Are you lying with statistics? Pitfalls to avoid when summarizing normalized results.

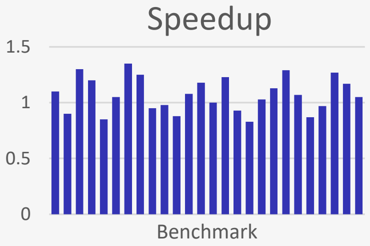
James Stevens

April 29, 2019

Motivation

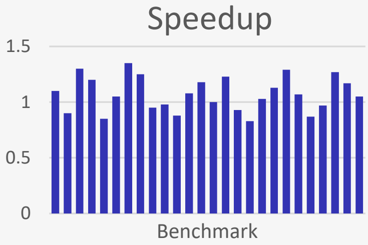
In scientific computing and high performance computing, we often produce sets of **normalized values**

- Speedups
- Changes in throughput rate
- Changes in algorithm data usage



Motivation

- Sometimes it makes sense to summarize this data with a single value
- How should we do this?
- Papers addressing this question:
  - [Fleming and Wallace(1986)]
  - [Smith(1988)]



Motivating example

	Processor time (speedup vs. **X**)		
Benchmark	**X**	**Y**	**Z**
B_1	20 (1.00)	10 (2.00)	40 (0.50)
B_2	30 (1.00)	60 (0.50)	15 (2.00)

- How do the processors compare in terms of speedup?

Table modified from [Fleming and Wallace(1986)]

Motivating example

	Processor time (speedup vs. **X**)		
Benchmark	**X**	**Y**	**Z**
B_1	20 (1.00)	10 (2.00)	40 (0.50)
B_2	30 (1.00)	60 (0.50)	15 (2.00)
Arithmetic mean speedup:	1.00	1.25	1.25

Table modified from [Fleming and Wallace(1986)]

Motivating example

What if we compute speedup vs. Y instead of X?

	Proc. time (**speedup vs. X**)			Proc. time (**speedup vs. Y**)		
Benchmark	X	Y	Z	X	Y	Z
B_1	20 (1.00)	10 (2.00)	40 (0.50)	20 (0.50)	10 (1.00)	40 (0.25)
B_2	30 (1.00)	60 (0.50)	15 (2.00)	30 (2.00)	60 (1.00)	15 (4.00)
A. mean speedup:	1.00	1.25	1.25	1.25	1.00	2.13

Tables modified from [Fleming and Wallace(1986)]

What can we conclude from this example?

- Arithmetic means of speedups and similarly normalized results are meaningless[1]

[1] [Fleming and Wallace(1986)]

Why does the arithmetic mean fail?

For benchmark i, if

- **X** is A times faster than **Y** and
- **Y** is B times faster than **Z**,

then

- **X** is $A \cdot B$ times faster than **Z**.

Suppose we want this logic to hold for mean speedups

Why does the arithmetic mean fail?

	s_i^{XvY}	s_i^{YvZ}	s_i^{XvZ}
B_1	0.50	4.00	2.00
B_2	2.00	0.25	0.50
A. mean	1.25	2.13	1.25

Mean speedup notation:

$$\text{mean}(s_0^{XvY}, \ldots, s_n^{XvY}) = s_m^{XvY}$$

With arithmetic mean, $s_m^{XvY} \cdot s_m^{YvZ} = 2.66 \neq s_m^{XvZ}$

Arithmetic mean does not have **multiplicative property**

Multiplicative property

We would like mean function for speedups to have multiplicative property,

$$f(a_1 \cdot b_1, \ldots, a_n \cdot b_n) = f(a_1, \ldots, a_n) \cdot f(b_1, \ldots, b_n),$$

which does not hold for arithmetic mean:

$$\frac{1}{n} \cdot \left(\sum_{i=1}^{n} a_i \cdot b_i\right) \neq \frac{1}{n} \cdot \left(\sum_{i=1}^{n} a_i\right) \cdot \frac{1}{n} \cdot \left(\sum_{i=1}^{n} b_i\right).$$

Alternative approach

Can we summarize these speedups in a more meaningful way?

Alternative approach

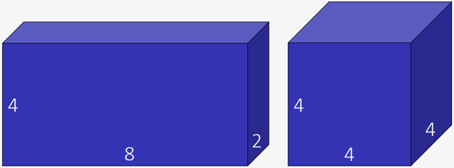
Can we summarize these speedups in a more meaningful way?

