Nonlinear Signal Processing
ELEG 833

Gonzalo R. Arce

Department of Electrical and Computer Engineering
University of Delaware
arce@ee.udel.edu

Fall 2008

Outline

Appendix: Linear Filter theory

Linear filters compute their output as a linear combination of observation samples:

$$Y(n) = \sum_{i=1}^{N} W_i X(n - i + 1)$$

where W_i are the filter coefficients. The impulse response of the filter is

$$h(n) = W_{n+1} \qquad n = 0, 1, \cdots, N - 1.$$

The output $Y(n)$ is the estimate of some desired signal $D(n)$ and is denoted as $\hat{D}(n)$. The set of N coefficients $W_1, W_2, \cdots, W_N$ define the column vector $\mathbf{W}$

$$\mathbf{W} = [W_1, W_2, \ldots, W_N]^T.$$

Similarly, the vector $\mathbf{X}(n)$ is defined as the N long observation vector

$$\begin{aligned} \mathbf{X}(n) &= [X(n),\ X(n-1), \ldots,\ X(n-N+1)]^T \\ &= [X_1(n),\ X_2(n), \ldots,\ X_N(n)]^T. \end{aligned} \tag{1}$$

The estimate of $D(n)$ is

$$\hat{D}(n) = \mathbf{W}^T \mathbf{X}(n),$$

and the estimation error is $e(n) = D(n) - \hat{D}(n)$.

Wiener Filter

The Wiener filter theory is based on the minimization of the mean squared error criterion

$$J_2(\mathbf{W}) = E\{e^2(n)\}.$$

The goal is to obtain the weight vector that minimizes $J_2(\mathbf{W})$. The optimal vector $\mathbf{W}$ which minimizes $J_2(\mathbf{W})$ is the Wiener filter.

The error is $e(n) = D(n) - \mathbf{W}^T\mathbf{X}(n)$; thus, $J_2(\mathbf{W})$ is

$$\begin{aligned} J_2(\mathbf{W}) &= E[(D(n) - \mathbf{W}^T\mathbf{X}(n))(D^T(n) - \mathbf{X}^T(n)\mathbf{W})] \\ &= E[D^2(n) - 2D(n)\mathbf{X}^T(n)\mathbf{W} + \mathbf{W}^T\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W}]. \end{aligned} \tag{2}$$

Assuming that the underlying random processes are stationary and that the desired signal is zero mean, the first expectation is σ_d^2. The second expectation is

$$\mathbf{p} = E\{D(n)\mathbf{X}(n)\} = E\left\{ D(n) \begin{bmatrix} X(n) \\ X(n-1) \\ \vdots \\ X(n-N+1) \end{bmatrix} \right\}. \tag{3}$$

The terms $E[D(n)\ X(n-m)]$ are the cross correlation coefficients, $r_{DX}(m)$.

The last expectation in (2) is the autocorrelation matrix

$$\mathbf{R}_x = E[\mathbf{X}(n)\mathbf{X}^T(n)] \tag{4}$$

$$= \begin{bmatrix} r_X(0) & r_X(1) & \cdots & r_X(N-1) \\ r_X(1) & r_X(0) & \cdots & r_X(N-2) \\ \vdots & & \ddots & \\ r_X(N-1) & r_X(N-2) & \cdots & r_X(0) \end{bmatrix} \tag{5}$$

where $r_X(m) = E[X(n)X(n-m)]$. Substituting the expectations in (2), we find the error function

$$J_2(\mathbf{W}) = \sigma_d^2 - 2\mathbf{p}^T\mathbf{W} + \mathbf{W}^T\mathbf{R}_x\mathbf{W} \tag{6}$$

which is a quadratic on $\mathbf{W}$; thus for different values in $W_1, W_2, \cdots, W_N$, $J_2(\mathbf{W})$ defines a bowl-shape surface in an $N+1$ dimensional space which has a unique minimum.

