Factorial Designs: Rank and Replicates

BIOE 498/598

2/17/2020

Rank revisited

The rank of a matrix quantifies the number of linearly independent rows or columns.

The column rank of a matrix is always equal to the row rank.

$$\mathrm{rank}(\mathbf{X}) = \mathrm{rank}(\mathbf{X}^{\mathrm{T}})$$

This limits the rank to be at most the smaller dimension of the matrix.

$$\mathrm{rank}(\mathbf{X}) \leq \min\{m, n\} \quad \text{if} \quad \dim(\mathbf{A}) = m \times n$$

If the above *equality* holds, we say that the matrix is **full rank**.

Rank and linear modeling

Each parameter in a linear model requires one independent piece of information.

The linear model $\mathbf{y} = \mathbf{X}\beta + \epsilon$ is solvable if and only if the design matrix $\mathbf{X}$ is full rank.

Solvability vs. power

A model is solvable if the design matrix is full rank. However, we need “extra" rows to estimate the uncertainty in the model.

Consider the one parameter model $y = \beta x + \epsilon$.

Given data $(x,y) = (3,6)$:

$$\hat{\beta} = x^{-1} y = 2$$

Solvability vs. power

A model is solvable if the design matrix is full rank. However, we need “extra" rows to estimate the uncertainty in the model.

Consider the one parameter model $y = \beta x + \epsilon$.

Given data $(x,y) = (3,6)$:

$$\hat{\beta} = x^{-1} y = 2$$

Substituting back we see that

$$\epsilon = y - \hat{\beta} x = 6 - 2 \times 3 = 0$$

With one data point our estimate is always exact!

Solvability vs. power

A model is solvable if the design matrix is full rank. However, we need “extra" rows to estimate the uncertainty in the model.

Consider the one parameter model $y = \beta x + \epsilon$.

Given data $(x,y) = (3,6)$:

$$\hat{\beta} = x^{-1} y = 2$$

Substituting back we see that

$$\epsilon = y - \hat{\beta} x = 6 - 2 \times 3 = 0$$

With one data point our estimate is always exact!

Now let's use two data points: $(x,y) = (3,6)$ and $(x,y) = (4,12)$.

$$\hat{\beta} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}^{+} \begin{pmatrix} 6 \\ 12 \end{pmatrix} = 2.64$$

$$\epsilon_1 = y_1 - \hat{\beta} x = 6 - 2.64 \times 3 = -1.92$$

$$\epsilon_2 = y_2 - \hat{\beta} x = 12 - 2.64 \times 4 = 1.44$$

What does this mean for factorial designs?

A full factorial design with n variables has 2^n experiments. It also has 2^n coefficients (intercept, first-order, and interaction). We can fit a model to a full factorial design but will have no information leftover to estimate the error.

We have three options if we want statistical power behind our factorial designs:

1. perform replicates of some (or all) runs
2. only estimate a subset of the 2^n coefficients
3. some combination of 1 & 2

What does this mean for factorial designs?

A full factorial design with n variables has 2^n experiments. It also has 2^n coefficients (intercept, first-order, and interaction). We can fit a model to a full factorial design but will have no information leftover to estimate the error.

We have three options if we want statistical power behind our factorial designs:

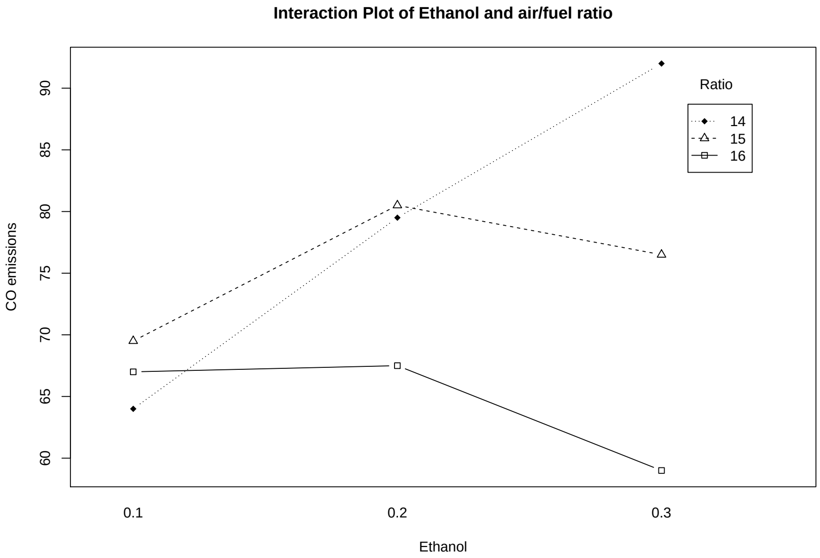
1. perform replicates of some (or all) runs
2. only estimate a subset of the 2^n coefficients
3. some combination of 1 & 2

- For small n designs we perform replicates since there are already few runs and the interactions are probably significant.
- For large n designs we drop coefficients for higher order terms since we already have lots of runs and the higher-order interactions are most likely zero.

CO emissions example

```
##
## Call:
## lm(formula = CO ~ Ratio * Eth, data = eth_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3333 -2.3333 -0.8333  1.0417  8.1667
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -101.167     45.637  -2.217  0.04370 *
## Ratio         11.000      3.038   3.621  0.00278 **
## Eth         1395.000    211.257   6.603 1.18e-05 ***
## Ratio:Eth    -90.000     14.063  -6.400 1.66e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.
##
## Residual standard error: 3.978 on 14 degrees of freedom
```

Interaction plot



Throwing example

```r
throw <- read.csv("AndersThrow.csv")
with(throw, cor.test(run, distance))
```

```
##
##  Pearson's product-moment correlation
##
## data:  run and distance
## t = 0.79594, df = 6, p-value = 0.4564
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5057812  0.8324307
## sample estimates:
##      cor
## 0.309035
```

Throwing example (continued)

```r
summary(lm(distance ~ 0 +hand + hat + boots + hand:hat + hand:bo
```

```
##
## Call:
## lm(formula = distance ~ 0 + hand + hat + boots + hand:hat + h
##     data = throw)
##
## Residuals:
##      1      2      3      4      5      6      7      8
## -0.875 -0.625  0.875  0.625  0.875  0.625 -0.875 -0.625
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## handleft               4.875      1.317   3.702   0.0659 .
## handright              8.125      1.317   6.170   0.0253 *
## hatyes                -0.750      1.521  -0.493   0.6707
## bootsyes               1.250      1.521   0.822   0.4975
## handright:hatyes      -1.500      2.151  -0.697   0.5577
## handright:bootsyes    -0.500      2.151  -0.232   0.8378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

Interaction plot

```
with(throw, interaction.plot(hat, hand, distance))
```