FSAN/ELEG815: Statistical Learning

Gonzalo R. Arce

Department of Electrical and Computer Engineering
University of Delaware

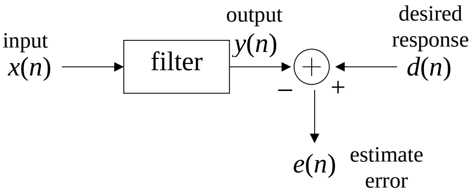
VI: The Wiener Filter

Outline of the Course

1. Review of Probability
2. Stationary processes
3. Eigen Analysis, Singular Value Decomposition (SVD) and Principal Component Analysis (PCA)
4. The Learning Problem
5. Training vs Testing
6. The Wiener Filter
7. Adaptive Optimization: Steepest descent and the LMS algorithm
8. Overfitting and Regularization
9. Logistic, Ridge and Lasso regression.
10. Neural Networks
11. Matrix Completion

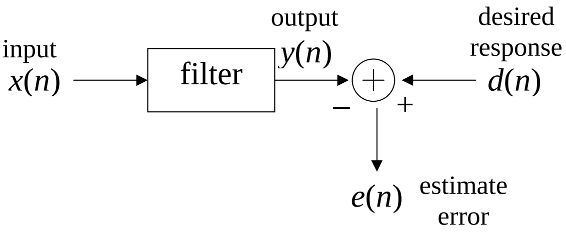
Problem Statement

Produce an estimate of a desired process statistically related to a set of observations



Historical Notes: The linear filtering problem was solved by

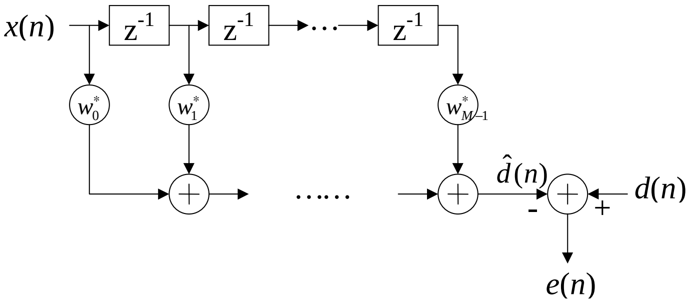
- Andrey Kolmogorov for discrete time – his 1938 paper “established the basic theorems for smoothing and predicting stationary stochastic processes”
- Norbert Wiener in 1941 for continuous time – not published until the 1949 paper *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*



System restrictions and considerations:

- Filter is linear
- Filter is discrete time
- Filter is finite impulse response (FIR)
- The process is WSS
- Statistical optimization is employed

For the discrete time case



- The filter impulse response is finite and given by

$$h_k = \begin{cases} w_k^* & \text{for } k = 0, 1, \cdots, M-1 \\ 0 & \text{otherwise} \end{cases}$$

- The output $\hat{d}(n)$ is an estimate of the desired signal $d(n)$
  - $x(n)$ and $d(n)$ are statistically related $\Rightarrow$ $\hat{d}(n)$ and $d(n)$ are statistically related

In convolution and vector form

$$\hat{d}(n) = \sum_{k=0}^{M-1} w_k^* x(n-k) = \mathbf{w}^H \mathbf{x}(n)$$

where

$$\begin{aligned} \mathbf{w} &= [w_0, w_1, \cdots, w_{M-1}]^T && \text{[filter coefficient vector]} \\ \mathbf{x} &= [x(n), x(n-1), \cdots, x(n-M+1)]^T && \text{[observation vector]} \end{aligned}$$

The error can now be written as

$$e(n) = d(n) - \hat{d}(n) = d(n) - \mathbf{w}^H \mathbf{x}(n)$$

Question: Under what criteria should the error be minimized?

Selected Criteria: Mean squared-error (MSE)

$$J(\mathbf{w}) = E\{e(n)e^*(n)\} \qquad (*)$$

Result: The $\mathbf{w}$ that minimizes $J(\mathbf{w})$ is the optimal (Wiener) filter

