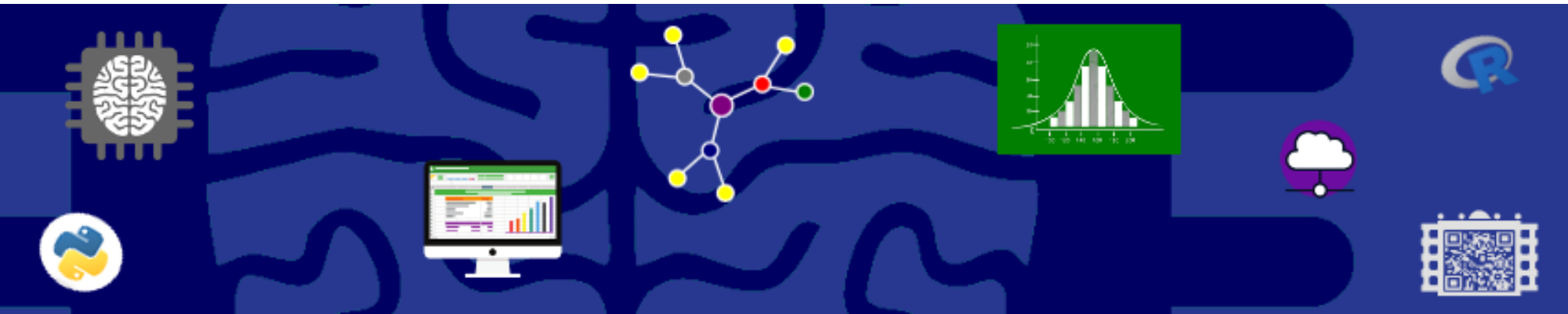
Introduction to Data Science

Winter Semester 2018/19

Oliver Ernst

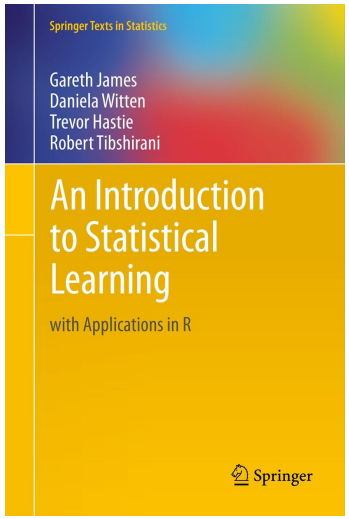
TU Chemnitz, Fakultät für Mathematik, Professur Numerische Mathematik

Lecture Slides



Organizational Issues

- Module: Introduction to Data Science (M24, Einführung in Data Science)
- Class web page: www.tu-chemnitz.de/mathematik/numa/lehre/ds-2018/
- Lecturer: Oliver Ernst (oernst@math.tu-chemnitz.de)
- Class meets Mo 9:15 and Thu 13:45 in Rh70 Room B202.
- Course Assistant: Jan Blechschmidt (jan.blechschmidt@math.tu-chemnitz.de)
- Lab exercises Tue 13:45 Rh39/41 Room 738.
- 50 % of coursework must be handed in correctly to register for exam.
- Oral exam, 30 minutes, at end of teaching term (February)
- **Textbook:** James, Witten, Hastie &Tibshirani. Introduction to Statistical Learning. Springer, 2013.



Note: the majority of images appearing in these lecture slides are taken from this textbook; permission for their use for teaching is gratefully acknowledged.

Contents I

Contents II

Contents III

Contents

What is Data Science?

Point of departure

- **Explosion of data:** sensor technology (e.g. weather); purchase histories (customer loyalty programs, fraud detection); soon every person on the planet ($\approx$ 8 billion) will have a smartphone and generate GPS traces, trillions of photos each year; DNA sequencing ...

  Nagel (2018):

  - Gigabyte (10^9); your hard disk
  - Terabyte (10^{12}); Facebook - 500 TB/day[1]
  - Petabyte (10^{15}); CERN - LHC: 15 PB/year
  - Zettabyte (10^{21}); mobile network traffic 2016
  - Yottabyte (10^{24}); event analysis
  - Brontobyte (10^{27}); sensor data from the IoT (Internet of Things)
- **Heterogeneity:** besides big, data unstructured, noisy, heterogeneous, not collected systematically.
- **Enabling technologies:** storage capacity; computing hardware; algorithms; 350 years of statistics; 100 years of numerical analysis.

[1] Facebook currently has 2.2 billion users worldwide,
Can Mark Zuckerberg Fix Facebook Before It Breaks Democracy? New Yorker, 09/2018

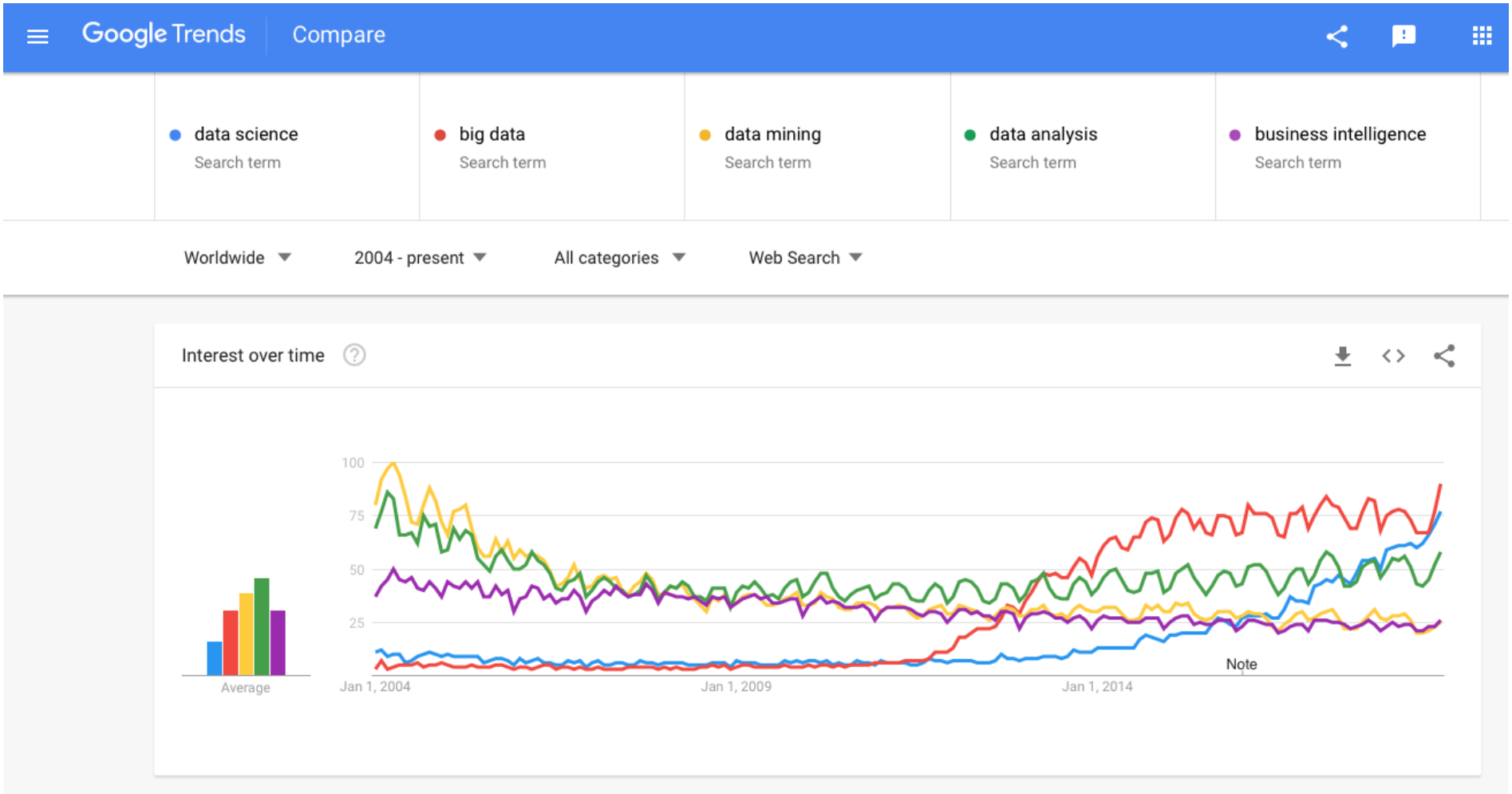
What is Data Science

Buzz words

- Data Science
- Big Data
- Data Mining
- Data Analysis
- Business Intelligence
- Analytics

What is Data Science

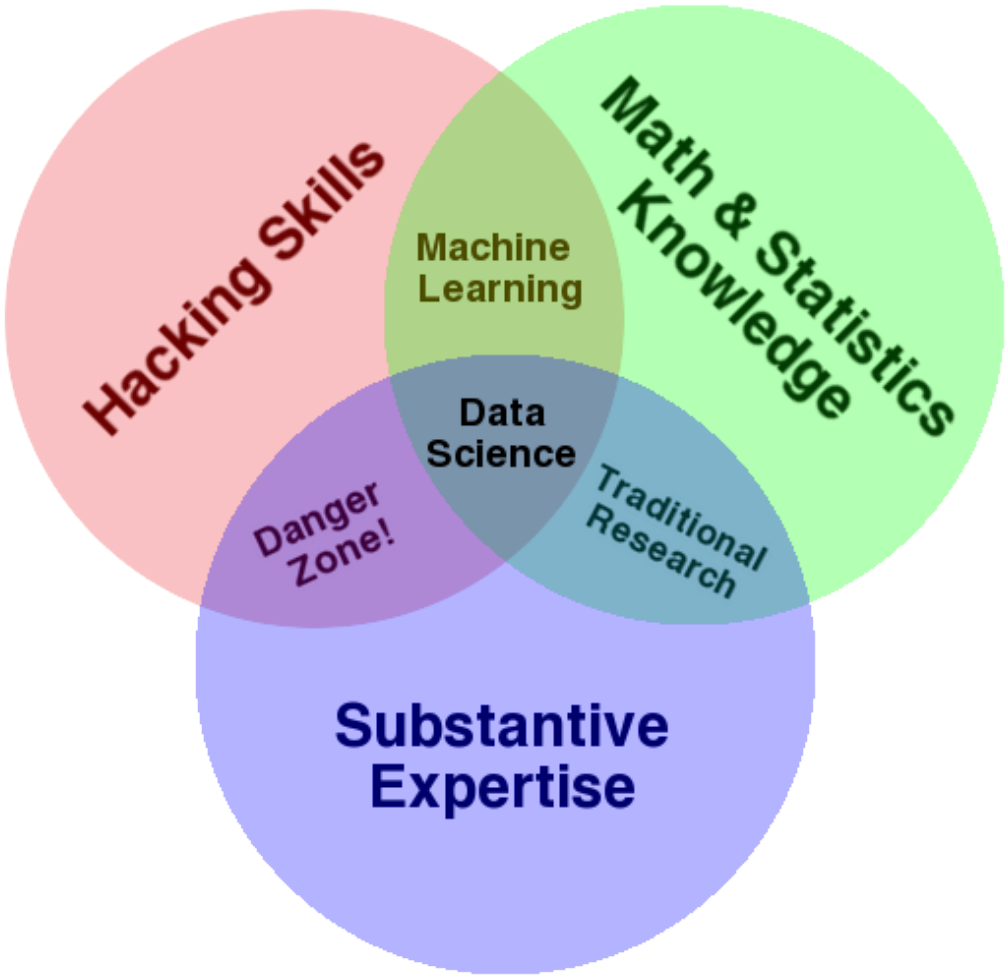
Buzz words



trends.google.com

What is Data Science

Conway's data science Venn diagram (2010) ...

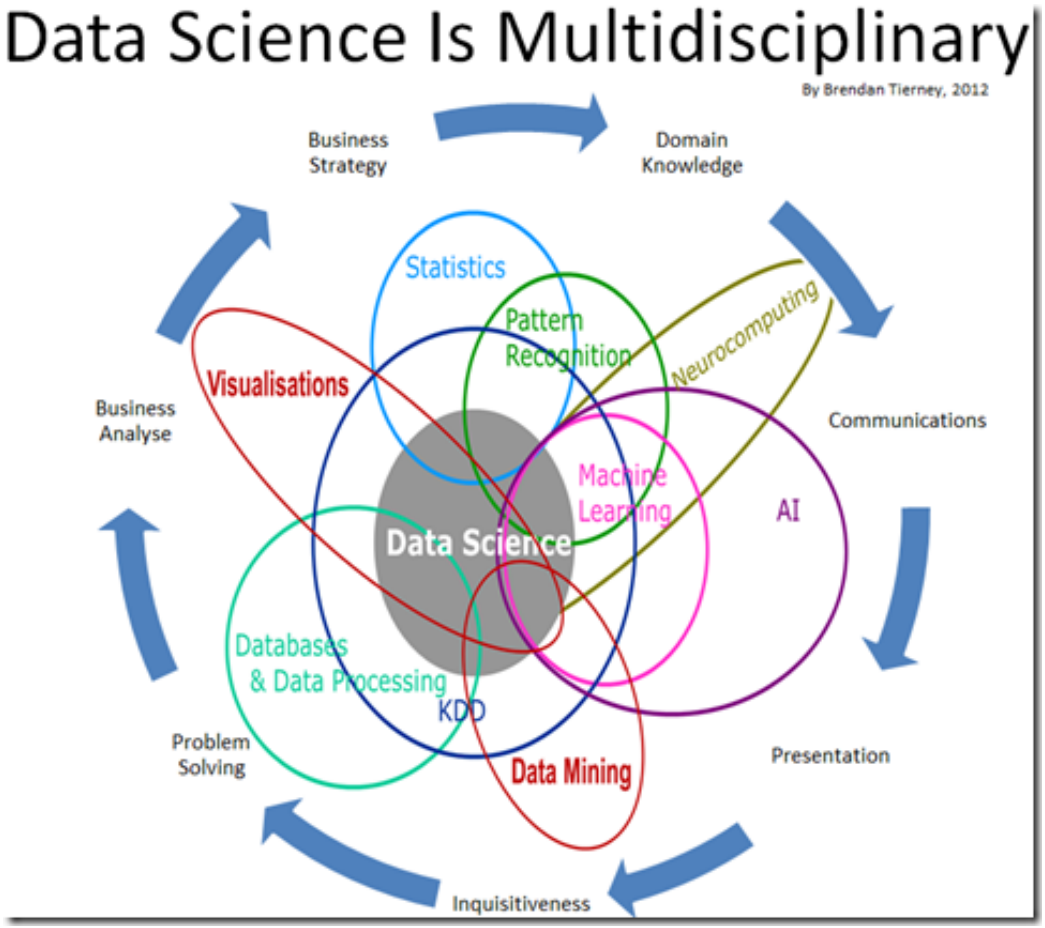


http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

What is Data Science

Conway's data science Venn diagram (2010) . . . and variations

Brendan Tierney (2012)



http://www.prooffreader.com/2016/09/battle-of-data-science-venn-diagrams.html

Joel Grus (2013), at the time of the Snowden revelations

What is Data Science

. . . and from the twitterverse:

"A data scientist is a statistician who lives in San Francisco."

"Data Science is statistics on a Mac"

"A data scientist is someone who is better at statistics than any software engineer and better at software engineering than any statistician."

Josh Wills, Director of Data Science at Cloudera

"A data scientist is someone who is worse at statistics than any statistician and worse at software engineering than any software engineer."

Will Cukierski, Data Scientist at Kaggle

https://twitter.com/cdixon/status/428914681911070720

What is Data Science

Donoho's first definition

David Donoho (2015)[2]

Classifies the activities of *Greater Data Science* (GDS) into six divisions.

(GDS1) Data Gathering, Preparation and Exploration
experimental design; collect; reformat, treat anomalies, expose unexpected features.

(GDS2) Data Representation and Transformation
databases; represent e.g. acoustic, image, sensor, network data.

(GDS3) Computing with Data
SW engineering; cluster computing; developing workflows.

(GDS4) Data Visualization and Presentation
classical EDA plots; high-dimensional or streaming data

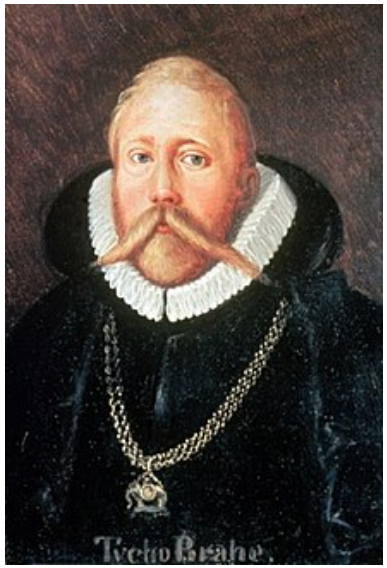
(GDS5) Data Modeling
generative vs. predictive modeling

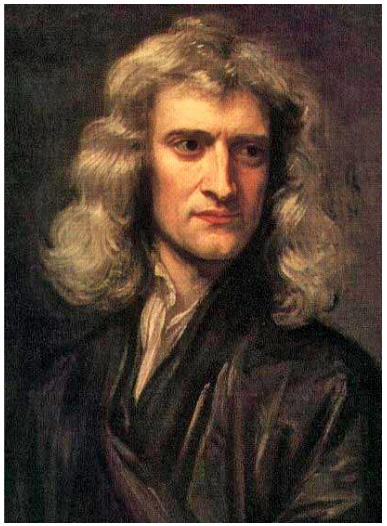
(GDS6) Science about Data Science
evaluate results, processes, workflows; analysis od methods.

[2] 50 Years of Data Science. Journal of Computational and Graphical Statistics. 26 (2017)

Some Examples

Brahe, Kepler & Newton





- Tycho Brahe (1546–1501) Accurate and detailed observations of positions of celestial bodies over many years. (data collection)
- Johannes Kepler (1571–1630) Based on Brahe's observations, derives laws of orbital motion. (model fitting)
- Isaac Newton (1643–1727) Derives Newtonian mechanics, from which Kepler's laws follow. (deeper underlying truths)

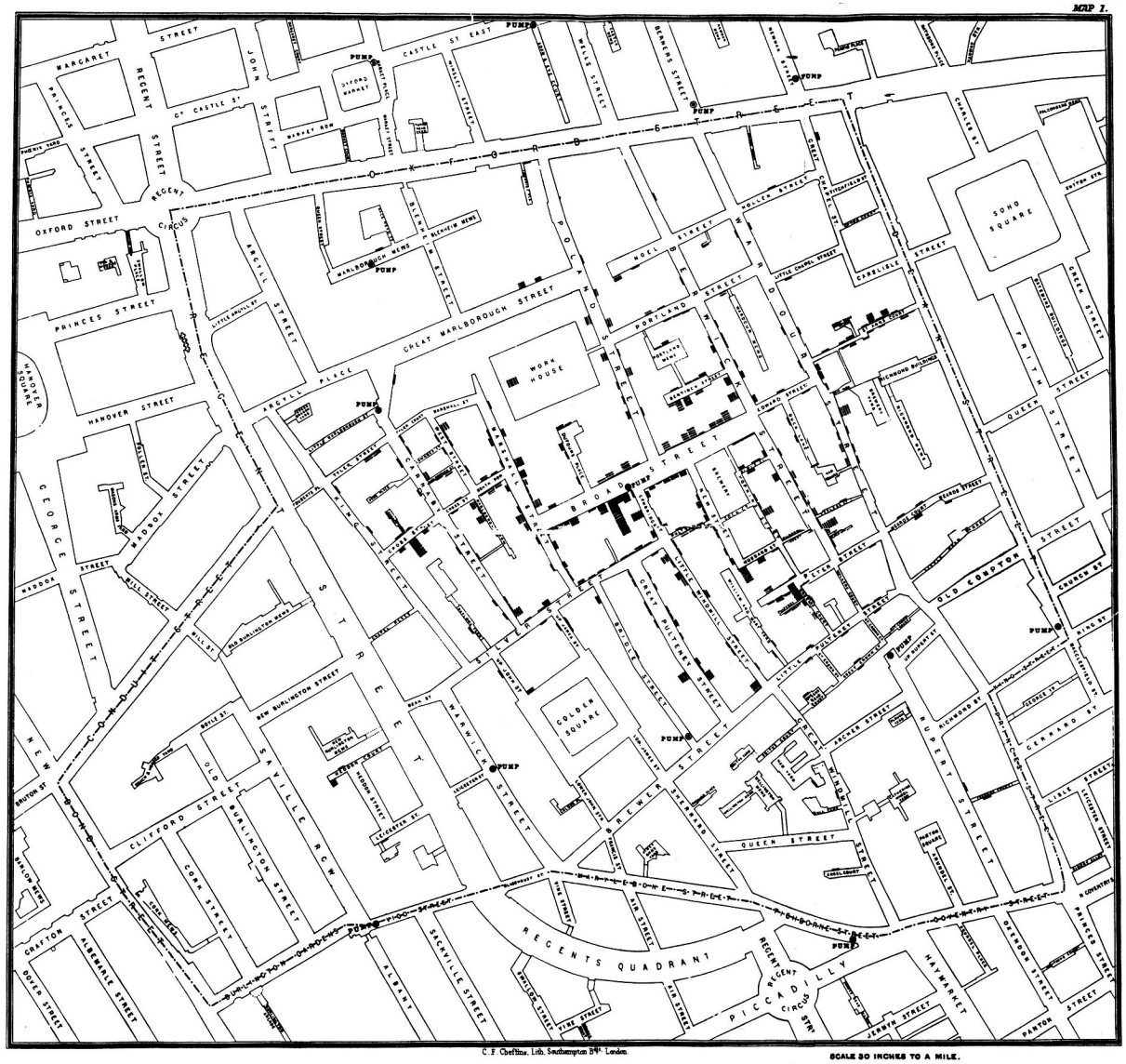
Some Examples

1854 cholera outbeak, Broad Street, London

- Mid 19-th century London was plagued by devastating outbreaks of Cholera.
- Competing medical theories of the time: miasma ("bad air") vs. as yet unidentified microbes.
- London doctor John Snow, microbe theory proponent, conjectured that Cholera-inducing microbes spread via drinking water.
- During a particularly severe outbreak in Soho district in 1854, Snow recorded the deaths on a *dot map*.
- The clustering pattern evident from the map pointed to the *Broad Street pump* as the likely source of contamination.
- Based on this evidence, he was able to convince the authorities to disable the pump, and Cholera incidence in the neighborhood declined.

Some Examples

1854 cholera outbeak, Broad Street, London



Original Snow map of 1854. Stacked rectangles indicate Cholera cases. (Wikipedia)

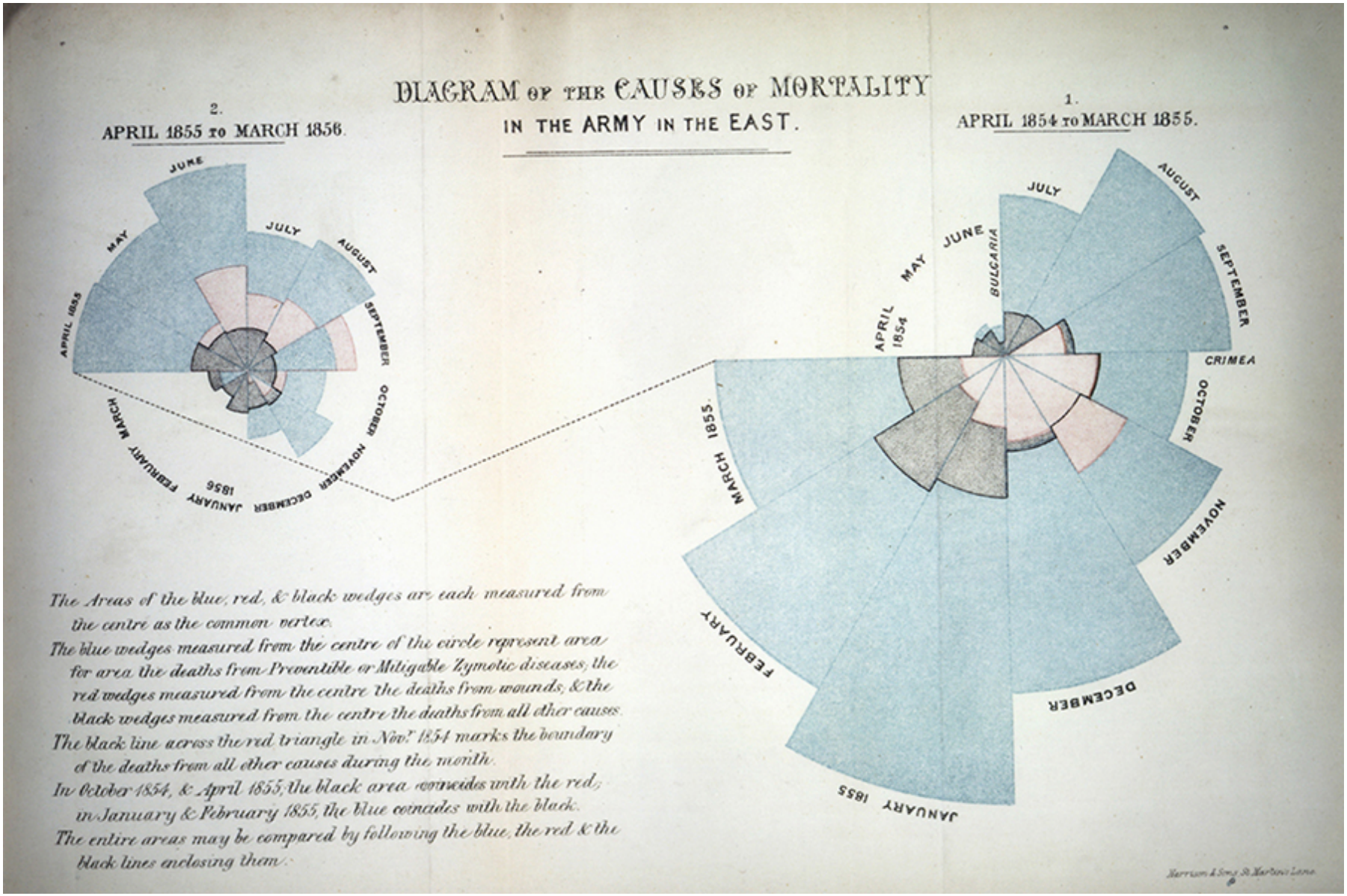
Some Examples

Florence Nightingale's data visualizations

- During the *Crimean War* (1853–1856), Florence Nightingale, a well-educated British social reformer and statistician, was in charge of a group of nurses tending the wounded British soldiers at Selimiye Barracks near Istanbul.
- She systematically documented the horrific medical conditions she encountered on arrival and reported her findings to the authorities, effecting dramatic hygenic improvements.
- Nightingale pioneered the graphical display of statistical data in pie-chart-like diagrams known then as *"coxcombs"*.
- In 1859, Nightingale was elected the first female member of the Royal Statistical Society. In 1874 she became an honorary member of the American Statistical Association.

Some Examples

Florence Nightingale's data visualizations



Some Examples

Data science today

- **Precision medicine:** the cost of sequencing the human genome has dropped by a factor of one million (3.5 billion to 1000 USD) in the last decade. New algorithms achieve a full genome analysis in 24 hours, looking at 6 billion genes, vs. standard technology taking 3 months, accounting for 5 genes only.

  E. A. Ashley. Towards Precision Medicine. Nature Reviews Generics (2017).
- **Predicting elections:** in 2008, the American statistician Nate Silver correctly predicted the outcome in 49 of 50 states in the US presidential election. In 2012 he correctly predicted in all 50 states.

  N. Silver. The Signal and the Noise. Penguin (2012).
- **Influencing elections:** A classical model for personality traits was combined with data gleaned from the accounts of a large number of Facebook users to produce microtargeted ads in recent political campains, including the 2016 US presidential election and the UK referendum on leaving the EU. Besides litigation, this has led to concerns over privacy issues and the possibilities for systematic disinformation based on data science techniques.

  The Data that Turned the World Upside Down. Motherboard Vice (2017).

Some Examples

Data science today

- **Beating humans at games**. 1997 IBM's Deep Blue beat world chess grandmaster Garry Kasparov. 2011 IBM's Deep Blue won at the TV game show 'Jeopardy'. In 2016 Google's Deep Mind beat world champion Lee Sedol at the ancient chinese game 'Go'.

  B. Dickson.. All the important games artificial intelligence has conquered. (2018).
- **Autonomous vehicles:** Automobiles already contain multiple software systems for control, navigation, communication and entertainment. By combining multiple sensor input (cameras, LIDAR, RADAR), artificial intelligence systems are moving ever closer to cars which can drive autonomously. Daimler has promised driverless cars by 2020, Ford by 2021.
- **Medical diagnostics:** In 2017, the Stanford Machine Learning Group introduced the algorithm CheXNet, a neural network which outperformed practicing radiologists in detecting pneumonia from chest X-ray images. In 2018, the same group presented a machine learning system which can summarize the key findings of radiology reports for x-ray images nearly as well as radiologists can.
- **Natural language processing:** Google, Google translate, Google ngrams,

What is Data Science?

Donoho's second definition/paradigm

David Donoho. Data Science: The End of Theory?
Lecture at Data Science Kickoff event, U Vienna, 22.06.2018

> **Def.** *Data Science concerns the recognition, formalization and exploitation of data phenomenology emerging from digital transformation of business, society and science itself.*

Hallmarks of DS activities:

Developments	Examples
Accumulating digital assets Emergent data/methodology universe Emergent technique of exploitation	

https://www.youtube.com/watch?v=PekBM76z2qE

What is Data Science?

Donoho's second definition/paradigm

In his Vienna lecture Donoho provides a series of examples for activities which qualify as DS under this paradigm, among these

- a study of false discovery rates in medical literature [Jager & Leek, 2014],
- a cross-model study on detection methods for ovarian cancer,
- the enterprise of machine learning, as well as
- the use of deep learning to accelerate existing algorithms.

What is Data Science?

Donoho's second definition/paradigm

Biostatistics (2014), **15**, 1, *pp.* 1–12
doi:10.1093/biostatistics/kxt007
Advance Access publication on September 25, 2013

R

An estimate of the science-wise false discovery rate and application to the top medical literature

LEAH R. JAGER
Department of Mathematics, United States Naval Academy, Annapolis, MD 21402, USA
JEFFREY T. LEEK*
Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, Baltimore, MD 21205, USA
jleek@jhsph.edu

SUMMARY

The accuracy of published medical research is critical for scientists, physicians and patients who rely on these results. However, the fundamental belief in the medical literature was called into serious question by a paper suggesting that most published medical research is false. Here we adapt estimation methods

Developments	Examples
Accumulating digital assets	database, SW for text scraping
Emergent data/methodology universe	all Pub-Med abstracts
Emergent technique of exploitation	stats about scientific activity

What is Data Science?

The end of theory?

- In 2008 an article in Wired Magazine[3] suggested that emerging data science techniques would render the traditional scientific method based on hypotheses obsolete, implying that automated data mining will lead directly to new discoveries.
- This is particularly relevant for areas of research such as bioinformatics, systems biology, epidemiology and ecology.
- This prompted an intense discussion on data-driven vs. hypothesis-driven research.
- The "No-Free-Lunch theorem" states that there can be no learning without knowledge.
- Currently, a "mathematization" of data science is setting in, leaving ample room for theoretical analysis, likely leading to improved methods. (Compare with the development of other computational disciplines such as the finite element method or uncertainty quantification)

[3] The end of theory:the data deluge makes the scientific method obsolete. Wired 6/2008.

Contents

Contents

- Given a data set containing the sales numbers for a given product in 200 markets, allocate an advertising budget across the three media channels **TV**, **radio** and **newspaper**.
- The sales numbers for each medium are available for different advertising budget values.
- We will try to model the dependence of sales on advertising budgets.
- Terminology:

X_1 : TV budget
X_2 : radio budget
X_3 : newpaper budget

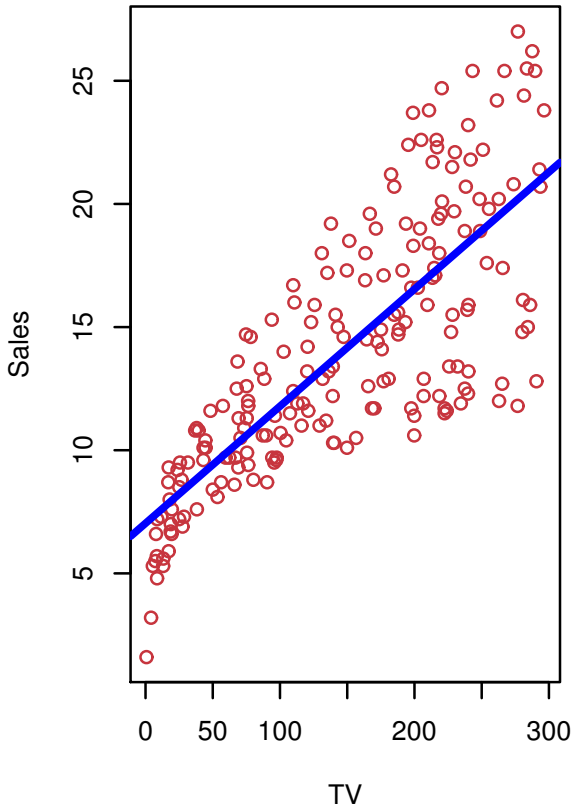
} input variables, predictors, independent variables, variables, features

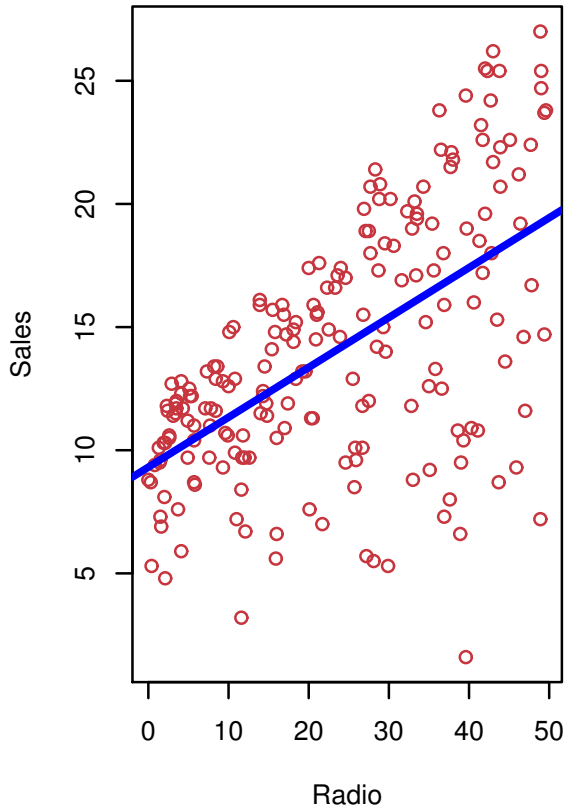
Y : sales

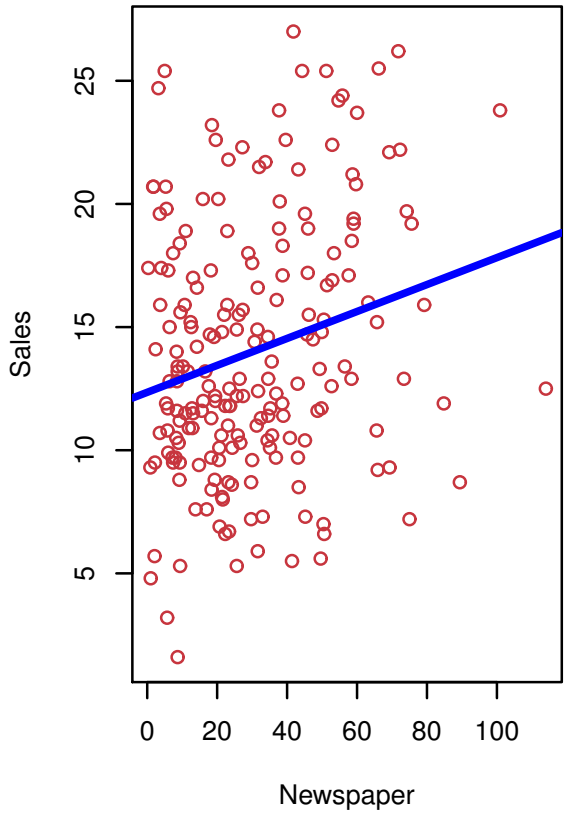
response, dependent variables.

Learning Theory

Example: Advertising channels







$$Y = f(X) + \varepsilon$$

$X = (X_1, \ldots, X_p)$, $p = \#$ predictors,

ε : random error term, $\mathbf{E}[\varepsilon] = 0$, (2.1)

f : systematic information X provides about Y.

