n W_{n-k}] \quad G_{YW}(z) = \sum_{-\infty}^{+\infty} g_{YW}(k) z^k. \quad (10)$$

Nous cherchons désormais la projection \hat{Y}_n de Y_n sur $\mathcal{L}_n(X) = \mathcal{L}_n(W)$.

On a :

$$\hat{Y}_n = \sum_0^\infty d_q W_{n-q}$$

$$E(Y_n - \hat{Y}_n)^2 = EY_n^2 - 2 \sum_{q=0}^{\infty} d_q g_{YW}(q) + \sum_{q=0}^{\infty} d_q^2 = EY_n^2 - \sum_0^\infty g_{YW}(q)^2 + \sum_0^\infty (d_q - g_{YW}(q))^2$$

Une expression qui est minimale lorsque $d_q = g_{YW}(q)$ de sorte que

$$D(z) = [G_{YW}(z)]_+.$$

Or, en remplaçant dans (9) et (10) X_{n-j} par son expression en fonction de W on déduit que

$$G_{YX}(z) = B(z^{-1}) G_{YW}(z).$$

Enfin, on voit comme précédemment que

$$\hat{Y}_n = \sum_{m=0}^{\infty} c_m X_{n-m}$$

et que

$$D(z) = C(z)B(z).$$

On en déduit finalement que la meilleure estimation de Y_n est

$$\hat{Y}_n = \sum_{m=0}^{\infty} c_m X_{n-m} \quad \text{où } C(z) = \sum_{m=0}^{\infty} c_m z^m$$

est donnée par

$$C(z) = \frac{\left[\frac{G_{YX}(z)}{B(z^{-1})} \right]_+}{B(z)}.$$

3.3 Cas Non-Stationnaires

3.3.1 ARIMA

En fait, les grandeurs économiques sont rarement des processus stationnaires et nous sommes amené à les représenter plutôt comme des processus dont les accroissements sont stationnaires ou plus généralement comme des processus tels que si

$$(\Delta X)_n = X_n - X_{n+1}$$

le processus $(\Delta^r X)_n$ des différences itérées d'ordre r soit un processus **ARMA** (p,q) . De tels processus s'appellent **ARIMA** (p,q,r) (le I vient de integrated). Les méthodes d'estimations et de prédictions sont fondées sur des idées voisines de celles exposées ci-dessus.

Ces méthodes font l'objet de perfectionnements divers compte tenu des problèmes posés notamment par la correction des variations saisonnières ou désaisonnalisation pour les prévisions mensuelles.

Un problème intéressant est que ces méthodes de prévisions économiques fondées sur un traitement purement probabiliste et statistique des observations, donnent, dans certains cas, de meilleurs résultats que les prévisions fondées sur des modèles économétriques explicatifs des phénomènes. Cela est notamment causé par des imperfections dans la spécification du modèle lui-même.

4 Filtre de Kalman

Dans la section précédente nous avons traité les modèles ARMA. Désormais, nous allons nous intéresser au filtre stochastique de Kalman. Ce filtre estime les états d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées. Il est particulièrement efficace dans le cas linéaire, avec un bruit gaussien. Cette estimation a de nombreuses applications dans la vie quotidienne. A titre d'exemple, nous pouvons citer la correction de signaux bruités, le système de guidage d'un robot ou même estimer les processus de diffusion en finance.

4.1 Chaînes de Markov Cachées et Filtre de Kalman

Comme vu précédemment, le filtre de Kalman estime les états inconnus à partir d'une information incomplète, bruitée ou même manquante.

Notons $\{x_k\}$, l'état que nous cherchons à estimer et $\{y_k\}$ les observations à notre disposition.

Les $\{x_{k+1}\}$ ne dépendent que des $\{x_k\}$ donc il s'agit d'une chaîne de Markov, ainsi notre problème revient à déterminer les états cachés d'une chaîne de Markov (Modèle de Markov Caché). Les $\{y_k\}$ sont des fonctions du bruit.

Nous allons donc chercher à estimer les états cachés à l'aide d'un estimateur optimal. Pour cela, nous allons utiliser un filtre tel que le filtre de Kalman afin d'éliminer le bruit de $\{y_k\}$ et ensuite nous lisserons, dans la mesure du possible, le résultat pour augmenter les performances de la prédiction. Nous allons supposer l'équation de Markov connue (dans le cas contraire nous pouvons les déterminer par Tuning).

