Dynamic Algorithms: Arithmetic, Geometric, Harmonic, and Power Means

Caleb Princewill Nwokocha

Department of Computer Science, The University of Manitoba

August 24, 2022

Acknowledgements

To my family and professors, I am very grateful for your support before and during this writing.

Abstract

Arithmetic, Geometric, and Harmonic means are the well-known Pythagorean means [1, 2, 4, 3]. Power mean, also known as Hölder mean, is a generalized mean for the Pythagorean means [5]. Formulas of these means have been used to derive brute-force algorithms that run at polynomial time and space complexities $O(n)$, but these brute-force algorithms cost more time and space when compared to their dynamic counterpart. I show that dynamic algorithms for computing the Pythagorean and the power means do exist; in fact, proofs of the dynamic algorithms are written, and a weblink to Java implementation of the dynamic algorithms is provided at the end of section 2. The dynamic mean algorithms, just like the brute-force mean algorithms are derived from mean formulas found in statistics resources like Moore et al, 1998; Stan, 2009; Chou, 1969. A good review of the formulas would be helpful for understanding the proofs, which shows that by using dynamic programming, the mean of a variable can be computed at constant time and space complexities $O(c)$, less costly than the brute-force algorithms. Utility of the dynamic algorithms may be found in applications that process instant mathematical mean of very large data with respect to the time each datum is acquired. An example application is provided in the introduction section, where Bob planned to instantly calculate the mean number of oranges that he receives each day from Alice.

Keywords: Dynamic algorithm, dynamic programming, arithmetic mean, geometric mean, harmonic mean, power mean.

1. Introduction

The objective of calculating the mean is finding an average of a variable. Consider a case where Alice and Bob are friends, and every day, Alice gives Bob x_t orange(s). The symbol x_t is a variable that represents the number of orange(s) that Bob receives from Alice each day. Suppose that on day 1, Alice gave Bob 2 oranges; on day 2 Alice gave Bob 6 oranges; and on day 3 Alice gave Bob only 1 orange. If the t subscript of x_t represent the day that Bob received orange(s), then this Alice and Bob case can be stated statistically as the following:

When $t = 1$, $x_t = 2$

When $t = 2$, $x_t = 6$

When $t = 3$, $x_t = 1$

The case can be stated even more succinctly as the following:

$$[x_{t=1} = 2, x_{t=2} = 6, x_{t=3} = 1] \rightarrow O(3) \text{ space}$$

Assuming that Bob planned to calculate the arithmetic mean of number of orange(s) he receives each day from Alice. Bob will typically do his calculation by the following brute-force mean algorithm:

$$\text{On day } t = 1\text{: } \bar{x}_{t=1} = \frac{x_{t=1}}{t=1} = \frac{2}{1} = 2 \ \rightarrow O(c) \text{ time}$$

The symbol $\bar{x}_t$ represents the mean on day t.

$$\text{On day } t = 2\text{: } \bar{x}_{t=2} = \frac{x_{t=1}+x_{t=2}}{t=2} = \frac{2+6}{2} = 4 \rightarrow O(c) \text{ time}$$

$$\text{On day } t = 3\text{: } \bar{x}_{t=3} = \frac{x_{t=1}+x_{t=2}+x_{t=3}}{t=3} = \frac{2+6+1}{3} = 3 \rightarrow O(c) \text{ time}$$

Observe that on the second day, Bob must remember $x_{t=1}$ to calculate $\bar{x}_{t=2}$ on day 2. Also, Bob must sum $x_{t=1}$ and $x_{t=2}$ as part of his calculation of $\bar{x}_{t=2}$ on day 2. Again, On the third day, Bob must remember $x_{t=1}$ and $x_{t=2}$, and sum $x_{t=1}$, $x_{t=2}$ and $x_{t=3}$ during his calculation of $\bar{x}_{t=3}$ on day 3. In general, Bob must remember $x_{t=1} \cdots x_{t=n}$ and sum $x_{t=1} \cdots x_{t=n+1}$ in the process of calculating $\bar{x}_{t=n+1}$ on day $n + 1$. Here n represents the last day t before day $n + 1$ that Bob received orange(s) from Alice. For example, if $n = 3$ then day n is equal to day 3, and day $n + 1$ is equal to day 4.