- One option: the **geometric mean**

$$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$$

The geometric mean

- Geometric interpretation: find edge length of hypercube with same volume as hyperrectangle with given edge lengths



$$(2 \cdot 4 \cdot 8)^{1/3} = 4$$

- Has multiplicative property (if data sets have equal size)

$$\left(\prod_{i=1}^{n} a_i \cdot b_i\right)^{\frac{1}{n}} = \left(\prod_{i=1}^{n} a_i\right)^{\frac{1}{n}} \cdot \left(\prod_{i=1}^{n} b_i\right)^{\frac{1}{n}}$$

Mean of product == product of means?

	s_i^{XvY}	s_i^{YvZ}	s_i^{XvZ}
B_1	0.50	4.00	2.00
B_2	2.00	0.25	0.50
A. mean	1.25	2.13	1.25

Mean speedup notation:

$$\text{mean}(s_0^{XvY}, \ldots, s_n^{XvY}) = s_m^{XvY}$$

With arithmetic mean, $s_m^{XvY} \cdot s_m^{YvZ} = 2.66 \neq s_m^{XvZ}$

Mean of product == product of means?

	s_i^{XvY}	s_i^{YvZ}	s_i^{XvZ}
B_1	0.50	4.00	2.00
B_2	2.00	0.25	0.50
A. mean	1.25	2.13	1.25
G. mean	1.00	1.00	1.00

Mean speedup notation:

$$\text{mean}(s_0^{XvY}, ..., s_n^{XvY}) = s_m^{XvY}$$

With arithmetic mean, $s_m^{XvY} \cdot s_m^{YvZ} = 2.66 \neq s_m^{XvZ}$

With geometric mean, $s_m^{XvY} \cdot s_m^{YvZ} = 1.00 = s_m^{XvZ}$

Speedup results with geometric mean

	Proc. time (**speedup vs. X**)			Proc. time (**speedup vs. Y**)		
Benchmark	X	Y	Z	X	Y	Z
B_1	20 (1.00)	10 (2.00)	40 (0.50)	20 (0.50)	10 (1.00)	40 (0.25)
B_2	30 (1.00)	60 (0.50)	15 (2.00)	30 (2.00)	60 (1.00)	15 (4.00)
G. mean speedup:	1.00	1.00	1.00	1.00	1.00	1.00

Tables modified from [Fleming and Wallace(1986)]

Speedup results with geometric mean

	Proc. time (**speedup vs. X**)			Proc. time (**speedup vs. Y**)		
Benchmark	X	Y	Z	X	Y	Z
B_1	20 (1.00)	10 (2.00)	40 (0.50)	20 (0.50)	10 (1.00)	40 (0.25)
B_2	30 (1.00)	60 (0.50)	15 (2.00)	30 (2.00)	60 (1.00)	15 (4.00)
G. mean speedup:	1.00	1.00	1.00	1.00	1.00	1.00

	Proc. time (**speedup vs. X**)			Proc. time (**speedup vs. Y**)		
Benchmark	X	Y	Z	X	Y	Z
B_1	20 (1.00)	**20 (1.00)**	40 (0.50)	20 (0.50)	**20 (1.00)**	40 (0.25)
B_2	30 (1.00)	**120 (0.25)**	15 (2.00)	30 (2.00)	**120 (1.00)**	15 (4.00)
G. mean speedup:	1.00	0.50	1.00	2.00	1.00	2.00

Tables modified from [Fleming and Wallace(1986)]

Satisfied with geometric mean of speedups?

	Proc. time (**speedup vs. X**)			Proc. time (**speedup vs. Y**)		
Benchmark	X	Y	Z	X	Y	Z
B_1	20 (1.00)	10 (2.00)	40 (0.50)	20 (0.50)	10 (1.00)	40 (0.25)
B_2	30 (1.00)	60 (0.50)	15 (2.00)	30 (2.00)	60 (1.00)	15 (4.00)
G. mean speedup:	1.00	1.00	1.00	1.00	1.00	1.00

Observe any issues with this?

Tables modified from [Fleming and Wallace(1986)]

Satisfied with geometric mean of speedups?

	Proc. time (**speedup vs. X**)			Proc. time (**speedup vs. Y**)		
Benchmark	X	Y	Z	X	Y	Z
B_1	20 (1.00)	10 (2.00)	40 (0.50)	20 (0.50)	10 (1.00)	40 (0.25)
B_2	30 (1.00)	60 (0.50)	15 (2.00)	30 (2.00)	60 (1.00)	15 (4.00)
G. mean speedup:	1.00	1.00	1.00	1.00	1.00	1.00

Observe any issues with this? Results consistent regardless of normalization, but is this a good way to characterize this data?