4.2 Algorithme de filtration

Montrons que l'on peut obtenir un système linéaire à partir de $\{x_k\}$ et $\{y_k\}$.

Soient $\{x_k\}$, $\{y_k\}$ deux processus définis sur (Ω, \mathcal{F}, P) .

Supposons X_0 connu tel que $X \in R^N$ et F_k la filtration adaptée.

La propriété de Markov nous donne :

$$P(x_{k+1} = e_j | F_k) = P(x_{k+1} | x_k) = P(x_{k+1} = e_j | x_k = e_i) = a_{ji} \quad (11)$$

On peut donc poser $A = (a_{j,i}) \in R^{N \times N}$ et donc

$$E[X_{k+1} | F_k] = E[X_{k+1} | X_k] = AX_k$$

Posons $V_{k+1} = X_{k+1} - AX_k$, ce qui nous donne une autre équation d'état.

De plus les v_{k+1} sont des martingales ainsi $E[v_{k+1} | F_k] = v_k$.

Comme nous observons les Y_k et non les X_k , nous allons exprimer les Y_{k+1} en fonction des X_k , de sorte que $Y_{k+1} = C(X_k, W_{k+1})$ où C est une fonction, $(w_k)_{k \geq 0}$ une suite de variables i.i.d avec, V_k indépendant de w_k et C indépendant du temps (k).

Ainsi, nous avons :

$$P(Y_{k+1} = f_j | x_0, x_1, \dots, y_1, \dots, y_k) = P(Y_{k+1} = f_j | x_k) = C_{ji} = P(Y_{k+1} = f_j | x_k = e_i)$$

De façon équivalente nous avons :

$$E[W_{k+1} \mid \mathcal{G}_k] = E[Y_{k+1} - C_k X_k \mid X_k] = CX_k - CX_k = 0$$

$$\Rightarrow Y_{k+1} = CX_k + WX_{k+1}$$

$$E[x_k \mid z_{k-1}] \Rightarrow \hat{x}_{k|k-1} \text{ si } k \geq 1 \text{ et } \hat{x}_0 = E[x_0]$$

$$x_{k+1} = F_k x_k + G_k w_k \quad (12)$$

$$z_k = H'_k x_k + v_k \quad (13)$$

Nous avons réussi à nous ramener à un problème linéaire à bruit gaussien.

Le filtre de Kalman combine à la fois un algorithme d'estimation ainsi qu'un algorithme de correction. Dans la suite nous allons étudier la manière dont s'opère ce type d'estimations à l'aide des x_k et des y_k .

4.2.1 Prédictions

La distribution de x_k sachant les mesures jusqu'à l'instant k est normale :

$$x_k \mid y_{1:k} \sim \mathcal{N}(\bar{x}_{k|k}, P_{k|k})$$

Comme $x_{k+1} = Fx_k + u_k$, on obtient directement que $x_{k+1} \mid y_{1:k}$ suit une loi normale comme somme de deux variables normales :

$$x_{k+1} \mid y_{1:k} \approx \mathcal{N}(\bar{x}_{k+1|k}, P_{k+1|k})$$

avec

$$\bar{x}_{k+1|k} = E(Fx_k + u_k \mid y_{1:k}) = F\bar{x}_{k|k}$$

et

$$P_{k+1|k} = E \left[(x_{k+1} - \bar{x}_{k+1|k}) (x_{k+1|k} - \bar{x}_{k+1|k})^T \mid y_{1:k} \right] = FP_{k|k}F^T + Q$$

4.2.2 Corrections

Grâce à l'étape précédente de prédiction nous disposons d'une estimation de l'état :

$$x_{k+1} \mid y_{1:k} \sim \mathcal{N}(\bar{x}_{k+1|k}, P_{k+1|k})$$

et des mesures

$$\begin{aligned} y_{k+1} &= Hx_{k+1} + v_k \\ y_{k+1} \mid x_{k+1} &\sim \mathcal{N}(Hx_{k+1}, R) \end{aligned}$$

On veux trouver la distribution de

$$x_{k+1} \mid y_{1:k+1}$$

c'est à dire de l'état connaissant les mesures jusqu'à l'instant $k+1$.

