

Gradient Boost: Least Mean Square Algorithm

B Swaroop Reddy and G V V Sharma*

CONTENTS

1	Convex Functions	1
2	Gradient Descent Method	2
3	Audio Source Files	2
4	Problem Formulation	2
5	LMS Algorithm	3

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

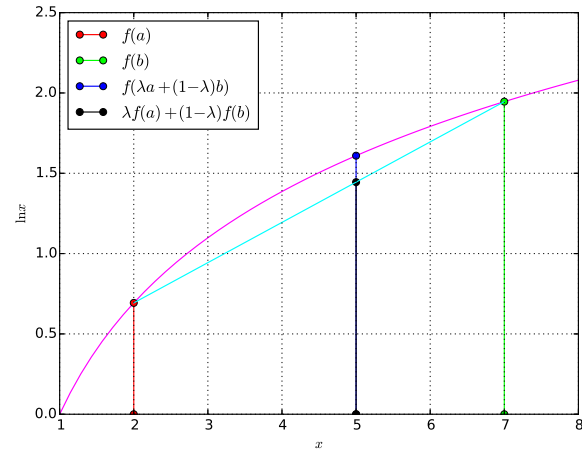


Fig. 1.1: $\ln x$ versus x

1 CONVEX FUNCTIONS

A single variable function f is said to be convex if

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

for $0 < \lambda < 1$.

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
```

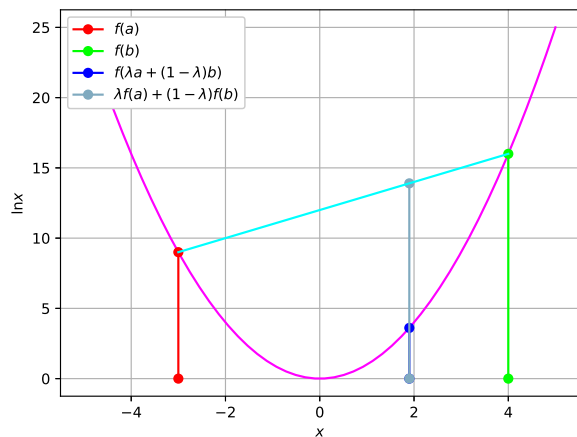


Fig. 1.2: x^2 versus x

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

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

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All content in this manual is released under GNU GPL. Free and open source.

/1.3.py

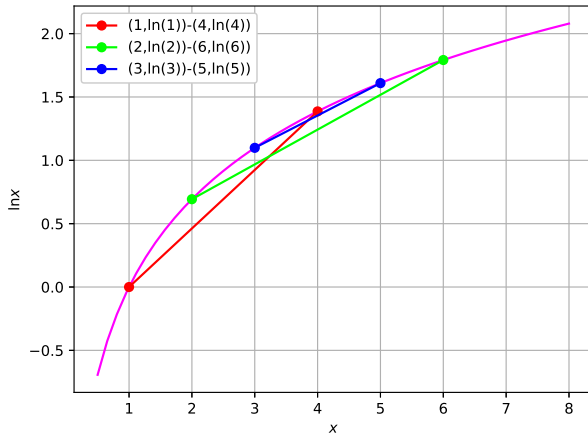


Fig. 1.3: Segments are below the curve

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

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 using (1.1)?

2 GRADIENT DESCENT METHOD

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

$$x^2 - c = 0 \quad (2.1)$$

2.1 Sketch the function for different values of c

$$f(x) = x^3 - 3xc \quad (2.2)$$

and comment upon its convexity.

2.2 Show that (2.1) results from

$$\min_x f(x) = x^3 - 3xc \quad (2.3)$$

2.3 Find a numerical solution for (2.1).

Solution: A numerical solution for (2.1) is obtained as

$$x_{n+1} = x_n - \mu f'(x) \quad (2.4)$$

$$= x_n - \mu (3x_n^2 - 3c) \quad (2.5)$$

where x_0 is an initial guess.

2.4 Write a program to implement (2.5).

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 4.1. 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
	$\mathbf{X}(n)$	Instrument	noise.wav
Unknown	$e(n)$	Human estimate	
	$\mathbf{W}(n)$	Weight Vector	

TABLE 4.1

$d(n)$ can be estimated as

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

where

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

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 (4.1) using the metric

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

The human voice can be then obtained as

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

5 LMS ALGORITHM

5.1 Show using (4.4) that

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

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

5.2 Use the gradient descent method to obtain an algorithm for solving (4.3)

Solution: The desired algorithm can be expressed as

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

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

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
```