Learning Theory

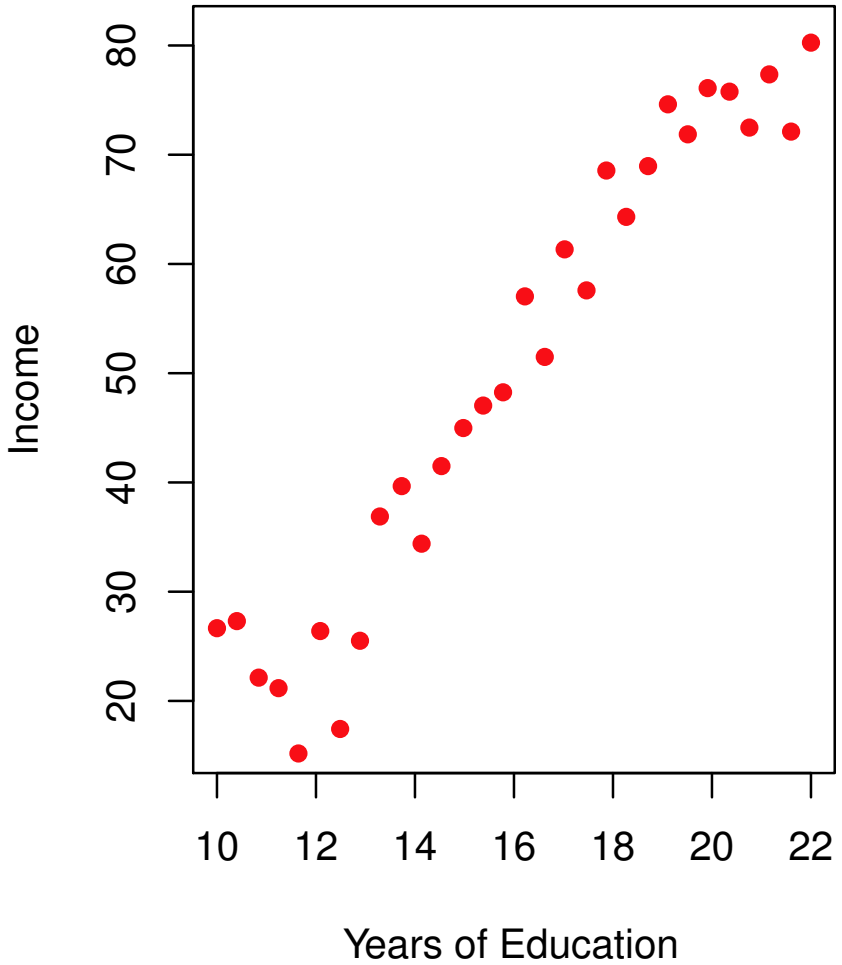
Example: Income

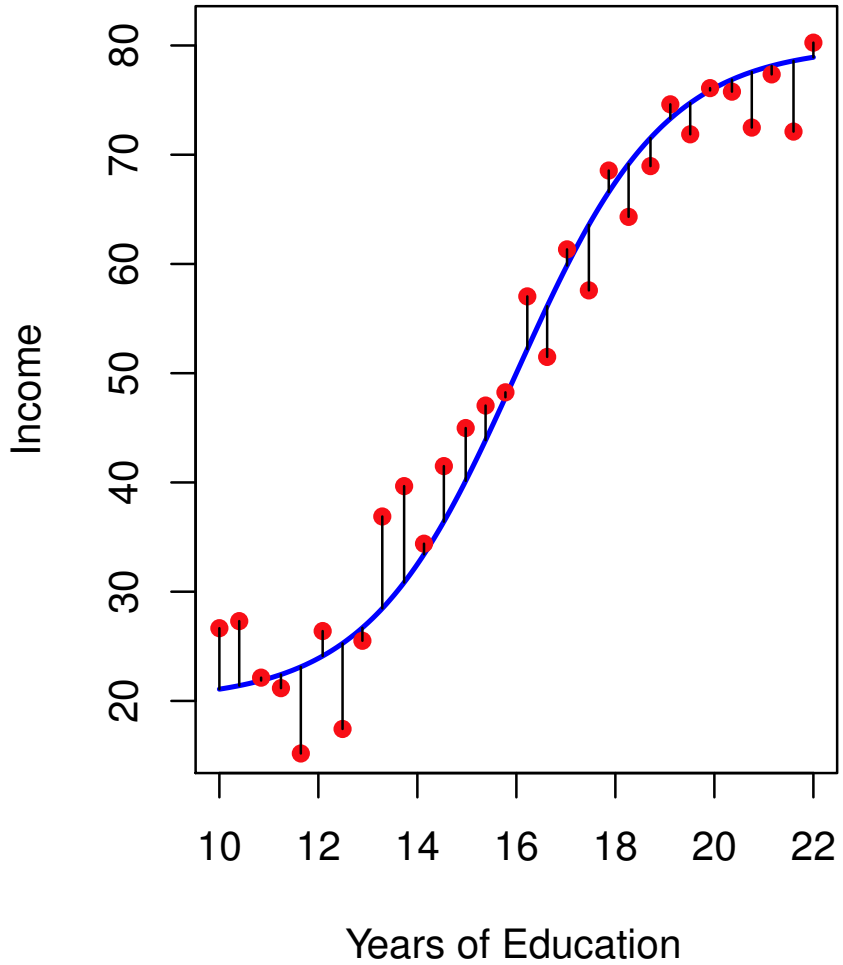
- Given data set showing **income** against **years of education** for 30 people.
- Objective: determine function f relating **income** as response to **years of education** as predictor.
- f generally unknown, must be estimated from the data.
- Here: data simulated, so f available.
- In another data set, **income** is given with respect to two input variables: **years of education** and **seniority**.

- Statistical learning is concerned with techniques for estimating f from a data set.

Learning Theory

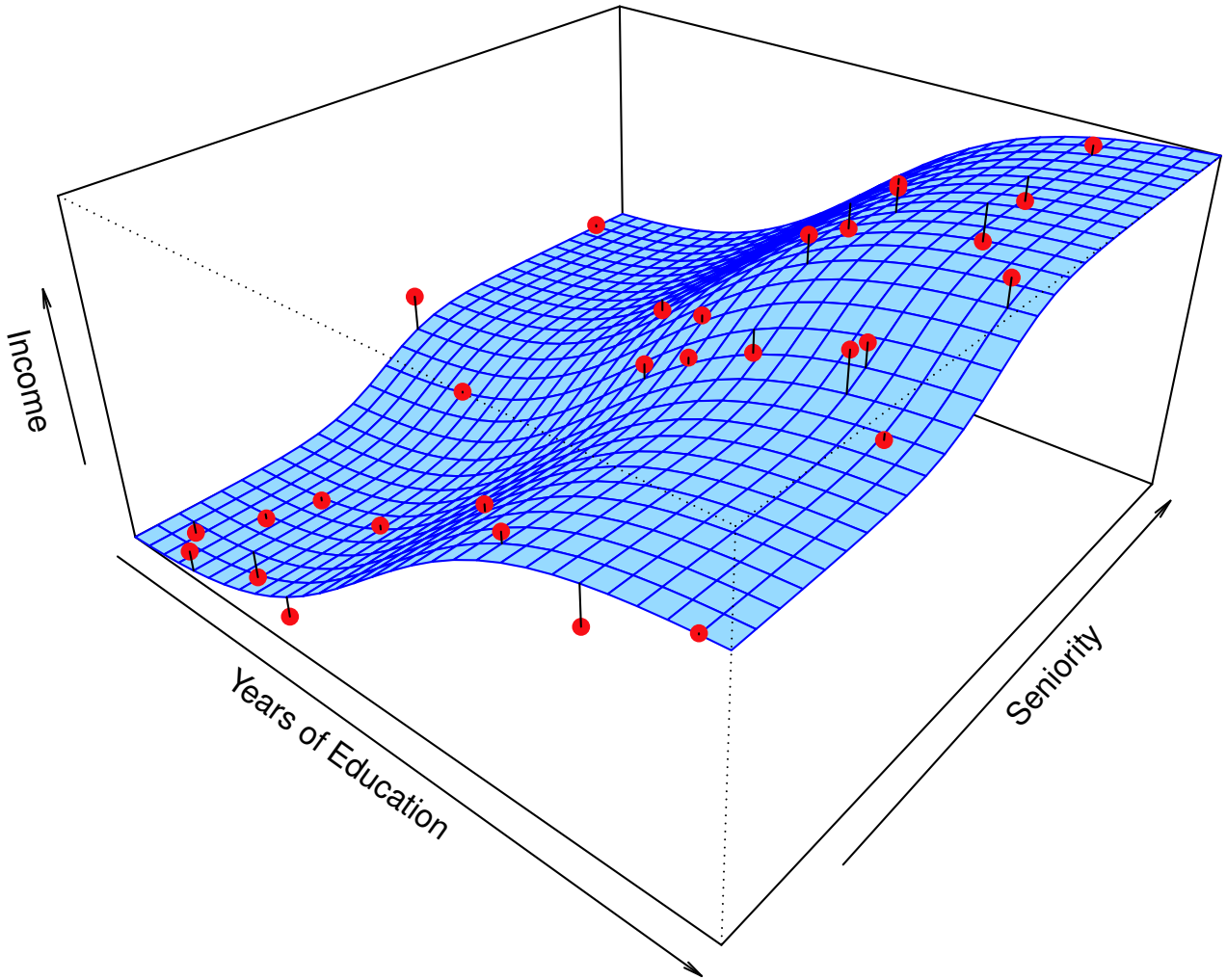
Example: Income





Learning Theory

Example: Income



Two main reasons for estimating f:
prediction
and
inference.

- Suppose inputs X readily available, but outputs Y difficult to obtain.
- Since errors average out, predict Y using

$$\hat{Y} = \hat{f}(X), \qquad \begin{aligned} &\hat{f} : \text{estimate for } f, \\ &\hat{Y} : \text{prediction for } Y = f(X). \end{aligned}$$

- Often $\hat{f}$ only available as a **black box**, i.e., a procedure for generating $\hat{Y}$ given X.

Example:

$X_1, \ldots, X_p$: characteristics of a patient's blood samples, measured in lab.
Y : patient's risk for severe adverse reaction to particular drug.

For obvious reasons, having an accurate estimate $\hat{Y} = \hat{f}(X)$ is preferable to evaluating $Y = f(X)$.

Accuracy of $\hat{Y} \approx Y$ depends on **reducible error** and **irreducible error**.

- **reducible error**: $f - \hat{f}$. Can be made smaller and smaller by employing increasingly sophisticated statistical learning techniques.
- **irreducible error**: ε. Present even for $f = \hat{f}$, cannot be predicted from X. Possible sources:
  - Additional variables Y may depend on but which are not observed/measured.
  - Unmeasurable variation.
    (E.g.: Adverse reaction may depend on manufacturing variations in drug or variations in patient's sensitivity over time.)
- Quantitative measure: **mean square error** (MSE)

$$\mathbf{E}\left[(Y - \hat{Y})^2\right] = \underbrace{[f(X) - \hat{f}(X)]^2}_{\textit{reducible}} + \underbrace{\mathbf{Var}\,\varepsilon}_{\textit{irreducible}}$$

- **Note:** irreducible error always lower bound on prediction accuracy.

Inference seeks to determine how the individual predictors $X_1, \ldots, X_p$ affect the response Y. In particular, this involves more detailed knowledge about $\hat{f}$ than simply considering it a black box.

Things to investigate:

- *Identify those predictors with the strongest effect on Y.*
  Can be a small subset of $X_1, \ldots, X_p$.
- *Determine relationship between response and each predictor.*
  Is it monotone increasing or decreasing with respect to an individual predictor? For more complex dependencies, such monotonicities can be affected by on the values of the remaining predictors.
- *Is a linear model sufficient?* Historically, most estimation methods have produced a linear (affine) function $\hat{f}$. If the true dependence of Y on X is more complicated, a linear model may not be accurate enough.

Learning Theory

Prediction example: direct-marketing campaign

- Company plans a direct-marketing campaign, wishes to identify individuals who would respond positively to a mailing.
- Respone is $Y \in \{\text{positive},\ \text{negative}\}$.
- Predictors X_j are demographic variables.
- Detailed relationship of response to demographic variables not of interest.
- A model which generates accurate predictions is all that is needed..

Learning Theory

Inference examples: advertising data set, purchase behavior

- In our first example (advertising by three media channels TV, radio and newspaper), one may also be interested in answers to
  - Which media increase sales?
  - Of those, which has the strongest positive effect?
  - At what rate do sales increase when the TV budget is raised?

- Another example: model brand of a product chosen by a customer as a function of predictor variables **price**, **store location**, **discount levels**, **competitor pricing** etc.
  Here detailed knowledge of how each variable affects outcome is of interest, e.g.
  - *What effect will changing the price of a product have on sales?*

- There are also **mixed situations**, involving both prediction and inference:
  Consider value of a house depending on prediction values **size**, **crime rate**, **zoning**, **distance from a river/ocean**, **air quality**, **schools**, **income level of community**, ...
  - *How much does an ocean view increase the value of a house?* (inference)
  - *Is this house over- or undervalued?* (prediction)

- **Note:** the two objectives may competing.
  Linear models allow for easier inference, but may not be accurate enough for given prediction goal.
  More sophisticated (highly nonlinear) approaches may yield high prediction accuracy, but the models they produce are often difficult to interpret.

Denote by

n : number of available data observations. ("**training data**")

x_{ij} : value of j-th predictor in i-th observation

$$i = 1, \ldots, n;\ j = 1, \ldots, p$$

y_i : value of response variable in i-th observation

Then training data consists of predictor-response pairs

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}, \qquad \boldsymbol{x}_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{ip} \end{bmatrix}.$$

Goal is estimating function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for all observations (X, Y).

Two basic approaches: **parametric** and **non-parametric**.

Two-step model-based approach

1. Assume specific functional form for f, popular example is the **linear model**

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p. \tag{2.2}$$

   Estimation of function f now consists only in determining values of the $p+1$ **parameters** $\beta_0, \beta_1, \ldots, \beta_p$. (huge simplification)

2. **Train** or **fit** the chosen model to the data, i.e., choose parameters $\{\beta_j\}_{j=0}^{p}$ in order that (here for linear model (2.2))

$$f(X) \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p.$$

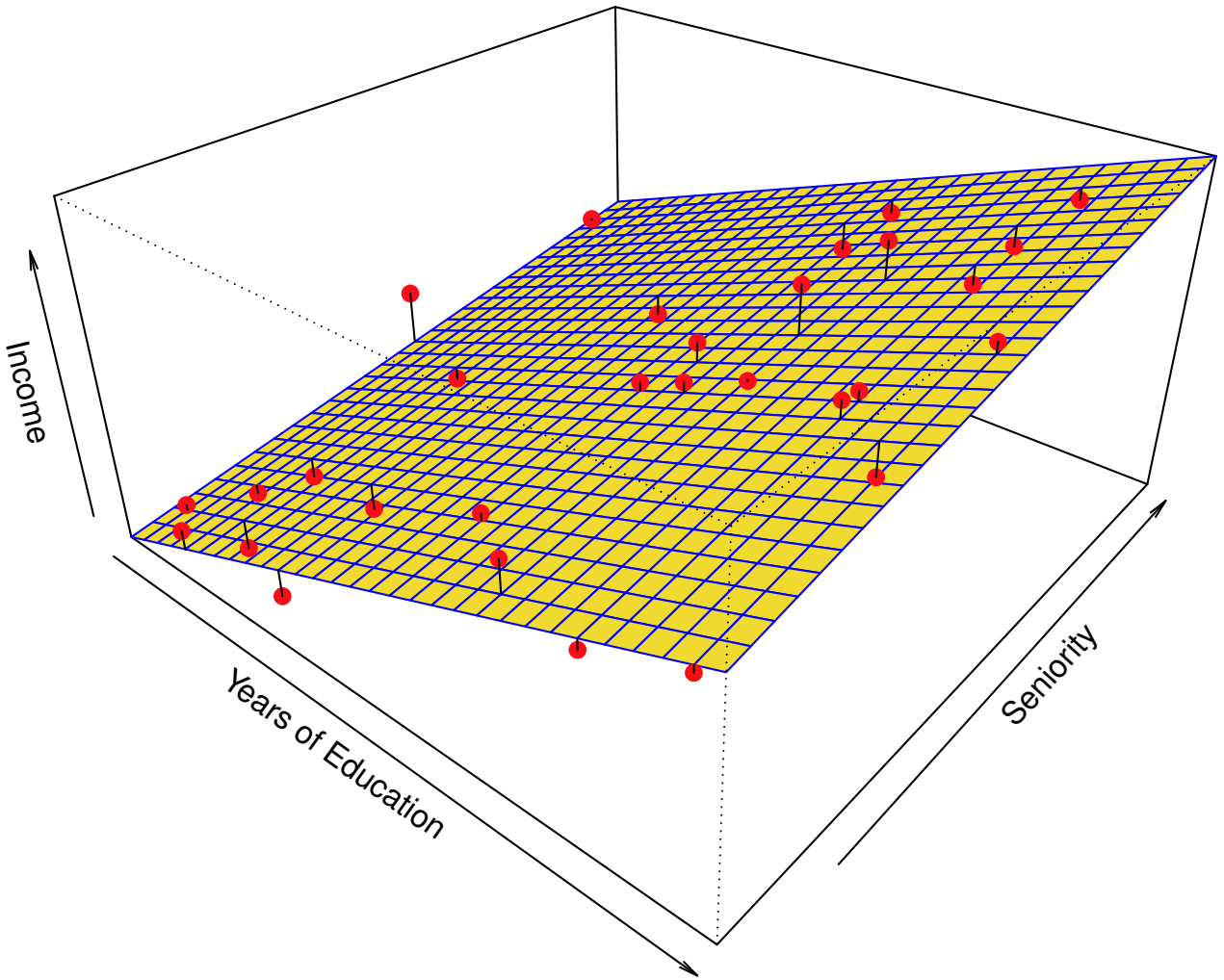
   Most common fitting technique: (ordinary) least squares, but many other techniques exist.

Problem of estimating f reduced to estimating a finite number of parameters.

Fundamental difficulty:

- Simplification comes at expense of strong restriction on type of dependence.
- For a bad choice, model cannot match the data well.
- More **flexible** models can better adapt to given data, but will generally involve more parameters to be estimated.
- Moreover, even if we are willing to fit our data extremely well with a flexible model, we may be adapting the model only to the fluctuations due to the random error contained in the data ("fitting the noise").
  In this case, our model will not **generalize** well, i.e., have a low prediction value for new data.
  This phenomenon is called **overfitting**.

Linear model: $\textbf{income} \approx \beta_0 + \beta_1 \times \textbf{education} + \beta_2 \times \textbf{seniority}$.



Learning Theory

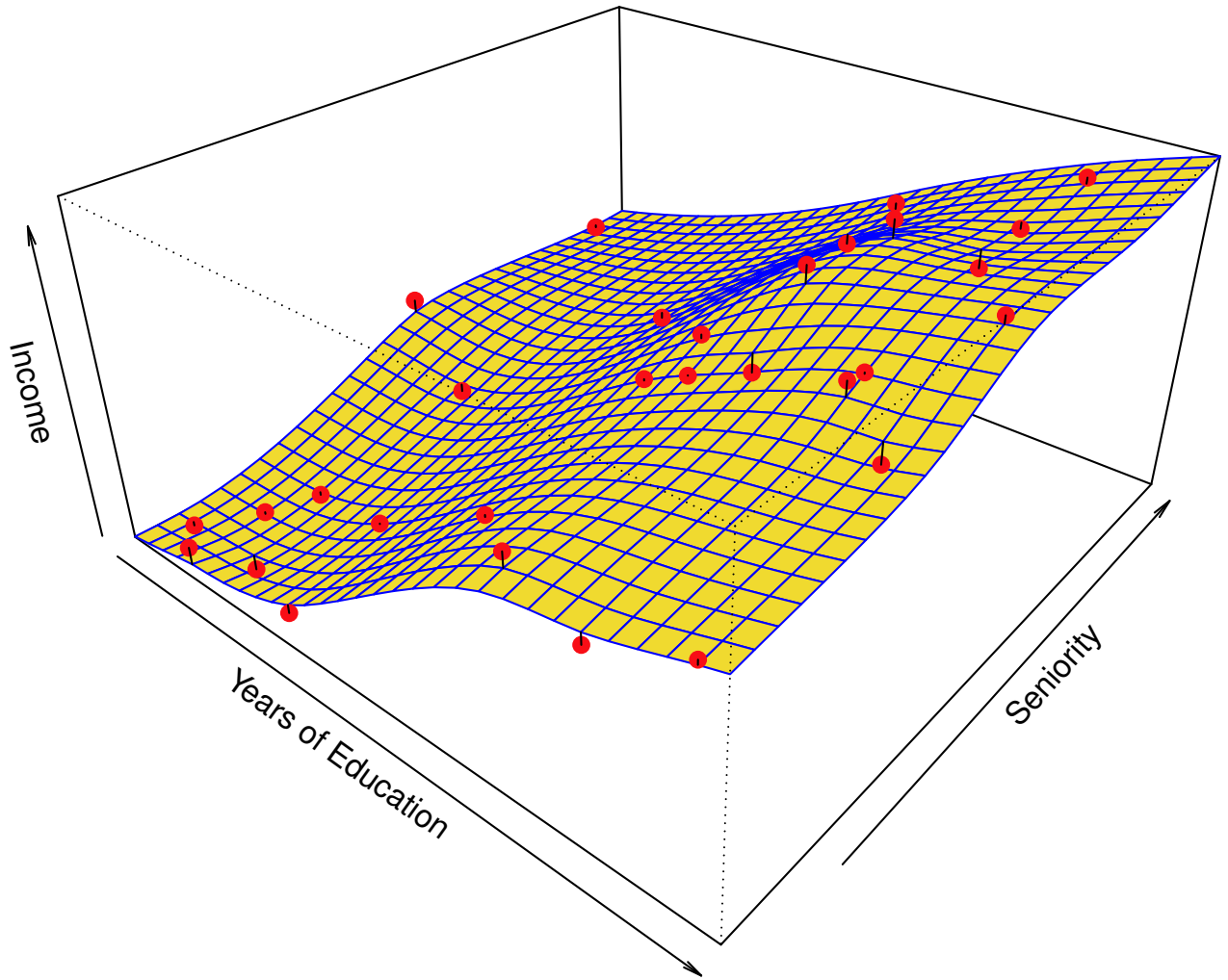
Non-parametric methods

- Non-parametric methods make no a priori assumptions on the functional form of f.
- Instead, they try to achieve as close an approximation to f as possible without being too rough or too oscillatory.

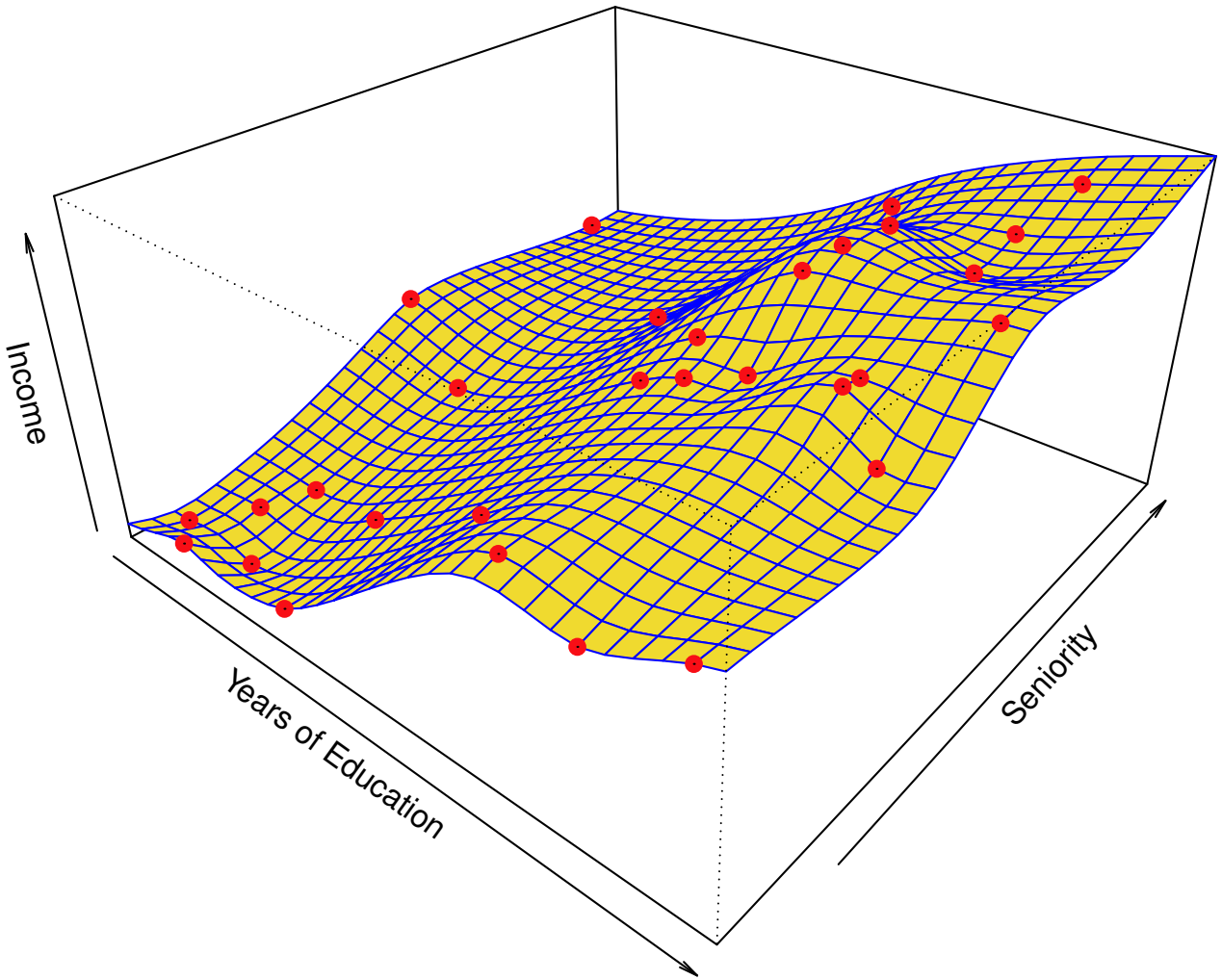
\+ Bad a priori assumption can't limit approximation accuracy.

\- Far more observations necessary than for parametric methods.

Smooth **thin-plate spline** model (later):



Rough **thin-plate spline** model:



Near perfect fit.
Are we overfitting?.

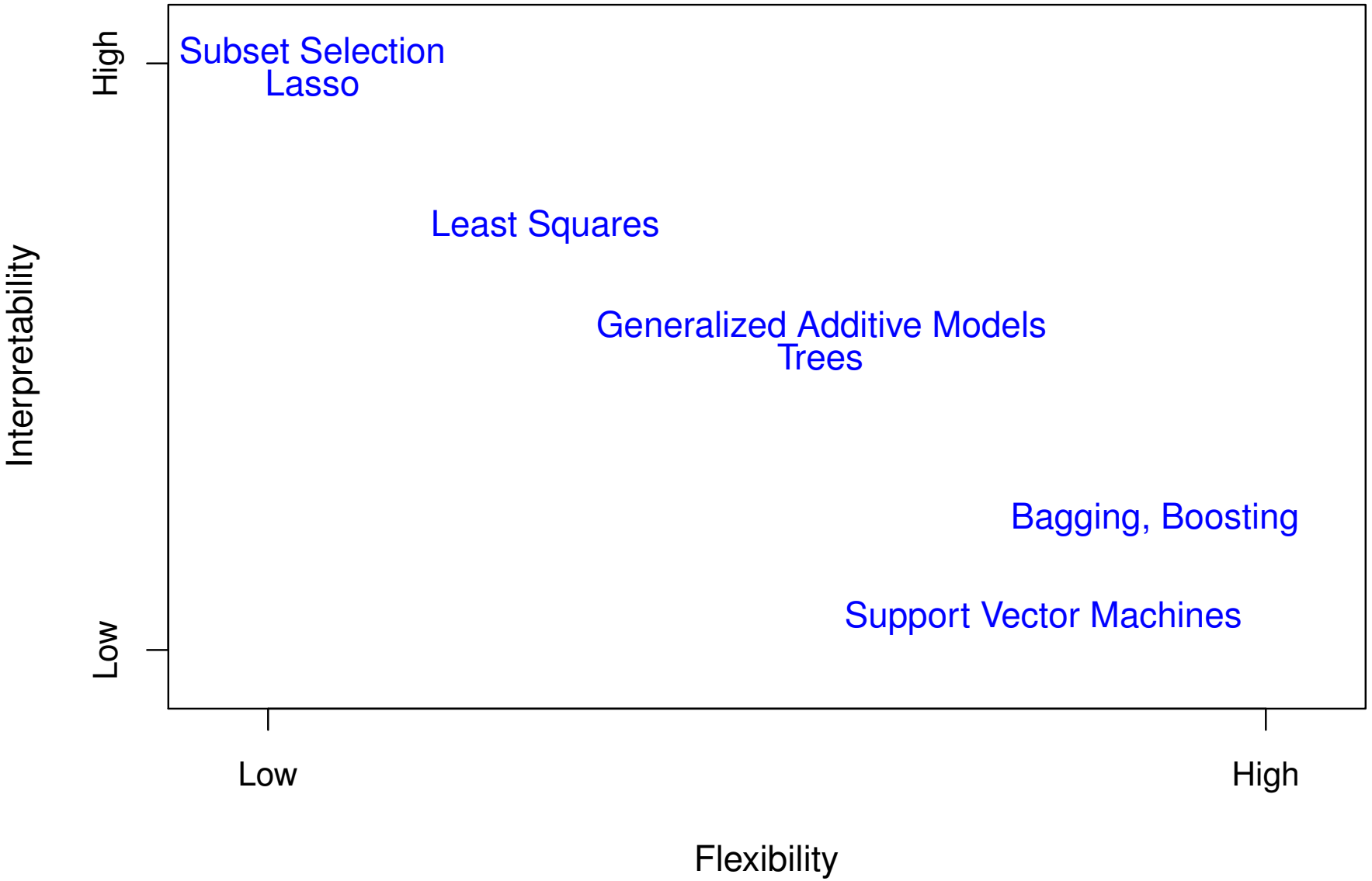
Learning Theory

Tradeoff: prediction accuracy vs. model interpretability

- Less flexible/more restrictive models can only produce a small range of shapes for f. E.g.: linear regression always provides *linear* approximation to f.
- More flexible methods (e.g. thin-plate splines) offer larger variety of function shapes.
- Advantage of restrictive methods:
  - \+ For inference, restrictive models much more interpretable.
  - Linear least-squares easy to interpret.
  - **Lasso**: linear model, different way of selecting coefficients, sets some to zero.
    More restrictive than least squares, but also more interpretable..
  - **Generalized additive models** (GAMs): extend model by certain nonlinear relationships.
    More flexible, less easy to interpret.
  - **Bagging**, **boosting**, **support-vector machines**: fully nonlinear methods, very flexible, very difficult to interpret.

Learning Theory

Tradeoff: prediction accuracy vs. model interpretability



Up to now: observation pairs $(\boldsymbol{x}_i, y_i), i = 1, \ldots, n$.

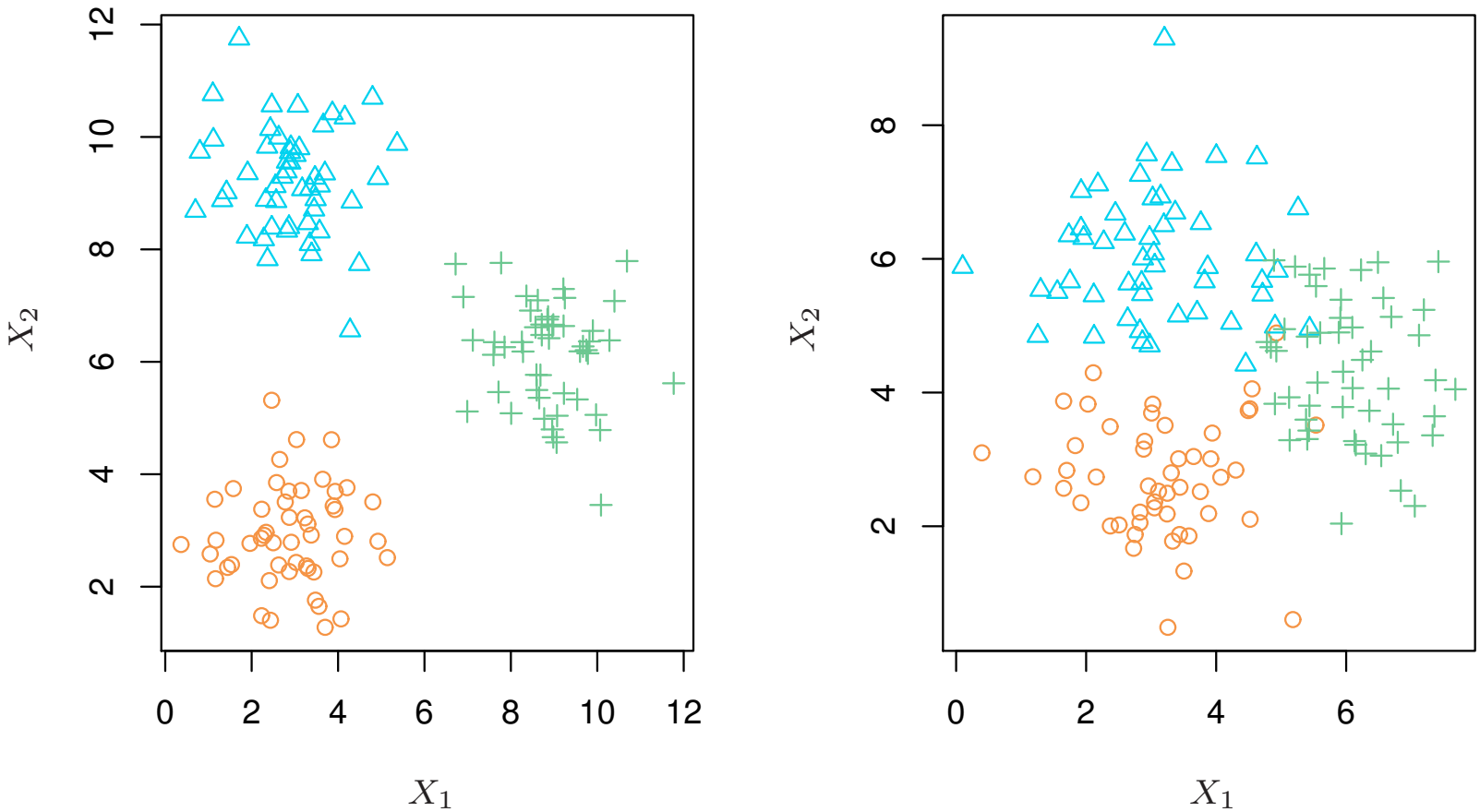
Seek model $\hat{f}$ such that $Y \approx \hat{f}(X)$ for all observations. This is called **supervised learning**.

In **unsupervised learning** only predictor variables X are observed, but no associated responses Y.

- No fitting is possible (nothing to fit to); we are, in a sense, working blind.
- Less ambitious goal: discover relationships between observations, draw conclusions for predictor variables.
- **Cluster analysis** (clustering): statistical learning tool to ascertain whether observations $\{\boldsymbol{x}_i\}_{i=1}^n$ fall into (more or less) distinct groups.
- **Example:** market segment analysis, observe multiple characteristics of potential customers (zip code, family income, shopping habits). Possible groups: big spenders, low spenders.
  In the absence of spending pattern data, clustering may reveal whether potential big spenders may be distinguished by the available data.

Learning Theory

Example: clustering



$n = 150$ observations of two variables X_1 and X_2, each belonging to one of three groups (colored for better distinction). Left: well-separated clusters, easily identified. Right: some overlap between groups, more challenging. Some observations will likely be mis-classified.

Note:

- Clustering more challenging in $p > 2$ dimensions, e.g. there are $p(p-1)/2$ possible scatterplots to look at. Automated methods needed.
- **Semi-supervised learning**: Only $m < n$ observations come with responses. (Responses could be very expensive to obtain compared to the predictor observations).
  Goal: incorporate both types of observations in an optimal way.

Another useful distinction is between **continuous** and **discrete** prediction and response variables.

- Continuous or **quantitative** variables – such as a person's height, age, income, the price of a house or stock – typically take on values in the real numbers.
- Discrete or **qualitative** variables – a person's gender, whether or not an event occurs, a cancer diagnosis – take on values in a in one of a finite number of different **classes** or categories.
- Problems with a quantitative response variable are typically referred to as **regression problems**, those with a qualitative response as **classification problems**.
- The distinction is not always sharp, e.g. logistic regression is used for (two-valued) qualitative responses (it estimates class probabilities).

Contents

Assessing Model Accuracy

Mean squared error

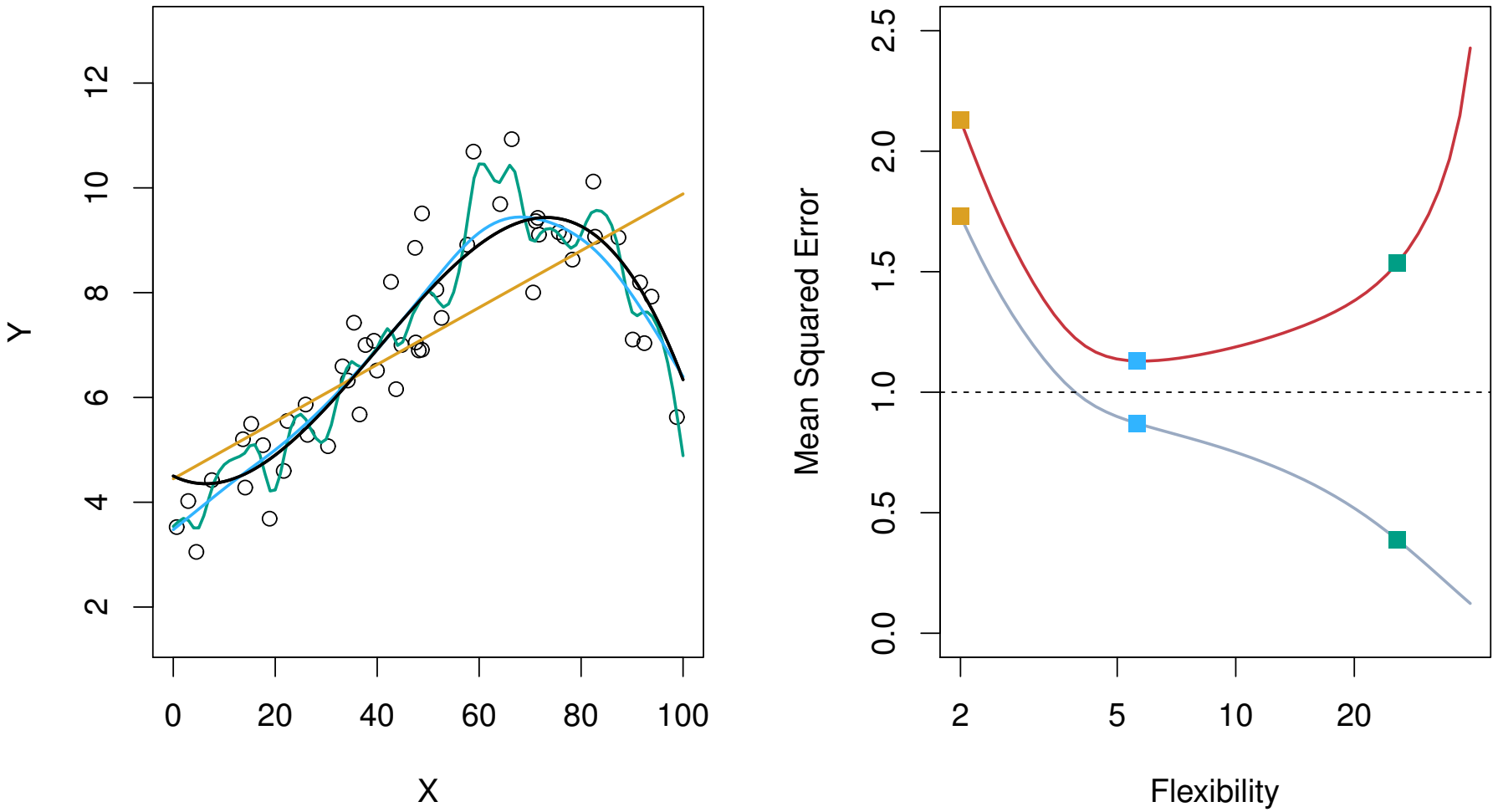
Most common error metric in regression: **mean squared error (MSE)**:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2. \tag{2.3}$$

- When applied to training data: **training MSE**.
- More interesting (particularly for prediction): **test MSE** resulting from data not used to train (fit) the model $\hat{f}$.
- If a **test data** set is available in addition to the training data, different learning (fitting) methods can be compared with respect to their test MSE values.
- In the absence of a test data set, choosing a learning method based solely on the training MSE can be deceptive.

