Least Mean Square Algorithm

B Swaroop Reddy and G V V Sharma*

Contents

***Abstract*—This manual provides an introduction to the LMS algorithm.**

1 Convex Functions

A single variable function f is said to be convex if

$$f\left[\lambda x + (1-\lambda) y\right] \le \lambda f(x) + (1-\lambda) f(y), \quad (1.1)$$

for $\quad 0 < \lambda < 1.$

1.1 Download and execute the following python script. Is $\ln x$ convex or concave?

```
wget https://raw.githubusercontent.com/
    gadepall/EE2250/master/manual/codes
    /1.1.py
```

1.2 Modify the above python script as follows to plot the parabola $f(x) = x^2$. Is it convex or concave?

```
wget https://raw.githubusercontent.com/
    gadepall/EE2250/master/manual/codes
    /1.2.py
```

1.3 Execute the following script to obtain Fig. **??**. Comment.

```
wget https://raw.githubusercontent.com/
    gadepall/EE2250/master/manual/codes
    /1.3.py
```

1.4 Modify the script in the previous problem for $f(x) = x^2$. What can you conclude?

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

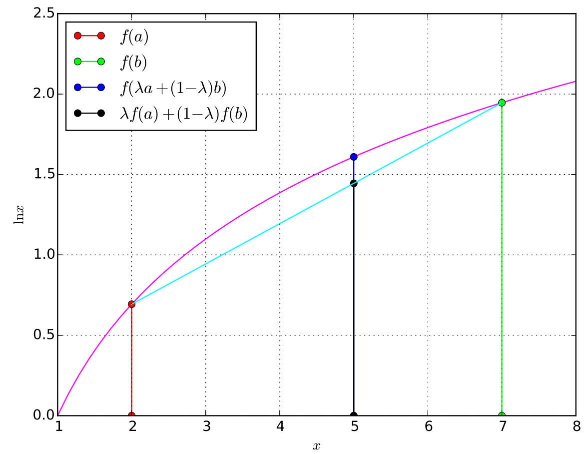

Fig. 1.1: $\ln x$ versus x

.

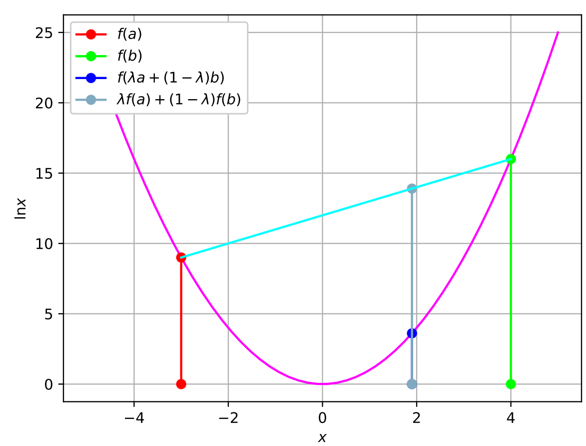

Fig. 1.2: x^2 versus x

.

1.5 Let

$$f(\mathbf{x}) = x_1 x_2, \quad \mathbf{x} \in \mathbf{R}^2 \quad (1.2)$$

Sketch $f(\mathbf{x})$ and deduce whether it is convex. Can you theoretically explain your observation

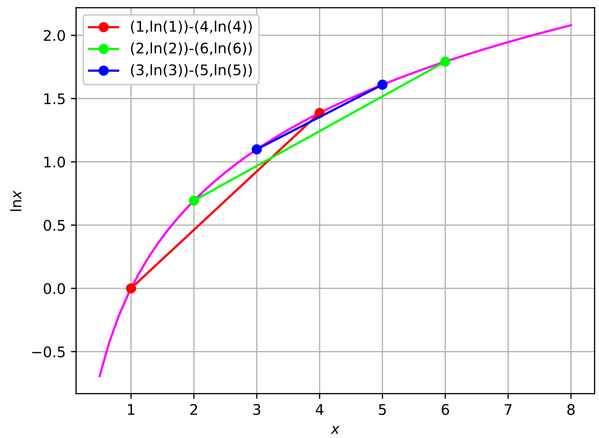

Fig. 1.3: Segments are below the curve

.

using (**??**)?

