60017 PERFORMANCE ENGINEERING

Design of Experiments

Last lecture

- Service demand
- Utilization law
- Bottleneck analysis

This lecture

- Screening for influential factors
  - Full factorial designs
- Accelerating estimation of system response
  - Fractional factorial designs

Motivating example: system performance tuning

- Suppose that we want to optimize a VM for a workload.
- We may run tests varying many configuration options:
  - Number of virtual cores: $2, 4, 8, 16, 32$
  - Virtual core clock (MHz): $2534, 2534, 2133, 1867, 1600$
  - VM RAM (GB): $2, 4, 8, 16, 32$
  - Virtual disk size: Small, Medium, Large
  - I/O cache size: Small, Large
  - Hyper-threading (HT) on the host machine: ON, OFF
- Unfortunately, testing each combination would require $2 \times 3 \times 5 \times 2 \times 5 \times 5 = 1500$ experiments...
- How to quickly find the most promising configurations?

Terminology

Before continuing we need some terminology:

- Response variable: a measurement representing the outcome of a test *(e.g., response time, throughput, ...)*
- Factor: a configuration option that affects the response variable *(e.g., HT)*
- Level: a feasible value of a factor *(e.g., ON/OFF)*
- Design: an experimental plan specifying
  - The number of experiments that we will run
  - For each experiment, the level to be assigned to each factor

Terminology

- Interaction: two factors interact if their levels jointly affect the response variable.
- Consider an experiment where the response variable is the system response time (*ms*), and let's vary just two factors.
- If factors interact, results depend on the levels of both factors.

Non-interacting Factors

Cache size/HT	*Off*	*On*
Small	10	8
Large	5 x0.5	4 x0.5

Interacting Factors

Cache size/HT	*Off*	*On*
Small	5	8
Large	5 x1	4 x0.5

Blackbox modelling

How to find the most promising configurations quickly?

- We can build a model to guide us in the decision.
- For example, we may only want to test the configurations where the model predicts the best performance.
- How to define a robust model? We could try simulation or analytical modelling, but they become error-prone as the system's internal complexity grows.
- Since the system exists, we can instead use measurements to build blackbox models, which are agnostic of the internals.
- We will focus on models based on multivariate polynomials.

Screening response models

- Consider a simple example with two factors A and B
- The simplest polynomial that can also capture interactions is

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

  - y: response variable.
  - x_A, x_B: levels of factors A and B, suitably encoded.
  - $x_A x_B$: interaction of A and B.
  - q_0, q_A, q_B, q_{AB} : effects, coefficients explaining the influence of the factors on the response y and the interactions among them.
  - ϵ : term capturing experimental noise
- We call the above a screening response model.
- With k factors we use terms of order up to k, e.g. for $k = 3$:

$$y = q_0 + \sum_j q_j x_j + \sum_{\substack{j,k \\ j \neq k}} q_{jk} x_j x_k + \sum_{\substack{j,k,h \\ j \neq k \neq h}} q_{jkh} x_j x_k x_h + \epsilon$$

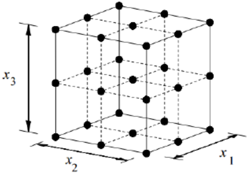
- We focus on the case without noise ($\epsilon = 0$).

Design methods

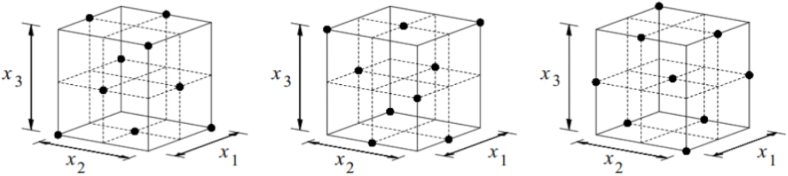
- Which experiments should we run to supply the model with informative data?
- Two strategies are in common use to design experiments.
- Full factorial design
  - Run every possible combination of levels
  - The study delivers maximal information to fit the model
  - Useful to identify the most important factors (screening)
  - Limited applicability outside screening, too expensive!
- Fractional factorial design
  - A fraction of the experiments of a full factorial design
  - Only some interactions may therefore be captured
  - Optimal results if some interactions are negligible
  - Applicable and popular in many areas of engineering

Example: full factorial vs fractional factorial designs

- Full factorial design (3 factors, 3 levels each)



- Fractional factorial design (here 1/3 of the experiments)



Source: L. F. Alvarez, PhD thesis, Univ. Bradford, 2000.