Assessing Model Accuracy

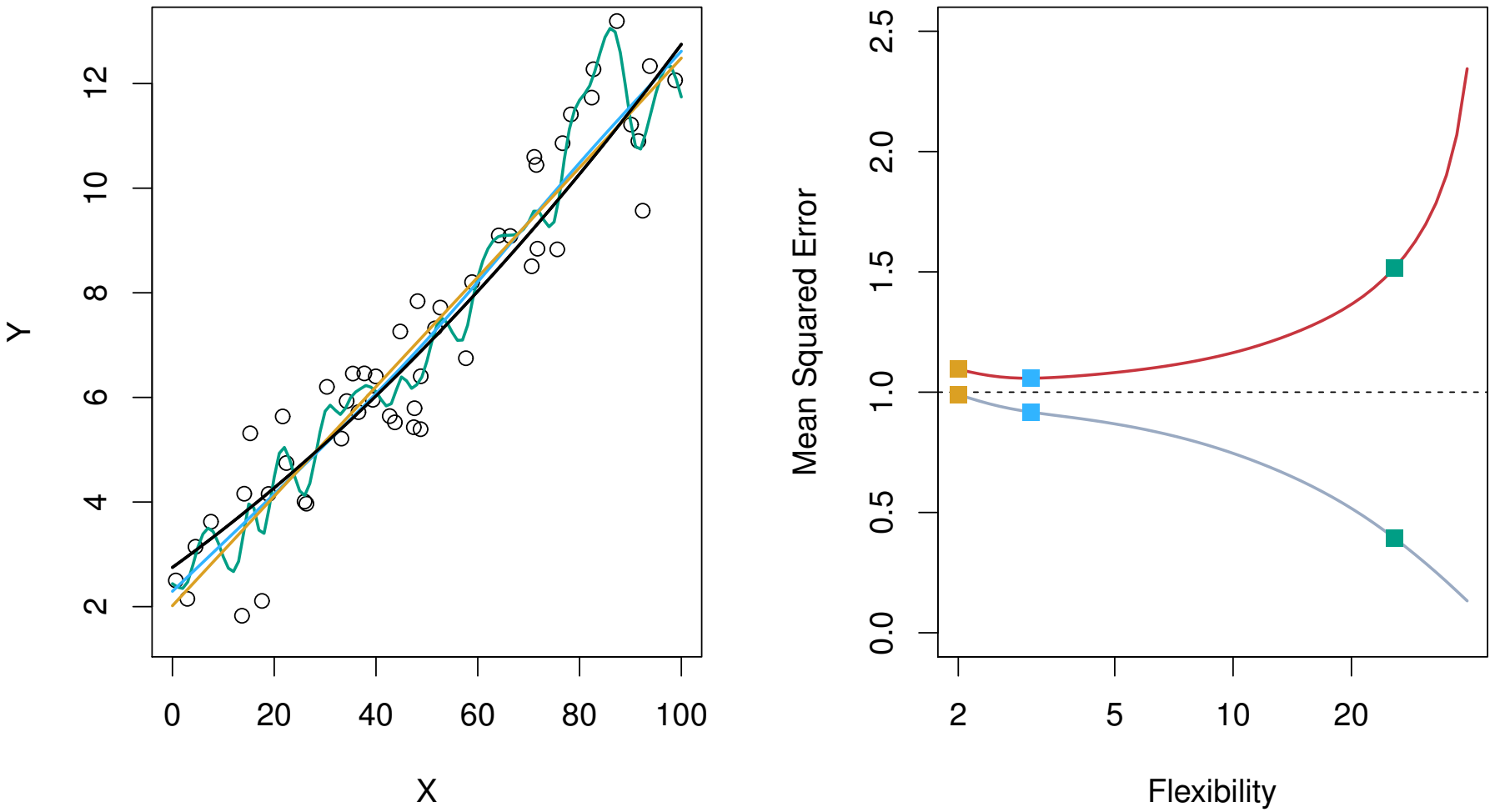
Example: smoothing spline models



Left: Observations from model (2.1), true f in black, estimates in orange, blue, green.
Right: Average MSE for training data (gray), test data (red) vs. flexibiliy parameter.
Horizontal dashed line denotes $\mathbf{Var}\,\varepsilon$.

Assessing Model Accuracy

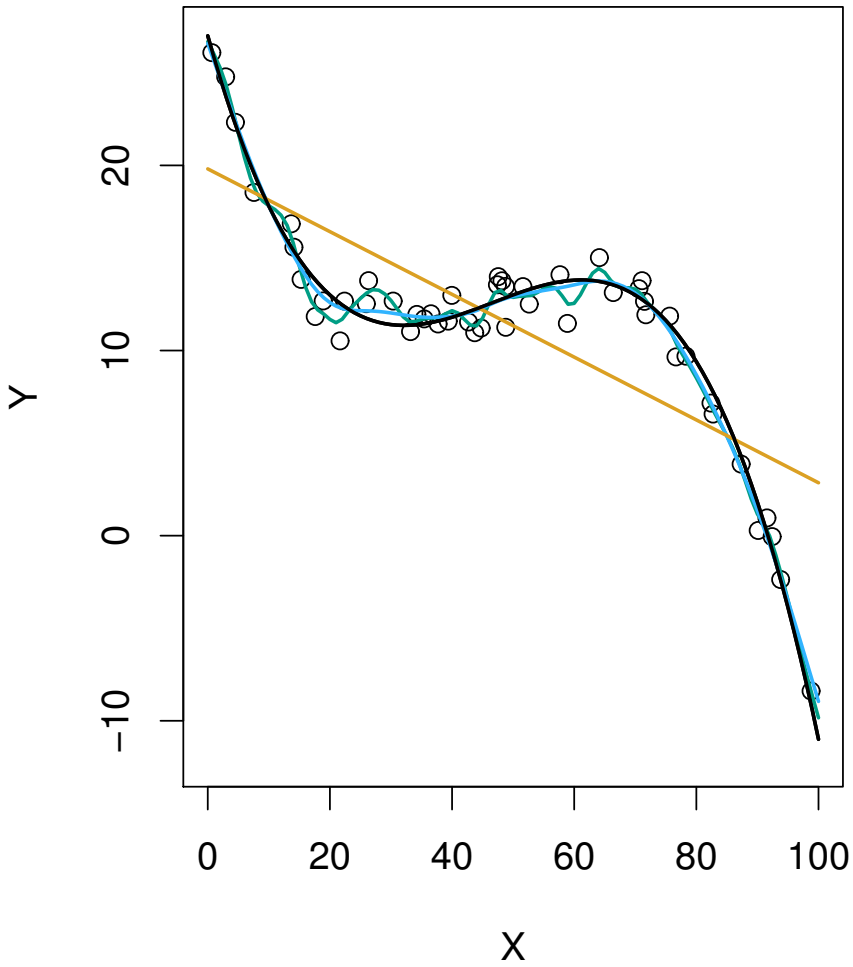
Example: smoothing spline models

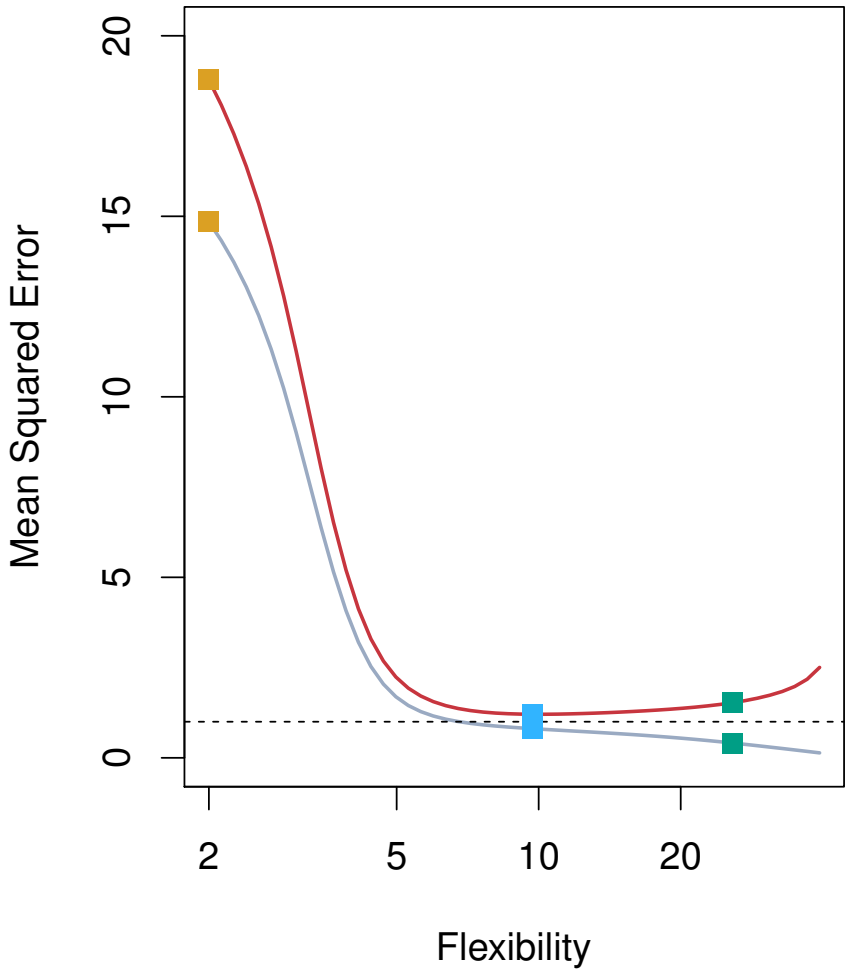


Same plots as on previous figure, but with a true model that is nearly linear. Initial estimate (with few degrees of freedon) already quite accurate.

Assessing Model Accuracy

Example: smoothing spline models





Another such plot, now the true model is highly nonlinear. Maximal accuracy for training and test data not attained until many degrees of freedom employed.

Assessing Model Accuracy

Example: smoothing spline models

Recap:

- Monotone decrease of training MSE as model becomes more flexible (more degrees of freedeom, DoF) and can more flexibly follow data variation.
- Typically test MSE curve U-shaped, rises again once overfitting sets in.
- This is a fundamental property of statistical learning, regardless of data set and regardless of statistical technique being used.
- Interpretation: in overfitting, estimate is finding patterns (signal variation) where there are none.
  James et al: "When we overfit the training data, the test MSE will be very large because the supposed patterns that the method found in the training data simply don't exist in the test data."
- Overfitting: less flexible model would have yielded smaller test MSE.

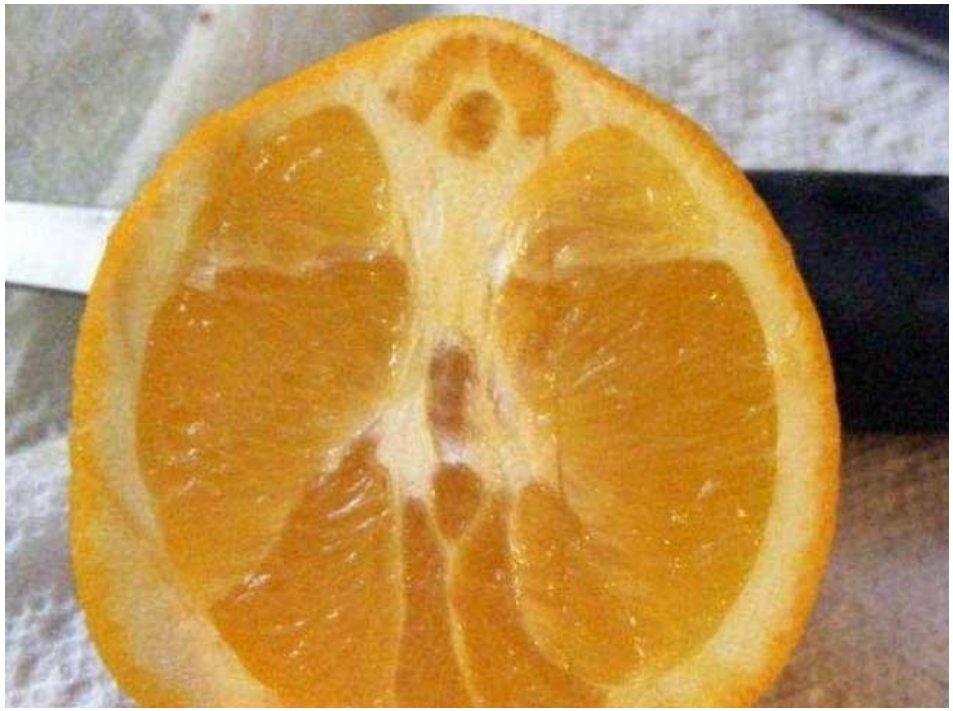
Note: Estimation methods based on minimizing MSE with respect to the DoF in the method, hence training MSE almost always less than test MSE.

Assessing Model Accuracy

Apophenia

The tendency to misclassify random events as systematic or, more generally, to see patterns where there are none, is common to human experience and known as **Apophenia**.

- It is believed to be an advantage in the process of natural selection.
- It encourages conspiracy theories.
- It is used to explain the **gambler's fallacy** in probability theory.
- In his bestselling book Thinking Fast and Low, the famous behavioral economist Kahneman calls this phenomenon the "law of small numbers".



Jesus and Mary in an orange.

Source: anorak.co.uk

Assessing Model Accuracy

Trade-off: bias vs variance

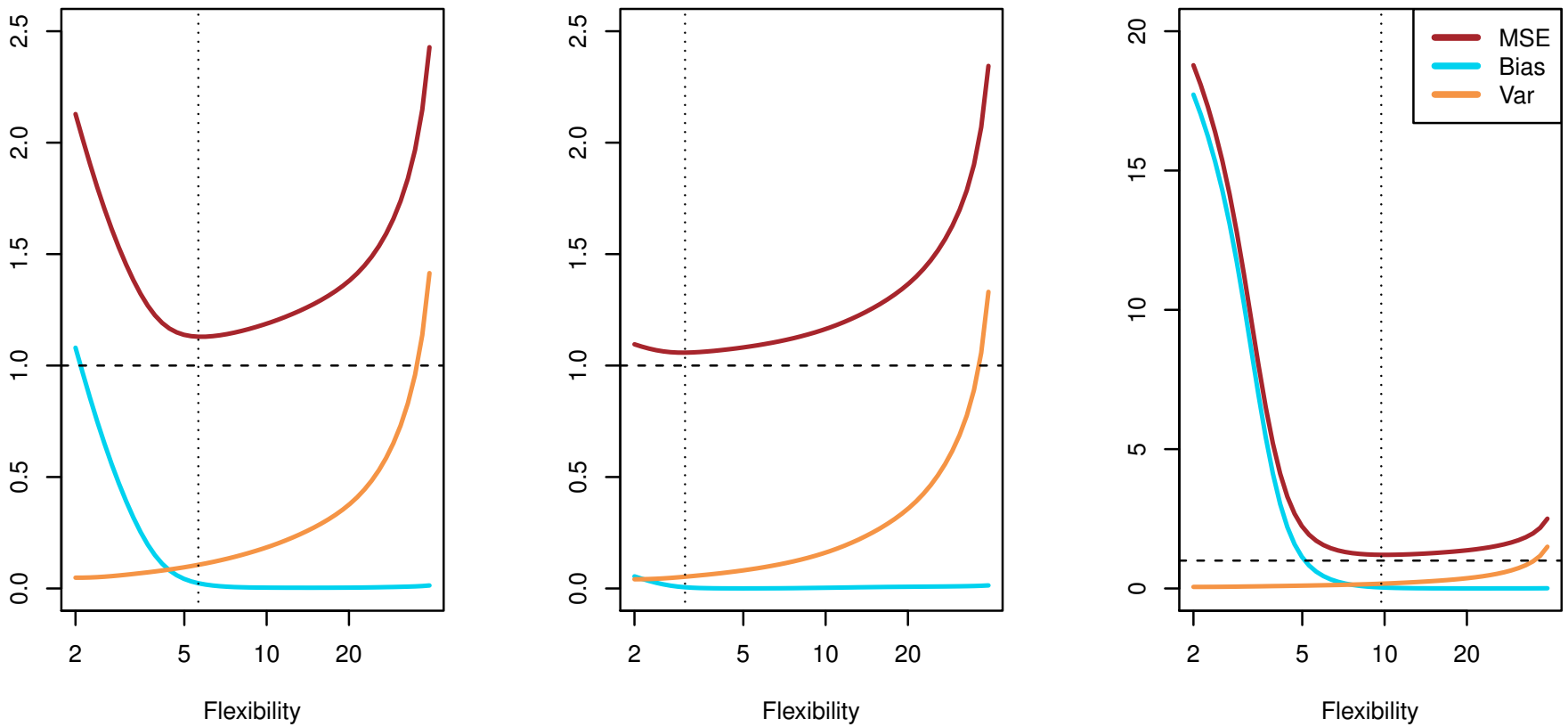
Can show: expected test MSE for new value x_0 of test data has representation

$$\mathbf{E}\left[\left(y_0 - \hat{f}(x_0)\right)^2\right] = \mathbf{Var}\, f(x_0) + [\text{Bias}\, \hat{f}(x_0)]^2 + \mathbf{Var}\, \varepsilon. \qquad (2.4)$$

- $\mathbf{E}\left[\left(y_0 - \hat{f}(x_0)\right)^2\right]$ is the expected test MSE with respect to the distribution of the predictor variable X, i.e., the average test MSE we would obtain by repeatedly estimating f using a large number of training sets, and testing each at x_0.
- (2.4) implies that a good statistical learning method needs to achieve both low bias and low variance.
- **Variance**: amount by which $\hat{f}$ would change if estimated using a different training data set. A method with high variance is sensitive to small changes in the data set.
- **Bias**: error introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model. Flexible methods have lower bias.

Assessing Model Accuracy

Trade-off: bias vs variance



Bias-variance decomposition for last 3 examples. Horizontal dashed line: $\mathbf{Var}\,\varepsilon$. Flexibility level of minimal test MSE varies due to different rates of change in bias and variance.

Assessing Model Accuracy

Bias vs. variance for classification

For qualitative (discrete) response variable Y, replace MSE with training **error rate**:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\{y_i\neq\hat{y}_i\}} \tag{2.5}$$

expressing the fraction of incorrect classifications, where

$$\hat{y}_i \text{ : predicted class label for } i\text{-th observation using } \hat{f},$$

$$\mathbb{1}_{\{y_i\neq\hat{y}_i\}} = \begin{cases} 1 & y_i \neq \hat{y}_i, \\ 0 & y_i = \hat{y}_i, \end{cases} \qquad \text{(indicator variable).}$$

As in regression setting, of more interest than training error rate (2.5) is **test error rate**, which averages classification errors $\mathbb{1}_{\{y_0\neq\hat{y}_0\}}$ over a test set of observations (x_0, y_0) with classification prediction $\hat{y}_0$ for predictor variable x_0.

Assessing Model Accuracy

Classification: Bayes classifier

One can show (we won't) that expectation of test rate error is minimized by the **Bayes classifier**: assign to test observation with predictor vector x_0 the class j for which conditional probability

$$\mathbf{P}\left(Y = j | X = x_0\right)$$

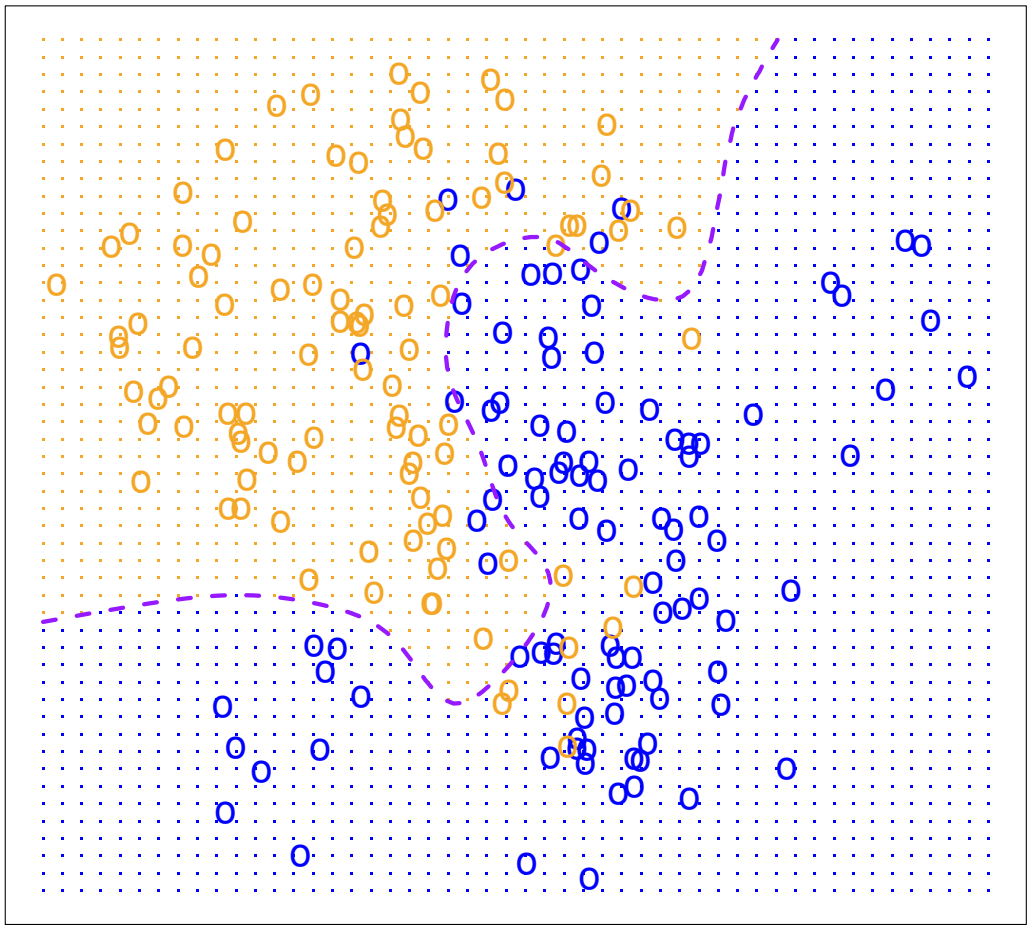
is maximized over all j.

Special case: two-class problem, i.e., $Y \in \{1, 2\}$; predict

$$\hat{y}_0 = \begin{cases} 1 & \text{if } \mathbf{P}\{Y = 1 | X = x_0\} > 0.5 \\ 2 & \text{otherwise.} \end{cases}$$

Assessing Model Accuracy

Example: Bayes classifier, 2 classes



Predictors: X_1, X_2
Response: $Y \in \{\text{orange, blue}\}$
Observations: circles

Orange shading:

$$\mathbf{P}\{Y = \text{orange}|X\} > 0.5$$

Blue shading

$$\mathbf{P}\{Y = \text{orange}|X\} < 0.5$$

(simulated data)

Dashed line: **Bayes decision boundary**

$$\mathbf{P}\{Y = \text{orange}|X\} = 0.5$$

Assessing Model Accuracy

Bayes error rate

- Bayes classifier produces lowest possible test error rate, the **Bayes error rate**.
- By definition, error rate at $X = x_0$ is

$$1 - \max_j \mathbf{P}\{Y = j|X\}.$$

- Overall Bayes error rate:

$$1 - \mathbf{E}\left[1 - \max_j \mathbf{P}\{Y = j|X\}\right],$$

  expectation with respect to distribution of X.
- Previous example: Bayes error rate is 0.1304, positive since some observations on wrong side of decision boundary, hence $\max_j \mathbf{P}\{Y = j|X = x_0\} < 1$ for some x_0.
- Bayes error rate analogous to irreducible error.

Assessing Model Accuracy

K-nearest neighbors

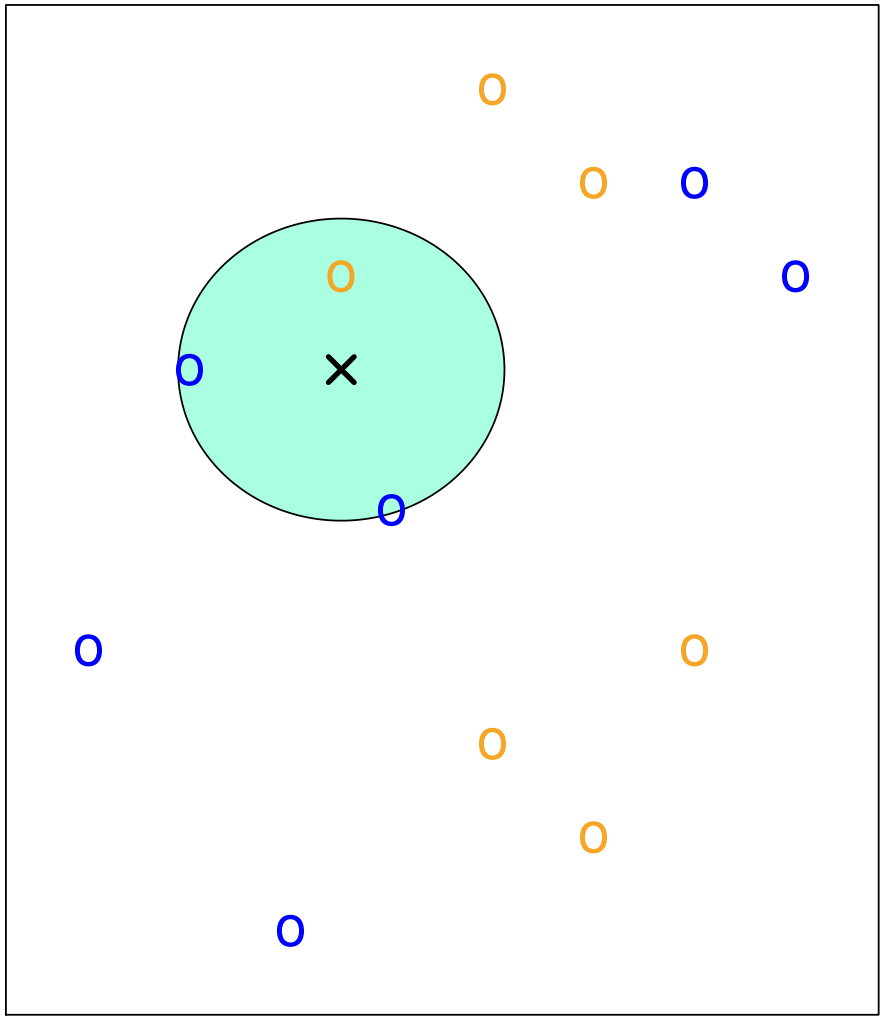
- Bayes classifier not realizable, since based on unknown conditional distribution, represents unattainable reference value.
- **K-nearest neighbors (KNN)** classifier: classify based on *estimate* of conditional distribution.
- Given $X = x_0$, denote by $\mathcal{N}_0$ the $K \in \mathbb{N}$ training set points closest to x_0 and estimate

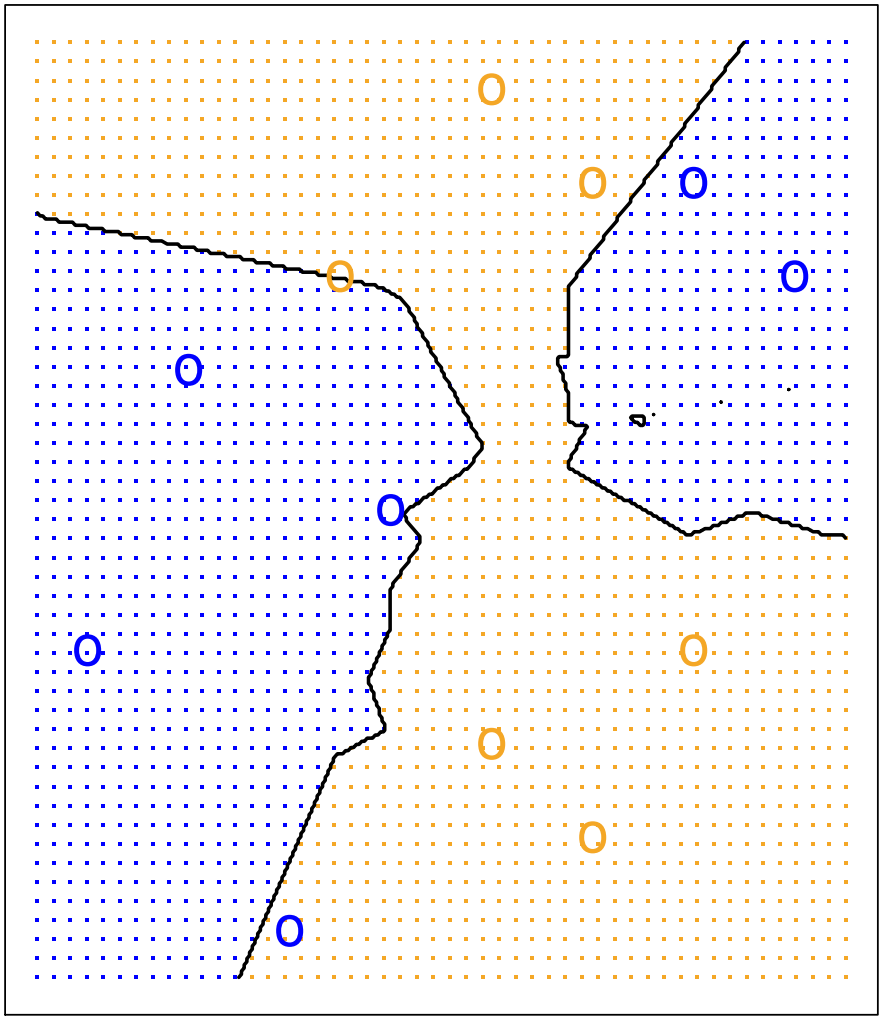
$$\mathbf{P}\{Y = j | X = x_0\} \approx \frac{1}{K} \sum_{i \in \mathcal{N}_0} \mathbb{1}_{\{y_i = j\}},$$

  i.e., by fraction of K nearest neighbors belonging to class j.
- Now proceed as in Bayes estimate with this approximate conditional distribution.

Assessing Model Accuracy

Example: KNN, K=3



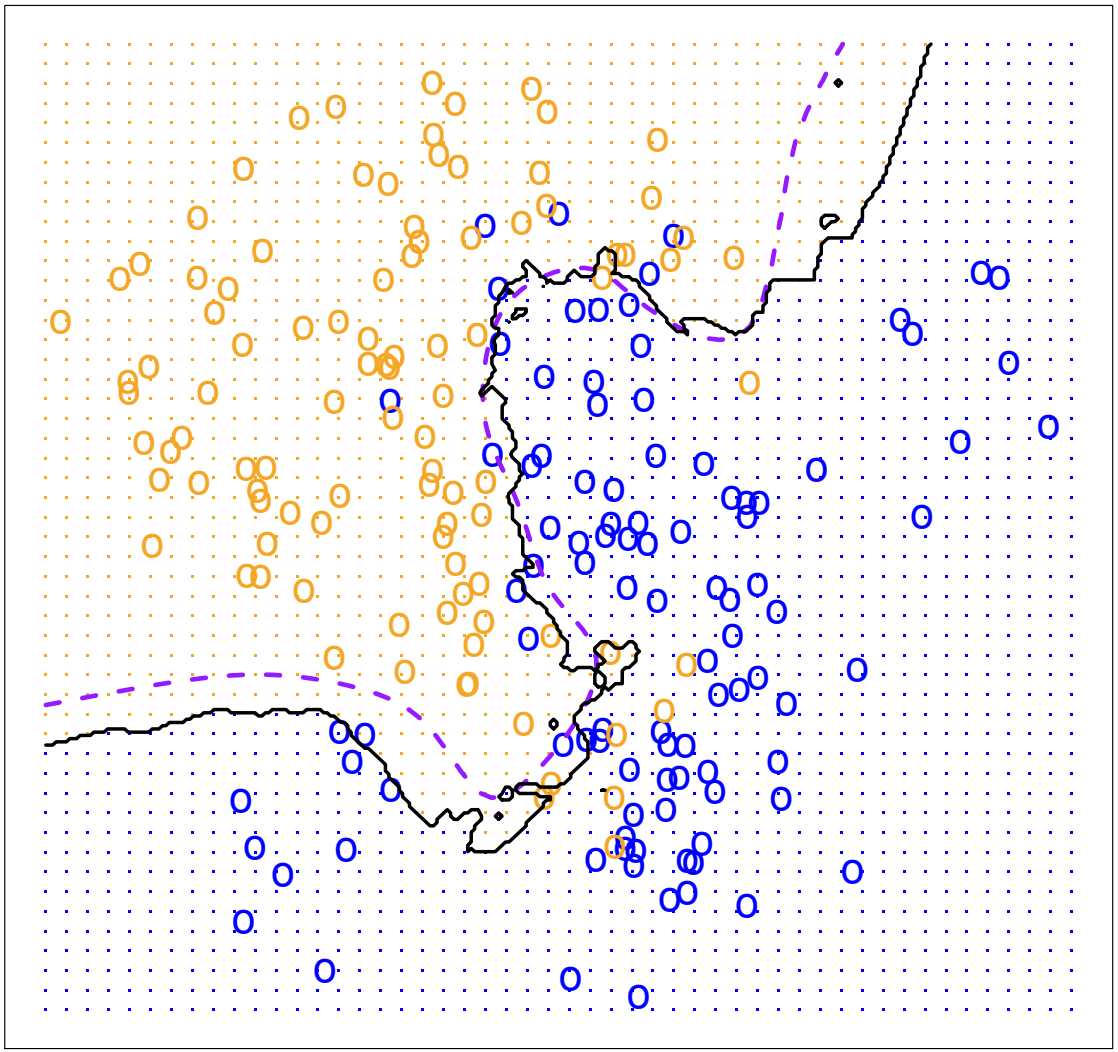


Left: $\times$: x_0; green circle: $\mathcal{N}_0$

Right: KNN applied to all shaded points, resulting decision boundary

Assessing Model Accuracy

Example: KNN, K=10



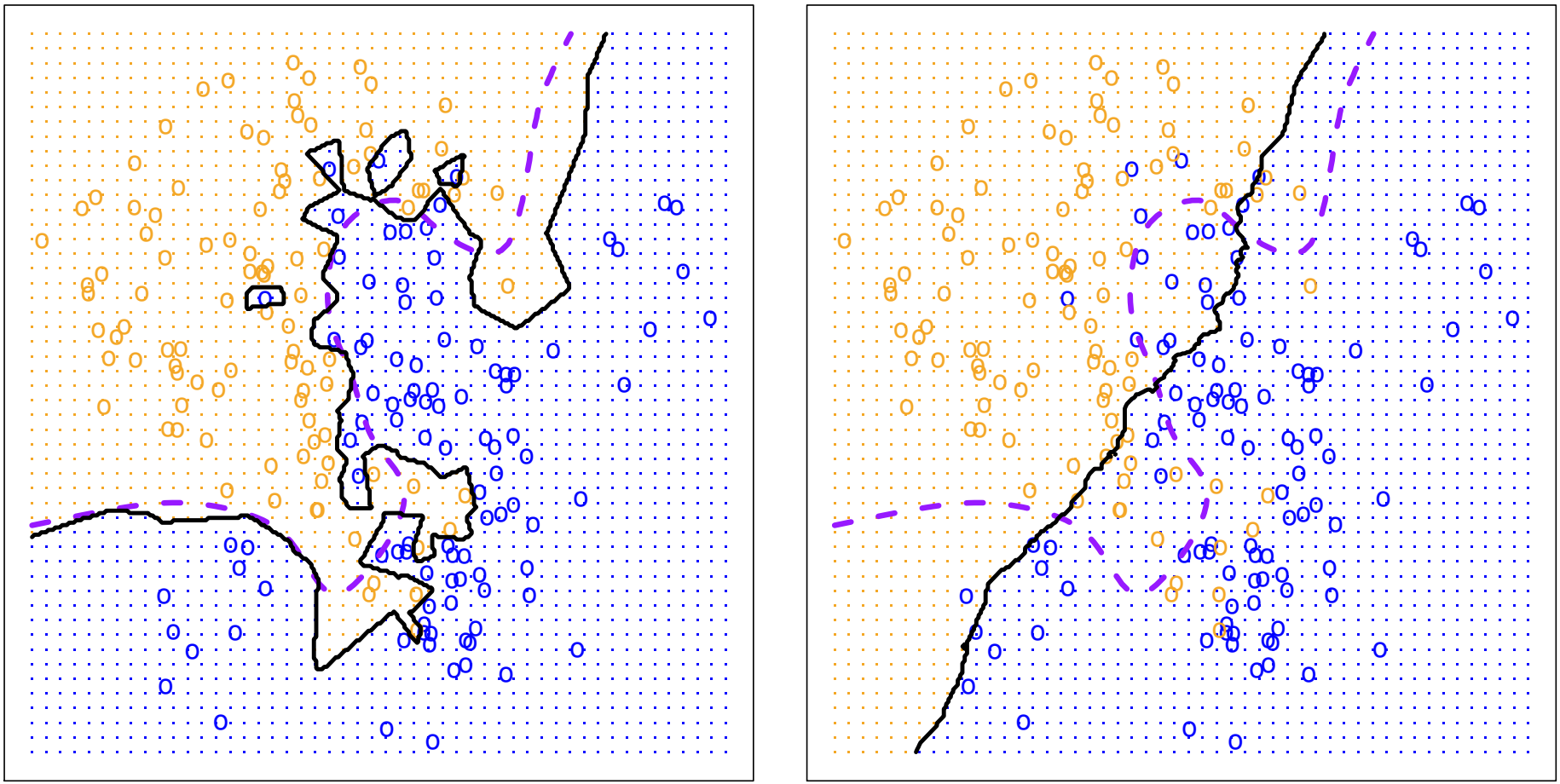
KNN decision boundary for $K = 10$ applied to data set from Slide 62.

Test error rates:
Bayes: 0.1304
KNN: 0.1363.