Utilizing $e(n) = d(n) - \mathbf{w}^H\mathbf{x}(n)$ in $(*)$ and expanding,

$$\begin{aligned} J(\mathbf{w}) &= E\{e(n)e^*(n)\} \\ &= E\{(d(n) - \mathbf{w}^H\mathbf{x}(n))(d^*(n) - \mathbf{x}^H(n)\mathbf{w})\} \\ &= E\{|d(n)|^2 - d(n)\mathbf{x}^H(n)\mathbf{w} - \mathbf{w}^H\mathbf{x}(n)d^*(n) \\ &\quad +\mathbf{w}^H\mathbf{x}(n)\mathbf{x}^H(n)\mathbf{w}\} \\ &= E\{|d(n)|^2\} - E\{d(n)\mathbf{x}^H(n)\}\mathbf{w} - \mathbf{w}^H E\{\mathbf{x}(n)d^*(n)\} \\ &\quad +\mathbf{w}^H E\{\mathbf{x}(n)\mathbf{x}^H(n)\}\mathbf{w} \qquad (**) \end{aligned}$$

Let

$$\begin{aligned} \mathbf{R} &= E\{\mathbf{x}(n)\mathbf{x}^H(n)\} && [\text{autocorrelation of } \mathbf{x}(n)] \\ \mathbf{p} &= E\{\mathbf{x}(n)d^*(n)\} && [\text{cross correlation between } \mathbf{x}(n) \text{ and } d(n)] \end{aligned}$$

Then $(**)$ can be compactly expressed as

$$J(\mathbf{w}) = \sigma_d^2 - \mathbf{p}^H\mathbf{w} - \mathbf{w}^H\mathbf{p} + \mathbf{w}^H\mathbf{R}\mathbf{w}$$

where we have assumed $x(n)$ & $d(n)$ are zero mean, WSS

The MSE criteria as a function of the filter weight vector $\mathbf{w}$

$$J(\mathbf{w}) = \sigma_d^2 - \mathbf{p}^H\mathbf{w} - \mathbf{w}^H\mathbf{p} + \mathbf{w}^H\mathbf{R}\mathbf{w}$$

Observation: The error is a quadratic function of $\mathbf{w}$

Consequences: The error is an M–dimensional bowl–shaped function of $\mathbf{w}$ with a unique minimum

Result: The optimal weight vector, $\mathbf{w}_0$, is determined by differentiating $J(\mathbf{w})$ and setting the result to zero

$$\nabla_{\mathbf{w}} J(\mathbf{w})\big|_{\mathbf{w}=\mathbf{w}_0} = 0$$

- A closed form solution exists

Example

Consider a two dimensional case, i.e., a $M = 2$ tap filter. Plot the error surface and error contours.

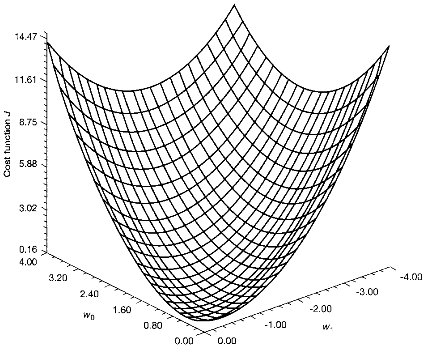


Figure 5.6 Error-performance surface of the two-tap transversal filter described in the numerical example.

Error Surface

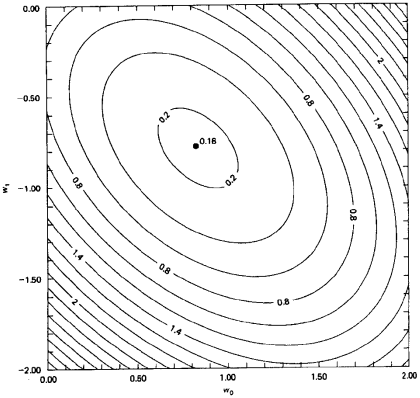


Figure 5.7 Contour plots of the error-performance surface depicted in Fig. 5.6.

Error Contours

Aside (Matrix Differentiation): For complex data,

$$w_k = a_k + jb_k, \qquad k = 0, 1, \cdots, M-1$$

the gradient, with respect to w_k, is

$$\nabla_k(J) = \frac{\partial J}{\partial a_k} + j\frac{\partial J}{\partial b_k}, \qquad k = 0, 1, \cdots, M-1$$

The complete gradient is thus given by

$$\nabla_{\mathbf{w}}(J) = \begin{bmatrix} \nabla_0(J) \\ \nabla_1(J) \\ \vdots \\ \nabla_{M-1}(J) \end{bmatrix} = \begin{bmatrix} \frac{\partial J}{\partial a_0} + j\frac{\partial J}{\partial b_0} \\ \frac{\partial J}{\partial a_1} + j\frac{\partial J}{\partial b_1} \\ \vdots \\ \frac{\partial J}{\partial a_{M-1}} + j\frac{\partial J}{\partial b_{M-1}} \end{bmatrix}$$

