Performance Metrics

Measuring and quantifying computer systems performance

Samuel Kounev

“Time is a great teacher, but unfortunately it kills all its pupils.”
-- Hector Berlioz

1

References

- „Measuring Computer Performance – A Practitioner's Guide“ by David J. Lilja, Cambridge University Press, New York, NY, 2000, ISBN 0-521-64105-5
- The supplemental teaching materials provided at http://www.arctic.umn.edu/perf-book/ by David J. Lilja
- Chapter 4 in „Performance Evaluation and Benchmarking“ – by Lizy Kurian John, ISBN 0-8493-3622-8

2

Roadmap

- Performance metrics
  - Characteristics of good performance metrics
  - Summarizing performance with a single value
  - Quantifying variability
  - Aggregating metrics from multiple benchmarks
- Errors in experimental measurements
  - Accuracy, precision, resolution
  - Confidence intervals for means
  - Confidence intervals for proportions

3

Performance Metrics

- Values derived from some fundamental measurements
  - *Count* of how many times an event occurs
  - *Duration* of a time interval
  - *Size* of some parameter
- Some basic metrics include
  - Response time
    - Elapsed time from request to response
  - Throughput
    - Jobs or operations completed per unit of time
  - Bandwidth
    - Bits per second
  - Resource utilization
    - Fraction of time the resource is used
- Standard benchmark metrics
  - For example, SPEC and TPC benchmark metrics

4

Characteristics of Good Metrics

- Linear
  - proportional to the actual system performance
- Reliable
  - Larger value → better performance
- Repeatable
  - Deterministic when measured
- Consistent
  - Units and definition constant across systems
- Independent
  - Independent from influence of vendors
- Easy to measure

5

Some Examples of Standard Metrics

- Clock rate
  - Easy-to-measure, Repeatable, Consistent, Independent, Non-Linear, Unreliable
- MIPS
  - Easy-to-measure, Repeatable, Independent, Non-Linear, Unreliable, Inconsistent
- MFLOPS, GFLOPS, TFLOPS, PFLOPS, …
  - Easy-to-measure, Repeatable, Non-Linear, Unreliable, Inconsistent, Dependent
- SPEC metrics (www.spec.org)
  - SPECcpu, SPECweb, SPECjbb, SPECjAppServer, etc.
- TPC metrics (www.tpc.org)
  - TPC-C, TPC-H, TPC-App

6

Speedup and Relative Change

- "Speed" refers to any rate metric → $R_i = D_i / T_i$
  - D_i ~ "distance traveled" by system i
  - T_i = measurement interval
- **Speedup** of system 2 w.r.t system 1
  - $S_{2,1}$ such that: $R_2 = S_{2,1} R_1$
- **Relative change**

$$\Delta_{2,1} = \frac{R_2 - R_1}{R_1}$$

$$\Delta_{2,1} > 0 \Rightarrow \text{System 2 is faster than system 1}$$
$$\Delta_{2,1} < 0 \Rightarrow \text{System 2 is slower than system 1}$$

7

Summarizing System Performance

- Two common scenarios
  - Summarize multiple measurements of a given metric
  - Aggregate metrics from multiple benchmarks
- Desire to reduce system performance to a single number
- Indices of central tendency used
  - Arithmetic mean, median, mode, harmonic mean, geometric mean
- Problem
  - Performance is multidimensional, e.g. response time, throughput, resource utilization, efficiency, etc.
  - Systems are often specialized → perform great for some applications, bad for others

8

Expected Value and Sample Mean

- Look at measured values $(x_1,\ldots,x_n)$ as a random **sample** from a population, i.e. measured values are values of a random variable X with an unknown distribution.
- The most common index of central tendency of X is its **mean E[X]** (also called **expected value** of X)
  - If X is discrete and $p_x = Pr(X = x) = Pr($"we measure x"$)$

$$E[X] = \sum_x x.\Pr(X = x) = \sum_x x.p_x$$

  - The **sample mean** (arithmetic mean) is an estimate of E[X]

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

9

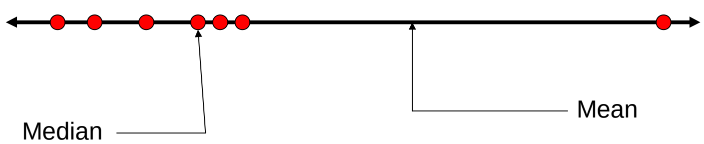
Common Indices of Central Tendency

- **Sample Mean**
  - Use when the sum of all values is meaningful
  - Incorporates all available information
- **Median**
  - the "middle" value (such that ½ of the values are above, ½ below)
  - Sort *n* values (measurements)
    - If n is odd, median = middle value
    - Else, median = mean of two middle values
  - Less influenced by outliers
- **Mode**
  - The value that occurs most often
  - Not unique if multiple values occur with same frequency
  - Use when values represent categories, i.e. data can be grouped into distinct types/categories (*categorical data*)

10

Sample Mean and Outliers

- Sample mean gives equal weight to all measurements
- *Outliers* can have a large influence on the computed mean value
- Distorts our intuition about the *central tendency*



11

Other Types of Means

- **Arithmetic Mean (Sample Mean)**
  - When sum of raw values has physical meaning
  - Typically used to summarize **times**

$$\overline{x_A} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- **Harmonic Mean**
  - Typically used to summarize **rates**

$$\overline{x_H} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

- **Geometric Mean**
  - Used when product of raw values has physical meaning

$$\overline{x_G} = \sqrt[n]{x_1 x_2 \cdots x_i \cdots x_n} = \left(\prod_{i=1}^{n} x_i \right)^{1/n}$$

12

Geometric Mean

- Maintains consistent relationships when comparing normalized values
  - Provides consistent rankings
  - Independent of basis for normalization
- Meaningful only when the product of raw values has physical meaning
- Example
  - If improvements in CPI and clock periods are given, the mean improvement for these two design changes can be found by the geometric mean.

