

Moindres carrés récurrents et algorithmes adaptatifs



Département Traitement du Signal et des Images
Laboratoire de Traitement et de Communication de l'Information-UMR5141 Télécom
ParisTech

5 février 2013



- 1 Introduction
- 2 LMS
- 3 Moindres Carrés Récursifs (MCR) - RLS

Introduction

Dans la suite on envisage des systèmes entrée-sortie qui varient dans le temps.

S'adapter : changer sa structure, ses modes de travail, ses paramètres pour répondre à des changements d'environnement.

Exemple : Annulation d'écho acoustique

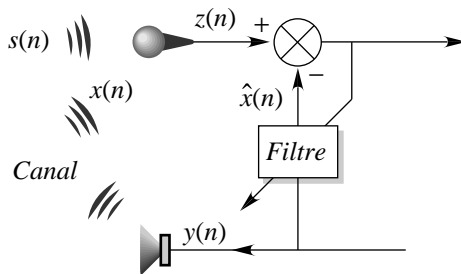


FIGURE: *Annulation d'écho acoustique*

En pratique, les coefficients du “filtre” doivent être modifiés de façon à suivre les variations du canal acoustique.

En l'absence du signal $s(n)$, le signal en sortie de l'annuleur d'écho doit être nul.

Estimation de la moyenne : convergence

On veut estimer la moyenne $m = \mathbb{E}[x_1]$ d'une suite de v.a. x_n supposées i.i.d. On utilise l'estimateur

$$\hat{m}_n = \frac{1}{n} \sum_{i=1}^n x_i$$

Acquisition/convergence : la loi des grands nombres assure ...

Mis sous forme récursive ... il vient

$$\hat{m}_n = \hat{m}_{n-1} + \mu_n(x_n - \hat{m}_{n-1})$$

où $m_0 = 0$ et où $\mu_n = \frac{1}{n}$; μ_n est appelé le pas d'adaptation
Le pas tend vers 0 quand n tend vers l'infini.'

Estimation de la moyenne : poursuite

On suppose à présent que

$$\mathbb{E}[x_n] = \begin{cases} m_1 & \text{si } n \leq T \\ m_2 & \text{si } n > T \end{cases}$$

Montrons que pour $n > T$, on a

$$\hat{m}_n - m_2 = \frac{T}{n}(m_1 - m_2) + \delta_n$$

où

$$\delta_n = \frac{1}{n} \sum_{i=1}^T (x_i - m_1) + \frac{1}{n} \sum_{i=T+1}^n (x_i - m_2)$$

Estimation de la moyenne : poursuite

Pour $n > T$ on a :

$$\hat{m}_n - m_2 = \frac{1}{n} \sum_{i=1}^T (x_i - m_2) + \sum_{i=T+1}^n (x_i - m_2)$$

On peut écrire :

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^T (x_i - m_2) &= \frac{1}{n} \sum_{i=1}^T (x_i - m_1 + m_1 - m_2) \\ &= \frac{1}{n} \sum_{i=1}^T (x_i - m_1) + \frac{1}{n} \sum_{i=1}^T (m_1 - m_2) \end{aligned}$$

Estimation de la moyenne : poursuite

Il vient :

$$\hat{m}_n - m_2 = \frac{T}{n}(m_1 - m_2) + \delta_n$$

avec

$$\delta_n = \frac{1}{n} \sum_{i=1}^T (x_i - m_1) + \frac{1}{n} \sum_{i=T+1}^n (x_i - m_2)$$

D'après les hypothèses, δ_n est centrée. Par conséquent pour $n > T$

$$\epsilon^2 = \mathbb{E} [(\hat{m}_n - m_2)^2] \geq \frac{T^2(m_1 - m_2)^2}{n^2}$$

Pour atteindre une précision de ϵ^2 il faut donc prendre $n > \frac{T|m_1 - m_2|}{\epsilon}$

Capacité de poursuite ...