Analysis shows that if a computer was performing the same calculation as Bob, the computer would require $O(1) + O(1) + O(1) = O(3)$ space for storage of $x_{t=1}, x_{t=2}, x_{t=3}$. For Alice and Bob case, this $x_{t=1}, x_{t=2}, x_{t=3}$ is equal to $x_{t=1} \cdots x_{t=n}$ if $t = n = 3$. Hence, the brute-force mean algorithm require $O(n)$ for storage of $x_{t=1} \cdots x_{t=n}$. It also require $O(n + 1)$ time for summation of $x_1 \cdots x_{n+1}$. If Bob knew an algorithm that require only $O(c)$ time and space, he would be better off using that algorithm for his calculation, as such algorithm would save Bob time and memory for use on other activities. In the case of a computer dealing with very large data $x_t \cdots x_{t+k}$, where each datum is available at certain times $t \cdots t + k$, such algorithm will not require stored data $x_t \cdots x_{t+k-1}$ of past times $t \cdots t + k - 1$ to calculate the current mean $\bar{x}_{t+k}$ at time $t + k$. It will only require the current datum x_{t+k} at current time $t + k$ and the stored mean of past data $\bar{x}_{t+k-1}$ for calculation of $\bar{x}_{t+k}$. Therefore, such algorithm run at $O(c)$ time and space.

In the proceeding sections, I show dynamic algorithms that calculate mathematical means at $O(c)$ time and space complexities.

2. Dynamic Arithmetic Mean

It is true that the arithmetic mean of a_t and b_{t+1}, where $t = 2$, is

$$\bar{x}_{t+1} = \frac{t\bar{x}_t + x_{t+1}}{t+1} = \frac{2a+b}{3}$$

2.1 Proof.

$$\frac{\frac{x_1 + \cdots + x_t}{t} + x_{t+1}}{t+1} \neq \frac{x_1 + \cdots + x_{t+1}}{t+1}$$

$$\frac{x_1 + \cdots + x_t}{t} + x_{t+1} \neq x_1 + \cdots + x_{t+1}$$

$$t\left(\frac{x_1 + \cdots + x_t}{t}\right) + x_{t+1} = x_1 + \cdots + x_{t+1}$$

$$\therefore \frac{t\left(\frac{x_1 + \cdots + x_t}{t}\right) + x_{t+1}}{t+1} = \frac{x_1 + \cdots + x_{t+1}}{t+1}$$

$$\left(\frac{t}{t+1}\right)\left(\frac{x_1 + \cdots + x_t}{t}\right) + \frac{x_{t+1}}{t+1} = \frac{x_1 + \cdots + x_{t+1}}{t+1}$$

$$\left(\frac{t}{t+1}\right)\left(\frac{x_1 + \cdots + x_t}{t}\right) + \frac{x_{t+1}}{t+1} = \left(\frac{t}{t+1}\right)\left(\frac{1}{t}\sum_{i=1}^{t} x_i\right) + \frac{x_{t+1}}{t+1}$$

$$\left(\frac{t}{t+1}\right)\left(\frac{1}{t}\sum_{i=1}^{t} x_i\right) + \frac{x_{t+1}}{t+1} = \frac{1}{t+1}\sum_{i=1}^{t+1} x_i$$

$$\frac{t}{t+1}\bar{x}_t + \frac{x_{t+1}}{t+1} = \bar{x}_{t+1}$$

$$\bar{x}_{t+1} = \frac{t\bar{x}_t + x_{t+1}}{t+1}$$

$$\bar{x}_{t+k} = \frac{1}{t+k}\left[\left(\sum_{i=t+1}^{t+k} x_i\right) + t\bar{x}_t\right]$$

2.2 Algorithm.

In this subsection, I show the algorithms and complexities of the brute-force and dynamic means.

The brute-force algorithm is the following:

bruteForceArithmeticMean (data): $\longrightarrow$ $O(n)$ space

sum $=$ 0.0

for each datum in data: sum $+=$ datum $\longrightarrow$ $O(n)$ time

arithmeticMean $=$ sum / length(data)

return arithmeticMean

Below is the dynamic algorithm:

dynamicArithmeticMean (time, datum): $\longrightarrow$ $O(c)$ space

$\text{currentArithmeticMean} = \frac{((\text{time}-1) * \text{previousArithmeticMean}) + \text{datum}}{\text{time}}$ $O(c)$ time

return currentArithmeticMean

There need to be a datum generator function called $\text{generateDatum}(t)$. Below is an algorithm for the datum generator:

$\text{generateDatum}(t)$:

$\text{data} = x_1 \cdots x_k$

return t index of data

The $\text{generateDatum}(t)$ algorithm provides datum at each time t for use by the brute-force and dynamic algorithms.

