Motivation

- $d$ be a scalar-valued random variable (desired output signal)
  - $E[d] = 0$
  - $E[d^2] = \sigma_d^2$
  - With realization $\{d(i) : i = 0, 1, 2, \ldots\}$

- $u \in \mathbb{R}^M (\mathbb{C}^M)$ be a random vector (input signal)
  - $E[u] = 0$
  - $R_u = E[u^*u] > 0$
  - $R_{du} = E[du^*]$
  - With realization $\{u_i : i = 0, 1, 2, \ldots\}$

Problem

*We want to solve*

$$\min_\omega E \left[ (d - u\omega)^2 \right]$$

(1)

*where $\omega$ is the weights vector.*
By the steepest-descent algorithm

$$\omega^o = R_u^{-1} R_{du}$$

which can be approximated by the following recursion with constant step-size $\mu > 0$

$$\omega_i = \omega_{i-1} + \mu [R_{du} - R_u \omega_{i-1}], \ \omega_{-1} = \text{initial guess}.$$  

**Remark**

$R_u$ and $R_{du}$ should be known, and fixed.
Adaptive Filters

"Smart Systems"
- Learning: Learns the Statistics of the Signal
- Tracking: Adjusts the Behavior to Signal Variations

Practical Reasons for Using Adaptive Filters
- Lack of Statistical Information
  - Mean, Variance, Auto-correlation, Cross-correlation, etc
- Variation in the Statistics of the Signal
  - Signal with Noise Randomly Moving in a Known/Unknown Bandwidth with Time

Types of Adaptive Filters
- Least Mean Square (LMS) Filters
  - Normalized LMS Filters
  - Non-Canonical LMS Filters
- Recursive Least Square (RLS) Filters
  - QR-RLS Filters
Least Mean Square (LMS) Filters
Development Using Instantaneous Approximation

- At time index $i$ approximate
  - $R_u = E[u^*u]$ by $\hat{R}_u = u_i^*u_i$
  - $R_{du} = E[du^*]$ by $\hat{R}_{du} = d(i)u_i^*$

- Corresponding steepest-descent iteration
  \[ \omega_i = \omega_{i-1} + \mu u_i^*[d(i) - u_i\omega_{i-1}], \ \omega_{-1} = \text{initial guess} \]
  where $\mu > 0$ is a constant stepsize.

- Remarks
  - Also known as the Widrow-Hoff algorithm.
  - Commonly used algorithm for simplicity.
  - $\mu$ is choosen to be $2^{-m}$ for $m \in \mathbb{N}$.

- Computational Cost
  - Complex-valued Signal: $8M + 2$ real multiplications, $8M$ real additions.
  - Real-values Signal: $2M + 1$ real multiplications, $2M$ real additions.
Least Mean Square (LMS) Filters

An Illustration

\[ u: \text{input signal} \]

\[ d: \text{desired output signal} \]

\[ \omega \]

\[ \overline{\omega} \]

\[ \text{interference} \]

Figure: An Illustration for Least Mean Square Filter
Least Mean Square (LMS) Filters
An Application (1/3)

Least Mean Square Filter

Wave Scope
Uniform Random Number

In
Gateway In

x(k)

Channel Model

d(k)

LMS Adaptive Filter

Out
Gateway Out1

Error

H. Ahsan (ECE BSU)  Adaptive Filters  April 12, 2010  7 / 17
Least Mean Square (LMS) Filters
An Application (2/3)
Solution to (1) using regularized Newton Recursion

\[ \omega_i = \omega_{i-1} + \mu(i) [\varepsilon(i) I - R_u]^{-1} [R_{du} - R_u \omega_{i-1}] , \omega_{-1} = \text{initial guess.} \]

where \( \mu(i) > 0 \) is the stepsize and \( \varepsilon(i) \) is the regularization factor.

With \( \mu(i) = \mu > 0 \) and \( \varepsilon(i) = \varepsilon \) fixed for all \( i \), using the instantaneous approximation

\[ \omega_i = \omega_{i-1} + \mu [\varepsilon I - u_i^* u_i]^{-1} u_i^* [d(i) - u_i \omega_{i-1}] \]

\[ = \cdots \]

\[ = \omega_{i-1} + \frac{\mu}{\varepsilon + \|u_i\|^2} u_i^* [d(i) - u_i \omega_{i-1}] \]

**Computational Cost**

- Complex-valued Signal: \( 10M + 2 \) real multiplications, \( 10M \) real additions and one real division.
- Real-values Signal: \( 3M + 1 \) real multiplications, \( 3M \) real additions and one real division.
Other LSM-Type Techniques

- Power Normalization
  - Replace $\frac{\mu}{\varepsilon + \|u_i\|^2}$ with $\frac{\mu / M}{\varepsilon / M + \|u_i\|^2 / M}$, where $M$ is the order of the filter.

