Prediction Model Parameter Identification Using LMS

Hsiaoping, Ni
ASU ID: 1220537786
hsiaopin@asu.edu

Abstract **– In this assignment, we have to estimate a particular parameter in the assigned equation for generating training data by using the LMS algorithm. Basically, LMS in Machine Learning is a type of supervised learning method.**

Keywords – LMS, Python, Gaussian, white noise, standard deviation, learning, instantaneous cost.

I. INTRODUCTION

LMS (Least Mean Square) is one of the supervised Machine Learning algorithms. A supervised learning method is a method in which we gather training data with input values. Generally, it is utilized as a process to find a meaningful structural function in order to predict output values optimally for its legitimate target. In other words, the program has the capacity to learn from a set of training data or features to improve performance.

LMS gives us a goal: based on a systematic computation through the algorithm, the variance (the sum of squares of the errors) needs to be minimized to find the best line of fit. In the human sense, as we learn according to errors, we realize of how to make appropriate adjustments for greater advancement towards the optimized outcome.

So, in LMS algorithm the error is fed back into the algorithm after each sample is input and the weights (in this example, they are values of parameter a) are adjusted. In addition, as the desired value meets the actual one, the learning process is terminated. Therefore, new iteration is not updating the weight anymore.

II. PROBLEM STATEMENT

Python is the main tool to implement and accomplish this assignment. In this assignment, we have to use the LMS algorithm on the given data generated by simulations due to our absent access to the physical process for data collection.

A. *Data collection*

Before applying the LMS algorithm, we first generate 500 data pairs of $\{x(k\text{-}1), x(k)\}$ from the following equation (1):

$$x(k) = \mathrm{a}x(k-1) + \varepsilon(k) \qquad (1)$$

Since the series is white noise, it is random and cannot be predicted. In that case, all the variables have the same variance and are independent. And they are identically distributed with a mean of zero.

From the equation (1), a = 0.99, and $\varepsilon(k)$ is a zero mean Gaussian white noise of variance 0.02. That shows that the variables are drawn from a Gaussian distribution. This can be written as: $\varepsilon(k) \sim N(0, \sigma^2)$, where $\sigma^2 = 0.02$. In this situation, the standard deviation (the square root of the variance) is nearly zero, meaning that those numbers are almost all equal.

However, as for the 500 random input samples generated, a Gaussian white noise with $x(k) \sim N(0, \sigma^2)$, where $\sigma^2 = 0.995$ can be used. Later, when all of facts are gathered, $x(k)$ can be calculated to become estimated values. As for the desired output values $d(k)$, we can discover them from the input values $x(k-1)$. All of these are organized in TABLE 1.

B. *LMS*

After collecting the data, we see that the desired values and the estimated values are different from each other accordingly. Therefore, we need to adjust the parameter a to make them identical. The LMS algorithm is the one that helps estimate the parameter a in equation (1). The algorithm is shown below:

$$\mathrm{a}(k+1) = \mathrm{a}(k) + \alpha[x(k)\mathrm{e}(k)]$$

$$\mathrm{e}(k) = d(k) - \mathrm{a}(k)x(k)$$

We can start from a(0) = 0 (here, the weight value is value a), and use the learning rate, α of 0.001. We want to find the final estimated a value in the process of running algorithm after the learning process is stopped.

III. RESULT

A. Data result from a mathematical model

As the $\varepsilon(k)$ values and the input values ($k-1$ from 1 to 500), $x(k-1)$ are generated, we can find the estimated output values (k from 2 to 501), $x(k)$ straightforwardly from the question (1). Then, we, at the same time, discovers the desired output values (k from 2 to 501 in 'input values $x(k-1)$'), $d(k)$, then comparing to the estimated outcomes, $x(k)$. All of the observation can be seen in TABLE I.

