How did *this* get published?

Pitfalls in experimental evaluation of computing systems

José Nelson Amaral

University of Alberta

Edmonton, AB, Canada
Thing #1: Aggregation

Thing #2: Learning

Thing #3: Reproducibility
So, a computing scientist entered a Store....

http://archive.constantcontact.com/fs042/1101916237075/archive/1102594461324.html

So, a computing scientist entered a Store….

They want $2,700 for the server and $100 for the iPod.

I will get both and pay only $2,240 altogether!

So, a computing scientist entered an Store….

Ma’am you are $560 short.

But the average of 10% and 50% is 30% and 70% of $3,200 is $2,240.

$ 3,000.00

$ 200.00

$ 3,000.00

$ 200.00

http://www.businessinsider.com/10-ways-to-fix-googles-busted-android-app-market-2010-1?op=1

So, a computing scientist entered an Store....

Ma’am you cannot take the arithmetic average of percentages!

But... I just came from at top CS conference in San Jose where they do it!

$h$�://www.businessinsider.com/10-­‐ways-­‐to-­‐fix-­‐googles-­‐busted-­‐android-­‐app-­‐market-­‐2010-­‐17op=1

The Problem with Averages
A Hypothetical Experiment

Execution Time

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Time (minutes)</th>
<th>Baseline</th>
<th>Transformed</th>
</tr>
</thead>
<tbody>
<tr>
<td>benchA</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>benchB</td>
<td>2</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>benchC</td>
<td>2</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>benchD</td>
<td>4</td>
<td>20</td>
<td>4</td>
</tr>
</tbody>
</table>

*With thanks to Iain Ireland*
Speedup

\[
\text{Speedup} = \frac{\text{Baseline Time}}{\text{Transformed Time}}
\]
The transformed system is, on average, 2.6 times faster than the baseline!
Normalized Time

\[
\text{Normalized Time} = \frac{\text{Transformed Time}}{\text{Baseline Time}}
\]
The transformed system is, on average, 2.6 times slower than the baseline!
Latency $\times$ Throughput

• What matters is latency:

• What matters is throughput:
Aggregation for Latency: Geometric Mean

$$GeoMean = \sqrt[n]{\prod_{i=0}^{n-1} S_i}$$
The performance of the transformed system is, on average, the same as the baseline!
The performance of the transformed system is, on average, the same as the baseline!
### Aggregation for Throughput

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Execution Time (minutes)</th>
<th>Baseline</th>
<th>Transformed</th>
</tr>
</thead>
<tbody>
<tr>
<td>benchA</td>
<td>2</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>benchB</td>
<td>10</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>benchC</td>
<td>20</td>
<td>5.25</td>
<td>8.25</td>
</tr>
<tr>
<td>benchD</td>
<td>20</td>
<td>8.25</td>
<td>5.25</td>
</tr>
<tr>
<td>Arith</td>
<td>8.25</td>
<td>5.25</td>
<td>1.6</td>
</tr>
<tr>
<td>Avg</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

The throughput of the transformed system is, on average, **1.6 times faster** than the baseline.

**Speedup** = \( \frac{8.25}{5.25} = 1.6 \)
The Evidence

• A careful reader will find the use of arithmetic average to aggregate normalized numbers in many top CS conferences.

• Papers that have done that have appeared in:
  – LCTES 2011
  – PLDI 2012 (at least two papers)
  – CGO 2012
    • A paper where the use of the wrong average changed a negative conclusion into a positive one.
  – 2007 SPEC Workshop
    • A methodology paper by myself and a student that won the best paper award.
This is not a new observation...

Edgar H. Sibley
Panel Editor

Using the arithmetic mean to summarize normalized benchmark results leads to mistaken conclusions that can be avoided by using the preferred method: the geometric mean.

HOW NOT TO LIE WITH STATISTICS: THE CORRECT WAY TO SUMMARIZE BENCHMARK RESULTS

PHILIP J. FLEMING and JOHN J. WALLACE

RULE 1: Do Not Use the Arithmetic Mean to Average Normalized Numbers

RULE 2: Use the Geometric Mean to Average Normalized Numbers

RULE 3: Use the Sum (or arithmetic mean) of Raw, Unnormalized Results whenever This “Total” Has Some Meaning

No need to dig dusty papers...

Geometric mean
From Wikipedia, the free encyclopedia

\[
GM\left(\frac{X_i}{Y_i}\right) = \frac{GM(X_i)}{GM(Y_i)}
\]

This makes the geometric mean the only correct mean when averaging normalized results, that is results that are presented as ratios to reference values.\[^4\] This is the case when presenting
So, the computing scientist returns to the Store...