On va donc utiliser les probabilités conditionnelles et notamment appliquer le théorème de Bayes :

$$\begin{aligned} p(a \mid b) &= \frac{p(b|a)p(a)}{p(b)} \\ p(x_{k+1} \mid y_{1:k}, y_{k+1}) &= p(y_{k+1} \mid x_{k+1}, y_{1:k}) p(x_{k+1} \mid y_{1:k}) \end{aligned}$$

On montre après quelques lignes de calcul que

$$x_{k+1} \mid y_{1:k+1} \sim \mathcal{N}(\tilde{x}_{k+1|k+1}, P_{k+1|k+1})$$

avec

$$K_{k+1} = P_{k+1|k} H (H P_{k+1,k} H + R)^{-1}$$

$$\bar{x}_{k+1|k+1} = \bar{x}_{k+1|k} + K_{k+1} (y_{k+1} - H\bar{x}_{k+1|k})$$

$$P_{k+1|k+1} = (I - K_{k+1} H) P_{k+1|k}$$

L'ALGORITHME DE KALMAN

Données

Modèles F et H , Covariances Q et R , Conditions Initiales $\hat{x}_{0,0}$ et $P_{0,0}$.

Prédictions

$$\hat{x}_{k+1,k} = F\hat{x}_{k,k}$$

$$P_{k+1,k} = F P_{k,k} F^* + Q$$

Corrections

$$K_{k+1} = P_{k+1|k} H (H P_{k+1|k} H + R)^{-1}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} (y_{k+1} - H\hat{x}_{k+1|k})$$

$$P_{k+1|k+1} = (I - K_{k+1} H) P_{k+1|k}$$

4.3 Lien entre filtre de Kalman et Modèle ARMA

Dans cette section nous allons montrer le lien entre le filtre de Kalman et les modèles ARMA. Le filtrage de Kalman peut servir à identifier les coefficients d'une équation ARMA de la forme suivante

$$y_k + a^{(1)}y_{k-1} + \cdots + a^{(n)}y_{k-n} = a^{(n+1)}u_{k-1} + \cdots + a^{(n+m)}u_{k-m} \quad (14)$$

Si (14) décrit le système et si les a_k sont des constantes, avec suffisamment de données le problème peut être résolu par un système linéaire. Cependant, dans la réalité les a_k suivent des variables aléatoires.

Supposons que pour tout i

$$a_{k+1}^i = a_{k+}^i + w_{k+}^i \quad (15)$$

où les $(w_k)_k$ suivent des lois normales centrées réduites i.i.d.

Nous allons remplacer (14) par :

$$y_k + a_k^{(1)}y_{k-1} + \cdots + a_k^{(n)}y_{k-n} = a_k^{(n+1)}u_{k-1} + \cdots + a_k^{(n+m)}u_{k-m} + v_k \quad (16)$$

où les $(v_k)_k$ suivent des lois normales centrées réduites indépendantes du processus w_k .

Concrètement, les v_k sont des bruits issus des appareils de mesures et les w_i sont les variations des a^i , en supposant que les a_i suivent des lois gaussiennes.

Nous pouvons maintenant poser le problème d'identification en terme de filtre de Kalman.

On définit un vecteur composé des états x_i de dimension $n+m$

$$x_k^{(1)} = a_k^{(1)}, \quad x_k^{(2)} = a_k^{(2)}, \quad \dots, \quad x_k^{(n+m)} = a_k^{(n+m)} \quad (17)$$

On définit également le processus de dimension $n+m$, w_k de bruit blanc gaussien, issu du vecteur w_k^i .

Ensuite les équations (16) et (17) nous donnent :

$$x_{k+1} = x_k + w_k \quad (18)$$

Ensuite, on définit la matrice :

$$H'_k = [-y_{k-1}, -y_{k-2}, \dots, u_{k-1}, u_{k-2}, \dots, u_{k-n}]$$

et le processus z_k par $z_k = y_k$ ce qui nous donne $z_k = H'_k x_k + v_k$ avec H_k connue. Ceci est suffisant pour définir le filtre de Kalman. Dans ce cas, le filtre devient

$$\hat{x}_{k+1|k} = [I - K_k H'_k] \hat{x}_{k|k-1} + K_k z_k$$

avec

$$K_k = \Sigma_{k|k-1} H_k [H'_k \Sigma_{k|k-1} H_k + R_k]^{-1}$$

et

$$\Sigma_{k+1|k} = \Sigma_{k|k-1} - \Sigma_{k|k-1} H_k [H'_k \Sigma_{k|k-1} H_k + R_k]^{-1} H'_k \Sigma_{k|k-1} + Q_k$$

$$R_k = E[v_k^2] \text{ et } Q_k = E[w_k w'_k].$$

On obtient ainsi un filtre de Kalman à partir d'un modèle ARMA.