2^k factorial designs

- In 2^k designs, there are k factors, each having only 2 levels.
- Hence there are 2^k possible combinations of levels
  - Important special case, results are easy to analyse
  - Also called a screening design, as it can help choosing the most important factors among several of them ($8 - 12$ max)
- For factors with more than two levels, pick min and max
  - number of cores: 2,4,8,16,32 $\rightarrow$ pick 2,32
- Let us introduce some additional notation:
  - y_i: response variable in the i-th experiment ($1 \leq i \leq 2^k$)
  - $x_{A,i}$: level of factor A in the i-the experiment
  - $x_{B,i}$: level of factor B in the i-the experiment

2^2 factorial designs ($k = 2$)

- Full factorial design with 2 factors having 2 levels each
- We encode the levels as -1 (low) or $+1$ (high)
  - $+1$ and -1 simplify calculations, we could use other encodings but formulas would be more complex.
- Running case:
  - response = measured response time of the benchmark
  - factor A = I/O cache size, Large = 1, Small = -1
  - factor B = HT, ON = 1, OFF = -1

A/B	*B=-1*	*B=+1*
A=-1	15	25
A=+1	45	75

Running case

We fit the model setting $x_A = -1$ if $A = -1$, $x_B = 1$ if $B = 1$, ...

$$y_1 = 15 = q_0 - q_A - q_B + q_{AB}$$
$$y_2 = 45 = q_0 + q_A - q_B - q_{AB}$$
$$y_3 = 25 = q_0 - q_A + q_B - q_{AB}$$
$$y_4 = 75 = q_0 + q_A + q_B + q_{AB}$$

This is a system of linear equations, solving for the q_i's we get

$$q_0 = (1/4)(y_1 + y_2 + y_3 + y_4) = 40$$
$$q_A = (1/4)(-y_1 + y_2 - y_3 + y_4) = 20$$
$$q_B = (1/4)(-y_1 - y_2 + y_3 + y_4) = 10$$
$$q_{AB} = (1/4)(y_1 - y_2 - y_3 + y_4) = 5$$

Sign table method

- The sign table does the same without explicit equations
  - I column entries are always set to 1
  - The other columns represent an assignment of x_A, x_B, $x_A x_B$
  - A and B columns enumerate all combinations of levels
  - AB obtained by multiplying A and B columns
- Multiply the responses by column i and scale by 2^k to get q_i

I	*A*	*B*	*AB*	*y*
1	-1	-1	1	15
1	1	-1	-1	45
1	-1	1	-1	25
1	1	1	1	75
40	20	10	5	*y*col/4*
$= q_0$	$= q_A$	$= q_B$	$= q_{AB}$	

Example: null effects

- $q_0 = 10.0, q_A = 0.0, q_B = 0.0, q_{AB} = 0.0$

Cache/HT (A/B)	*Off (-1)*	*On (+1)*
Small (-1)	10	10
Large (1)	10	10

- $q_0 = 7.5, q_A = 0.0, q_B = 0.0, q_{AB} = 2.5$

Cache/HT (A/B)	*Off (-1)*	*On (+1)*
Small (-1)	10	5
Large (1)	5	10

Example: sign of effects

- $q_0 = 7.5, q_A = -2.5, q_B = 0.0, q_{AB} = 0.0$

Cache/HT (A/B)	Off (-1)	On (+1)
Small (-1)	10	10
Large (1)	5	5

- $q_0 = 7.5, q_A = 2.5, q_B = 0.0, q_{AB} = 0.0$

Cache/HT (A/B)	Off (-1)	On (+1)
Small (-1)	5	5
Large (1)	10	10

Allocation of Variation

- We now study the influence of the factors on the response
- The method we see explains the variance of the y_i values
- A factor is more important if it contributes more to the SST
  - We consider a scaled variance, called variation (the SST):

