Name: Hank Rinchart 10/13/08

Problem	Points	Score
1(a)	15	
1(b)	15	
1(c)	15	
2(a)	15	
2(b)	15	
3(a)	15	
3(b)	10	
Total	100	

Notes:

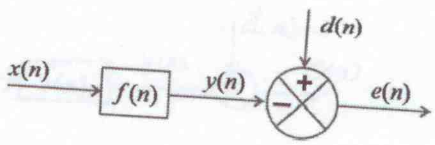
(1) The exam is closed books and notes except for one double-sided sheet of notes.

(2) Please indicate clearly your answer to the problem.

(3) The details of your solutions are more important than the answers. Please explain your solutions clearly and include as many details as possible.

1. For the adaptive system shown to the right, assuming $f(n)$ is a linear, time-invariant moving average filter:

$$y(n) = \sum_{i=0}^{L-1} f(i)x(n-i)$$



(a) Derive the normal equations for the minimum least squares error estimate of the filter coefficients.

(b) Develop the concept of a linear prediction filter based on this model (explain how this model is modified to produce a linear prediction estimate of the filter coefficients).

(c) Derive the expression for the autocorrelation estimate of the linear prediction coefficients.

2. Modify the block diagram shown above to produce the basic LMS adaptive filter.

(a) Derive an expression for estimation of the filter coefficients using an iterative-in-time approach.

(b) Compare and contrast this to the approach in Prob. 1.

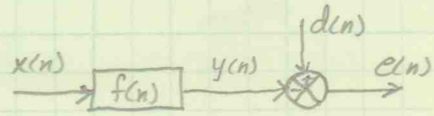
3. Suppose the input signal to the adaptive filter shown above is as follows: $x(n) = ax(n-1) + w(n)$. Assume $w(n)$ is zero-mean white Gaussian noise.

(a) Explain how successful the filter in no. 2 will be at correctly estimating the underlying parameters of this signal. Be as specific as possible and use terms such as the bias, variance, and convergence.

(b) Explain under what conditions we prefer approach no. 2 over approach no. 1. Again, be as specific as possible in your explanation, discussing issues such as, but not limited to, computational complexity.

1) For the adaptive system shown, assuming $f(n)$ is a linear, time-invariant moving average filter:

$$y(n) = \sum_{i=0}^{L-1} f(i)\,x(n-i)$$



(a) Derive the normal equations for the minimum least squares error estimate of the filter coefficients.

Soln: Define the objective fnc. to be minimized: $J = E\{e^2(n)\} = \sum_n e^2(n)$

Error Signal is given by: $e(n) = d(n) - f(n) \circledast x(n)$

or: $e(n) = d(n) - \sum_i f(i)\,x(n-i)$

Differentiate the objective function, J, wrt each filter coeff, $f(j)$:

$$\frac{\partial J}{\partial f(j)} = \frac{\partial E\{e^2(n)\}}{\partial f(j)} = 2\,E\left\{e(n)\,\frac{\partial e(n)}{\partial f(j)}\right\}$$

Substituting w/ the error expression: $\frac{\partial e(n)}{\partial f(j)} = \frac{\partial}{\partial f(j)}\left(d(n) - \sum_i f(i)x(n-i)\right) = -x(n-j)$

Combining:

$$\frac{\partial J}{\partial f(j)} = 2E\left\{e(n)\,\frac{\partial e(n)}{\partial f(j)}\right\} = 2\,E\{e(n)(-x(n-j))\}$$

$$= 2E\left\{\left(d(n) - \sum_i f(i)x(n-i)\right)(-x(n-j))\right\}$$

$$= -2E\{x(n-j)\,d(n)\} + 2E\left\{\sum_i f(i)\,x(n-i)\,x(n-j)\right\}$$

Cross Correlation: $E\{x(n-j)\,d(n)\} = g(n-j,n)$
Auto Correlation: $E\{x(n-i)\,x(n-j)\} = R(n-i,n-j)$

$$= -2(g(n-j,n)) + 2\sum_i R(n-i,n-j)\,f(i)$$

$$= 2\left[\sum_i R(n-i,n-j)f(i) - g(n-j,n)\right]$$

Set this expression to zero: $2\left[\sum_i R(n-i,n-j)f(i) - g(n-j,n)\right] = 0$

Normal Equations: $\sum_i R(n-i,n-j)\,f(i) = g(n-j,n)$

---

1) (b) Develop the concept of a linear prediction filter based on this model (explain how this model is modified to produce a linear prediction estimate of the filter coefficients).

Soln: For a zero mean, stationary signal $x(n)$, the linear prediction is of the form:

$$\hat{x}(n) = \sum_{i=1}^{p} a_i \, x(n-n_0-i)$$

where n_0 = prediction distance
a_i = prediction coefficients

W/out loss of generality, assume $n_0 = 0$

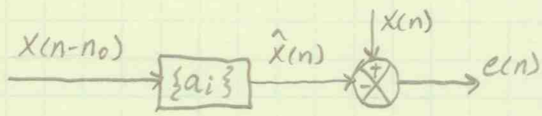
Prediction error is: $e(n) = x(n) - \hat{x}(n) = x(n) - \sum_{i=1}^{p} a_i x(n-i)$

The least squares technique from (1a) will be used:

Minimize the objective function: $J = E\{e^2(n)\} = E\{(x(n) - \hat{x}(n))^2\}$

$$= E\left\{\left(x(n) - \sum_{i=1}^{p} a_i x(n-i)\right)^2\right\}$$

This approach is essentially the same as the least squares approach with the following differences:
Input: $x(n-n_0)$
Desired signal: $x(n)$
Filter Coeff. replaced w/ Pred. Coeff.: a_i



1) (c) Derive the expression for the autocorrelation estimate of the linear prediction coefficients.