13

Weighted Means

- Standard definitions of means assume all measurements are equally important
- If that's not the case, one can use weights to represent the relative importance of measurements
- E.g. if application 1 is run more often than application 2 it should have a higher weight

$$\sum_{i=1}^{n} w_i = 1$$

$$\bar{x}_{A,w} = \sum_{i=1}^{n} w_i x_i$$

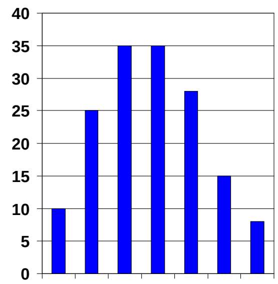
$$\bar{x}_{H,w} = \frac{1}{\sum_{i=1}^{n} \frac{w_i}{x_i}}$$

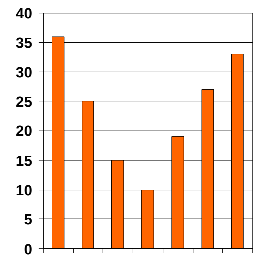
$$\bar{x}_{G,w} = \prod_{i=1}^{n} x_i^{w_i}$$

14

Quantifying Variability

- Means hide information about variability
- How "spread out" are the values?
- How much spread relative to the mean?
- What is the shape of the distribution of values?





15

Indices of Dispersion

- Used to quantify variability
  - Range = (max value) – (min value)
  - Maximum distance from the mean = Max of | x_i – mean |
  - Neither efficiently incorporates all available information
- Most commonly the **sample variance** is used

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} = \frac{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}{n(n-1)}$$

  - Referred to as having "**(n-1) degrees of freedom"**
  - Second form good for calculating "on-the-fly"
    - One pass through data

16

Most Common Indices of Dispersion

- **Sample Variance**
  - In “units-squared” compared to mean
  - Hard to compare to mean
- **Standard Deviation s**
  - s = square root of variance
  - Has units same as the mean
- **Coefficient of Variation (COV)** $COV = \frac{s}{\bar{x}}$
  - Dimensionless
  - Compares relative size of variation to mean value

17

Aggregating Performance Metrics From Multiple Benchmarks

- Problem: How should metrics obtained from component benchmarks of a benchmark suite be aggregated to present a summary of the performance over the entire suite?
- What central tendency measures are valid over the whole benchmark suite for speedup, CPI, IPC, MIPS, MFLOPS, cache miss rates, cache hit rates, branch misprediction rates, and other measurements?
- What would be the appropriate measure to summarize speedups from individual benchmarks?

18

MIPS as an Example

- Assume that the benchmark suite is composed of *n* benchmarks, and their individual MIPS are known:
  - I_i is the instruction count of i^{th} benchmark (in millions)
  - t_i is the execution time of i^{th} benchmark
  - MIPS_i is the MIPS rating of the i^{th} benchmark
  - The overall MIPS is the MIPS when the n benchmarks are considered as part of a big application:

$$Overall\ MIPS = \frac{\sum_{i=1}^{n} I_i}{\sum_{i=1}^{n} t_i}$$

19

MIPS as an Example (2)

- The overall MIPS of the suite can be obtained by computing:
  - a ***weighted harmonic mean (WHM)*** of the MIPS of the individual benchmarks weighted according to the instruction counts

    OR
  - a ***weighted arithmetic mean (WAM)*** of the individual MIPS with weights corresponding to the execution times spent in each benchmark in the suite.

20

MIPS as an Example (3)

$w_i^{ic} = \frac{I_i}{\sum_{k=1}^{n} I_k}$ is the weight of i^{th} benchmark according to instruction count

$$\text{WHM with weights corresponding to instruction counts} = \frac{1}{\sum_{i=1}^{n} \frac{w_i^{ic}}{MIPS_i}} =$$

$$= \frac{1}{\frac{1}{\sum_{k=1}^{n} I_k} \sum_{i=1}^{n} \frac{I_i}{MIPS_i}} = \frac{\sum_{k=1}^{n} I_k}{\sum_{i=1}^{n} \frac{I_i}{MIPS_i}} = \frac{\sum_{k=1}^{n} I_k}{\sum_{i=1}^{n} \frac{I_i t_i}{I_i}} = \frac{\sum_{k=1}^{n} I_k}{\sum_{i=1}^{n} t_i} =$$

$$= Overall\ MIPS$$

21

MIPS as an Example (4)

