Performance evaluation

How to deal with experiments

Jean-Marc Vincent

MESCAL-INRIA Project
Laboratoire d'Informatique de Grenoble
Universities of Grenoble, France
{Jean-Marc.Vincent}@imag.fr

This work was partially supported by CAPES/COFECUB and STIC-AMSUD

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Outline

Introduction

Aim of this course

Give basic concepts of experimental design

- Statistical analysis of experimental data
- Modelling and parameters estimation
- Measurment in distributed systems and large scale systems
- Design of experiments

Interactive course : discussion about your own experiments

Outline

Application problems

Procedure TRC

```
int TRC(int *T,int p, int r)
{
int cpt=0;
int i, j,x;
if (r-p>0)
{
i=p; j=r;
while (j-i>0)
{
if (T[i] >= T[i+1]) { x=T[i]; T[i]=T[i+1]; T[i+1]=x; i++;}
else {x=T[j]; T[j]=T[i+1]; T[i+1]=x; j--;}
cpt++;
}
cpt+=TRC(T,p,i-1); cpt+=TRC(T,i+1,r);
}
return cpt;
}
```

First step

Specification

- The procedure TRC(T,i,j) sorts in place elements of an array T from index i to j included;

How to check it ?

Methods

- Exhaustive checking : enumerate all arrays and check them one by one
- Subset checking : use a representative subset of all arrays
- Statistical testing : generate uniformly an arbitrary subset size of array (confidence)

Example

Second step

Specification

- The procedure TRC(T,i,j) sorts in place elements of an array T from index i to j included;
- The procedure TRC(T,i,j) sorts n elements in $\mathcal{O}(n \log n)$ comparisons

How to evaluate it ?

Methods

- Exhaustive evaluation : enumerate all arrays and compute cost for each one
- Statistical evaluation : generate uniformly an arbitrary subset size of array (confidence) compute the empirical distribution of the cost, test the model $n \log n$

Example

Third step

Specification

- The procedure TRC(T,i,j) sorts in place elements of an array T from index i to j included;
- The procedure TRC(T,i,j) sorts n elements in $\mathcal{O}(n \log n)$ comparisons
- The procedure TRC(T,i,j) sorts n elements efficiently

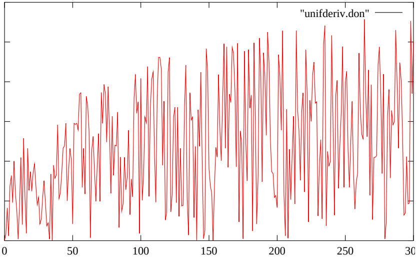
How to measure it ?

Methods

- Put probes inside the program/system/processor...
- Performance evaluation : generate uniformly an arbitrary subset size of array (confidence) measure the execution time, compute the empirical distribution of the execution time, fit with some models

Example

Sample analysis



Tendency analysis

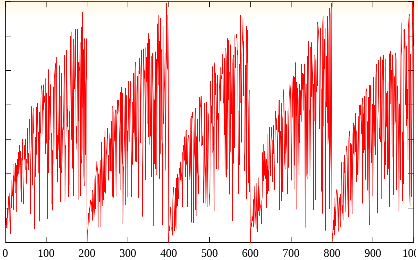
non homogeneous experiment

⇒ model the evolution of experiment

estimate and compensate tendency

explain why

Sample analysis (2)



Periodicity analysis

periodic evolution of the experimental environment ?

$\Rightarrow$ model the evolution of experiment

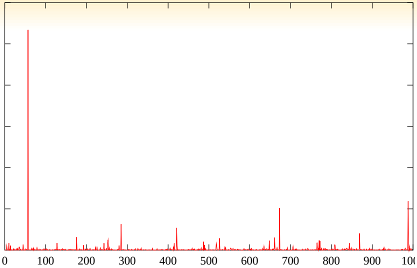
Fourier analysis of the sample

Integration on time (sliding window analysis) Danger : size of the window

Wavelet analysis

explain why

Sample analysis (3)



Non significant values

extraordinary behaviour of experimental environment

rare events with different orders of magnitude

$\Rightarrow$ threshold by value

Danger : choice of the threshold : indicate the rejection rate

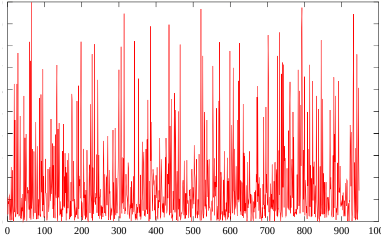
$\Rightarrow$ threshold by quantile

Danger : choice of the percentage : indicate the rejection value

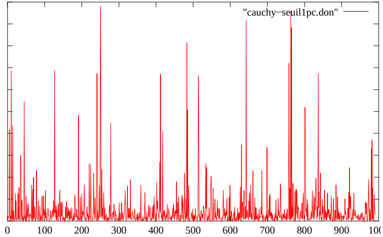
explain why

Sample analysis (4)

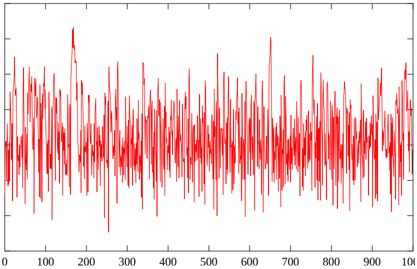
Threshold value : 10



Threshold percentage : 1%



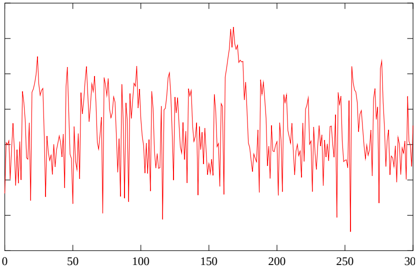
Sample analysis (5)



looks like correct experiments
Statistically independent
Statistically homogeneous

