Department of Electrical & Computer Engineering
University of California, Santa Barbara

ECE 245
Spring 2011
Shynk
H.O. #21

EXAMPLE FINAL EXAM

INSTRUCTIONS

This exam is open book and open notes. It consists of 4 problems and is worth a total of 160 points. The problems are not of equal difficulty so use discretion in allocating your time. Attempt to answer all questions in any order.

ECE 245
Spring 2011

1. LINEARLY CONSTRAINED FILTERING (40 points)

Suppose we want to minimize the mean-square error (MSE) subject to the constraint $\mathbf{a}^T\mathbf{w} = b$ where $e(n) = d(n) - \mathbf{w}^T\mathbf{x}(n)$, $\mathbf{w} = [w_1, \ldots, w_N]^T$ is the weight vector, $\mathbf{x}(n) = [x(n), \ldots, x(n-N+1)]^T$ is the input signal vector, $d(n)$ is the desired signal, $\mathbf{a}$ is a constraint vector, and b is a scalar.

(a) Show that the corresponding cost function can be written as

$$\xi^c = \xi_{\min} + \mathbf{v}^T\mathbf{R}\mathbf{v} + \lambda(\mathbf{a}^T\mathbf{v} - c) \tag{1}$$

where $\xi_{\min}$ is the minimum MSE for the *unconstrained* formulation, $\mathbf{v} = \mathbf{w} - \mathbf{w}_o$, $\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$ is the unconstrained Wiener weight vector, $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$, $\mathbf{p} = E[\mathbf{x}(n)d(n)]$, $c = b - \mathbf{a}^T\mathbf{w}_o$, and λ is a Lagrange multiplier.

(b) By minimizing (1) with respect to $\mathbf{v}$ and λ, show that the (translated) optimal weight vector is given by

$$\mathbf{v}_o^c = c\frac{\mathbf{R}^{-1}\mathbf{a}}{\mathbf{a}^T\mathbf{R}\mathbf{a}}. \tag{2}$$

Find an expression for the minimum MSE $\xi^c_{\min}$ and show that the constraint cannot result in a value lower than the unconstrained minimum MSE $\xi_{\min}$.

(c) Suppose now that we want to derive a least-mean-square (LMS) algorithm that minimizes an instantaneous estimate of the constrained MSE in (1). The algorithm can be written using the following two steps:

$$\mathbf{w}^c(n) = \mathbf{w}(n) + 2\mu\mathbf{x}(n)e(n) \tag{3}$$

$$\mathbf{w}(n+1) = \mathbf{w}^c(n) + \mathbf{u}(n) \tag{4}$$

where $\mathbf{w}^c(n)$ is an intermediate weight vector (i.e., before the constraint is applied), and $\mu > 0$ is the step-size parameter. By minimizing the energy $\mathbf{u}^T(n)\mathbf{u}(n)$ subject to the constraint $\mathbf{a}^T\mathbf{w}(n+1) = b$, show that

$$\mathbf{u}(n) = \frac{b - \mathbf{a}^T\mathbf{w}^c(n)}{\mathbf{a}^T\mathbf{a}}\mathbf{a}. \tag{5}$$

(d) By combining (3)-(5) and taking the expectation of both sides of the resulting update equation, we can write

$$E[\mathbf{w}(n+1)] = \mathbf{B}E[\mathbf{w}(n)] + \mathbf{d}. \tag{6}$$

Show that

$$\mathbf{B} = \mathbf{A}(\mathbf{I} - 2\mu\mathbf{R}) \quad \text{and} \quad \mathbf{d} = 2\mu\mathbf{A}\mathbf{p} + \frac{b\,\mathbf{a}}{\mathbf{a}^T\mathbf{a}} \tag{7}$$

and specify the form of $\mathbf{A}$. The stability of this recursion is governed by the eigenvalues of the transition matrix $\mathbf{B}$.

2. LMS ALGORITHM (40 points)

Consider an Nth-order tapped-delay line with input vector $\mathbf{x}(n)$ and desired response $d(n)$. Assume the coefficients $\mathbf{w}(n)$ are adjusted by the least-mean-square (LMS) algorithm with step-size parameter $\mu > 0$.

(a) Suppose we use the modified LMS algorithm

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mathbf{g}(n+1) \tag{8}$$

where $\mathbf{g}(n+1)$ is a *smoothed* (filtered) estimate of the gradient, as follows:

$$\mathbf{g}(n+1) = (1-\mu)\mathbf{g}(n) - \mu e(n)\mathbf{x}(n), \tag{9}$$

and $e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n)$ is the usual a priori error. Let the Wiener weight vector be $\mathbf{w}_o$. Rewrite (8) and (9) in terms of the translated/rotated weight vector $\mathbf{v}'(n) = \mathbf{Q}^T[\mathbf{w}(n) - \mathbf{w}_o]$ where $\mathbf{Q}$ contains the normalized eigenvectors of the input signal autocorrelation matrix $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)] = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, and $\mathbf{\Lambda}$ contains the eigenvalues $\{\lambda_k\}$. Define the quantities $\mathbf{g}'(n) = \mathbf{Q}^T\mathbf{g}(n)$ and $\mathbf{x}'(n) = \mathbf{Q}^T\mathbf{x}(n)$ in your result. Also define $e_o(n)$, the error when operating with the Wiener weights.

(b) Rewrite your answer from part (a) in the following matrix/vector form, and take the expectation of each term:

$$E[\mathbf{u}'(n+1)] = \mathbf{F}E[\mathbf{u}'(n)] + E[\mathbf{h}'(n)] \tag{10}$$

where

$$\mathbf{u}'(n+1) = \begin{bmatrix} \mathbf{v}'(n+1) \\ \mathbf{g}'(n+1) \end{bmatrix}. \tag{11}$$

(Hint: You should rewrite the expression for $\mathbf{v}'(n+1)$ so that it is a function of $\mathbf{g}'(n)$ instead of $\mathbf{g}'(n+1)$. The matrix $\mathbf{F}$ should be a function only of μ and $\mathbf{\Lambda}$. Note that the independence assumption has been used.)

