Performance Modeling of Computer Systems and Networks

Prof. Vittoria de Nitto Personè

Sample statistics

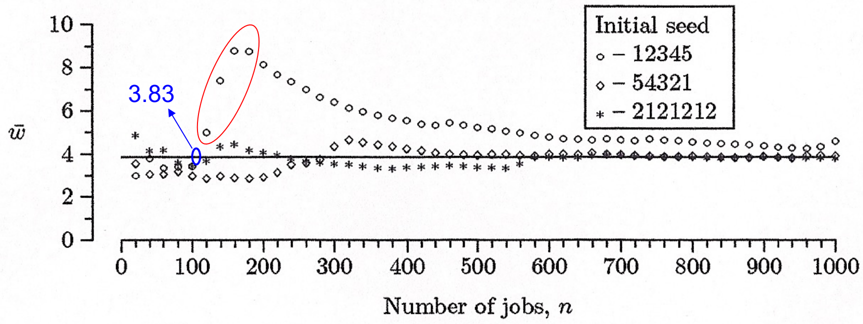
Università degli studi di Roma Tor Vergata
Department of Civil Engineering and Computer Science Engineering

1

- Simulation involves *a lot* of data
- Must "compress" the data into meaningful statistics
- Collected data is a *sample* from a much larger *population*
- Two types of statistical analysis
  - "Within-the-run"
  - "Between-the-runs" (replication)
- Essence of statistics: analyze a sample and draw inferences

2

- The accumulated average wait was printed every 20 job



- The convergence of w is slow, erratic, and dependent on the initial seed

arrival and service processes are *uncoupled*

stream 0 for arrivals, stream 1 for services

for 10025 jobs
average interarrival time = 1.99
average wait .... = 3.92
average delay .... = 2.41
average service time .... = 1.50
average # in the node ... = 1.96
average # in the queue .. = 1.21
utilization .... = 0.75

stream 0 for arrivals, stream 2 for services (or e.g. stream 128 to get more separation)

for 10025 jobs
average interarrival time = 1.99
average wait .... = 3.86
average delay .... = 2.36
average service time .... = 1.50
average # in the node ... = 1.93
average # in the queue .. = 1.18
utilization .... = 0.75

Theoretical values

$\bar{r}$	$\bar{w}$	$\bar{d}$	$\bar{s}$	$\bar{l}$	$\bar{q}$	$\bar{x}$
2.00	3.83	2.33	1.50	1.92	1.17	0.75

Sample Mean and Standard Deviation

Consider a sample $x_1, x_2, \ldots, x_n$ (continuous or discrete), let us define:

- *sample mean* $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
- *sample variance* $s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$
- *sample standard deviation* $s = \sqrt{s^2}$
- *coefficient of variation* $s / \bar{x}$

- *mean*: a measure of central tendency
- *variance* and *deviation*: measures of dispersion about the mean

easier math
(no square root)

same units as data,
mean

- note that *coefficient of variation (CV)* is unit-less, but a common shift in data changes the CV

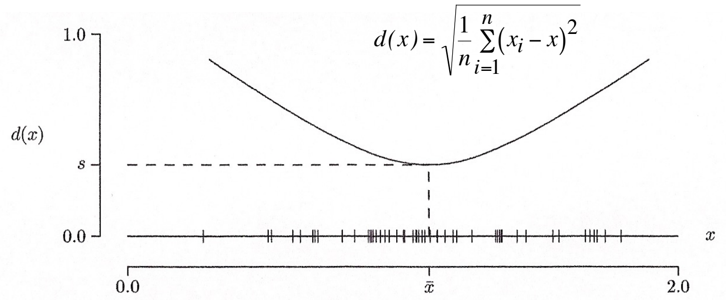
Relating the mean and standard deviation

Consider the root-mean-square (rms) function

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

- $d(x)$ measures dispersion about any value x
- the mean $\bar{x}$ gives the smallest possible value for $d(x)$ (Theorem 4.1.1)
- The standard deviation s is that smallest value

