Metrics and Review of Basic Statistics

CS 239

Experimental Methodologies for System Software

Peter Reiher

April 5, 2007

Introduction

- Metrics
- Why are we talking about statistics?
- Important statistics concepts
- Indices of central tendency
- Summarizing variability

Metrics

- A metric is a measurable quantity
- For our purposes, one whose value describes an important phenomenon
- Most of performance evaluation is about properly gathering metrics

Common Types of Metrics

- Duration/ response time
  - How long did the simulation run?
- Processing rate
  - How many transactions per second?
- Resource consumption
  - How much disk is currently used?
- Error rates
  - How often did the system crash?
- What metrics can we use to describe security?

Examples of Response Time

- Time from keystroke to echo on screen
- End-to-end packet delay in networks
- OS bootstrap time
- Leaving UCLA to getting on 405

Some Measures of Response Time

- *Response time:* request-response interval
  - Measured from end of request
  - Ambiguous: beginning or end of response?
- *Reaction time:* end of request to start of processing
- *Turnaround time:* start of request to end of response

Processing Rate

- How much work is done per unit time?
- Important for:
  - Provisioning systems
  - Comparing alternative configurations
  - Multimedia

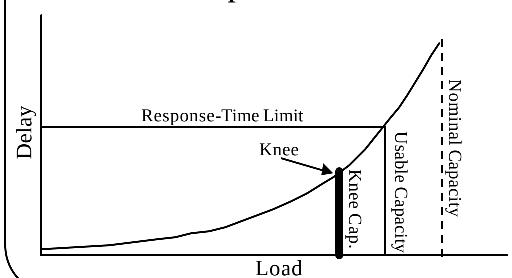
Examples of Processing Rate

- Bank transactions per hour
- Packets routed per second
- Web pages crawled per night

Common Measures of Processing Rate

- *Throughput:* requests per unit time: MIPS, MFLOPS, Mb/s, TPS
- *Nominal capacity:* theoretical maximum: bandwidth
- *Knee capacity:* where things go bad
- *Usable capacity:* where response time hits a specified limit
- *Efficiency:* ratio of usable to nominal cap.

Nominal, Knee, and Usable Capacities



Resource Consumption

- How much does the work cost?
- Used in:
  - Capacity planning
  - Identifying bottlenecks
- Also helps to identify “next” bottleneck

Examples of Resource Consumption

- CPU non-idle time
- Memory usage
- Fraction of network bandwidth needed
- How much of your salary is paid for rent

Measures of Resource Consumption

- *Utilization:* $\int_0^t u(t)\,dt$

  where ***u*(*t*)** is instantaneous resource usage

  – Useful for memory, disk, etc.
- If ***u*(*t*)** is always either 1 or 0, reduces to *busy time* or its inverse, *idle time*
  – Useful for network, CPU, etc.

Error Metrics

- Failure rates
- Probability of failures
- Time to failure

Examples of Error Metrics

- Percentage of dropped Internet packets
- ATM down time
- Lifetime of a component
- Wrong answers from IRS tax preparation hotline

Measures of Errors

- *Reliability:* P(error) or *Mean Time Between Errors* (MTBE)
- *Availability:*
  – *Downtime:* Time when system is unavailable, may be measured as *Mean Time to Repair* (MTTR)
  – *Uptime:* Inverse of downtime, often given *as Mean Time Between Failures* (MTBF/MTTF)

Security Metrics

- A difficult problem
- Often no good metrics to express security goals and achievements
  – Equally bad, some definable metrics are impossible to measure
- Some failure metrics are applicable
  – Expected time to break a cipher

Choosing What to Measure

- Core question in any performance study
- Pick metrics based on:
  – Completeness
  – (Non-)redundancy
  – Variability
  – Feasibility

Completeness

- Must cover everything relevant to problem
  - Don't want awkward questions at conferences!
- Difficult to guess everything *a priori*
  - Often have to add things later

Redundancy

- Some factors are functions of others
- Measurements are expensive
- Look for minimal set
- Again, often an interactive process

Variability

- Large variance in a measurement makes decisions impossible
- Repeated experiments can reduce variance
  - Very expensive
  - Can only reduce it by a certain amount
- Better to choose low-variance measures to start with

Feasibility

- Some things are easy to measure
- Others are hard
- A few are impossible
- Choose metrics you can actually measure
- But beware of the "drunk under the streetlamp" phenomenon

Variability and Performance Measurements

- Performance of a system is often complex
  - Perhaps not fully explainable
- One result is variability in most metric readings
- Good performance measurement takes this into account

An Example

- 10 pings from UCLA to MIT Tuesday night
- Each took a different amount of time (expressed in msec):

84.0	84.9	84.5	84.3	84.5
84.5	84.8	86.8	84.1	84.5

- How do we understand what this says about how long a packet takes to get from LA to Boston?

How to Get a Handle on Variability?

