MONASH University

MONASH BUSINESS SCHOOL

ETC3580: Advanced Statistical Modelling

Week 1: Visualizing linear models

Outline

Linear Models

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

- Response: y
- Predictors: $x_1, \ldots, x_p$
- Error: $\varepsilon \sim$ IID

Let $\boldsymbol{y} = (y_1, \ldots, y_n)'$, $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)'$, $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)'$ and

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{1,1} & x_{2,1} & \ldots & x_{p,1} \\ 1 & x_{1,2} & x_{2,2} & \ldots & x_{p,2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1,n} & x_{2,n} & \ldots & x_{p,n} \end{bmatrix} \qquad \text{(the model matrix).}$$

Linear Models

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

- Response: y
- Predictors: $x_1, \ldots, x_p$
- Error: $\varepsilon \sim$ IID

Let $\boldsymbol{y} = (y_1, \ldots, y_n)'$, $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)'$, $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)'$ and

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{1,1} & x_{2,1} & \ldots & x_{p,1} \\ 1 & x_{1,2} & x_{2,2} & \ldots & x_{p,2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1,n} & x_{2,n} & \ldots & x_{p,n} \end{bmatrix} \quad \text{(the model matrix).}$$

Then

$$\boldsymbol{y} = \boldsymbol{X}\beta + \boldsymbol{\varepsilon}.$$

Matrix formulation

Least squares estimation

Minimize: $(\boldsymbol{y} - \boldsymbol{X}\beta)'(\boldsymbol{y} - \boldsymbol{X}\beta)$

Matrix formulation

Least squares estimation

Minimize: $(\boldsymbol{y} - \boldsymbol{X}\beta)'(\boldsymbol{y} - \boldsymbol{X}\beta)$

Differentiate wrt β gives

The "normal" equations

$$\hat{\beta} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$$

Likelihood

If the errors are iid and normally distributed, then

$$\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{X}\beta, \sigma^2\boldsymbol{I}).$$

Likelihood

If the errors are iid and normally distributed, then

$$\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{X}\beta, \sigma^2\boldsymbol{I}).$$

So the likelihood is

$$L = \frac{1}{\sigma^n (2\pi)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\beta)'(\boldsymbol{y} - \boldsymbol{X}\beta)\right)$$

Likelihood

If the errors are iid and normally distributed, then

$$\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{X}\beta, \sigma^2\boldsymbol{I}).$$

So the likelihood is

$$L = \frac{1}{\sigma^n(2\pi)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X}\beta)'(\boldsymbol{y}-\boldsymbol{X}\beta)\right)$$

which is maximized when $(\boldsymbol{y}-\boldsymbol{X}\beta)'(\boldsymbol{y}-\boldsymbol{X}\beta)$ is minimized.

Likelihood

If the errors are iid and normally distributed, then

$$\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{X}\beta, \sigma^2\boldsymbol{I}).$$

So the likelihood is

$$L = \frac{1}{\sigma^n (2\pi)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\beta)'(\boldsymbol{y} - \boldsymbol{X}\beta)\right)$$

which is maximized when $(\boldsymbol{y} - \boldsymbol{X}\beta)'(\boldsymbol{y} - \boldsymbol{X}\beta)$ is minimized.

So MLE = OLS.

Outline

R modelling notation

```r
fit <- lm(y ~ x1 + x2 + x3,
  data=tibble)
```

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon, \qquad \varepsilon \sim N(0, \sigma^2)$$

```
## # A tibble: 200 x 4
##         y    x1    x2    x3
##     <dbl> <dbl> <dbl> <dbl>
##   1  78.6  51.3  47.8   2.8
##   2  19.4  76.6  37.8   1.05
##   3  97.6  26.1  24.2   4.63
##   4  81.5  21.6 -19.3   1.9
```