50 samples from program buffon

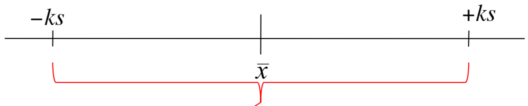


$$d(x) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - x)^2}$$

$\bar{x} \cong 1.095, s \cong 0.354$

Chebyshev's inequality

Consider the number of points that lie within k standard deviations of the mean



- Points farthest from the mean make the most contribution to *s*

Define the set $S_k = \{x_i | \bar{x} - ks < x_i < \bar{x} + ks\}$

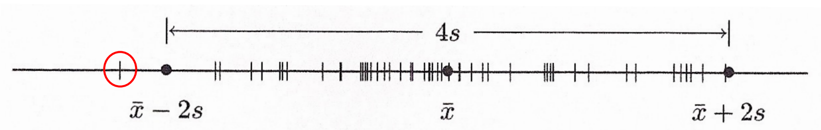
Let $p_k = |S_k|/n$ be the proportion of x_i within $\pm ks$ of $\bar{x}$

$$p_k \geq 1 - \frac{1}{k^2} \qquad (k > 1)$$

Chebyshev's inequality

- for *any* sample, at least 75% of the points lie within $\pm 2s$ of $\bar{x}$
- for k=2, the inequality is very conservative:

  typically 95% lie within $\pm 2s$ *of* $\bar{x}$
- $\bar{x} \pm 2s$ defines the "effective width" of a sample



- most (but not all) points will lie in this interval
- *outliers* should be viewed with suspicion

Linear data transformations

- Often need to convert to different units after data has been collected
- let x'_i be the "new data": $x'_i = ax_i + b$
- *sample mean*

$$\bar{x}' = \frac{1}{n}\sum_{i=1}^{n} x'_i = \frac{1}{n}\sum_{i=1}^{n}(ax_i + b) = \frac{a}{n}\left(\sum_{i=1}^{n} x_i\right) + b = a\bar{x} + b$$

- *sample variance* $\quad (s')^2 = \frac{1}{n}\sum_{i=1}^{n}(x'_i - \bar{x}')^2 = \ldots = a^2 s^2$
- *sample standard deviation* $\quad s' = |a| s$

Examples of Linear Data Transformations

- suppose $x_1, x_2, \ldots, x_n$ measured in seconds
  - to convert to minutes, let $x'_i = x_i/60$

  (a=1/60, b=0)

$$\bar{x}' = \frac{45}{60} = 0.75 \qquad s' = \frac{15}{60} = 0.25 \quad \text{(minutes)}$$

- *standardize* data (a=1/s, b= $-\bar{x}/s$) $\quad x'_i = \frac{1}{s}x_i - \frac{\bar{x}}{s} \qquad x'_i = \frac{x_i - \bar{x}}{s}$

  Then $\quad \bar{x}' = 0 \qquad s' = 1$

  Used to avoid problems with very large (or small) valued samples

Nonlinear data transformations

- usually involves a Boolean (two-state) outcome
- the *value* of x_i is not as important as the *effect*
- let A be a fixed set; then

$$x'_i = \begin{cases} 1 & x_i \in \mathrm{A} \\ 0 & \text{otherwise} \end{cases}$$

- let p be the proportion of x_i that fall in A:

$$p = \frac{\text{the number of } x_i \text{ in A}}{n}$$

then

$$\bar{x}' = p \qquad s' = \sqrt{p(1-p)}$$

Examples of Nonlinear Data Transformations

For the single server service queue

- let $x_i = d_i$ be the delay for job i
- let A = R$^+$, then x_i'=1 iff d_i>0
- from exerc. 1.2.3 proportion of job delayed is p = 0.723
- then $\bar{x}' = 0.723$ and $s = \sqrt{(0.723)(0.277)} = 0.448$

Computational considerations

Consider the sample standard deviation equation

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

Requires two passes through the data:

1. Compute the mean $\bar{x}$
2. Compute the squared differences about $\bar{x}$

Must store or re-create the entire sample!
bad when n is large!

The conventional one-pass Algorithm

Consider the sample standard deviation equation

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

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

by separating and simplifying

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

round-off error, overflow

Welford's one-pass algorithm

- running sample mean until *i*

$$\bar{x}_i = \frac{1}{i}\left(x_1 + x_2 + \ldots + x_i\right)$$

- running sample sum of squared deviations until *i*

$$v_i = \left(x_1 - \bar{x}_i\right)^2 + \left(x_2 - \bar{x}_i\right)^2 + \ldots + \left(x_i - \bar{x}_i\right)^2$$

- $\bar{x}_i$ and v_i can be computed recursively $\left(\bar{x}_0 = 0, v_0 = 0\right)$:

$$\bar{x}_i = \bar{x}_{i-1} + \frac{1}{i}\left(x_i - \bar{x}_{i-1}\right)$$

$$v_i = v_{i-1} + \left(\frac{i-1}{i}\right)\left(x_i - \bar{x}_{i-1}\right)^2$$

- $\bar{x}_n$ is the sample mean, v_n / n is the variance

Welford's Algorithm
(program uvs)

