G. M. Georgiou and K. Voigt, "Stochastic computation of moments, mean, variance, skewness, and kurtosis," *Electronics Letters,* Volume 51, Issue 9, 30 April 2015, p. 673 - 674.

Stochastic computation of moments, mean, variance, skewness, and kurtosis

George M. Georgiou and Kerstin Voigt

Stochastic computation of statistical moments and related quantities such as the mean, variance, skewness, and kurtosis, is performed with simple neural networks. The computed quantities can be used to estimate parameters of input data probability distributions, gauge normality of data, add useful features to the inputs, preprocess data, and for other applications. Such neural networks can be embedded in larger ones that perform signal processing or pattern recognition tasks. Convergence to the correct values is demonstrated with experiments.

Introduction: Statistical moments and related quantities such as mean, variance, skewness, and kurtosis have been used in pattern recognition, adaptive filtering, signal processing and neural networks, and in general are useful quantities in stochastic processes [1]. In this communication, these statistical quantities will be computed stochastically, using instantaneous gradient descent techniques that minimize the appropriate error functional. To the authors' knowledge, besides the mean, and that only incidentally, for example, in self-organizing maps (SOMs) [2] and related algorithms, the variance, skewness, kurtosis, and moments have not been computed stochastically. The standard formulas of computing these quantities require use of the number of input data vectors N, the sample size [3]. Computing them stochastically has the advantage that no knowledge of the number of input patterns is needed, and that they can be available for use even in environments with time-varying input statistics, which LMS algorithms are inherently able to do.

The mean and standard deviation, which is the square root of the variance, are often used to preprocess data before presenting them to a neural network, commonly to make each input component centered around the origin and have unit standard deviation. Uses of the skewness and kurtosis include gauging whether the underlying distribution is normal [4] and characterizing the sharpness of tuning curves in the brain [5]. Raw moments can be used in the *method of moments* to estimate the parameters of an assumed underlying probability distribution function (pdf) [6]. The kurtosis of the error signal, i.e. a mean-fourth cost function, has been used for the LMS algorithm [7].

We show that the optimal learning rate for the introduced LMS rules does not depend on the input data vector. The optimal learning rate is derived from two different perspectives. In the usual LMS algorithm there is such dependency which is accounted for in the Normalized LMS (NLMS) algorithm [8].

We performed experiments that show that the algorithms converge to accurate estimations of the various statistical quantities.

The rule for the mean: For input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$, where $\mathbf{x}_i \in \mathbb{R}^n$, consider the error functional F_1:

$$F_1 = \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{w}_1\|^2, \tag{1}$$

where $\mathbf{w}_1 \in \mathbb{R}^n$ is an adjustable weight vector. The symbol $\|\cdot\|$ indicates the Euclidean norm. The instantaneous gradient of F_1 with respect to $\mathbf{w}_1$ is

$$\nabla_{\mathbf{w}_1} F_1 = -\sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{w}_1). \tag{2}$$

Setting the gradient to zero, at equilibrium, the value $\mathbf{w}_1$ that minimizes F_1 is

$$\overline{\mathbf{w}}_1 = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i, \tag{3}$$

which is the mean μ, a well-known fact. For a given input vector $\mathbf{x}_i$, from (2), the stochastic, i.e. the online, as opposed to the batch method, gradient descent learning rule is

$$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) + \alpha(\mathbf{x}_i(n) - \mathbf{w}_1(n)). \tag{4}$$

Parameter α is the learning rate, a small positive constant. As usual, $\mathbf{w}_1$ is initialized to small random values. At convergence, $\mathbf{w}_1 = \mu$, the mean vector. Computing the mean using this rule does not require knowledge of the number of input vectors. The shape of the error function F_1, except in degenerate cases, is bowl-shaped with a single minimum at the mean. This learning rule is akin to the update rule of self-organizing maps (SOMs) [2]. In SOMs, during training each input vector is assigned to a winning neuron, hence the computed mean, or centroid as it is called in context of SOMs, is local to that specific neuron, and also temporal since in the course of training input vectors are assigned and de-assigned to a particular neuron. At convergence, however, the input vectors settle to specific neurons, and each weight vector of a neuron converges to the mean of the associated input vectors.

