Recursive Mean and Stdev formulas of O(1) and one-step-back storage requirements for Normal distributions

Harris V. Georgiou (MSc,PhD)

Short Communication

Abstract—Updating basic running statistics often requires keeping track of cumulants as the data set grows or evolves through time. Common approaches that are based on running cumulants are inefficient for intensive-processing or Big data contexts, since they introduce running windows to the original data set, in order to avoid arithmetic overflows. This work formulates recursive estimators for arithmetic mean and (sample) standard deviation of Normal distributions, which are of minimum storage complexity (one-step-back), arithmetically robust for error resiliency and inherently parallelizable at the lowest possible level (arithmetic operators).

Index Terms—online statistics, parallel processing, Big data

1 Introduction

In massive data sets usually found in streaming or Big data tasks, it is often that updates to basic statistics such as the arithmetic mean and (sample) standard deviation need to be performed online, with minimum storage requirements for previous history, minimum number of calculations, robust arithmetic to avoid cumulative errors and inherently parallelizable design.

The most common approaches, such as keeping track of cumulants as the data set grows or evolves, are sensitive to arithmetic overflow and require some bounding 'window', i.e., a running frame or an exponentially decreasing weighting profile. Both these approaches are inefficient for intensive-processing or Big data contexts, since they introduce hard or soft filtering, respectively, to the original data set, essentially forcing the calculations upon a smaller subset rather than the complete data set.

The goal of this work is to formulate recursive estimators for arithmetic mean and (sample) standard deviation of Normal distributions, which are of minimum storage complexity (one-step-back), arithmetically robust for error resiliency and inherently parallelizable at the lowest possible level (arithmetic operators).

2 Recursive statistics

Theorem 1 (Recursive Generalized Means (RGM))**.** *Let:*

$$\bar{x}_n(k) = \left(\frac{1}{n} \cdot \sum_{i=1}^{n} x_i^k \right)^{\frac{1}{k}} \tag{1}$$

the Generalized Means (GM) [2] of order $k \in \mathbf{Z}$ of a data set $x_i \in \mathbf{R}$. Then Eq.1 can be calculated recursively via:

$$\bar{x}_{n+1}(k) = \left(\frac{n}{n+1} \cdot \bar{x}_n(k)^k + \frac{1}{n+1} \cdot x_{n+1}^k \right)^{\frac{1}{k}} \tag{2}$$

where x_{n+1}^k is an additional data value, $n \geq 1$ and $\bar{x}_1(k) = \sqrt[k]{x_1^k} = x_1$.

Proof: From Eq.1 follows that:

$$\begin{aligned} \bar{x}_{n+1}(k)^k &= \frac{1}{n+1} \cdot \sum_{i=1}^{n+1} x_i^k \\ &= \frac{1}{n+1} \cdot \sum_{i=1}^{n} x_i^k + \frac{x_{n+1}^k}{n+1} \\ &= \frac{n}{n+1} \cdot \left(\frac{1}{n} \cdot \sum_{i=1}^{n} x_i^k \right) + \frac{x_{n+1}^k}{n+1} \\ &= \frac{n}{n+1} \cdot \bar{x}_n(k)^k + \frac{x_{n+1}^k}{n+1} \end{aligned}$$

which derives Eq.2. □

Corollary 2 (Recursive Arithmetic Mean (RAM))**.** *Let:*

$$m_n = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i \tag{3}$$

the arithmetic mean value (Normal distribution) [1] of a data set $x_i \in \mathbf{R}$. Then Eq.3 can be calculated recursively via:

$$m_{n+1} = \frac{n}{n+1} \cdot m_n + \frac{x_{n+1}}{n+1} \tag{4}$$

where x_{n+1} is an additional data value, $n \geq 1$ and $m_1 = x_1$.

Proof: Eq.4 is directly derived from Eq.2 for $k = 1$. □

Corollary 3 (Recursive Standard Deviation (RSD))**.** *Let:*

$$s_n = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - m_n)^2} \tag{5}$$

• *Harris Georgiou is a post-doctorate research associate with the Data Science Lab (datastories.org) at the Department of Informatics, University of Piraeus (UniPi), Athens, Greece – E-mail: hgeorgiou@unipi.gr.*

Last updated: October 30, 2017.

the sample standard deviation value (Normal distribution) [1] of a data set $x_i \in \mathbf{R}$. Then Eq.5 can be calculated recursively via:

$$s_{n+1} = \sqrt{\frac{n-1}{n} \cdot s_n^2 + \frac{(x_{n+1} - m_n)^2}{n}} \tag{6}$$

where x_{n+1} is an additional data value, $n \geq 1$ and $s_1 = 0$.

Proof: Following the same outline for the proof of Eq.2 for $k = 2$ and data set values $y_i = x_i - m_n$, it follows that for sample standard deviation (i.e., dividing by $n-1$):

$$\begin{aligned} {s_{n+1}}^2 &= \frac{1}{(n-1)+1} \cdot \sum_{i=1}^{n+1} (x_i - m_n)^2 \\ &= \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - m_n)^2 + \frac{(x_{n+1} - m_n)^2}{n} \\ &= \frac{n-1}{n} \cdot \left(\frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - m_n)^2 \right) + \frac{(x_{n+1} - m_n)^2}{n} \\ &= \frac{n-1}{n} \cdot {s_n}^2 + \frac{(x_{n+1} - m_n)^2}{n} \end{aligned}$$

which derives Eq.6. □

3 Experimental Evaluation

For the actual assessment of the estimation accuracy when using the recursive formulas of Eq.4 and Eq.6, a simple program in Matlab/Octave was used. The idea is to use a standard Normal distribution ($m = 0$, $s = 1$) to create randomized sets of increasing size and track the errors between the recursive versus the true (full-set) estimations of arithmetic means and sample standard deviations. The listing of the program is illustrated in Algorithm 1.