- No *a priori* knowledge of the sample size *n* required
- Less prone to accumulated round-off error

$$\bar{x}_i = \bar{x}_{i-1} + \frac{1}{i}\left(x_i - \bar{x}_{i-1}\right)$$

```
n = 0;
x = 0.0;
v = 0.0;
while (more data ) {
    x = GetData();
    n++;
    d = x - x;
    v = v + d * d * (n - 1) / n;
    x = x + d / n;
}
s = sqrt(v / n);
return n, x, s;
```

Welford's Algorithm
(program `uvs`)

- No *a priori* knowledge of the sample size *n* required
- Less prone to accumulated round-off error

```
n = 0;
x = 0.0;
v = 0.0;
while (more data ) {
    x = GetData();
    n++;
    d = x - x;
    v = v + d * d * (n - 1) / n;
    x = x + d / n;
}
s = sqrt(v / n);
return n, x, s;
```

$$v_i = v_{i-1} + \left(\frac{i-1}{i}\right)(x_i - \bar{x}_{i-1})^2$$

Example

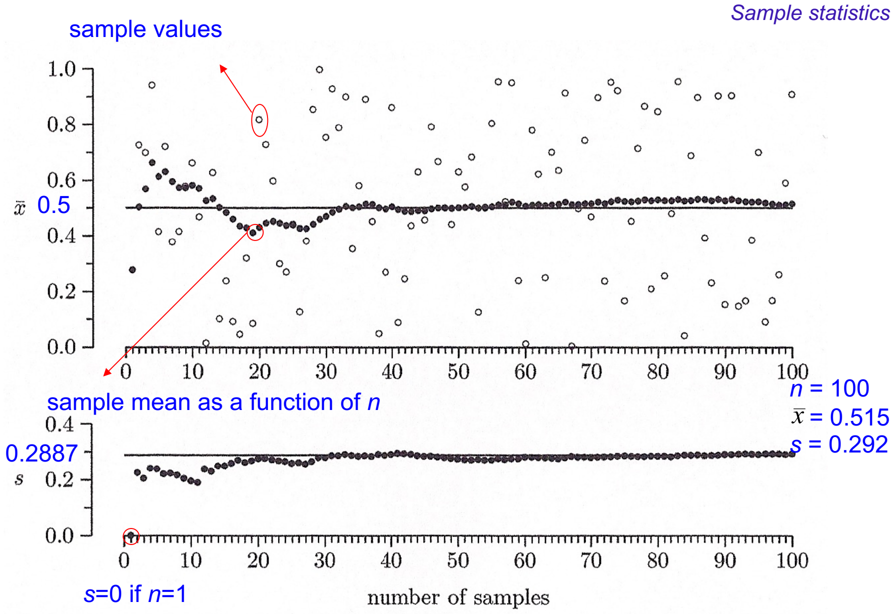
- let $x_1, x_2, \ldots, x_n$ be *Uniform*(a, b) random variates
- in the limit as $n \to \infty$

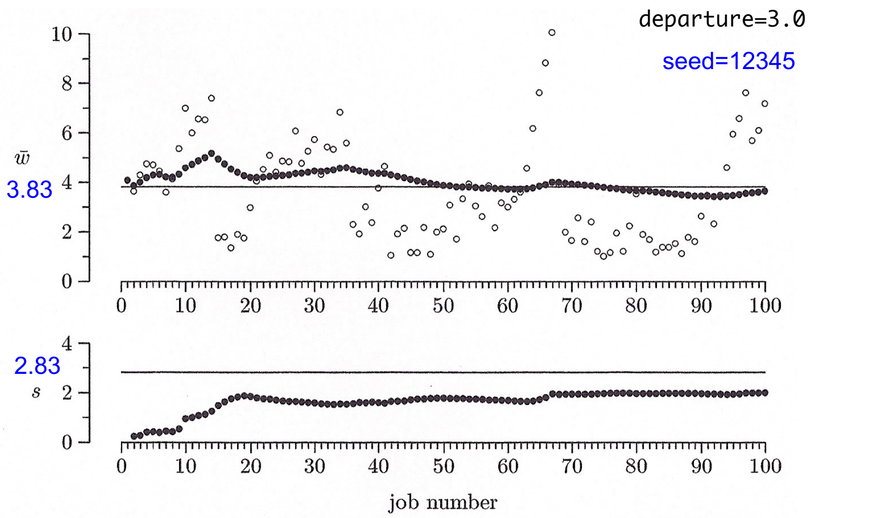
$$\bar{x} \to \frac{a+b}{2} \qquad s \to \frac{b-a}{\sqrt{12}}$$

