

## 3 Prédiction linéaire

### 3.1 Principe

Soit une séquence de données quelconque  $x[n]$ . On peut se poser le problème de décrire chacun des termes de cette séquence par une combinaison linéaire de ses  $p$  prédécesseurs (avec des poids  $a_1, \dots, a_p$ ), et ceci à une erreur de prédiction près, de puissance  $P$  : il s'agit alors de trouver le "meilleur" jeu de paramètres  $(p, a_1, \dots, a_p, P)$  correspondant à ces données, c'est-à-dire celui qui minimise  $P$ . Par construction de cette *prédiction linéaire*, chaque valeur  $x[n]$  est estimée par une quantité

$$\hat{x}[n] = - \sum_{k=1}^p a_k x[n-k],$$

avec une erreur de prédiction

$$\begin{aligned} e[n] &= x[n] - \hat{x}[n] \\ &= x[n] + \sum_{k=1}^p a_k x[n-k]. \end{aligned} \tag{7}$$

Puisque l'estimation  $\hat{x}[n]$  est engendrée linéairement à partir de  $p$  observations, l'estimée la meilleure (c'est-à-dire celle qui minimise  $P$ ) n'est autre que la projection orthogonale de  $x[n]$  sur l'espace engendré par les observations. Le jeu des coefficients optimaux est donc celui pour lequel (principe d'orthogonalité) l'erreur est orthogonale aux observations :

$$\mathbb{E}\{e[n] x[n-j]\} = 0; j \geq 1 \tag{8}$$

et la puissance de l'erreur minimum associée vaut

$$P = \mathbb{E}\{e[n] (x[n] - \hat{x}[n])\} = \mathbb{E}\{e[n] x[n]\}. \tag{9}$$

Reportant la forme explicite (7) dans (8), on en déduit que le jeu de coefficients  $\{a_1, \dots, a_p\}$  assurant la meilleure prédiction est solution du système linéaire (dit *de Yule-Walker*) donné par :

$$r[j] = - \sum_{k=1}^p a_k r[j-k]; j \geq 1 \tag{10}$$

(avec  $r[k] := \mathbb{E}\{x[n]x[n \pm k]\}$ ), l'erreur minimale associée prenant la forme

$$P = r[0] + \sum_{k=1}^p a_k r[k]. \quad (11)$$

D'un point de vue purement formel, l'équation (10) dit que le jeu des coefficients de prédiction optimaux est solution d'un système linéaire à  $p$  inconnues pour lequel il suffit de disposer de  $p$  équations reposant sur la connaissance des valeurs de la fonction de corrélation pour  $p$  retards strictement positifs. L'inversion directe d'un tel système pose néanmoins plusieurs problèmes qui en limitent l'application directe :

1. *Conditionnement* — L'inversion du système linéaire (10) suppose en théorie que l'on connaisse parfaitement les échantillons  $r[k]$  de la fonction de corrélation, ce qui n'est bien sûr jamais le cas en pratique. Passer des valeurs exactes  $r[k]$  à des valeurs estimées  $\hat{r}[k]$  est donc susceptible de poser des problèmes de mauvais conditionnement à la matrice de corrélation à inverser, ceci étant d'autant plus probable que les valeurs de  $r[k]$  sont proches de zéro et/ou mal estimées, les deux phénomènes se conjuguant pour les grands retards. À cet égard, une première précaution est de considérer le système de Yule-Walker (valable en théorie pour tout  $j \geq 1$ ) pour les seules  $p$  premières valeurs de la corrélation, c'est-à-dire,  $j = 1, \dots, p$ .
2. *Ordre inconnu* — Une autre hypothèse implicite est que l'on sache quel est l'ordre du filtre de prédiction à choisir, mais celui-ci est en fait aussi une variable du problème. En effet, l'idée sous-jacente est de trouver la meilleure prédiction possible, c'est-à-dire le meilleur jeu de coefficients pour un ordre donné mais aussi, pour différents ordres envisageables, celui qui fournira la plus faible des erreurs minimales. La conséquence de cette difficulté est que, en résolvant le problème sous sa forme de Yule-Walker telle qu'en (10), il faudra chercher autant de solutions que l'on testera d'ordres, les solutions pour des ordres différents nécessitant *a priori* de résoudre à chaque fois un nouveau système.
3. *Complexité* — La troisième difficulté est que, pour un ordre fixé  $p$ , l'inversion directe du système linéaire (10) est en  $O(p^3)$  pour le nombre d'opérations élémentaires (additions et multiplications) mises en jeu. Si l'on conjugue cette observation avec le point précédent, on voit que

