Adaptive Filtering: Part III

Dr. Deepa Kundur

University of Toronto

Discrete-Time Signals and Systems

Reference:

Section 13.3 of

John G. Proakis and Dimitris G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th edition, 2007.

Advantages and Disadvantages of the LMS Algorithm

- Advantages:
  - computational simplicity!
  - single parameter Δ to tune for stability
- Disadvantages:
  - low rate of convergence
  - single parameter Δ to tune for convergence

RLS Algorithm: Background

- RLS = Recursive Least Squares
- Adopt optimization of a least squares criterion rather than MSE, so we deal directly with the data sequence $x(n)$.
- Assume $x(n) = 0$ for $n < 0$.
- Let $\mathbf{h}_M(n) \equiv$ filter coefficient vector at time n.
- Let $\mathbf{X}_M(n) \equiv$ input signal vector at time n.

$$\mathbf{h}_M(n) = \begin{bmatrix} h(0,n) \\ h(1,n) \\ h(2,n) \\ \vdots \\ h(M-1,n) \end{bmatrix} \qquad \mathbf{X}_M(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ x(n-2) \\ \vdots \\ x(n-M+1) \end{bmatrix}$$

RLS Problem Formulation

- Given observations: $\mathbf{X}_M(l)$, $l = 0, 1, \ldots, n$

$$x(0)\ x(1)\ x(2)\ x(3)\ x(4)\ \cdots x(n-3)\ x(n-2)\ x(n-1)\ x(n)$$

Example: $M = 3$:

$$\begin{aligned}
\mathbf{X}_3(0) &= [\ 0 \quad 0 \quad x(0)]^t \\
\mathbf{X}_3(1) &= [\ 0 \quad x(0)\ x(1)]^t \\
\mathbf{X}_3(2) &= [x(0)\ x(1)\ x(2)]^t \\
\mathbf{X}_3(3) &= [x(1)\ x(2)\ x(3)]^t \\
\mathbf{X}_3(4) &= [x(2)\ x(3)\ x(4)]^t \\
&\ \vdots \\
\mathbf{X}_3(n-1) &= [x(n-3)\ x(n-2)\ x(n-1)]^t \\
\mathbf{X}_3(n) &= [x(n-2)\ x(n-1)\quad x(n)\quad]^t
\end{aligned}$$

RLS Problem Formulation

- Given observations: $\mathbf{X}_M(l)$, $l = 0, 1, \ldots, n$
- Given desired sequence: $d(l)$, $l = 0, 1, \ldots, n$
- Determine: $\mathbf{h}_M(n)$ that minimizes

$$\mathcal{E}_M = \sum_{l=0}^{n} w^{n-l} |e_M(l, n)|^2$$

where

$$\begin{aligned}
e_M(l, n) &= d(l) - \hat{d}(l, n) \\
&= d(l) - \mathbf{h}_M^t(n)\ \mathbf{X}_M(l)
\end{aligned}$$

and $0 < w < 1$ is a weighting factor.

Output of Adaptive FIR Filter

Consider $l = 0, 1, \ldots, n$:

$$\begin{aligned}
\hat{d}(l, n) &= \mathbf{h}_M^t(n)\ \mathbf{X}_M(l) \\
&= [h(0, n)\ h(1, n) \cdots h(M-1, n)] \begin{bmatrix} x(l) \\ x(l-1) \\ \vdots \\ x(l-M+1) \end{bmatrix} \\
&= \sum_{k=0}^{M-1} h(k, n) x(l-k) = h(l, n) * x(l)
\end{aligned}$$

Why are there two arguments for the error e_M?

