

Estimación y Predicción en Series Temporales

Filtro Adaptativo LMS

Departamento de Procesamiento de Señales

Instituto de Ingeniería Eléctrica
Facultad de Ingeniería

2022

Haykin, Adaptive Filter Theory, 4.^a edición, Cap. 5 y 6).

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

Introducción - Motivación

- El algoritmo de *steepest descent* supone el conocimiento de $\nabla_{\mathbf{w}} J(\mathbf{w}(n))$ y una elección correcta de μ .
- En muchos casos es imposible contar con esta información, y debemos estimar $\nabla_{\mathbf{w}} J(\mathbf{w}(n))$ a partir de los datos.
- Entre los algoritmos que estiman el gradiente, el LMS es el más simple: no necesita medir correlaciones ni inversiones matriciales.
- La forma de hacerlo es substituir \mathbf{p} y \mathbf{R} en la expresión de $\nabla_{\mathbf{w}} J(\mathbf{w}(n))$ por estimaciones. LMS utiliza la estimación más trivial: los valores instantáneos.

$$\nabla_{\mathbf{w}} J(\mathbf{w}(n)) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n)$$

$$\hat{\mathbf{R}}(n) = \mathbf{u}(n)\mathbf{u}(n)^H$$

$$\hat{\mathbf{p}}(n) = \mathbf{u}(n)d^*(n)$$

El algoritmo LMS

De esta forma, $\widehat{\nabla_{\mathbf{w}} J(\mathbf{w}(n))} = -2\mathbf{u}(n)d^*(n) + 2\mathbf{u}(n)\mathbf{u}(n)^H \widehat{\mathbf{w}}(n)$.

$$\text{Teníamos } \mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{2}\mu \nabla_{\mathbf{w}} J(\mathbf{w}(n))$$

$$\Rightarrow \widehat{\mathbf{w}}(n+1) = \widehat{\mathbf{w}}(n) + \mu \mathbf{u}(n)[d^*(n) - \mathbf{u}^H(n)\widehat{\mathbf{w}}(n)].$$

O de forma equivalente,

$$\begin{cases} y(n) &= \widehat{\mathbf{w}}^H(n)\mathbf{u}(n) \\ e(n) &= d(n) - y(n) \\ \widehat{\mathbf{w}}(n+1) &= \widehat{\mathbf{w}}(n) + \mu e^*(n)\mathbf{u}(n) \end{cases}$$

Observaciones

- 1 $\mu e^*(n)\mathbf{u}(n)$ es la corrección en el estimado de $\mathbf{w}(n)$.
- 2 Dirección del gradiente muy ruidosa, puede estar muy lejos de la real.
- 3 A pesar de que los estimados de \mathbf{R} y \mathbf{p} son instantáneos, la performance global es buena. ¿Porqué? Porque el mismo algoritmo, al acumular, promedia los estimados.

Complejidad del método LMS

Si el filtro tiene M coeficientes:

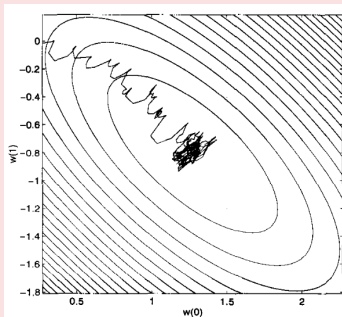
	\times	$+$
Cálculo de la salida $y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n)$	M	$M - 1$
Actualización de coeficientes	M	M
Cálculo del error $e(n) = d(n) - y(n)$	$-$	1
$\mu e^*(n)$	1	$-$
TOTAL:	$2M + 1$	$2M.$

Convergencia del algoritmo LMS

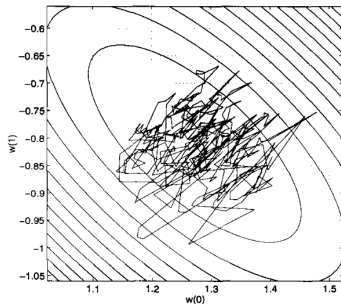
- LMS estima el gradiente mediante valores instantáneos. En general, la corrección aplicada a los coeficientes del filtro no va a ser en la dirección de máxima pendiente.
- Como $\mathbb{E}[\widehat{\nabla_{\mathbf{w}} J(\mathbf{w}(n))}] = -2\mathbb{E}[\mathbf{u}(n)e^*(n)] = \nabla_{\mathbf{w}} J(\mathbf{w}(n))$, la corrección se hace, en media, en la dirección de máxima pendiente.