2 Gradient Descent Method

Consider the problem of finding the square root of a number c. This can be expressed as the equation

$$x^2 - c = 0 \tag{2.1}$$

2.1 Sketch the function for different values of c

$$f(x) = x^3 - 3xc \tag{2.2}$$

and comment upon its convexity.

2.2 Show that (**??**) results from

$$\min_x f(x) = x^3 - 3xc \tag{2.3}$$

2.3 Find a numerical solution for (**??**).
Solution: A numerical solution for (**??**) is obtained as

$$x_{n+1} = x_n - \mu f'(x) \tag{2.4}$$
$$= x_n - \mu\left(3x_n^2 - 3c\right) \tag{2.5}$$

where x_0 is an inital guess.

2.4 Write a program to implement (**??**).
Solution: Download and execute

```
wget
https://raw.githubusercontent.com/gadepall/
    EE2250/master/manual/codes/square_root
    .py
```

3 Audio Source Files

3.1 Get the **audio_source**

```
svn checkout https://github.com/gadepall/
    EE5347/trunk/audio_source
cd audio_source
```

3.2 Play the **signal_noise.wav** and **noise.wav** file. Comment.
Solution: signal_noise.wav contains a human voice along with an instrument sound in the background. This instrument sound is captured in **noise.wav**.

4 Problem Formulation

4.1 See Table **??**. The goal is to extract the human voice $e(n)$ from $d(n)$ by suppressing the component of $\mathbf{X}(n)$. Formulate an equation for this.
Solution: The maximum component of $\mathbf{X}(n)$ in

Signal	Label	Type	Filename
Known	d(n)	Human+Instrument	signal noise.wav
	X(n)	Instrument	noise.wav
Unknown	e(n)	Human estimate	
	W(n)	Weight Vector	

TABLE 4.1

$d(n)$ can be estimated as

$$\mathbf{W}^T(n)\mathbf{X}(n) \tag{4.1}$$

where

$$\mathbf{W}(n) = \begin{bmatrix} w_1(n) \\ w_2(n) \\ w_3(n) \\ .. \\ .. \\ w_{n-M+1}(n) \end{bmatrix}_{MX1} \tag{4.2}$$

Intuitively, the human voice $e(n)$ is obtained after removing as much of $\mathbf{X}(n)$ from $d(n)$ as possible. The first step in this direction is to estimate $\mathbf{W}$ in (**??**) using the metric

$$\min_{\mathbf{W}(n)} \|d(n) - \mathbf{W}^T(n)\mathbf{X}(n)\|^2 \tag{4.3}$$

The human voice can be then obtained as

$$e(n) = d(n) - \mathbf{W}^T(n)\mathbf{X}(n) \tag{4.4}$$

5 LMS Algorithm

5.1 Show using (**??**) that

$$\nabla_{\mathbf{W}(n)} e^2(n) = \frac{\partial e^2(n)}{\partial \mathbf{W}(n)} \tag{5.1}$$

$$= -2\mathbf{X}(n)d(n) + 2\mathbf{X}(n)X^T(n)\mathbf{W}(n) \tag{5.2}$$

5.2 Use the gradient descent method to obtain an algorithm for solving (**??**)
Solution: The desired algorithm can be expressed as

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \bar{\mu}[\nabla_{\mathbf{W}(n)} e^2(n)] \tag{5.3}$$

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu\mathbf{X}(n)e(n) \tag{5.4}$$

where $\mu = \bar{\mu}$.

5.3 Write a program to suppress $\mathbf{X}(n)$ in $d(n)$.
Solution: Execute

```
wget https://raw.githubusercontent.com/
    gadepall/EE5347/master/lms/codes/
    LMS_NC_SPEECH.py
```

6 Wiener-Hopf Equation

6.1 Using (**??**), show that

$$E[e^2(n)] = r_{dd} - W^T(n)r_{xd} - r_{xd}^T\mathbf{W}(n) + W^T(n)R\mathbf{W}(n) \tag{6.1}$$

where

$$r_{dd} = E[d^2(n)] \tag{6.2}$$

$$r_{xd} = E[\mathbf{X}(n)d(n)] \tag{6.3}$$

$$R = E[\mathbf{X}(n)\mathbf{X}^T(n)] \tag{6.4}$$

6.2 By computing

$$\frac{\partial J(n)}{\partial \mathbf{W}(n)} = 0, \tag{6.5}$$

show that the optimal solution for

$$W^*(n) = \min_{\mathbf{W}(n)} E\left[e^2(n)\right] = R^{-1}r_{xd} \tag{6.6}$$

This is the Wiener optimal solution.