Useful helper functions

Base functions

- `summary`
- `coef`
- `fitted`
- `predict`
- `residuals`

`broom` functions

- `tidy`
- `augment`
- `glance`

R formulas

Categorical predictors:

- R will create the required dummy variables from a categorical *factor*.
- The first level is used as the reference category.
- Use `relevel` to change the reference category

Expression	Description
`y ~ x`	Simple regression
`y ~ 1 + x`	Explicit intercept
`y ~ -1 + x`	Through the origin
`y ~ x + I(x^2)`	Quadratic regression
`y ~ x1 + x2 + x3`	Multiple regression
`sqrt(y) ~ x + I(x^2)`	Transformed
`y ~ . -x1`	All variables except `x1`

Outline

Partial residuals

$$\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{X}\hat{\beta}$$

denote residuals for a given model fit.

Then the partial residuals for variable j are given by

$$\boldsymbol{r}_j = \boldsymbol{y} - \boldsymbol{X}_{-j}\hat{\beta}_{-j}$$

where the $-j$ subscript indicates the removal of the jth column/element.

- Equivalent to y adjusted for all variables other than x_j.
- Plotting $\boldsymbol{r}_j$ vs $\boldsymbol{x}_j$ shows the relationship of $\boldsymbol{y}$ vs $\boldsymbol{x}_j$ after adjustment.

Conditional plots

- Slope of regression of $\boldsymbol{r}_j$ on $\boldsymbol{x}_j$ is β_j.
- Conditional plots show $\boldsymbol{r}_j + \boldsymbol{x}_{-j|m}\beta_{-j}$ vs $\boldsymbol{x}_j$, where $\boldsymbol{x}_{-j|m}$ corresponds to median of numeric variables and mode for factors.
- Let $\boldsymbol{x}^{*\prime}$ denote row of design matrix constructed from $x_j = x$ and $\boldsymbol{x}_{-j|m}$. Then equation of line is $\boldsymbol{x}^{*\prime}\hat{\beta}$ and standard error at x is

$$\mathrm{se}(x) = \sqrt{\boldsymbol{x}^{*\prime}\mathrm{Var}(\hat{\beta})\boldsymbol{x}^{*}}.$$

- Construct confidence interval using

$$\boldsymbol{x}^{*\prime}\hat{\beta} \pm t_{n-p,1-\alpha/2}\mathrm{se}(x)$$

Visualization using visreg

- `visreg(ls_object)`
- `visreg(ls_object, "xvar", gg=TRUE)`

Outline

Spotting an interaction in data

Interactions occur when effect of one predictor on response changes with value of another predictor.

Spotting an interaction in data

Interactions occur when effect of one predictor on response changes with value of another predictor.

To see 2-way interactions in raw data

- Plot y vs x_j for different values of x_k
- Plot y vs x_k for different values of x_j

Spotting an interaction in data

Interactions occur when effect of one predictor on response changes with value of another predictor.

To see 2-way interactions in raw data

- Plot y vs x_j for different values of x_k
- Plot y vs x_k for different values of x_j

To see 2-way interactions after adjustment

- Plot r_j vs x_j for different values of x_k
- Plot r_k vs x_k for different values of x_j

Spotting an interaction in data

Interactions occur when effect of one predictor on response changes with value of another predictor.

To see 2-way interactions in raw data

- Plot y vs x_j for different values of x_k
- Plot y vs x_k for different values of x_j

To see 2-way interactions after adjustment

- Plot r_j vs x_j for different values of x_k
- Plot r_k vs x_k for different values of x_j

Much harder to see higher-order interactions

Interactions

Interactions:

- One possible type of interaction is obtained by multiplying the relevant columns of the model matrix.
- Use `a:b` for the interaction between `a` and `b`.
- Use `a*b` to mean `a + b + a:b`
- Need to specify explicit functions for other types of interaction. e.g., `I(a/b)`

Limited order interactions:

- Interactions up to 2nd order can be specified using the `^` operator.
- `(a+b+c)^2` is identical to `(a+b+c)*(a+b+c)`

Nested factors:

- `a + b %in% a` expands to `a + a:b`

Interpretation

- Each coefficient gives effect of one unit increase of predictor on response variable, *holding all other variables constant*.
- Be careful with interactions: can't easily interpret main effects when predictors interact.