Finally, let there be a main algorithm that call the above algorithms:

main ():

initialize data

for range $t = 1$ to $t \leq n$: add generateDatum(t) to index t of data bruteForceArithmeticMean (data)	$O(n^2)$ time and $O(n)$ space
for range $t = 1$ to $t \leq n$: datum = generateDatum(t); dynamicArithmeticMean (t, datum)	$O(n)$ time and $O(c)$ space

It is shown that the dynamicArithmeticMean algorithm is more efficient than the bruteForceArithmeticMean. Similar algorithms with minor changes can be used to compute the geometric, harmonic, and power means shown in section 3, 4, and 5. The dynamic algorithm remain more efficient for geometric, harmonic, and power means.

Java implementation of these algorithms can be found at *github.com/calebnwokocha/Dynamic-Means*.

3. Dynamic Geometric Mean

It is true that the geometric mean of a_t and b_{t+1}, where $t = 2$, is

$$\bar{x}_{t+1} = (x_{t+1}(\bar{x}_t)^t)^{\frac{1}{t+1}} = (b(a)^2)^{\frac{1}{3}}$$

3.1 Proof.

$$\left((x_1 \cdots x_t)^{\frac{1}{t}} \cdot x_{t+1}\right)^{\frac{1}{t+1}} \neq (x_1 \cdots x_{t+1})^{\frac{1}{t+1}}$$

$$(x_1 \cdots x_t)^{\frac{1}{t}} \cdot x_{t+1} \neq x_1 \cdots x_{t+1}$$

$$\left[(x_1 \cdots x_t)^{\frac{1}{t}}\right]^t \cdot x_{t+1} = x_1 \cdots x_{t+1}$$

$$\therefore \left(\left[(x_1 \cdots x_t)^{\frac{1}{t}}\right]^t \cdot x_{t+1}\right)^{\frac{1}{t+1}} = (x_1 \cdots x_{t+1})^{\frac{1}{t+1}}$$

$$\left(\left[\left(\prod_{i=1}^{t} x_i\right)^{\frac{1}{t}}\right]^t \cdot x_{t+1}\right)^{\frac{1}{t+1}} = \left(\prod_{i=1}^{t+1} x_i\right)^{\frac{1}{t+1}}$$

$$((\bar{x}_t)^t \cdot x_{t+1})^{\frac{1}{t+1}} = \bar{x}_{t+1}$$

$$\bar{x}_{t+1} = (x_{t+1}(\bar{x}_t)^t)^{\frac{1}{t+1}}$$

$$\bar{x}_{t+k} = \left((\bar{x}_t)^t \prod_{i=t+1}^{t+k} x_i\right)^{\frac{1}{t+k}}$$

4. Dynamic Harmonic Mean

It is true that the harmonic mean of a_t and b_{t+1}, where $t = 2$, is

$$\bar{x}_{t+1} = t + 1\left(\frac{t}{\bar{x}_t} + \frac{1}{x_{t+1}}\right)^{-1} = 3\left(\frac{2}{a} + \frac{1}{b}\right)^{-1}$$

4.1 Proof.

$$\frac{t+1}{\frac{t}{\frac{1}{x_1} + \cdots + \frac{1}{x_t}} + \frac{1}{x_{t+1}}} \neq \frac{t+1}{\frac{1}{x_1} + \cdots + \frac{1}{x_{t+1}}}$$

$$\frac{t}{\frac{1}{x_1} + \cdots + \frac{1}{x_t}} + \frac{1}{x_{t+1}} \neq \frac{1}{x_1} + \cdots + \frac{1}{x_{t+1}}$$

$$\left[\left(\frac{t}{\frac{1}{x_1} + \cdots + \frac{1}{x_t}}\right)\left(\frac{1}{t}\right)\right]^{-1} + \frac{1}{x_{t+1}} = \frac{1}{x_1} + \cdots + \frac{1}{x_{t+1}}$$

$$\therefore \frac{t+1}{\left[\left(\frac{t}{\frac{1}{x_1} + \cdots + \frac{1}{x_t}}\right)\left(\frac{1}{t}\right)\right]^{-1} + \frac{1}{x_{t+1}}} = \frac{t+1}{\frac{1}{x_1} + \cdots + \frac{1}{x_{t+1}}}$$