la complexité algorithmique va croissant lorsque l'on teste des ordres  $p$  de plus en plus grands, situation qui (comme mentionné au point 1) s'assortit de conditionnement de plus en plus mauvais de la matrice à inverser.

L'ensemble de ces remarques a conduit à rechercher des résolutions du système (10) qui soient meilleures que par un inversion directe, ordre par ordre. L'idée est d'exploiter la structure particulière du système linéaire considéré, qui n'est pas quelconque, afin d'obtenir une solution qui soit *récursive sur l'ordre*, de telle sorte qu'en explorant les solutions à ordre  $p$  croissant, celle à l'ordre  $p$  puisse se construire sur la base de celle à l'ordre  $p-1$ , moyennant la connaissance d'une valeur (estimée) supplémentaire de la fonction de corrélation.

Pour ce faire, il convient de ré-écrire les deux équations (10)-(11) en les regroupant en une seule selon ;

$$\mathbf{R}(p) \mathbf{a}(p) = \mathbf{P}(p), \quad (12)$$

avec

$$\begin{aligned} \mathbf{R}(p) &:= \{r[i-j]\}_{i,j=1,\dots,p}, \\ \mathbf{a}(p) &:= (1 \ a_1(p) \ \dots \ a_p(p))^T, \\ \mathbf{P}(p) &:= (P(p) \ 0 \ \dots \ 0)^T. \end{aligned}$$

(Dans cette ré-écriture, la dépendance des différentes quantités par rapport à l'ordre considéré a été rendue explicite.)

Si l'on suppose maintenant connue la solution à l'ordre  $p$ , la solution à l'ordre  $p+1$ , qui s'écrit

$$\mathbf{R}(p+1) \mathbf{a}(p+1) = \mathbf{P}(p+1),$$

met en jeu une matrice de corrélation  $\mathbf{R}(p+1)$  augmentée d'une ligne et d'une colonne par rapport à la matrice  $\mathbf{R}(p)$  intervenant dans (12). Or une matrice de corrélation a une structure très particulière, dite *Toeplitz symétrique*, qui la rend entièrement déterminée par la connaissance de sa seule première ligne (ou colonne), les autres éléments s'en déduisant par translation le long de chacune de ses diagonales. Il s'ensuit que l'on retrouve la matrice  $\mathbf{R}(p)$  comme sous-matrice de  $\mathbf{R}(p+1)$ , et ceci de deux façons différentes :

$$\begin{aligned} \mathbf{R}(p) &= [\mathbf{R}(p+1)]_{i,j=1,\dots,p}, \\ \mathbf{R}(p) &= [\mathbf{R}(p+1)]_{i,j=2,\dots,p+1}, \end{aligned}$$

C'est cette structure que l'on va exploiter en faisant apparaître dans le système à l'ordre  $p+1$  la solution à l'ordre  $p$ . Pour ce faire, on met le vecteur qui est solution à l'ordre  $p+1$  sous la forme :

$$\mathbf{a}(p+1) = \mathbf{b}_+(p) + k(p) \mathbf{b}_-(p), \quad (13)$$

avec

$$\mathbf{b}_+(p) = (\mathbf{a}(p)^T \mathbf{0})^T$$

et  $\mathbf{b}_-(p)$  tel que

$$[\mathbf{b}_-(k) = \mathbf{b}_+(p+2-k)]_{k=1\dots p+1},$$

le facteur  $k(p)$  étant un terme à déterminer.

