Dependable Distributed Systems Master of Science in Engineering in Computer Science

AA 2022/2023

LECTURE 21 : INTRO TO EXPERIMENTAL DESIGN

Part of the material inside is taken from slides of prof Raj Jain

Note on output analysis

Each simulation run is just a particular realization of the random variables

If not properly analysed, results of the executed runs may hide an high variance and lead to wrong inference about the system under analysis

Independence Across Runs Property

Let $Y_1, Y_2, \ldots$ be an output stochastic process from a single simulation run

- e.g., Y_i might be the throughput (production) in the i-th hour for a manufacturing system

Let $y_{1,1}, y_{1,2}, \ldots, y_{1,m}$ be a realization of the random variables $Y_1, Y_2, \ldots, Y_m$ resulting from making the simulation run 1 of length *m* by using the random numbers $u_{1,1}, u_{1,2}, \ldots$

OBSERVATION

If we run a second simulation (run 2) with a different set of random numbers $u_{2,1}, u_{2.2}, \ldots$, then we will obtain a different realization $y_{2,1}, y_{2,2}, \ldots, y_{2,m}$ of the random variables $Y_1, Y_2, \ldots, Y_m$

Independence Across Runs Property

Let $Y_1, Y_2, \dots$ be an output stochastic process from a single simulation run

- e.g., Y_i might be the throughput (production) in the i-th hour for a manufacturing system

Let $y_{1,1}, y_{1,2}, \dots, y_{1,m}$ be a realization of the random variables $Y_1, Y_2, \dots, Y_m$ resulting from making the simulation run 1 of length *m* by using the random numbers $u_{1,1}, u_{1,2}, \dots$

Performing *n* independent runs we get

RUN 1	$y_{1,1}$	$\cdots$	$y_{1,i}$	$\cdots$	$y_{1,m}$
RUN2	$y_{2,1}$	$\cdots$	$y_{2,i}$	$\cdots$	$y_{2,m}$
	$\vdots$		$\vdots$		$\vdots$
RUN n	$y_{n,1}$	$\cdots$	$y_{n,i}$	$\cdots$	$y_{n,m}$

The observations from a particular replication (row) are clearly not IID

The observation of a particular realization *i* across multiple runs is IID

Experimental Design and Analysis

- Design a proper set of experiments for measurement or simulation
- Develop a model that best describes the data obtained
- **Estimate the contribution of each alternative to the performance**
- Isolate the measurement errors
- Estimate confidence intervals for model parameters
- **Check if the alternatives are significantly different**
- Check if the model is adequate

An old example

Personal workstation design

1. Processor: 68000, Z80, or 8086.

2. Memory size: 512K, 2M, or 8M bytes

3. Number of Disks: One, two, three, or four

Terminology

Response Variable: *Outcome*. E.g., throughput, response time

Factors: *Variables that affect the response variable*. E.g., CPU type, memory size, number of disk drives.

Levels: *The values that a factor can assume*, E.g., the CPU type has three levels: 68000, 8080, or Z80

Replication: *Repetition of all or some experiments*

Design: *The number of experiments, the factor level and number of replications for each experiment*

Experimental Unit: *Any entity that is used for experiments*

Terminology

Interaction: Two factors A and B are said to interact if the effect of one depends upon the level of the other

Table 1: Noninteracting Factors

	A_1	A_2
B_1	3	5
B_2	6	8

Table 2: Interacting Factors

	A_1	A_2
B_1	3	5
B_2	6	9

Types of Experimental Designs

Given k factors, with the i-th factor having n_i levels

- **Simple Designs**: Vary one factor at a time

$$\#\text{ of Experiments} = 1 + \sum_{i=1}^{k} (n_i - 1)$$

  - Not statistically efficient.
  - Wrong conclusions if the factors have interaction.
  - Not recommended.

The response of a base configuration is measured first

- **Full Factorial Design**: All combinations.

$$\#\text{ of Experiments} = \prod_{i=1}^{k} n_i$$

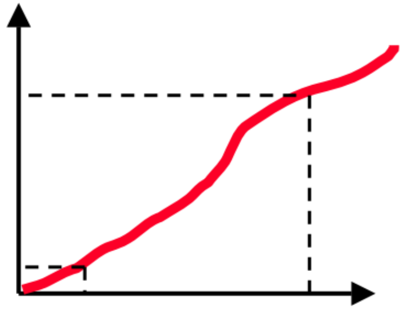
  - Can find the effect of all factors.
  - Too much time and money.
  - May try 2^k design first.

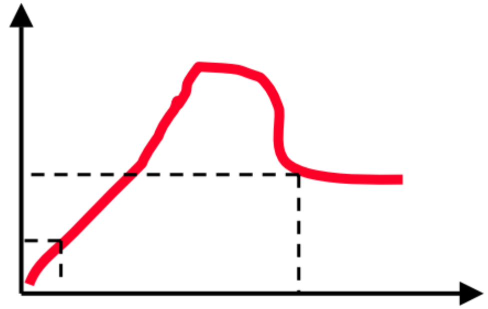
Types of Experimental Designs

- Fractional Factorial Designs: Less than Full Factorial
  - Save time and expense.
  - Less information.
  - May not get all interactions.
  - Not a problem if negligible interactions

2^k Factorial Designs

- k factors, each at two levels.
- Easy to analyze.
- Helps in sorting out impact of factors.
- Good at the beginning of a study.
- Valid only if the effect is unidirectional.
  E.g., memory size, the number of disk drives





2^2 Factorial Factorial Designs

- Two factors, each at two levels.