(c) Show that $E[\mathbf{h}'(n)] = \mathbf{0}$.

3. LEAST-SQUARES (40 points)

Let the output *vector* for an FIR filter with N coefficients be written as $\mathbf{y}(n) = \mathbf{X}^T(n)\mathbf{w}(n)$ where

$$\mathbf{X}(n) = [\mathbf{x}(1)| \cdots |\mathbf{x}(n)] \tag{12}$$

is an $N \times n$ matrix containing the filter input vectors up to the present time instant n. The cost function for the method of least squares (LS) (without weighting) is $J(n) = \mathbf{e}^T(n)\mathbf{e}(n)$ where $\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n)$ and $\mathbf{d}(n)$ is the desired signal vector.

(a) For the LS method, show that the error vector at the optimal LS weight vector $\hat{\mathbf{w}}$ (i.e., the weights that minimize $J(n)$) can be expressed as follows:

$$\hat{\mathbf{e}}(n) = [\mathbf{I} - \mathbf{P}(n)]\mathbf{d}(n). \tag{13}$$

Specify $\mathbf{P}(n)$ (which is called a projection matrix).

(b) For large n and $\lambda \approx 1$, it can be shown that

$$\Phi(n) = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)\mathbf{x}^T(i) \approx \left(\frac{1-\lambda^n}{1-\lambda}\right)\mathbf{R} \tag{14}$$

where $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$. Demonstrate that with this approximation, the recursive LS (RLS) algorithm has the form of Newton's Method.

(c) Consider the following modified LS cost function:

$$J_\beta(n) = \sum_{i=1}^{n} \lambda^{n-i}e^2(i) + \lambda^n\beta[\mathbf{w}(n) - \mathbf{w}(0)]^T[\mathbf{w}(n) - \mathbf{w}(0)] \tag{15}$$

where $\beta > 0$ is a weighting factor, and $\mathbf{w}(0) \neq \mathbf{0}$ is the initial weight vector. For $\lambda = 1$, find the LS weight vector which minimizes $J_\beta(n)$. For notational convenience, define $\mathbf{v}(n) = \mathbf{w}(n) - \mathbf{w}(0)$ (which is like the usual translated weight vector, but with respect to the initial weight vector).

(d) For the cost function in part (c) and $\lambda < 1$, show how the RLS algorithm for updating the weight vector $\mathbf{w}(n)$ differs from the usual RLS algorithm. (You do *not* need to show the version of the RLS algorithm that is based on the matrix inversion lemma.)

4. LATTICE FILTERING (40 points)

Consider the least-squares (LS) lattice filter with joint-process estimation. Define the backward residual vector $\mathbf{b}_{m+1}(i) = [b_0(i), \ldots, b_m(i)]^T$ and the joint-process weight vector $\mathbf{k}_m(n) = [k_0(n), \ldots, k_m(n)]^T$. The corresponding normal equation is

$$\mathbf{D}_{m+1}(n)\mathbf{k}_m(n) = \mathbf{t}_{m+1}(n) \tag{16}$$

where $\mathbf{D}_{m+1}(n)$ is a diagonal matrix with nonzero elements $\{B_m(n)\}$ (the backward residual energies) and

$$\mathbf{t}_{m+1}(n) = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{b}_{m+1}(i)d(i) \tag{17}$$

where $0 < \lambda \leq 1$ is the forgetting factor, and $d(i)$ is the desired signal.

(a) We know that $\mathbf{D}_{m+1}(n) = \mathbf{L}_m(n)\mathbf{\Phi}_{m+1}\mathbf{L}_m^T(n)$ where $\mathbf{\Phi}_{m+1}(n)$ is the (deterministic) autocorrelation matrix of the input signal vector $\mathbf{u}_{m+1}(n)$ (similar to (17)), and $\mathbf{L}_m(n)$ is a lower triangular matrix that depends on the backward prediction coefficients $\{c_{m,i}\}$. Show that

$$\mathbf{t}_{m+1}(n) = \mathbf{L}_m(n)\phi_{m+1}(n) \tag{18}$$

and specify the cross-correlation vector $\phi_{m+1}(n)$.

(b) Show that

$$\mathbf{k}_m(n) = \mathbf{L}_m^{-T}(n)\hat{\mathbf{w}}_m(n) = \mathbf{D}_{m+1}^{-1}(n)\mathbf{c}_m^T(n)\phi_{m+1}(n) \tag{19}$$

where $\hat{\mathbf{w}}_m(n) = \mathbf{\Phi}_{m+1}^{-1}(n)\phi_{m+1}(n)$ are the LS weights for the tapped-delay-line filter, and $\mathbf{c}_m(n) = [c_{m,m}, \ldots, c_{m,1}, 1]^T$.

(c) Define the scalar $\rho_m(n) = \mathbf{c}_m^T(n)\phi_{m+1}(n)$. By substituting time-update recursions for $\mathbf{c}_m^T(n)$ and $\phi_{m+1}(n)$, show that

$$\rho_m(n) = \lambda\rho_m(n-1) + \frac{b_m(n)}{\gamma_m(n)}e_m(n) \tag{20}$$

where $\gamma_m(n)$ is the conversion factor, and

$$e_m(n) = d(n) - \mathbf{w}_{m-1}^T(n)\mathbf{u}_m(n) \tag{21}$$

is the estimation error based on $m-1$ weights.