Assessing Model Accuracy

Example: KNN, K=1,100

KNN applied to data set from Slide 62:

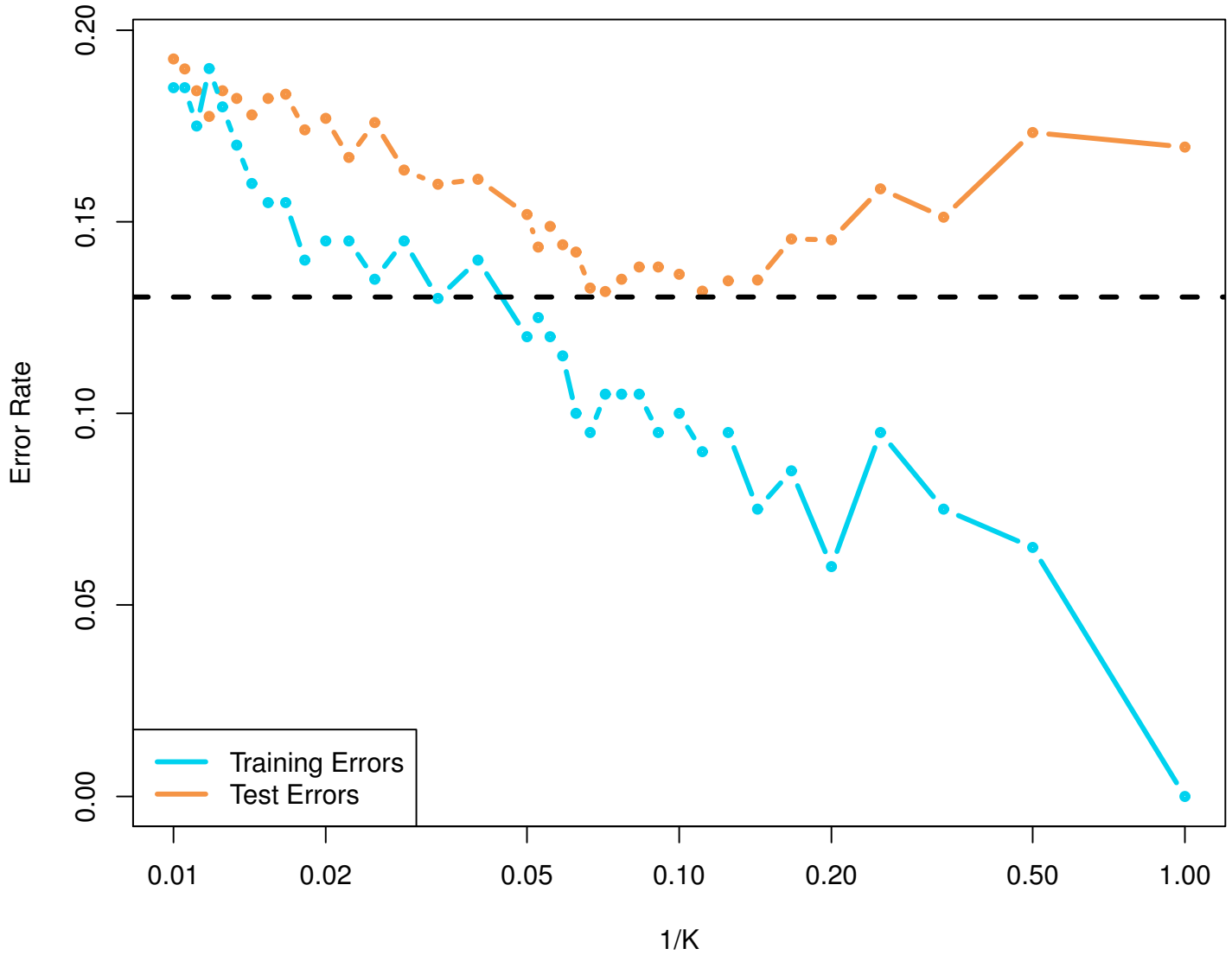


Left: $K = 1$, high variance; test error rate 0.1695.

Right: $K = 100$, high bias; test error rate 01925.

Assessing Model Accuracy

Example: KNN error rates against K



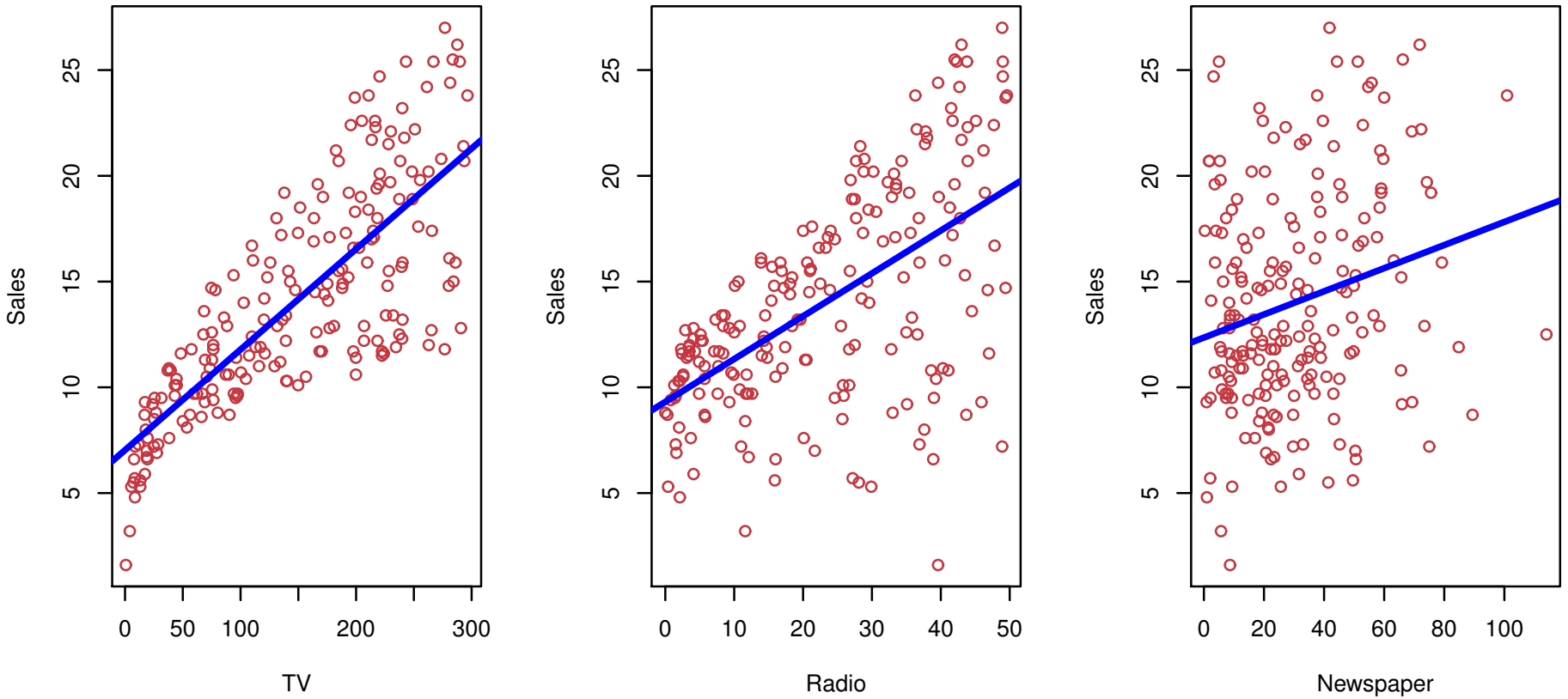
Training and test errors of KNN classification for same data plotted against $1/K$.

Contents

Linear Regression

Advertising again

Recall advertising data set from Slide 28:



We will use the simple and well-established statistical learning technique known as **linear regression** to answer the following questions:

Linear Regression

Questions about advertising data set

1. Is there a relationship between advertising budget and sales?
   Otherwise, why bother?
2. How strong is this relationship between advertising budget and sales?
   Prediction possibly better than random guess?
3. Which media contribute to sales?
   Separate individual contributions
4. How accurately can we estimate the effect of each medium on sales?
   Euro by Euro?
5. How accurately can we predict future sales?
   Precise prediction for each medium?
6. Is the relationship linear?
   If yes, linear regression appropriate (possibly after transforming data)
7. Is there synergy among the advertising media?
   Called **interaction effect** *in statistics.*

Contents

Simple Linear Regression

Definition, terminology, notation

Linear model for quantitative response Y of single predictor X:

$$Y \approx \beta_0 + \beta_1 X. \tag{3.1}$$

Statistician: "We are regressing Y onto X."

E.g., with predictor **TV** advertising and response **sales**,

$$\textbf{sales} \approx \beta_0 + \beta_1 \times \textbf{TV}.$$

The values of **coefficients** or **parameters** β_0, β_1 obtained from fitting to the training data are denoted by $\hat{\beta}_0, \hat{\beta}_1$, leading to the prediction values

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \tag{3.2}$$

when $X = x$, where the hat on $\hat{y}$ denotes the predicted value of the reponse.

Determining **intercept** $\hat{\beta}_0$ and **slope** $\hat{\beta}_1$ in (3.1) amounts to choosing these parameters such that the **residuals** or **data misfits**

$$r_i := y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i), \qquad i = 1, \ldots, n,$$

are minimized.

There are many options for defining smallness here, in **least squares estimation** this is measured by the **residual sum of squares (RSS)**

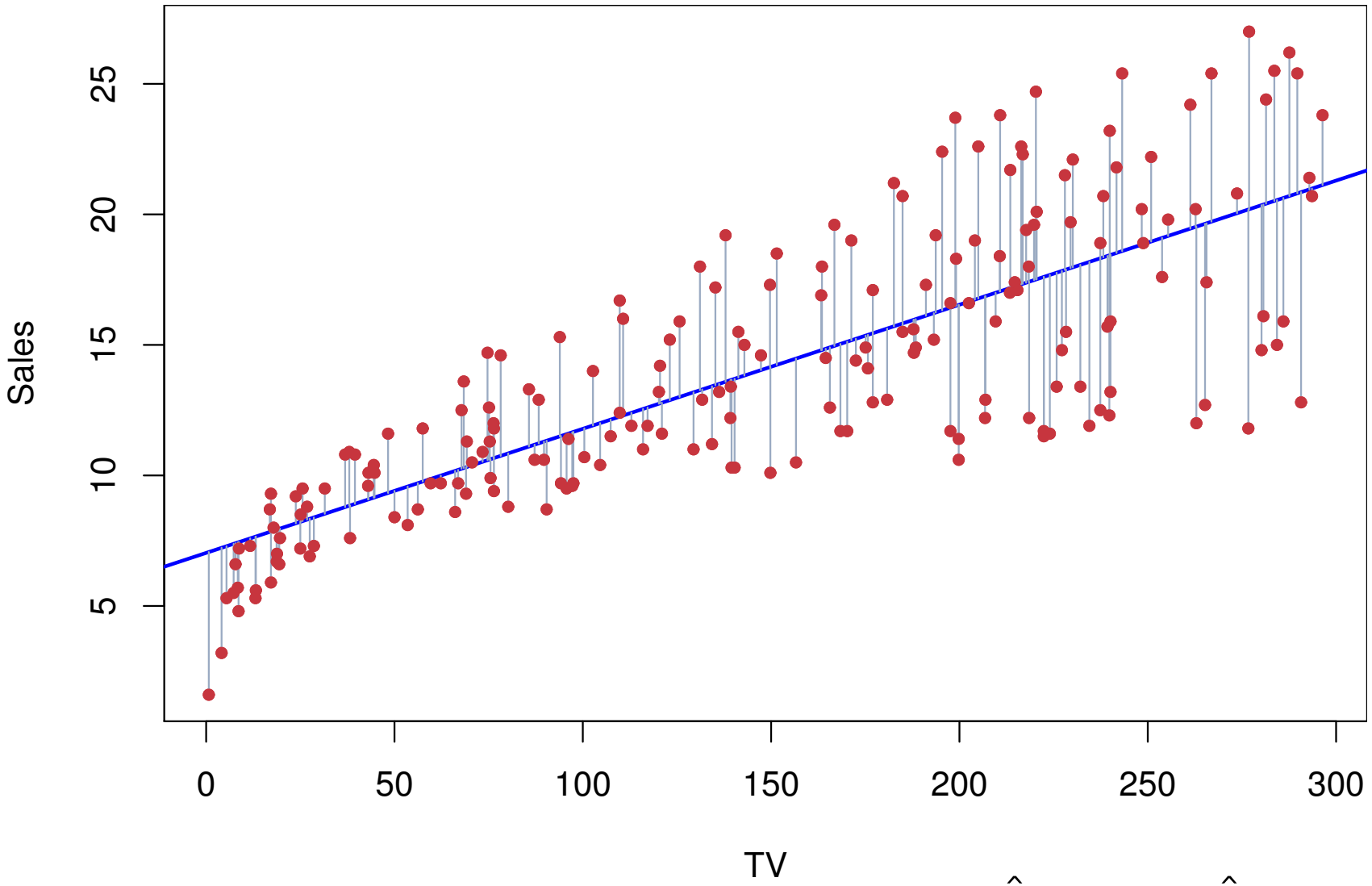
$$\text{RSS} := r_1^2 + \cdots + r_n^2 = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + \cdots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2. \tag{3.3}$$

An easy calculation reveals

$$\begin{aligned} \hat{\beta}_0 &= \overline{y} - \hat{\beta}_1 \overline{x}, & \overline{x} &:= \frac{1}{n}\sum_{i=1}^n x_i, \\ \hat{\beta}_1 &= \frac{\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^n (x_i - \overline{x})^2}, & \overline{y} &:= \frac{1}{n}\sum_{i=1}^n y_i. \end{aligned} \tag{3.4}$$

Simple Linear Regression

Example: LS fit for advertising data

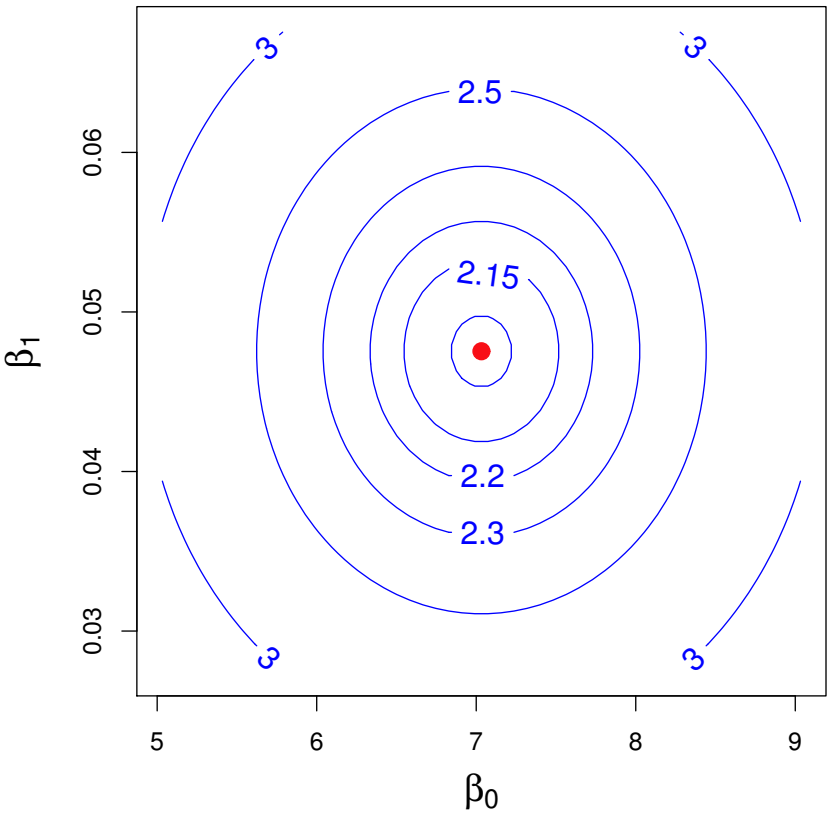


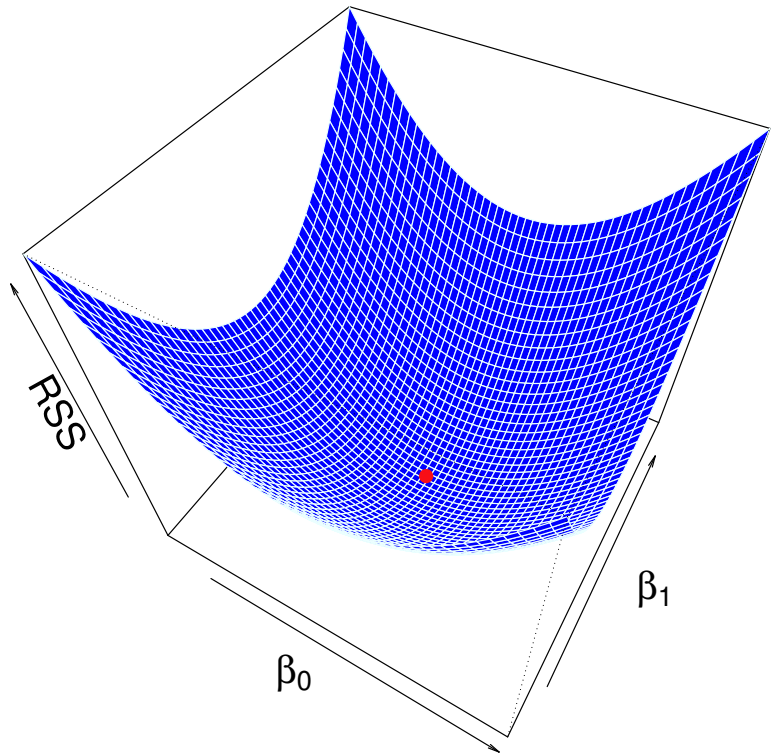
$\hat{\beta}_0 = 7.03, \hat{\beta}_1 = 0.0475$

Simple Linear Regression

Example: LS fit for advertising data

LS fit of sales vs. TV budget: RSS as a function of (β_0, β_1)





Left: Level curves. Right: Surface plot.

Simple Linear Regression

Assessing the accuracy of the coefficient estimates

Linear regression yields a linear model

$$Y = \beta_0 + \beta_1 X + \varepsilon \tag{3.5}$$

where β_0 : intercept

β_1 : slope

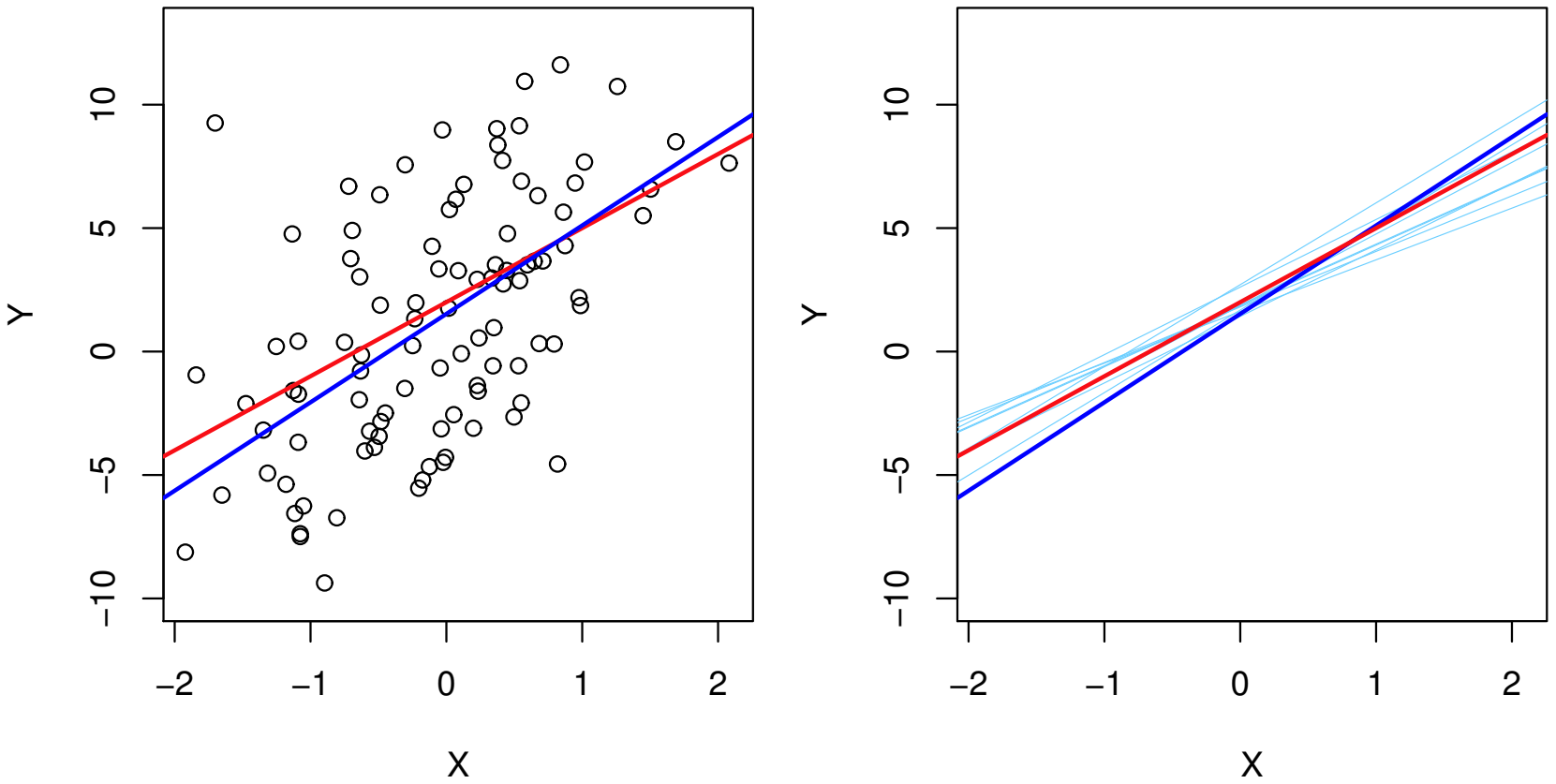
ε : model error, modeled as centered random variable, independent of X.

Model (3.5) defines the **population regression line**, the best linear approximation to the true (generally unknown) relationship between X and Y.

The linear relation (3.2) containing the coefficients $\hat{\beta}_0, \hat{\beta}_1$ estimated from a given data set is called the **least squares line**.

Simple Linear Regression

Example: population regression line, least squares line



- Left: Simulated data set ($n = 100$) from model $f(X) = 2 + 3X$.
  Red line: population regression line (true model).
  Blue line: least squares line from data (black dots).
- Right: Additionally ten (light blue) least squares lines obtained from ten separate randomly generated data sets from same model; seen to average to the red line.

Simple Linear Regression

Analogy: estimation of mean

- Standard statistical approach: use information contained in a sample to estimate characteristics of a large (possibly infinite) population.
- Example: approximate **population mean** μ (expectation, expected value) of random variable Y from observations $y_1, \ldots, y_n$ by **sample mean** $\hat{\mu} := \overline{y} := \frac{1}{n} \sum_{i=1}^{n} y_i$.
- Just like $\hat{\mu} \approx \mu$ but, in general, $\hat{\mu} \neq \mu$, the coefficients $\hat{\beta}_0, \hat{\beta}_1$ defining the least squares line are estimates of the true values β_0, β_1 of the model.
- Sample mean $\hat{\mu}$ is an **unbiased estimator** of μ, i.e., it does not *systematically* over- or underestimate the true value μ.
  Same holds for estimators $\hat{\beta}_0, \hat{\beta}_1$.
- How accurate is $\hat{\mu} \approx \mu$?
  Standard error[4] of $\hat{\mu}$, denoted $\mathrm{SE}(\hat{\mu})$, satisfies

$$\mathbf{Var}\,\hat{\mu} = \mathrm{SE}(\hat{\mu})^2 = \frac{\sigma^2}{n}, \qquad \text{where } \sigma^2 = \mathbf{Var}\,Y. \tag{3.6}$$

[4] Standard deviation of the sample distribution, i.e., average amount $\hat{\mu}$ differs from μ.

Simple Linear Regression

Standard error of regression coefficients

For the regression coefficients (assuming uncorrelated observation errors)

$$\begin{aligned} \mathrm{SE}(\hat{\beta}_0)^2 &= \sigma^2 \left[\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^n (x_i - \overline{x})^2} \right], \\ \mathrm{SE}(\hat{\beta}_1)^2 &= \frac{\sigma^2}{\sum_{i=1}^n (x_i - \overline{x})^2}, \qquad\qquad \sigma^2 = \mathbf{Var}\,\varepsilon. \end{aligned} \tag{3.7}$$

- $\mathrm{SE}(\hat{\beta}_1)$ smaller when x_i more spread out (provides more leverage to estimate slope).
- $\mathrm{SE}(\hat{\beta}_0) = \mathrm{SE}(\hat{\mu})$ if $\overline{x} = 0$. (Then $\hat{\beta}_0 = \overline{y}$.)
- σ generally unknown, can be estimated from the data by **residual standard error**

$$\mathrm{RSE} := \sqrt{\frac{\mathrm{RSS}}{n-2}}.$$

When RSE used in place of σ, should write $\widehat{\mathrm{SE}}(\hat{\beta}_1)$.

- 95% **confidence interval**: range of values containing true unknown value of parameter with probability 95%.
- For linear regression: 95% CI for β_1 approximately

$$\hat{\beta}_1 \pm 2 \cdot \mathsf{SE}(\hat{\beta}_1), \tag{3.8}$$

  i.e., with probability 95%,

$$\beta_1 \in [\hat{\beta}_1 - 2 \cdot \mathsf{SE}(\hat{\beta}_1), \hat{\beta}_1 + 2 \cdot \mathsf{SE}(\hat{\beta}_1)]. \tag{3.9}$$

- Similarly, for β_0, 95% CI approximately given by

$$\hat{\beta}_0 \pm 2 \cdot \mathsf{SE}(\hat{\beta}_0). \tag{3.10}$$

- For advertising example: with 95% probability

$$\beta_0 \in [6.130, 7.935], \qquad \beta_1 \in [0.042, 0.053].$$

Use SE to test **null hypothesis**

$$H_0 : \text{no relationship between } X \text{ and } Y \tag{3.11}$$

and **alternative hypothesis**

$$H_a : \text{some relationship between } X \text{ and } Y \tag{3.12}$$

or, mathematically,

$$H_0 : \beta_1 = 0 \quad \text{vs.} \quad H_a : \beta_1 \neq 0.$$

- Reject H_0 if $\hat{\beta}_1$ sufficiently far from 0 relative to $\mathsf{SE}(\hat{\beta}_1)$.
- **t-statistic**

$$t = \frac{\hat{\beta}_1 - 0}{\mathsf{SE}(\hat{\beta}_1)} \tag{3.13}$$

measures distance of $\hat{\beta}_1$ from 0 in # standard deviations.

Simple Linear Regression

Hypothesis tests

- $\beta_1 = 0$ implies t follows t-distribution with $n - 2$ degrees of freedom.
- We compute probability of observing $|t|$ or larger under assumption $\beta_1 = 0$, its *p*-**value**.
- Small p-value: unlikely to observe substantial relation between X and Y due to purely random variation, unless the two actually are related.
- In this case we reject H_0.
- Typical cutoffs for p-value: 1%, 5%; for $n = 30$ corresponds to t-statistic (3.13) values 2 and 2.75. respectively.

For **TV** sales data in advertising data set:

	Estimate	SE	t-statistic	p-value
β_0	7.0325	0.4578	15.36	< 0.0001
β_1	0.0475	0.0027	17.67	< 0.0001

Simple Linear Regression

Reminder: Student's t distribution

- Given $X_1, \cdots, X_n$ i.i.d. $\sim N(\mu, \sigma^2)$
- Sample mean:

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

- (Bessel corrected) sample variance:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \overline{X})^2$$

- RV

$$\frac{\overline{X} - \mu}{\sigma/\sqrt{n}}$$

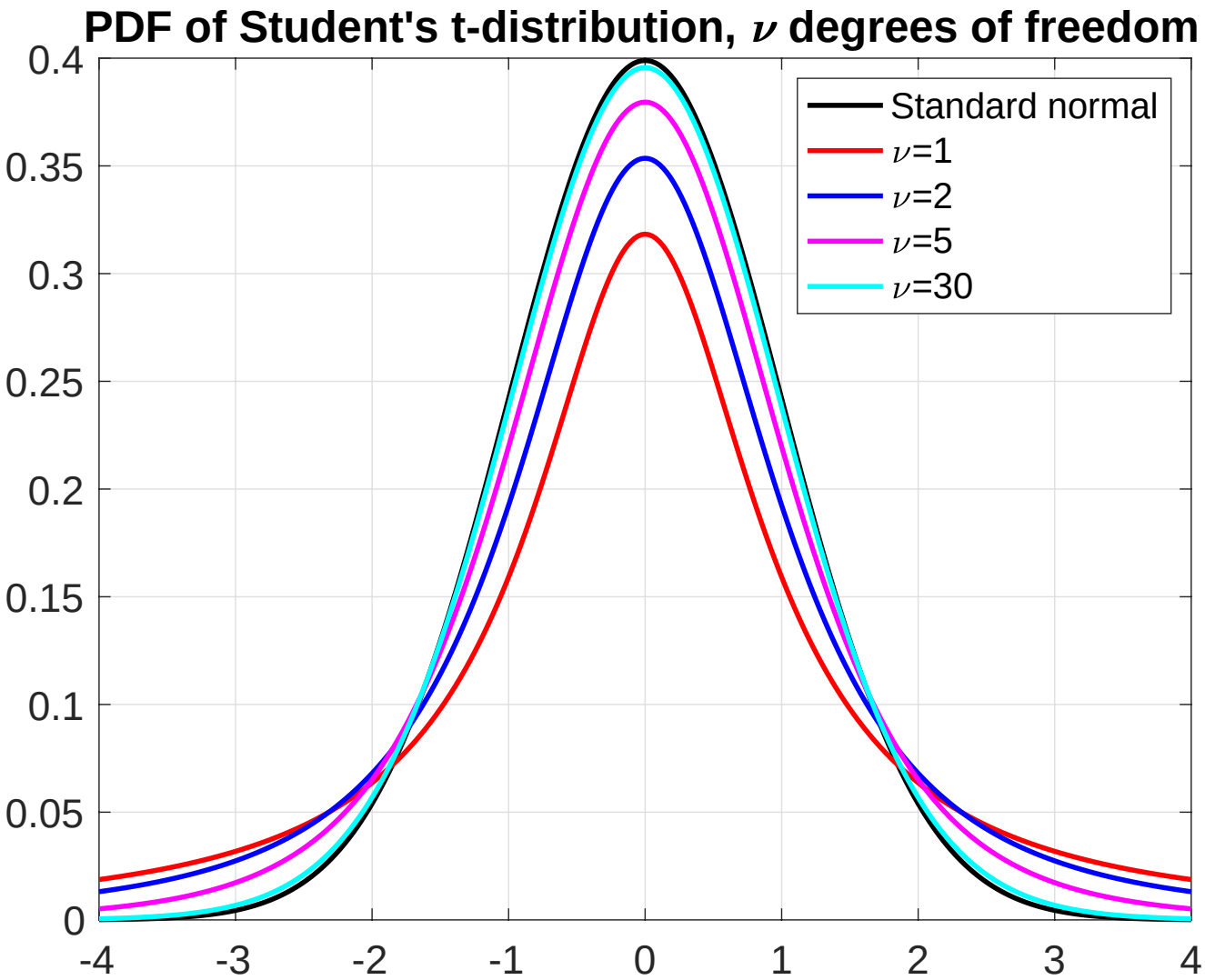
  distributed according to $N(0, 1)$.
- RV

$$\frac{\overline{X} - \mu}{S/\sqrt{n}}$$

  distributed according to Student's t-distribution with $n - 1$ DoF.

Simple Linear Regression

Student's t distribution



- **Residual standard error**: estimate of standard deviation of ε (model error)

$$\mathrm{RSE} = \sqrt{\frac{\mathrm{RSS}}{n-2}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{3.14}$$

- For **TV** data RSS $= 3.26$, i.e., deviation of sales from true regression line on average by 3,260 units (even if exact β_0, β_1 known).
  Corresponds to $3,260/14,000 = 23\%$ error relative to mean value of all sales.
- RSE measures *lack of model fit*.

- R^2 **statistic**: alternative measure of fit: proportion of variance explained.
- $\in [0, 1]$, independent of scale of Y.
- Defined in terms of **total sum of squares (TSS)** as

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}, \qquad \text{TSS} = \sum_{i=1}^{n} (y_i - \overline{y})^2. \tag{3.15}$$

- TSS : total variance in response Y,
  RSS : amount of variability left unexplained after regression,
  $\text{TSS} - \text{RSS}$: response variability explained by regression model,
  R^2 : proportion of variability in Y explained using X.
- $R^2 \approx 0$: linear model wrong, high model error variance.
- For **TV** data $R^2 = 0.61$: 2/3 of **sales** variability explained by (linear regression on) TV budget.
- $R^2 \in [0, 1]$, but sufficient value problem dependent.

- Measure of linear relationship between X and Y: **(sample) correlation**:

$$\mathsf{Cor}(X, Y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}. \tag{3.16}$$

- In simple linear regression: $\mathsf{Cor}(X, Y)^2 = R^2$.
- Correlation expresses association between single pair of variables; R^2 between larger number of variables in multivariate linear regression.

Contents

Multiple Linear Regression

Justification

- $p > 1$ predictor variables
  (as in **advertising** data set: **TV**, **newspaper**, **radio**)
- Easiest option: simple linear regression for each

For **radio** sales data in advertising data set:

	Estimate	SE	t-statistic	p-value
β_0	9.312	0.563	16.54	< 0.0001
β_1	0.203	0.020	9.92	< 0.0001

For **newspaper** sales data in advertising data set:

	Estimate	SE	t-statistic	p-value
β_0	12.351	0.621	19.88	< 0.0001
β_1	0.055	0.017	3.30	< 0.00115

Multiple Linear Regression

Justification

- How to predict total sales given 3 budgets?
- Each separate regression equation ignores the other 2 media.
- For correlated media budgets this can lead to misleading estimates of individual media effects.

Multiple linear regression model for p predictor variables:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon \tag{3.17}$$

β_j : average effect on Y of 1-unit increase in X_j holding other predictors fixed.

In advertising example:

$$\textbf{sales} = \beta_0 + \beta_1 \times \textbf{TV} + \beta_2 \times \textbf{radio} + \beta_3 \times \textbf{newspaper} \tag{3.18}$$

Multiple Linear Regression

Estimating the coefficients

- Given estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$, obtain prediction formula

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p. \tag{3.19}$$

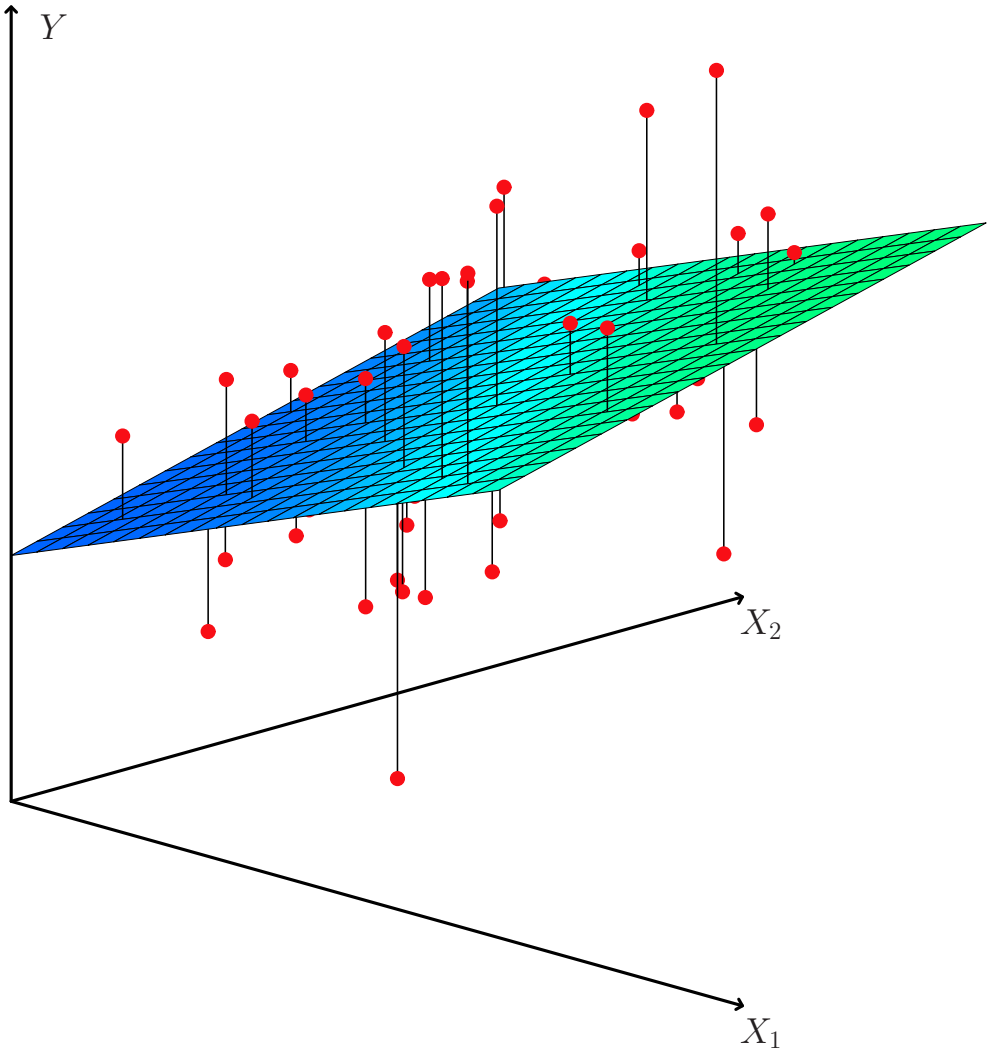
- Same fitting approach: choose $\{\hat{\beta}_j\}_{j=0}^p$ to minimize

$$\text{RSS} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i,1} - \cdots - \hat{\beta}_p x_{i,p})^2, \tag{3.20}$$

yielding the **multiple least squares regression coefficients**

Multiple Linear Regression

Example: multiple linear regression, 2 predictors, 1 response



- Determining the coefficients $\{\hat{\beta}_j\}_{j=0}^p$ to minimize the RSS in (3.20) is equivalent to minimizing $\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|_2^2$, where we have introduced the notation

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad \boldsymbol{X} = \begin{bmatrix} 1 & x_{1,1} & \dots & x_{1,p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n,1} & \dots & x_{n,p} \end{bmatrix}, \quad \widehat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\beta}_0 \\ \vdots \\ \hat{\beta}_p \end{bmatrix}$$

  for the vector $\boldsymbol{y} \in \mathbb{R}^n$ of response observations, the matrix $\boldsymbol{X} \in \mathbb{R}^{n\times(p+1)}$ of predictor observations and vector $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}$ of coefficient estimates.
- The problem of finding a vector $\boldsymbol{x} \in \mathbb{R}^n$ such that $\boldsymbol{b} \approx \boldsymbol{A}\boldsymbol{x}$ for given $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$ is called a **linear regression problem**.
- One (of many) possible approaches for achieving this is choosing $\boldsymbol{x}$ to minimize $\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2$, which is a **linear least squares problem**.