Algorithm 1 Matlab/Octave code for testing arithmetic accuracy of RAM and RSD.

```
clear all;
R=[1 0 0];
Nmax=100000;
for N=1000:1000:Nmax
    X=randn(N,1);    % norm.distr.(m=0,s=1)
    m=X(1);  s=0;
    for k=2:length(X)
        m=(k-1)/k*m+X(k)/k;
        s=(k-2)/(k-1)*s+(X(k)-m)^2/(k-1);
    end;
    c=[N m-mean(X) sqrt(s)-std(X)];
    R=[R; c];  disp(c);
end;
figure(1);
plotyy(R(:,1),R(:,2),R(:,1),R(:,3));
title('Accuracy: Recursive Mean and Stdev');
xlabel('sample size (N)');
ylabel('error: (recursive)-(true)');
legend('RAM-AM','RSD-SD',...
       'Location','SouthEast');
```

Figure 1 illustrates a test run of Algorithm 1 for sample set sizes of $N = 10^3, ..., 10^5$. The test was conducted in double precision arithmetic (default), which provides arithmetic accuracy of $eps \simeq 5.5 \cdot 10^{-15}$. From the plot it is clear that the recursive estimation of the arithmetic mean (RAM) remains within the region of eps as expected, since Eq.4 is an unbiased estimator of first-order arithmetic for the true arithmetic mean (AM). On the other hand, the plot illustrates that the recursive estimation of the sample standard deviation (RSD) is significantly larger than eps, since Eq.6 is a biased estimator (depends on m_n) of second-order arithmetic of the true sample standard deviation (SD), but asymptotically approaches zero as the data set size increases.

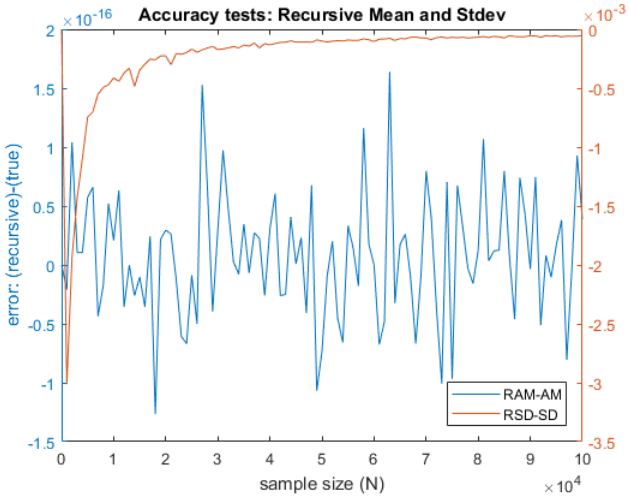

Fig. 1. Experimental evaluation of RAM and RSD ($eps \simeq 5.5 \cdot 10^{-15}$)

4 Discussion

As mentioned above, the goal of this work is to formulate recursive estimators that are of minimum storage complexity, arithmetically robust and inherently parallelizable.

Both Eq.4 and Eq.6 are formulated in a way that minimizes the arithmetic errors and the possibility of arithmetic overflow in case of excessively large data set size or number of iterations. In practice, this depends almost entirely on the two divisions in each formula, i.e., first calculate the fraction and then multiply it with the previous estimation (see Algorithm 1).

The storage required by these estimators is clearly one-step-back. This means that, no matter what the data set size is or (equivalently) how many times the recursion is executed, Eq.4 and Eq.6 never uses more than the previous estimation of m_n and s_n, respectively, plus the newly arrived data value x_{n+1}. This is particularly useful in extremely large data sets for avoiding arithmetic overflows, e.g. if the calculations were based on successive updates to $\sum_{i=1}^{n}(x_i)$ and $\sum_{i=1}^{n}(x_i - m_n)^2$, as these are usually implemented iteratively.

Finally, it should be noted that Eq.4 and Eq.6 are inherently parallelizable with a fixed number of calculations independently of data set size N, i.e., exhibit algorithmic complexity O(1). Additionally, even Eq.6 can be implemented using only additions and multiplications (keep s_n^2 instead of s_n), as it is the case for Eq.4. This means that these

formulations are not only parallel-ready but, morevover, implementable in DSP or GPU hardware, which provide massively parallel, simple-arithmetic processing cores in the sense of *single instruction multiple data* (SIMD).

REFERENCES

[1] M.R. Spiegel, J. Schiller, R.A. Srinivasan, *Probability and Statistics*, 3rd/Ed. (McGraw-Hill: 2009).
[2] S. Theodoridis, K. Koutroumbas, *Pattern Recognition*, 4th/Ed. (Academic Press: 2009).

Harris Georgiou is a MSc, PhD and post-doctorate research fellow with the Dept. of Informatics & Telecommunications, National Kapodistrian University of Athens (NKUA/UoA), Greece and currently a post-doctorate research associate with the Data Science Lab (datastories.org) at the Dept. of Informatics, University of Piraeus (UniPi), Athens, Greece. His main research areas include Machine Learning, Pattern Recognition, Signal Processing, Medical Imaging, Game Theory and is currently working on Big Data Analytics, Data Mining and Mobility Patterns.