Example

Let $\mathbf{c}$ and $\mathbf{w}$ be $M \times 1$ complex vectors.
For $g = \mathbf{c}^H\mathbf{w}$, find $\nabla_{\mathbf{w}}(g)$

Note

$$g = \mathbf{c}^H\mathbf{w} = \sum_{k=0}^{M-1} c_k^* w_k = \sum_{k=0}^{M-1} c_k^*(a_k + jb_k)$$

Thus

$$\begin{aligned} \nabla_k(g) &= \frac{\partial g}{\partial a_k} + j\frac{\partial g}{\partial b_k} \\ &= c_k^* + j(jc_k^*) = 0, \qquad k = 0, 1, \cdots, M-1 \end{aligned}$$

Result: For $g = \mathbf{c}^H\mathbf{w}$

$$\nabla_{\mathbf{w}}(g) = \begin{bmatrix} \nabla_0(g) \\ \nabla_1(g) \\ \vdots \\ \nabla_{\text{M-1}}(g) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{0}$$

Example

Now suppose $g = \mathbf{w}^H \mathbf{c}$.

Find $\nabla_{\mathbf{w}}(g)$

In this case,

$$g = \mathbf{w}^H \mathbf{c} = \sum_{k=0}^{M-1} w_k^* c_k = \sum_{k=0}^{M-1} c_k(a_k - jb_k)$$

and

$$\begin{aligned}\nabla_k(g) &= \frac{\partial g}{\partial a_k} + j\frac{\partial g}{\partial b_k} \\ &= c_k + j(-jc_k) = 2c_k, \qquad k = 0, 1, \cdots, M-1\end{aligned}$$

Result: For $g = \mathbf{w}^H \mathbf{c}$

$$\nabla_{\mathbf{w}}(g) = \begin{bmatrix} \nabla_0(g) \\ \nabla_1(g) \\ \vdots \\ \nabla_{\text{M-1}}(g) \end{bmatrix} = \begin{bmatrix} 2c_0 \\ 2c_1 \\ \vdots \\ 2c_{M-1} \end{bmatrix} = 2\mathbf{c}$$

Example

Lastly, suppose $g = \mathbf{w}^H \mathbf{Q} \mathbf{w}$.

Find $\nabla_{\mathbf{w}}(g)$

In this case,

$$
\begin{aligned}
g &= \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} w_i^* w_j q_{i,j} \\
&= \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} (a_i - jb_i)(a_j + jb_j) q_{i,j} \\
\Rightarrow \nabla_k(g) &= \frac{\partial g}{\partial a_k} + j \frac{\partial g}{\partial b_k} \\
&= 2 \sum_{j=0}^{M-1} (a_j + jb_j) q_{k,j} + 0 \\
&= 2 \sum_{j=0}^{M-1} w_j q_{k,j}
\end{aligned}
$$

Result: For $g = \mathbf{w}^H \mathbf{Q} \mathbf{w}$

$$\nabla_{\mathbf{w}}(g) = \begin{bmatrix} \nabla_0(g) \\ \nabla_1(g) \\ \vdots \\ \nabla_{\text{M-1}}(g) \end{bmatrix} = 2 \begin{bmatrix} \sum_{i=0}^{M-1} q_{0,i} w_i \\ \sum_{i=0}^{M-1} q_{1,i} w_i \\ \vdots \\ \sum_{i=0}^{\text{M-1}} q_{M-1,i} w_i \end{bmatrix} = 2\mathbf{Q}\mathbf{w}$$

- Observation: Differentiation result depends on matrix ordering

Returning to the MSE performance criteria

$$J(\mathbf{w}) = \sigma_d^2 - \mathbf{p}^H\mathbf{w} - \mathbf{w}^H\mathbf{p} + \mathbf{w}^H\mathbf{R}\mathbf{w}$$

Approach: Minimize error by differentiating with respect to $\mathbf{w}$ and set result to 0

$$\begin{aligned} \nabla_{\mathbf{w}}(J) &= \mathbf{0} - \mathbf{0} - 2\mathbf{p} + 2\mathbf{R}\mathbf{w} \\ &= \mathbf{0} \\ \Rightarrow \mathbf{R}\mathbf{w}_0 &= \mathbf{p} \qquad \text{[normal equation]} \end{aligned}$$