- A somewhat more general fitting approach using a model

$$y \approx \beta_0 + \beta_1 f_1(\boldsymbol{x}) + \cdots + \beta_p f_p(\boldsymbol{x})$$

  with fixed **regression functions** $\{f_j\}_{j=1}^p$ also leads to a linear regression problem, where now $[\boldsymbol{X}]_{i,j} = f_j(x_i)$.
- A linear least squares problem $\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2 \to \mathbf{min}$ with $m \geq n$ has a unique solution if the columns of $\boldsymbol{A}$ are linearly independent, i.e., when $\boldsymbol{A}$ has full rank, given by $\boldsymbol{x} = (\boldsymbol{A}^\mathsf{T}\boldsymbol{A})^{-1}\boldsymbol{A}^\mathsf{T}\boldsymbol{b}$.
  In this case the solution can be computed using a Cholesky decomposition.
- In the (nearly) rank-deficient case, more sophisticated techniques of numerical linear algebra like the QR decomposition or the SVD are required to obtain a (stable) solution.
- When $\boldsymbol{A}$ is large and sparse or structured, iterative methods such as CGLS or LSQR can be employed which require only matrix-vector products in place of manipulations of matrix entries.

Multiple Linear Regression

Advertising data

	Estimate	SE	t-statistic	p-value
β_0	2.939	0.3119	9.42	< 0.0001
β_1 (TV)	0.046	0.0014	32.81	< 0.0001
β_2 (radio)	0.189	0.0086	21.89	< 0.0001
β_3 (newspaper)	-0.001	0.0059	-0.18	0.8599

- Newspaper slope differs from simple regression.
  Small estimate, p-value no longer significant.
- Now no relation between sales and newspaper budget. Contradiction?

Multiple Linear Regression

Advertising data

Correlation matrix:

	TV	radio	newapaper	sales
TV	1.0000	0.0548	0.0567	0.7822
radio		1.0000	0.3541	0.5762
newspaper			1.0000	0.2283
sales				1.0000

- Correlation between newspaper and radio: ≈ 0.35:
  Tend to spend more on radio ads where more is spent on newspaper ads.
- If correct, i.e., $\beta_{\text{newspaper}} \approx 0$, $\beta_{\text{radio}} > 0$, radio increased sales, and where radio budget high, newpaper budget tends to also be high.
- Simple linear regression: indicates newspaper associated with higher sales.
  Multiple regression reveals no such affect.
- Newspaper receives credit for radio's affect on sales.
  Sales due to newspaper advertising is a **surrogate** for sales due to radio advertising.

Multiple Linear Regression

Absurd example, same effect

- Counterintuitive but not uncommon. Consider following (absurd) example.
- Data on **shark attacks** versus **ice cream sales** at beach community would show similar positive relationship as **newpaper** and **radio** ads.
- Should one ban ice cream sales to reduce risk of shark attacks?
- Answer: High temperatures cause both (more people at beach for shark encounters, more ice cream customers).
- Multiple regression reveals icre cream sales not a predictor for shark attacks after adjusting for temperature.

Multiple Linear Regression

Questions to consider

1. Is at least one of the predictors $X_1, X_2, \ldots, X_p$ useful in predicting the response?
2. Do all predictors help to explain Y, or is only a subset of the predictors useful?
3. How well does the model fit the data?
4. Given a set of predictor values, what response value should we predict, and how accurate is our prediction?

Multiple Linear Regression

(1) Is there a relationship between response and predictors?

- As for simple regression, perform statistical hypothesis test: null hpothesis

$$H_0: \quad \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

versus alternative

$$H_a: \quad \text{at least one } \beta_j (j = 1, \ldots, p) \text{ is nonzero.}$$

- Such a test can be based on the **F-statistic**

$$F = \frac{(\mathrm{TSS} - \mathrm{RSS})/p}{\mathrm{RSS}/(n-p-1)} \tag{3.21}$$

where, as before,

$$\mathrm{TSS} = \sum_{i=1}^{n} (y_i - \overline{y})^2, \qquad \mathrm{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

Multiple Linear Regression

(1) Is there a relationship between response and predictors?

$$F = \frac{(\mathrm{TSS} - \mathrm{RSS})/p}{\mathrm{RSS}/(n-p-1)}$$

- Under linear model assumption, can show

$$\mathbf{E}\left[\frac{\mathrm{RSS}}{n-p-1}\right] = \sigma^2.$$

- If also H_0 is true, can show

$$\mathbf{E}\left[\frac{\mathrm{TSS} - \mathrm{RSS}}{p}\right] = \sigma^2.$$

- Hence $F \approx 1$ if no relationship between response and predictors. Alternatively, if H_a true, $\mathbf{E}\left[(\mathrm{TSS} - \mathrm{RSS})/p\right] > \sigma^2$, hence $F > 1$.

Multiple Linear Regression

(1) Is there a relationship between response and predictors?

Statistics for multiple regression of **sales** onto **radio**, **TV** and **newspaper** in the advertising data set:

Quantity	Value
RSE	1.69
R^2	0.897
F	570

- $F \gg 1$ strong evidence against H_0.
- Proper threshold value for F depends on n, p.
  Larger F needed to reject H_0 for small n.
- H_0 true, ε_i Gaussian, then F follows **F-distribution**; calculate p-value using statistical software.
- Here, p-value ≈ 0 for $F = 590$ in this example, hence we safely reject H_0.

Multiple Linear Regression

(1) Is there a relationship between response and predictors?

- To test whether *subset* of last $q < p$ coefficients relevant, use null hypothesis

$$H_0: \quad \beta_{p-q+1} = \beta_{p-q+2} = \cdots = \beta_p = 0.$$

- Fit model using all variables *except* last q, obtaining residual sum of squares RSS_0.
- Appropriate F-statistic now

$$F = \frac{(\mathsf{RSS}_0 - \mathsf{RSS})/q}{\mathsf{RSS}\,/(n-p-1)}$$

- For multiple regression, t-statistic and p values for each variable indicate whether each predictor related to response after adjusting for the remaining variables.
  Equivalent to F-test omitting single variable ($q = 1$).
  Reports partial effect of adding each variable.

Multiple Linear Regression

(1) Is there a relationship between response and predictors?

What does F statistic tell us that individual p-values don't?

- Does single small p-value indicate at least one variable relevant? No.
- Example: $p = 100$, $H_0 : \beta_1 = \cdots = \beta_p = 0$ true.
  Then by chance, 5% of p-values below 0.05.
  Almost guaranteed that $p < 0.05$ for at least one variable by chance.
- Thus, for large p, looking only at p-values of individual t-statistics tends to discover spurious relationships.
- For F-statistic, if H_0 true, only 5% chance of p-value < 0.05 independently of n, p.

Note: F-statistic approach works for $p < n$.

Multiple Linear Regression

(2) Deciding on important variables

- Typically, not all predictors related to response (**variable selection** problem).
- One approach: try all possible models, select best one. Criteria? **Mallow's C_p**, **Akaike information criterion (AIC)**, **Bayesian information criterion (BIC)** (later)
- For p large, trying 2^p models with subsets of variables impractical.
- *Forward selection:* Start with null model (only β_0), fit p simple regressions, add variable leading to lowest RSS, then add variable leading to two-variable model with lowest RSS, continue until stopping criterion met.
- *Backward selection:* Start with full model, remove variable with largest p-value, fit new $(p-1)$-variable model, keep removing least significant variable, until stopping criterion met.
- *Mixed selection:* Start with null model, adding variables with best fit one-by-one, remove variables whenever its p-value rises above threshold, until model contains only variables with low p-values and excludes those with high p-value.

Multiple Linear Regression

(3) Model fit

RSE, R^2 computed and interpreted as in simple linear regression.

- $R^2 = \mathsf{Cor}(X, Y)^2$ for simple linear regression.
- $R^2 = \mathsf{Cor}(\hat{Y}, Y)^2$ for multiple linear regression, maximized by fitted model.
- $R^2 \approx 1$: model explains large portion of response variance.
- *Advertising example:*

$\{\mathbf{TV}, \mathbf{radio}, \mathbf{newspaper}\}$	$R^2 = 0.8972$
$\{\mathbf{TV}, \mathbf{radio}\}$	$R^2 = 0.89719$

  Small *increase* on including **newspaper** (even though **newspaper** not significant)

- Note: R^2 always increases when variables are added.
- Tiny increase in R^2 on including **newspaper** more evidence this variable can be dropped.
- Including redundant variables promotes overfitting.

Multiple Linear Regression

(3) Model fit

- *Advertising example:*

$$\{\textbf{TV}\} \qquad R^2 = 0.61$$
$$\{\textbf{TV}, \textbf{radio}\} \qquad R^2 = 0.89719$$

Substantial improvement on adding **radio**.
(Could also look at p-value of **radio**'s coefficient in last model.)

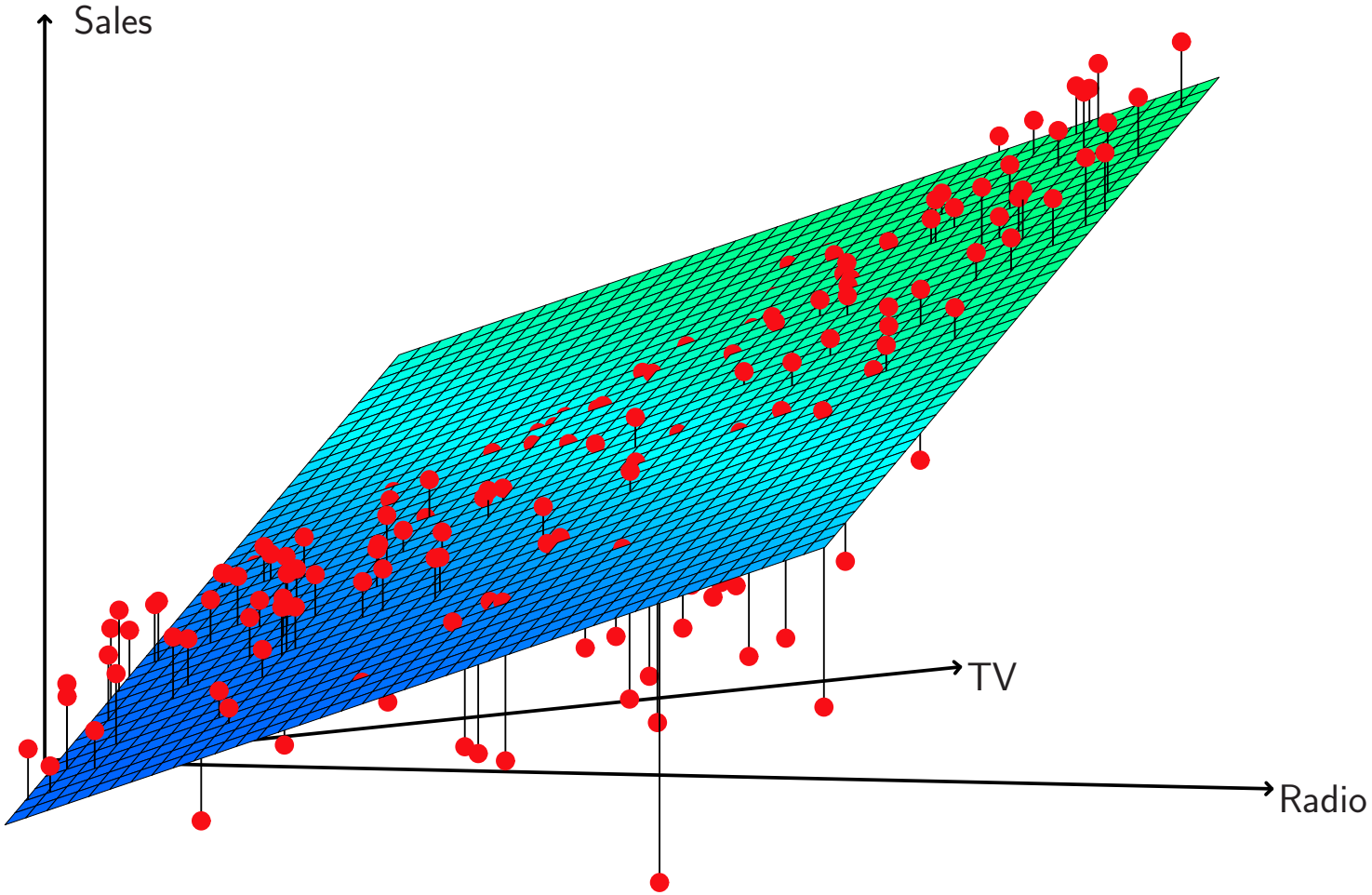
- *Advertising example:*

$$\{\textbf{TV}, \textbf{radio}, \ \textbf{newspaper}\} \qquad \text{RSE} = 1.686$$
$$\{\textbf{TV}, \textbf{radio}\} \qquad \text{RSE} = 1.681$$
$$\{\textbf{TV}\} \qquad \text{RSE} = 3.26$$

- Note: for multiple linear regression RSE defined as

$$\text{RSE} = \sqrt{\frac{\text{RSS}}{n-p-1}}.$$

$\{\mathbf{TV}, \mathbf{radio}\}$



Multiple Linear Regression

(3) Model fit

Previous figure:

- Some observations above, some below least squares regression plane.
- Linear model overestimates sales where most of budget spent either exclusively on **TV** or **radio**.
- Underestimation where budget split between two media.
- Such *nonlinear pattern* not reflected by linear model; suggests *synergy* effect between these two media.

Multiple Linear Regression

(4) Predictions

We note three sources of prediction uncertainty:

1. Reducible error: $\hat{Y} \approx f(X)$ since $\hat{\beta}_j \approx \beta_j$.
   Can construct confidence intervals to ascertain closeness $\hat{Y}$ to $f(X)$.
2. *Model bias*: linear model can only yield best *linear* approximation.
3. Irreducible error: $Y = f(X) + \varepsilon$.
   Assess prediction error with **prediction intervals**: incorporate both reducible and irreducible errors.

Example: Prediction using $\{\textbf{TV}, \textbf{radio}\}$ model.
$X_{\text{TV}} = 100\,000$ \$, $X_{\text{radio}} = 20\,000$ \$.

Confidence interval on **sales** : 95% confidence interval : $[10.985, 11.528]$.

Prediction interval on **sales**: 95% prediction interval : $[7.930, 14.580]$.

Increased uncertainty about sales for given city in contrast with average sales over many locations.

Contents

Other Considerations in the Regression Model

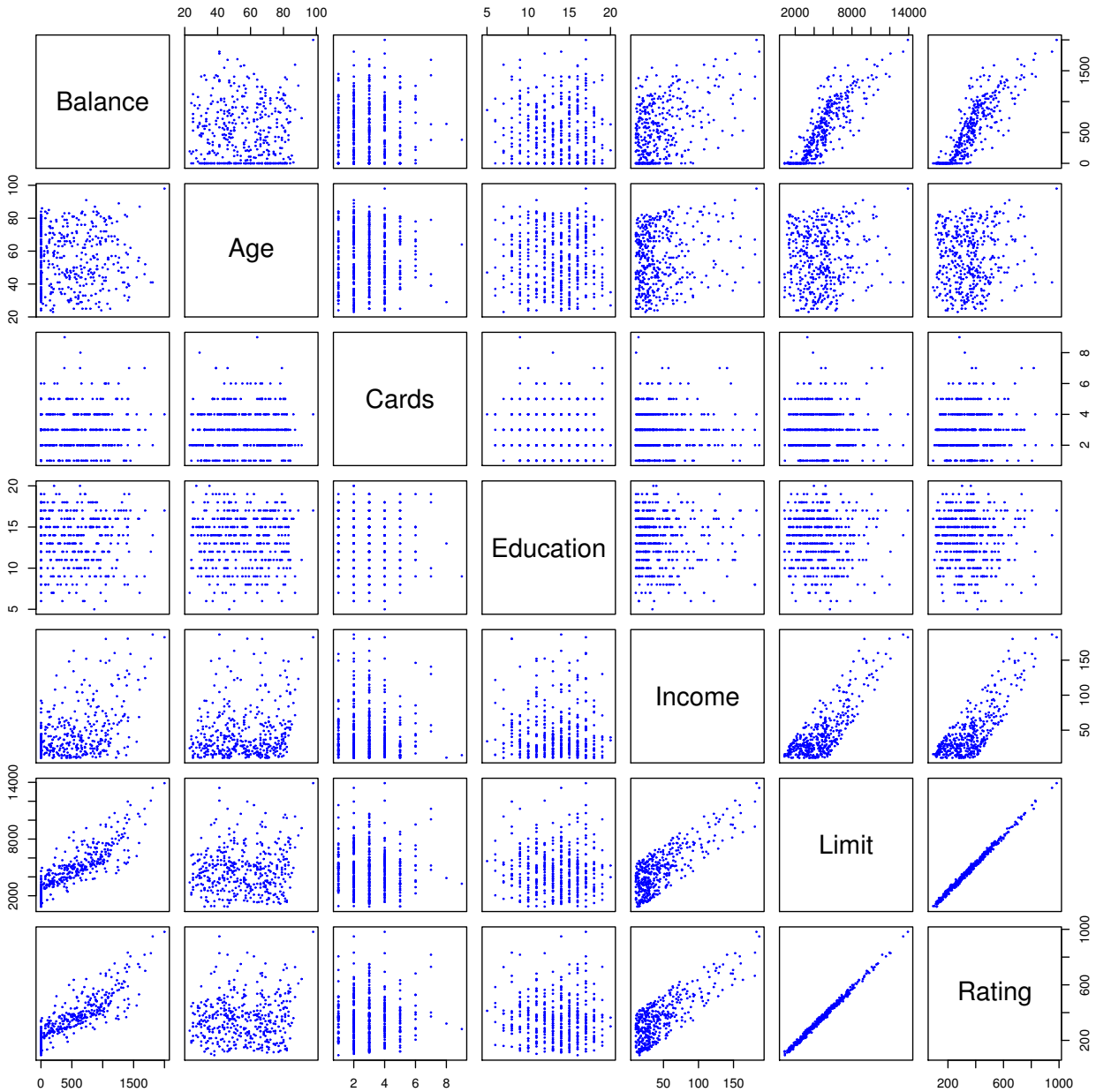
Qualitative predictors

Credit data set:

- Quantitative predictors:
  - **balance**: average credit card debt for a number of individuals
  - **age**
  - **cards** (# credit cards)
  - **education** (years of education)
  - **income** (in thousands of dollars)
  - **limit** (credit limit)
  - **rating** (credit rating)
- Qualitative predictors:
  - **gender**
  - **student** (student status)
  - **status** (marital status)
  - **ethnicity** (Caucasian, African American or Asian)

Other Considerations in the Regression Model

Qualitative predictors



Other Considerations in the Regression Model

Two-valued predictors

- Goal: investigate differences in credit card balance between males/females.
- **Gender** (qualitative variable, factor) represented with **indicator** (dummy variable)

$$x_i = \begin{cases} 1 & \text{if } i\text{-th person female,} \\ 0 & \text{if } i\text{-th person male.} \end{cases} \tag{3.22}$$

- Using x_i in regression equation results in model

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i = \begin{cases} \beta_0 + \beta_1 + \varepsilon_i & \text{if } i\text{-th person female,} \\ \beta_0 + \varepsilon_i & \text{if } i\text{-th person male.} \end{cases} \tag{3.23}$$

- Interpretation

$$\begin{aligned} \beta_0 &: \text{ average credit card balance among males,} \\ \beta_0 + \beta_1 &: \text{ average credit card balance among females,} \\ \beta_1 &: \text{average difference in credit card balance male/female.} \end{aligned}$$

Other Considerations in the Regression Model

Two-valued predictors

	Coefficient	Standard error	t-statistic	p-value
β_0	509.80	33.13	15.389	< 0.0001
β_1	19.73	46.05	0.429	0.6690

- Average credit card debt males: \$509.80.
- Average additional credit card debt females: \$19.73.
- Total average female credit card debt: \$529.53.
- High p value for dummy variable. Conclusion?
  Gender not a statistically significant factor for credit card debt.
- Switching male/female coding yields estimates

$$\hat{\beta}_0 = \$529.53, \qquad \hat{\beta}_1 = \$ - 19.73, \qquad \hat{\beta}_0 + \hat{\beta}_1 = \$509.80.$$

Other Considerations in the Regression Model

Two-valued predictors

Another alternative coding of two-valued **gender** predictor:

$$x_i = \begin{cases} 1 & \text{if } i\text{-th person female,} \\ -1 & \text{if } i\text{-th person male.} \end{cases}$$

Results in model

$$y_i = \beta_0 + \beta_1 x + \varepsilon_i = \begin{cases} \beta_0 + \beta_1 + \varepsilon_i & \text{if } i\text{-th person female,} \\ \beta_0 - \beta_1 + \varepsilon_i & \text{if } i\text{-th person male,} \end{cases}$$

with interpretation

β_0 : average credit card balance (ignoring gender),

β_1 : amount females are above/males below this average,

giving estimates

$\hat{\beta}_0 = \$519.665$ (half way between male and female averages)

$\hat{\beta}_1 = \$ \quad 9.865$ (half of \$19.63, average male/female difference).

Other Considerations in the Regression Model

Multi-valued qualitative predictors

To encode **ethnicity** $\in \{$Caucasian, African American, Asian$\}$, use multiple dummy variables (# values -1)

$$x_{i,1} = \begin{cases} 1 & \text{if } i\text{-th person Asian,} \\ 0 & \text{if } i\text{-th person not Asian,} \end{cases} \tag{3.24}$$

$$x_{i,2} = \begin{cases} 1 & \text{if } i\text{-th person Caucasian,} \\ 0 & \text{if } i\text{-th person not Caucasian,} \end{cases} \tag{3.25}$$

resulting in model

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i.2} + \varepsilon_i = \begin{cases} \beta_0 + \beta_1 + \varepsilon_i & \text{if } i\text{-th person Asian} \\ \beta_0 + \beta_2 + \varepsilon_i & \text{if } i\text{-th person Caucasian} \\ \beta_0 + \varepsilon_i & \text{if } i\text{-th person African American} \end{cases} \tag{3.26}$$

Interpretation: β_0 : average credit card balance for African Americans (**baseline**),

β_1 : difference between Asian and African Americans,

β_2 : difference between Caucasian and African Americans

Other Considerations in the Regression Model

Multi-valued qualitative predictors

	Coefficient	Standard error	t-statistic	p-value
β_0	531.00	46.32	11.464	< 0.0001
β_1 (Asian)	−18.69	65.02	−0.287	0.7740
β_2 (Caucasian)	−12.50	56.68	−0.221	0.8260

- Estimated balance for African Americans (baseline): \$531.00.
- Asians estimated to have \$18.69 less debt than African Americans.
- Caucasians estimated to have \$12.50 less debt than African Americans.
- β_1, β_2 have high p-values, indicating no statistical significance for ethnicity as factor in credit card balance.
- Coefficients and p-values depend on coding, result does not.
  F-test to reject $H_0 : \beta_1 = \beta_2 = 0$ has p-value 0.96 (cannot reject).
- Dummy variable approach works for combining qualitative and quantitative predictors.
  (Other coding schemes for qualitative variables possible.)

Other Considerations in the Regression Model

Extending the linear model

- Restrictive assumptions in linear model: **linearity**, **additivity**.
- Additivity: effect on Y of changing X_j independent of remaining variables.
- Linearity: rate of change in Y with respect to X_j constant in X_j.
- Recall advertising data set: indication that higher **radio** budget made effect of **TV** spending stronger (interaction effect, synergy).
- Add **interaction term** to two-predictor model:

$$\begin{aligned} Y &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon \\ &= \beta_0 + (\beta_1 + \beta_3 X_2) X_1 + \beta_2 X_2 + \varepsilon \\ &= \beta_0 + \tilde{\beta}_1 X_1 + \beta_2 X_2 + \varepsilon, \qquad \tilde{\beta}_1 := \beta_1 + \beta_3 X_2. \end{aligned}$$

$\tilde{\beta}_1$ changes with X_2, hence effect of X_1 on Y changes with X_2.

Other Considerations in the Regression Model

Extending the linear model: factory example

Example: facttory productivity.

- Predict # produced **units** based on # production **lines** and # **workers**.
- Expected: increase in # production **lines** will depend on # **workers**.
- In linear model of **units**, include interaction term between **lines** and **workers**:

$$\begin{aligned}\textbf{units} &\approx 1.2 + 3.4 \times \textbf{lines} + 0.22 \times \textbf{workers} + 1.4 \times (\textbf{lines} \times \textbf{workers}) \\ &= 1.2 + (3.4 + 1.4 \times \textbf{workers}) \times \textbf{lines} + 0.22 \times \textbf{workers}\end{aligned}$$

- Adding additional line will increase # produced units by $3.4 + 1.4 \times \textbf{workers}$. The more **workers**, the stronger the effect of adding a **line**.

Other Considerations in the Regression Model

Extending the linear model: advertising example

Linear model for **sales** predicted by interacting **TV**, **radio** terms:

$$\begin{aligned} \textbf{sales} &= \beta_0 + \beta_1 \times \textbf{TV} + \beta_2 \times \textbf{radio} + \beta_3 \times (\textbf{radio} \times \textbf{TV}) + \varepsilon \\ &= \beta_0 + (\beta_1 + \beta_3 \times \textbf{radio}) \times \textbf{TV} + \beta_2 \times \textbf{radio} + \varepsilon \end{aligned} \tag{3.27}$$

Interpretation of β_3 : increase in effectiveness of TV advertising for one-unit increase in radio advertising.

	Coefficient	Standard error	t-statistic	p-value
β_0	6.7502	0.248	27.23	< 0.0001
β_1	0.0191	0.002	12.70	< 0.0001
β_2	0.0289	0.009	3.24	0.0014
β_3	0.0011	0.000	20.73	< 0.0001

- Model with interaction term superior to that including only **main effects**.
- Low p-value of interaction term strong evidence for rejecting $H_0 : \beta_3 = 0$.

Other Considerations in the Regression Model

Extending the linear model: advertising example

- Model (3.27) has $R^2 = 96.8\%$
  (vs. $R^2 = 89.7\%$ for model without interaction term).
- Interpretation: of the variability remaining after fitting the model without interaction term,
$$\frac{96.8\% - 89.7\%}{100\% - 89.7\%} = 69\%$$
  is explained by model (3.27) which includes the interaction term.
- \$1000 increase in TV budget associated with sales increase of
  $(\hat{\beta}_1 + \hat{\beta}_3 \times \textbf{radio}) \times 1000 = 19 + 1.1 \times \textbf{radio}$ units.
  \$1000 increase in radio budget associated with sales increase of
  $(\hat{\beta}_2 + \hat{\beta}_3 \times \textbf{TV}) \times 1000 = 29 + 1.1 \times \textbf{TV}$ units.
- **Hierarchical principle**: for every interaction term, include all associated main effects, even if the p values of their coefficients not significant.
  Rationale: If X_1X_2 related to response, vanishing coefficients for X_1, X_2 unimportant. X_1X_2 typically correlated with X_1, X_2; leaving these out alters meaning of interaction.

Other Considerations in the Regression Model

Extending the linear model: credit example

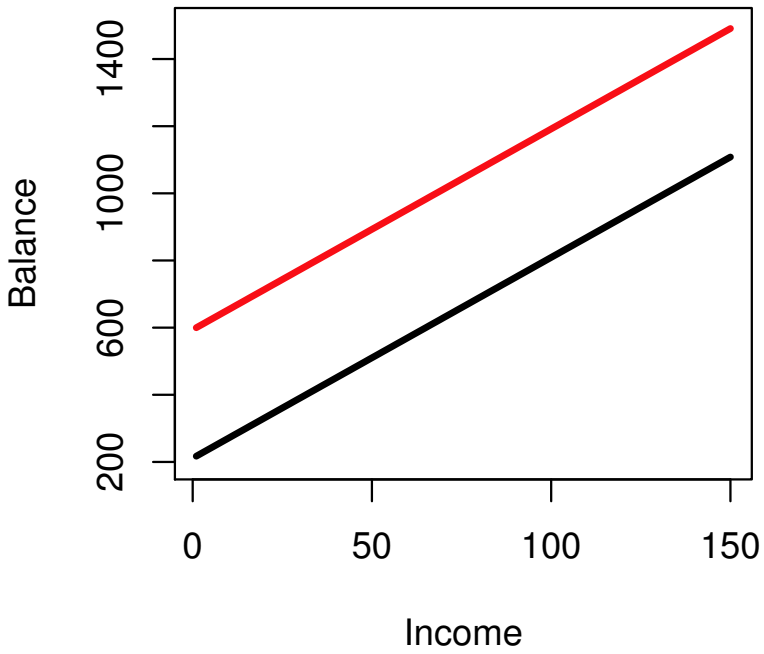
Credit data set: predict **balance** using **income** (quantitative) and **student** (qualitative). Without interaction term:

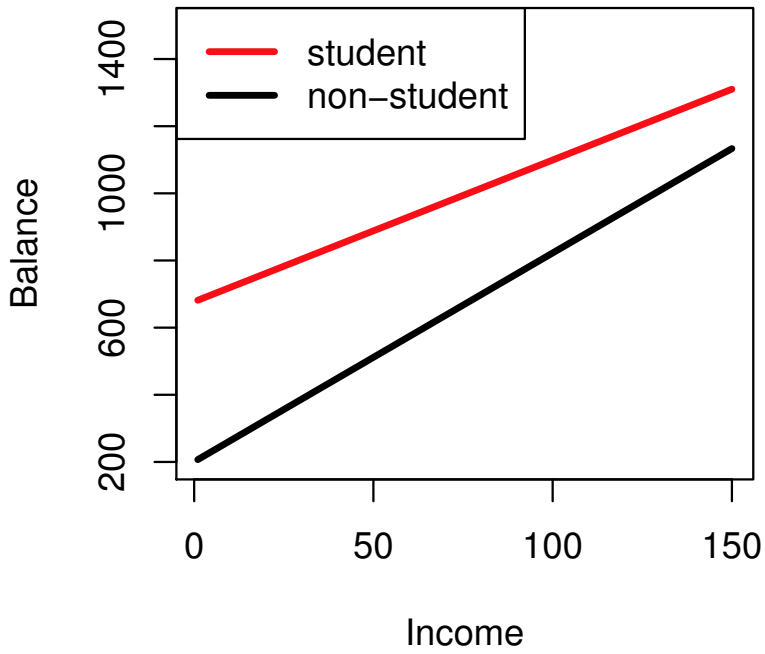
$$\begin{aligned} \textbf{balance}_i &\approx \beta_0 + \beta_1 \times \textbf{income}_i + \begin{cases} \beta_2 & \text{if } i\text{-th person student} \\ 0 & \text{otherwise} \end{cases} \\ &= \beta_1 \times \textbf{income}_i + \begin{cases} \beta_0 + \beta_2 & \text{if } i\text{-th person student} \\ \beta_0 & \text{otherwise.} \end{cases} \end{aligned} \tag{3.28}$$

- Results in fitting two parallel lines to data (one each for students and non-students).
- Parallel implies: average affect on balance of one-unit increase in **income** independent of **Student** status.
- Reflects model shortcoming: change in **income** may have very different effect on credit card balance for students and non-students.

Other Considerations in the Regression Model

Extending the linear model: credit example





With interaction term: multiply **income** with dummy variable for **student**

$$\textbf{balance}_i \approx \beta_0 + \beta_1 \times \textbf{income}_i + \begin{cases} \beta_2 + \beta_3 \times \textbf{income}_i & \text{if } i\text{-th person student} \\ 0 & \text{otherwise} \end{cases}$$

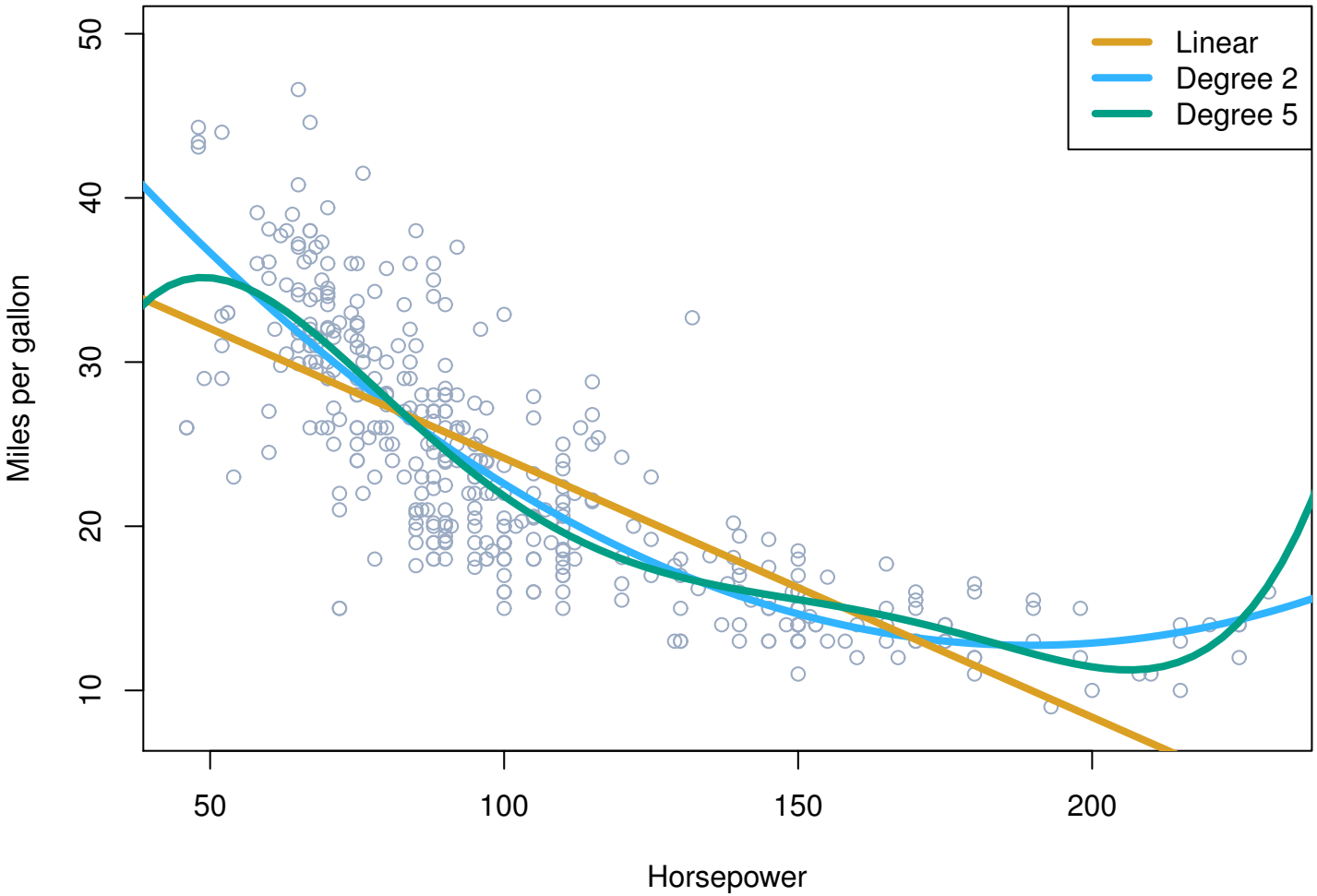
$$= \begin{cases} (\beta_0 + \beta_2) + (\beta_1 + \beta_3) \times \textbf{income}_i & \text{if } i\text{-th person student} \\ \beta_0 + \beta_1 \times \textbf{income}_i & \text{otherwise.} \end{cases} \tag{3.29}$$

Other Considerations in the Regression Model

Extending the linear model: credit example

- Now the two lines have different intercepts and different slopes.
- Slope for students lower, indicates increases in income associated with smaller increase in credit card balance than for non-students.

Polynomial regression vs. linear regression:



Auto data set showing **mpg** (miles per gallon) versus **horsepower** for different cars.

Other Considerations in the Regression Model

Extending the linear model: nonlinear relationships

Since the data seem to suggest curved relationship, add quadratic term:

$$\textbf{mpg} = \beta_0 + \beta_1 \times \textbf{horsepower} + \beta_2 \times \textbf{horsepower}^2 + \varepsilon. \tag{3.30}$$

	Coefficient	Standard error	t-statistic	p-value
β_0	56.9001	1.8004	31.6	< 0.0001
β_1	-0.4662	0.0311	-15.0	< 0.0001
β_2	0.0012	0.0001	10.1	< 0.0001

- Linear fit has $R^2 = 0.606$, quadratic fit has $R^2 = 0.688$.
- p-value for quadratic term highly significant.
- Degree 5 fit more oscillatory, doesn't appear to explain data any better than quadratic.

Other Considerations in the Regression Model

Potential problems

Most common problems when fitting a linear regression model to a data set: (identification and solution as much an art as a science)

1. Nonlinear dependence of response on predictors
2. Correlated error terms
3. Non-constant variance of error terms
4. Outliers
5. High-leverage points
6. Collinearity

Other Considerations in the Regression Model

Potential problems: (1) Nonlinear dependence

Inference and prediction from linear regression model suspect when true model nonlinear.