TABLE I
GENERATED DATA BY PYTHON FOR THE LMS ALGORITHM

	k-1	input values $x(k$-1)	k	estimated output values $x(k)$
0	0	-0.040257	1	-0.066687
1	1	-0.007259	2	-0.036614
2	2	-1.016622	3	-1.028231
3	3	0.385505	4	0.404845
4	4	-0.339148	5	-0.320510
...	...	...	...	...
497	497	1.538949	498	1.536485
498	498	-0.446621	499	-0.435470
499	499	0.277009	500	0.287490
500	500	0.430049	501	0.453369
501	501	1.201860	502	1.214361

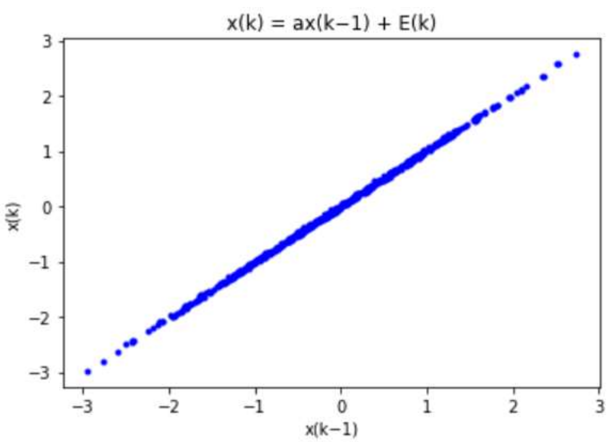

Fig.1 The correlation between the inputs $x(k$-1) and the outputs $x(k)$.

B. Prediction parameters result from the LMS algorithm.

From the equation (1), once e(k) is zero, showing that errors are no more, the equation becomes like this: a(k+1) = a(k). There would be no longer need for learning since the model becomes perfect.

After the completion of the algorithm, we record that the last updated parameter a is about 0.9999999999999937 (about the 79th of iteration begins this value). Therefore, the estimated of a comes out finally (from the for loop in Python, when error becomes zero, the estimate of a will stay 0.9999999999999937) and there would be no necessity to update anymore.

In addition, plot a(k) and the instantaneous cost $E(k)$ for k = 0, 1, 2, …, 5000 are also provided. As for the cost function, the equation looks like this:

$$E(k) = \frac{1}{2}(\mathrm{e}(k))^2$$

As the nth of iteration reaches about 79, the result becomes zero (no more error). Both of the plots are displayed below to show the pattern that there is a starting point when the y-axis value begins to be the same.

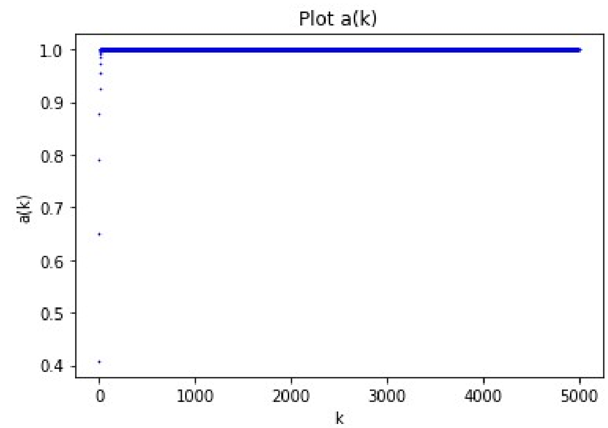

Fig.2 The plot of the correlation between a(k) and k.

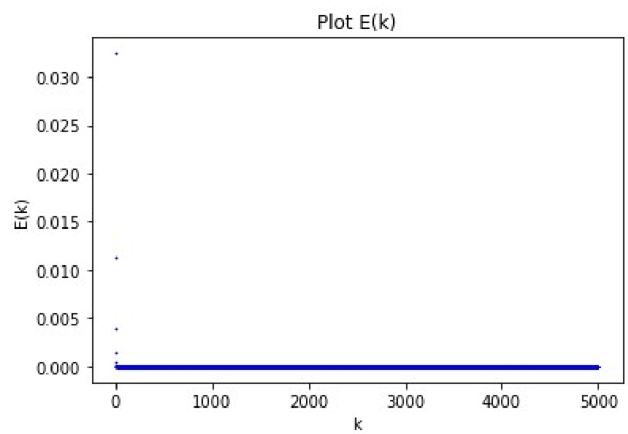

Fig.3 The plot of the correlation between E(k) and k.

REFERENCE

[1] "Statistics how to." https://www.statisticshowto.com/least-squares-regression-line/ (accessed Nov 5, 2014).

[2] EEE 551 Lecture notes