Sample analysis (5bis)

Zooming

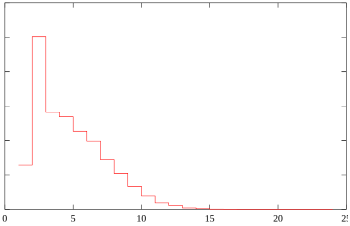


Autocorrelation

Danger time correlation among samples
experiments impact on experiments
$\Rightarrow$ stationarity analysis
autocorrelation estimation (ARMA)

Distribution analysis

Summarize data in a **histogram**

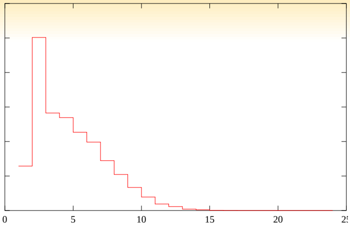


Shape analysis

- unimodal / multimodal
- variability
- symmetric / dissymmetric (skewness)
- flatness (kurtosis)

$\Longrightarrow$ **Central tendency analysis**
$\Longrightarrow$ **Variability analysis around the central tendency**

Mode value



Mode

- Categorical data
- Most frequent value
- highly unstable value
- for continuous value distribution depends on the histogram step
- interpretation depends on the flatness of the histogram

⟹ **Use it carefully**
⟹ **Predictor function**

Median value

Median

- Ordered data
- Split the sample in two equal parts

$$\sum_{i \leqslant Median} f_i \leqslant \frac{1}{2} \leqslant \sum_{i \leqslant Median+1} f_i.$$

- more stable value
- does not depends on the histogram step
- difficult to combine (two samples)

$\Longrightarrow$ Randomized algorithms

Mean value

Mean

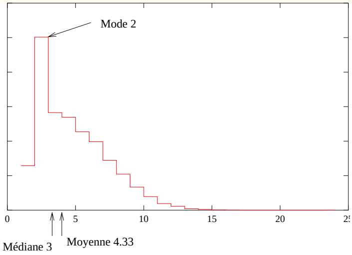
- Vector space
- Average of values

$$Mean = \frac{1}{Sample\_Size} \sum x_i = \sum_x x.f_x.$$

- stable value
- does not depends on the histogram step
- easy to combine (two samples ⇒ weighted mean)

⟹ **Additive problems (cost, durations, length,...)**

Central tendency



Complementarity

- Valid if the sample is "Well-formed"
- Semantic of the observation
- Goal of analysis

$\Longrightarrow$ **Additive problems (cost, durations, length,...)**

Central tendency (2)

Summary of Means

- Avoid means if possible
  Loses information
- Arithmetic mean
  When sum of raw values has physical meaning
  Use for summarizing times (not rates)
- Harmonic mean
  Use for summarizing rates (not times)
- Geometric mean
  Not useful when time is best measure of perf
  Useful when multiplicative effects are in play

Computational aspects

- Mode : computation of the histogram steps, then computation of max $O(n)$ "off-line"
- Median : sort the sample $O(nlog(n))$ or $O(n)$ (subtile algorithm) "off-line"
- Mean : sum values $O(n)$ "on-line" computation

Is the central tendency significant ?
$\Rightarrow$ Explain variability.

Computational aspects

- Mode : computation of the histogram steps, then computation of max $O(n)$ "off-line"
- Median : sort the sample $O(nlog(n))$ or $O(n)$ (subtile algorithm) "off-line"
- Mean : sum values $O(n)$ "on-line" computation

Is the central tendency significant ?
$\Rightarrow$ Explain variability.

Variability

Categorical data (finite set)

f_i : empirical frequency of element i
Empirical entropy

$$H(f) = \sum_i f_i \log f_i.$$

Measure the empirical distance with the uniform distribution

- $H(f) \geqslant 0$
- $H(f) = 0$ iff the observations are reduced to a unique value
- $H(f)$ is maximal for the uniform distribution

Variability (2)

Ordered data

Quantiles : quartiles, deciles, etc
Sort the sample :

$$(x_1, x_2, \cdots, x_n) \longrightarrow (x_{(1)}, x_{(2)}, \cdots, x_{(n)});$$

$$Q_1 = x_{(n/4)};\ Q_2 = x_{(n/2)} = Median;\ Q_3 = x_{(3n/4)}.$$

For deciles

$$d_i = argmax_i\{\sum_{j \leqslant i} f_j \leqslant \frac{i}{10}\}.$$

Utilization as quantile/quantile plots to compare distributions

Variability (3)

Vectorial data

Quadratic error for the mean

$$Var(X) = \frac{1}{n}\sum_{1}^{n}(x_i - \bar{x}_n)^2.$$

Properties:

$$\begin{aligned} Var(X) &\geqslant 0; \\ Var(X) &= \overline{x^2} - (\bar{x})^2, \text{ où } \overline{x^2} = \frac{1}{n}\sum_{i=1}^{n} x_i^2. \\ Var(X + cste) &= Var(X); \\ Var(\lambda X) &= \lambda^2 Var(X). \end{aligned}$$

Bibliography

- **The Art of Computer Systems Performance Analysis : Techniques for Experimental Design, Measurment, Simulation and Modeling.** Raj Jain *Wiley 1991*
- **Measuring Computer Performance: A Practitioner's Guide** David J. Lilja Cambridge University Press, 2000.

L I G