$w_i^{et} = \frac{t_i}{\sum_{k=1}^{n} t_k}$ is the weight of i^{th} benchmark according to execution time

$$\text{WAM with weights corresponding to execution time} = \sum_{i=1}^{n} w_i^{et} MIPS_i =$$

$$= \frac{1}{\sum_{k=1}^{n} t_k} \left[\sum_{i=1}^{n} t_i MIPS_i \right] = \frac{1}{\sum_{k=1}^{n} t_k} \left[\sum_{i=1}^{n} t_i \frac{I_i}{t_i} \right] = \frac{\sum_{i=1}^{n} I_i}{\sum_{k=1}^{n} t_k}$$

$$= Overall\ MIPS$$

22

Example

Benchmark	Instruction Count (in millions)	Time (sec)	Individual MIPS
1	500	2	250
2	50	1	50
3	200	1	200
4	1000	5	200
5	250	1	250

Example (cont.)

- Weights of the benchmarks with respect to instruction counts:

  {500/2000, 50/2000, 200/2000, 1000/2000, 250/2000} =

  {0.25, 0.025, 0.1, 0.5, 0.125}
- Weights of the benchmarks with respect to time:

  {0.2, 0.1, 0.1, 0.5, 0.1}
- WHM of individual MIPS (weighted with *I*-counts) =

  1 / (0.25/250 + 0.025/50 + 0.1/200 + 0.5/200 + 0.125/250) = 200
- WAM of individual MIPS (weighted with time) =

  250*0.2 + 50*0.1 + 200*0.1 + 200*0.5 + 250*0.1 = 200

Example (cont.)

$$Overall\ MIPS = \left(\sum_{i=1}^{n} I_i\right) / \left(\sum_{i=1}^{n} t_i\right) = 2000/10 = 200$$

- WHM of individual MIPS (weighted with *I*-counts) = 200
- WAM of individual MIPS (weighted with time) = 200

- Unweighted arithmetic mean of individual MIPS = 190
- Unweighted harmonic mean of individual MIPS = 131.58

- Neither of the unweighted means is indicative of the overall MIPS!

25

Arithmetic vs. Harmonic Mean

- If a metric is obtained by dividing A by B, either *harmonic mean* with weights corresponding to the measure in the numerator or *arithmetic mean* with weights corresponding to the measure in the denominator is valid when trying to find the aggregate measure from the values of the measures in the individual benchmarks.

- If A is weighted equally among the benchmarks, simple (unweighted) harmonic mean can be used.

- If B is weighted equally among the benchmarks, simple (unweighted) arithmetic mean can be used.

26

Aggregating Metrics

Measure	Valid Central Tendency for Summarized Measure Over a Benchmark Suite	
A/B	WAM weighted with Bs	WHM weighted with As
IPC	WAM weighted with cycles	WHM weighted with *I*-count
CPI	WAM weighted with *I*-count	WHM weighted with cycles
MIPS	WAM weighted with time	WHM weighted with *I*-count
MFLOPS	WAM weighted with time	WHM weighted with FLOP count
Cache hit rate	WAM weighted with number of references to cache	WHM weighted with number of cache hits

27

Aggregating Metrics (cont.)

Measure	Valid Central Tendency for Summarized Measure Over a Benchmark Suite	
Cache misses per instruction	WAM weighted with *I*-count	WHM weighted with number of misses
Branch misprediction rate per branch	WAM weighted with branch counts	WHM weighted with number of mispredictions
Normalized execution time	WAM weighted with execution times in system considered as base	WHM weighted with execution times in the system being evaluated
Transactions per minute	WAM weighted with exec times	WHM weighted with proportion of transactions for each benchmark

28

Exercise

- A benchmark consists of two parts: part 1 runs image processing for 1 hour, and part 2 runs compression for 1 hour.
- Assume that benchmark is run on a system and part 1 achieves MIPS1, part 2 achieves MIPS2
- How can these two results be summarized to derive an overall MIPS of the system?

29

Speedup

- What would be the appropriate measure to summarize *speedups* from individual benchmarks of a suite?
  - WHM of the individual speedups with weights corresponding to the execution times in the *baseline system*
  - WAM of the individual speedups with weights corresponding to the execution times in the *enhanced system*

30

Example

Benchmark	Time on Baseline System	Time on Enhanced System	Individual Speedup
1	500	250	2
2	50	50	1
3	200	50	4
4	1000	1250	0.8
5	250	200	1.25

- Total time on baseline system = 2000 sec
- Total time on enhanced system = 1800 sec
- Overall speedup = 2000/1800 = 1.111

31

Example (cont.)

