Linear Methods for Regression

Matthew S. Shotwell, Ph.D.

Department of Biostatistics
Vanderbilt University School of Medicine
Nashville, TN, USA

January 17, 2020

Simple vs Multiple Linear Regression

- We've mostly discussed "simple" linear regression; one predictor:

$$Y = \beta_0 + \beta_1 X_1$$

- In general, we want to consider many predictors:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

- Sometimes written in matrix notation:

$$Y = X\beta$$

  where X is a row vector of predictors and β is a column vector of coefficients/parameters

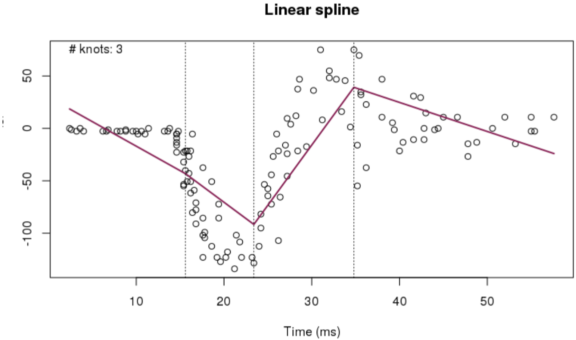
Where do the predictors come from?

The predictors $X_1, \dots, X_p$ can be any of:

- quantitative inputs
- transformations e.g., $X_2 = \sqrt{X_1}$
- basis expansions of one or more predictors, e.g., linear splines
- numeric "dummy" coding of categories, e.g.
  if $G =$ 'green' $X_1 = 1$ and 0 otherwise
  if $G =$ 'red' $X_2 = 1$ and 0 otherwise
- interactions, e.g., $X_3 = X_1 \cdot X_2$

What does "linear" mean?

The word "linear" in "linear model" means that Y is a linear function of the parameters β. For a fixed input X, the association between Y and any β_j is linear. "Linear model" does NOT mean that Y is a linear function of the inputs (think about linear splines).



How do we fit linear models?

- generally fit by minimizing training error:

$$\overline{\text{err}}(\beta) = \sum_{i=1}^{n} L(y_i, x_i\beta))$$

  where y_i and x_i are training examples and $x_i\beta$ is in matrix notation: $x_i\beta = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$

- almost always use squared error loss

$$\overline{\text{err}} = \sum_{i=1}^{n} (y_i - x_i\beta)^2$$

- can sometimes have a roughness penalty, e.g., lasso

$$\overline{\text{err}} = \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

How do we fit linear models in R?

Although the homework asks you to fit models manually, but minimizing average loss, there are usually more convenient ways to fit most models in R:

- Linear model with squared error loss: `lm`
- Linear models with penalized squared error loss: `glmnet`

After a linear model is fit

Once we've fit a linear model:

- the values of β that minimize training error are called "estimates" and denoted $\hat{\beta}$ (vector)
- can make predictions $\hat{Y} = X\hat{\beta}$
- $r_i = \hat{y}_i - y_i$ are called "residuals"
- we want residuals to be small
- R-squared (R^2): fraction of variability among y_i that can be explained by model; bigger means model more closely fits the training data
- when R^2 is 1, training error is zero

Code example

`linear-regression-examples.R`

Bias-variance tradeoff

Linear model design affects:

- number of model parameters
- bias-variance tradeoff
- quality of predictor (test error!)

Linear model design is a tuning “parameter”

Linear model design

- linear model design is a manual process
- must specify maximum complexity, e.g., using splines, interactions, etc
- modern ML methods (e.g., neural networks) do this in a more algorithmic fashion
- once maximum complexity specified, we have methods to tune the model by “weeding out” parts of the linear model that don’t contribute much, e.g., best-subset selection, stepwise selection, lasso, ridge