Estimation de la moyenne : algorithme à pas constant

Pour éviter la perte de la capacité de poursuite l'idée est de construire un algorithme à pas constant :

$$\hat{m}_n = \hat{m}_{n-1} + \mu (x_n - \hat{m}_{n-1})$$

Étudions d'abord la capacité de poursuite. On remarque que (par récurrence) :

$$\hat{m}_n = \mu (x_n + \cdots + (1 - \mu)^j x_{n-j} + \cdots + (1 - \mu)^{n-1} x_1)$$

Ainsi, si $0 < \mu < 2$, quand n croît, les termes du passé sont “oubliés” de façon exponentielle avec le taux $(1 - \mu)$

→ Capacité de poursuite

Algorithme à pas constant - Acquisition

Étudions maintenant la capacité d'acquisition

On suppose que $x_n = m + Z_n$ avec $Z_n \sim BB(0, \sigma^2)$.

On pose $\epsilon_n = \hat{m}_n - m$. Alors

$$\epsilon_n = (1 - \mu) \epsilon_{n-1} + \mu Z_n$$

Cette équation est celle d'un processus AR-1. On sait qu'elle a une solution stationnaire ssi $|1 - \mu| < 1$ cad $0 < \mu < 2$

Algorithme à pas constant - Acquisition

On montre que la solution stationnaire a pour moyenne :

$$\mathbb{E}[\epsilon_n] = 0$$

et, par conséquent, \hat{m}_n fluctue autour de la vraie valeur m et que sa variance s'écrit :

$$\text{var}(\epsilon_n) = \frac{\mu^2}{1 - (1 - \mu)^2} \sigma^2 = \frac{\mu}{2 - \mu} \sigma^2$$

Si $\mu \ll 2$ alors ...

On doit accepter un compromis entre convergence et capacité de poursuite.

Algorithme récursif général

Pour prendre en compte d'éventuelles non-stationnarités, on considère des algorithmes de la forme suivante :

$$\hat{\theta}_n = \hat{\theta}_{n-1} + \Delta_n(e(n))$$

où Δ_n tend vers 0 quand $e(n)$ tend vers 0. On note que cette forme d'algorithmes ne nécessite ni une opération de minimisation ni une opération de calcul de moyenne. On souhaite vérifier les points suivants :

- la simplicité d'implémentation
- la capacité d'acquisition,
- la capacité de poursuite.

Théorème de projection

- On considère un espace de Hilbert \mathcal{H} .
- Soit $x \in \mathcal{H}$ et $\mathcal{C} \subset \mathcal{H}$.
- Il existe un unique élément de \mathcal{C} , noté $(x|\mathcal{C})$, tel que

$$\forall y \in \mathcal{C} \quad \|x - (x|\mathcal{C})\| \leq \|x - y\|$$

- $(x|\mathcal{C})$ vérifie

$$\begin{aligned} x - (x|\mathcal{C}) &\perp \mathcal{C}, \quad \text{principe d'orthogonalité} \\ \|x - (x|\mathcal{C})\|^2 &= \|x\|^2 - \|(x|\mathcal{C})\|^2 \end{aligned}$$

Filtre de Wiener

Dans la suite, l'espace de Hilbert considéré est celui des v.a. de carré intégrable $\mathbb{E} [|x_n|^2] < +\infty$.

On considère une suite d'observations (x_n, y_n) stationnaires au second ordre, centrées et de covariance stationnaire et on cherche $\underline{\theta} = [\theta_0, \dots, \theta_{P-1}]^T$ qui minimise

$$J(\theta) = \mathbb{E} [|x_n - \underline{\theta}^T \underline{y}_n|^2]$$

où $\underline{y}_n = [y_n, \dots, y_{n-P+1}]^T$.