Ceci permet, par identification des deux sous-systèmes qui correspondent à la solution connue à l'ordre  $p$  dans le système à l'ordre  $p+1$ , d'aboutir aux deux égalités :

$$\begin{aligned} P(p) + k(p) Q(p) &= P(p+1), \\ Q(p) + k(p) P(p) &= 0, \end{aligned}$$

avec

$$Q(p) := r[p+1] + a_1(p)r[p] + \dots + a_p(p)r[0].$$

On en déduit les deux valeurs :

$$k(p) = -\frac{Q(p)}{P(p)} \quad (14)$$

et

$$P(p+1) = P(p) [1 - k^2(p)].$$

Revenant à la définition (13), on obtient la forme récursive pour les coefficients de prédiction :

$$a_m(p+1) = a_m(p) + k(p) a_{p+1-m}(p); 1 \leq m \leq p$$

et

$$a_{p+1}(p+1) = k(p).$$

Disposant ainsi des relations de récursion permettant de passer de la solution à l'ordre  $p$  à la solution à l'ordre  $p+1$ , il suffit alors d'initialiser

l'algorithme (dit de *Levinson-Durbin*) en partant de la solution à l'ordre  $p = 1$  qui s'écrit trivialement

$$a_1(1) = -\frac{r[1]}{r[0]},$$

avec

$$P(1) = r[0] [1 - a_1^2(1)].$$

## 3.2 Vers la modélisation

Si l'on repart de l'équation (7) qui définit l'erreur de la prédiction linéaire, il est possible de lui attacher une interprétation alternative en voyant le signal  $x[n]$  comme définissant la sortie d'un filtre linéaire dont l'entrée serait  $e[n]$ . Étant donné que la relation d'entrée-sortie de ce filtre s'écrit

$$x[n] - \sum_{k=1}^p a_k x[n-k] = e[n],$$

il possède une structure récursive (la valeur de la sortie du filtre à un instant donné dépend de l'entrée à cet instant et des *valeurs passées de la sortie*) lui valant la dénomination de *filtre auto-régressif* (ou AR en abrégé) d'ordre  $p$ . Le rôle joué par la puissance de l'erreur de prédiction  $P$  est alors tenu par la variance  $\sigma^2$  de l'entrée  $e[n]$  (qualifiée d'*innovation*).

En ce sens, *prédire* linéairement un signal est équivalent à le *modéliser* en considérant qu'il est la sortie d'un filtre générateur de type AR.

## 3.3 Exemples

*Exemple 1. AR\_predict.m* (Prédiction simple) — Étant donné un filtre auto-régressif (d'ordre maximal 4), on génère des échantillons d'une trajectoire du processus associé  $x[n]$  en excitant ce filtre en entrée par un bruit blanc gaussien  $e[n]$ . Si l'on part maintenant de cette réalisation et qu'on la filtre en combinant linéairement, pour chaque instant  $n$ , les  $p$  échantillons passés avec les poids correspondant aux coefficients du filtre, on obtient une valeur estimée  $\hat{x}[n]$  différant de la valeur vraie d'exactly  $e[n]$ . Si l'on ne réactualise par contre pas la prédiction à chaque instant et si l'on s'appuie, non sur les  $p$  dernières valeurs observées, mais sur celles prédites précédemment, on observe que la prédiction  $\hat{x}[n]$  tend vers 0, qui est la moyenne de  $x[n]$ .

*Exemple 2. AR\_change.m et ARMA\_change.m (Détection de rupture)* — Étant donné un processus stationnaire, l'erreur associée à sa prédiction linéaire est de puissance constante au cours du temps, ce qui, sous hypothèse gaussienne, maintient cette puissance inférieure à un seuil avec une probabilité que l'on peut fixer. Si par contre l'observation est brutalement changée en commutant sur un nouveau processus stationnaire pour lequel la prédiction ne peut plus être assurée avec le jeu de coefficients initial, l'erreur augmente et sa puissance dépasse le seuil avec une probabilité plus grande que celle attendue dans le cas précédent. Ceci permet une détection de changement dans un processus par suivi de la puissance de son erreur de prédiction.