- If something we're trying to measure varies from run to run, how do we express its behavior?
- That's what statistics is all about
- Which is why a good performance analyst needs to understand them

Some Basic Statistics Concepts

- Independence of events
- Random variables
- Cumulative distribution functions (CDFs)

Independent Events

- Events are independent if:
  - Occurrence of one event doesn't affect probability of other
- Examples:
  - Coin flips
  - Inputs from separate users
  - "Unrelated" traffic accidents

Non-Independent Events

- Not all events are independent
- Second person accessing a web page might get it faster than the first
  - Or than someone asking for it the next day
- Kids requesting money from their parents
  - Sooner or later the wallet is empty

Random Variables

- Variable that takes values probabilistically
  - Not necessarily just any value, though
- Variable usually denoted by capital letters, particular values by lowercase
- Examples:
  - Number shown on dice
  - Network delay
  - CS239 attendance

Cumulative Distribution Function (CDF)

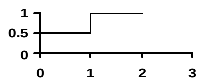
- Maps a value *a* of random variable *x* to probability that the outcome is less than or equal to *a*:

$$F_x(a) ? P(x ? a)$$

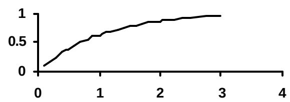
- Valid for discrete and continuous variables
- Monotonically increasing
- Easy to specify, calculate, measure

CDF Examples

- Coin flip (T = 1, H = 2):



- Exponential packet interarrival times:



Probability Density Function (pdf)

- A "relative" of CDF
- Derivative of (continuous) CDF:

$$f(x) \; ? \; \frac{dF(x)}{dx}$$

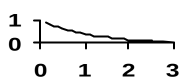
- Useful to find probability of a range:

$$P(x_1 \; ? \; x \; ? \; x_2) \; ? \; F(x_2) \; ? \; F(x_1)$$

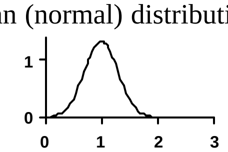
$$? \; ?_{x_1}^{x_2} f(x)dx$$

Examples of pdf

- Exponential interarrival times:



- Gaussian (normal) distribution:



Probability Mass Function (pmf)

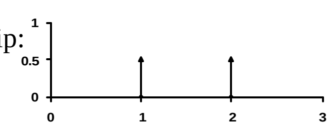
- PDF doesn't exist for discrete random variables
  - Because their CDF not differentiable
- pmf instead: $f(x_i) = p_i$ where p_i is the probability that x will take on value x_i

$$P(x_1 \; ? \; x \; ? \; x_2) \; ? \; F(x_2) \; ? \; F(x_1)$$

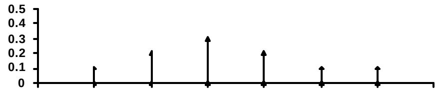
$$? \; \sum_{x_1 ? x_i ? x_2} p_i$$

Examples of pmf

- Coin flip:



- Typical CS grad class size:



Summarizing Data With a Single Number

- Most condensed form of presentation of set of data
- Usually called the **average**
  - Average isn't necessarily the mean
- More formal term is index of central tendency
- Must be representative of a major part of the data set

Indices of Central Tendencies

- Specify center of location of the distribution of the observations in the sample
- Common examples:
  - Mean
  - Median
  - Mode

Sample Indices

- The mean (or other index) is the mean of all possible elements of random variable
- You usually don't test them all
- The mean of the ones you test is the sample mean
- Sample mean ? mean
  - For a different set of samples, you get a different sample mean

An Example

If we assign value 1 to red and value 2 to blue, mean value in jar is 1.5

Sample mean is 1.75

Many different sample means would be possible

Sample Mean

- Take sum of all observations
- Divide by the number of observations
- More affected by outliers than median or mode
- Mean is a linear property
  - Mean of sum is sum of means
  - Not true for median and mode

Sample Median

- Sort the observations
  - In increasing order
- Take the observation in the middle of the series
- More resistant to outliers
  - But not all points given "equal weight"

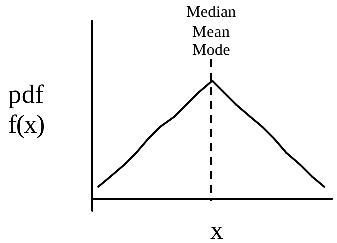
Sample Mode

- Plot a histogram of the observations
  - Using existing categories
  - Or dividing ranges into buckets
- Choose the midpoint of the bucket where the histogram peaks
  - For categorical variables, the most frequently occurring
- Effectively ignores much of the sample

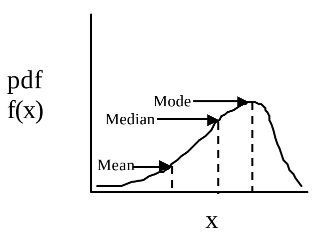
Characteristics of Mean, Median, and Mode

- Mean and median always exist and are unique
- Mode may or may not exist
  - If there is a mode, there may be more than one
- Mean, median and mode may be identical
  - Or may all be different
  - Or some of them may be the same

Mean, Median, and Mode Identical



Median, Mean, and Mode All Different



So, Which Should I Use?

