

Estimación y Predicción en Series Temporales

Filtros Adaptivos – descenso por gradiente

Departamento de Procesamiento de Señales

Instituto de Ingeniería Eléctrica
Facultad de Ingeniería

2022

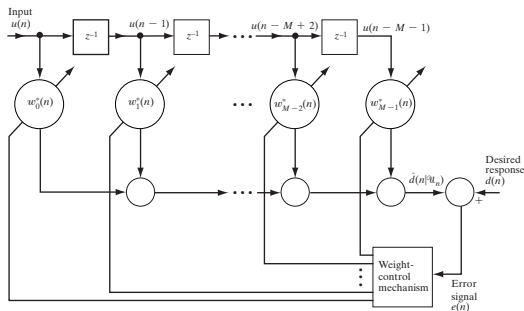
Haykin, Adaptive Filter Theory, 4.^a edición, Cap. 4 y 5).

Hayes, Statistical Digital Signal Processing and Modeling (1996), Cap. 9.

Agenda

- Desarrollaremos la teoría de filtros adaptivos contruídos sobre la base de un filtro FIR.
- Este tipo de filtro no requiere un conocimiento exacto de la estadística de las señales a ser filtradas.
- Identificamos **dos procesos básicos en un filtrado adaptivo**:
 - 1 **El proceso adaptivo**, que incluye el ajuste automático de los coeficientes del filtro de acuerdo a cierto algoritmo.
 - 2 **El proceso de filtrado**, que incluye generar la salida del filtro y estimar el error en la estimación, que luego se utiliza para modificar los coeficientes. **De esta forma se cierra el lazo de realimentación.**
- Veremos:
 - **Método de máxima pendiente (steepest descent)**. Vieja técnica de optimización recursiva para buscar el mínimo de una superficie cuando este no se conoce explícitamente.
 - El algoritmo **LMS (Least Mean Squares)**. Menos exacto pero menos costoso: no requiere estimar funciones de correlación ni invertir matrices.

Estructura del filtro adaptivo



Recordemos: $e(n) = d(n) - \hat{d}(n|\mathcal{U}_n) = d(n) - \mathbf{w}^H(n)\mathbf{u}(n)$

$$J(\mathbf{w}) = \sigma_d^2 - \mathbf{w}^H(n)\mathbf{p} - \mathbf{p}^H\mathbf{w}(n) + \mathbf{w}^H(n)\mathbf{R}\mathbf{w}(n)$$
$$\mathbf{R}\mathbf{w}_o = \mathbf{p}, \quad J_{min} = \sigma_d^2 - \mathbf{p}^H\mathbf{w}_o.$$

- En **condiciones estacionarias** el filtro adaptivo se diseña para estar cerca de $\nabla_{\mathbf{w}} J = 0$.
- En **condiciones no-estacionarias**, la superficie de error cambia (estacionaridad local, \mathbf{R} y \mathbf{p} van variando con n).

El método de máxima pendiente

- El filtro debe satisfacer $\mathbf{R}\mathbf{w}_o = \mathbf{p}$ en todo momento, con \mathbf{R} y \mathbf{p} variando en el tiempo.
- Una forma es obtener una solución analítica para cada n lo cual puede ser muy costoso (M grande) y necesitar gran velocidad de procesamiento.

Una alternativa es el **método de máxima pendiente**:

- 1 $\mathbf{w}(0) = \mathbf{w}_o$. (cierto valor elegido, en general puede ser 0).
- 2 Luego nos movemos en la dirección opuesta al gradiente:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{2}\mu \nabla_{\mathbf{w}} J(\mathbf{w}(n)), \quad \mu > 0,$$

$$\text{dónde } \nabla_{\mathbf{w}} J(\mathbf{w}(n)) = 2 \frac{\partial J(\mathbf{w}(n))}{\partial \mathbf{w}^*(n)} = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n).$$

$$\Rightarrow \mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)], \quad n = 0, 1, 2, \dots$$

Obs: μ determina el paso de ajuste.