Par conséquent $\underline{\theta}$ vérifie :

$$x_n - \underline{\theta}^T \underline{y}_n \perp \underline{y}_n \quad \text{cad} \quad \mathbb{E} [\underline{y}_n (x_n - \underline{\theta}^T \underline{y}_n)^T] = 0$$

Filtre de Wiener

Montrons que la solution s'écrit :

$$\underline{\theta}_w = \underline{R}_{yy}^{-1} \mathbb{E} \left[(\underline{y}_n \ x_n) \right]$$

avec $\underline{R}_{yy} = \mathbb{E} \left[\underline{y}_n \underline{y}_n^T \right]$

→ $\underline{\theta}_w$ est le filtre de Wiener

Montrons que :

$$J(\underline{\theta}_w) = \mathbb{E} \left[|x_n|^2 \right] - \mathbb{E} \left[x_n \underline{y}_n^T \right] \underline{R}_{yy}^{-1} \mathbb{E} \left[(\underline{y}_n \ x_n) \right]$$

On en déduit :

$$J(\underline{\theta}) = J(\underline{\theta}_w) + (\underline{\theta} - \underline{\theta}_w)^T \underline{R}_{yy} (\underline{\theta} - \underline{\theta}_w)$$

Filtre de Wiener

Reprenons :

$$\underline{\theta}_w = \underline{\underline{R}}_{yy}^{-1} \mathbb{E} \left[(\underline{y}_n \ x_n) \right] \quad (1)$$

Les problèmes liés au calcul de θ_w (expression (1)) sont :

- Estimation des covariances,
- Calcul de l'inverse de matrice,
- Absence d'adaptation

Pour répondre à ces différents points, une idée consiste à déduire, dans un premier temps, un algorithme récursif pour la solution stationnaire puis de remplacer certaines espérances par des estimations basées sur les observations passées.

Algorithme du gradient

Un algorithme récursif classique en optimisation est l'algorithme dit du gradient. Soit la fonction d'objectif à minimiser

$$J(\underline{\theta}) = \mathbb{E} [|x_n|^2] - 2\underline{\theta}^T \mathbb{E} [x_n \underline{y}_n] + \underline{\theta}^T \mathbb{E} [\underline{y}_n \underline{y}_n^T] \underline{\theta}$$

Son gradient s'écrit :

$$\nabla_{\underline{\theta}}(J) = -2\mathbb{E} [x_n \underline{y}_n] + 2\mathbb{E} [\underline{y}_n \underline{y}_n^T] \underline{\theta}$$

Algorithme du gradient

L'algorithme du gradient a pour expression :

$$\begin{aligned}
 \underline{\theta}_n &= \underline{\theta}_{n-1} - \mu \nabla_{\underline{\theta}}(J), \quad \text{où } \mu \geq 0 \\
 &= \underline{\theta}_{n-1} + 2\mu \mathbb{E} \left[\underline{y}_n (x_n - \underline{y}_n^T \underline{\theta}_{n-1}) \right] \\
 &= (I - 2\mu \mathbb{E} \left[\underline{y}_n \underline{y}_n^T \right]) \underline{\theta}_{n-1} + 2\mu \mathbb{E} \left[\underline{y}_n x_n \right] \\
 &= \left(I - 2\mu \underline{\underline{R}}_{yy} \right) \underline{\theta}_{n-1} + 2\mu \mathbb{E} \left[\underline{y}_n x_n \right]
 \end{aligned} \tag{2}$$

On montre que si $0 < \mu < 2/\lambda_{\max}$ – avec λ_{\max} : plus grande valeur propre de $\underline{\underline{R}}_{yy}$ – alors :

- la suite $\underline{\theta}_n$ converge
- et sa limite est $\underline{\theta}_w = \underline{\underline{R}}_{yy}^{-1} \mathbb{E} \left[(\underline{y}_n x_n) \right]$