- Weights corresponding to execution times on baseline system:
  - {500/2000, 50/2000, 200/2000, 1000/2000, 250/2000}
- Weights corresponding to execution times on enhanced system:
  - {250/1800, 50/1800, 50/1800, 1250/1800, 200/1800}
- WHM of individual speedups =
  - 1 / (500/(2000*2) + 50/(2000*1) + 200/(2000*4) + 1000/(2000*0.8) + 250/(2000*1.25)) = … = 1.111
- WAM of individual speedups =
  - 2*250/1800 + 1*50/1800 + 4*50/1800 + 0.8*1250/1800 + 1.25*200/1800 = … = 1.111

32

Use of Simple (Unweighted) Means

Measure	To Summarize Measure over a Benchmark Suite	
	Simple arithmetic mean valid?	Simple harmonic mean valid?
A/B	If Bs are equal	If As are equal
Speedup	If equal execution times in each benchmark in the improved system	If equal execution times in each benchmark in the baseline system
IPC	If equal cycles in each benchmark	If equal *I*-count in each benchmark
CPI	If equal *I*-count in each benchmark	If equal cycles in each benchmark
MIPS	If equal times in each benchmark	If equal *I*-count in each benchmark
MFLOPS	If equal times in each benchmark	If equal FLOPS in each benchmark

33

Use of Simple (Unweighted) Means (2)

Measure	To Summarize Measure over a Benchmark Suite	
	Simple arithmetic mean valid?	Simple harmonic mean valid?
Cache hit rate	If equal number of references to cache for each benchmark	If equal number of cache hits in each benchmark
Cache misses per instruction	If equal *I*-count in each benchmark	If equal number of misses in each benchmark
Branch misprediction rate per branch	If equal number of branches in each benchmark	If equal number of mispredictions in each benchmark
Normalized execution time	If equal execution times in each benchmark in the system considered as base	If equal execution times in each benchmark in the system evaluated
Transactions per minute	If equal times in each benchmark	If equal number of transactions in each benchmark

34

Weighting Based on Target Workload

- Ideally, when aggregating metrics each benchmark should be weighted for whatever fraction of time it will run in the user's target workload.
- For example if benchmark 1 is a compiler, benchmark 2 is a digital simulation, and benchmark 3 is compression, for a user whose actual workload is digital simulation for 90% of the day, and 5% compilation and 5% compression, WAM with weights 0.05, 0.9, and 0.05 will yield a valid overall MIPS on the target workload.
- If each benchmark is expected to run for an equal period of time, finding a simple (unweighted) arithmetic mean of the MIPS is not an invalid approach.

35

Roadmap

- Performance metrics
  - Characteristics of good performance metrics
  - Summarizing performance with a single value
  - Quantifying variability
  - Aggregating metrics from multiple benchmarks
- Errors in experimental measurements
  - Accuracy, precision, resolution
  - Confidence intervals for means
  - Confidence intervals for proportions

36

Experimental Errors

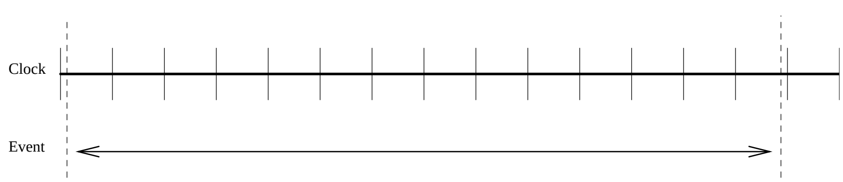
- Errors → *noise* in measured values
- *Systematic* errors
  - Result of an experimental "mistake"
  - Typically produce constant or slowly varying bias
  - Controlled through skill of experimenter
  - Example: forget to clear cache before timing run
- *Random* errors
  - Unpredictable, non-deterministic, unbiased
  - Result of
    - Limitations of measuring tool
    - Random processes within system
  - Typically cannot be controlled
    - Use statistical tools to characterize and quantify

37

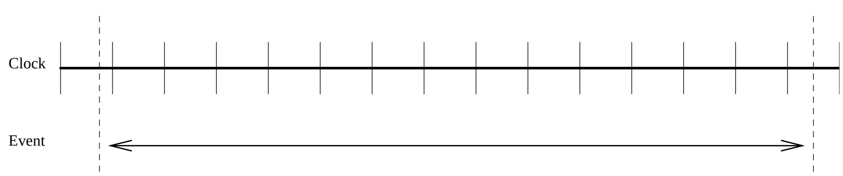
Example: Quantization

Timer resolution → quantization error

Repeated measurements X ± Δ (completely unpredictable)



(a) Interval timer reports event duration of n = 13 clock ticks.



(b) Interval timer reports event duration of n = 14 clock ticks.

38

A Model of Errors

1 error source →

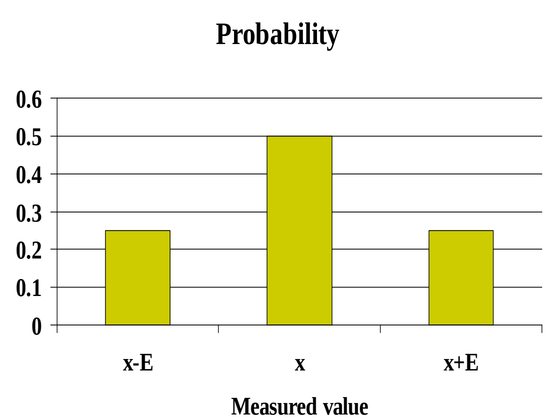
Error	*Measured value*	*Probability*
-E	*x* – E	½
+E	*x* + E	½

2 error sources →

Error 1	*Error 2*	*Measured value*	*Probability*
-E	-E	*x* – 2E	¼
-E	+E	*x*	¼
+E	-E	*x*	¼
+E	+E	*x* + 2E	¼

