Journal of Telematics and Informatics (JTI)
Vol.9, No.2, 2021, pp. 52~57
ISSN: 2303-3703

A Case of Identity for NLMS and Kaczmarz Algorithms

Muhammad Yasir Siddique Anjum
Department of Electrical Engineering, National University of Modern Languages, Rawalpindi Campus, Pakistan
e-mail: yasir.siddique@numl.edu.pk

Abstract

In this paper, we will demonstrate, by employing basic linear algebra, that two seemingly disconnected algorithms, the Kaczmarz algorithm, familiar to mathematics community as an iterative solver of linear systems, and the Normalized LMS (NLMS) algorithm, known to the signal processing community as a self-learning adaptive filter, are identical. In this paper, we have provided a simple linear algebraic proof of the relationship between Kaczmarz and NLMS algorithms which demonstrates that both algorithms are identical

Keywords: *Kaczmarz, LMS, NLMS, adaptive filter*

1. Introduction

Recent proposition of randomized Kaczmarz algorithm has generated an upheaval in scientific community. It has been shown that Kaczmarz algorithm can achieve better convergence when it selects the rows of a linear system, the one it is trying to solve, in a random fashion rather than a progressive one [1]. This upheaval has attracted a lot of researchers towards the Kaczmarz algorithm, and engineers towards its potential applications. As a result there has been a revival of Kaczmarz algorithm, as one would say it has experienced a renaissance. Many others have shared the lead and claims are being made for yet faster and faster convergence of the algorithm [2-4]. On theoretical side, the algorithm is paving its way in linear algebra [5, 6], approximation theory [7], statistics and data analysis [8], non-linear analysis [7], etc. Also on the applied side, the algorithm is gaining much popularity and is being used to solve the problems arising in atmospheric tomography [9], medical tomography [10], optics [11], image reconstruction [12], GPU computing [13], etc. All this is being attributed to its ability to solve a linear system in an iterative fashion, and to do it quickly, the convergence.

But here we are compelled to say that yet there exists another algorithm that is exactly the same as Kaczmarz algorithm with similar convergence properties. Only it has a different name. It is known by the name of Normalized LMS and is familiar to signal processing community as an adaptive filter [14]. In adaptive signal processing, it has been widely used for system identification, array beam-forming, channel equalization, acoustic echo cancellation, etc. The NLMS filter iteratively solves a system of linear equations by minimizing the error between its output and the desired output and, to speed up convergence, selects an adaptive sep-size that minimizes the error with respect to aposteriori output. The system of linear equations and, hence, the rows of the system matrix in question are formed by a time-delayed input vector whose every entry is a Gaussian random variable with a zero mean and a finite variance. Therefore, it can be argued that, in the context of row selection, the NLMS algorithm is already random as the rows are statistically uncorrelated. And its convergence properties are already very well-defined.

But surprisingly, the two algorithms seem disconnected at the moment. Former is familiar to mathematics community and the latter has the approval of signal processing community. Signal processing literature, including books and journals, barely include the Kaczmarz algorithm while the mathematics community is even less inclined to NLMS algorithm. Whenever there is the problem of solving a system of linear equations in an iterative manner with good convergence properties, both work with their own solutions. While some have discussed them in entirely disengaged manner [15], even fewer have hinted a relationship between them but without the explicit proof [16]. Therefore, we will show in this paper, in unequivocal terms, that both algorithms are identical. Only difference is that of the nomenclature, presumably arising due to their different application contexts and, hence, the notations in vogue in their respective communities. Let it be the scientific historians who debate which preceded the other but the fact is both are same. It is this fact we will try to demonstrate in this paper by employing basic linear algebraic. As a side note, we will also establish a serial link between Steepest Descent, LMS, NLMS, and Kaczmarz algorithms. We will show, in systematic sequential steps, that how they are related and how one can be derived from the other by simple substitutions which will lead us to the ultimate case of identity for the Kaczmarz algorithm. The case may also be viewed, from a pedagogical viewpoint, as an alternate proof of the Kaczmarz algorithm.