$$t + 1\left(\left[t\left(\sum_{i=1}^{t} \frac{1}{x_i}\right)^{-1}\left(\frac{1}{t}\right)\right]^{-1} + \frac{1}{x_{t+1}}\right)^{-1} = t + 1\left(\sum_{i=1}^{t+1} \frac{1}{x_i}\right)^{-1}$$

$$t + 1\left(\left[\bar{x}_t\left(\frac{1}{t}\right)\right]^{-1} + \frac{1}{x_{t+1}}\right)^{-1} = \bar{x}_{t+1}$$

$$\bar{x}_{t+1} = t + 1\left(\frac{t}{\bar{x}_t} + \frac{1}{x_{t+1}}\right)^{-1}$$

$$\bar{x}_{t+k} = t + k\left(\frac{t}{\bar{x}_t} + \sum_{i=t+1}^{t+k} \frac{1}{x_i}\right)^{-1}$$

5. Dynamic Power Mean

It is true that the power mean of a_t and b_{t+1}, where $t = 2$ and $m = 1$, is

$$\bar{x}_{t+1} = \left(\frac{\bar{x}_t^m t + x_{t+1}^m}{t+1}\right)^{\frac{1}{m}} = \left(\frac{(a^1 \cdot 2) + b^1}{3}\right)^{\frac{1}{1}}$$

5.1 Proof.

$$\left(\frac{\left(\frac{x_1^m + \cdots + x_t^m}{t}\right)^{\frac{1}{m}} + x_{t+1}^m}{t+1}\right)^{\frac{1}{m}} \neq \left(\frac{x_1^m + \cdots + x_{t+1}^m}{t+1}\right)^{\frac{1}{m}}$$

$$\frac{\left(\frac{x_1^m + \cdots + x_t^m}{t}\right)^{\frac{1}{m}} + x_{t+1}^m}{t+1} \neq \frac{x_1^m + \cdots + x_{t+1}^m}{t+1}$$

$$\left(\frac{x_1^m + \cdots + x_t^m}{t}\right)^{\frac{1}{m}} + x_{t+1}^m \neq x_1^m + \cdots + x_{t+1}^m$$

$$t\left(\left[\left(\frac{x_1^m + \cdots + x_t^m}{t}\right)^{\frac{1}{m}}\right]^m\right) + x_{t+1}^m = x_1^m + \cdots + x_{t+1}^m$$

$$\therefore \left(\frac{t\left(\left[\left(\frac{x_1^m + \cdots + x_t^m}{t}\right)^{\frac{1}{m}}\right]^m\right) + x_{t+1}^m}{t+1}\right)^{\frac{1}{m}} = \left(\frac{x_1^m + \cdots + x_{t+1}^m}{t+1}\right)^{\frac{1}{m}}$$

$$\left(\left[t\left(\left[\left(\frac{1}{t}\sum_{i=1}^{t} x_i^m\right)^{\frac{1}{m}}\right]^m\right) + x_{t+1}^m\right] \cdot (t+1)^{-1}\right)^{\frac{1}{m}} = \left(\frac{1}{t+1}\sum_{i=1}^{t+1} x_i^m\right)^{\frac{1}{m}}$$

$$\left([\bar{x}_t^m t + x_{t+1}^m] \cdot (t+1)^{-1}\right)^{\frac{1}{m}} = \bar{x}_{t+1}$$

$$\bar{x}_{t+1} = \left(\frac{\bar{x}_t^m t + x_{t+1}^m}{t+1}\right)^{\frac{1}{m}}$$

$$\bar{x}_{t+k} = \left(\frac{1}{t+k}\left[\left(\sum_{i=t+1}^{t+k} x_i^m\right) + \bar{x}_t^m t\right]\right)^{\frac{1}{m}}$$

6. Conclusion

Dynamic algorithms are more efficient than brute-force algorithms for computing Pythagorean and generalized power means of increasing data.

References

[1] Moore, D. S., McCabe, G. P., & Craig, B. A. (2009). Looking at Data – Distributions. In *Introduction to the practice of statistics.* Seventh edition. New York: W.H. Freeman.

[2] Geometric Mean. (2021, July 28). https://stats.libretexts.org/@go/page/4551

[3] *Statistical Analysis*, Ya-lun Chou, Holt International, 1969, ISBN 0030730953.

[4] http://www.scientificlib.com/en/Mathematics/Analysis/HarmonicMean.html

[5] Sykora, Stan. (2009). Mathematical Means and Averages: Basic Properties. 10.3247/SL3Math09.001.