39

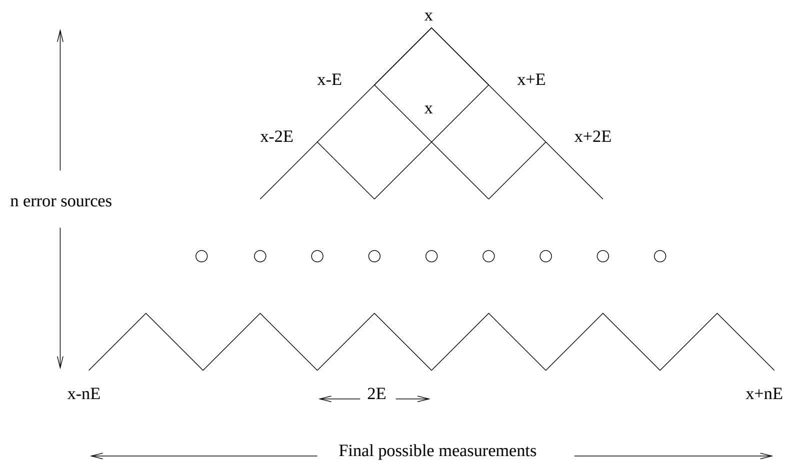
A Model of Errors (2)



40

A Model of Errors (3)

Probability of obtaining a specific measured value



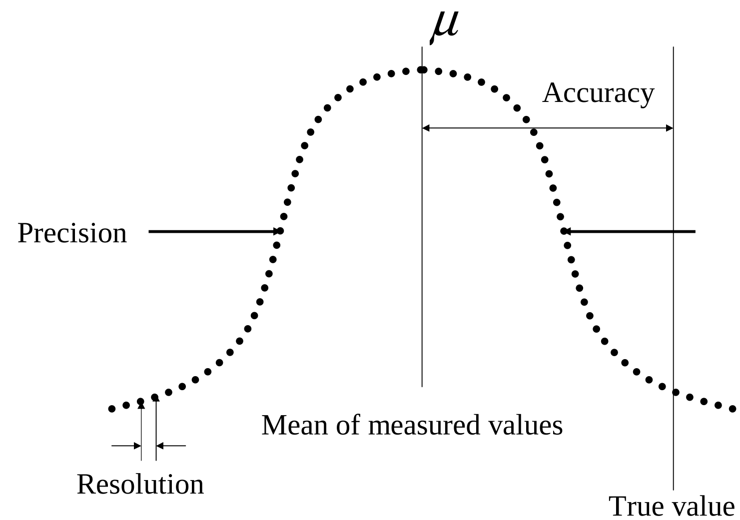
41

A Model of Errors (4)

- Look at the measured value as a random variable X
- $Pr(X=x_i)$ = Pr(measure x_i) is proportional to the number of paths from real value to x_i
- $Pr(X=x_i)$ ~ binomial distribution
- As number of error sources becomes large
  - $n \rightarrow \infty$,
  - Binomial $\rightarrow$ Gaussian (Normal)
- Thus, the bell curve

42

Frequency of Measuring Specific Values



43

Accuracy, Precision and Resolution

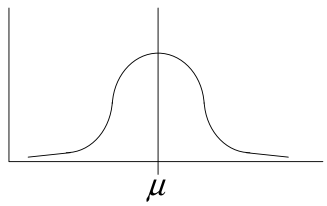
- **Accuracy**
  - How close mean of measured values is to true value?
  - Systematic errors cause inaccuracy
- **Precision**
  - Random errors cause imprecision
  - Quantify amount of *imprecision* using statistical tools
- **Resolution**
  - Smallest increment between measured values
  - Dependent on measurement tools used

44

Confidence Interval for the Mean μ

- Assume errors are **normally distributed**, i.e. measurements are samples from a normally distributed population
- Will now show how to quantify the *precision* of measurements using confidence intervals
- Assume *n* measurements $x_1,\ldots,x_n$ are taken
- Measurements form a set of IID random variables

$$x_i \in N(\mu, \sigma^2)$$



45

Confidence Interval for the Mean μ (2)

- Looking for an interval $[c_1,c_2]$ such that

$$\Pr[c_1 \le \mu \le c_2] = 1-\alpha$$

- Typically, a symmetric interval is used so that

$$\Pr[\mu < c_1] = \Pr[\mu > c_2] = \frac{\alpha}{2}$$

- The interval $[c_1,c_2]$ is called **confidence interval** for the mean μ
- α is called the **significance level** and $(1-\alpha)\times 100$ is called the **confidence level**.