2. Proof

We begin the proof by first writing Kaczmarz's recursion equation for reference [3].

$$\boldsymbol{x}[\boldsymbol{n}+\boldsymbol{1}] = \boldsymbol{x}[\boldsymbol{n}] + \left(b_j - \boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}]\right)\frac{\boldsymbol{a_j}}{\boldsymbol{a_j}\boldsymbol{a_j}^{\boldsymbol{T}}} \tag{1}$$

Now we lay out nomenclature and proceed with the proof. Let a system of linear equations be defined as,

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{2}$$

such that $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{x} \in \mathbb{R}^{n}$, and $\boldsymbol{b} \in \mathbb{R}^{m}$. Since we are seeking an iterative solution, Eq. (2) is modified as,

$$\boldsymbol{Ax}[\boldsymbol{n}] = \boldsymbol{b} \tag{3}$$

n denotes the iteration number. Re-writing Eq. (3) as,

$$\begin{bmatrix} \boldsymbol{a_1}^{\boldsymbol{T}} \\ \boldsymbol{a_2}^{\boldsymbol{T}} \\ \vdots \\ \boldsymbol{a_m}^{\boldsymbol{T}} \end{bmatrix} \boldsymbol{x}[\boldsymbol{n}] = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{4}$$

Selecting the j-th row $\boldsymbol{a_j}^{\boldsymbol{T}}$ of $\boldsymbol{A}$ matrix from Eq. (4),

$$\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}] = b_j \tag{5}$$

$b_j \in \mathbb{R}$ is the j-th component of $\boldsymbol{b}$ vector. Since our initial choice of $\boldsymbol{x}[\boldsymbol{n}]$ is purely arbitrary, there can be an error $e_j[n]$,

$$e_j[n] = b_j - \boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}] \tag{6}$$

Squaring Eq. (6), we obtain the cost function $\xi \in \mathbb{R}$.

$$\xi = {b_j}^2 - 2\boldsymbol{x}[\boldsymbol{n}]^{\boldsymbol{T}}\boldsymbol{a_j}b_j + \boldsymbol{x}[\boldsymbol{n}]^{\boldsymbol{T}}\boldsymbol{a_j}\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}] \tag{7}$$

Computing the instantaneous gradient of Eq. (7) by minimizing ξ with respect to $\boldsymbol{x}[\boldsymbol{n}]$,

$$\boldsymbol{\nabla\xi} = -2\boldsymbol{a_j}b_j + 2\boldsymbol{a_j}\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}] \tag{8}$$

Re-arranging Eq. (8),

$$\boldsymbol{\nabla\xi} = -2\left(b_j - \boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}]\right)\boldsymbol{a_j} \quad (9)$$

Employing SD algorithm [17] to compute $\boldsymbol{x}[\boldsymbol{n}]$ by using the gradient computed in Eq. (9),

$$\boldsymbol{x}[\boldsymbol{n}+\boldsymbol{1}] = \boldsymbol{x}[\boldsymbol{n}] + \mu\boldsymbol{\nabla\xi} \quad (10)$$

Substituting Eq. (9) in Eq. (10),

$$\boldsymbol{x}[\boldsymbol{n}+\boldsymbol{1}] = \boldsymbol{x}[\boldsymbol{n}] - 2\mu\left(b_j - \boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}]\right)\boldsymbol{a_j} \quad (11)$$

Eq. (11) is known as LMS algorithm. It is almost close to Kaczmarz equation except for the denominator term $\boldsymbol{a_j}\boldsymbol{a_j}^{\boldsymbol{T}}$. We will obtain this term by solving for an optimal μ. By optimal we mean a value of μ that minimizes the aposteriori error defined as [14],

$$e_j[n+1] = b_j - \boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}+\boldsymbol{1}] \quad (12)$$

Substituting Eq. (11) in Eq. (12),

$$e_j[n+1] = b_j - \boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}] + 2\mu\boldsymbol{a_j}^{\boldsymbol{T}}\left(b_j - \boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}]\right)\boldsymbol{a_j} \quad (13)$$

Substituting Eq. (6) in Eq. (13) and observing that $e_j[n]$ is a scalar,

$$e_j[n+1] = e_j[n] + 2\mu\boldsymbol{a_j}^{\boldsymbol{T}}e_j[n]\boldsymbol{a_j} = e_j[n] + 2\mu e_j[n]\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{a_j} \quad (14)$$