- What if sum of results has meaning? Or weighted sum/average?
- Lost information about total execution times

Tables modified from [Fleming and Wallace(1986)]

Another perspective

[Smith(1988)]: yes, arithmetic mean of *speedups* (or flop rates) is meaningless, but total or weighted **total exec. time is more informative**

- Geometric mean of normalized values yields consistent, but uninformative result
- Instead, perform appropriate aggregate computation *before* normalizing, not after
- Single-value measure for benchmark times should be directly proportional to total time consumed by benchmarks
    - G. mean of times fails this test
    - G. mean of speedups lacks this info

	Proc. time		
	X	Y	Z
B_1	20	10	40
B_2	30	60	15
Sum	50	70	55
A. mean	25.0	35.0	27.5
G. mean	24.5	24.5	24.5
G. mean speedups	1.00	1.00	1.00

Another perspective

[Smith(1988)]: use **harmonic mean** to summarize performance rates (flop/s) because equivalent to dividing total ops by total time[1]

- Harmonic mean: $n \cdot \left(\sum_{i=1}^{n} a_i^{-1}\right)^{-1}$
- If B_i executes f_i flops in t_i seconds,

$$\underbrace{n \cdot \left(\sum_{i=1}^{n} \left(\frac{f_i}{t_i}\right)^{-1}\right)^{-1}}_{\text{Harmonic mean of rates}} = n \cdot \left(\sum_{i=1}^{n} \frac{t_i}{f_i}\right)^{-1} \stackrel{?}{=} \underbrace{\frac{\sum_{i=1}^{n} f_i}{\sum_{i=1}^{n} t_i}}_{\text{total flop/s}}$$

Another perspective

[Smith(1988)]: use **harmonic mean** to summarize performance rates (flop/s) because equivalent to dividing total ops by total time[1]

- Harmonic mean: $n \cdot \left(\sum_{i=1}^{n} a_i^{-1}\right)^{-1}$
- If B_i executes f_i flops in t_i seconds,

$$\underbrace{n \cdot \left(\sum_{i=1}^{n} \left(\frac{f_i}{t_i}\right)^{-1}\right)^{-1}}_{\text{Harmonic mean of rates}} = n \cdot \left(\sum_{i=1}^{n} \frac{t_i}{f_i}\right)^{-1} \stackrel{?}{=} \underbrace{\frac{\sum_{i=1}^{n} f_i}{\sum_{i=1}^{n} t_i}}_{\text{total flop/s}}$$

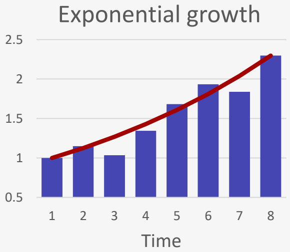
- [1]Only true if f_i constant across programs, which paper assumes

Other uses of geometric mean

- To treat constant factor changes equally
  - If we consider 20% error twice as bad as 10% error, and 40% error twice as bad as 20% error, and we want

$$\mathrm{mean}(0.1, 0.2, 0.4) = 0.2$$

- When product of values has meaning, e.g., compound annual growth rate



Paper 1 conclusions

Three rules from [Fleming and Wallace(1986)]:

1. Do not use the arithmetic mean to average normalized numbers
2. Use the geometric mean to average normalized numbers
3. Use the sum (or arithmetic mean) of raw, unnormalized results whenever this "total" has some meaning

Paper 2 conclusions

Conclusions from [Smith(1988)]:

- Total exec. time of benchmarks more informative than mean speedup
- Perform appropriate aggregate computation *before* normalizing, not after
- Use harmonic mean to summarize flop rates (only works w/constant flop counts)

Conclusions

Questions when choosing mean

- Are values normalized? If so, should we aggregate before normalizing?
- What properties should mean have?
- Does (weighted) sum of values have meaning?
- Does product of values have meaning?
- How large is variance? Large variance reduces meaningfulness of means. Does it even make sense to summarize data with single value?
    - "The uselessness of arithmetic mean as a performance predictor cannot be emphasized enough. Giving additional statistics such as standard deviation [...] does not mitigate the situation. [Adding] standard deviation is similar to saying: Here is a meaningless performance measure, and here is a measure of just how meaningless it is."[2]

[2] [Smith(1988)]

Bibliography

Philip J Fleming and John J Wallace.
How not to lie with statistics: the correct way to summarize benchmark results.
Communications of the ACM, 29(3):218–221, 1986.

James E Smith.
Characterizing computer performance with a single number.
Communications of the ACM, 31(10):1202–1206, 1988.