Central moments: The $k-$th central moment of component j of input vector $\mathbf{x}_i$ is defined as

$$m_k^j = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i^j - \overline{\mathbf{x}}^j)^k, \tag{5}$$

where $\overline{\mathbf{x}}^j$ is the mean. We can rewrite (5) in vector format:

$$\mathbf{m}_k = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \overline{\mathbf{x}})^{\cdot k}, \tag{6}$$

where the superscript $.k$ indicates element-wise exponentiation. All components are processed independently. There is no need to use new element-wise operations for the mean in (4) since vector subtraction and scalar multiplication are already element-wise operations. It is noted that $\mathbf{m}_1 = \mathbf{0}$, the first central moment. The second central moment $\mathbf{m}_2$ is a vector that has the variance σ^2 for each component. To derive the stochastic gradient rule on $\mathbf{w}_k^j$ that will converge central moment $\mathbf{m}_k^j$, we define a cost functional F_k^j:

$$F_k^j = \frac{1}{2}\sum_{i=1}^{N} \left((\mathbf{x}_i^j - \overline{\mathbf{x}}^j)^k - \mathbf{w}_k^j\right)^2 \tag{7}$$

The functional F_k^j is minimized when $\mathbf{w}_k^j$ equals the central moment $\mathbf{m}_k^j$. We substitute $\overline{\mathbf{x}}^j$ with the $\mathbf{w}_1$ in (4), the mean as is being computed. The partial derivative of F_k^j with respect to $\mathbf{w}_k^j$ is

$$\frac{\partial F_k^j}{\partial \mathbf{w}_k^j} = -\sum_{i=1}^{N} \left((\mathbf{x}_i^j - \overline{\mathbf{w}}_1^j)^k - \mathbf{w}_k^j\right). \tag{8}$$

When the partial derivatives for all j become zero, the weight vector $\overline{\mathbf{w}}_k = \mathbf{m}_k$, the $k-$th moment of the inputs. The online gradient descent rule takes the form

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) + \alpha((\mathbf{x}_i - \mathbf{w}_1(n))^{\cdot k} - \mathbf{w}_k(n)). \tag{9}$$

For each each input $\mathbf{x}_i$ that is presented, weight vectors $\mathbf{w}_1$ and $\mathbf{w}_k$ are being updated in parallel or sequentially using Equations 4 and 9, respectively. The learning rate α may be chosen to be the same or be different for the two equations.

Raw moments, as opposed to central moments, can be computed with the rule of Equation 9 and setting $\mathbf{w}_1$ to the zero vector:

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) + \alpha((\mathbf{x}_i)^{\cdot k} - \mathbf{w}_k(n)). \tag{10}$$

Skewness and kurtosis: Skewness is a measure of the bias of the data around the mean: positive implies data are spread to the right of the mean and negative to the left. The sample skewness γ_1 for component j of $\mathbf{x}_i$, the single random variable $\mathbf{x}_i^j$, is defined as

$$\gamma_1 = \frac{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i^j - \overline{\mathbf{x}}^j)^3}{(\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i^j - \overline{\mathbf{x}}^j)^2)^{3/2}} = \frac{\mathbf{m}_3^j}{(\mathbf{m}_2^j)^{3/2}}. \tag{11}$$

Central moments $\mathbf{m}_3^j$ and $\mathbf{m}_2^j$, the variance, can be computed stochastically using Equations 4 and 9, and hence skewness can be computed stochastically.

