The Least-Mean-Square(LMS)Algorithm

Lecture note

Yajie Wu

Shanghai University

2024 年 9 月 26 日

1. Introduction
2. The LMS Algorithm
3. Some Properties of the LMS Algorithm

- The least-mean-square (LMS) is a search algorithm in which a simplification of the gradient vector computation is made possible by appropriately modifying the objective function .
- The convergence characteristics of the LMS algorithm are examined in order to establish a range for the convergence factor that will guarantee stability. The convergence speed of the LMS is shown to be dependent on the eigenvalue spread of the input signal correlation matrix .
- The main features that attracted the use of the LMS algorithm are low computational complexity, proof of convergence in stationary environment,unbiased convergence in the mean to the Wiener solution, and stable behavior when implemented with finite-precision arithmetic.

1. Introduction
2. The LMS Algorithm
   - basic problem
   - LMS Algorithm
3. Some Properties of the LMS Algorithm

1. Introduction
2. The LMS Algorithm
   - basic problem
   - LMS Algorithm
3. Some Properties of the LMS Algorithm

- In Chap.2 , the optimal solution leads to the minimum mean-square error in estimating the reference signal d(k).
- The solution is given by

$$w_o = R^{-1}p \tag{1}$$

where $R = E[x(k)x^T(k)]$ and $p = E[d(k)x(k)]$, assuming that d(k) and x(k) are jointly WSS.

- If good estimates of matrix R, denoted by $\hat{R}(k)$, and of vector p, denoted by $\hat{p}(k)$, are available, a steepest-descent-based algorithm can be used to search the Wiener solution of (1) as follows:

$$\begin{aligned} w(k+1) &= w(k) - \mu\hat{g}_w(k) \\ &= w(k) + 2\mu(\hat{p}(k) - \hat{R}(k)w(k)) \end{aligned} \tag{2}$$

- for $k = 0, 1, 2, ...,$where$\hat{g}_w(k)$represents an estimate of the gradient vector of the objective function with respect to the filter coefficients.

- One possible solution is to estimate the gradient vector by employing instantaneous estimates for R and p as follows:

$$\hat{R}(k) = x(k)x^T(k)$$
$$\hat{p}(k) = d(k)x(k) \tag{3}$$

- The resulting gradient estimate is given by

$$\begin{aligned}\hat{g}_w(k) &= -2d(k)x(k) + 2x(k)x^T(k)w(k) \\ &= 2x(k)(-d(k) + x^T(k)w(k)) \\ &= -2e(k)x(k)\end{aligned} \tag{4}$$

- Note that if the objective function is replaced by the instantaneous square error $e^2(k)$, instead of the MSE, the above gradient estimate represents the true gradient vector since

$$\begin{aligned}\frac{\partial e^2(k)}{\partial w} &= \left[2e(k)\frac{\partial e(k)}{\partial w_0(k)}\ 2e(k)\frac{\partial e(k)}{\partial w_1(k)} \cdots 2e(k)\frac{\partial e(k)}{\partial w_N(k)}\right]^T \\ &= -2e(k)x(k) \\ &= \hat{g}_w(k)\end{aligned} \tag{5}$$

- The resulting gradient-based algorithm is known as the least-mean-square (LMS) algorithm, whose updating equation is

$$w(k+1) = w(k) + 2\mu e(k)x(k) \tag{6}$$

where the convergence factor should be chosen in a range to guarantee convergence.

