Factorial design and optimization methods

Krzysztof Podgórski
Department of Mathematics and Statistics
University of Limerick

November 23, 2009

Factorial design

- n factors

Factorial design

- n factors
- each factor is on two levels: ‘low’ (-) and ‘high’ (+)

Factorial design

- n factors
- each factor is on two levels: ‘low’ (-) and ‘high’ (+)
- there is 2^n combinations of factors and their levels

Factorial design

- *n* factors
- each factor is on two levels: 'low' (-) and 'high' (+)
- there is 2^n combinations of factors and their levels
- complete factorial design requires 2^n data

Factorial design

- n factors
- each factor is on two levels: ‘low’ (-) and ‘high’ (+)
- there is 2^n combinations of factors and their levels
- complete factorial design requires 2^n data
- if $n = 3$, then at least 8 data points are needed.

3-way factorial design

The factors are symbolically denoted by A, B, and C.

Combination	A	B	C	Response
1	−	−	−	y1
a	+	−	−	y2
b	−	+	−	y3
c	−	−	+	y4
ab	+	+	−	y5
ac	+	−	+	y6
bc	−	+	+	y7
abc	+	+	+	y8

Effects of individual factors and interactions

- Effect – the average of observations when the factor is ‘high’ minus the average when it is ‘low’.

Effects of individual factors and interactions

- Effect – the average of observations when the factor is 'high' minus the average when it is 'low'.
- First order interactions – Consider A and B. Set B at - than take the difference between average when A is at + and average when A is at - . Repeat the same for the case when B is at + and take half the difference between the second one and the first one.
- Second order interactions – AB interactions can be split into two components, those obtained when C is at + and those obtained when C is at -. Half of the difference between the former and the latter is three way interaction.

Effects of individual factors and interactions

- Effect – the average of observations when the factor is ‘high’ minus the average when it is ‘low’.

Effects of individual factors and interactions

- Effect – the average of observations when the factor is ‘high’ minus the average when it is ‘low’.
- First order interactions – Consider A and B. Set B at - than take the difference between average when A is at + and average when A is at - . Repeat the same for the case when B is at + and take half the difference between the second one and the first one.
- Second order interactions – AB interactions can be split into two components, those obtained when C is at + and those obtained when C is at -. Half of the difference between the former and the latter is three way interaction.

Example 7.7.1

In a liquid chromatography experiment, the dependence of the retention parameter, k', on three factors was investigated. The factors are: pH (P), the concentration of a counter-ion (T), and the concentration of the organic solvent in the mobile phase (C).

Combinations of factor levels	k′
1	4.7
p	9.9
t	7.0
c	2.7
pt	15.0
pc	5.3
tc	3.2
ptc	6.0

Example 7.7.1 – Effects of individual factors

Effect of P: (9.9+15.0+5.3+6.0-(4.7+7.0+2.7+3.2))/4 = 4.65

Example 7.7.1 – Effects of individual factors

Effect of P: (9.9+15.0+5.3+6.0-(4.7+7.0+2.7+3.2))/4 = 4.65
Effect of C: (2.7+5.3+3.2+6.0-(4.7+9.9+7.0+15.0))/4 = -4.85

Example 7.7.1 – Effects of individual factors

Effect of P: (9.9+15.0+5.3+6.0-(4.7+7.0+2.7+3.2))/4 = 4.65
Effect of C: (2.7+5.3+3.2+6.0-(4.7+9.9+7.0+15.0))/4 = -4.85
Effect of T: (7.0+15.0+3.2+6.0-(4.7+9.9+2.7+5.3))/4 = 2.15

Example 7.7.1 – Interaction between two and three factors

Effect of PT: 0.75

Example 7.7.1 – Interaction between two and three factors

Effect of PT: 0.75
Effect of PC: -1.95

Example 7.7.1 – Interaction between two and three factors

Effect of PT: 0.75
Effect of PC: -1.95
Effect of TC: -1.55

Example 7.7.1 – Interaction between two and three factors

Effect of PT: 0.75
Effect of PC: -1.95
Effect of TC: -1.55
Effect of PTC: -0.65

Example 7.7.1 – Sums of squares

- Sum of squares = $N(\mathit{effect})^2/4$, where N is the total number of measurements.