Algorithme du gradient

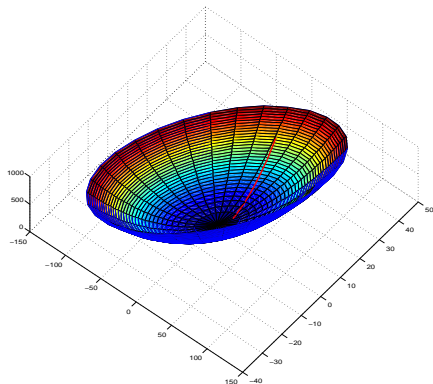


FIGURE: *Critère du gradient déterministe*

LMS - Least Mean Squares

L'algorithme LMS, appelé aussi algorithme du gradient stochastique, consiste à supprimer les espérances dans

$$\underline{\theta}_n = \underline{\theta}_{n-1} + 2\mu \mathbb{E} \left[\underline{y}_n (x_n - \underline{y}_n^T \underline{\theta}_{n-1}) \right]$$

ce qui donne :

Valeur initiale : $\underline{\theta}_0 = [0, \dots, 0]^T$,

Répéter :

$$\left| \begin{array}{l} e_n = x_n - \underline{y}_n^T \underline{\theta}_{n-1} \\ \underline{\theta}_n = \underline{\theta}_{n-1} + \mu e_n \underline{y}_n \end{array} \right. \quad (3)$$

Le problème de la convergence est un problème difficile dans le cas général.

LMS normalisé

On a vu que le pas du gradient doit être choisi comme l'inverse de la plus grande valeur propre. D'où l'idée de prendre, quand le signal n'est pas stationnaire, un pas variant comme l'inverse de la puissance "instantanée".

Valeur initiale : $\underline{\theta}_0 = [0, \dots, 0]^T$,

Répéter :

$$\begin{aligned} e_n &= x_n - \underline{y}_n^T \underline{\theta}_{n-1} \\ \underline{\theta}_n &= \underline{\theta}_{n-1} + \frac{\nu}{\delta + \underline{y}_n^T \underline{y}_n} e_n \underline{y}_n \end{aligned} \quad (4)$$

δ est une constante qui évite la division par 0

Moindres Carrés Récursifs avec facteur d'oubli exponentiel

On reprend l'idée des moindres carrés qui, dans le cas de la moyenne s'écrit

$$\min_m \sum_{i=1}^n (x_i - m)^2$$

qui donne comme estimateur $\hat{m}_n = n^{-1} \sum_{i=1}^n x_i$ dont l'expression récursive s'écrit

$$\hat{m}_n = \hat{m}_{n-1} + \frac{1}{n}(x_n - \hat{m}_{n-1}) \quad (5)$$

Moindres Carrés Récursifs avec facteur d'oubli exponentiel

Un inconvénient de l'expression (5) en terme de poursuite est le gain de la forme $1/n$ qui tend vers 0 quand n tend vers l'infini. Pour contourner cette difficulté on introduit un facteur d'oubli sur les échantillons passés :

$$\min_m \sum_{i=1}^n (x_i - m)^2 \lambda^{n-i} \quad \text{avec} \quad 0 < \lambda \leq 1$$

Montrons que pour $\lambda \neq 1$

$$\hat{m}_n = \hat{m}_{n-1} + \frac{1 - \lambda}{1 - \lambda^n} (x_n - \hat{m}_{n-1})$$

Moindres Carrés Récursifs avec facteur d'oubli exponentiel

On va concevoir un algorithme adaptatif permettant d'estimer les coefficients d'un filtre à l'instant n à partir de ses coefficients à l'instant $n - 1$, à la lumière des nouvelles observations, avec un facteur d'oubli

On observe une suite scalaire x_n et une suite vectorielle \underline{y}_n de dimension P . On cherche $\hat{\underline{\theta}}_n$ qui minimise

$$J(\underline{\theta}_n) = \sum_{i=1}^n \lambda^{n-i} (x_i - \underline{y}_i^T \underline{\theta}_n)^2$$

- $\underline{y}_i = [y_i, \dots, y_{i-P+1}]^T$
- $\underline{\theta}_n = [\theta_0^n, \dots, \theta_n^{P-1}]^T$: est le filtre à l'instant n
- λ est un facteur d'oubli ($0 << \lambda \leq 1$)