Interpretation

- Each coefficient gives effect of one unit increase of predictor on response variable, *holding all other variables constant*.
- Be careful with interactions: can't easily interpret main effects when predictors interact.

Visualization using visreg

- `visreg2d(ls_object, "xvar1", "xvar2")`
- `visreg2d(ls_object, "xvar1", "xvar2", plot.type='persp')`
- `visreg2d(ls_object, "xvar1", "xvar2", plot.type='rgl')`
- `visreg2d(ls_object, "xvar1", "xvar2", plot.type='gg')`

Outline

Hypothesis testing

- Use F-tests between models:
  - Model 1: p_1 parameters.
  - Model 2 (nested within Model 1): p_2 parameters

$$F = \frac{(\mathrm{RSS}_2 - \mathrm{RSS}_1)/(p_1 - p_2)}{\mathrm{RSS}_1/(n - p_1)} \sim F_{p_1 - p_2, n - p_1}$$

- Helper functions: `anova`, `drop1`
- If one term dropped, this is equivalent to a t-test on coefficient.

Hypothesis testing

`anova(model)`

- provides sequential testing of terms (conditional on all previous terms)
- Order of terms will usually affect the p-values.
- Uses “Type 1” SS

`anova(model1, model2)`

- Tests two nested models.
- Avoids ordering problems

`drop1(model, test="F")`

- Equivalent to series of `anova(model1, model2)` calls where `model2` drops one variable at a time.
- Equivalent to “Type 3” SS

Outline

Akaike's Information Criterion

Akaike's Information Criterion

$$\text{AIC} = -2\log L + 2q$$

where q is the number of parameters in the model.

- Select model with smallest AIC

Akaike's Information Criterion

Akaike's Information Criterion

$$\text{AIC} = -2\log L + 2q$$

where q is the number of parameters in the model.

- Select model with smallest AIC
- For Gaussian errors:

$$L = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\beta)'(\boldsymbol{y} - \boldsymbol{X}\beta)\right)$$

$$\begin{aligned} -2\log L &= n\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\beta)'(\boldsymbol{y} - \boldsymbol{X}\beta) \\ &= c + \frac{\text{SSE}}{2\sigma^2} \end{aligned}$$

Akaike's Information Criterion

Akaike's Information Criterion

$$\text{AIC} = -2\log L + 2q$$

where q is the number of parameters in the model.

- Select model with smallest AIC
- For Gaussian errors:

$$L = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\beta)'(\boldsymbol{y} - \boldsymbol{X}\beta)\right)$$

$$\begin{aligned} -2\log L &= n\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\beta)'(\boldsymbol{y} - \boldsymbol{X}\beta) \\ &= c + \frac{\text{SSE}}{2\sigma^2} \end{aligned}$$

`extractAIC()` used by `step()` handles c and q differently from `AIC()`

Variable selection

`step()` will minimize AIC using backwards selection

Best model with only main effects

```
mod1 <- lm(y ~ x1 + x2 + x3, data=data) %>%
  step()
```

Best model with up to 2-way interactions

```
mod2 <- lm(y ~ (x1 + x2 + x3)^2, data=data) %>%
  step()
```

Best model with up to 3-way interactions

```
mod3 <- lm(y ~ (x1 + x2 + x3)^3, data=data) %>%
  step()
```

Variable selection and inference

- Do not use coefficient t-tests for variable selection.
- Beware of *any* statistical tests after variable selection.
- Confidence intervals after variable selection are too narrow.
- Variable selection is most useful for prediction. If you are only interested in inference, don’t do it!