- using *Uniform*(0, 1) $\bar{x}$ and s should converge to

$$\frac{0+1}{2} = 0.5 \qquad \frac{1-0}{\sqrt{12}} \cong 0.2887$$

- The convergence to theoretical values is not necessarily monotone


sample values
1.0
0.8
0.6
$\bar{x}$ 0.5
0.4
0.2
0.0
0 10 20 30 40 50 60 70 80 90 100
sample mean as a function of n
n = 100
$\bar{x}$ = 0.515
s = 0.292
0.4
0.2887
s 0.2
0.0
0 10 20 30 40 50 60 70 80 90 100
s=0 if n=1
number of samples


departure=3.0
seed=12345
10
8
6
$\bar{w}$
3.83 4
2
0
0 10 20 30 40 50 60 70 80 90 100
4
2.83
s 2
0
0 10 20 30 40 50 60 70 80 90 100
job number

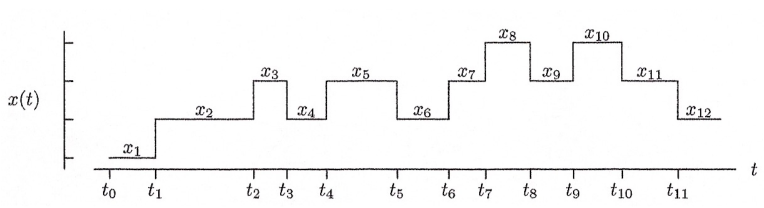
time-averaged sample statistics

- Let $x(t)$ be the sample path of a stochastic process for $0 < t < \tau$
- *Sample-path mean* $\bar{x} = \frac{1}{\tau} \int_0^\tau x(t)dt$
- *Sample-path variance* $s^2 = \frac{1}{\tau} \int_0^\tau (x(t) - \bar{x})^2 dt$
- *Sample-path standard deviation* $s = \sqrt{s^2}$
- One-pass equation for variance $s^2 = \left(\frac{1}{\tau} \int_0^\tau x^2(t)dt \right) - \bar{x}^2$

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

Computational considerations

- For DES, a sample path is *piecewise constant*
- Changes in the sample path occur at *event times* $t_0, t_1, \ldots$



- For computing statistics, integrals reduce to summations

Computational sample-path formulas

Consider a piecewise constant sample path

$$x(t) = \begin{cases} x_1 & t_0 < t \le t_1 \\ x_2 & t_1 < t \le t_2 \\ . & . \\ . & . \\ x_n & t_{n-1} < t \le t_n \end{cases}$$

- *Sample-path mean* $\bar{x} = \frac{1}{\tau}\int_0^\tau x(t)dt = \frac{1}{t_n}\sum_{i=1}^{n} x_i \delta_i$ with $\delta_i = t_i - t_{i-1}$ inter-event time
- *Sample-path variance*

$$s^2 = \frac{1}{\tau}\int_0^\tau (x(t) - \bar{x})^2 dt = \frac{1}{t_n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \delta_i = \left(\frac{1}{t_n}\sum_{i=1}^{n} x_i^2 \delta_i\right) - \bar{x}^2$$

Welford's sample path Algorithm

- based on the definitions

$$\bar{x}_i = \frac{1}{i}(x_1\delta_1 + x_2\delta_2 + \ldots + x_i\delta_i)$$
$$v_i = (x_1 - \bar{x}_i)^2\delta_1 + (x_2 - \bar{x}_i)^2\delta_2 + \ldots + (x_i - \bar{x}_i)^2\delta_i$$

- $\bar{x}_i$ is the sample-path mean of $x(t)$ for $t_0 \le t \le t_i$
- v_i / t_i is the sample-path variance
- $\bar{x}_i$ and v_i can be computed recursively $(\bar{x}_0 = 0, v_0 = 0)$

$$\bar{x}_i = \bar{x}_{i-1} + \frac{\delta_i}{t_i}(x_i - \bar{x}_{i-1})$$
$$v_i = v_{i-1} + \frac{\delta_i t_{i-1}}{t_i}(x_i - \bar{x}_{i-1})^2$$

Exercises

- Exercises: 4.1.7, 4.1.8