$$\mathcal{E}_M = \sum_{l=0}^{n} w^{n-l}\ |e_M(l, n)|^2$$

- There is an error associated with each (indexed with l) input $\mathbf{X}_M(l)$.
  - The number of error terms grow as $n \to \infty$.
- Each error is dependent on the time-varying adaptive filter coefficients used to compute it:
  $\mathbf{h}_M(n) = [h(0, n)\ h(1, n)\ \cdots\ h(M-1, n)]^t$
- Therefore,

$$e_M(l, n) = d(l) - \hat{d}(l, n) = d(l) - \mathbf{h}_M^t(n)\ \mathbf{X}_M(l)$$

Weighting Factor

$$\mathcal{E}_M = \sum_{l=0}^{n} \boxed{w^{n-l}} \, |e_M(l,n)|^2$$

- $0 < w < 1$
  - $l = 0, 1, \ldots, n \implies n - l = n, n-1, \ldots, 0$
  - $l = 0, 1, \ldots, n \implies 0 < w^n < w^{n-1} < \ldots, < w^0 = 1$
- Objective: weight more recent data points more heavily to allow filter coefficients to adapt to time-varying statistical characteristics of the data

Minimization of $\mathcal{E}_M$

The RLS cost function is:

$$\mathcal{E}_M = \sum_{l=0}^{n} w^{n-l} |e_M(l,n)|^2$$

Recall, MSE cost function:

$$\mathcal{E}_M = E[|e(n)|^2]$$

minimization with respect to $\mathbf{h}_M(n)$ gives the Wiener-Hopf Equations . . .

Wiener-Hopf Equations

The Wiener-Hopf equations can be represented as:

$$\mathbf{\Gamma}_M \mathbf{h}_M = \gamma_d$$

where

- $\mathbf{h}_M$ denotes the vector of adaptive filter coefficients
- γ_d is an $M \times 1$ crosscorrelation vector
- $\mathbf{\Gamma}_M$ is an $M \times M$ Hermitian autocorrelation matrix

Similarly, minimization of

$$\mathcal{E}_M = \sum_{l=0}^{n} w^{n-l} |e_M(l,n)|^2$$

with respect to $\mathbf{h}_M(n)$ yields the solution

$$\mathbf{R}_M(n)\mathbf{h}_M(n) = \mathbf{D}_M(n)$$

where

- $\mathbf{R}_M(n)$ is the signal estimated correlation matrix:

$$\mathbf{R}_M(n) = \sum_{l=0}^{n} w^{n-l} \mathbf{X}_M^*(l) \mathbf{X}_M^t(l)$$

- $\mathbf{D}_M(n)$ is the estimated crosscorrelation vector:

$$\mathbf{D}_M(n) = \sum_{l=0}^{n} w^{n-l} \mathbf{X}_M^*(l) d(l)$$

The solution is given by

$$\mathbf{h}_M(n) = \mathbf{R}_M^{-1}(n)\mathbf{D}_M(n)$$

Note: $\mathbf{R}_M(n)$ is not Toeplitz as in the LMS case for $\mathbf{\Gamma}_M$.

Recall, for LMS $\mathbf{\Gamma}_M$ is a Hermitian autocorrelation matrix.

$$\mathbf{\Gamma}_5 = \begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}(-1) & \gamma_{xx}(-2) & \gamma_{xx}(-3) & \gamma_{xx}(-4) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(-1) & \gamma_{xx}(-2) & \gamma_{xx}(-3) \\ \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(-1) & \gamma_{xx}(-2) \\ \gamma_{xx}(3) & \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(-1) \\ \gamma_{xx}(4) & \gamma_{xx}(3) & \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) \end{bmatrix} = \begin{bmatrix} \gamma_{xx}(0) & \gamma^*{}_{xx}(1) & \gamma^*{}_{xx}(2) & \gamma^*{}_{xx}(3) & \gamma^*{}_{xx}(4) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma^*{}_{xx}(1) & \gamma^*{}_{xx}(2) & \gamma^*{}_{xx}(3) \\ \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma^*{}_{xx}(1) & \gamma^*{}_{xx}(2) \\ \gamma_{xx}(3) & \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma^*{}_{xx}(1) \\ \gamma_{xx}(4) & \gamma_{xx}(3) & \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) \end{bmatrix}$$

Note: $\mathbf{\Gamma}_5^t = \mathbf{\Gamma}_5^*$ and more generally $\mathbf{\Gamma}_M^t = \mathbf{\Gamma}_M^*$ because $\boxed{\gamma_{xx}(n) = \gamma_{xx}^*(-n)}$

The solution is given by

$$\mathbf{h}_M(n) = \mathbf{R}_M^{-1}(n)\mathbf{D}_M(n)$$

Note: $\mathbf{R}_M(n)$ is not Toeplitz as in the LMS case for $\mathbf{\Gamma}_M$.