Résolution

Montrons que la solution qui minimise $J(\underline{\theta}_n) = \sum_{i=1}^n \lambda^{n-i} (x_i - \underline{y}_i^T \underline{\theta}_n)^2$ est telle que :

$$\underline{\underline{Y}}_n \underline{\underline{\Lambda}}_n \underline{x}_n = \underline{\underline{Y}}_n \underline{\underline{\Lambda}}_n \underline{\underline{Y}}_n^T \hat{\underline{\theta}}_n$$

où

- $\underline{\underline{Y}}_n = [\underline{y}_1, \dots, \underline{y}_n]$,
- $\underline{x}_n = [x_1, \dots, x_n]^T$,
- $\underline{\underline{\Lambda}}_n = \text{diag}[\lambda^{n-1}, \dots, 1]$

Résolution

La solution est telle que

$$\nabla_{\underline{\theta}_n}(J) = [0, \dots, 0]^T$$

soit :

$$\sum_{i=1}^n \lambda^{n-i} \left[x_i - \underline{y}_i^T \hat{\underline{\theta}}_n \right] \underline{y}_i = [0, \dots, 0]^T$$

ce qui donne :

$$\sum_{i=1}^n \lambda^{n-i} x_i \underline{y}_i = \sum_{i=1}^n \lambda^{n-i} \underline{y}_i^T \hat{\underline{\theta}}_n \underline{y}_i$$

ou encore :

$$\underline{\underline{Y}}_n \underline{\underline{\Lambda}}_n \underline{x}_n = \underline{\underline{Y}}_n \underline{\underline{\Lambda}}_n \underline{\underline{Y}}_n^T \hat{\underline{\theta}}_n$$

Résolution

Posons

$$\underline{\underline{R}}_n = \underline{\underline{Y}}_n \underline{\underline{\Lambda}}_n \underline{\underline{Y}}_n^T = \sum_{i=1}^n \lambda^{n-i} \underline{y}_i \underline{y}_i^T$$

$$\underline{p}_n = \underline{\underline{Y}}_n \underline{\underline{\Lambda}}_n \underline{x}_n = \sum_{i=1}^n \lambda^{n-i} x_i \underline{y}_i$$

On obtient

$$\hat{\underline{\theta}}_n = \underline{\underline{R}}_n^{-1} \underline{p}_n$$

On souhaite calculer la solution de façon récursive : on cherche à calculer $\hat{\underline{\theta}}_n$ en fonction de $\hat{\underline{\theta}}_{n-1}$

On commence par exprimer $\underline{\underline{R}}_n$ et \underline{p}_n en fonction de $\underline{\underline{R}}_{n-1}$ et \underline{p}_{n-1}

Résolution récursive

$$\underline{\underline{R}}_n = \lambda \underline{\underline{R}}_{n-1} + \underline{y}_n \underline{y}_n^T$$

$$\underline{p}_n = \lambda \underline{p}_{n-1} + x_n \underline{y}_n$$

On cherche une expression récursive de $\underline{\underline{R}}_n^{-1}$

Utilisons pour cela le lemme d'inversion matricielle :

$$(\underline{\underline{R}} + \underline{b} \underline{b}^T)^{-1} = \underline{\underline{R}}^{-1} - \frac{\underline{\underline{R}}^{-1} \underline{b} \underline{b}^T \underline{\underline{R}}^{-1}}{1 + \underline{b}^T \underline{\underline{R}}^{-1} \underline{b}}$$

Résolution récursive

$$\underline{\underline{R}}_n^{-1} = \lambda^{-1} \underline{\underline{R}}_{n-1}^{-1} - \frac{\lambda^{-2} \underline{\underline{R}}_{n-1}^{-1} \underline{y}_n \underline{y}_n^T \underline{\underline{R}}_{n-1}^{-1}}{1 + \lambda^{-1} \underline{y}_n^T \underline{\underline{R}}_{n-1}^{-1} \underline{y}_n}$$