Hello. I am just back from Beijing. Now I know that we should take the geometric average of percentages.

Hello.

I am just back from Beijing.

Now I know that we should take the geometric average of percentages.
So, a computing scientist entered a Store....

Sorry Ma’am, we don’t average percentages...

Thus I should get \( \sqrt{50 \times 10} \)

\[= 22.36\% \text{ discount and pay } 0.7764 \times \$3,200 = \$2,484.48 \]
So, a computing scientist entered a Store....

The original price is $3,200. You pay $2,700 + $100 = $2,800. If you want an aggregate summary, your discount is $400/3,200 = 12.5%
Disregard to methodology when using automated learning
Example:
Evaluation of Feedback Directed Optimization (FDO)
We have:

We want to measure the effectiveness of an FDO-based code transformation.
Generic relations were moved in Django revision 5172:
try:
    from django.contrib.contenttypes import generic
except ImportError:
    from django.db.models import generic

class Tag(models.Model):
    A basic tag

    name = models.CharField(max_length=50, unique=True,
                            db_index=True, error_message=[u'isTag()]
    objects = TagManager()

class Meta:
    db_table = 'tag'
    verbose_name = 'Tag'
    verbose_name_plural = 'Tags'
    ordering = ('name',)
The FDO transformation produces code that is XX faster for this application.
The Evidence

• Many papers that use a single input for training and a single input for testing appeared in conferences (notably CGO).
• For instance, a paper that uses a single input for training and a single input for testing appears in:
  – ASPLOS 2004
Generic relations were moved in Django revision 5172.
```python
from django.contrib.contenttypes import import壁垒
except ImportError:
    import django.db.models as generic

class Tag(models.Model):
    A basic
    name = models.CharField(max_length=50, unique=True,
                            db_index=True)
    objects = TagManager()

class Meta:
    db_table = 'tag'
    verbose_name = 'Tag'
    verbose_name_plural = 'Tags'
    ordering = ('name',)
```

A diagram illustrates the evaluation set, application code, compiler, optimized code, and profile. The diagram shows the process of optimizing code through profiling and compilation. The website http://www.orchardoo.com is mentioned as a reference for more information.
Combined Profiling (Berube, ISPASS12)

Cross-Validated Evaluation (Berube, SPEC07)

http://www.orchardoo.com

Generic relations were moved in Django revision 5172

try:
    from django.contrib.contenttypes import generic
except ImportError:
    import django.db.models as generic

class Tag(models.Model):
    A basic class
    name = models.CharField(max_length=50, unique=True,
        db_index=True, null=True, blank=True)
    objects = TagManager()

    class Meta:
        db_table = 'tag'
        verbose_name = 'Tag'
        verbose_name_plural = 'Tags'
        ordering = ('name',)
Evaluation Set

Application Code

Compiler

Optimized Code

Wrong Evaluation!

Profile

Performance

[FDO]

http://www.orchardoo.com

Generic relations were moved in Django revision 5172
try:
    from django.contrib.contenttypes import generic
except ImportError:
    import django.db.models as generic

class Tag(models.Model):
    A basic Tag
    name = models.CharField(max_length=50, unique=True,
                             db_index=True, error_messages={'unique': '[tag]'
                                           ' must be unique'})
    objects = TagManager()

class Meta:
    db_table = 'tag'
    verbose_name = 'Tag'
    verbose_name_plural = 'Tags'
    ordering = ('name',)
The Evidence

• For instance, a paper that incorrectly uses the same input for training and testing appeared in:
  – PLDI 2006
Expectation:
When reproduced, an experimental evaluation should produce similar results.
Issues

Have the measurements been repeated a sufficient number of times to capture measurement variations?

Availability of code, data, and precise description of experimental setup.

Lack of incentives for reproducibility studies.
Progress

Program committees/reviewers starting to ask questions about reproducibility.

Steps toward infrastructure to facilitate reproducibility.
SPEC Research Group

http://research.spec.org/

14 industrial organizations
20 universities or research institutes
SPEC Research Group

http://research.spec.org/

Performance Evaluation

Benchmarks for New Areas

Performance Evaluation Tools

Evaluation Methodology

Repository for Reproducibility

http://icpe2013.ipd.kit.edu/

4th ACM/SPEC International Conference on Performance Engineering

ICPE 2013

Prague - Czech Republic - April 21-24
Evaluate Collaboratory:
http://evaluate.inf.usi.ch/

Open Letter to PC Chairs

Anti Patterns

Evaluation in CS education
Parting Thoughts....

Creating a culture that enables full reproducibility seems daunting...

Initially we could aim for:

Reasonable expectation by a reasonable reader that, if reproduced, the experimental evaluation would produce similar results.