Kurtosis γ_2 is a measure of "peaketness", i.e. how flat or how peaked the data distribution is. For $\mathbf{x}_i^j$, it is defined as

$$\gamma_2 = \frac{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i^j - \overline{\mathbf{x}}^j)^4}{(\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i^j - \overline{\mathbf{x}}^j)^2)^2} = \frac{\mathbf{m}_4^j}{(\mathbf{m}_2^j)^2}. \tag{12}$$

As for skewness, kurtosis can be computed stochastically using Equations 4 and 9 to compute the moments $\mathbf{m}_4^j$ and $\mathbf{m}_2^j$.

Normalization rules: The stability and convergence properties of LMS can improved by using the Normalized LMS (NLMS) [8], which uses a variable learning rate $\alpha(n) = \frac{1}{\mathbf{x}(n)^T \mathbf{x}(n)+\gamma}$, where γ is a small positive real constant added to prevent division by zero. [9] In a like manner we will derive the optimal value of $\alpha(n)$ in the learning rules of (4) and (9). NLMS can be derived from different vantage points. We will use the minimization of the *a posteriori* error [9], i.e. the to find optimal $\alpha(n)$ that will minimizes the square of the error $(\mathbf{e}^j(n))^2$ after the weight vector $\mathbf{w}_k$ has been updated:

$$\mathbf{e}^j(n) = (\mathbf{x}_i^j(n) - \mathbf{w}_1^j(n))^k - \mathbf{w}_k^j(n+1). \tag{13}$$

Substituting $\mathbf{w}_k^j(n+1)$ from (9) and omitting time step n,

$$\mathbf{e}^j = (\mathbf{x}_i^j - \mathbf{w}_1^j)^k - (\mathbf{w}_k^j + \alpha((\mathbf{x}_i^j - \mathbf{w}_1^j)^k - \mathbf{w}_k^j)) \tag{14}$$

$$\mathbf{e}^j = ((\mathbf{x}_i^j - \mathbf{w}_1^j)^k - \mathbf{w}_k^j)(1-\alpha) \tag{15}$$

The partial derivative of $(\mathbf{e}^j(n))^2$ with respect to $\alpha(n)$ is zero when $\alpha(n) = 1$. This implies that the optimal learning rate α, unlike the usual LMS algorithm, does not depend on input vector $\mathbf{x}(n)$.

The same result can be arrived at by solving the analogous to the NMLS constraint optimization problem, that minimize the square of the Euclidean norm of the weight change under a constraint:

$$\text{Minimize } \|\mathbf{w}_k^j(n+1) - \mathbf{w}_k^j(n\|^2 \tag{16}$$

subject to $\mathbf{w}_k^j(n+1) = \mathbf{x}_i^j(n)$. Using component j of (9) and the constraint, the weight change is written as follows:

$$\mathbf{w}_k^j(n+1) - \mathbf{w}_k^j(n) = \alpha(\mathbf{x}_i^j(n) - \mathbf{w}_k^j(n)) = \alpha(\mathbf{w}_k^j(n+1) - \mathbf{w}_k^j(n)). \tag{17}$$

Again, it is concluded, that $\alpha(n) = 1$, independent of $\mathbf{x}(n)$. This result is applicable to the update rule of SOMs, and could imply that these algorithms are less sensitive to sudden changes in the magnitude of the inputs as is the case in the usual LMS algorithm which is stabilized with the NLMS.

Results: As test cases we present two runs that show the convergence behavior of the algorithms in computing the mean (Equation 4) and central moments $k = 2$ (variance), 3, and 4 (Equation 9) in Fig. 1; in Fig. 2, the mean (Equation 4), variance (Equation 9), skewness and kurtosis are shown. The latter two quantities are computed using Equation 9 to compute the appropriate central moments and Equations 11 and 12, respectively. The horizontal lines are the corresponding computed values via the statistical formulas. In each case, 100 sample points were drawn from a gaussian distribution with mean 0.5 and variance 1.44. A fixed learning rate $\alpha = 0.001$ was used. As it can be seen, within about 70 epochs, the values converged to the computed equivalents. Convergence of the central moments, of course, depends on the convergence of the mean. Skewness was the slowest to converge after the mean had converged.