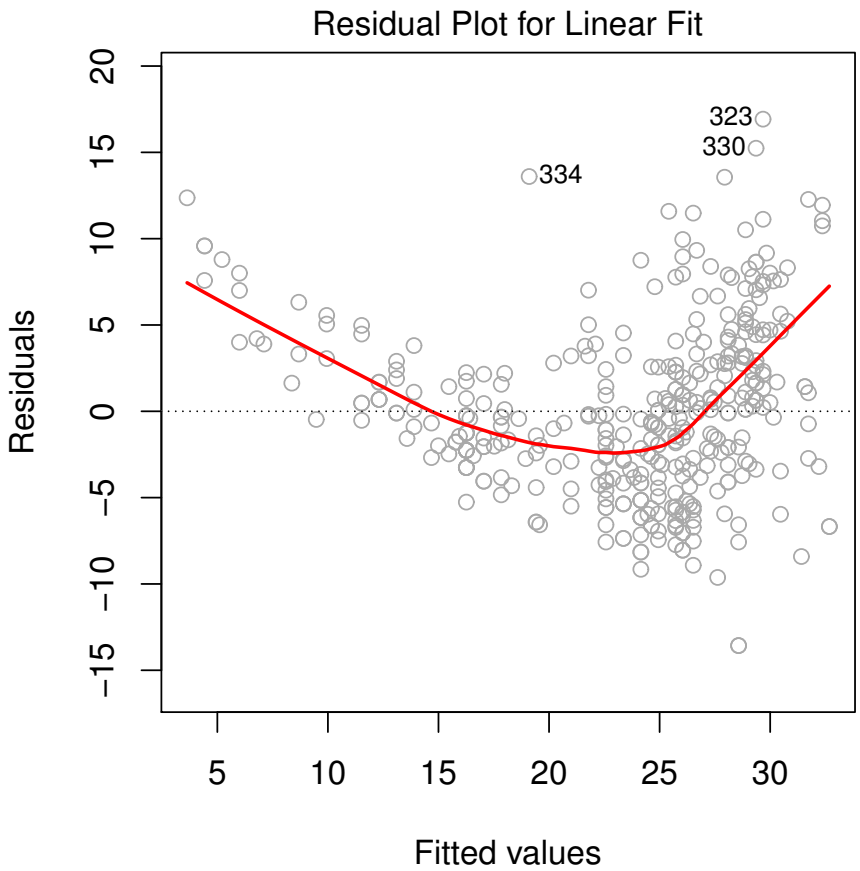
- Identifying nonlinearity aided by **residual plots**

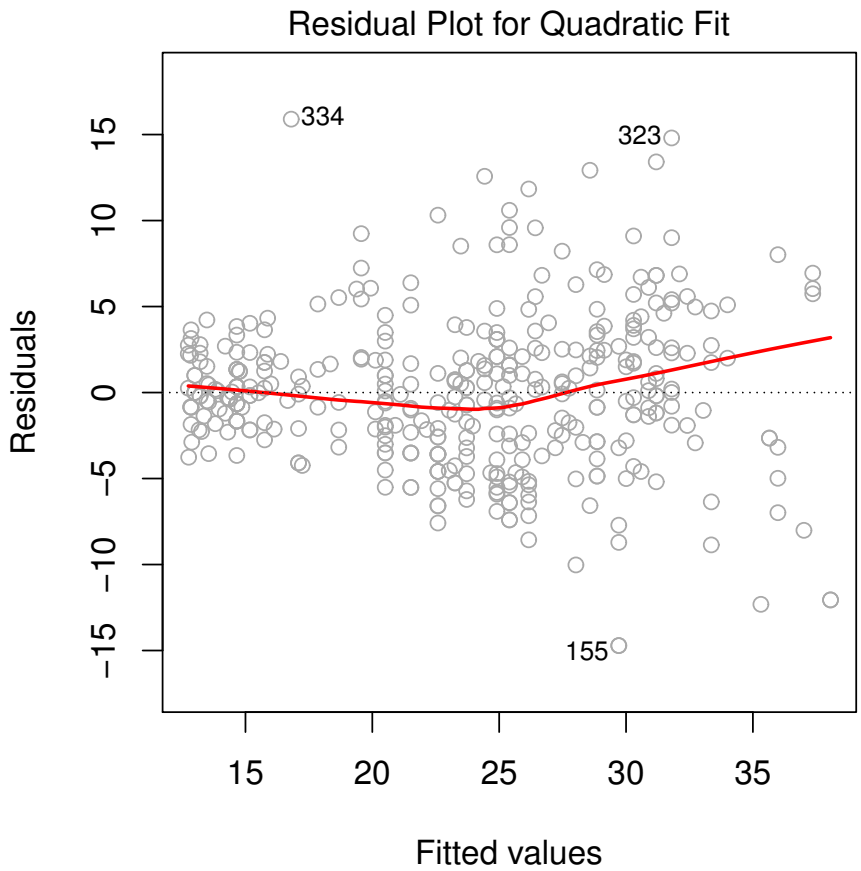
$$e_i = y_i - \hat{y}_i \qquad \text{against predictors } x_i.$$

- For multiple regression models, plot residuals against predicted (fitted) values $\hat{y}_i$.
- Ideal picture: no discernible pattern.
- Pattern indicates possible problem with model.
- When nonlinearity is suggested, introduce nonlinear functions of predictors as **regression functions** into the regression model.

Other Considerations in the Regression Model

Potential problems: (1) Nonlinear dependence





Residuals versus predicted values for **Auto** data set.
Red line is smooth fit to residuals to aid in identifying trends.
Left: linear regression of **mpg** on **horsepower** (strong pattern).
Right: linear regression of **mpg** on **horsepower** and **horsepower**^{2} (little pattern).

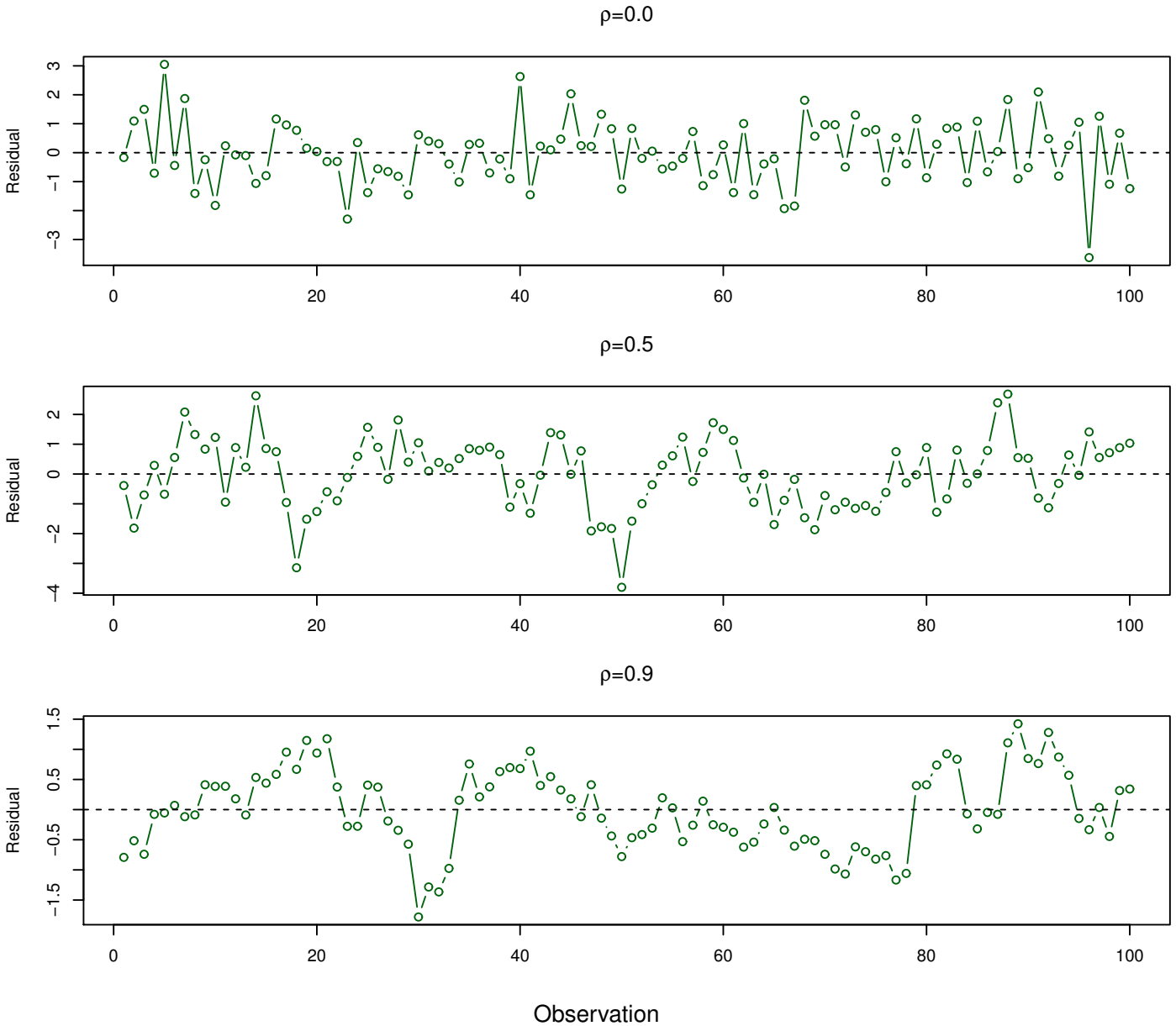
Other Considerations in the Regression Model

Potential problems: (2) Correlated error terms

- Linear regression assumes uncorrelated errors ε_i.
- Computation of SE for coefficient estimates, fitted values, based on this assumption. Otherwise estimated SE tend to underestimate true SE, confidence and prediction intervals too optimistic (narrow), p-values lower than they should be.
- Extreme example: double data (observations, error terms identical in pairs). SE calculations use sample size $2n$ in place of n, hence CI narrower by factor of $\sqrt{2}$.
- Detection for **time series**: plot residuals as function of time. No correlations implies no visible pattern; correlations lead to **tracking** of residuals.
- Example (next slide): time series with error correlation $\rho = 0, 0.5, 0.9$
- Example: study of persons' heights predicted from their weights. Uncorrelatedness assumption violated if, e.g., individuals related, same diet or environmantal factors.

Other Considerations in the Regression Model

Potential problems: (2) Correlated error terms



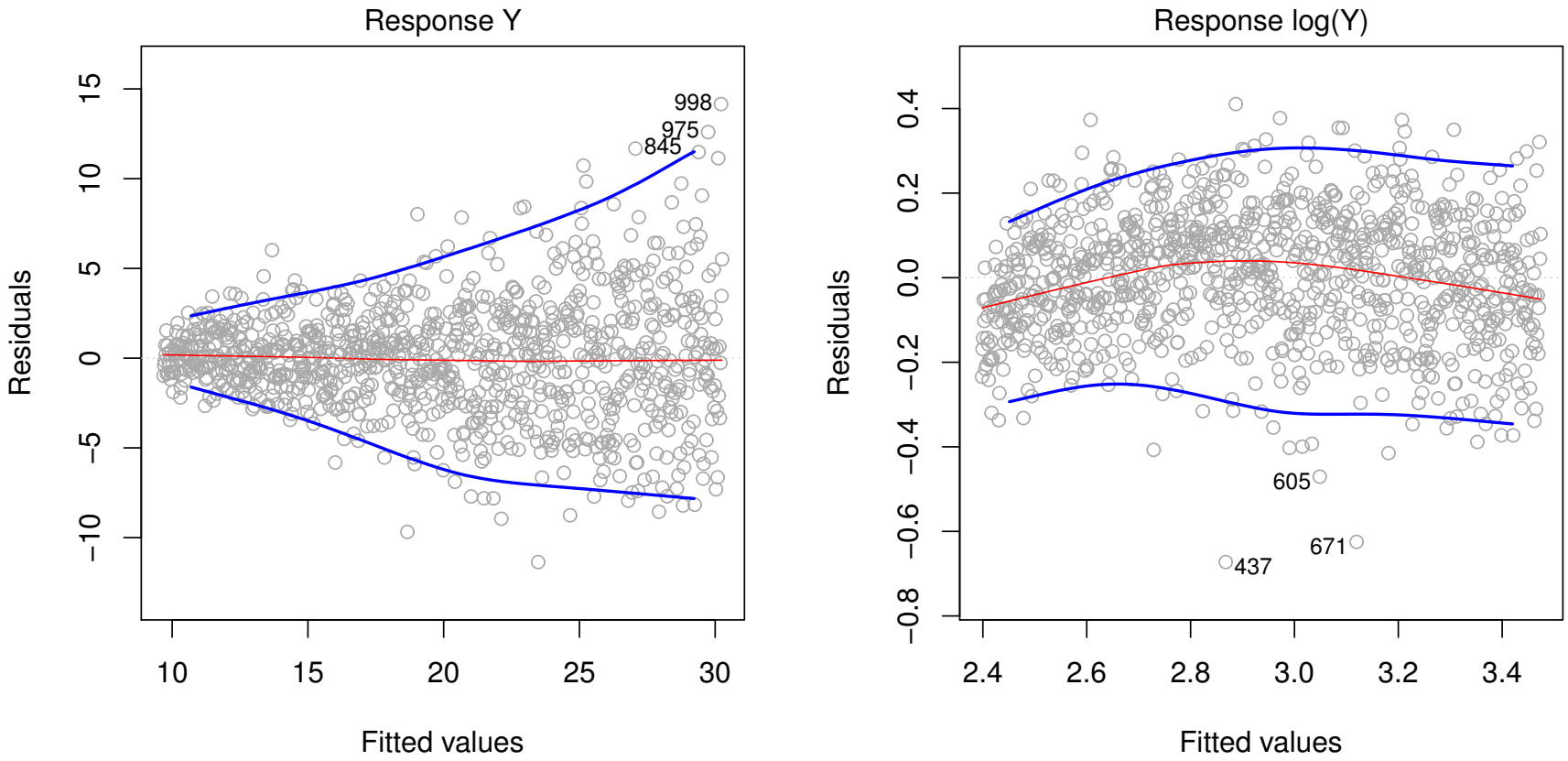
Other Considerations in the Regression Model

Potential problems: (3) Non-constant variance of error terms

- SE, CI, hypothesis tests associated with linear model rely on assumption $\mathbf{Var}\,\varepsilon_i = \sigma^2$ $(\forall i)$.
- Non-constant error variance (**heteroscedasticity**), e.g. increase with response value, leads to *funnel-shaped* residual plot.
- Possible solution: transform response Y using concave function such as $\log Y$ or $\sqrt{Y}$, leads to damping of larger responses, reducing heteroscedasticity.
- When variation of response variance known, e.g., i-th response average of n_i observations which are uncorrelated with variance σ^2, then average has variance $\sigma_i^2 = \sigma^2/n_i$. Remedy: **weighted least squares** with weights proportional to inverse variances.

Other Considerations in the Regression Model

Potential problems: (3) Non-constant variance of error terms

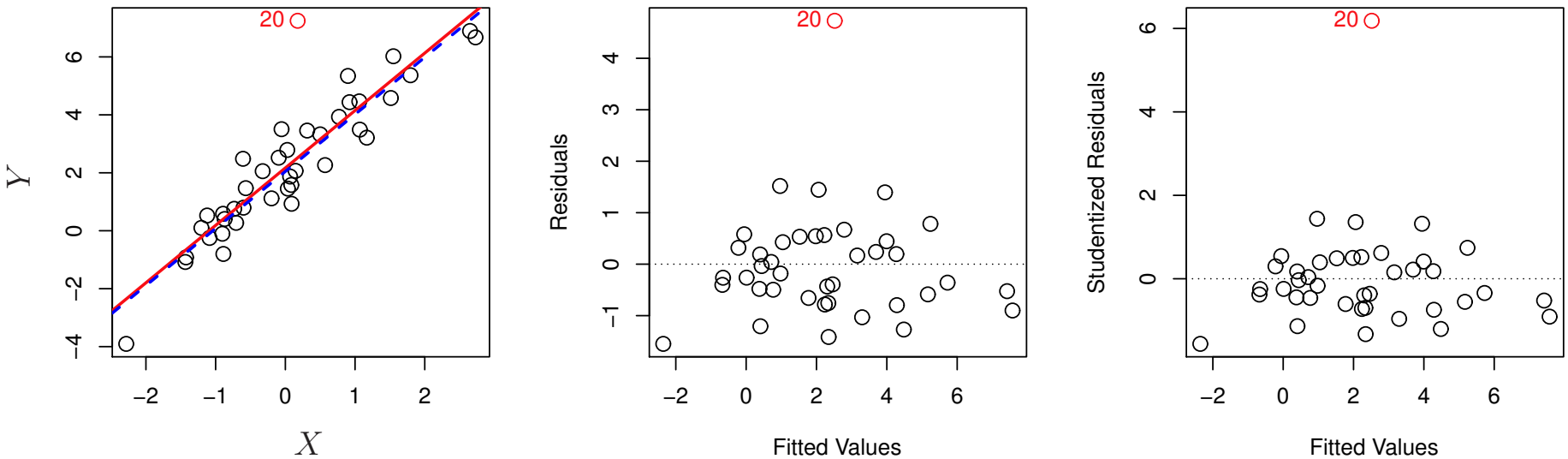


Residual plots. Red: smooth fit of residuals. Blue: track outer quantiles of residuals.
Left: funnel shape indicating heteroscedasticity.
Right: After log-transforming respone, heteroscedasticity removed.

Other Considerations in the Regression Model

Potential problems: (4) Outliers

- **Outlier**: point where y_i far from value predicted by model.
- Possible causes: observation errors.



Left: red solid line: least squares line with outlier, blue: without.
Center: Residual plot identifies outlier.
Right: Outlier seen to have studentized residual (divide e_i by its estimated standard error) of 6 (between -3 and 3 expected).
R^2 declines from 0.892 to 0.805 on including outlier.

Other Considerations in the Regression Model

Potential problems: (5) High-leverage points

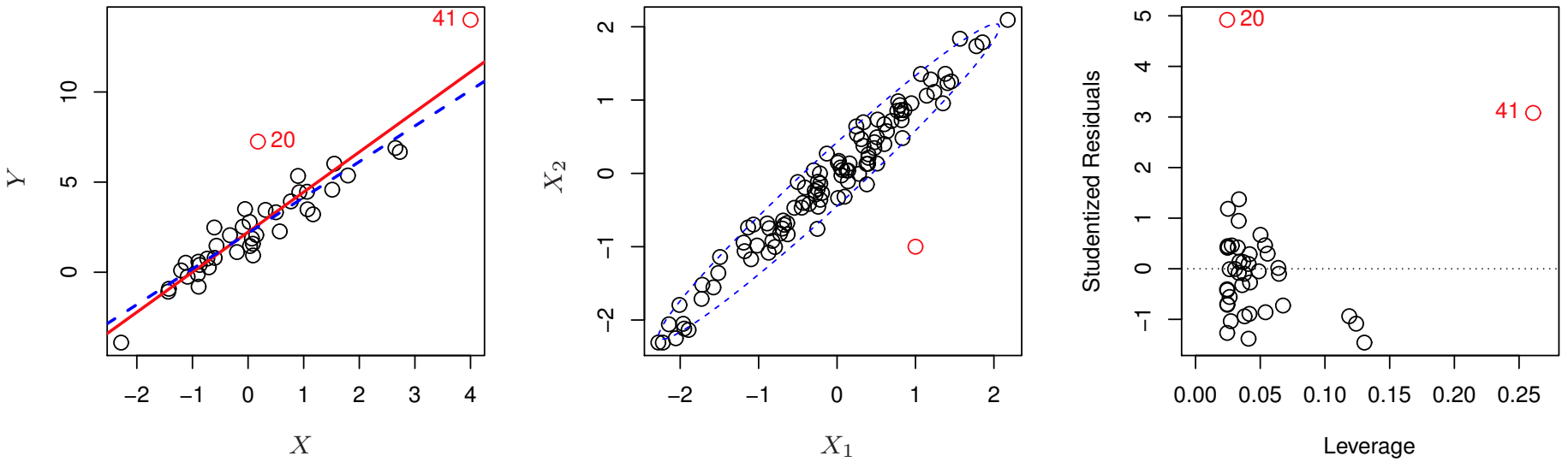
- Outliers: observations where y_i is unusual given x_i.
- Observations with **high leverage** have unusual value for x_i.
- If least squares line strongly affected by certain points, problems with these may invalidate entire fit, hence important to identify such observations.
- Simple linear regression: extremal x-values; multiple linear regression: in range of all other observation coordinates, but unusual (difficult to detect for more than two predictors).
- Large value of **leverage statistic** indicates high leverage.
  For simple linear regression:

$$h_i = \frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_{i'=1}^{n}(x_{i'} - \overline{x})^2} \in \left(\frac{1}{n}, 1\right). \tag{3.31}$$

  Average value always $\frac{p+1}{n}$, deviation from average indicates high leverage.

Other Considerations in the Regression Model

Potential problems: (5) High-leverage points

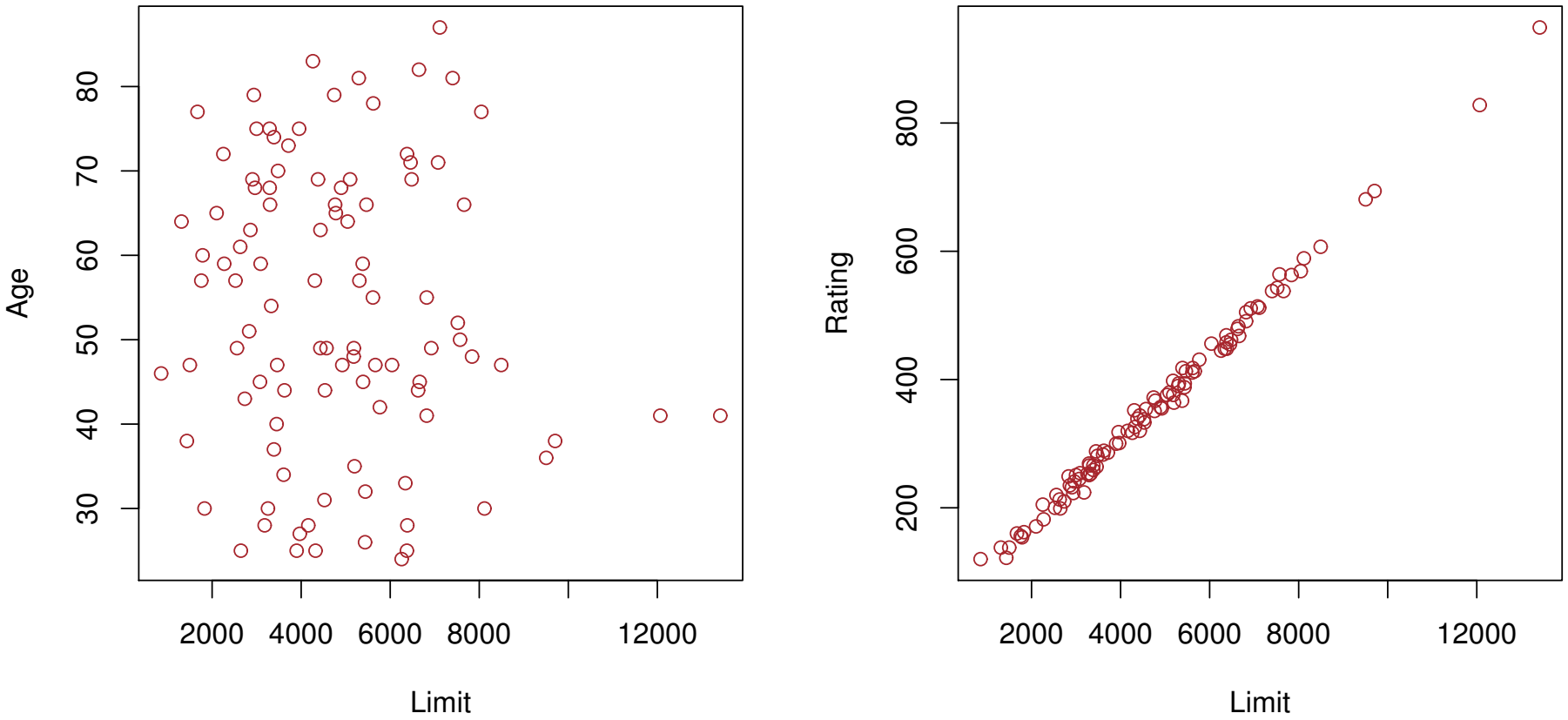


Left: Same data as previous figure, with added observation 41 (red) of high leverage. Red solid line is least squares fit with, blue dashed without observation 41.
Center: two predictor variables, most observations within blue dashed ellipse, red observation distinctly outside.
Right: same data as in left panel, studentized redisulas vs. leverage statistic. Observation 41 has high leverage and high residual, i.e., outlier *and* high-leverage point.
Outlier observation 20 has low leverage.

Other Considerations in the Regression Model

Potential problems: (6) Collinearity

Collinearity: two or more predictor variables closely related.

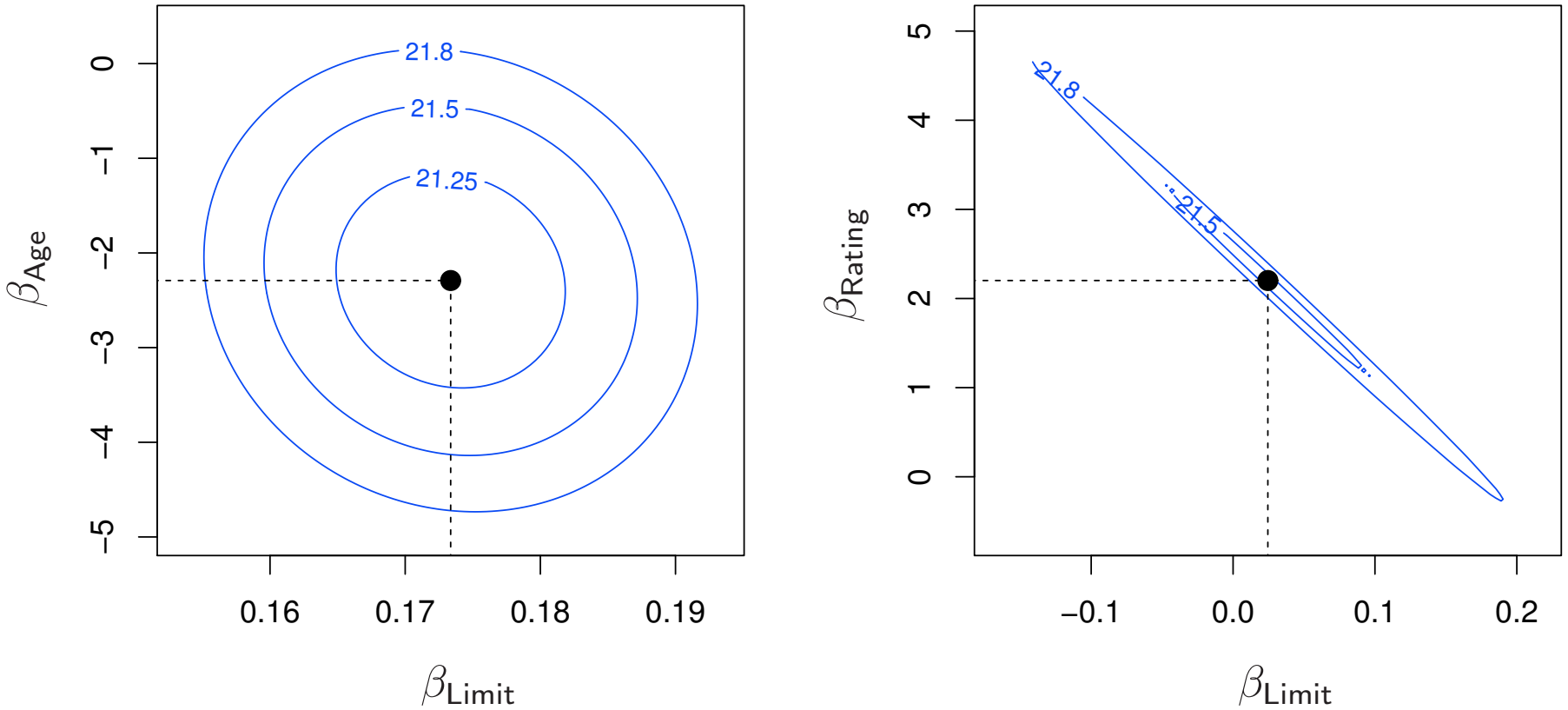


From **Credit** data set. Left: **limit** vs. **age**. Right: **limit** vs. **rating** (strongly collinear).

Other Considerations in the Regression Model

Potential problems: (6) Collinearity

Difficult to separate individual effects of collinear variables on response.



Contour plot of RSS associated with different coefficient estimates for **Credit** data set.
Axes scaled to include 4 SE on either side of optimum.
Left: for regression of **balance** on **limit** and **age**.
Right: for regression of **balance** on **limit** and **rating**.

Other Considerations in the Regression Model

Potential problems: (6) Collinearity

- Collinearity increases SE, hence reduces t-statistic, and we will more likely fail to reject $H_0 : \beta_j = 0$. This reduces the **power** of the hypothesis test, i.e., the probability of correctly detecting a nonzero coefficient.

	Coefficient	Standard error	t-statistic	p-value
		Model 1		
β_0	-173.411	43.828	-3.957	< 0.0001
β_1 (**age**)	-2.292	0.672	-3.407	0.0007
β_2 (**limit**)	0.173	0.005	34.496	< 0.0001
		Model 2		
β_0	-377.537	45.254	-8.343	< 0.0001
β_1 (**rating**)	2.202	0.952	2.312	0.0213
β_2 (**limit**)	0.025	0.064	0.384	0.7012

- Model 1: **age**, **limit** both highly significant.
  Model 2: collinearity between **rating** and **limit** increases SE for **limit** coefficient by factor 12, p-value increases to 0.701. Collinearity masks importance of **limit** variable.

Other Considerations in the Regression Model

Potential problems: (6) Collinearity

- Important to detect collinearity when fitting a model.
- Correlation matrix may give indication.
- **Multicollinearity**: collinearity between 3 or more variables which each have low pairwise correlation.
- **Variance inflation factor (VIF)**: ratio of variance of $\hat{\beta}_j$ when fitting the full model and variance of $\hat{\beta}_j$ when fitted separately.
- VIF ≥ 1, minimum at complete absence of collinearity.
  Problematic if VIF exceeds 5 or 10.
- $$\text{VIF}(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}$$

  $R^2_{X_j|X_{-j}}$: R^2 from regression of X_j onto all other predictors.
- In **Credit** data example: predictors have VIF values of 1.01, 160.67, 160.59.
- Remedies: drop problematic variables, combine collinear variables into single predictor.

Contents

Revisiting the Marketing Data Questions

Recall the seven questions relating to the **Advertising** data set we set out to answer on Slide 71:

1. Is there a relationship between advertising budget and sales?
2. How strong is this relationship between advertising budget and sales?
3. Which media contribute to sales?
4. How accurately can we estimate the effect of each medium on sales?
5. How accurately can we predict future sales?
6. Is the relationship linear?
7. Is there synergy among the advertising media?

We revisit each in turn.

Revisiting the Marketing Data Questions

1. Is there a relationship between advertising budget and sales?

- Fit multiple regression model of **sales** onto **TV**, **radio** and **newspaper**.
- Test hypothesis $H_0 : \beta_{\text{TV}} = \beta_{\text{radio}} = \beta_{\text{newspaper}} = 0$.
- Rejection/non-rejection based on F-statistic (Slide 100).
- For advertising data: low p-value of F-statistic (table on Slide 102) strong evidence for rejecting H_0.

Revisiting the Marketing Data Questions

2. How strong is this relationship between advertising budget and sales?

- Measure of model error: RSE (see Slide 80), estimates standard deviation of response from (true) population regression line.
- Advertising data:
  For multiple regression model of **sales** on **TV** and **radio**, RSE $= 1,681$ units (Slide 107).
  Relative to response sample mean of $14,022$ units, this is an error of 12%.
- Measure of model error: R^2 (Slide 87), measures proportion of response variability explained by model.
- Advertising data:
  For multiple regression model of **sales** on **TV**, **radio** and **newspaper**, $R^2 = 0.897$, i.e., $\approx 90\%$ of **sales** variability explained by multiple linear regression model (Slide 102).

Revisiting the Marketing Data Questions

3. Which media contribute to sales?

- p-values of t-statistic in multiple regression model of **sales** on **TV**, **radio** and **newspaper**: small for **TV** and **radio**, large for **newspaper**.
- Suggest only **TV** and **radio** budgets related to **sales**.

Revisiting the Marketing Data Questions

4. How accurately can we estimate the effect of each medium on sales?

- Confidence intervals for β_j constructed from SE of $\hat{\beta}_j$.
- Advertising data: 95%-confidence intervals for multiple regression coefficients are

TV	$(0.043, 0.049)$
radio	$(0.172, 0.206)$
newspaper	$(-0.013, 0.011)$

- Wide SE due to collinearity? (Slide 138).
  VIF scores for **TV**, **radio** and **newspaper** are 1.005, 1.145, 1.145, so not likely.
- Separate simple regressions of **sales** on **TV**, **radio** and **newspaper** show strong association of **TV** and **radio** with **sales**, mild association of **newspaper** with **sales**, when remaining two predictors ignored.

Revisiting the Marketing Data Questions

5. How accurately can we predict future sales?

- Can use (3.19) for prediction.
- Precition intervals assess accuracy of predicting individual responses $Y = f(X) + \varepsilon$.
- Confidence intervals assess accuracy of predicting average responses $Y = f(X)$.
- Former always wider due to accounting for additional variability due to irreducible error ε.

Revisiting the Marketing Data Questions

6. Is the relationship linear?

- Identify nonlinearity using residual plots of linear model (Slide 129).
- Advertising data:
  Nonlinear effects visible in figure on Slide 108.
- Discussed regression functions which are nonlinear in the predictor variables.

Revisiting the Marketing Data Questions

7. Is there synergy among the advertising media?

- Non-additive relationships modeled by interaction term in model (Slide 119).
- Presence of interaction (synergy) confirmed by small p-value of interaction term.
- Advertising data:
  Including interaction term increased R^2 from $\approx 90\%$ to $\approx 97\%$.

Contents

Linear Regression vs. K-Nearest Neighbors

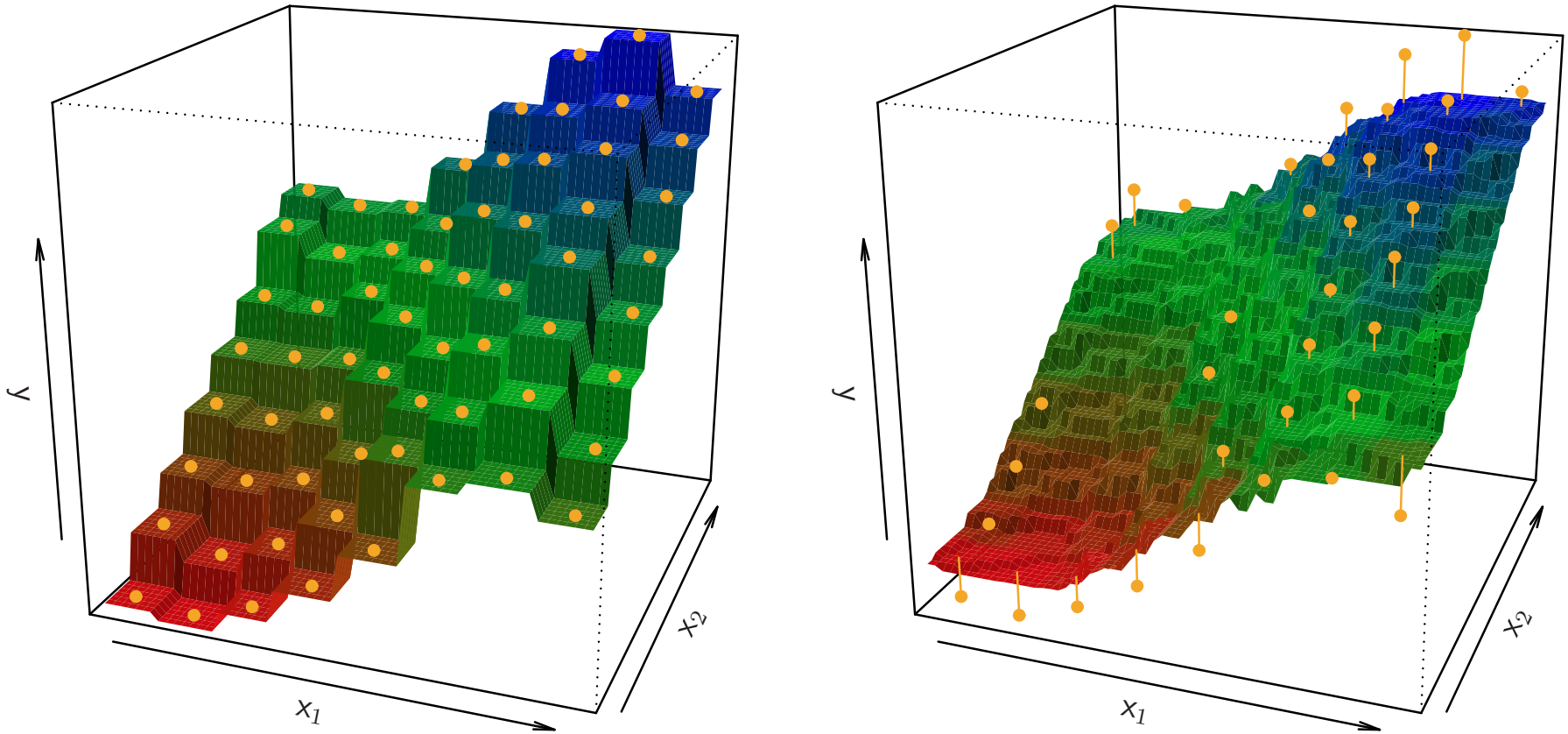
Non-parametric approach

- Linear regression is a parametric method.
- Non-parametric methods make no strong a priori assumptions on functional form of model $Y \approx f(X)$, more flexibility in adapting to data.
- Here: **K-nearest neighbors (KNN) regression** (Cf. KNN classifier in Chapter 2).
- Given prediction point x_0, first determine the set $\mathcal{N}_0$ consisting of the K ($K \in \mathbb{N}$) training observations closest to x_0.
- Predict $\hat{y}_0$ to be average training response in $\mathcal{N}_0$, i.e.,

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in \mathcal{N}_0} y_i.$$

Linear Regression vs. K-Nearest Neighbors

Non-parametric approach



Two KNN fits on a data set with 64 observations using $p = 2$ predictors.
Left: $K = 1$. Interpolation, rough step-like function.
Right: $K = 9$. Not interpolatory, smoother.