Example 7.7.1 – Sums of squares

- Sum of squares = $N(\mathit{effect})^2/4$, where N is the total number of measurements.
- Each sum of squares for effects has one degree of freedom.

Example 7.7.1 – Sums of squares

- Sum of squares = $N(\mathit{effect})^2/4$, where N is the total number of measurements.
- Each sum of squares for effects has one degree of freedom.
- Residuals sum of squares has to be computed either by subtraction from the total sum of squares or directly as described before.

Example 7.7.1 – Sums of squares

- Sum of squares = $N(\mathit{effect})^2/4$, where N is the total number of measurements.
- Each sum of squares for effects has one degree of freedom.
- Residuals sum of squares has to be computed either by subtraction from the total sum of squares or directly as described before.
- If all interaction are considered and no replicates are made, there is not enough degrees of freedom to test for significance of main effects and interactions

Example 7.7.1 – Sums of squares

- Sum of squares = $N(\mathit{effect})^2/4$, where N is the total number of measurements.
- Each sum of squares for effects has one degree of freedom.
- Residuals sum of squares has to be computed either by subtraction from the total sum of squares or directly as described before.
- If all interaction are considered and no replicates are made, there is not enough degrees of freedom to test for significance of main effects and interactions
- If such tests are desired the highest order interaction should be dropped from considerations.

Example 7.7.1 in *R*

Mercury is lost from solutions stored in polypropylene flasks by combination with traces of tin in the polymer. The absorbance of a standard aqueous solution of mercury stored in such flasks was measured for two levels of the following factors:

Factor	Low	High
P - pH	-1	1
T - counter-ion concentration	-1	1
C - organic solvent concentr.	-1	1

The following results were obtained.

Combination of factor levels	k'
1	4.7
p	9.9
t	7.0
c	2.7
pt	15.0
pc	5.3
tc	3.2
ptc	6.0

Reading and formating data

We read the data into *R*:

```
k=c(4.7,9.9,7.0,2.7,15.0,5.3,3.2,6.0)
P=c(-1, 1,-1,-1, 1, 1,-1,1)
T=c(-1,-1, 1,-1, 1,-1, 1,1)
C=c(-1,-1,-1, 1,-1, 1, 1,1)
```

Analysis of the data

The following lines of codes allows to compute the effects which are twice the coefficients to the linear model fitted to the above data.

```
Result=lm(k~P+T+C+P*T+P*C+T*C+P*C*T)
summary(Result)
Call:
lm(formula = k ~ P + T + C + P * T + P * C + T * C + P * C * T)
Residuals:
ALL 8 residuals are 0: no residual degrees of freedom!
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.725         NA      NA       NA
P              2.325         NA      NA       NA
T              1.075         NA      NA       NA
C             -2.425         NA      NA       NA
P:T            0.375         NA      NA       NA
P:C           -0.975         NA      NA       NA
T:C           -0.775         NA      NA       NA
P:T:C         -0.325         NA      NA       NA
Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:     1, Adjusted R-squared:   NaN
F-statistic:   NaN on 7 and 0 DF,  p-value: NA
```

Effects=Twice estimated coefficients

```
2*coef(Result)
(Intercept)           A          C          T
     0.1545     -0.0215     0.0005    -0.0265
        A:C         A:T        C:T      A:C:T
    -0.0005     -0.0065     0.0025    -0.0005
```

Two way factorial design with replicates

Suppose that the following result has been obtained for replicates of the experiment

```
k1=c(4.5,9.8,6.8,2.9,14.8,5.3,3.2,5.6)
```

We read in the data and design

```
k1=c(4.5,9.8,6.8,2.9,14.8,5.3,3.2,5.6)
kk=c(k,k1)
P=c(P,P)
T=c(T,T)
C=c(C,C)
```

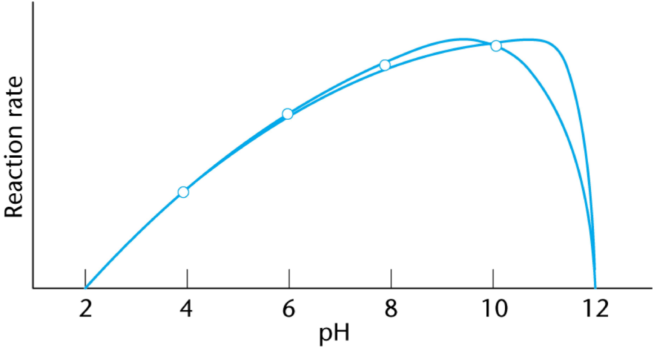
Optimization

Optimization

- The design and its analysis allows to test for effects of factors.