The gradient vector ∇ is needed to search for the bottom of the bowl where

$$\begin{aligned}\nabla &= \frac{\partial(J_2(\mathbf{W}))}{\partial \mathbf{W}} \\ &= \left[\frac{\partial(J_2(\mathbf{W}))}{\partial W_1}, \frac{\partial(J_2(\mathbf{W}))}{\partial W_2}, \ldots, \frac{\partial(J_2(\mathbf{W}))}{\partial W_N}\right]^T \\ &= 2\mathbf{R}_x\mathbf{W} - 2\mathbf{p}.\end{aligned}$$

Setting the gradient to zero, yields

$$\mathbf{R}_X\mathbf{W}_0 = \mathbf{p}. \tag{7}$$

This is known as the *normal equation*. The optimal linear (Wiener) filter is

$$\mathbf{W}_0 = \mathbf{R}_X^{-1}\mathbf{p}, \tag{8}$$

which only requires the knowledge of second order statistics of the underlying processes.

The estimate is "normal" or at "right angles" with respect to the observations. The orthogonality property of the error is noticed in the normal equation as $\mathbf{p}_x - \mathbf{R}_X\mathbf{W}_0 = \mathbf{0}$. This equation is

$$\begin{aligned} \mathbf{0} &= E\{\mathbf{X}(n)(D(n) - \mathbf{X}^T(n)\mathbf{W}_0)\} \\ &= E\{\mathbf{X}(n)e_0(n)\} \end{aligned}$$

implying that, $e_0(n)$, is orthogonal to each input sample that enters into the estimation

$$E\begin{bmatrix} X(n)e_0(n) \\ X(n-1)e_0(n) \\ \vdots \\ X(n-N+1)e_0(n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{9}$$

Hence, the term *normal* describes the set of N conditions in (9).

Once the optimal linear estimate is obtained as

$$\hat{D}(n) = \mathbf{W}_0^T \mathbf{X}(n)$$

where $\mathbf{W}_0$ is given in (8), the mean square error is obtained as

$$\begin{aligned} J_2(\mathbf{W}_0) &= \sigma_d^2 - 2\mathbf{p}^T\mathbf{W}_0 + \mathbf{W}_0^T\mathbf{R}_X\mathbf{W}_0 && (10)\\ &= \sigma_d^2 - \mathbf{p}^T\mathbf{W}_0 && (11)\\ &= \sigma_d^2 - \mathbf{p}^T\mathbf{R}_X^{-1}\mathbf{p}, && (12) \end{aligned}$$

which is the minimum point of the bowl error sample or $J_{\min}$.

A method to describe the error surface is to determine its principal axes. Rewrite (12) as

$$J_2(\mathbf{W}_0) - J_{min} = (\mathbf{W} - \mathbf{W}_0)^T \mathbf{R}_X (\mathbf{W} - \mathbf{W}_0)^T$$

When $\mathbf{W}$ is the Wiener solution, the obtained MSE is the minimum, but when this optimal weight vector is perturbed the MSE increases.

A method to display the shape of the MSE surface is the *contour*. Since the error surface is quadratic, these contours are ellipsoids which aid in the visualization of the MSE surface.