5 Application du filtre de Kalman

Dans cette section nous allons illustrer une application du filtre de Kalman à un signal sinusoïdal bruité selon une variable aléatoire qui suit une loi normale centrée et réduite.

5.1 Codage du filtre de Kalman

Dans la *Figure.4* nous allons coder les différentes fonctions essentielles à l'algorithme de filtration. Cette première partie de code se décompose en trois fonctions :

5.1.1 Fonction d'initialisation

- \mathbf{F}_k , États de transitions ;
- \mathbf{H}_k , Données observées ;
- \mathbf{Q}_k , Matrice de covariance du bruit issue du processus ;
- \mathbf{R}_k , Matrice de covariance du bruit issue des moyens d'observations ;
- \mathbf{B}_k , Matrice de contrôle et \mathbf{u}_k , le vecteur de contrôle.

On passe de l'état $k - 1$ à l'état k de la manière suivante :

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k$$

où

- \mathbf{F}_k l'état de transition du modèle appliquée à l'état précédent \mathbf{x}_{k-1} ;
- \mathbf{B}_k la matrice de contrôle du modèle appliquée au vecteur de contrôle \mathbf{u}_k ;
- \mathbf{w}_k bruit qui suit une loi normale centrée, \mathcal{N} , avec la matrice de covariance \mathbf{Q}_k : $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$.

Au temps k une observation \mathbf{z}_k est faite de l'état \mathbf{x}_k selon :

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k$$

où

- \mathbf{v}_k bruit qui suit une loi normale centrée, \mathcal{N} , avec la matrice de covariance \mathbf{R}_k : $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$.

5.1.2 Fonction de prédiction

Estimateur d'état :

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$$

Estimateur de la matrice de covariance :

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}_k$$

5.1.3 Fonction de correction

Préajustement de mesure :

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$$

Préajustement de covariance :

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k$$

Gain de Kalman:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \mathbf{S}_k^{-1}$$

Mise à jour de l'estimateur d'état :

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$$

Mise à jour de l'estimateur de la matrice de covariance :

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$$

Mesures post-ajustements :

$$\tilde{\mathbf{y}}_{k|k} = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k}$$

```
import numpy as np
|
class KalmanFilter(object):
    def __init__(self, F = None, B = None, H = None, Q = None, R = None, P = None, x0 = None):
        if(F is None or H is None):
            raise ValueError("Set proper system dynamics.")

        self.n = F.shape[1]
        self.m = H.shape[1]

        self.F = F
        self.H = H
        self.B = 0 if B is None else B
        self.Q = np.eye(self.n) if Q is None else Q
        self.R = np.eye(self.n) if R is None else R
        self.P = np.eye(self.n) if P is None else P
        self.x = np.zeros((self.n, 1)) if x0 is None else x0

    def predict(self, u = 0):
        self.x = np.dot(self.F, self.x) + np.dot(self.B, u)
        self.P = np.dot(np.dot(self.F, self.P), self.F.T) + self.Q
        return self.x

    def update(self, z):
        y = z - np.dot(self.H, self.x)
        S = self.R + np.dot(self.H, np.dot(self.P, self.H.T))
        K = np.dot(np.dot(self.P, self.H.T), np.linalg.inv(S))
        self.x = self.x + np.dot(K, y)
        I = np.eye(self.n)
        self.P = np.dot(np.dot(I - np.dot(K, self.H), self.P),
                       (I - np.dot(K, self.H)).T) + np.dot(np.dot(K, self.R), K.T)
```

FIG. 4 – Part 1 : Kalman Coding

5.2 Exemple d'application du filtre de Kalman

Dans la *Figure.5* nous allons donc reprendre le code créé précédemment (*Figure.4*) afin d'illustrer la pertinence du filtre de Kalman.