Ejemplo: convergencia típica de LMS

Dos coeficientes, 500 iteraciones



$$\mathbf{w}(0) = (0, 0)$$



$$\mathbf{w}(0) = \mathbf{R}^{-1}\mathbf{p}$$

Convergencia de LMS: marco estadístico

- Suponemos $u(n)$ y $d(n)$ conjuntamente WSS.
- Nos interesa saber las condiciones para las cuales

$$\lim_{n \rightarrow +\infty} \mathbb{E}[\mathbf{w}(n)] = \mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}.$$

- Tomando esperanzas en las ecuaciones del LMS:

$$\mathbb{E}[\hat{\mathbf{w}}(n+1)] = \mathbb{E}[\hat{\mathbf{w}}(n)] + \mu \mathbb{E}[\mathbf{u}(n)d^*(n)] - \mu \mathbb{E}[\mathbf{u}(n)\mathbf{u}^H(n)\hat{\mathbf{w}}(n)].$$

El último término es difícil de evaluar; **supondremos para simplificar que $\mathbf{u}(n)$ y $\mathbf{w}(n)$ son estadísticamente independientes.** Esta hipótesis es claramente no cierta del todo, pero conduce a resultados que concuerdan relativamente con las simulaciones.

- De esta forma,

$$\begin{aligned}\mathbb{E}[\hat{\mathbf{w}}(n+1)] &= \mathbb{E}[\hat{\mathbf{w}}(n)] + \mu \mathbb{E}[\mathbf{u}(n)d^*(n)] - \mu \mathbb{E}[\mathbf{u}(n)\mathbf{u}^H(n)]\mathbb{E}[\hat{\mathbf{w}}(n)] \\ &= (\mathbf{I} - \mu\mathbf{R})\mathbb{E}[\hat{\mathbf{w}}(n)] + \mu\mathbf{p}.\end{aligned}$$

Convergencia de LMS: marco estadístico (cont.)

- La ecuación anterior es la misma que fue obtenida para el algoritmo de máxima pendiente, por lo que vale el mismo análisis.
- Por lo tanto: Para un proceso WSS, el algoritmo LMS converge en media si $0 < \mu < \frac{2}{\lambda_{max}}$, y si la hipótesis de independencia se satisface.

La condición anterior tiene dos inconvenientes:

- 1 La convergencia de la media de $\mathbf{w}(n)$ no implica que permanezca acotado.
- 2 No conocemos λ_{max} .

Para el segundo punto, tenemos una forma razonable de solucionarlo: sabemos que $\lambda_{max} \leq \sum_{k=1}^M \lambda_k = \text{Traza}(\mathbf{R})$. Luego, como \mathbf{R} es Toeplitz, $\text{Traza}(\mathbf{R}) = Mr(0) = M\mathbb{E}[|u(n)|^2]$.

Tenemos entonces una condición suficiente para la convergencia del algoritmo LMS: $0 < \mu < \frac{2}{M\mathbb{E}[|u(n)|^2]}$.

Podemos por ejemplo estimar $\mathbb{E}[|u(n)|^2] = \frac{1}{N} \sum_{k=0}^{N-1} |u(n-k)|^2$.

Criterios de convergencia (cont.)

Algunos criterios posibles:

- **Convergencia de la media:** $\mathbb{E}[\hat{\mathbf{w}}(n) - \mathbf{w}_o] \xrightarrow{n \rightarrow +\infty} \mathbf{0}$ (la solución de Wiener). Criterio de poco valor práctico, cualquier secuencia de valor medio cero converge en este sentido.
- **Convergencia en media:** $\mathbb{E}[\|\hat{\mathbf{w}}(n) - \mathbf{w}_o\|] \xrightarrow{n \rightarrow +\infty} 0$. Más fuerte. Difícil de probar debido a la singularidad en 0.
- **Convergencia en media cuadrática:**

$$\mathcal{D}(n) := \mathbb{E}[\|\hat{\mathbf{w}}(n) - \mathbf{w}_o\|^2] \xrightarrow{n \rightarrow +\infty} 0, \quad (\mathcal{D}(n): \text{Desvío del error cuadrado})$$

Es fácil de minimizar (funcional convexo diferenciable).