The signal estimated correlation matrix is:

$$\mathbf{R}_M(n) = \sum_{l=0}^{n} w^{n-l} \underbrace{\mathbf{X}_M^*(l)\mathbf{X}_M^t(l)}_{\text{NOT Toeplitz mx}}$$

$$\mathbf{X}_3^*(l)\mathbf{X}_3^t(l) = \underbrace{\begin{bmatrix} |x(l)|^2 & x^*(l)x(l-1) & x^*(l)x(l-2) \\ x^*(l-1)x(l) & |x(l-1)|^2 & x^*(l-1)x(l-2) \\ x^*(l-2)x(l) & x^*(l-2)x(l-1) & |x(l-2)|^2 \end{bmatrix}}_{\text{NOT Toeplitz mx}}$$

Computation of $\mathbf{R}_M^{-1}(n)$

- Adapting the filter coefficients at each time step n:

  $$\mathbf{h}_M(n) = \mathbf{R}_M^{-1}(n)\mathbf{D}_M(n)$$

  requires an $M \times M$ matrix inversion.
- Computing $\mathbf{R}_M^{-1}(n)$ from scratch each time is impactical.
- Consider the Matrix Inversion Lemma (Householder, 1964):

$$\mathbf{R}_M^{-1}(n) = \frac{1}{w}\left[\mathbf{R}_M^{-1}(n-1) - \frac{\mathbf{R}_M^{-1}(n-1)\mathbf{X}_M^*(n)\mathbf{X}_M^t(n)\mathbf{R}_M^{-1}(n-1)}{w + \mathbf{X}_M^t(n)\mathbf{R}_M^{-1}(n-1)\mathbf{X}_M^*(n)}\right]$$

Matrix Inversion Lemma Rewritten

Let

$$\begin{aligned} \mathbf{P}_M(n) &= \mathbf{R}_M^{-1}(n) \\ \mathbf{K}_M(n) &= \frac{1}{w + \mu_M(n)}\mathbf{P}_M(n-1)\mathbf{X}_M^*(n) \\ \mu_M(n) &= \mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\mathbf{X}_M^*(n) \end{aligned}$$

$$\mathbf{R}_M^{-1}(n) = \frac{1}{w}\left[\mathbf{R}_M^{-1}(n-1) - \frac{\mathbf{R}_M^{-1}(n-1)\mathbf{X}_M^*(n)\mathbf{X}_M^t(n)\mathbf{R}_M^{-1}(n-1)}{w + \mathbf{X}_M^t(n)\mathbf{R}_M^{-1}(n-1)\mathbf{X}_M^*(n)}\right]$$

becomes

$$\boxed{\mathbf{P}_M(n) = \frac{1}{w}\left[\mathbf{P}_M(n-1) - \mathbf{K}_M(n)\mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\right]}$$

Note: recursive relationship

Also, notice:

$$
\begin{aligned}
\mathbf{P}_M(n) &= \frac{1}{w}\left[\mathbf{P}_M(n-1) - \mathbf{K}_M(n)\mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\right] \\
\mathbf{P}_M(n)\mathbf{X}_M^*(n) &= \frac{1}{w}\left[\underbrace{\mathbf{P}_M(n-1)\mathbf{X}_M^*(n)}_{=[w+\mu_M(n)]\mathbf{K}_M(n)} - \mathbf{K}_M(n)\underbrace{\mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\mathbf{X}_M^*(n)}_{=\mu_M(n)}\right] \\
&= \frac{1}{w}\left[w\mathbf{K}_M(n) + \mu_M(n)\mathbf{K}_M(n) - \mathbf{K}_M(n)\mu_M(n)\right] \\
&= \mathbf{K}_M(n)
\end{aligned}
$$

where $\boxed{\mathbf{K}_M(n) = \mathbf{P}_M(n)\mathbf{X}_M^*(n)}$ is called the Kalman gain vector.