Nous allons prendre :

$$dt = \frac{1}{60} ; F = \begin{pmatrix} 1 & dt & 0 \\ 0 & 1 & dt \\ 0 & 0 & 1 \end{pmatrix} ; Q = \begin{pmatrix} 0,05 & 0,05 & 0,0 \\ 0,05 & 0,05 & 0,0 \\ 0,0 & 0,0 & 0,0 \end{pmatrix} ; H = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$x = [-5,5]$ partitionné en 1000

measurement = $-\sin(x) + N$ avec $N \sim N(0,1)$

etat-cache = $-\sin(x)$

```
def example():
    dt = 1.0/60
    F = np.array([[1, dt, 0], [0, 1, dt], [0, 0, 1]])
    H = np.array([1, 0, 0]).reshape(1, 3)
    Q = np.array([[0.05, 0.05, 0.0], [0.05, 0.05, 0.0], [0.0, 0.0, 0.0]])
    R = np.array([0.5]).reshape(1, 1)

    x = np.linspace(-5, 5, 1000)
    measurements = - np.sin(x) + np.random.normal(0, 1, 1000)
    etat_cache = - np.sin(x)

    kf = KalmanFilter(F = F, H = H, Q = Q, R = R)
    predictions = []

    for z in measurements:
        predictions.append(np.dot(H, kf.predict())[0])
        kf.update(z)

    import matplotlib.pyplot as plt
    plt.plot(range(len(measurements)), measurements, label = 'Measurements', linestyle='dashed')
    plt.plot(range(len(predictions)), np.array(predictions), label = 'Kalman Filter Prediction')
    plt.plot(range(len(etat_cache)), etat_cache, label = 'etat_cache')

    plt.legend()
    plt.show()

if __name__ == '__main__':
    example()
```

FIG. 5 – Part 2 : Kalman Example

Après avoir créé la fonction *filtre de Kalman* et après l'avoir appliquée aux différentes valeurs prises plus haut, nous obtenons le résultat suivant :

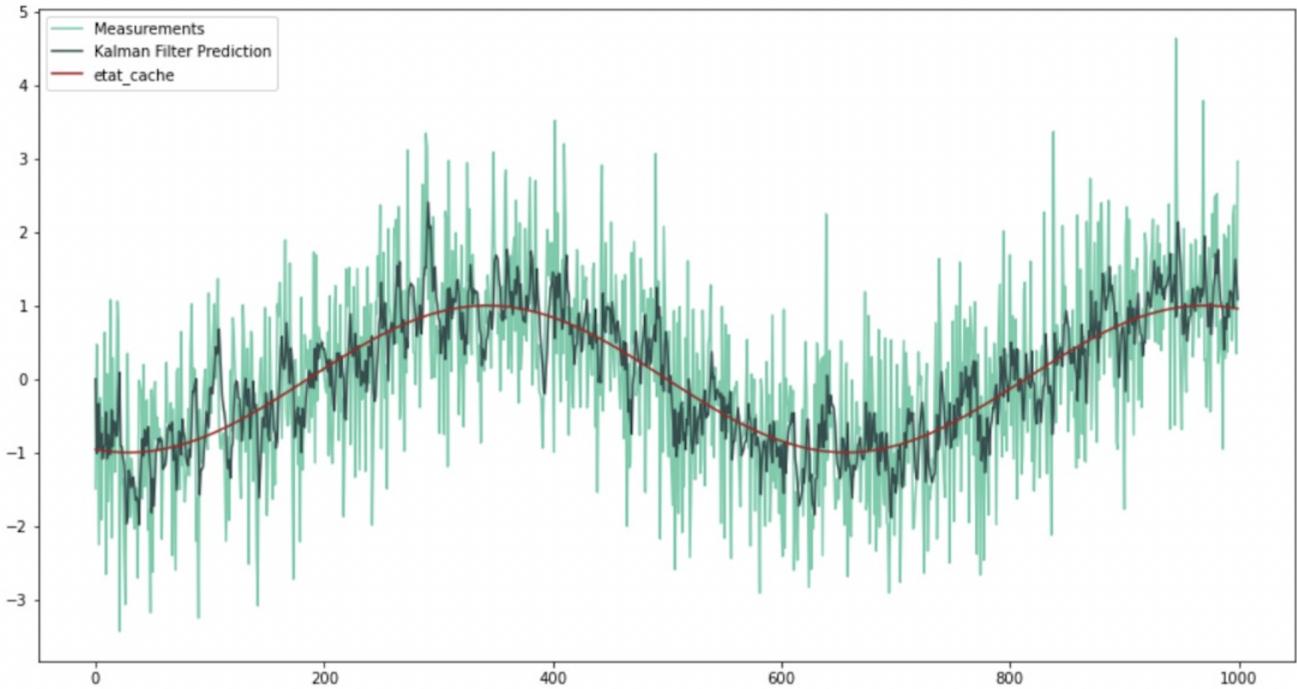


FIG. 6 – Illustration Kalman Filter

Nous observons ainsi un signal vert correspondant aux mesures effectuées avec le bruit gaussien. La grande amplitude de ce signal est dûe à la variance du bruit. Rappelons que, dans notre exemple, le bruit est une variable aléatoire centrée et réduite.