Otra condición para describir la convergencia del LMS es

$$J(\mathbf{w}(n)) = \mathbb{E}[e^2(n)] \xrightarrow{n \rightarrow +\infty} \text{cte.}$$

Convergencia en media cuadrática del método LMS

Definimos el error en exceso como: $J_{ex}(n) = J(\mathbf{w}(n)) - J_{min}$.
Es fácil mostrar que

$$\lambda_{min}(\mathbf{R})\mathcal{D}(n) \leq J_{ex}(n) \leq \lambda_{max}(\mathbf{R})\mathcal{D}(n), \quad \forall n.$$

Entonces el decaimiento de $J_{ex}(n)$ y $\mathcal{D}(n)$ son matemáticamente equivalentes. Nos focalizaremos entonces en el estudio de $J(\mathbf{w}(n))$.

Tenemos

$$\begin{aligned} J(\mathbf{w}(n)) &= J_{min} + (\mathbf{w}(n) - \mathbf{w}_o)^T \mathbf{R}(\mathbf{w}(n) - \mathbf{w}_o) \\ &= J_{min} + \mathbf{c}(n)^T \mathbf{R} \mathbf{c}(n) \\ &= J_{min} + J_{ex}(n). \end{aligned}$$

El cálculo de $J_{ex}(n)$ no es sencillo ([ver Haykin](#)). Es posible establecer la siguiente propiedad asintótica:

$$\left\{ \begin{array}{l} J(\infty) = J_{min} + J_{ex}(\infty) \\ J_{ex}(\infty) = \frac{J_{min}}{1 - \mu \sum_{k=0}^{M-1} \lambda_k / (2 - \mu \lambda_k)} \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} 1. \quad 0 < \mu < 2/\lambda_{max} \\ 2. \quad \mu \sum_{k=0}^{M-1} \lambda_k / (2 - \mu \lambda_k) < 1 \end{array} \right.$$

Convergencia en media cuadrática del método LMS (cont.)

Luego,

$$J_{ex}(\infty) = J(\infty) - J_{min} = \mu J_{min} \frac{\sum_{k=0}^{M-1} \lambda_k / (2 - \mu \lambda_k)}{1 - \mu \sum_{k=0}^{M-1} \lambda_k / (2 - \mu \lambda_k)}.$$

Si $\mu \ll 2/\lambda_{max}$ (cierto en general), entonces:

- $\mu \lambda_k \ll 2 \Rightarrow$ la condición (2) anterior se vuelve $\mu < 2/\text{Tr}[\mathbf{R}]$.
- $J(\infty) \approx J_{min} \frac{1}{1 - \frac{1}{2}\mu \text{Tr}[\mathbf{R}]}$
- $J_{ex}(\infty) \approx \mu J_{min} \frac{\frac{1}{2}\text{Tr}[\mathbf{R}]}{1 - \frac{1}{2}\mu \text{Tr}[\mathbf{R}]} \approx \frac{1}{2}\mu J_{min} \text{Tr}[\mathbf{R}]$

Defs.: Desajuste, valor propio promedio

Desajuste: $\mathcal{M} = \frac{J_{ex}(\infty)}{J_{min}} \stackrel{\mu \ll 2/\lambda_{max}}{\approx} \mu \frac{\frac{1}{2}\text{Tr}[\mathbf{R}]}{1 - \frac{1}{2}\mu \text{Tr}[\mathbf{R}]} \approx \frac{1}{2}\mu \text{Tr}[\mathbf{R}]$

OBS: recordar que $\text{Tr}[\mathbf{R}] =$ potencia en los retardos, potencia en el vector de observación.

Valor propio promedio: $\lambda = \frac{1}{M} \sum_{k=0}^{M-1} \lambda_k$

Convergencia en media cuadrática del método LMS (cont.)

