Recursive Least Squares Algorithms

Introduction

Up to now we have considered gradient descent algorithms for minimizing our mean-square error or MSE (cost function)

$$J(n) = E[|e(n)|^2] .$$

The difficulty with these methods (steepest descent) is that they all require knowledge of

$$\begin{aligned} \mathbf{R} &= E[\mathbf{u}(n)\mathbf{u}^H(n)] \\ \boldsymbol{\pi} &= E[\mathbf{u}(n)d^*(n)] \end{aligned} .$$

When this statistical information is unknown, we are forced to estimate these statistics from the data. In the case of LMS, we use instantaneous estimates of the ensemble averages

$$\begin{aligned} \hat{\mathbf{R}}(n) &= \mathbf{u}(n)\mathbf{u}^H(n) \\ \hat{\boldsymbol{\pi}}(n) &= \mathbf{u}(n)d^*(n) \end{aligned} .$$

For many applications this *statistical approach* is adequate, in others this gradient estimate may not provide a sufficiently rapid rate of convergence or sufficiently small J_{ex}.

As an alternative we may consider the least squares (LS) error cost function which does not include expectations

$$\begin{aligned} J(n) &= \varepsilon(n) \\ &= \sum_{i=1}^{n} |e(i)|^2 . \end{aligned}$$

Note that this cost function requires no statistical information about $u(n)$ or $d(n)$ and is evaluated directly from the data itself over the observation interval $1 \le i \le n$. The method of LS is a *deterministic approach*.

Philosophy

There is an important philosophical difference between minimizing the LS error and the MSE.

Minimizing the MSE (when using exact **R** and **p**) produces the *same* set of filter coefficients for *all* sequences having the *same* statistics. Therefore the coefficients do not depend on the incoming data only on their ensemble averages. As a result the filter coefficients will be *statistically optimal over a particular class of process*.

With LS error we are minimizing a squared error that *depends* explicitly *on* $u(n)$ and $d(n)$. Consequently for *different signals* we get *different filters* even if the statistics of the sequences are the same. As a result the filter coefficients will be *optimal for the given data*.

Some Preliminaries

In the method of exponentially-weighted LS we minimize the cost function

$$\varepsilon(n) = \sum_{i=1}^{n} \lambda^{n-i} |e(i)|^2$$

where $0 < \lambda < 1$ is called the forgetting factor since information from the distant past has an increasingly negligible effect on the coefficient updating.

Problem. Find $\hat{\mathbf{w}}(n)$ which minimizes

where

$$e(i) = \delta(i) - \hat{\boldsymbol{\omega}}^H(n)\boldsymbol{\upsilon}(i).$$

Solution. We set the gradient of $\varepsilon(n)$ w.r.t. the coefficient vector $\hat{\mathbf{w}}(n)$ equal to zero as we did earlier when we solved the statistical filtering problem. We can compute the gradient as

$$\nabla\varepsilon(n) = -2\sum_{i=1}^{n}\lambda^{n-i}\mathbf{u}(i)[d^*(i) - \mathbf{u}^H(i)\hat{\mathbf{w}}(n)].$$

Then

$$\nabla\varepsilon(n) = \mathbf{0}$$
$$\Leftrightarrow$$
$$\sum_{i=1}^{n}\lambda^{n-i}\mathbf{u}(i)d^*(i) - \sum_{i=1}^{n}\lambda^{n-i}\mathbf{u}(i)\mathbf{u}^H(i)\hat{\mathbf{w}}(n) = \mathbf{0}.$$
$$\Leftrightarrow$$
$$\sum_{i=1}^{n}\lambda^{n-i}\mathbf{u}(i)\mathbf{u}^H(i)\hat{\mathbf{w}}(n) = \sum_{i=1}^{n}\lambda^{n-i}\mathbf{u}(i)d^*(i)$$

The above defines the *deterministic normal equations*:

$$\boxed{\boldsymbol{\Phi}(n)\hat{\mathbf{w}}(n) = \mathbf{z}(n)}$$

where

$$\boxed{\boldsymbol{\Phi}(n) = \sum_{i=1}^{n}\lambda^{n-i}\mathbf{u}(i)\mathbf{u}^H(i)}$$

and

$$\boxed{\mathbf{z}(n) = \sum_{i=1}^{n}\lambda^{n-i}\boldsymbol{\upsilon}(i)d^*(i)}.$$

We note that an alternate solution would employ the orthogonality principle.

Instead of solving the normal equations directly for each n (deterministic equivalent to Wiener solution), we want to derive a recursive solution of the form

$$\hat{\mathbf{w}}(n) = \hat{\boldsymbol{\omega}}(n-1) + \underbrace{\Delta\hat{\boldsymbol{\omega}}(n-1)}_{\text{χορρεχτιον}}.$$

Since

$$\boxed{\hat{\mathbf{w}}(n) = \boldsymbol{\Phi}^{-1}(n)\boldsymbol{\zeta}(n)}$$

we would need our recursive relation to evaluate $\boldsymbol{\Phi}^{-1}(n)$ in terms of $\boldsymbol{\Phi}^{-1}(n-1)$ and $\mathbf{z}(n)$ in terms of $\mathbf{z}(n-1)$.

We first observe the following

$$\boxed{\begin{aligned}\boldsymbol{\Phi}(n) &= \lambda\left[\sum_{i=1}^{n-1}\lambda^{n-1-i}\mathbf{u}(i)\mathbf{u}^H(i)\right] + \mathbf{u}(n)\mathbf{u}^H(n)\\ &= \lambda\boldsymbol{\Phi}(n-1) + \mathbf{u}(n)\mathbf{u}^H(n)\end{aligned}}$$

and

$$\boxed{\begin{aligned}\mathbf{z}(n) &= \lambda\left[\sum_{i=1}^{\nu-1}\lambda^{\nu-1-i}\mathbf{u}(i)\,d^{*}(i)\right]+\mathbf{u}(\nu)\,d^{*}(\nu)\\ &= \lambda\boldsymbol{\zeta}(\nu-1)+\mathbf{u}(\nu)\,d^{*}(\nu)\end{aligned}}.$$

We next employ Woodbury's identity to the recursive evaluation of $\boldsymbol{\Phi}^{-1}(n)$.