Estabilidad del algoritmo de máxima pendiente. Caso WSS

Al incluir realimentación, el algoritmo puede tornarse inestable. Es claro que la inestabilidad dependerá de μ y de \mathbf{R} . Para estudiar la estabilidad debemos observar los modos naturales del sistema, es decir qué sucede en la base propia de \mathbf{R} :

- $\mathbf{c}(n) := \mathbf{w}(n) - \mathbf{w}_o$: vector de error en los coeficientes en tiempo n .
- Teníamos $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)]$. Usando que $\mathbf{R}\mathbf{w}_o = \mathbf{p}$, tenemos

$$\mathbf{c}(n+1) = (\mathbf{I} - \mu\mathbf{R})\mathbf{c}(n) = (\mathbf{I} - \mu\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H)\mathbf{c}(n),$$

siendo las columnas de \mathbf{Q} los vect. ppios de \mathbf{R} , \mathbf{Q} unitaria ($\mathbf{Q}^H = \mathbf{Q}^{-1}$) y $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_M)$ con los λ_i positivos (teorema espectral + \mathbf{R} definida positiva).

Usando el cambio a coordenadas intrínsecas $\mathbf{v}(n) = \mathbf{Q}^H \mathbf{c}(n)$ demostrar que para cada modo normal,

$v_i(n+1) = (1 - \mu\lambda_i)v_i(n)$, $i = 1, 2, \dots, M$. Deduzcamos la CNG

Teníamos que $\mathbf{c}(n+1) = (\mathbf{I} - \mu \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H) \mathbf{c}(n)$, por lo que

$$\mathbf{Q}^H \mathbf{c}(n+1) = (\mathbf{I} - \mu \mathbf{\Lambda}) \mathbf{Q}^H \mathbf{c}(n).$$

Utilizando el cambio de coordenadas $\mathbf{v}(n) = \mathbf{Q}^H \mathbf{c}(n)$, tenemos

$$\mathbf{v}(n+1) = (\mathbf{I} - \mu \mathbf{\Lambda}) \mathbf{v}(n).$$

Esto es para cada modo natural, $k = 1, 2, \dots, M$,

$$v_k(n+1) = (1 - \mu \lambda_k) v_k(n);$$

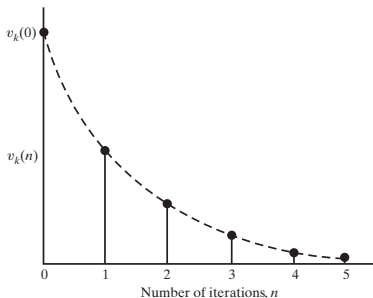
$$\text{Luego, } v_k(n) = (1 - \mu \lambda_k)^n v_k(0).$$

La convergencia de esta serie geométrica se verifica si

$$-1 < 1 - \mu \lambda_k < 1 \stackrel{\lambda_k > 0}{\iff} 0 < \mu < \frac{2}{\lambda_k} \quad \forall k = 1, 2, \dots, M.$$

$$\Rightarrow \boxed{0 < \mu < \frac{2}{\lambda_{max}}} \quad \text{CNS de convergencia para } \mathbf{w}(n) \xrightarrow{n \rightarrow \infty} \mathbf{w}_o$$

Convergencia del algoritmo de máxima pendiente, caso WSS



Definimos τ_k como el tiempo que demora el modo natural k -ésimo en decaer a $1/e$ de su valor inicial:

$$(1 - \mu\lambda_k)^{\tau_k} = \exp(-1) \Rightarrow \tau_k = \frac{-1}{\ln(1 - \mu\lambda_k)},$$

y si μ es lo suficientemente pequeño para que $\mu\lambda_k \ll 1$, $\tau_k \approx \frac{1}{\mu\lambda_k}$. Luego, si definimos el tiempo global, este viene dado por

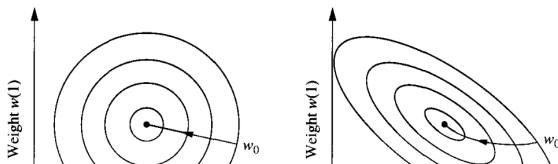
Convergencia del algoritmo de máxima pendiente, caso WSS