Squaring and expanding Eq. (14) to obtain cost function $\rho \in \mathbb{R}$ for aposteriori error,

$$\rho = e_j[n]^2 + 4\mu e_j[n]^2\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{a_j} + 4\mu^2 e_j{}^2[n]\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{a_j}\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{a_j} \quad (15)$$

Minimizing Eq. (15) with respect to μ,

$$\boldsymbol{\nabla\rho} = 4e_j[n]^2\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{a_j} + \boldsymbol{8}\mu e_j{}^2[n]\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{a_j}\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{a_j} = 0 \quad (16)$$

Where $\boldsymbol{\nabla\rho} \in \mathbb{R}$. Solving Eq. (16) for μ,

$$\mu = -\frac{1}{2\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{a_j}} \quad (17)$$

Substituting Eq. (17) in Eq. (11),

$$\boldsymbol{x}[\boldsymbol{n}+\boldsymbol{1}] = \boldsymbol{x}[\boldsymbol{n}] + \frac{1}{\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{a_j}}\left(b_j - \boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}]\right)\boldsymbol{a_j} \quad (18)$$

Since $\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{a_j}$ is a scalar, we can re-arrange it,

$$\boldsymbol{x}[\boldsymbol{n}+\boldsymbol{1}] = \boldsymbol{x}[\boldsymbol{n}] + \left(b_j - \boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}]\right)\frac{\boldsymbol{a_j}}{\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{a_j}} \quad (19)$$

Hence, we arrive at Kackzmarz's equation. Whereas, the equation for NLMS is [14] ,

$$\boldsymbol{w}[\boldsymbol{n}+\mathbf{1}] = \boldsymbol{w}[\boldsymbol{n}] + (d - \boldsymbol{x}^{\boldsymbol{T}}[\boldsymbol{n}]\boldsymbol{w}[\boldsymbol{n}])\frac{\boldsymbol{x}[\boldsymbol{n}]}{\boldsymbol{x}^{\boldsymbol{T}}[\boldsymbol{n}]\boldsymbol{x}[\boldsymbol{n}]} \tag{20}$$

Eq. (19) and Eq. (20) are identical except for the nomenclature. This difference of nomenclature is depicted in Figure 1 and illustrated in Table 1.

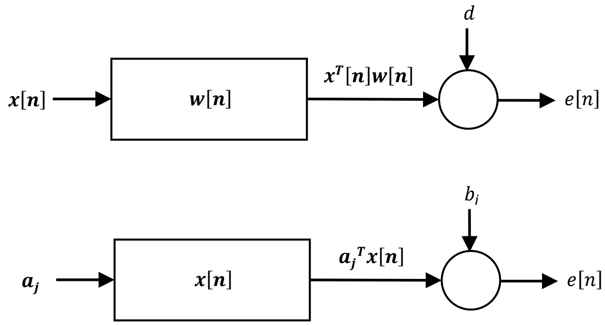

Figure 1. Comparison of NLMS and Kaczmarz algorithms

The box at the top in the Figure 1 generally implies a physical system, a smart antenna for example. Such a system is described by a system vector. The box computes the system output by forming the dot product of the input vector with the system vector. As the time progresses, input vector is delayed and again the dot product is computed with the system vector to form the output. This process, in signal processing community, is known by the name of convolution. The box is then called a filter. At each time step, the filter updates its system vector in an attempt to minimize the error between its output and desired output unless further system update brings no significant change in the error. In this way, a best possible solution with respect to the error is achieved in an adaptive fashion and so follows the term adaptive filter. Therefore, Kaczmarz algorithm can be thought of an adaptive filter. Other way round, NLMS adaptive filter can be viewed as an iterative linear system solver. Only caveat is that Kaczmarz algorithm operates on a restricted set of inputs that keep recurring whereas NLMS has no such limitations.