On pose :

$$\underline{\underline{Q}}_n = \underline{\underline{R}}_n^{-1}$$

$$\underline{k}_n = \frac{\lambda^{-1} \underline{\underline{Q}}_{n-1} \underline{y}_n}{1 + \lambda^{-1} \underline{y}_n^T \underline{\underline{Q}}_{n-1} \underline{y}_n}$$

On obtient l'équation de récursion pour $\underline{\underline{Q}}_n$:

$$\underline{\underline{Q}}_n = \lambda^{-1} \underline{\underline{Q}}_{n-1} - \lambda^{-1} \underline{k}_n \underline{y}_n^T \underline{\underline{Q}}_{n-1}$$

On peut alors voir que $\underline{k}_n = \underline{\underline{Q}}_n \underline{y}_n = \underline{\underline{R}}_n^{-1} \underline{y}_n$

Résolution récursive

Il reste à trouver une équation récursive pour $\hat{\underline{\theta}}_n = \underline{Q}_n \underline{p}_n$

Montrons que

$$\hat{\underline{\theta}}_n = \hat{\underline{\theta}}_{n-1} + \underline{k}_n \left[x_n - \underline{y}_n^T \hat{\underline{\theta}}_{n-1} \right]$$

Résolution récursive

$$\begin{aligned}
 \hat{\underline{\theta}}_n &= \underline{Q}_{\underline{n}\underline{n}} p_n \\
 &= \lambda \underline{Q}_{\underline{n}\underline{n-1}} p_{n-1} + \underline{Q}_{\underline{n}\underline{n}} y_n x_n \\
 &= \lambda \left(\lambda^{-1} \underline{Q}_{\underline{n-1}\underline{n-1}} - \lambda^{-1} \underline{k}_n y_n^T \underline{Q}_{\underline{n}\underline{n-1}} \right) p_{n-1} + \underline{Q}_{\underline{n}\underline{n}} y_n x_n \\
 &= \underline{Q}_{\underline{n-1}\underline{n-1}} p_{n-1} - \underline{k}_n y_n^T \underline{Q}_{\underline{n}\underline{n-1}} p_{n-1} + \underline{Q}_{\underline{n}\underline{n}} y_n x_n \\
 &= \hat{\underline{\theta}}_{n-1} - \underline{k}_n y_n^T \hat{\underline{\theta}}_{n-1} + \underline{k}_n x_n \\
 &= \hat{\underline{\theta}}_{n-1} + \underline{k}_n \left[x_n - y_n^T \hat{\underline{\theta}}_{n-1} \right] \\
 &= \hat{\underline{\theta}}_{n-1} + \underline{k}_n e_n
 \end{aligned}$$

où $e_n = x_n - y_n^T \hat{\underline{\theta}}_{n-1}$ est l'erreur *a priori*

Algorithme MCR

Initialisation :

$$\begin{aligned}\underline{\theta}_0 &= [0, \dots, 0]^T \\ \underline{Q}_{\underline{=0}} &= \delta^{-1} \underline{I}_P, \text{ avec } 0 < \delta \ll 1\end{aligned}$$

Répéter :

$$\begin{aligned}\underline{k}_n &= \frac{\lambda^{-1} \underline{Q}_{\underline{=n-1}} \underline{y}_n}{1 + \lambda^{-1} \underline{y}_n^T \underline{Q}_{\underline{=n-1}} \underline{y}_n} \\ e_n &= x_n - \underline{y}_n^T \hat{\underline{\theta}}_{n-1} \\ \hat{\underline{\theta}}_n &= \hat{\underline{\theta}}_{n-1} + \underline{k}_n e_n \\ \underline{Q}_{\underline{=n}} &= \lambda^{-1} \underline{Q}_{\underline{=n-1}} - \lambda^{-1} \underline{k}_n \underline{y}_n^T \underline{Q}_{\underline{=n-1}}\end{aligned}$$