Relación con el número de condición de \mathbf{R}

- Por un lado, tenemos la CNS de convergencia,
 $0 < \mu < \frac{2}{\lambda_{max}}$, o de forma equivalente $\mu = \alpha \frac{2}{\lambda_{max}}$, con
 $0 < \alpha < 1$.
- Por otro lado, el tiempo global de convergencia es
 $\tau \approx \frac{1}{\mu \lambda_{min}}.$

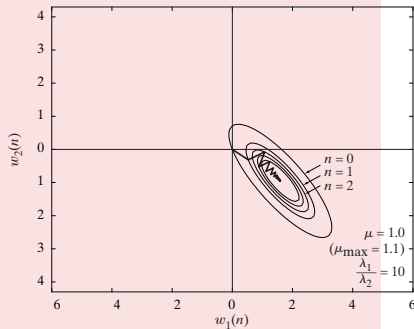
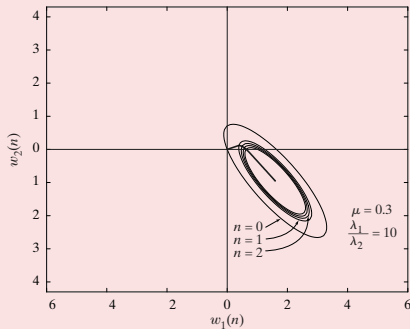
Entonces:

$$\tau \approx \frac{1}{2\alpha} \frac{\lambda_{max}}{\lambda_{min}} = \frac{1}{2\alpha} \chi(\mathbf{R})$$



Convergencia del algoritmo de máxima pendiente, caso WSS

Influencia de μ



Comportamiento Transitorio de $\mathbf{w}(n)$, caso WSS

Teníamos $\mathbf{v}(n) = \mathbf{Q}^H \mathbf{c}(n) = \mathbf{Q}^H (\mathbf{w}(n) - \mathbf{w}_o)$.

Pre-multiplicando por \mathbf{Q} :

$$\begin{aligned}\mathbf{w}(n) &= \mathbf{w}_o + \mathbf{Q}\mathbf{v}(n) \\ &= \mathbf{w}_o + [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M]\mathbf{v}(n), \quad \mathbf{v}(n) = (v_1(n), v_2(n), \dots, v_M(n))^T \\ &= \mathbf{w}_o + \sum_{k=1}^M v_k(n)\mathbf{q}_k = \mathbf{w}_o + \sum_{k=1}^M v_k(0)(1 - \mu\lambda_k)^n \mathbf{q}_k\end{aligned}$$

Error cuadrático medio

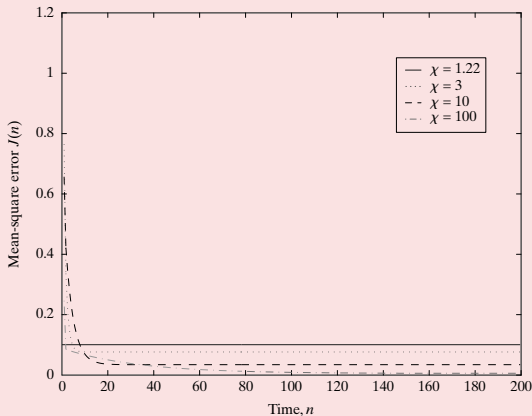
Teníamos $J(n) = J_{min} + \sum_{k=1}^M \lambda_k |v_k(n)|^2$. Sustituyendo,

$$J(n) = J_{min} + \sum_{k=1}^M \lambda_k (1 - \mu\lambda_k)^{2n} |v_k(0)|^2.$$

- Si el algoritmo converge, entonces $\lim_{n \rightarrow +\infty} J(n) = J_{min}$, cualquiera sean las condiciones iniciales

Comportamiento Transitorio de $w(n)$, caso WSS

Curva de aprendizaje $n \mapsto J(n)$



$$\mu = 0.3$$