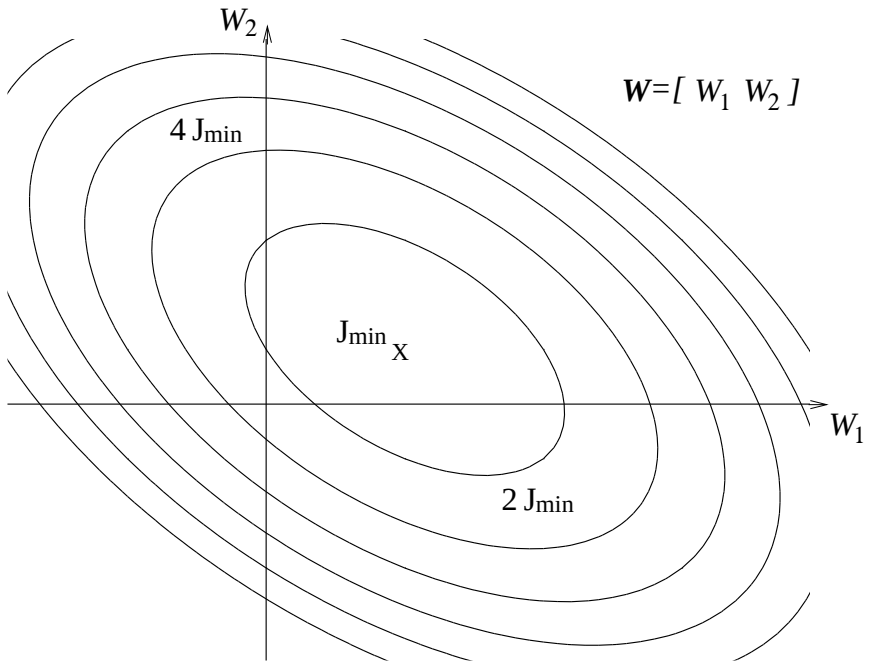

Figure: Contours of an MSE surface

A similarity transformation of $\mathbf{R}$ exists such that

$$\mathbf{R}_X = \mathbf{Q}\Lambda\mathbf{Q}^T \tag{13}$$

where Λ is the diagonal matrix of eigenvalues λ_i of the matrix $\mathbf{R}_X$. Applying the similarity transformation we obtain

$$J(\mathbf{W}) = J_{\min} + (\mathbf{W} - \mathbf{W}_0)^T\mathbf{Q}\Lambda\mathbf{Q}^T(\mathbf{W} - \mathbf{W}_0)$$

Defining the vector coordinate ν as

$$\nu = \mathbf{Q}^T(\mathbf{W} - \mathbf{W}_0).$$

the quadratic error function becomes

$$\begin{aligned} J &= J_{\min} + \nu^T\mathbf{\Lambda}\nu \\ &= J_{\min} + \sum_{i=1}^{N}\lambda_i\nu_i^2. \end{aligned}$$

This representation shows that the components of ν constitute the principal axis of the error-performance surface.

Method of Steepest Descent

The steepest descent algorithm solves the normal equations for $\mathbf{W}_0$. Given a set of weights $\mathbf{W}$, the linear estimate $\hat{D}(n) = \mathbf{W}^T\mathbf{X}(n)$ leads to

$$J = \sigma_d^2 - 2\mathbf{W}^T\mathbf{p} + \mathbf{W}^T\mathbf{R}\mathbf{W}$$

which is a quadratic surface in $N+1$ dimensions.
The steepest descent method starts by assigning an initial value $\mathbf{W}(0)$.
Next it corrects the initial guess by moving the weight values towards the direction of the negative of the gradient vector.
This successive procedure leads the bottom of the error surface.

Denoting the gradient vector at time n as $\nabla(J(n))$, the value of the weight vector at time $n+1$ is then computed as

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \frac{1}{2}\mu[-\nabla(j(n))] \tag{14}$$

where μ is a positive real-valued constant. The gradient vector is given by

$$\begin{aligned} \nabla(J(n)) &= \left[\frac{\partial J(n)}{\partial W_1}, \frac{\partial J(n)}{\partial W_2}, \cdots, \frac{\partial J(n)}{\partial W_N}\right]^T \\ &= -2\mathbf{p} + 2\mathbf{R}\mathbf{W}(n). \end{aligned} \tag{15}$$

Replacing (15) in (14) leads to the steepest descent recursion

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{W}(n)] \tag{16}$$

for $n \geq 0$, and where μ is a parameter that controls the incremental update.

The steepest descent algorithm is recursive. It is natural to establish conditions for covergence and stability. Begin by defining the weight error vector at time n as

$$\mathbf{c}(n) = \mathbf{W}(n) - \mathbf{W}_0$$

where $\mathbf{W}_0$ is the optimal solution. Replacing $\mathbf{p}$ by $\mathbf{R}\mathbf{W}_0$ in the steepest descent recursion leads to

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu\mathbf{R}(-\mathbf{c}(n)) \tag{17}$$

which can be written as the matrix difference equation

$$\begin{aligned} \mathbf{c}(n+1) &= \mathbf{c}(n) - \mu\mathbf{R}\mathbf{c}(n) \\ &= (\mathbf{I} - \mu\mathbf{R})\mathbf{c}(n). \end{aligned} \tag{18}$$

Using $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H$, where Λ is a diagonal matrix with the eigenvalues of $\mathbf{R}$ as the diagonal elements, we obtain

$$\mathbf{c}(n+1) = \mathbf{c}(n) - \mu\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H\mathbf{c}(n). \tag{19}$$