46

Case 1: Number of Measurements >= 30

- Measurements $x_1,\dots,x_n$ form a sample from a normal distribution
- The sample mean $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$

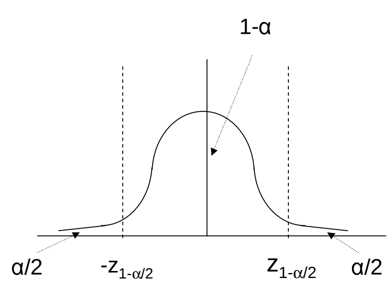
$$x_i \in N(\mu, \sigma^2) \Rightarrow \bar{x} \in N(\mu, \sigma^2/n) \Rightarrow z = \frac{\bar{x}-\mu}{\sqrt{\sigma^2/n}} \in N(0,1)$$

$$z \in N(0,1) \Rightarrow \Pr\left(-z_{1-\alpha/2} \le z \le z_{1-\alpha/2}\right) = 1-\alpha$$

$z_{1-\alpha/2}$ is the upper $\left(1-\frac{\alpha}{2}\right)$ critical point of the standard normal distribution (tabulated data)

47

Case 1: Number of Measurements >= 30



$$\Pr\left(-z_{1-\alpha/2} \le z \le z_{1-\alpha/2}\right) = 1-\alpha$$

$$\Pr\left(-z_{1-\alpha/2} \le \frac{\bar{x}-\mu}{\sqrt{\sigma^2/n}} \le z_{1-\alpha/2}\right) = 1-\alpha$$

$$\Pr\left(\bar{x} - z_{1-\alpha/2}\sqrt{\sigma^2/n} \le \mu \le \bar{x} + z_{1-\alpha/2}\sqrt{\sigma^2/n}\right) = 1-\alpha$$

48

Case 1: Number of Measurements >= 30

$$\Pr\left(\bar{x} - z_{1-\alpha/2}\sqrt{\sigma^2/n} \le \mu \le \bar{x} + z_{1-\alpha/2}\sqrt{\sigma^2/n}\right) = 1-\alpha$$

Since $n \ge 30$, we can approximate the varience σ^2 with the sample varience s^2

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

$$\Pr\left(\bar{x} - z_{1-\alpha/2}\sqrt{s^2/n} \le \mu \le \bar{x} + z_{1-\alpha/2}\sqrt{s^2/n}\right) = 1-\alpha$$

$$c_1 = \bar{x} - z_{1-\alpha/2}\sqrt{s^2/n}$$

$$c_2 = \bar{x} + z_{1-\alpha/2}\sqrt{s^2/n}$$

Case 1: Number of Measurements >= 30

- We found an interval $[c_1,c_2]$ such that

$$c_1 = \bar{x} - z_{1-\alpha/2}\sqrt{s^2/n}$$
$$c_2 = \bar{x} + z_{1-\alpha/2}\sqrt{s^2/n}$$

$$\Pr[c_1 \le \mu \le c_2] = 1-\alpha$$

- The interval $[c_1,c_2]$ is an *approximate* $100(1-\alpha)\%$ confidence interval (CI) for the mean μ (an interval estimate of μ)
- The larger *n* is, the better the estimate.

Case 1: Number of Measurements < 30

- Problem: Cannot assume that the sample variance provides a good estimate of the population variance.

However, since $x_i \in N(\mu, \sigma^2)$ it can be shown that

$$z = \frac{\bar{x} - \mu}{\sqrt{s^2 / n}}$$ has a Student ***t*** distribution with (n-1) d.f.

- An *exact* 100(1-α) CI for μ is then given by

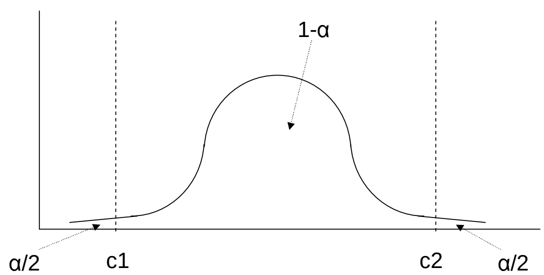
$$c_1 = \bar{x} - t_{1-\alpha/2;n-1}\sqrt{s^2/n}$$
$$c_2 = \bar{x} + t_{1-\alpha/2;n-1}\sqrt{s^2/n}$$

$t_{1-\alpha/2;n-1}$ is the upper $\left(1 - \frac{\alpha}{2}\right)$ critical point of the t distr. with n-1 d.f. (tabulated)

51

The Student *t* distribution

- The *t* distribution is similar to the Normal distribution
- They are both bell-shaped and symmetric around the mean
- The *t* distribution tends to be more "spread out" (has greater variance)
- The *t* distribution becomes the same as the standard normal distribution as *n* tends to infinity



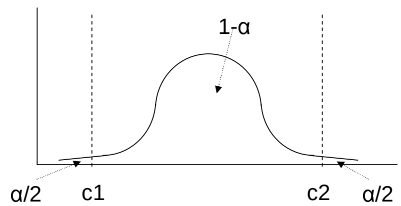
52

Example

Experiment	Measured value
1	8.0 s
2	7.0 s
3	5.0 s
4	9.0 s
5	9.5 s
6	11.3 s
7	5.2 s
8	8.5 s

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} = 7.94$$

$$s = \text{sample standard deviation} = 2.14$$



- 90% CI → 90% chance actual value in interval
- 90% CI → α = 0.10
  - 1 – (α / 2) = 0.95
- *n* = 8 → 7 degrees of freedom

53

Example (cont.)