Estimated Crosscorrelation Vector

Recall $\mathbf{h}_M(n) = \mathbf{R}_M^{-1}(n)\mathbf{D}_M(n) = \mathbf{P}_M(n)\mathbf{D}_M(n)$

$$
\begin{aligned}
\mathbf{D}_M(n) &= \sum_{l=0}^{n} w^{n-l}\mathbf{X}_M^*(l)d(l) = \sum_{l=0}^{n-1} w^{n-l}\mathbf{X}_M^*(l)d(l) + \mathbf{X}_M^*(n)d(n) \\
&= \sum_{l=0}^{n-1} w \cdot w^{n-1-l}\mathbf{X}_M^*(l)d(l) + \mathbf{X}_M^*(n)d(n) \\
&= \underbrace{w\sum_{l=0}^{(n-1)} w^{(n-1)-l}\mathbf{X}_M^*(l)d(l)}_{=\mathbf{D}_M(n-1)} + \mathbf{X}_M^*(n)d(n)
\end{aligned}
$$

$$
\boxed{\mathbf{D}_M(n) = w\ \mathbf{D}_M(n-1) + d(n)\mathbf{X}_M^*(n)}
$$

Note: <u>recursive</u> relationship

<u>Goal</u>: Obtain current filter coefficients in terms of previous

$$
\begin{aligned}
\mathbf{h}_M(n) &= \mathbf{R}_M^{-1}(n)\mathbf{D}_M(n) \\
&= \mathbf{P}_M(n)\mathbf{D}_M(n) \\
&= \frac{1}{w}\left[\mathbf{P}_M(n-1) - \mathbf{K}_M(n)\mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\right] \times \\
&\qquad \left[w\ \mathbf{D}_M(n-1) + d(n)\mathbf{X}_M^*(n)\right] \\
&= \underbrace{\mathbf{P}_M(n-1)\mathbf{D}_M(n-1)}_{=\mathbf{h}_M(n-1)} + \frac{1}{w}d(n)\underbrace{\mathbf{P}_M(n-1)\mathbf{X}_M^*(n)}_{=(w+\mu_M(n))\mathbf{K}_M(n)} \\
&\qquad -\mathbf{K}_M(n)\mathbf{X}_M^t(n)\underbrace{\mathbf{P}_M(n-1)\mathbf{D}_M(n-1)}_{=\mathbf{h}_M(n-1)} \\
&\qquad -\frac{1}{w}d(n)\mathbf{K}_M(n)\underbrace{\mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\mathbf{X}_M^*(n)}_{=\mu_M(n)} \\
&= \mathbf{h}_M(n-1) + \mathbf{K}_M(n)\left[d(n) - \mathbf{X}_M^t(n)\mathbf{h}_M(n-1)\right]
\end{aligned}
$$

$$
\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + \mathbf{K}_M(n)\left[d(n) - \mathbf{X}_M^t(n)\mathbf{h}_M(n-1)\right]
$$

Recall,

$$
\begin{aligned}
\hat{d}(l,n) &= \mathbf{h}_M^t(n)\ \mathbf{X}_M(l) = \mathbf{X}_M^t(l)\ \mathbf{h}_M(n) \\
\hat{d}(n,n-1) &= \mathbf{X}_M^t(n)\ \mathbf{h}_M(n-1)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\mathbf{h}_M(n) &= \mathbf{h}_M(n-1) + \mathbf{K}_M(n)\left[d(n) - \hat{d}(n,n-1)\right] \\
&= \mathbf{h}_M(n-1) + \mathbf{K}_M(n)\ \underbrace{e_M(n,n-1)}_{\equiv e_M(n)} \\
&= \mathbf{h}_M(n-1) + \mathbf{K}_M(n)\ e_M(n)
\end{aligned}
$$