Premultiplying the above by $\mathbf{Q}^H$

$$\mathbf{Q}^H\mathbf{c}(n+1) = \mathbf{Q}^H\mathbf{c}(n) - \mu\mathbf{Q}^H\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H\mathbf{c}(n). \tag{20}$$

Since $\mathbf{Q}$ is a unitary matrix satisfying $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$, and if we define $\mathbf{v}(n) = \mathbf{Q}^H\mathbf{c}(n)$, then (20) reduces to

$$\mathbf{v}(n+1) = (\mathbf{I} - \mu\mathbf{\Lambda})\mathbf{v}(n), \tag{21}$$

whose initial value equals $\mathbf{v}(0) = -\mathbf{Q}^H\mathbf{W}_0$ if we assume that the initial weight vector $\mathbf{W}(0)$ is zero. This recursion is fully decoupled where the $k'th$ elements of $\mathbf{v}(n)$, $v_k(n)$ satisfies

$$v_k(n+1) = (1 - \mu\lambda_k)v_k(n) \qquad k = 1, 2, \cdots, N \tag{22}$$

where λ_k is the $k'th$ eigenvalue of $\mathbf{R}$.

The solution follows easily as

$$v_k(n) = (1 - \mu\lambda_k)^n v_k(0) \tag{23}$$

which converges if $| 1 - \mu\lambda_k | < 1$. Since the eigenvalues of $\mathbf{R}$ are all real and postive, all modes will converge if

$$0 < \mu < \frac{2}{\lambda_{max}}, \tag{24}$$

where λ_{max} is the largest eigenvalue of $\mathbf{R}$. From the definition,

$$\mathbf{v}(n) = \mathbf{Q}^H[\mathbf{W}(n) - \mathbf{W}_0]. \tag{25}$$

Premultiplying the above by $\mathbf{Q}$ leads to

$$\mathbf{W}(n) = \mathbf{W}_0 + \mathbf{Q}\mathbf{v}(n). \tag{26}$$

Since each of the elements of $\mathbf{v}(n)$ decays geometrically to zero, the steepest descent algorithm converges to the optimal solution $\mathbf{W}_0$.

Outline

Least-Mean-Square (LMS) Algorithm

If the gradient vector $\nabla(J(n))$ could be exactly measured after each iteration, then the system would converge to the optimum Wiener solution. The gradient of the error surface was shown before to be

$$\nabla\left(J(n)\right) = -2\mathbf{p}_x + \ 2\mathbf{R}_x\mathbf{W}(n). \tag{27}$$

In order to estimate the gradient we must first obtain the estimates for $\mathbf{p}_x$ and $\mathbf{R}_x$. The simplest choice is used by the (LMS) algorithm.

$$\hat{\mathbf{R}}_X(n) = \mathbf{X}(n)\mathbf{X}^T(n) \tag{28}$$

and

$$\hat{\mathbf{p}}_X(n) = \mathbf{X}(n)D(n). \tag{29}$$

The instantaneous estimate of the gradient is obtained as

$$\hat{\nabla}\left(J(n)\right) = -2\mathbf{X}(n)\ D(n) + \ 2\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W}(n), \tag{30}$$

Using this estimate we obtain the weight vector recursion

$$\begin{aligned} \mathbf{W}(n+1) &= \mathbf{W}(n) + \mu\mathbf{X}(n)\left[\ D(n) - \ \mathbf{X}^T(n)\mathbf{W}(n)\right] \\ &= \mathbf{W}(n) + \mu\mathbf{X}(n)\ e(n), \end{aligned} \tag{31}$$

where $e(n)$ is the estimation error. The term $\mu\mathbf{X}(n)e(n)$ in (31) is the correction applied to the the weight vector on each iteration.