Result: The Wiener filter coefficients are defined by

$$\mathbf{w}_0 = \mathbf{R}^{-1}\mathbf{p}$$

Question: Does $\mathbf{R}^{-1}$ always exist? Recall $\mathbf{R}$ is positive semi-definite, and usually positive definite

Orthogonality Principle

Consider again the normal equation that defines the optimal solution

$$\begin{aligned} \mathbf{R}\mathbf{w}_0 &= \mathbf{p} \\ \Rightarrow E\{\mathbf{x}(n)\mathbf{x}^H(n)\}\mathbf{w}_0 &= E\{\mathbf{x}(n)d^*(n)\} \end{aligned}$$

Rearranging

$$\begin{aligned} E\{\mathbf{x}(n)d^*(n)\} - E\{\mathbf{x}(n)\mathbf{x}^H(n)\}\mathbf{w}_0 &= \mathbf{0} \\ E\{\mathbf{x}(n)[d^*(n) - \mathbf{x}^H(n)\mathbf{w}_0]\} &= \mathbf{0} \\ E\{\mathbf{x}(n)e_0^*(n)\} &= \mathbf{0} \end{aligned}$$

Note: $e_0^*(n)$ is the error when the optimal weights are used, i.e.,

$$e_0^*(n) = d^*(n) - \mathbf{x}^H(n)\mathbf{w}_0$$

Thus

$$E\{\mathbf{x}(n)e_0^*(n)\} = E\begin{bmatrix} x(n)e_0^*(n) \\ x(n-1)e_0^*(n) \\ \vdots \\ x(n-M+1)e_0^*(n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Orthogonality Principle

A necessary and sufficient condition for a filter to be optimal is that the estimate error, $e^*(n)$, be orthogonal to each input sample in $\mathbf{x}(n)$

Interpretation: The observations samples and error are orthogonal and contain no mutual "information"

Objective: Determine the minimum MSE

Approach: Use the optimal weights $\mathbf{w}_0 = \mathbf{R}^{-1}\mathbf{p}$ in the MSE expression

$$\begin{aligned} J(\mathbf{w}) &= \sigma_d^2 - \mathbf{p}^H\mathbf{w} - \mathbf{w}^H\mathbf{p} + \mathbf{w}^H\mathbf{R}\mathbf{w} \\ \Rightarrow J_{\min} &= \sigma_d^2 - \mathbf{p}^H\mathbf{w}_0 - \mathbf{w}_0^H\mathbf{p} + \mathbf{w}_0^H\mathbf{R}(\mathbf{R}^{-1}\mathbf{p}) \\ &= \sigma_d^2 - \mathbf{p}^H\mathbf{w}_0 - \mathbf{w}_0^H\mathbf{p} + \mathbf{w}_0^H\mathbf{p} \\ &= \sigma_d^2 - \mathbf{p}^H\mathbf{w}_0 \end{aligned}$$

Result:

$$J_{\min} = \sigma_d^2 - \mathbf{p}^H\mathbf{R}^{-1}\mathbf{p}$$

where the substitution $\mathbf{w}_0 = \mathbf{R}^{-1}\mathbf{p}$ has been employed

Objective: Consider the excess MSE introduced by using a weighted vector that is not optimal.

$$J(\mathbf{w})-J_{\min}=(\sigma_d^2-\mathbf{p}^H\mathbf{w}-\mathbf{w}^H\mathbf{p}+\mathbf{w}^H\mathbf{R}\mathbf{w})-(\sigma_d^2-\mathbf{p}^H\mathbf{w}_0-\mathbf{w}_0^H\mathbf{p}+\mathbf{w}_0^H\mathbf{R}\mathbf{w}_0)$$

Using the fact that

$$\mathbf{p}=\mathbf{R}\mathbf{w}_0 \quad \text{and} \quad \mathbf{p}^H=\mathbf{w}_0^H\mathbf{R}$$