Performance in MIPS

Cache Size	Memory Size 4M Bytes	Memory Size 16M Bytes
1K	15	45
2K	25	75

$$x_A = \begin{cases} -1 & \text{if 4M bytes memory} \\ 1 & \text{if 16M bytes memory} \end{cases}$$

$$x_B = \begin{cases} -1 & \text{if 1K bytes cache} \\ 1 & \text{if 2K bytes cache} \end{cases}$$

2^2 Factorial Factorial Designs, Computation of effects

Experiment	A	B	y
1	-1	-1	y_1
2	1	-1	y_2
3	-1	1	y_3
4	1	1	y_4

$$y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B$$

y = response
qi = effects

$$y_1 = q_0 - q_A - q_B + q_{AB}$$

$$y_2 = q_0 + q_A - q_B - q_{AB}$$

$$y_3 = q_0 - q_A + q_B - q_{AB}$$

$$y_4 = q_0 + q_A + q_B + q_{AB}$$

2^2 Factorial Factorial Designs, Computation of effects

$$q_0 = \frac{1}{4}(y_1 + y_2 + y_3 + y_4)$$
$$q_A = \frac{1}{4}(-y_1 + y_2 - y_3 + y_4)$$
$$q_B = \frac{1}{4}(-y_1 - y_2 + y_3 + y_4)$$
$$q_{AB} = \frac{1}{4}(y_1 - y_2 - y_3 + y_4)$$

2^2 Factorial Factorial Designs, Computation of effects

Experiment	A	B	y
1	-1	-1	y_1
2	1	-1	y_2
3	-1	1	y_3
4	1	1	y_4

$$q_A = \frac{1}{4}(-y_1 + y_2 - y_3 + y_4)$$

$$q_B = \frac{1}{4}(-y_1 - y_2 + y_3 + y_4)$$

Notice:

q_A = Column A × Column y

q_B = Column B × Column y

Allocation of variation

The importance of a factor is measured by the proportion of the total variation in the response that is explained by the factor

$$\text{Total Variation of } y = \text{SST} = \sum_{i=1}^{2^2} (y_i - \bar{y})^2$$

the mean of responses from all four experiments

- For a 2^2 design:

$$\text{SST} = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2 = \text{SSA} + \text{SSB} + \text{SSAB}$$

- Variation due to A = SSA = $2^2\, q_A^{\,2}$
- Variation due to B = SSB = $2^2\, q_B^{\,2}$
- Variation due to interaction = SSAB = $2^2\, q_{AB}^{\,2}$
- Fraction explained by A = $\frac{SSA}{SST}$

Variation ≠ Variance

Allocation of variation example

- Memory-cache study:

$$\bar{y} = \frac{1}{4}(15 + 55 + 25 + 75) = 40$$

$$\begin{aligned} \text{Total Variation} &= \sum_{i=1}^{4} (y_i - \bar{y})^2 \\ &= (25^2 + 15^2 + 15^2 + 35^2) \\ &= 2100 \\ &= 4 \times 20^2 + 4 \times 10^2 + 4 \times 5^2 \end{aligned}$$

- Total variation= 2100

  Variation due to Memory = 1600 (76%)

  Variation due to cache = 400 (19%)

  Variation due to interaction = 100 (5%)

2^k Design Example

- Three factors in designing a machine:
  - Cache size
  - Memory size
  - Number of processors

	Factor	Level -1	Level 1
A	Memory Size	4MB	16MB
B	Cache Size	1kB	2kB
C	Number of Processors	1	2

2^k Design Example

Cache Size	4M Bytes		16M Bytes	
	1 Proc	2 Proc	1 Proc	2 Proc
1K Byte	14	46	22	58
2K Byte	10	50	34	86

I	A	B	C	AB	AC	BC	ABC	y
1	-1	-1	-1	1	1	1	-1	14
1	1	-1	-1	-1	-1	1	1	22
1	-1	1	-1	-1	1	-1	1	10
1	1	1	-1	1	-1	-1	-1	34
1	-1	-1	1	1	-1	-1	1	46
1	1	-1	1	-1	1	-1	-1	58
1	-1	1	1	-1	-1	1	-1	50
1	1	1	1	1	1	1	1	86
320	80	40	160	40	16	24	9	Total
40	10	5	20	5	2	3	1	Total/8

2^k Design Example

$$
\begin{aligned}
\text{SST} &= 2^3(q_A^2 + q_B^2 + q_C^2 + q_{AB}^2 + q_{AC}^2 + q_{BC}^2 + q_{ABC}^2) \\
&= 8(10^2 + 5^2 + 20^2 + 5^2 + 2^2 + 3^2 + 1^2) \\
&= 800 + 200 + 3200 + 200 + 32 + 72 + 8 = 4512 \\
&= 18\% + 4\% + 71\% + 4\% + 1\% + 2\% + 0\% \\
&= 100\%
\end{aligned}
$$

- Number of Processors (C) is the most important factor.

$2^k r$ Factorial Designs

- *r* replications of 2^k Experiments
  $\Rightarrow$ $2^k r$ observations.
  $\Rightarrow$ Allows estimation of experimental errors.

- Model:

$$y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B + e$$

- e = Experimental error

References

- R. Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling,“ https://www.cin.ufpe.br/~rmfl/ADS_MaterialDidatico/PDFs/performanceAnalysis/Art%20of%20Computer%20Systems%20Performance%20Analysis%20Techniques.pdf

- A. M. Law - Simulation modeling and analysis https://industri.fatek.unpatti.ac.id/wp-content/uploads/2019/03/108-Simulation-Modeling-and-Analysis-Averill-M.-Law-Edisi-5-2014.pdf