Soln: Minimize the objective function and set to zero:

$$J = E\{(x(n) - \sum_{i=1}^{p} a_i x(n-i))^2\}$$

$$= E\{x^2(n)\} - 2E\{\sum_{i=1}^{p} a_i x(n)x(n-i)\} + E\{(\sum_{i=1}^{p} a_i x(n-i))^2\}$$

$$= E\{x^2(n)\} - 2\sum_{i=1}^{p} a_i (E\{x(n)x(n-i)\}) + E\{(\sum_{i=1}^{p} a_i x(n-i))^2\}$$

Differentiate the objective function wrt prediction coefficients, a_i:

$$\frac{\partial J}{\partial a_i} = \frac{\partial}{\partial a_i}\left\{ E\{x^2(n)\} - 2\sum_{i=1}^{p} a_i (E\{x(n)-x(n-i)\}) + E\{(\sum_{i=1}^{p} a_i x(n-i))^2\}\right\}$$

$$= -2E\{x(n)x(n-\ell)\} + 2E\{\sum_{i=1}^{p} a_i x(n-i)\}x(n-\ell)$$

Setting to zero and rearranging:

$$E\{\sum_{i=1}^{p} a_i x(n-i)\}x(n-\ell) = E\{x(n)x(n-\ell)\}$$

$$\sum_{i=1}^{p} a_i E\{x(n-i)x(n-\ell)\} = E\{x(n)-x(n-\ell)\}$$

Define the Covariance Function: $c(i,j) = E\{x(n-i, n-j)\}$

Rewriting the Prediction Equation becomes the Yule-Walker Equation:

$$\sum_{i=1}^{p} a_i c(i,\ell) = c(0,\ell)$$

This is the Covariance Method for Linear Prediction, and in matrix form becomes:

$$Ca = c$$

Assuming stationary inputs, the covariance can be converted to autocorrelation:

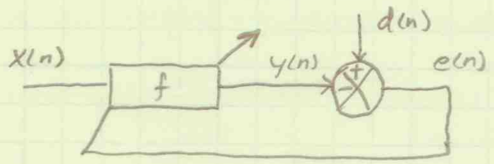
$$Ca = c \implies Ra = r$$

Prediction coefficients can be determined by solving:

$$a = R^{-1}r \quad \text{where } a = \begin{Bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{Bmatrix}, \; R = \begin{Bmatrix} r(0) & r(1) & \cdots & r(p-1) \\ r(1) & \cdots & \cdots & r(p-2) \\ \cdots & \cdots & \cdots & \vdots \\ r(p-1) & r(p-2) & \cdots & r(0) \end{Bmatrix}, \; r = \begin{Bmatrix} r(1) \\ r(2) \\ \vdots \\ r(p) \end{Bmatrix}$$

Since R is Toeplitz, the inverse can be computed efficiently using an iterative algorithm.

2) Modify the block diagram from Prob. 1 to produce the basic LMS adaptive filter.



(a) Derive an expression for estimation of the filter coefficients using an iterative-in-time approach

Soln: Estimate the gradient of the mean-squared error by the instantaneous gradient of the squared error:

$$\nabla \hat{J}_n = \frac{\partial \hat{J}}{\partial f_n} = \frac{\partial e^2(n)}{\partial f_n}$$

In terms of the error signal and input signal:

$$f_{n+1} = f_n - \frac{\alpha}{2} \nabla \hat{J}_n \quad \text{where} \quad \hat{J}_n = e^2(n) = (d(n) - f_n^T x_n)^2$$

$$\therefore \nabla \hat{J}_n = \frac{\partial \hat{J}}{\partial f_n} = 2\, e(n) \frac{\partial e(n)}{\partial f_n} = -2\, e(n)\, x_n$$

Therefore: $\underline{f_{n+1} = f_n + \alpha\, e(n)\, x_n}$

(b) Compare and contrast this to the approach in Prob. 1

Soln: The LMS update is computationally simple. It requires no matrix inversions or averages. The total operation count per update is two times the filter length: $2L$.
As an example, the Levinson LS algorithm requires $2L^2$ operations
However, by replacing the performance index J with the instantaneous version, $\hat{J}$, the LMS algorithm is not expected to converge to the exact least squares solution.

3) Suppose the input signal to the adaptive filter is as follows: $x(n) = a x(n-1) + w(n)$
Assume $w(n)$ is zero-mean white Gaussian noise.

(a) Explain how successful the filter in Prob. 2 will be at correctly estimating the underlying parameters of this signal. Be as specific as possible and use terms such as bias, variance, and convergence.

Approach no. 2 is not exactly solving the same problem as no. 1 due to using the instantaneous value of the error signal. There is no guarantee of convergence to the exact solution. The steady state error will be higher than for the least squares solution, even if the adaptive filter converges in the mean. From the independence assumption, the instantaneous gradient estimate is unbiased, but an ill-chosen step size could introduce bias in the algorithm. Algorithm stability is based on stepsize which is constrained by the filter length, L.

$$0 < \alpha < \frac{2}{L E\{x^2(n)\}}$$

The algorithm will become unstable if α is chosen outside this bound.

(b) Explain under what conditions we prefer approach no. 2 over approach no. 1. Again be as specific as possible in your explanation, discussing ideas such as, but not limited to, computational complexity.

The LMS algorithm would be preferred for problems involving large filter lengths and in applications such as speech recognition where the algorithm is required to track a changing signal. The simple update equation is much more computationally efficient than the LS approach. This makes the LMS algorithm applicable to real-time scenarios.