yields

$$\begin{aligned}
J(\mathbf{w})-J_{\min} &= -\mathbf{p}^H\mathbf{w}-\mathbf{w}^H\mathbf{p}+\mathbf{w}^H\mathbf{R}\mathbf{w}+\mathbf{p}^H\mathbf{w}_0+\mathbf{w}_0^H\mathbf{p}-\mathbf{w}_0^H\mathbf{R}\mathbf{w}_0\\
&= -\mathbf{w}_0^H\mathbf{R}\mathbf{w}-\mathbf{w}^H\mathbf{R}\mathbf{w}_0+\mathbf{w}^H\mathbf{R}\mathbf{w}+\mathbf{w}_0^H\mathbf{R}\mathbf{w}_0\\
&\quad +\mathbf{w}_0^H\mathbf{R}\mathbf{w}_0-\mathbf{w}_0^H\mathbf{R}\mathbf{w}_0\\
&= -\mathbf{w}_0^H\mathbf{R}\mathbf{w}-\mathbf{w}^H\mathbf{R}\mathbf{w}_0+\mathbf{w}^H\mathbf{R}\mathbf{w}+\mathbf{w}_0^H\mathbf{R}\mathbf{w}_0\\
&= (\mathbf{w}-\mathbf{w}_0)^H\mathbf{R}(\mathbf{w}-\mathbf{w}_0)\\
\Rightarrow J(\mathbf{w}) &= J_{\min}+(\mathbf{w}-\mathbf{w}_0)^H\mathbf{R}(\mathbf{w}-\mathbf{w}_0)
\end{aligned}$$

Finally, using the eigenvalue and vector representation $\mathbf{R} = \mathbf{Q}\mathbf{\Omega}\mathbf{Q}^H$

$$J(\mathbf{w}) = J_{\min} + (\mathbf{w} - \mathbf{w}_0)^H \mathbf{Q}\mathbf{\Omega}\mathbf{Q}^H (\mathbf{w} - \mathbf{w}_0)$$

or defining the eigenvector transformed difference

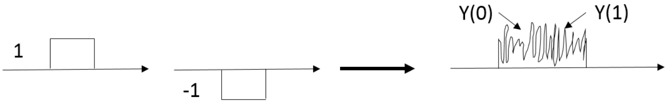
$$\begin{aligned} \mathbf{v} &= \mathbf{Q}^H(\mathbf{w} - \mathbf{w}_0) \qquad (*) \\ \Rightarrow J(\mathbf{w}) &= J_{\min} + \mathbf{v}^H \mathbf{\Omega} \mathbf{v} \\ &= J_{\min} + \sum_{k=1}^{M} \lambda_k v_k v_k^* \end{aligned}$$

Result:

$$J(\mathbf{w}) = J_{\min} + \sum_{k=1}^{M} \lambda_k |v_k|^2$$

Note: $(*)$ shows that v_k is the difference $(\mathbf{w} - \mathbf{w}_0)$ projected onto eigenvector $\mathbf{q}_k$

Example: Binary Phase-Shift Keying Symbol Estimate



Let x be a signal that is either -1 or 1 with probability $1/2$.
Collect two noisy measurements of the same value of x:

$$y(0) = x + v(0);$$

$$y(1) = x + v(1);$$

where $v(0)$ and $v(1)$ are independent zero-mean Gaussian with $\sigma_v^2 = 1$.
The optimal linear estimator of x given $\mathbf{y} = [y(0), y(1)]^T$ is

$$\hat{x} = \mathbf{w}^{\mathbf{H}}\mathbf{y}.$$

The autocorrelation matrix of $\mathbf{y}$ is

$$\mathbf{R}_y = \begin{bmatrix} E[y(0)^2] & E[y(0)y^*(1)] \\ E[y(1)y^*(0)] & E[y(1)^2] \end{bmatrix}.$$

Notice that x, $v(0)$ and $v(1)$ are independent, we get

$$E[y(0)^2] = E[x^2] + E[v(0)^2] = 1 + 1 = 2;$$

$$E[y(1)^2] = E[x^2] + E[v(1)^2] = 1 + 1 = 2;$$

$$E[y(0)y^*(1)] = E[(x + v(0))(x + v(1))^*] = E[x^2] = 1;$$

$$E[y(1)y^*(0)] = E[(x + v(1))(x + v(0))^*] = E[x^2] = 1.$$

So we have

$$\mathbf{R}_y = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

The cross-correlation vector of the desired value x and the measurements y is

$$\mathbf{P} = \begin{bmatrix} E[xy^*(0)] & E[xy^*(1)] \end{bmatrix}^H,$$

where

$$E[xy^*(0)] = E[x(x+v(0))] = E[x^2] = 1;$$
$$E[xy^*(1)] = E[x(x+v(1))] = E[x^2] = 1.$$

So we have

$$\mathbf{P} = \begin{bmatrix} 1 & 1 \end{bmatrix}^H,$$

The weights of the estimator are:

$$\mathbf{w} = \mathbf{R}_y^{-1}\mathbf{P} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \end{bmatrix}.$$

That is

$$\hat{x} = \frac{1}{3}(y(0) + y(1)).$$