$$
\begin{aligned}
SST &= \text{Sum of Squares Total} \\
&= \sum_{1 \le i \le 2^k} (y_i - \bar{y})^2 \\
&= \sum_{1 \le i \le 2^k} (q_A x_{A,i} + q_B x_{B,i} + q_{AB} x_{A,i} x_{B,i})^2 \\
&= \sum_{1 \le i \le 2^k} (q_A x_{A,i})^2 + \sum_{1 \le i \le 2^k} (q_B x_{B,i})^2 + \sum_{1 \le i \le 2^k} (q_{AB} x_{A,i} x_{B,i})^2 \\
&\quad + \textit{product terms}
\end{aligned}
$$

where $\bar{y}$ is the average response across experiments, thus $\bar{y} = q_0$.

Allocation of Variation

- Using that $x_{A,i}^2 = x_{B,i}^2 = 1$, the product terms simplify to

$$2q_A q_{AB} \sum_{1 \le i \le 2^k} x_{B,i} + 2q_B q_{AB} \sum_{1 \le i \le 2^k} x_{A,i} + 2q_A q_B \sum_{1 \le i \le 2^k} x_{A,i} x_{B,i}$$

- Note that the A and B columns of the sign table sum to zero

$$\sum_{1 \le i \le 2^k} x_{A,i} = \sum_{1 \le i \le 2^k} x_{B,i} = 0$$

- Also, due to the orthogonality of A and B in the sign table

$$\sum_{1 \le i \le 2^k} x_{A,i} x_{B,i} = 0$$

- Therefore the product terms sum to zero.

Allocation of Variation

- Putting everything together

$$SST = q_A^2 \sum_{1 \le i \le 2^k} x_{A,i}^2 + q_B^2 \sum_{1 \le i \le 2^k} x_{B,i}^2 + q_{AB}^2 \sum_{1 \le i \le 2^k} (x_{A,i} x_{B,i})^2$$

- Due to the squaring, sums are all $\sum_{1 \le i \le 2^k} 1 = 2^k$, hence since $k = 2$

$$\boxed{SST = 4(q_A^2 + q_B^2 + q_{AB}^2)}$$

Allocation of Variation

$$\boxed{SST = 4(q_A^2 + q_B^2 + q_{AB}^2)}$$

- The most appealing property of 2^k designs is that the contributions of the factors to the SST are easy to interpret.
  - $SSA =$ variation explained by $A = 4q_A^2$
  - $SSB =$ variation explained by $B = 4q_B^2$
  - $SSAB =$ variation explained by $AB = 4q_{AB}^2$
  - $SST = SSA + SSB + SSAB$
- The ratios SSA/SST, SSB/SST, $SSAB/SST$ show the percentage of the variation explained by each factor and by their interaction.
- Factors or interactions that explain a higher percentage of variation are considered more important.

Example: allocating variation in the running case

- $q0 = 40, q_A = 20.0, q_B = 10.0, q_{AB} = 5.0$
- $SST = 4(q_A^2 + q_B^2 + q_{AB}^2) = 2100$
- $SSA/SST = 4q_A^2/SST = 76.19\%$
- $SSB/SST = 4q_B^2/SST = 19.05\%$
- $SSAB/SST = 4q_{AB}^2/SST = 4.76\%$
- The variation is mostly explained by the levels of A and (to a moderate extent) of B, not by their interaction.