Algorithm 1 LMS algorithm

Initialization

$x(-1) = w(0) = [0\ 0 \cdots 0]^T$

Do for k $\geq$ 0

$e(k) = d(k) - x^T(k)w(k)$

$w(k+1) = w(k) + 2\mu e(k)x(k)$

LMS algorithm

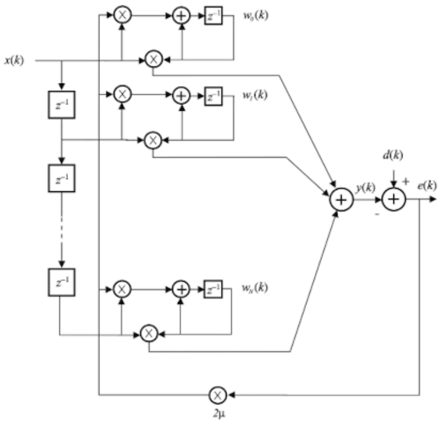


Fig. 3.1 LMS adaptive FIR filter

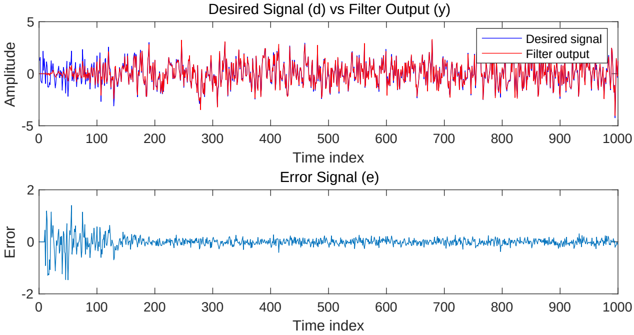
图 1: 1

code

```matlab
% LMS Adaptive FIR Filter Implementation in MATLAB

% Parameters
N = 1000;            % Number of samples
M = 10;              % Number of filter taps (filter order)
mu = 0.01;           % Step size (learning rate)
x = randn(1, N);     % Input signal (white Gaussian noise)
d = filter([1 0.5], 1, x) + 0.1*randn(1, N); % Desired signal with noise

% Initialization
w = zeros(M, 1);     % Initial filter weights (zero)
y = zeros(1, N);     % Filter output
e = zeros(1, N);     % Error signal

% LMS Algorithm
for k = M:N
    % Extract input signal segment
    x_segment = x(k:-1:k-M+1)';

    % Filter output (dot product of weights and input signal)
    y(k) = w' * x_segment;

    % Error calculation (desired signal - output)
    e(k) = d(k) - y(k);

    % Update filter weights using LMS update rule
    w = w + 2 * mu * e(k) * x_segment;
end
```

图 2: 2

simulation


Desired Signal (d) vs Filter Output (y)
Desired signal
Filter output
Amplitude
5
0
-5
0
100
200
300
400
500
600
700
800
900
1000
Time index
Error Signal (e)
Error
2
0
-2
0
100
200
300
400
500
600
700
800
900
1000
Time index

1. Introduction
2. The LMS Algorithm
3. Some Properties of the LMS Algorithm
   - Gradient Behavior
   - Convergence Behavior of the Coefficient Vector

Gradient Behavior

- he ideal gradient direction required to perform a search on the MSE surface for the optimum coefficient vector solution is

$$\begin{aligned} g_w(k) &= 2\{E[x(k)x^T(k)]w(k) - E(d(k)x(k))\} \\ &= 2[Rw(k) - p] \end{aligned} \quad (7)$$

- In the LMS algorithm, instantaneous estimates of R and p are used to determine the search direction, i.e.,

$$\hat{g}_w(k) = 2[x(k)x^T(k)w(k) - d(k)x(k)] \quad (8)$$

Gradient Behavior

- The LMS gradient direction has the tendency to approach the ideal gradient direction since for a fixed coefficient vector w .

$$E[\hat{g}_w(k)] = 2E[x(k)x^T(k)]w - E[d(k)x(k)] = g_w \qquad (9)$$

- vector $\hat{g}_w(k)$ can be interpreted as an unbiased estimate of g_w.
- In an ergodic environment, if, for a fixed w vector, $\hat{g}_w(k)$ is calculated for a large number of inputs and reference signals, the average direction tends to g_w, i.e.,

$$\lim_{M\to\infty} \frac{1}{M} \sum_{i=1}^{M} \hat{g}_w(k+i) \to g_w \qquad (10)$$