- Supongamos que aproximamos la curva de aprendizaje (promedio de realizaciones o ensambles) por una sola exponencial.
- La constante de tiempo será (siguiendo el mismo rationale que para máxima pendiente): $\tau_{mse,av} = \frac{1}{2\mu\lambda_{av}}$.

- Teníamos: $\mathcal{M} \approx \frac{\mu}{2} \sum_{k=1}^M \lambda_k$
$$\Rightarrow \mathcal{M} \approx \frac{\mu}{2} M \lambda_{av} \approx \frac{M}{4\tau_{mse,av}}.$$

Observaciones:

- 1 Para $\tau_{mse,av}$ fijo, $\mathcal{M} \propto M$.
- 2 $\mathcal{M} \propto \frac{1}{\tau_{mse,av}} \propto \frac{1}{\text{tiempo para alcanzar el valor final}}$
- 3 $\left. \begin{array}{l} \mathcal{M} \propto \mu \\ \tau_{mse,av} \propto 1/\mu \end{array} \right\} \Rightarrow \mu \text{ es elección de compromiso.}$

Comparación: Máxima pendiente y LMS

- El error cuadrático medio es mínimo cuando $\mathbf{w}(n) = \mathbf{w}_o$.
- El algoritmo de máxima pendiente:
 - Converge a \mathbf{w}_o cuando $n \rightarrow +\infty$. Puede hacerlo porque utiliza el valor exacto del gradiente para cada iteración.
 - Tiene una curva de aprendizaje ($n \mapsto J(n)$) bien definida, constituida por exponenciales decrecientes según los modos, tantos como el orden del filtro.
- El algoritmo LMS:
 - Utiliza estimadores ruidosos del gradiente y por lo tanto sólo puede aproximarse a \mathbf{w}_o y luego fluctuar entorno a éste.
 - Definimos el error cuadrático medio como $J(\infty)$, y el error cuadrático medio en exceso como $J_{ex}(\infty) = J(\infty) - J_{min}$.
 - La curva de aprendizaje es ruidosa. Para analizar la dinámica del filtro necesitamos estudiar $\mathbb{E}[J(n)]$. En la práctica el ensamble de promedios se aproxima con 50 a 200 simulaciones independientes.

Los tres factores que afectan el comportamiento del algoritmo son: μ , M , y $\{\lambda_i\}$. Podemos resumir los efectos individuales de la siguiente forma:

μ :

- Se puede interpretar como la memoria del algoritmo, ya que determina el peso relativo que se le asigna a las nuevas observaciones.
- Cuando μ es grande, la adaptación es rápida. Utilizando menor cantidad de datos, el error en exceso es mayor.
- Cuando μ es pequeño, la adaptación es más lenta. El filtro utiliza más datos (tiene más memoria), el error en exceso es menor.

M :

- Las propiedades de convergencia del error cuadrático medio $\mathbb{E}[J(n)]$ dependen de M .
- $\mathbb{E}[\mathbf{w}(n)]$ no depende de M .
- $\mathbb{E}[J(n)]$ converge $\Leftrightarrow 0 < \mu < \frac{2}{\sum_{k=1}^M \lambda_k}$. En este caso se dice que el LMS converge en media cuadrática.
- $\mathbb{E}[\hat{\mathbf{w}}(n)]$ converge $\Leftrightarrow 0 < \mu < \frac{2}{\lambda_{max}}$. En este caso se dice que LMS converge en la media.
- Dado que $\lambda_{max} \leq \sum_{k=1}^M \lambda_k$, si $\mathbb{E}[J(n)]$ converge $\Rightarrow \mathbb{E}[\hat{\mathbf{w}}(n)]$ converge.

$\{\lambda_i\}$:

- Cuando los valores propios de \mathbf{R} están muy separados, tenemos:
 - J_{ex} queda determinado fundamentalmente por los λ_i grandes.
 - El tiempo de convergencia de $\mathbb{E}[\hat{\mathbf{w}}(n)]$ está limitado por los λ_i pequeños.
 - $J_{ex}(\infty) \approx \frac{1}{2}\mu J_{min} \text{Tr}[\mathbf{R}]$, $\mathcal{M} = \frac{J_{ex}}{J_{min}} \approx \frac{1}{2}\mu \text{Tr}[\mathbf{R}]$