Linear Regression vs. K-Nearest Neighbors

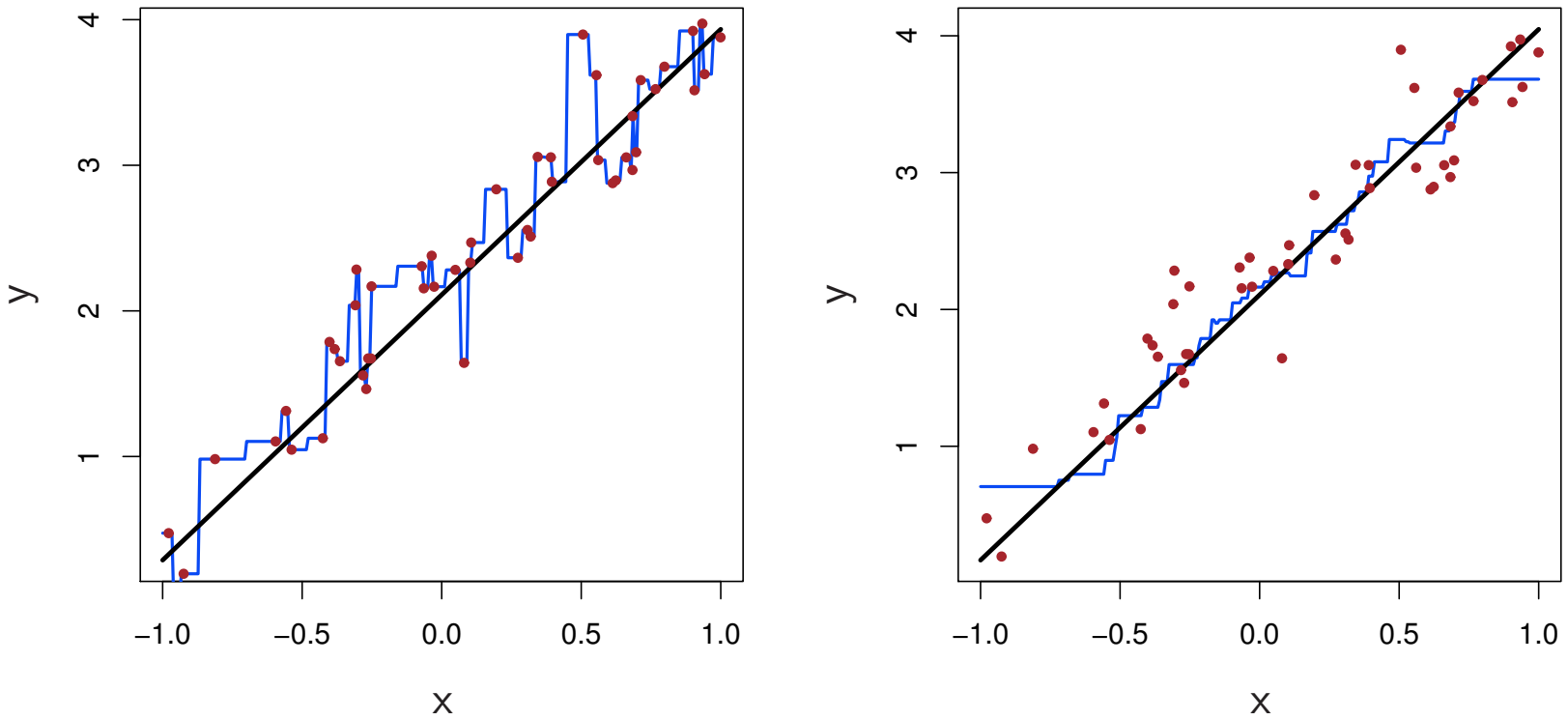
Tuning K

- Flexibility of model controlled by K: less flexible. smoother fit, for large K.
- Bias-variance tradeoff.
- Flexible model: low bias, high variance
  (prediction depends on only one nearby observation).
  Unflexible model: high bias, low variance (changing one observation has smaller effect, averaging introduces bias).
- Optimal value of K? (later)

Linear Regression vs. K-Nearest Neighbors

Parametric vs. non-parametric

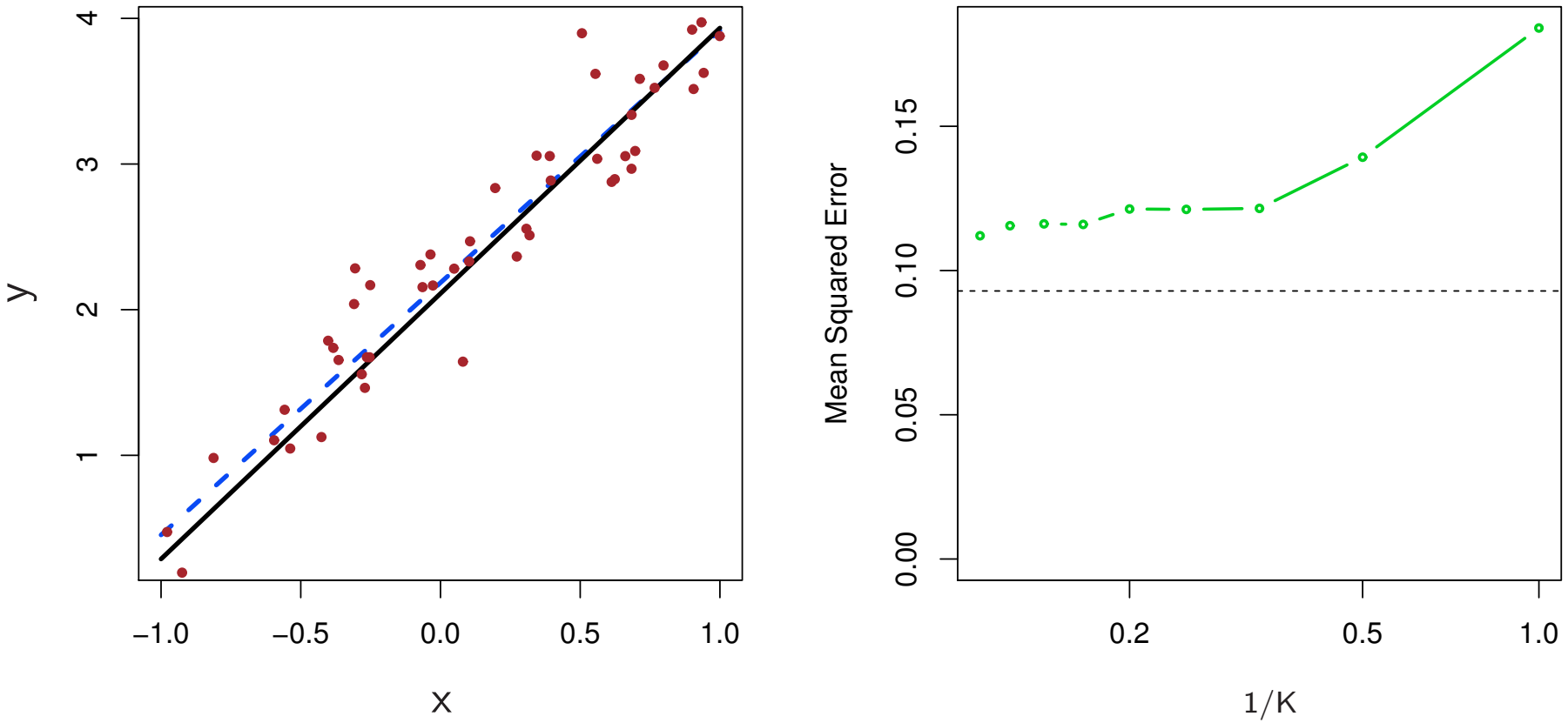
Q: In what setting will a parametric approach outperform a non-parametric approach?
A: Depends on how closely assumed form of f matches true form.



1D data, 100 observations (red), linear true model (black), KNN regression (blue).
Left: $K = 1$, right: $K = 9$.

Linear Regression vs. K-Nearest Neighbors

Parametric vs. non-parametric

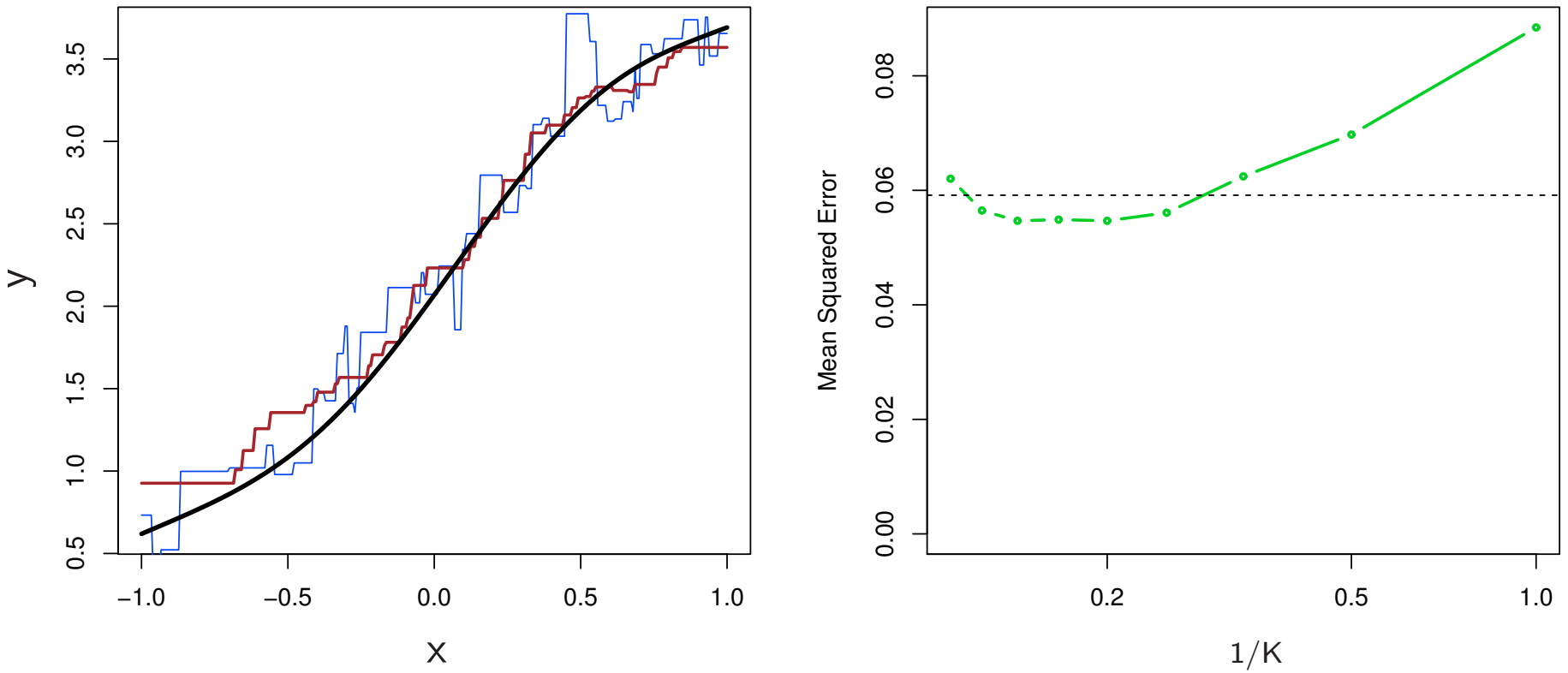


Left: same data, linear regression fit.

Right: test set MSE for linear regression (dotted line) and KNN for different values of K (plotted against $1/K$).

Linear Regression vs. K-Nearest Neighbors

Parametric vs. non-parametric

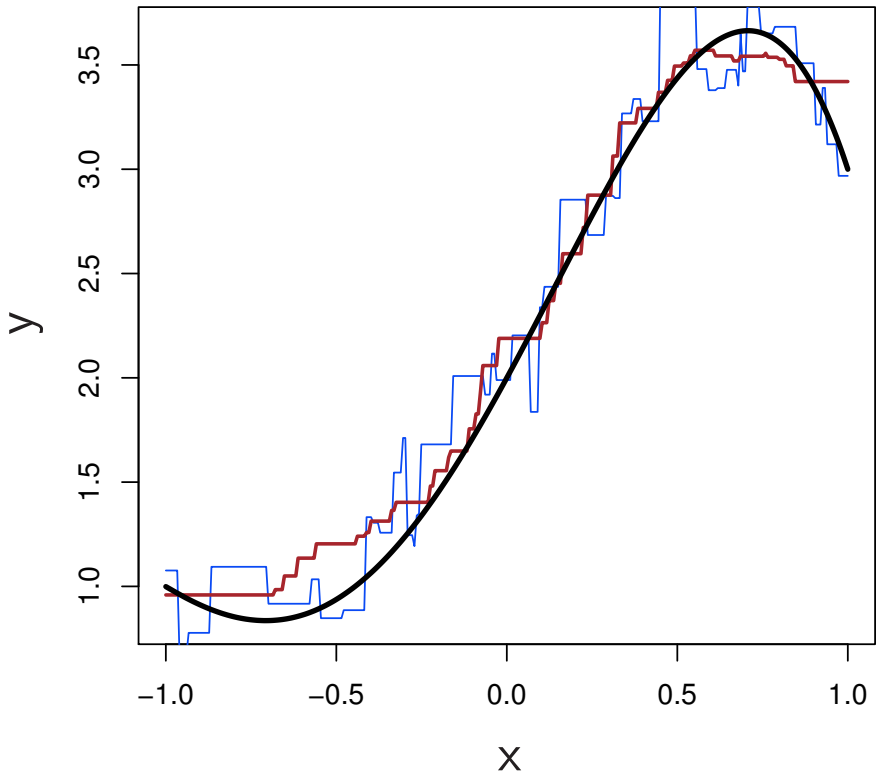


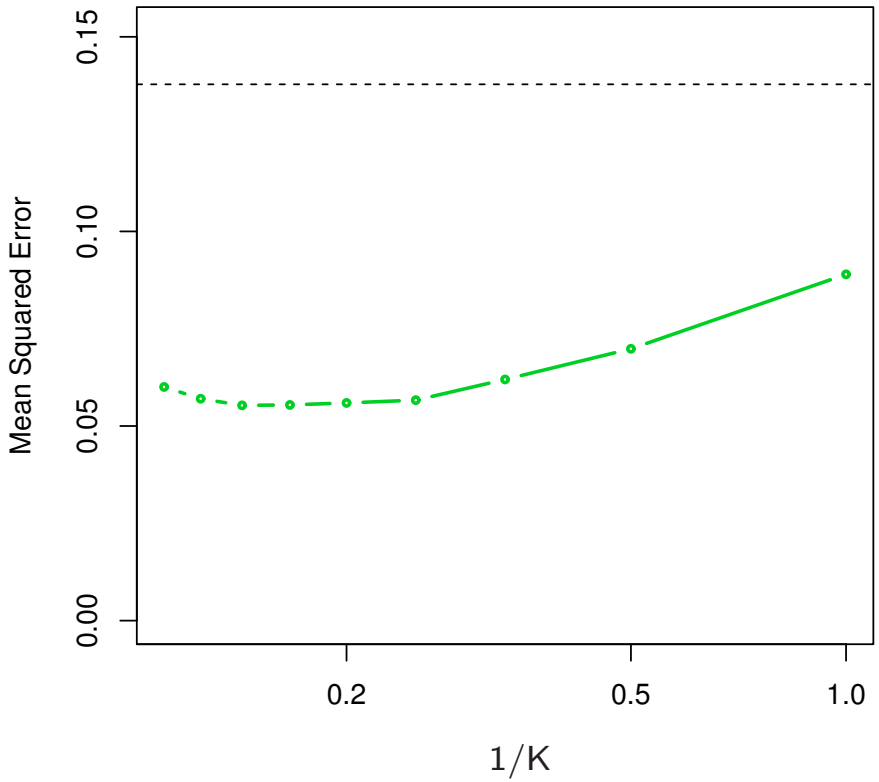
Left: slightly nonlinear data, true model (black), KNN regression with $K = 1$ (blue) and $K = 9$ (red).

Right: test set MSE for linear regression (dotted line) and KNN (against $1/K$). KNN wins for $K \geq 4$.

Linear Regression vs. K-Nearest Neighbors

Parametric vs. non-parametric



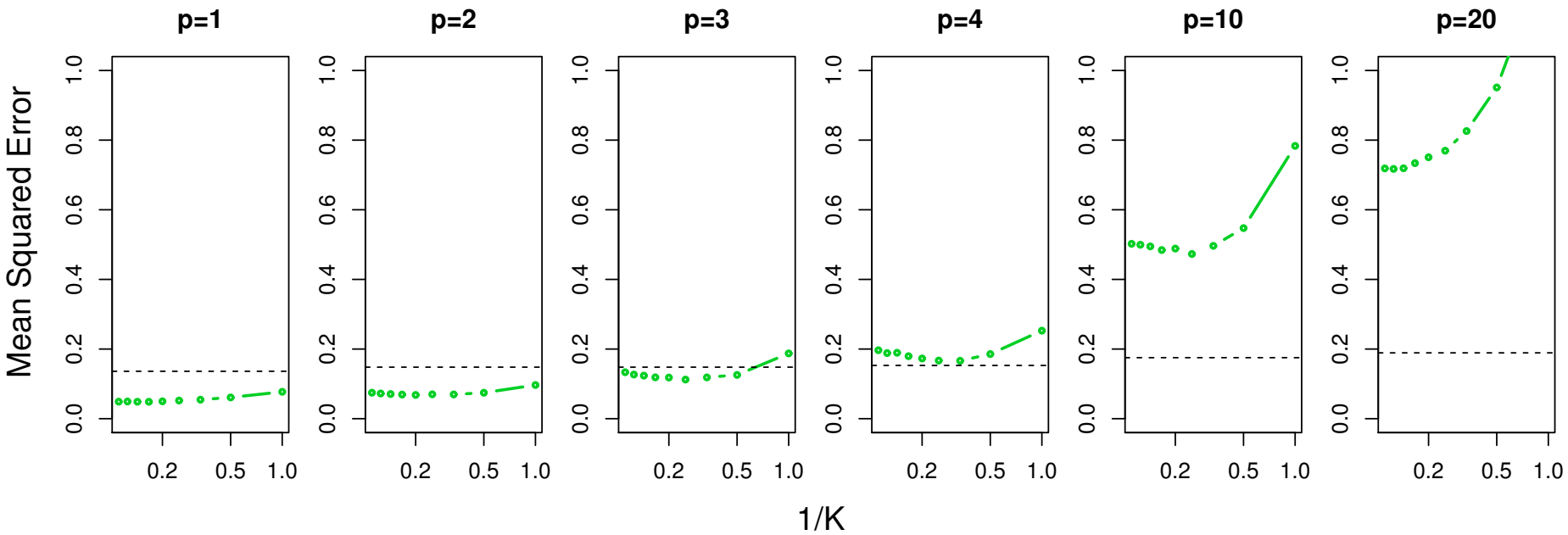


Left: stronly nonlinear data, true model (black), KNN regression with $K = 1$ (blue) and $K = 9$ (red).

Right: test set MSE for linear regression (dotted line) and KNN (against $1/K$). KNN wins for all K displayed.

Linear Regression vs. K-Nearest Neighbors

Parametric vs. non-parametric



Strongly nonlinear case, added noise predictors not associated with response. Linear regression MSE deteriorates only slightly as p rises, KNN regression MSE much more sensitive.

- For $p = 1$ KNN seems at most slightly worse than linear regression. For $p > 1$ this is no longer true.
- **Curse of dimensionality**: for $p = 20$, many of the 100 observations have no nearby observations.

Linear Regression vs. K-Nearest Neighbors

Parametric vs. non-parametric

- General rule: parametric methods tend to outperform non-parametric methods when there is a small number of observations per predictor.
- Even for small p, parametric methods offer the added advantage of better interpretability.

Contents

Classification

- **Classification**: response variable is **qualitative** or **categorical**.
- Involves assigning a predictor observation to a finite number of **classes** or **categories**.
- Likely more fundamental to human experience than regression.
  Examples: military triage, spam classification, fraud detection, tumor diagnostics, friend-foe distinction ...
- Common formulation: perform a linear regression, view (continuous) response result as probability of belonging to each class, choose class with larges probability.
- This chapter: 3 widely used classifiers:
  - **logistic regression**
  - **linear discriminant analysis** (LDA)
  - K-**nearest neighbors**

Contents

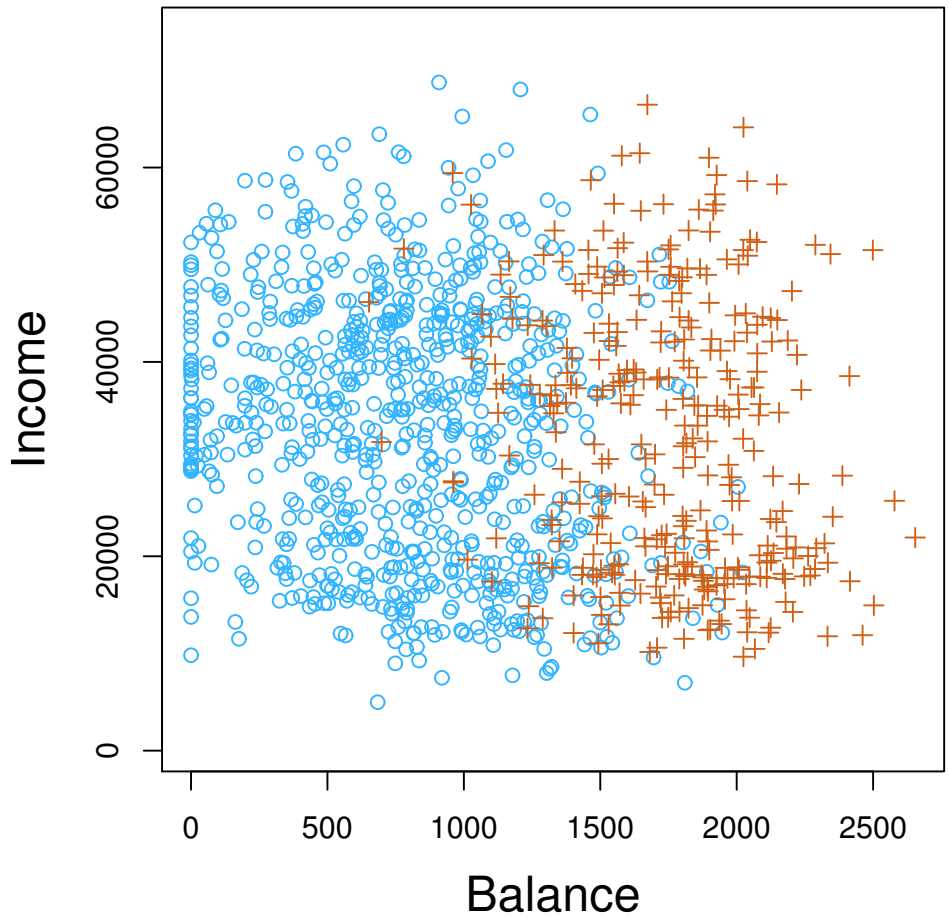
Overview of Classification

Setting

As for regression: use training observations $\{(x_i, y_i)_{i=1}^n\}$, to construct **classifier** able to perform classification also for test data not used in training.

`Default` data set: 10,000 individuals'

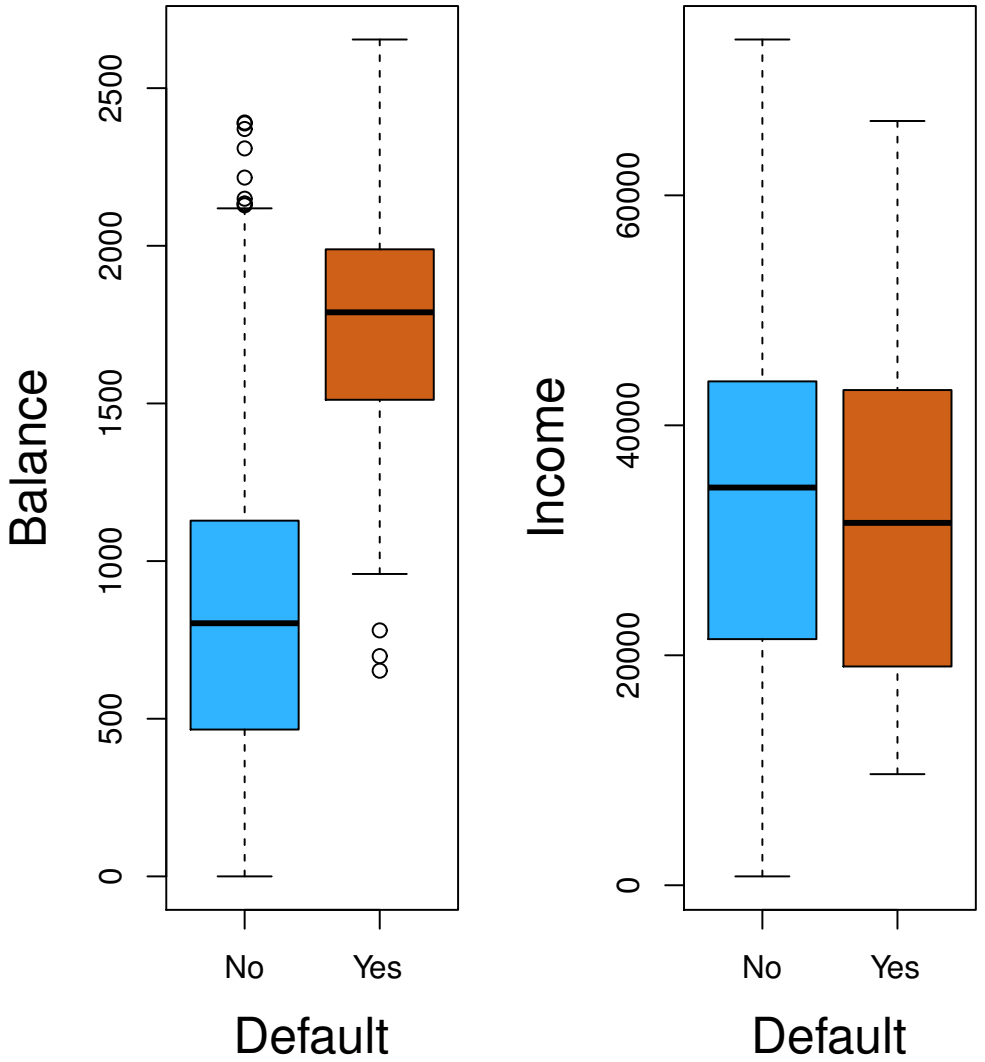
- annual `income` and
- monthly credit card `balance`.
- Response: binary `default` variable, i.e., whether or not person defaulted on their credit card payment in a given month.
- Overall default rate: 3%.



Overview of Classification

Setting

- **Box plots**: distributions of `balance` and `income` split by binary `default` variable.
- Objective: predict `default` (Y) for any pair of `balance` (X_1) and `income` (X_2) values.
- In this data: pronounced relationship between predictor `balance` and response `default`.



Contents

Why Not Linear Regression?

Ordering problem

Simplified model for predicting condition of incoming emergency room patients with possible diagnoses `stroke`, `drug overdose` or `epileptic seizure`.

Possible coding:

$$Y = \begin{cases} 1 & \text{if } \texttt{stroke}, \\ 2 & \text{if } \texttt{drug overdose}, \\ 3 & \text{if } \texttt{epileptic seizure}. \end{cases}$$

- Could perform linear regression based on available predictors $X_1, \ldots, X_p$.
- Coding implies (unnatural) ordering in outcome: places `drug overdose` between `stroke` and `epileptic seizure`.
- Also assumes distance between `stroke` and `drug overdose` is the same as between `drug overdose` and `epileptic seizure`.
- Different (equally reasonable) coding would lead to different linear model (and different predictions) for same data.

Why Not Linear Regression?

Ordering problem

- Sometimes underlying natural ordering exists (*mild*, *moderate*, *severe*).
- In general no way to map qualitative variable with > 2 values to quantitative response variable amenable to linear regression.
- For *binary* respone, e.g., only `stroke` and `drug overdose`, could use dummy variable approach and code

$$Y = \begin{cases} 0 & \text{if } \texttt{stroke}, \\ 1 & \text{if } \texttt{drug overdose}. \end{cases}$$

Following linear regression, could predict `drug overdose` if $\hat{Y} > 0.5$ and `stroke` otherwise.
Here flipping coding gives same results.
$X\hat{\beta}$ from linear regression yields estimate of probability

$$\mathbf{P}(\texttt{drug overdose}|X).$$

- For qualitative responses with > 2 values another approach is needed.

Contents

Logistic Regression

Idea

- `Default` data set, response variable `default` $\in \{$`Yes`, `No`$\}$.
- Logistic regression models *probability* of Y belonging to a particular class.
- Here: probability of default given `balance` denoted by

$$p(\texttt{balance}) := \mathbf{P}(\texttt{default} = \texttt{Yes} | \texttt{balance}) \in [0, 1].$$

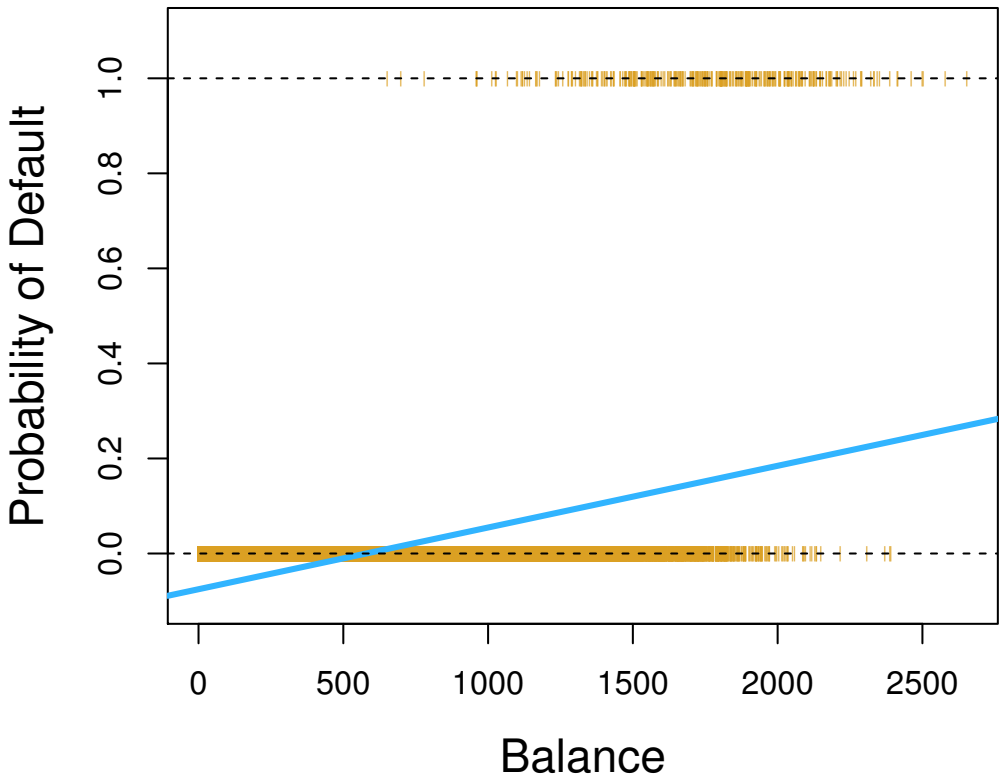
- Predict `default` $=$ `Yes` whenever, e.g., $p($`balance`$) > 0.5$.
- More conservative credit card company might prefer lower threshold, e.g., $p($`balance`$) > 0.1$.

- Predicting `default = Yes` by modeling relationship between $p(X) = \mathbf{P}(Y = 1|X)$ and X by linear regression

$$p(X) = \beta_0 + \beta_1 X \qquad (4.1)$$

gives fit on the right.

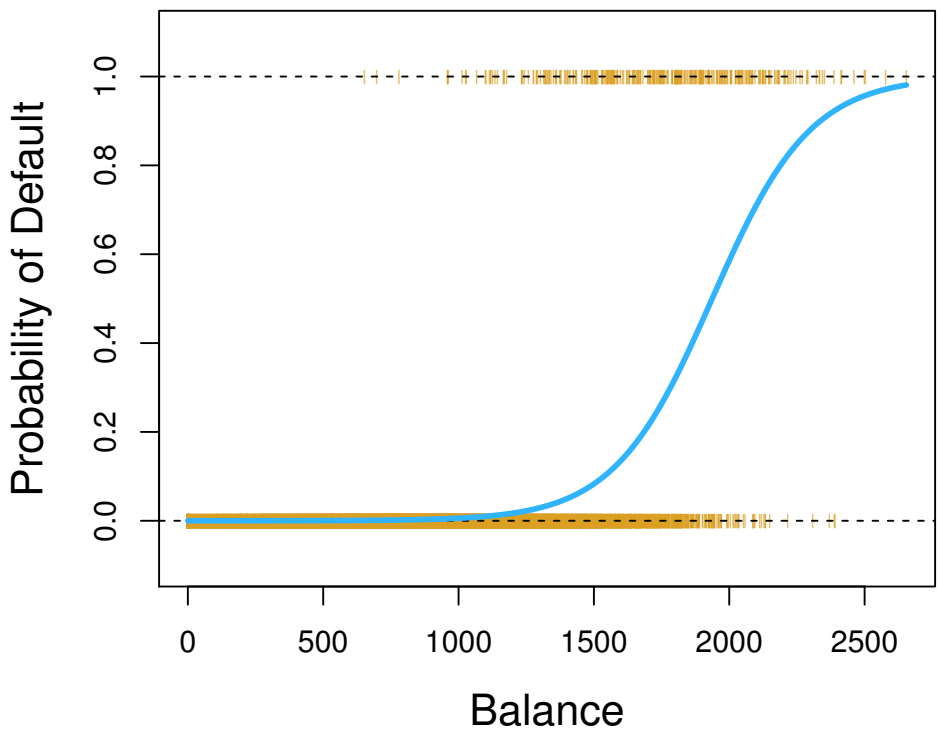
- Illustrates basic problem of fitting binary response coded with $\{0, 1\}$ with straight line: unless range of X limited, can always obtain probabilities outside $[0, 1]$.



Compose linear function with a **sigmoid** (monotone, S-shaped) function with values in $[0, 1]$, e.g., **logistic function**

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}. \tag{4.2}$$

- Fit for `Default` data on the right.
- Average default rate in both cases (linear and logistic) 0.0333, close to overall proportion in data set.



- Rearranging (4.2) gives

$$\frac{p(X)}{1-p(X)} = \mathrm{e}^{\beta_0+\beta_1 X} \tag{4.3}$$

- Ratio on left: **odds**, $\in [0, \infty]$.
- Example: if 1 in 5 people default, odds are 1/4; for 9 in 10, odds are 9.
- Popular horse-racing terminology, as reflects appropriate betting strategy.
- Take logarithms on both sides of (4.3):

$$\log \frac{p(X)}{1-p(X)} = \beta_0 + \beta_1 X. \tag{4.4}$$

Lhs: **log-odds** or **logit**. Logistic regression model (4.2) has logit which is linear in X.

Logistic Regression

Logistic model: parameter β_1

- β_1: in linear regression, gives average change in Y per unit change in X; in logistic regression, reflects change in log-odds per unit change in X.
- Unit change in X changes odds by factor e^{β_1}.
  Due to nonlinearity, β_1 does not correspond to change in $p(X)$ due to unit change in X.
- Amount $p(X)$ changes depends on value of X.
- $\beta_1 > 0$ implies monotone increase of $p(X)$ with X, decrease for $\beta_1 < 0$.

$$p(X) = \frac{\mathrm{e}^{\beta_0+\beta_1 X}}{1+\mathrm{e}^{\beta_0+\beta_1 X}}$$

- **Maximum-likelihood estimation** (MLE) to determine estimates $\hat{\beta}_0, \hat{\beta}_1$ of coefficients β_0, β_1.
- **Likelihood function**

$$\ell(\beta_0, \beta_1) := \prod_{y_i=1} p(x_i) \cdot \prod_{y_i=0} (1 - p(x_i)), \tag{4.5}$$

  $p(x_i)$ determined from the observations (frequency).
- Estimates $\hat{\beta}_0, \hat{\beta}_1$ determined as $(\hat{\beta}_0, \hat{\beta}_1) := \arg\max \ell(\beta_0, \beta_1)$.
  This is a problem of numerical optimization methods, plenty of software available.
- Least squares can be viewed as a special case of MLE.

Logistic Regression

Estimating the regression coefficients

Coefficient estimates and statistics for logistic regression model on `Default` data set for predicting $\mathbf{P}(\texttt{default} = \texttt{Yes})$ with predictor `balance`:

	Coefficient	Standard error	z-statistic	p-value
β_0	-10.6513	0.3612	-29.5	< 0.0001
β_1	0.0055	0.0002	24.9	<0.0001

- Estimation accuracy measured by standard errors.
- z-statistic : analogous role here as t statistic in simple linear regression. For coefficient β_1:

$$z = \frac{\hat{\beta}_1}{\mathrm{SE}(\hat{\beta}_1)}$$

- p-values strong evidence against $H_0 : \beta_1 = 0$, implying $p(X) = \mathrm{e}^{\beta_0}/(1+\mathrm{e}^{\beta_0})$.
- Intercept β_0 not of interest.

Logistic Regression

Making predictions: predictor `balance`

- Given this logistic regression model for `default` on `balance`, what probability for defaulting on payment can we predict for an individual with a balance of \$1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} \approx 0.00576 \approx 0.5\%.$$

- What about a balance of \$2000? Here

$$\hat{p}(X) \approx 0.5863 \approx 58\%.$$

Logistic Regression

Making predictions: predictor `student`

- For qualitative predictor variables, e.g., `student` in `Default` data set, use dummy variable taking value 1 for students, 0 for non-students.
- Resulting model: logistic regression of `default` on `student` status

	Coefficient	Standard error	z-statistic	p-value
β_0	-3.5041	0.0707	-49.55	< 0.0001
β_1	0.4049	0.1150	3.52	0.0004

- $\beta_1 > 0$, statistically significant.
- Model predicts higher default probability for students:

$$\hat{\mathbf{P}}(\texttt{default} = \texttt{Yes} | \texttt{student} = \texttt{Yes}) = \frac{\mathrm{e}^{\hat{\beta}_0 + \hat{\beta}_1 \cdot 1}}{1 + \mathrm{e}^{\hat{\beta}_0 + \hat{\beta}_1 \cdot 1}} \approx 0.0431,$$

$$\hat{\mathbf{P}}(\texttt{default} = \texttt{Yes} | \texttt{student} = \texttt{No}) = \frac{\mathrm{e}^{\hat{\beta}_0 + \hat{\beta}_1 \cdot 0}}{1 + \mathrm{e}^{\hat{\beta}_0 + \hat{\beta}_1 \cdot 0}} \approx 0.0292.$$

- For multiple predictor variables $X = (X_1, \ldots, X_p)$, generalize (4.4) to

$$\log \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p, \tag{4.6}$$

or

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}, \tag{4.7}$$

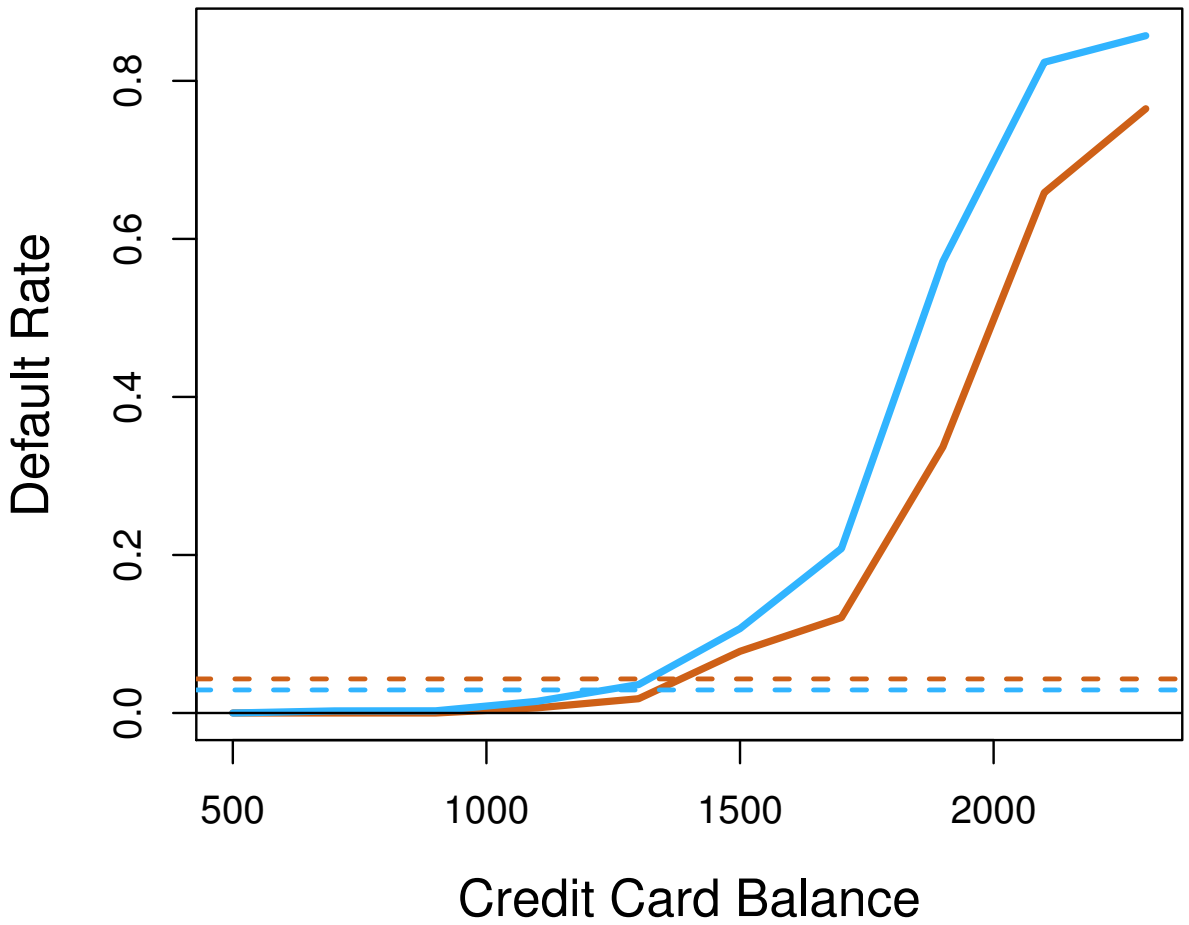
- Fit parameters again by MLE.
- Logistic regression predicting `default` based on `balance`, `income` and `student` status:

	Coefficient	Standard error	z-statistic	p-value
β_0	-10.8690	0.4923	-22.08	< 0.0001
β_1 (`balance`)	0.0057	0.0002	24.74	< 0.0001
β_2 (`income`)	0.0030	0.0082	0.37	0.7115
β_3 (`student`)	-0.6468	0.2362	-2.74	0.0062

Logistic Regression

Multiple logistic regression

- Coefficients of `balance` and `student` significant.
- Coefficient of `student` now negative! Explanation?