90 % Confidence Interval

$$a = 1 - \alpha / 2 = 1 - 0.10/2 = 0.95$$

$$t_{a;n-1} = t_{0.95;7} = 1.895$$

$$c_1 = 7.94 - \frac{1.895(2.14)}{\sqrt{8}} = 6.5$$

$$c_2 = 7.94 + \frac{1.895(2.14)}{\sqrt{8}} = 9.4$$

	a		
n	0.90	0.95	0.975
…	…	…	…
5	1.476	2.015	2.571
6	1.440	1.943	2.447
7	1.415	1.895	2.365
…	…	…	…
∞	1.282	1.645	1.960

95 % Confidence Interval

$$a = 1 - \alpha / 2 = 1 - 0.10/2 = 0.975$$

$$t_{a;n-1} = t_{0.975;7} = 2.365$$

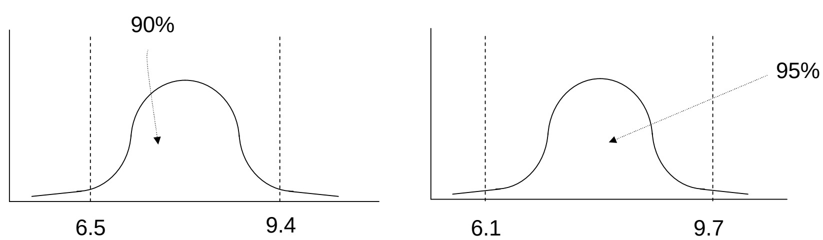
$$c_1 = 7.94 - \frac{2.365(2.14)}{\sqrt{8}} = 6.1$$

$$c_2 = 7.94 + \frac{2.365(2.14)}{\sqrt{8}} = 9.7$$

	a		
n	0.90	0.95	0.975
…	…	…	…
5	1.476	2.015	2.571
6	1.440	1.943	2.447
7	1.415	1.895	2.365
…	…	…	…
∞	1.282	1.645	1.960

What Does it Mean?

- 90% CI = [6.5, 9.4]
  - 90% chance mean value is between 6.5, 9.4
- 95% CI = [6.1, 9.7]
  - 95% chance mean value is between 6.1, 9.7
- Why is interval wider when we are more confident?



55

What If Errors Not Normally Distributed?

- Can use the Central Limit Theorem (CLT)

  *Sum of a "large number" of values from **any** distribution will be Normally (Gaussian) distributed.*
- "Large number" typically assumed to be >≈ 6 or 7.
- If n >= 30 the approximate CI based on the normal distribution remains valid and can be used.

$$[\bar{x} - z_{1-\alpha/2}\sqrt{s^2/n}\,,\, \bar{x} + z_{1-\alpha/2}\sqrt{s^2/n}]$$

- If n < 30, we can normalize the measurements by grouping them info groups of 6 or more and using their averages as input data.
- We can now use the CI based on the *t*-distribution:

$$[\bar{x} - t_{1-\alpha/2;n-1}\sqrt{s^2/n}, \bar{x} + t_{1-\alpha/2;n-1}\sqrt{s^2/n}]$$

56

What If Errors Not Normally Distributed? (2)

- What if impossible to measure the event of interest directly, e.g. duration of the event too short.
- Measure the duration of several repetitions of the event and calculate the average time for one occurrence.

$$\bar{x}_j = T_j / m_j \qquad T_j \text{ is the time required to repeat event } m_j \text{ times}$$

$$\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$$

- Now apply the CI formula to the *n* mean values.
- The normalization has a penalty!
  - Number of measurement reduced → loss of information
- Provides CI for mean value of the aggregated events, not the individual events themselves!
- Tends to smooth out the variance

57

How Many Measurements?

- Width of interval inversely proportional to √n
- Want to find how many measurements needed to obtain a CI with a given width

$$(c_1, c_2) = (1 \mp e)\bar{x} = \bar{x} \mp z_{1-\alpha/2} \frac{s}{\sqrt{n}}$$

$$z_{1-\alpha/2} \frac{s}{\sqrt{n}} = \bar{x}e \quad \Rightarrow \quad n = \left(\frac{z_{1-\alpha/2}\, s}{\bar{x}e} \right)^2$$

- But *n* depends on knowing mean and standard deviation
  - Estimate $\bar{x}$ and *s* with small number of measurements
  - Use the estimates to find *n* needed for desired interval width

58

Example

- Assume that based on 30 measurements we found:
  - Mean = 7.94 s
  - Standard deviation = 2.14 s
- Want 90% confidence true mean is within 3.5% of measured mean?
  - *α = 0.90*
  - *(1-α/2) = 0.95*
  - *Error = ± 3.5%*
  - *e = 0.035*

$$n = \left(\frac{z_{1-\alpha/2} s}{\bar{x} e} \right)^2 = \left(\frac{1.895(2.14)}{0.035(7.94)} \right) = 212.9$$

- 213 measurements

→ 90% chance true mean is within ± 3.5% interval

59

Confidence Intervals for Proportions

- Assume we are counting the number of times several events occur and want to estimate the fraction of time each event occurs?
- Can model this using a binomial distribution
  - *p* = Pr(success) in *n* trials of binomial experiment
  - Need a confidence interval for *p*
- Let *m* be the number of successes
- *m* has a binomial distribution with parameters *p* and *n*

$$E[m] = pn \qquad \sigma^2[m] = p(1-p)n$$

Can estimate p using the sample proportion $\bar{p} = m/n$

If $\mathrm{pn} \geq 10$, can approximate the binomial distribution

with Gaussian $N(pn, p(1-p)n) \;\Rightarrow\; m \approx N(pn, \bar{p}(1-\bar{p})n)$