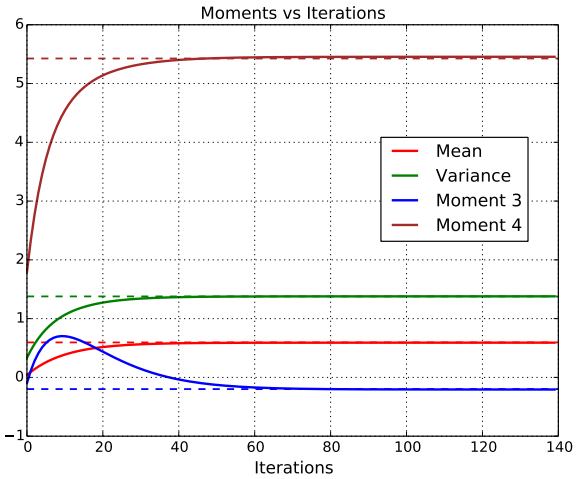

Fig. 1. Convergence of the moments (mean, variance, moment 3, moment 4)

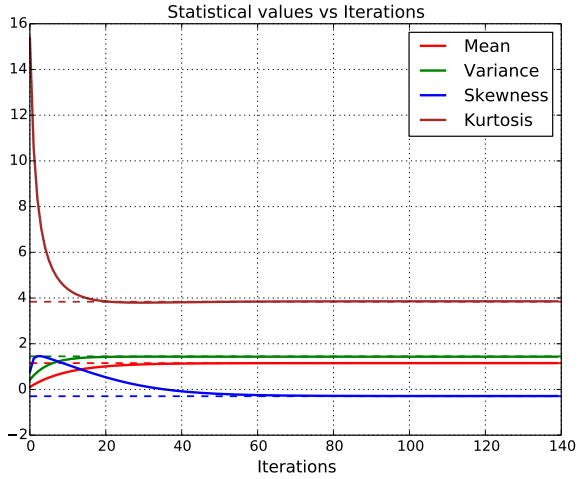

Fig. 2. Convergence of mean, variance, skewness, and kurtosis)

Conclusion: Although basic statistical quantities such as the mean, variance, skewness. kurtosis, and moments are of importance in pattern recognition, signal processing, neural networks and related fields, they do not seem to have been computed stochastically, as weights in a gradient descent process. The closest to these computations is that for the mean in SOMs. The derived rules allow these statistical quantities to be stochastically computed, and thus be read, interpreted and used in real time and in time-varying environments. The experiments have shown the efficacy of the rules.

References

1 Papoulis, A.: 'Probability, Random Variables and Stochastic Processes' (McGraw-Hill, 1991), 3rd edition
2 Kohonen, T.: 'Self-organizing maps' (Springer, 2001)
3 Sheskin, D.: 'Handbook of Parametric and Nonparametric Statistical Procedures: Third Edition' (CRC Press, 2003)
4 Sanei, S.: 'Adaptive Processing of Brain Signals' (Wiley, 2013)
5 Samonds, J., Potetz, B., and Lee, T.: 'Sample skewness as a statistical measurement of neuronal tuning sharpness', *Neural Computation*, 2014, **26**, (5), pp. 860–906
6 Ayyub, B. and McCuen, R.: 'Probability, Statistics, and Reliability for Engineers and Scientists, Third Edition' (Taylor & Francis, 2011)
7 Tanrikulu, O. and Constantinides, A.: 'Least-mean kurtosis: a novel higher-order statistics based adaptive filtering algorithm', *Electronics Letters*, 1994, **30**, (3), pp. 189–190
8 Haykin, S.: 'Adaptive Filter Theory (3rd Ed.)' (Prentice-Hall, Inc., 1996)
9 Farhang-Boroujeny, B.: 'Adaptive filters : theory and applications' (Wiley, 1998)

George M. Georgiou and Kerstin Voigt (*School of Computer Science and Engineering, California State University, San Bernardino, San Bernardino, CA 92407-2393, USA*)

E-mail: georgiou@csusb.edu