- Depends on characteristics of the metric
  - If data is categorical, use mode
  - If a total of all observations makes sense, use mean
  - If not, and the distribution is skewed, use median
  - Otherwise, use mean
- But think about what you're choosing

Some Examples

- Most-used resource in system
  - Mode
  - Ficus replica that receives the most original updates
- Interarrival times
  - Mean
  - Time between file access requests in Conquest
- Load
  - Median
  - Number of packets a DefCOM classifier handles per second

Don't Always Use the Mean

- Means are often overused and misused
  - Means of significantly different values
  - Means of highly skewed distributions
  - Multiplying means to get mean of a product
    - Only works for independent variables
- Errors in taking ratios of means

Geometric Means

- An alternative to the arithmetic mean

$$\dot{x} = \left(\prod_{i=1}^{n} x_i \right)^{1/n}$$

- Use geometric mean if product of observations makes sense

Good Places To Use Geometric Mean

- Layered architectures
- Performance improvements over successive versions
- Average error rate on multihop network path

Harmonic Mean

- Harmonic mean of sample $\{x_1, x_2, ..., x_n\}$ is

$$\dot{x} = \frac{n}{1/x_1 + 1/x_2 + \dots + 1/x_n}$$

- Use when arithmetic mean of $1/x_1$ is sensible

Example of Using Harmonic Mean

- When working with MIPS numbers from a single benchmark
  - Since MIPS calculated by dividing constant number of instructions by elapsed time

$$x_i = \frac{m}{t_i}$$

- Not valid if different m's (e.g., different benchmarks for each observation)

Means of Ratios

- Given *n* ratios, how do you summarize them?
- Can't always just use harmonic mean
  - Or similar simple method
- Consider numerators and denominators

Considering Mean of Ratios: Case 1

- Both numerator and denominator have physical meaning
- Then the average of the ratios is the ratio of the averages

Example: CPU Utilizations

Measurement Duration	CPU Busy (%)
1	40
1	50
1	40
1	50
100	20
Sum	200 %
Mean?	

CS 239, Spring 2007 Lecture 2 Page 55

Mean for CPU Utilizations

Measurement Duration	CPU Busy (%)
1	40
1	50
1	40
1	50
100	20
Sum	200 %
Mean?	Not 40%

CS 239, Spring 2007 Lecture 2 Page 56

Properly Calculating Mean For CPU Utilization

- Why not 40%?
- Because CPU Busy percentages are ratios
  - And their denominators aren't comparable
- The duration-100 observation must be weighed more heavily than the duration-1 observations

CS 239, Spring 2007 Lecture 2 Page 57

So What Is the Proper Average?

- Go back to the original ratios

$$\text{Mean CPU Utilization} = \frac{0.45 + 0.50 + 0.40 + 0.50 + 0.20}{1 + 1 + 1 + 1 + 100}$$

$$= 21\ \%$$

CS 239, Spring 2007 Lecture 2 Page 58

Considering Mean of Ratios: Case 1a

- Sum of numerators has physical meaning, denominator is a constant
- Take the arithmetic mean of the ratios to get the mean of the ratios

CS 239, Spring 2007 Lecture 2 Page 59

For Example,

- What if we calculated the CPU utilization from the last example using only the four duration 1 measurements?
- Then the average is

$$\frac{1}{4}\left(\frac{.40}{1} + \frac{.50}{1} + \frac{.40}{1} + \frac{.50}{1}\right) = .45$$

CS 239, Spring 2007 Lecture 2 Page 60

Considering Mean of Ratios: Case 1b

- Sum of the denominators has a physical meaning, numerator is a constant
- Take the harmonic mean of the ratios

Considering Mean of Ratios: Case 2

- The numerator and denominator are expected to have a multiplicative, near-constant property

  $a_i = c\ b_i$
- Estimate c with geometric mean of a_i/b_i

Example for Case 2

- An optimizer reduces the size of code
- What is the average reduction in size, based on its observed performance on several different programs?
- Proper metric is percent reduction in size
- And we're looking for a constant *c* as the average reduction

Program Optimizer Example, Continued

	Code Size		
Program	Before	After	Ratio
BubbleP	119	89	.75
IntmmP	158	134	.85
PermP	142	121	.85
PuzzleP	8612	7579	.88
QueenP	7133	7062	.99
QuickP	184	112	.61
SieveP	2908	2879	.99
TowersP	433	307	.71

Why Not Use Ratio of Sums?

- Why not add up pre-optimized sizes and post-optimized sizes and take the ratio?
  - Benchmarks of non-comparable size
  - No indication of importance of each benchmark in overall code mix
- When looking for constant factor, not the best method

So Use the Geometric Mean

- Multiply the ratios from the 8 benchmarks
- Then take the 1/8 power of the result

$\bar{x}$? ?.75*.85*.85*.88*.99*.61*.99*.71?$^{1/8}$

= .82