Resumen del algoritmo LMS

- Parámetros: M : número de coeficientes; μ : paso de ajuste, $0 < \mu < \frac{2}{\text{potencia de entrada}} = \frac{2}{Mr(0)}$.
- Condiciones iniciales: $\hat{\mathbf{w}}(0) = \mathbf{0}$.
- Datos: $\mathbf{u}(n)$: vector de entrada; $d(n)$: respuesta deseada
- Valores a calcular: $\hat{\mathbf{w}}(n+1)$
- Cálculo:

$$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n)$$
$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)e^*(n)$$

Algoritmo LMS en un entorno no estacionario

El entorno no estacionario puede deberse a dos motivos:

- 1 La estadística de la respuesta deseada varía en el tiempo. Ejemplo: identificación de sistemas. En este caso \mathbf{R} queda fija y \mathbf{p} cambia.
- 2 $\{u(n)\}$ no estacionario. Ejemplo: ecualizador de canal variante en el tiempo. En este caso \mathbf{R} y \mathbf{p} cambian.

El algoritmo tiene la tarea de encontrar el mínimo de la superficie de error y también seguir ("track") la posición de éste.

Sea $\mathbf{w}_o(n)$ el valor óptimo en el tiempo n (cambia).

$$\begin{aligned}\varepsilon(n) &= \hat{\mathbf{w}}(n) - \mathbf{w}_o(n) \\ &= \underbrace{\hat{\mathbf{w}}(n) - \mathbb{E}[\hat{\mathbf{w}}(n)]}_{\varepsilon_1(n)} + \underbrace{\mathbb{E}[\hat{\mathbf{w}}(n)] - \mathbf{w}_o(n)}_{\varepsilon_2(n)}\end{aligned}$$

$\varepsilon_1(n)$: errores en la estimación del gradiente (*weight vector noise*);

$\varepsilon_2(n)$: retraso en el proceso adaptativo (*weight vector lag*).

Obso. En el caso estacionario

LMS Normalizado (NLMS)

Una de las dificultades en la implementación del LMS es la elección de μ .

Para un proceso estacionario:

- LMS converge en la media si $0 < \mu < \frac{2}{\lambda_{max}}$.
- LMS converge en media cuadrática si $0 < \mu < \frac{2}{\text{Tr}[\mathbf{R}]}$.

En general no conocemos \mathbf{R} y se debe estimar, o al menos estimar su traza. Una forma de hacerlo es

$$\text{Tr}[\mathbf{R}] = M\mathbb{E}[|u(n)|^2],$$

y la condición de media cuadrática puede substituirse por

$$\left. \begin{aligned} 0 < \mu < \frac{2}{M\mathbb{E}[|u(n)|^2]}, \\ \mathbb{E}[|u(n)|^2] \approx \frac{1}{M} \sum_{k=1}^M |u(n-k+1)|^2 \end{aligned} \right\} \Rightarrow \begin{aligned} &\text{convergencia en media} \\ &\text{cuadrática vale si} \\ &0 < \mu < \frac{2}{\mathbf{u}(n)^H \mathbf{u}(n)}. \end{aligned}$$

LMS Normalizado (NLMS) (cont.)

Una forma de incorporar esto en LMS es usando

$$\mu(n) = \frac{\beta}{\mathbf{u}(n)^H \mathbf{u}(n)} = \frac{\beta}{\|\mathbf{u}(n)\|^2}, \quad \text{con } 0 < \beta < 2.$$

Esto da lugar al algoritmo adaptivo conocido como NLMS (Normalized LMS):

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \beta \frac{\mathbf{u}(n)}{\|\mathbf{u}(n)\|^2} e^*(n), \quad 0 < \beta < 2$$

Observaciones:

- 1 En LMS la corrección es proporcional a $\mathbf{u}(n)$. Cuando es grande, podemos tener el problema de amplificar el ruido en el cálculo del gradiente.
- 2 En el NLMS podemos tener problemas con valores de $\|\mathbf{u}(n)\|$ chicos. Una alternativa que corrige este problema es:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \beta \frac{\mathbf{u}(n)}{\|\mathbf{u}(n)\|^2 + \varepsilon} e^*(n), \quad 0 < \beta < 2, \quad \varepsilon > 0 \text{ chico}$$