Case study: modelling hyper-threading effects

- We run a TPC-W benchmark on a quad-core Intel Xeon 5540
- 2 factors: HT and CPU frequency
- 2 possible response variables: power and response time

CPU Freq. (MHz)	Hyper-threading ON/OFF	Power Consumption (W)	Mean Response Time [ms]
2534	ON	**184**	13.50
2534	OFF	193	**11.60**
2133	ON	**170**	16.10
2133	OFF	176	**14.50**
1867	ON	**167**	**17.50**
1867	OFF	174	18.70
1600	ON	**167**	**21.20**
1600	OFF	173	43.10

Case study: modelling hyper-threading effects

- Response variable = system response time

Freq/HT (F/H)	*Off (-1)*	*On (+1)*
1600 (-1)	43.10	21.20
2534 (1)	11.60	13.50

- $q_0 = 22.35$, $q_F = -9.80$, $q_H = -5.00$, $q_{FH} = 5.95$
- Explained variation:
  - SSF/SST = 61.4%
  - SSH/SST = 16.0%
  - SSFH/SST = 22.6%

⇒ Response time tuning requires managing both frequency & HT

Case study: modelling hyper-threading effects

- Response variable = power consumption

Freq/HT	*Off (-1)*	*On (+1)*
1600 (-1)	173	167
2534 (1)	193	184

- $q_0 = 179.25$, $q_F = -9.25$, $q_H = 3.75$, $q_{FH} = -0.75$
- Explained variation:
  - SSF/SST = 85.4%
  - SSH/SST = 14.0%
  - SSFH/SST= 0.6%

$\Rightarrow$ Frequency explains power usage, HT has a limited influence

General 2^k designs

- The results can be easily generalised, however higher-order interaction need to be included
- For example with $k = 3$

$$
\begin{aligned}
SST &= 8(q_A^2 + q_B^2 + q_C^2 + q_{AB}^2 + q_{BC}^2 + q_{AC}^2 + q_{ABC}^2) \\
&= (SSA + SSB + SSC) + (SSAB + SSBC + SSAC) \\
&\quad + SSABC
\end{aligned}
$$

- Factors with low explained variation can be removed from follow-up experiments, which are typically carried out using the fractional factorial method.

Example: sign table for $k = 3$

- Easily automated, e.g., MATLAB's *fullfact* function

I	*A*	*B*	*C*	*AB*	*AC*	*BC*	*ABC*
1	-1	-1	-1	1	1	1	-1
1	1	-1	-1	-1	-1	1	1
1	-1	1	-1	-1	1	-1	1
1	1	1	-1	1	-1	-1	-1
1	-1	-1	1	1	-1	-1	1
1	1	-1	1	-1	1	-1	-1
1	-1	1	1	-1	-1	1	-1
1	1	1	1	1	1	1	1

Fractional factorial designs: 2^{k-p} designs

- 2^k screening designs often require too many experiments
  - A recent study considered how GCC compiler options affect sofware energy usage (Pallister *et al.*, 2013).
  - The authors observed they would need at least a 2^{82} design.
- After using a 2^k screening design we know which factors strongly interact with each others.
- We can then decompose the problem in smaller subproblems, with just a few interacting factors.
- Another simplification consists in adopting a 2^{k-p} design
  - Analyze k two-level factors using only 2^{k-p} experiments
  - p is a user-specified parameter, which controls precision
- Compared to a 2^k design this can save a considerable effort
  - 2^{k-1} design needs half of the experiments of a 2^k design
  - 2^7 design needs 128 experiments, a 2^{7-4} design only $2^3 = 8$

Key Idea of a 2^{k-p} factorial design

- Goal: keep simplicity of result interpretation of a 2^k design
- Assumption: interactions among the factors are sparse and we can guess which ones they are.
- Under this assumption, we can sacrifice some interaction terms of the response model to allow for more factors.
- 2^2 response model

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

- 2^{3-1} response model

$$y = q_0 + q_A x_A + q_B x_B + q_C x_C$$

- In larger designs, some interactions will still appear in the response model.
- Problem: ignoring interactions can bias the q_i values!