Therefore,

$$
\boxed{\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + \mathbf{K}_M(n)\ e_M(n)}
$$

Direct-Form Recursive Least Squares Algorithm

Set $\mathbf{h}_M(-1) = \mathbf{0}$
Set $\mathbf{P}_M(-1) = (1/\delta)\mathbf{I}_M$, for $\delta \ll 1$
Note: $\mathbf{X}_M(-1) = \mathbf{0}$

For $n = 0, 1, 2, \ldots$

Given: $\mathbf{h}_M(n-1)$, $\mathbf{P}_M(n-1)$, $\mathbf{X}_M(n-1)$
New observation: $x(n)$

0. Form Input Vector:
Form $\mathbf{X}_M(n)$ from $\mathbf{X}_M(n-1)$ as follows:

$$\begin{aligned}
\mathbf{X}_M(n-1) &= [x(n-M)\ x(n-M+1)\ x(n-M+2)\ \cdots\ x(n-2)\ x(n-1)] \\
&= [\underbrace{x(n-M)}_{\text{DELETE}}\ x(n-M+1)\ x(n-M+2)\ \cdots\ x(n-2)\ x(n-1)]\ \underbrace{[x(n)]}_{\text{INSERT}} \\
\mathbf{X}_M(n) &= [x(n-M+1)\ x(n-M+2)\ \cdots\ x(n-2)\ x(n-1)\ x(n)]
\end{aligned}$$

Direct-Form Recursive Least Squares Algorithm

ASIDE:

$$\begin{aligned}
\mathbf{X}_3(-1) &= [\ 0\quad 0\quad 0\ ]^t = \mathbf{0} \\
\mathbf{X}_3(0) &= [\ 0\quad 0\quad x(0)]^t \\
\mathbf{X}_3(1) &= [\ 0\quad x(0)\ x(1)]^t \\
\mathbf{X}_3(2) &= [x(0)\ x(1)\ x(2)]^t \\
\mathbf{X}_3(3) &= [x(1)\ x(2)\ x(3)]^t \\
\mathbf{X}_3(4) &= [x(2)\ x(3)\ x(4)]^t \\
&\vdots \\
\mathbf{X}_3(n-1) &= [x(n-3)\ x(n-2)\ x(n-1)]^t \\
\mathbf{X}_3(n) &= [x(n-2)\ x(n-1)\quad x(n)\ ]^t
\end{aligned}$$

$\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + \mathbf{K}_M(n)\ e_M(n)$

1. Compute the filter output:
$$\hat{d}(n) = \mathbf{X}_M^t(n)\mathbf{h}_M(n-1)$$
2. Compute the error:
$$e_M(n) = d(n) - \hat{d}(n)$$
3. Compute the Kalman gain vector:
$$\mathbf{K}_M(n) = \frac{\mathbf{P}_M(n-1)\mathbf{X}_M^*(n)}{w + \mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\mathbf{X}_M^*(n)}$$
4. Update the inverse of the correlation matrix:
$$\mathbf{P}_M(n) = \frac{1}{w}\left[\mathbf{P}_M(n-1) - \mathbf{K}_M(n)\mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\right]$$
5. Update the coefficient vector of the filter:
$$\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + \mathbf{K}_M(n)\ e_M(n)$$

LMS versus RLS

LMS:
$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \Delta\ e(n)\ \mathbf{X}_M^*(n)$$
RLS:
$$\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + \mathbf{K}_M(n)\ e_M(n)$$

- Adjustment rate is controlled by a single-parameter Δ for the LMS algorithm and the Kalman gain vector for $\mathbf{K}_M(n)$ for the RLS algorithm.
- RLS has rapid rate of convergence compared to LMS.
- LMS is computationally simpler than RLS.

■