Convergence Behavior of the Coefficient Vector

- Measurement white noise n(k) with zero mean and variance σ_n^2 is added to the output of the unknown system.
- The error in the adaptive filter coefficients as related to the ideal coefficient vector w_o, in each iteration, is described by the N + 1-length vector.

$$\Delta w(k) = w(k) - w_o \tag{11}$$

Convergence Behavior of the Coefficient Vector

- LMS algorithm can alternatively be described by

$$\begin{aligned}\Delta w(k+1) &= \Delta w(k) + 2\mu e(k)x(k) \\ &= \Delta w(k) + 2\mu x(k)[x^T(k)w_o + n(k) - x^T(k)w(k)] \\ &= \Delta w(k) + 2\mu x(k)[e_o(k) - x^T(k)\Delta w(k)] \\ &= [I - 2\mu x(k)x^T(k)]\Delta w(k) + 2\mu e_o(k)x(k) \quad (12)\end{aligned}$$

$e_o(k)$ is the optimal output error given by

$$e_o(k) = d(k) - w_o^T x(k) = w_o^T x(k) + n(k) - w_o^T x(k) = n(k) \quad (13)$$

Convergence Behavior of the Coefficient Vector

- The expected error in the coefficient vector is then given by

$$E[\Delta w(k+1)] = E\{[I-2\mu x(k)x^T(k)]\Delta w(k)\}+2\mu E[e_o(k)x(k)] \quad (14)$$

- If it is assumed that the elements of $x(k)$ are statistically independent of the elements of $w(k)$ and orthogonal to $e_o(k)$,(3.14) can be simplified as follows:

$$\begin{aligned} E[\Delta w(k+1)] &= \{I - 2\mu E[x(k)x^T(k)]\}E[\Delta w(k)] \\ &= (I - 2\mu R)E[\Delta w(k)] \end{aligned} \quad (15)$$

- The above expression leads to

$$E[\Delta w(k+1)] = (I - 2\mu R)^{k+1}E[\Delta w(0)] \quad (16)$$

Convergence Behavior of the Coefficient Vector

- Equation (3.15) premultiplied by Q^T , where Q is the unitary matrix that diagonalizes R through a similarity transformation,yields

$$
\begin{aligned}
E[Q^T \Delta w(k+1)] &= (I - 2\mu Q^T R Q) E[Q^T \Delta w(k)] \\
&= E[\Delta w'(k+1)] \\
&= (I - 2\mu\Lambda) E[\Delta w'(k)] \\
&= \begin{bmatrix} 1-2\mu\lambda_0 & 0 & \cdots & 0 \\ 0 & 1-2\mu\lambda_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1-2\mu\lambda_N \end{bmatrix} E[\Delta'(k)] \quad (17)
\end{aligned}
$$

Convergence Behavior of the Coefficient Vector

- where $\Delta w^{'}(k+1) = Q^T \Delta w(k+1)$ is the rotated-coefficient-error vector. The applied rotation yielded an equation where the driving matrix is diagonal, making it easier to analyze the equation's dynamic behavior. Alternatively, the above relation can be expressed as

$$E[Q^T \Delta w(k+1)] = (I - 2\mu\Lambda)^{k+1} E[\Delta w^{'}(0)]$$
$$= \begin{bmatrix} (1-2\mu\lambda_0)^{k+1} & 0 & \cdots & 0 \\ 0 & (1-2\mu\lambda_1)^{k+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (1-2\mu\lambda_N)^{k+1} \end{bmatrix} E[\Delta^{'}(0)] \tag{18}$$

Convergence Behavior of the Coefficient Vector

- This equation shows that in order to guarantee convergence of the coefficients in the mean, the convergence factor of the LMS algorithm must be chosen in the range

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{19}$$

- The choice of μ as above explained ensures that the mean value of the coefficient vector approaches the optimum coefficient vector w_o.
- It should be mentioned that if the matrix R has a large eigenvalue spread, it is advisable to choose a value for μ much smaller than the upper bound.

Thanks!