Sign table for a 2^k design

2^k columns

k columns | $2^k - k - 1$ columns

Example: 2^3 design

I	*A*	*B*	*C*	*AB*	*AC*	*BC*	*ABC*
1	-1	-1	-1	1	1	1	-1
1	1	-1	-1	-1	-1	1	1
1	-1	1	-1	-1	1	-1	1
1	1	1	-1	1	-1	-1	-1
1	-1	-1	1	1	-1	-1	1
1	1	-1	1	-1	1	-1	-1
1	-1	1	1	-1	-1	1	-1
1	1	1	1	1	1	1	1

Sign table for a 2^{k-p} design

We sacrifice the $q_{ABC}x_Ax_Bx_C$ term to make room for factor D.

2^{k-p} columns

2^{4-1} design

$p = 1$

	$k - p$			$2^{k-p} - k - 1$			p
I	*A*	*B*	*C*	*AB*	*AC*	*BC*	*D*
1	-1	-1	-1	1	1	1	-1
1	1	-1	-1	-1	-1	1	1
1	-1	1	-1	-1	1	-1	1
1	1	1	-1	1	-1	-1	-1
1	-1	-1	1	1	-1	-1	1
1	1	-1	1	-1	1	-1	-1
1	-1	1	1	-1	-1	1	-1
1	1	1	1	1	1	1	1

D will be assigned to these levels in the experiments

Confounding

D is thus dependent on A, B, C, but in this way we run less experiments. The problem of confounding is the price to pay!

Using columns products we see that interactions overlap with main effects biasing the effects

2^{4-1} design

$p = 1$

I	*A*	*B*	*C*	*AB*	*AC*	*BC*	*D*
ABCD	*BCD*	*ACD*	*ABD*	*CD*	*BD*	*AD*	*ABC*
1	-1	-1	-1	1	1	1	-1
1	1	-1	-1	-1	-1	1	1
1	-1	1	-1	-1	1	-1	1
1	1	1	-1	1	-1	-1	-1
1	-1	-1	1	1	-1	-1	1
1	1	-1	1	-1	1	-1	-1
1	-1	1	1	-1	-1	1	-1
1	1	1	1	1	1	1	1

Resolution of a Fractional Design

- Order of an effect = number of factors included in it
  - e.g., ABCD is a 4th-order effect
- The confoudings in the 2^{4-1} design are
  - $A = BCD$, $B = ACD$, $C = ABD$ (1st order and 3rd order)
  - $AB = CD$, $AC = BD$, $BC = AD$ (2nd order and 2nd order)
  - $I = ABCD$ (0th order and 4th order)
- The full list of confoundings is algorithmically generated starting from the known 0th or 1st order confoundings, e.g.,

$$D = ABC \Rightarrow D * D = D * ABC \Rightarrow I = ABCD$$

  since the squaring of the entries of column D is I.
- If the sum of the orders of the confoundings is r or more, we say that the design has resolution r
  - Resolution numbers are indicated with Roman literals
  - The 2^{4-1} example is a resolution IV design
  - The higher the resolution the less severe the confounding is.

Resolution of a Fractional Design

Resolution III design

I	A	B	C	D	AC	BC	ABC
ABD	BD	AD	$ABCD$	AB	BCD	ACD	CD
1	-1	-1	-1	1	1	1	-1
1	1	-1	-1	-1	-1	1	1
1	-1	1	-1	-1	1	-1	1
1	1	1	-1	1	-1	-1	-1
1	-1	-1	1	1	-1	-1	1
1	1	-1	1	-1	1	-1	-1
1	-1	1	1	-1	-1	1	-1
1	1	1	1	1	1	1	1

Caution!
Main effects are confounded with 2^{nd} order interactions

Resolution of a Fractional Design

Resolution IV design

I	A	B	C	AB	AC	BC	D
ABCD	BCD	ACD	ABD	CD	BD	AD	ABC
1	-1	-1	-1	1	1	1	-1
1	1	-1	-1	-1	-1	1	1
1	-1	1	-1	-1	1	-1	1
1	1	1	-1	1	-1	-1	-1
1	-1	-1	1	1	-1	-1	1
1	1	-1	1	-1	1	-1	-1
1	-1	1	1	-1	-1	1	-1
1	1	1	1	1	1	1	1

Confounding of main terms is with 3rd order interactions. This is better.