Definition
Non-Blind algorithms are so called since they employ a reference sequence $\{d(i) : i = 0, 1, 2, \ldots\}$.

Non-Blind Algorithm
- Leaky LMS Algorithm
- LMF Algorithm
- LMMN Algorithm

Blind Algorithm
- CMA1-2, NCMA Algorithm
- CMA2-2 Algorithm
- RCA Algorithm
- MMA Algorithm
Solution to (1) using regularized Newton Recursion

\[ \omega_i = \omega_{i-1} + \mu(i) \left[ \varepsilon(i) I - R_u \right]^{-1} \left[ R_{du} - R_u \omega_{i-1} \right], \omega_{-1} = \text{initial guess.} \]

where \( \mu(i) > 0 \) is the stepsize and \( \varepsilon(i) \) is the regularization factor.

Approximate \( R_u \) by \( \hat{R}_u = \frac{1}{i+1} \sum_{j=0}^{i} \lambda^{i-j} u_j^* u_j \), i.e. by an exponential average of previous regressors.

- If \( \lambda = 1 \) then all regressors have equal weight.
- If \( 0 \ll \lambda < 1 \) then recent regressors \((i-1, i-2, \ldots)\) are more relevant and remote regressors are forgotten.
- Generally \( \lambda \) is chosen so that \( 0 \ll \lambda < 1 \), therefore RLS has a memory or forgetting property.

Assume \( \mu(i) = \frac{1}{i+1} \) and \( \varepsilon(i) = \frac{\lambda^{i+1} \varepsilon}{i+1} \) for all \( i \). Then \( \varepsilon(i) \to 0 \) as \( i \to \infty \), i.e. as time increases the regularization factor disappears.
Recursive Least Square (RLS) Filters

- Development using the instantaneous approximation
  \[\omega_i = \omega_{i-1} + \left[\lambda^{i+1}\varepsilon l + \sum_{j=0}^{i} \lambda^{i-j} u_j^* u_j\right]^{-1} u_i^* [d(i) - u_i \omega_{i-1}]\]

- Define
  \[\Phi_i = \lambda^{i+1}\varepsilon l + \sum_{j=0}^{i} \lambda^{i-j} u_j^* u_j\]
  then
  \[\Phi_i = \lambda \Phi_{i-1} + u_i^* u_i, \quad \Phi_{-1} = \varepsilon l\]

- The matrix inversion formula for \(P_i = \Phi_i^{-1}\) is given by
  \[P_i = \lambda^{-1} \left[P_{i-1} - \frac{\lambda^{-1} P_{i-1} u_i^* u_i P_{i-1}}{1 + \lambda^{-1} u_i P_{i-1} u_i^*}\right], \quad P_{-1} = \varepsilon^{-1} l\]

- With the simplification we obtain the RLS algorithm
  \[\omega_i = \omega_{i-1} + P_i u_i^* [d(i) - u_i \omega_{i-1}], \quad i = 0, 1, 2, \ldots\]
Least-Squares Problem

- Replace $E \left[ |d - u\omega|^2 \right]$ by $\frac{1}{N} \sum_{i=0}^{N-1} |d - u\omega|^2$, then problem (1) is modified to
  \[
  \min_{\omega} \sum_{i=0}^{N-1} |d(i) - u_i\omega|^2 = \min_{\omega} \|y - H\omega\|^2
  \]  
  (2)

  where

  \[
  y = \begin{bmatrix} d(0) & d(1) & \cdots & d(N-1) \end{bmatrix}
  \]  
  and

  \[
  H = \begin{bmatrix} u_0^T & u_1^T & \cdots & u_{N-1}^T \end{bmatrix}^T
  \]

- Weighted Least-Squares
  - Let $W$ be a weights matrix, then (2) can be modified to
    \[
    \min_{\omega} (y - H\omega)^* W (y - H\omega).
    \]

- Regularized Least-Squares
  - Let $\Pi > 0$ be a regularization matrix, then (2) can be modified to
    \[
    \min_{\omega} \left[ \omega^*\Pi\omega + \|y - H\omega\|^2 \right].
    \]
Not Presented

- Weighted, Regularized and Weighted and Regularized Least-Square Algorithms
- Array Methods for Adaptive Filters
- Given’s Rotation
- CORDIC Cells
- QR-Recursive Least Square Algorithm
References

- Dr. Rafia’s Notes for ECE 635
- Adaptive Filters by Ali H. Sayed