60

Confidence Intervals for Proportions (2)

$$m \approx N(pn, \overline{p}(1-\overline{p})n)$$

$$\Pr\left(-z_{1-\alpha/2} \le \frac{m - pn}{\sqrt{\overline{p}(1-\overline{p})n}} \le z_{1-\alpha/2}\right) \approx 1-\alpha$$

$$\Pr\left(m - z_{1-\alpha/2}\sqrt{\overline{p}(1-\overline{p})n} \le pn \le m + z_{1-\alpha/2}\sqrt{\overline{p}(1-\overline{p})n}\right) \approx 1-\alpha$$

$$\Pr\left(\overline{p} - z_{1-\alpha/2}\sqrt{\frac{\overline{p}(1-\overline{p})}{n}} \le p \le \overline{p} + z_{1-\alpha/2}\sqrt{\frac{\overline{p}(1-\overline{p})}{n}}\right) \approx 1-\alpha$$

$$\Rightarrow c_1 = \overline{p} - z_{1-\alpha/2}\sqrt{\frac{\overline{p}(1-\overline{p})}{n}} \qquad c_2 = \overline{p} + z_{1-\alpha/2}\sqrt{\frac{\overline{p}(1-\overline{p})}{n}}$$

61

Example

- How much time does processor spend in OS?
- Interrupt every 10 ms and increment counters
  - n = number of interrupts
  - m = number of interrupts when PC within OS
- Run for 1 minute
  - n = 6000, m = 658

$$(c_1, c_2) = \overline{p} \mp z_{1-\alpha/2}\sqrt{\frac{\overline{p}(1-\overline{p})}{n}}$$

$$= 0.1097 \mp 1.96\sqrt{\frac{0.1097(1-0.1097)}{6000}} = (0.1018, 0.1176)$$

- Can claim with 95% confidence that the processor spends 10.2-11.8% of its time in OS

62

How Many Measurements?

$$(1-e)\overline{p} = \overline{p} - z_{1-\alpha/2}\sqrt{\frac{\overline{p}(1-\overline{p})}{n}} \Rightarrow \quad e\overline{p} = z_{1-\alpha/2}\sqrt{\frac{\overline{p}(1-\overline{p})}{n}}$$

$$\Rightarrow \quad n = \frac{z^2_{1-\alpha/2}\overline{p}(1-\overline{p})}{(e\overline{p})^2}$$

- Example: How long to run OS experiment?
- Want 95% confidence interval with ± 0.5% width

$$\alpha = 0.05 \qquad e = 0.005$$

$$\overline{p} = m/n = 658/6000 = 0.1097$$

$$n = \frac{z^2_{1-\alpha/2}\overline{p}(1-\overline{p})}{(e\overline{p})^2} = \frac{(1.960)^2(0.1097)(1-0.1097)}{[0.005(0.1097)]^2} = 1{,}247{,}102$$

- 10 ms interrupts → 3.46 hours

63

Further Reading

- R. K. Jain, *"The Art of Computer Systems Performance Analysis : Techniques for Experimental Design, Measurement, Simulation, and Modeling"*, Wiley (April 1991), ISBN: 0471503363, 1991
- Kishor Trivedi, *"Probability and Statistics with Reliability, Queuing, and Computer Science Applications"*, John Wiley and Sons, ISBN 0-471-33341-7, New York, 2001
- *Electronic Statistics Textbook* http://www.statsoft.com/textbook/stathome.html
- See http://www.arctic.umn.edu/perf-book/bookshelf.shtml
- N.C. Barford, "*Experimental Measurements: Precision, Error, and Truth"* (Second Edition), John Wiley and Sons, New York, 1985
- John Mandel, *"The Statistical Analysis of Experimental Data"*, Interscience Publishers, a division of John Wiley and Sons, New York, 1964.

64

Further Reading (cont.)

- P.J.Fleming and J.J.Wallace, *“How Not To Lie With Statistics: The Correct Way To Summarize Benchmark Results”,* Communications of the ACM, Vol.29, No.3, March 1986, pp. 218-221
- James Smith, *„Characterizing Computer Performance with a Single Number“,* Communications of the ACM, October 1988, pp.1202-1206
- Patterson and Hennessy, *Computer Architecture: The Hardware/Software Approach*, Morgan Kaufman Publishers, San Francisco, CA.
- Cragon, H., *Computer Architecture and Implementation*, Cambridge University Press, Cambridge, UK.
- Mashey, J.R., *War of the benchmark menas: Time for a truce,* Computer Architecture News, 32 (1), 4, 2004.
- John, L.K., *More on finding a single number to indicate overall performance of a benchmark suite*, Computer Architecture News, 32 (1) 3, 2004.

65

Exercise 1

- Many compilers have several different levels of optimization that can be selected to improve performance. Using some appropriate benchmark program, determine whether these different optimization levels actually make a statistically significant difference in the overall execution time of this program. Run the program 4 times for each of the different optimizations. Use a 90% and a 99% confidence interval to determine whether each of the optimizations actually improves the performance. Explain your results.

66