Ensuite, nous remarquons un signal tracé en noir qui suit parfaitement le premier signal vert. Cependant, nous constatons que son amplitude est bien plus faible. Ce signal correspond donc au résultat de la filtration du signal vert par l'algorithme de Kalman.

Enfin, en rouge est tracé le signal non bruité, une courbe sinusoïdale classique.

Nous pouvons constater que le filtre de Kalman est très efficace dans notre exemple car il atténue grandement le bruit et approche ainsi le signal non bruité. Cependant, le résultat n'est pas lissé : il comporte de nombreux changements de taux de variation.

6 Conclusion

Un filtre est une reconstruction $\hat{X}(t)$ d'un signal $X(t)$ à partir du signal observé (bruité) ($Y(s), s \leq t$). Il s'agit donc un algorithme récursif d'estimation et d'autocorrection, c'est-à-dire un algorithme qui puisse s'écrire de manière formelle pour un accroissement infinitésimal du temps d'observation $\hat{X}(t + dt)$ en fonction de $\hat{X}(t)$ et de $dY(t), dt$. L'algorithme étant récursif, il ne va pas recalculer les $\hat{X}(t)$ à l'instant $t + dt$ à partir des $Y(s), s \leq t$ mais va reprendre $\hat{X}(t)$ pour prédire $\hat{X}(t + 1)$.

Dans tout ce document, nous avons étudié un cas particulier et bien défini qui est celui des problèmes de types linéaires avec un bruit blanc gaussien. La réalité est rarement aussi simple. Ainsi, les limites de ces différents modèles et algorithmes sont généralement vite atteintes lorsque l'on veut traiter des problèmes physiques ou même financiers. De plus le filtre de Kalman n'a pas d'effets de lissage donc les performances de la prédition s'en retrouvent amoindries, impactant ainsi la filtration qui est donc un peu moins efficace.

Les modèles ARMA/ARIMA possèdent également ces faiblesses. Ainsi, pour exploiter leurs plein potentiel, il est nécessaire de connaître l'ensemble des paramètres, dans le cas où ils nous seraient inconnus il faudrait passer par des estimateurs généralement calculés grâce à l'algorithme EM.

7 Bibliographie

7.1 Ouvrages et Articles

- Elliott, Aggoun, Moore, 1995, *Hidden Markov Models*, Springer
- Anderson, Moore, 1979, *Optimal Filtering*, Prentice Hall
- Augeri, 2021, *Introduction aux chaînes de Markov*
- Campillo, 2006, *Modèle de Markov cachés et filtrage particulaire*
- Chafaï, Djalil, Lévy-Leduc, Céline, Roche, Angelina, 2017, *Introduction aux séries temporelles*
- Hastie, Tibshirani, Sherlock, Eisen, Brown, Botstein, 1999, *Imputing missing data for gene expression arrays*, Technical report, Division of Biostatistics, Stanford University
- Honaker, King, Blackwell, 2011, Amelia II. *A program for missing data*, Journal of Statistical Software
- Rubin, Wiley, 1987, *Multiple Imputation for Nonresponse in Surveys*
- Stekhoven, Buehlmann, 2012, *Non-parametric missing value imputation for mixed-type data*, Bioinformatics
- Van Buuren, 2012, *Flexible Imputation of Missing Data*
- Shen, Zetik, Thoma, 2008, *Performance Comparison of ToA and TDoA Based Location Estimation Algorithms in LOS Environment*
- Welch, Bishop, 1995, *An Introduction to the Kalman Filter*, University of North Carolina, Department of Computer Science
- Kalman, Kalman, 1960, *A New Approach to Linear Filtering and Prediction Problems*, Transaction of the SME-Journal of Basic Engineering
- Comets, Meyer, 2015, *Calcul stochastique et modèles de diffusions*
- Bouleau, 2004, *Processus stochastiques et applications*

7.2 Liens

- <https://github.com/zziz/kalman-filter>