Table 1. Comparison of nomenclature of NLMS and Kaczmarz algorithms

Kaczmarz algorithm		NLMS algorithm	
a_j	j-th row of A matrix	$x[n]$	Input signal vector
$x[n]$	Solution	$w[n]$	Weight vector of the filter
$a_j{}^T x[n]$	Output	$x^T[n]w[n]$	Output vector
b_j	j-th component of b vector	d	Desired output

3. Proof of an alternate formula

Some authors prefer to write Kaczmarz equation as [15],

$$\boldsymbol{x}[\boldsymbol{n}+\mathbf{1}] = \boldsymbol{x}[\boldsymbol{n}] + \mu\left(b_j - \boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{x}[\boldsymbol{n}]\right)\frac{\boldsymbol{a_j}}{\boldsymbol{a_j}\boldsymbol{a_j}^{\boldsymbol{T}}} \quad (21)$$

An implication of this approach is that if we try to derive the optimal step-size for Eq. (21) such that it minimizes the aposteriori error criteria laid down in Eq. (12), the resulting value of step-size thus obtained for Eq. (21) is always equal to 1. We can understand this by observing the factor $\boldsymbol{a_j}^{\boldsymbol{T}}\boldsymbol{a_j}$ in the denominator of Kaczmarz's Equation. This factor already achieves the desired goal of optimization as in Eq. (19). Minimizing Eq. (21) with respect to μ will result in no new information. Therefore, Eq. (1) and Eq. (21) are identical in terms of optimality criteria laid down in Eq. (12).

4. Conclusion

In this paper, we have provided a simple linear algebraic proof of the relationship between Kaczmarz and NLMS algorithms which demonstrates that both algorithms are identical.

References

[1] T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.* 2009; 15: 262-278.

[2] H. Mansour and O. Yilmaz. A fast randomized Kaczmarz algorithm for sparse solutions of consistent linear systems. *CoRR.* 2013; 1305.3803.

[3] X. Chen and A. Powell. Almost Sure Convergence of the Kaczmarz Algorithm with Random Measurements. *Journal of Fourier Analysis and Applications.* 2012; 18: 1195-1214.

[4] Y. Cai, Y. Zhao, and Y. Tang. *Exponential Convergence of a Randomized Kaczmarz Algorithm with Relaxation*. Proceedings of the 2011 2nd International Congress on Computer Applications and Computational Science. 2012; 145: 467-473.

[5] D. Needell and J. A. Tropp. Paved with good intentions: Analysis of a randomized block Kaczmarz method. *Linear Algebra and its Applications*. 2014; 199-221.

[6] C. Popa, T. Preclik, H. Köstler, and U. Rüde. On Kaczmarz's projection iteration as a direct solver for linear least squares problems. *Linear Algebra and its Applications.* 2012; 436: 389-404.

[7] M. Haltmeier. Convergence analysis of a block iterative version of the loping Landweber–Kaczmarz iteration. *Nonlinear Analysis: Theory, Methods & Applications.* 2009; 71: e2912-e2919.

[8] S. Gaure. OLS with multiple high dimensional category variables. *Computational Statistics & Data Analysis.* 2013; 66: 8-18.

[9] M. Eslitzbichler, C. Pechstein, and R. Ramlau. An H1-Kaczmarz reconstructor for atmospheric tomography. *Journal of Inverse and Ill-Posed Problems.* 2013; 21: 431–450.

[10] L. Taoran, K. Tzu-Jen, I. David, C. N. Jonathan, and J. S. Gary. Adaptive Kaczmarz method for image reconstruction in electrical impedance tomography. *Physiological Measurement.* 2013; 34: 595.

[11] M. Rosensteiner and R. Ramlau. Kaczmarz algorithm for multiconjugated adaptive optics with laser guide stars. *Journal of the Optical Society of America.* 2013; 30: 1680-1686.

[12] Z. Zhu, K. Wahid, P. Babyn, D. Cooper, I. Pratt, and Y. Carter. Improved Compressed Sensing-Based Algorithm for Sparse-View CT Image Reconstruction. *Computational and Mathematical Methods in Medicine.* 2013.

[13] J. M. Elble, N. V. Sahinidis, and P. Vouzis. GPU computing with Kaczmarz's and other iterative algorithms for linear systems. *Parallel Comput.* 2010; 36: 215-231.

[14] B. Farhang-Boroujeny. Adaptive Filters: Theory and Applications. New York: John Wiley & Sons. 1998.

[15] E. Chong and S. Zak. An Introduction to Optimization. New York: John Wiley & Sons. 2013.

[16] P. Diniz. Adaptive Filtering: Algorithms and Practical Implementation. Springer, 2010.
[17] G. Strang. Computational Science and Engineering. Massachusetts: Wellesley-Cambridge Press. 2007.