Optimization

- The design and its analysis allows to test for effects of factors.
- The next step is to find the level of factors that gives an optimal (maximal or minimal) response.



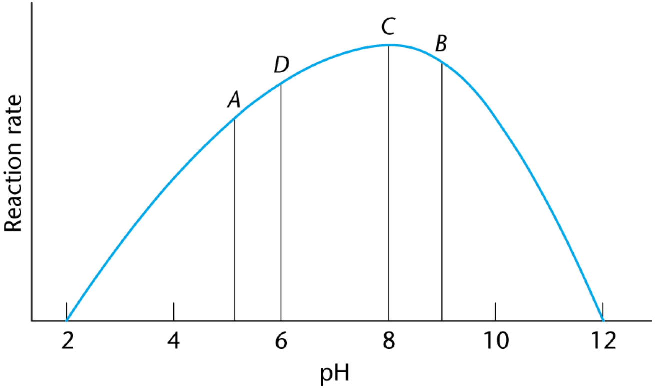
Equidistant vs. efficient search algorithms

Equidistant vs. efficient search algorithms

- The equidistant search may be very inefficient if we are out of luck.

Equidistant vs. efficient search algorithms

- The equidistant search may be very inefficient if we are out of luck.
- There exist more uniformly efficient forms of finding optimum



Fibonacci series and efficient search algorithms

Fibonacci series and efficient search algorithms

- $F_0 = F_1 = 1$, $F_{n-1} + F_n = F_{n+1}$.

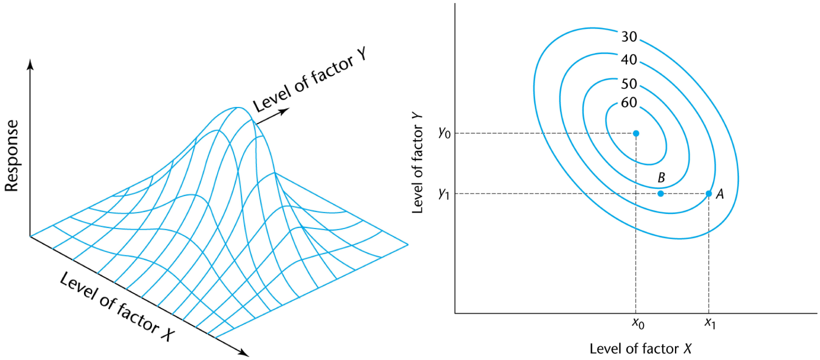
Fibonacci series and efficient search algorithms

- $F_0 = F_1 = 1$, $F_{n-1} + F_n = F_{n+1}$.
- If the precision of identifying the optimal location is supposed to be N-fold reduction of the initial range, i.e. $\epsilon = R/N$, where the R is the length of the original interval. Then for the minimal n such that $F_n \geq N$, we take F_{n-2}/F_n to be the proportion of the whole range for the distance of A from the left end point. The same proportion is taken for B and the right end point of the entire interval.

Fibonacci series and efficient search algorithms

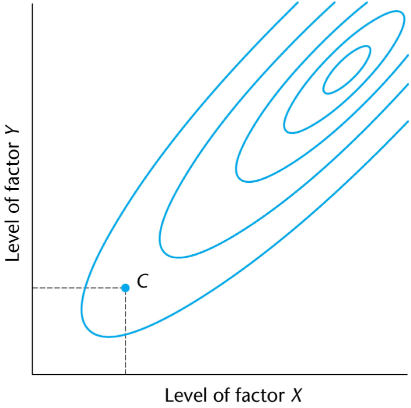
- $F_0 = F_1 = 1$, $F_{n-1} + F_n = F_{n+1}$.
- If the precision of identifying the optimal location is supposed to be N-fold reduction of the initial range, i.e. $\epsilon = R/N$, where the R is the length of the original interval. Then for the minimal n such that $F_n \geq N$, we take F_{n-2}/F_n to be the proportion of the whole range for the distance of A from the left end point. The same proportion is taken for B and the right end point of the entire interval.
- This grants that optimal value with the error ϵ will be found in n steps.