Convergence in the mean

The goal is to find conditions on the step size such that the expected valued of the weight vector $E\{\hat{\mathbf{W}}(n)\}$ converges to the optimum Wiener solution $\mathbf{W}_o$ as the number of iterations approaches infinity. The LMS weight recursion is

$$\begin{aligned}\mathbf{W}(n+1) &= \mathbf{W}(n) + \mu\mathbf{X}(n)e(n) \\ &= \mathbf{W}(n) + \mu\mathbf{X}(n)\left[D(n) - \mathbf{X}^T(n)\mathbf{W}(n)\right] \end{aligned} \tag{32}$$

Defining $\mathbf{c}(n+1) = \mathbf{W}_0 - \mathbf{W}(n+1)$ where $\mathbf{W}_0$ is the Wiener filter solution, we can rewrite (32) as

$$\begin{aligned}\mathbf{c}(n+1) &= \mathbf{c}(n) - \mu\mathbf{X}(n)\left[D(n) - \mathbf{X}^T(n)(\mathbf{W}_0 - \mathbf{c}(n))\right] \\ &= \left(\mathbf{I} - \mu\mathbf{X}(n)\mathbf{X}^T(n)\right)\mathbf{c}(n) + \mu\mathbf{X}(n)\left[D(n) - \mathbf{X}^T(n)\right] \\ &= \left(\mathbf{I} - \mu\mathbf{X}(n)\mathbf{X}^T(n)\right)\mathbf{c}(n) + \mu\mathbf{X}(n)e_0(n), \end{aligned} \tag{33}$$

where $\mathbf{c}_0(n) = D(n) - \mathbf{W}_0^T\mathbf{X}(n)$. Taking the expection of the above:

$$E\{\mathbf{c}(n+1)\} = E\{\left(\mathbf{I} - \mu\mathbf{X}(n)\mathbf{X}^T(n)\right)\mathbf{c}(n)\} + E\left\{\mu\mathbf{X}(n)e_0(n)\right\} \tag{34}$$

From the orthogonality property, the second term in the right side of (34) vanishes. The independence asumptions are used to simplify the remaining terms of (34). In particular if the weights $\mathbf{c}(n)$ are assumed independent of the vector $\mathbf{X}(\mathbf{n})$, then

$$E\{\mathbf{c}(n+1)\} = E\{\left(\mathbf{I} - \mu\mathbf{X}(n)\mathbf{X}^T(n)\right)\}E\{\mathbf{c}(n)\} \quad (35)$$

$$= \left(\mathbf{I} - \mu\mathbf{R}\right)E\{\mathbf{c}(n)\}. \quad (36)$$

This equation is of the same form as the steepest descent algorithm, which was shown to converge to the null vector if the step size μ satisfies.

$$o < \mu < \frac{2}{\lambda_{MAX}}$$

where λ_{MAX} is the maximum eigenvalue of $\mathbf{R}$.

Since $E\{\mathbf{c}(n+1)\}$ converges to $\underline{\mathbf{0}}$, and since $\mathbf{c}(n) = \mathbf{W}(n) + \mathbf{W}_0$, then the LMS equation converges as

$$\lim_{n\to\infty} E\{\mathbf{W}(n)\} = \mathbf{W}_0.$$

Estimating the largest eigenvalue is not simple. On the other hand, the upper bound

$$\lambda_{MAX} \leq trace[\mathbf{R}] = N\sigma^2$$

is easily found which leads to the more conservative bound for μ required for the LMS convergence on the mean

$$o < \mu < \frac{2}{N\sigma^2}.$$

The LMS algorithm produces a mean-squared error $J(n)$ that is always in excess of the minimum mean squared error J_{min} attained with the Wiener solution. The excess mean-squared error is defined as:

$$J_{ex}(n) = J(n) - J_{min}. \tag{37}$$

The dynamic behavior of the mean-square error $J(n)$ as the weight coefficients are adapted from the initial guess $\mathbf{W}(0)$ to the weight vector attained in steady state, is referred to as the learning curve.

The *misadjustment* is defined for this purpose as the ratio of the excess mean-squared error $J_{ex}(\infty)$ to the minimum mean-squared error J_{min} giving

$$\mathcal{M} = \frac{J_{ex}(\infty)}{J_{min}} \tag{38}$$

$$\tag{39}$$

When the step size μ is small compared to the upper bound $2/\lambda_{max}$ it has been shown that

$$\mathcal{M} = \frac{\mu}{2}\sum_{i=1}^{N}\lambda_i \tag{40}$$

$$= \frac{\mu}{2} Trace[\mathbf{R}_x]. \tag{41}$$