7 Convergence of the LMS Algorithm

7.1 *Convergence in the Mean*

7.1.1 Show that R in (**??**) is symmetric as well as positive definite.

Let

$$\tilde{W}(n) = \mathbf{W}(n) - W_* \tag{7.1}$$

where W_* is obtained in (**??**). Also, according to the LMS algorithm,

$$W(n+1) = \mathbf{W}(n) + \mu\mathbf{X}(n)e(n) \tag{7.2}$$

$$e(n) = d(n) - X^T(n)\mathbf{W}(n) \tag{7.3}$$

7.1.2 Show that

$$E\left[\tilde{W}(n+1)\right] = [I - \mu R]E\left[\tilde{W}(n)\right] \tag{7.4}$$

7.1.3 Show that

$$R = U\Lambda U^T \tag{7.5}$$

for some U, Λ, such that Λ is a diagonal matrix and $U^TU = I$.

7.1.4 Show that

$$\lim_{n\to\infty} E\left[\tilde{W}(n+1)\right] = 0 \iff \lim_{n\to\infty}[I - \mu\Lambda]^n = 0 \tag{7.6}$$

7.1.5 Using (**??**), show that

$$0 < \mu < \frac{2}{\lambda_{\max}} \tag{7.7}$$

where $\lambda_{\max}$ is the largest entry of Λ.

7.2 *Convergence in Mean-square sense*

Let

$$\mathbf{X}(n) = \begin{bmatrix} X_1(n) \\ X_2(n) \end{bmatrix} \tilde{W}(n) = \begin{bmatrix} \tilde{W}_1(n) \\ \tilde{W}_2(n) \end{bmatrix} \tag{7.8}$$

7.2.1 Show that

$$E[\tilde{W}^T(n)\mathbf{X}(n)X^T(n)\tilde{W}(n)] = E[\tilde{W}^T(n)R\tilde{W}(n)] \tag{7.9}$$

for R defined in (**??**).

7.2.2 Show that

$$J(n) = E[e^2(n)] = E[e_*^2(n)] + E[\tilde{W}(n)\mathbf{X}(n)\mathbf{X}(n)^T\tilde{W}(n)^T] - E[\tilde{W}(n)\mathbf{X}(n)e_*(n)] - E[e_*(n)X^T(n)\tilde{W}^T(n)] \tag{7.10}$$

where

$$\tilde{W}(n) = W(n) - W_* \tag{7.11}$$

$$e_*(n) = d(n) - W_*\mathbf{X}(n) \tag{7.12}$$

7.2.3 Show that

$$E\left[\tilde{W}(n)\mathbf{X}(n)e_*(n)\right] = E\left[e_*(n)X^T(n)\tilde{W}^T(n)\right] = 0 \tag{7.13}$$

7.2.4 Show that

$$E\left[\tilde{W}^T(n)R\tilde{W}(n)\right] = \text{trace}\left(E\left[\tilde{W}^T(n)R\tilde{W}(n)\right]\right) \tag{7.14}$$

$$= \text{trace}\left(E\left[\tilde{W}(n)\tilde{W}^T(n)\right]R\right) \tag{7.15}$$

7.2.5 Using (**??**), (**??**) and (**??**), show that

$$\tilde{W}(n+1) = \left[I - \mu\mathbf{X}(n)X^T(n)\right]\tilde{W}(n)+\mu\mathbf{X}(n)e_*(n) \tag{7.16}$$

7.2.6 Let $\mu^2 \to 0$. Using (**??**) and (**??**), show that

$$E\left[\tilde{W}(n+1)\tilde{W}^T(n+1)\right]$$
$$= (I - 2\mu R)\, E\left[\tilde{W}(n)\tilde{W}^n(n)\right] \tag{7.17}$$

7.2.7 Show that

$$\lim_{n\to\infty} E\left[\tilde{W}(n)\tilde{W}^T(n)\right] = 0 \iff 0 < \mu < \frac{1}{\lambda_{max}} \tag{7.18}$$

7.2.8 Find the value of the cost function at infinity i.e. $J(\infty)$

7.2.9 How can you choose the value of μ from the convergence of both in mean and mean-square sense?