Alternating variable search method

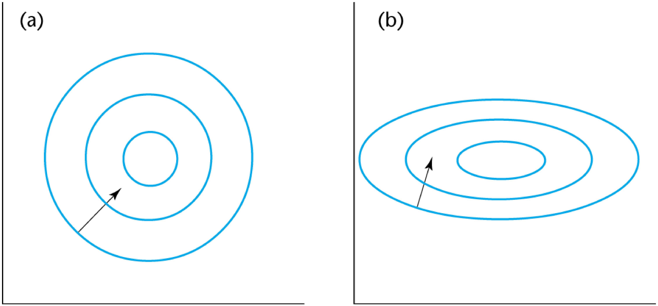


Alternating variable search method and interaction

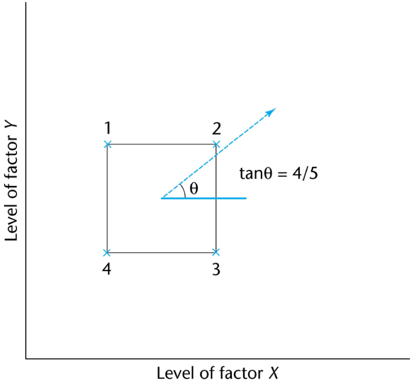
The methods fails if there are strong interactions:



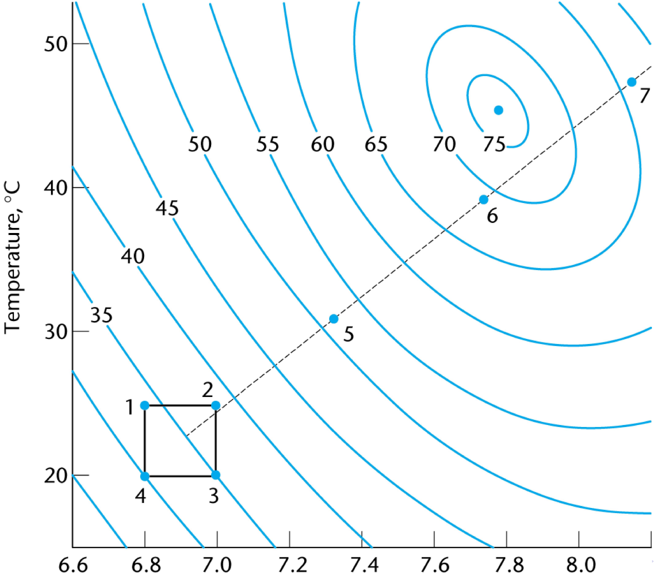
The method of steepest ascent



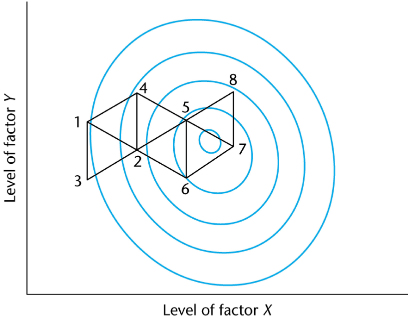
Factorial design and the steepest ascent direction

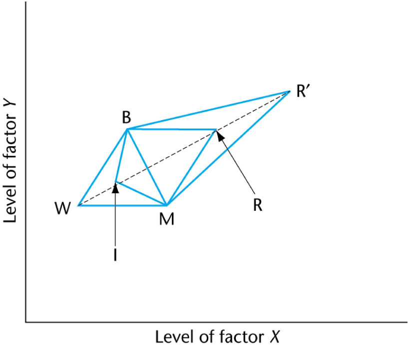


Close to the optimum – interactions



Simplex optimization





Local vs. Global maxima – Simulated annealing