Search Based Big Data Android App Energy Genetic Improvement in the Cloud

Karl Spearman, Charles Pearson & Hermann Mark
UCL, CRUST Center, Londen, Grate Britton.

Search Based Big Data Android App Energy Genetic Improvement in the Cloud

Karl Spearman, Charles Pearson & **Hermann Mark**
UCL, CRUST Center, Londen, Grate Britton.

Results

- Comparing Algorithms Advocated and Base-of-the-art.
- Lower is better (in our study).

Results

- Comparing Algorithms Advocated and Base-of-the-art.
- Lower is better (in our study).

	1	2	3	4	5	6	7	8	9	10
A	6.02	0.01	0	0.02	0.04	0.05	0.01	0.03	15	9.04
B	0.05	0.05	2	0.09	0.08	0.06	0.05	0.09	0.09	0.08

Transformed Vargha-Delaney Effect Size

Results

- Comparing Algorithms Advocated and Base-of-the-art.
- Lower is better (in our study).

	1	2	3	4	5	6	7	8	9	10
A	6.02	0.0 1	0	0.02	0.04	0.05	0.01	0.03	15	9.04
B	0.05	0.0 5	2	0.09	0.08	0.06	0.05	0.09	0.09	0.08

- Effect size is 0.315. ***Success!***
- ***AND SO***: <u>Our Algorithm A performs better!</u>

Results

- Comparing Algorithms Advocated and Base-of-the-art.
- Lower is better (in our study).

	1	2	3	4	5	6	7	8	9	10
A	6.02	0.01	0	0.02	0.04	0.05	0.01	0.03	15	9.04
B	0.05	0.05	2	0.09	0.08	0.06	0.05	0.09	0.09	0.08

- Effect size is 0.315. ***Success!***
- ***AND SO***: **Our Algorithm A performs better!**

Transformed Vargha-Delaney Effect Size

This work introduces

- Guidelines for ensuring that the Vargha Delaney effect size test tells us whether results are <u>usefully</u> better, not just whether they are better.
- Specific to a Search Based Software Engineering context.

Effect Size Testing

- For comparing randomized sets of results (common in SBSE).
- Hypothesis testing indicates whether a difference is significant.
- Effect size testing indicates how big the difference is.

Vargha Delaney A test for effect size

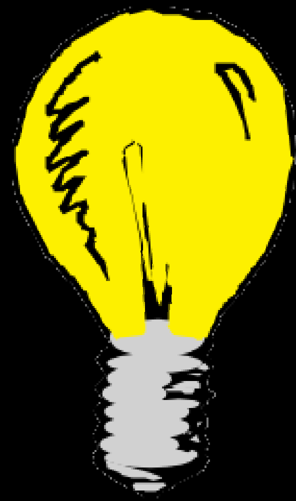
Calculates A_{12}, the probability that a randomly chosen value from group 1 is higher than one from group 2:

$$A_{12} = Prob(X_1 > X_2) + 0.5Prob(X_1 = X_2)$$

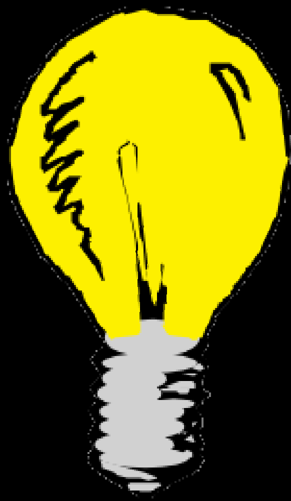
The Problem

- A difference that is small enough to be irrelevant is counted the same as a large difference.

- So if solution A wins by an insignificant margin in 70% of cases and solution B wins by a significant margin in 30% solution A wins, even if its benefits are <u>of no practical use</u>.



The solution is to ensure that only <u>meaningful</u> differences are considered.



Transform data to be meaningful

Well-known statistical approach

... yet not often done in SBSE

Two Approaches

- Pre Transformed Data (PTD)

  The simplest to implement - just modify the data beforehand.

- Modified Comparison Function (MCF)

  Some things can't be done through PTD and so instead modify the comparison function.

$$A_{12} = P(\mathbf{X_1 > X_2}) + 0.5P(\mathbf{X_1 = X_2})$$

How this may be implemented: SBSE examples.

- Implementation differences: Only a speed up greater than could be achieved by parallelization or different hardware is counted.
- Moore's law: An improvement needs to be at least double its competition.
- Delays: Delays of less than 10 seconds are ignored. Overnight delays are all counted as identical.

Uses for MCF

- With decomposable fitness functions solutions can be compared using every part of the fitness function:
  - Pareto dominance.
  - Has to improve in *n* instances.
  - Comparing growth function across instances.
  - And much more…

The Results

The same example as before BUT

All results under 0.1 are now discounted as "meaningless"

	1	2	3	4	5	6	7	8	9	10
A	6.02	0	0	0	0	0	0	0	15	9.04
B	0	0	2	0	0	0	0	0	0	0

The effect size is now 0.615

Algorithm B now wins

Final word

This can bridge the gap between saying results are "better" and saying they are "meaningfully better".

BUT:

- Caution advised.
- Standards across the research community or agreed with clients should be reached.

References

1. Ali, S., Briand, L.C., Hemmati, H., Panesar-Walawege, R.K.: A systematic review of the application and empirical investigation of search-based test case generation. Software Engineering, IEEE Transactions on 36(6), 742–762 (2010)

2. Amdahl, G.M.: Validity of the single processor approach to achieving large scale computing capabilities. In: Proceedings of the April 18-20, 1967, spring joint computer conference. pp. 483–485. ACM (1967)

3. Arcuri, A., Briand, L.: A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering. Software Testing, Verification and Reliability (2012)

4. Hsu, H.Y., Orso, A.: Mints: A general framework and tool for supporting test-suite minimization. In: Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on. pp. 419–429. IEEE (2009)

5. Moore, G.E., et al.: Cramming more components onto integrated circuits. Proceedings of the IEEE 86(1), 82–85 (1998)

6. Vargha, A., Delaney, H.D.: A critique and improvement of the CL common language effect size statistics of McGraw and Wong. Journal of Educational and Behavioral Statistics 25(2), 101–132 (2000)

Any questions?