Orange: student
Blue: non-student

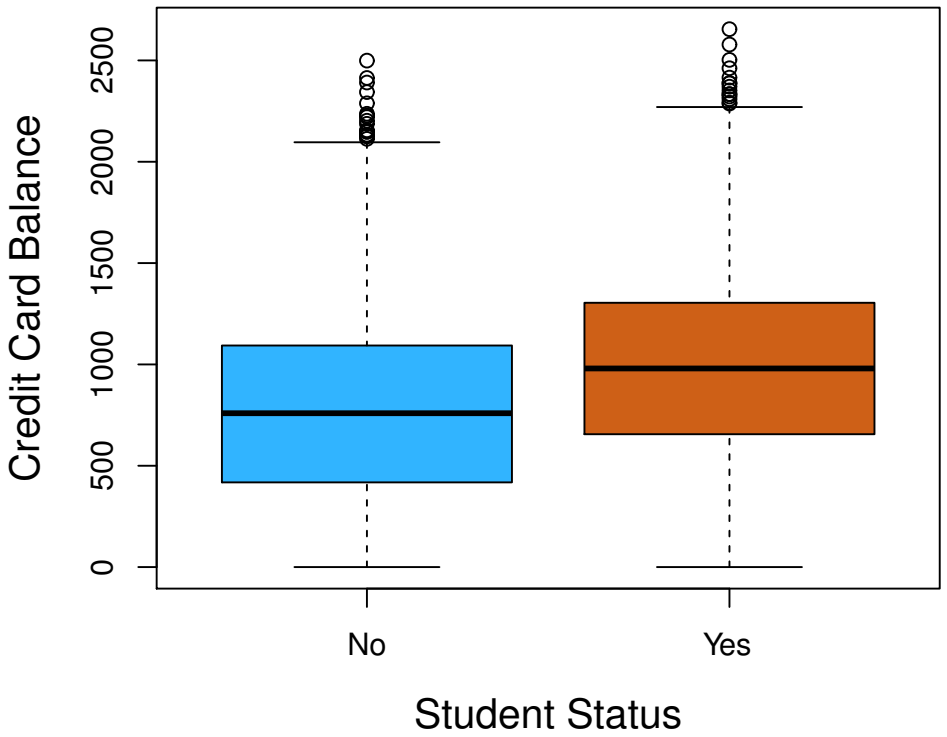
Logistic Regression

Multiple logistic regression

- Negative coefficient of `student`: for fixed value of `balance` and `income`, student *less* likely to default than non-student.
- Figure shows: student default rate at or below non-student rate for each value of `balance`.
- Horizontal broken lines: overall student default rate higher than non-student. Explains positive coefficient for `student` in single variable logistic regression.

Logistic Regression

Multiple logistic regression



- Variables `student` and `balance` correlated.
- Students tend to hold higher debt level, hence higher probability of default.
- Individual student with given balance will have lower default probability than non-student with same balance.

- Overall: student riskier than non-student.
- But: student less risky than non-student with same balance.
- Illustrates subtleties of ignoring further relevant predictors.
- Phenomenon: **confounding**.

Logistic Regression

Multiple logistic regression: example predictions

- Student with credit card balance of \$1 500, income of \$40 000 has estimated probability of default

$$\hat{p}(X) = \frac{\mathrm{e}^{\hat{\beta}_0+\hat{\beta}_1\cdot 1500+\hat{\beta}_2\cdot 40+\hat{\beta}_3\cdot 1}}{1+\mathrm{e}^{\hat{\beta}_0+\hat{\beta}_1\cdot 1500+\hat{\beta}_2\cdot 40+\hat{\beta}_3\cdot 1}} \approx 0.0549.$$

- For non-student, same credit card balance and income, estimate is

$$\hat{p}(X) = \frac{\mathrm{e}^{\hat{\beta}_0+\hat{\beta}_1\cdot 1500+\hat{\beta}_2\cdot 40+\hat{\beta}_3\cdot 0}}{1+\mathrm{e}^{\hat{\beta}_0+\hat{\beta}_1\cdot 1500+\hat{\beta}_2\cdot 40+\hat{\beta}_3\cdot 0}} \approx 0.1054.$$

Note: model fit was performed with units of \$1 000 for variable `income`.

Logistic Regression

Logistic regression for several response classes

- Recall emergency room example with 3 response classes `stroke`, `drug overdose` and `epileptic seizure`.
- Would like to model

$$\begin{aligned}
&\mathbf{P}(Y = \texttt{stroke}|X),\\
&\mathbf{P}(Y = \texttt{drug overdose}|X),\\
&\mathbf{P}(Y = \texttt{epileptic seizure}|X)\\
&\qquad = 1 - \mathbf{P}(Y = \texttt{stroke}|X) - \mathbf{P}(Y = \texttt{drug overdose}|X).
\end{aligned}$$

- Can extend two-class logistic regression to more than two, software available, but LDA more popular for this case.

Contents

Linear Discriminant Analysis

Conditional distribution

- Recall (ideal) Bayes classifier: assign to x_0 class $k \in \{1, \ldots, K\}$ such that

$$\hat{y}_0 = \hat{f}(x_0) = \underset{1 \le k \le K}{\arg\max}\, \mathbf{P}(Y = k | X = x_0).$$

- Logistic regression: model $\mathbf{P}(Y = k | X = x_0)$ using logistic function (4.7) when $K = 2$.
- Alternative approach LDA: model distribution of *predictors* X, then use **Bayes' rule** to turn these into estimates for $\mathbf{P}(Y = k | X = x_0)$.
- Motivation:
  - Logistic regression often unstable even for well-separated classes.
  - For small n, predictors approximately Gaussian across classes, LDA more stable than logistic regression.
  - LDA popular for $K > 2$.

Given **probability space** $(\Omega, \mathfrak{A}, \mathbf{P})$, $A, B \in \mathfrak{A}$, $\mathbf{P}(B) > 0$, then the **conditional probability** of A given B is defined by

$$\mathbf{P}(A|B) := \frac{\mathbf{P}(A \cap B)}{\mathbf{P}(B)}.$$

Solving for $\mathbf{P}(A \cap B)$, exchanging roles of A and B, assuming $\mathbf{P}(A) > 0$, gives

$$\mathbf{P}(A|B) = \frac{\mathbf{P}(B|A)\,\mathbf{P}(A)}{\mathbf{P}(B)} \qquad \textbf{Bayes' rule} \quad [\text{Bayes, 1763}]$$

- A: unobservable state of nature, with **prior probability** $\mathbf{P}(A)$ of occurring;
- B: observable event, probability $\mathbf{P}(B)$ known as **evidence**;
- $\mathbf{P}(B|A)$: probability that A causes B to occur (**likelihood**);
- $\mathbf{P}(A|B)$: **posterior probability** of A knowing that B has occurred.
- Terms: **inverse probability**, **Bayesian inference**.

Linear Discriminant Analysis

Bayes' rule (partitions)

Given partition $\{A_j\}_{j\in\mathbb{N}}$ of Ω into exhaustive and exclusive disjoint events, de Morgan's rule and countable additivity give, assuming all $\mathbf{P}(A_j) > 0$,

$$\mathbf{P}(B) = \sum_{j\in\mathbb{N}} \mathbf{P}(B|A_j)\,\mathbf{P}(A_j) \qquad \text{(law of total probability)},$$

leading to another variant of Bayes' rule:

$$\mathbf{P}(A_k|B) = \frac{\mathbf{P}(B|A_k)\,\mathbf{P}(A_k)}{\sum_{j\in\mathbb{N}} \mathbf{P}(B|A_j)\,\mathbf{P}(A_j)},$$

giving posterior probability of each A_k after observing B.

Linear Discriminant Analysis

Bayes' rule (densities)

Given real-valued **random variables** X, Y with **probability density functions** (pdfs)

- $f_X(x)$, $f_Y(y)$: density of X, Y at value x, y,
- $f_{X|Y}(x|y)$: density of $(X|Y)$ at x having observed $Y = y$,
- $f_{Y|X}(y|x)$: analogously.

Then Bayes' theorem states that

$$f_{X|Y}(x|y) = \frac{f_{Y|X}(y|x)\, f_X(x)}{f_Y(y)} = \frac{f_{Y|X}(y|x)\, f_X(x)}{\int f_{Y|X}(y|x)\, f_X(x)\, \mathrm{d}x}.$$

- $f_{Y|X}(y|x)$ is now called the **likelihood function**.
- $\int f_{Y|X}(y|x)\, f_X(x)\, \mathrm{d}x$ is calles the **normalizing factor** or **marginal**.
- Short form:

$$f_{X|Y} \propto f_{Y|X}\, f_X.$$

Linear Discriminant Analysis

Using Bayes' rule for classification

- Goal: classify observation into one of $K \geq 2$ classes.
- $\pi_k := \mathbf{P}(Y(X) = k)$, $1 \leq k \leq K$, for randomly chosen X: **prior** probability.
- $f_k(x) := \mathbf{P}(X = x | Y = k)$, $1 \leq k \leq K$, **density function**[5] of X in class k. In other words: $f_k(x)$ large if probability that $X = x$ is large in class k.
- Bayes' rule:

$$p_k(x) := \mathbf{P}(Y = k | X = x) = \frac{\pi_k f_k(x)}{\sum_{j=1}^{K} \pi_j f_j(x)}. \tag{4.8}$$

- Idea: instead of computing $p_k(X)$ directly, estimate $f_k(X)$ and π_k, $k = 1, \ldots, K$ and insert into (4.8).
- If all estimates accurate, should come close to Bayes classifier.

[5] Modify accordingly for non-discrete predictor variable.

Linear Discriminant Analysis

LDA, $p = 1$

- Assumption: assume single predictor X has Gaussian distribution in each class, i.e.,

$$f_k(x) = \frac{1}{\sigma_k\sqrt{2\pi}} \exp \frac{-(x-\mu_k)^2}{2\sigma_k^2}, \qquad k = 1, \ldots, K.$$

- Assume further that $\sigma_1 = \cdots = \sigma_K = \sigma$.
- Insert into (4.8):

$$p_k(x) = \frac{\pi_k \exp \frac{-(x-\mu_k)^2}{2\sigma^2}}{\sum_{j=1}^K \pi_j \exp \frac{-(x-\mu_j)^2}{2\sigma^2}} \tag{4.9}$$

- Classification: assign x to class k for which (4.9) is largest.
- Equivalent: class k for which

$$\delta_k(x) := \frac{x\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k$$

is largest.

Linear Discriminant Analysis

LDA, $p = 1$

Example:

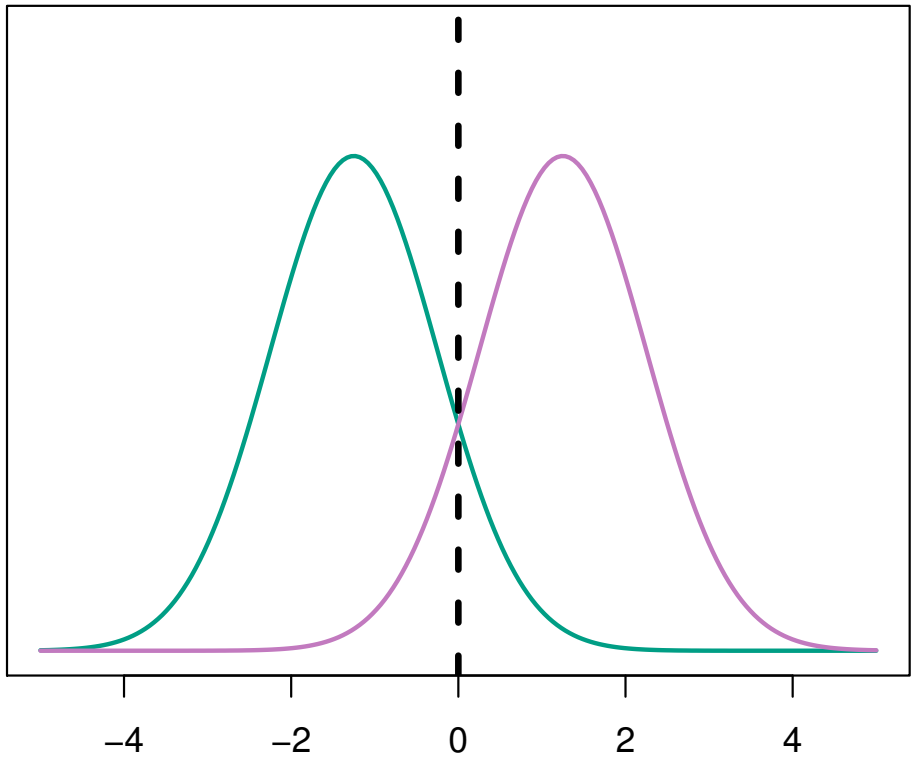
- $K = 2$, $\pi_1 = \pi_2$, assign x to class 1 if $\delta_1(x) > \delta_2(x)$ or

$$2x(\mu_1 - \mu_2) > \mu_1^2 - \mu_2^2.$$

- Bayes decision boundary at

$$x = \frac{\mu_1 + \mu_2}{2}.$$

- In this case we can compute the Bayes classifier.



Two univariate normal densities with $\sigma_1 = \sigma_2 = 1$ and $\mu_1 = -\mu_2 = 1.25$, Bayes decision boundary (dashed black line).

Linear Discriminant Analysis

LDA, $p = 1$, estimating mean and variances

- LDA uses estimates for (ususally unknown) mean and variance:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{y_i = k} x_i, \qquad \hat{\sigma} = \frac{1}{n - K} \sum_{k=1}^{K} \sum_{y_i = k} (x_i - \hat{\mu}_k)^2, \tag{4.10}$$

  N: total # observations, n_k: # observations in class k.

- Prior probabilities estimated as

$$\hat{\pi}_k = \frac{n_k}{n}. \tag{4.11}$$

- Classifier now assigns new observation to class k such that

$$k = \underset{1 \le k \le K}{\arg\max}\, \hat{\delta}_k(x), \qquad \hat{\delta}_k(x) := x \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log \hat{\pi}_k. \tag{4.12}$$

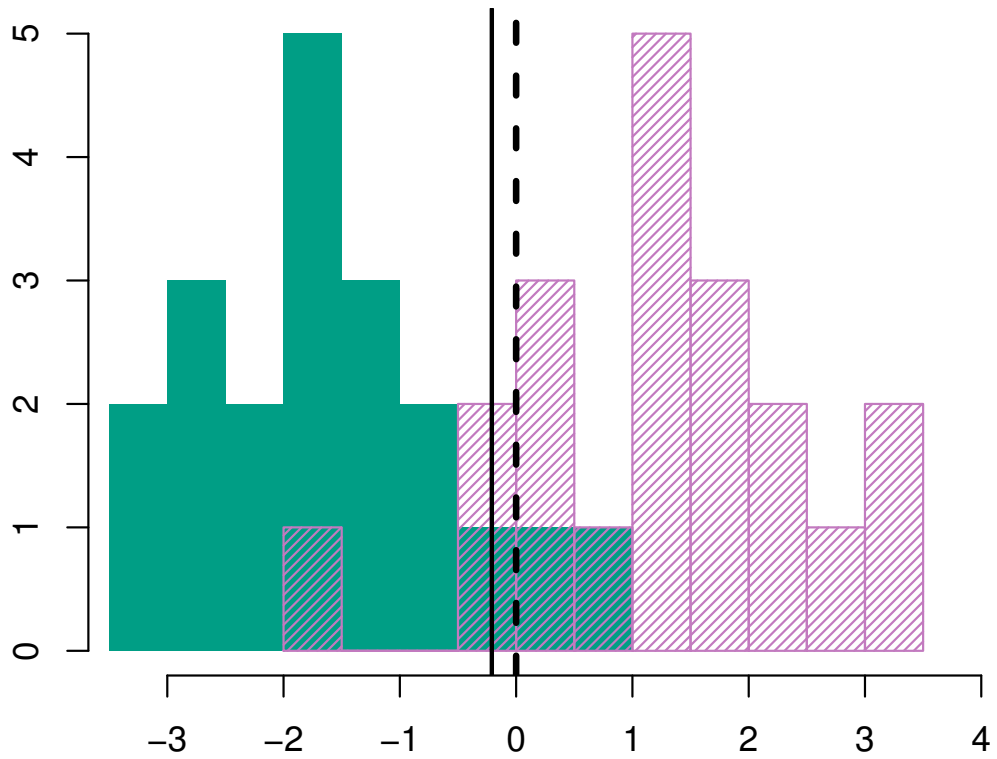
- LDA: **discriminant functions** $\hat{\delta}_k(x)$ are *linear* in x.

Linear Discriminant Analysis

LDA, $p = 1$, example

Example: (right)

- $K = 2$, $n = 20$ random observations from each class, estimate σ, μ_k, π_k.
- Bayes decision boundary given by solid black line; observations to the left assigned to green, otherwise purple.
- $n_1 = n_2 = 20 \Rightarrow \hat{\pi}_1 = \hat{\pi}_2$ and decision boundary at $(\hat{\mu}_1 + \hat{\mu}_2)/2$, slightly to left of Bayes decision boundary at $(\mu_1 + \mu_2)/2 = 0$.
- Bayes error rate: 10.6%, LDA test error rate: 11.1 %, i.e., only 0.5% over optimal!



Simulated data from 2 classes, LDA / Bayes decision boundaries.

Linear Discriminant Analysis

LDA, $p = 1$

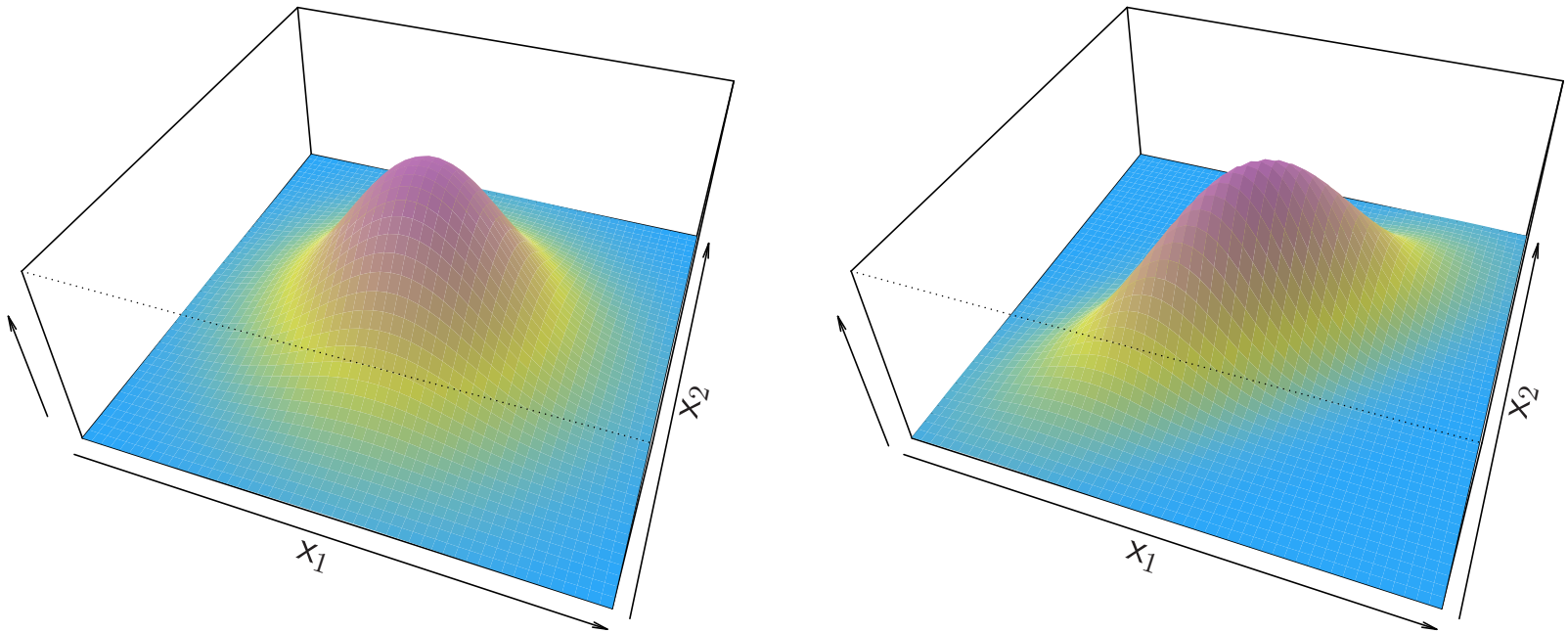
Recap: LDA classifier

- assumes observations within each class follow normal distribution,
- class-specific mean, common variance σ^2,
- estimates lead to classifier (4.12).

Linear Discriminant Analysis

LDA, $p > 1$

- For multiple predictor variables $X = (X_1, \ldots, X_p)$, assume observations follow multivariate normal distributions with class-specific mean, common covariance matrix.



Probability density functions (pdf) of two bivariate ($p = 2$) Gaussian distributions. Left: uncorrelated, right: correlation 0.7.

Linear Discriminant Analysis

LDA, $p > 1$

- Multivariate Gaussian:

$$X \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \boldsymbol{\mu} = \mathbf{E}[X] \in \mathbb{R}^p, \quad \boldsymbol{\Sigma} = \mathbf{Cov}(X) \in \mathbb{R}^{p \times p}.$$

- Pdf:

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^p \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{4.13}$$

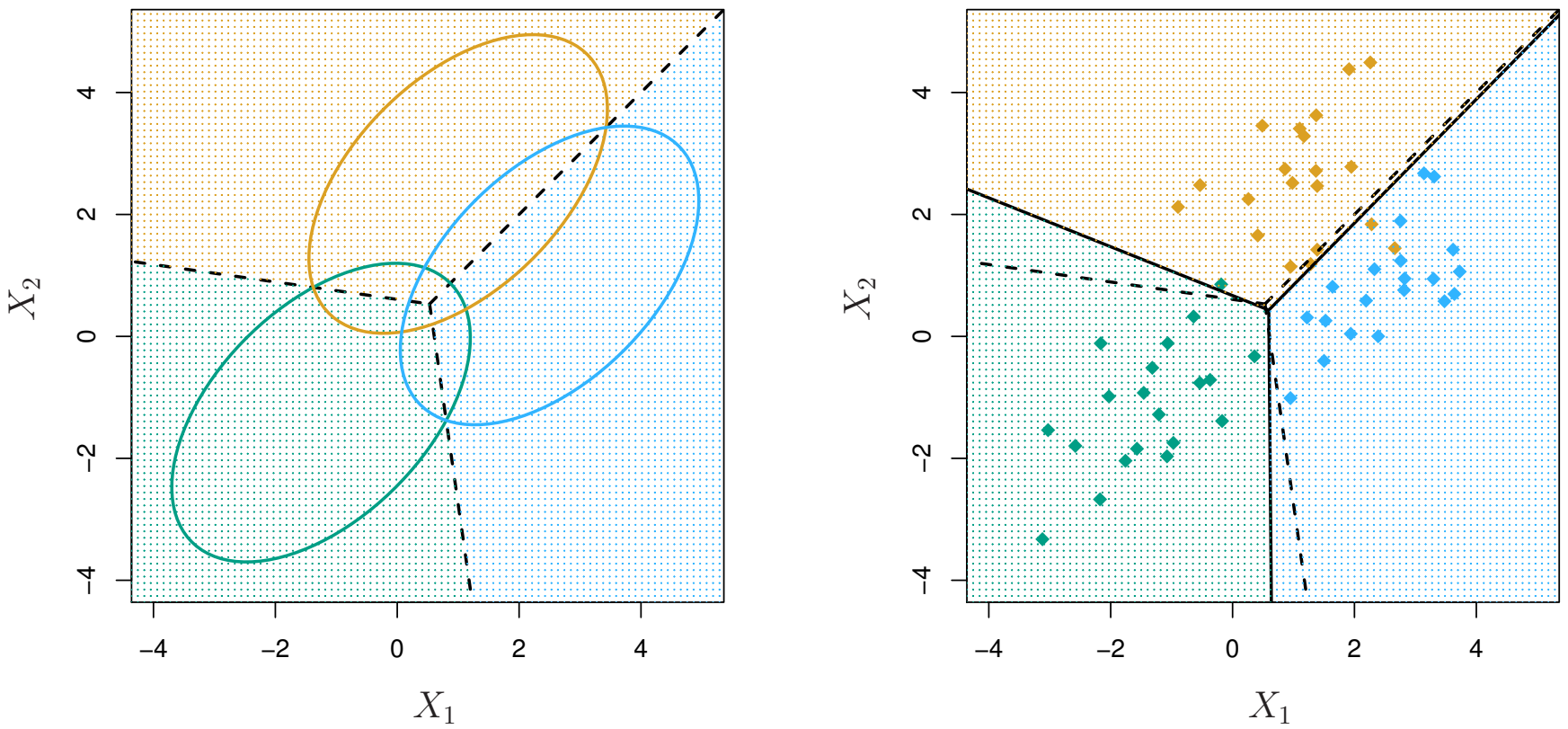
- For $p > 1$ assume within each class k: $X \sim \mathrm{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$.
- Inserting pdf f_k into (4.8), we obtain Bayes classifier assigning observation $\boldsymbol{x}$ to class

$$k = \underset{1 \le k \le K}{\arg\max}\ \delta_k(\boldsymbol{x}), \qquad \delta_k(\boldsymbol{x}) := \boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \log \pi_k. \tag{4.14}$$

This is the vector version of (4.12).

Linear Discriminant Analysis

LDA, $p > 1$, example



$p = 2$, $K = 3$, samples from three Gaussian distributions with means $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\boldsymbol{\mu}_3$, common covariance matrix.

Left: 95%-ellipses, Bayes decision boundaries dashed.

Right: $n = 20$ random samples drawn from each class, their LDA classifications, Bayes decision boundary dashed, LDA decision boundary solid.

Linear Discriminant Analysis

LDA, $p > 1$, example

- Bayes decision boundaries: $\delta_j(\boldsymbol{x}) = \delta_k(\boldsymbol{x}), \quad j, k = 1, 2, 3,\ j < k.$
- LDA decision boundaries: $\hat{\delta}_j(\boldsymbol{x}) = \hat{\delta}_k(\boldsymbol{x}), \quad j, k = 1, 2, 3,\ j < k.$
- Unknown parameters

$$\{\pi_k\}_{k=1}^K, \quad \{\boldsymbol{\mu}_k\}_{k=1}^K, \quad \boldsymbol{\Sigma}$$

  estimated using formulas analogous to $p = 1$ case.
- Test error rates:
  Bayes: 0.0746
  LDA: 0.0770
- Again, conditional probability $\delta_k(\boldsymbol{x})$ in (4.14) *linear* in $\boldsymbol{x}$.

Linear Discriminant Analysis

LDA applied to `Default` data set

- Predict probability of defaulting on credit card payments given `balance` and `student` status.
- LDA model fit to $n = 10,000$ training samples results in *training* error rate of 2.75%. Low?
- Caveats:
  - Training error rates generally lower than test error rates.
  - High ratio of free parameters p to n poses overfitting danger, but here $p = 2$, $n = 10,1000$.
  - Overall, true default rate in `Default` training data only 3.33%.
    Implies (useless) constant classifier $Y \equiv 0$ has this low error rate.

Linear Discriminant Analysis

LDA applied to `Default` data set

Confusion matrix for LDA applied to `Default`:

		True default status		
		No	Yes	Total
Predicted default status	No	9,644	252	9,896
	Yes	23	81	104
	Total	9,667	333	10,000

- Two types of misclassification errors.
- LDA: predicts 104 of 10,000 will default; of those, only 81 really defaulted. Hence, only 23 of 9,667 incorrectly labelled.
- However: of 333 who really defaulted, 252 (75.7%) missed by LDA.
- For credit card company trying to identify high-risk individuals: this false negative error rate probably unacceptable.

Linear Discriminant Analysis

Binary misclassification errors

- **Class-specific** classification errors can be crucial.
- In screening procedures (medical, airport):
  sensitivity: ratio of true positives identified;
  specificity: ratio of true negatives identified.
- In `Default` example:
  sensitivity $= 81/333 \approx 24.3\%$;
  specificity $= 9\,644/9\,667 \approx 99.8\%$.
- In hypothesis testing:
  type-I error: rejection of true null hypothesis (**false positive** finding);
  type-II error: failing to reject false null hypothesis (**false negative** finding).

Linear Discriminant Analysis

Example: mammography screening

What is the probability that a woman has breast cancer given (only) a positive result after undergoing a mammography screening?

Data on breast cancer screening test: [Kerlikowske & al., 1996]

Prevalence	1% (proportion of women who have breast cancer)
Sensitivity	90%
Specificity	91%

Bayes' rule: $Y \in \{0, 1\}$ (cancer?), $X \in \{0, 1\}$ (test positive?)

$$\mathbf{P}(Y = 1|X = 1) = \frac{\mathbf{P}(X = 1|Y = 1) \cdot \mathbf{P}(Y = 1)}{\mathbf{P}(X = 1|Y = 1) \cdot \mathbf{P}(Y = 1) + \mathbf{P}(X = 1|Y = 0) \cdot \mathbf{P}(Y = 0)}$$

$$= \frac{0.9 \cdot 0.01}{0.9 \cdot 0.01 + (1 - 0.91)(1 - 0.01)} \approx 9.2\%.$$

Do medical professionals know this? Don't count on it!
[Hoffrage & Gigerenzer, 1998]

Linear Discriminant Analysis

Modified LDA

- LDA approximates Bayes classifier, which has lowest *overall* error rate.
- However, sometimes important to achieve low error within a particular class of interest (credit card company, interested in defaulting customers).
- Bayes classifier: assign observation x to class k for which $p_k(x)$ largest.
  In two-class case of `Default` data set: assign to `default` class if

$$\mathbf{P}(\texttt{default} = \texttt{Yes} | X = x) > 0.5.$$

- To increase sensitivity to default, instead use lower threshold of

$$\mathbf{P}(\texttt{default} = \texttt{Yes} | X = x) > 0.2.$$

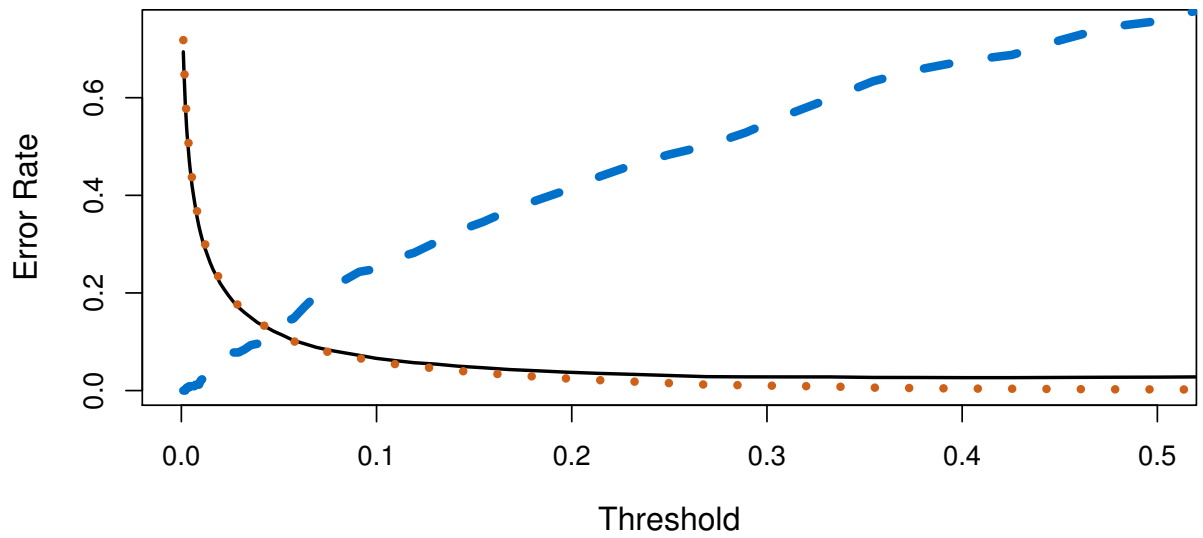
  Modifies confusion table as follows: (cf. Slide 201)

		True default status		
		No	Yes	Total
Predicted	No	9,432	138	9,570
default status	Yes	235	195	430
	Total	9,667	333	10,000

Linear Discriminant Analysis

Modified LDA

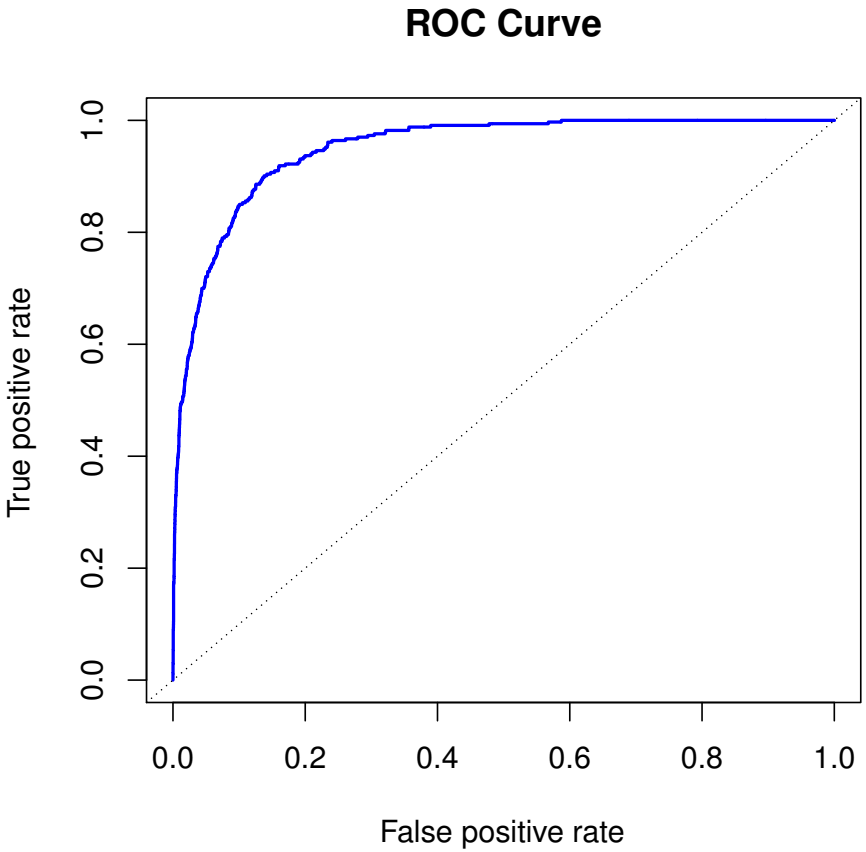
- LDA default prediction increases from 104 to 430. Default prediction error rate improves from $252/333 \approx 75.7\%$ to $138/333 \approx 41.4\%$.
- However, now 235 individuals who did not default are misclassified, raising the classification error in this class from $23/9,667 \approx 0.24\%$ to $235/9,667 \approx 2.4\%$, with an overall classification error of $(138 + 235)/10,000 = 3.73\%$.



`Default` data set: error rates versus threshold for LDA-assignment into defaulting class: black: overall, blue: fraction of defaulting customers misclassified; red: misclassified non-defaulting customers.

Linear Discriminant Analysis

ROC curve



- Traces out false positive/true positive rate for all threshold values of LDA classifier in `Default` data set.
- True positive: sensitivity (ratio defaulters correctly classified)
- False positive: 1−specificity (ratio of non-defaulters incorrectly classified).
- Optimal ROC curve: follows left/top boundaries.
- Dotted line: "no-information classifier", i.e., if student status and credit card balance unrelated to default.

- **Receiver Operating Characteristics** (ROC): simultaneous plot of both error types for all possible thresholds.
- **Area under the ROC curve** (AUC): overall performance of classifier summarized over all threshold values. Here AUC= 0.95 close to optimum 1.

Linear Discriminant Analysis

Summary of terminology

Possible results when applying a classifier (diagnostic test) to a population:

		Predicted class		
		− or Null	+ or Non-null	Total
True class	− or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

- Epidemiology context:
  +: disease, −: non-disease.
- Hypothesis testing context:
  −: null hypothesis, +: alternative (non-null) hypothesis.
- `Default` data set context:
  +: defaulting customer, −: non-defaulting customer.

Linear Discriminant Analysis

Performance measures for binary classification

Name	Definition	Synonyms
False Pos. rate	FP/N	Type-I error, 1− specificity
True. Pos. rate	TP/P	1− Type-II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision, 1− false discovery proportion
Neg. Pred. value	TN/N*	

N: population negative
P: population positive
N*: predicted negative
P*: predicted positive

- **Quadratic discriminant analysis** (QDA): assume observations within each class follow Gaussian distribution, but each class has distinct covariance matrix, i.e., observation in k-th class given by random variable

$$X \sim \mathrm{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

- Assign observation $X = \boldsymbol{x}$ to class k which maximizes discriminant

$$\begin{aligned}\delta_k(\boldsymbol{x}) &= -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) - \frac{1}{2}\log\det\boldsymbol{\Sigma}_k + \log\pi_k \\ &= -\frac{1}{2}\boldsymbol{x}^\top\boldsymbol{\Sigma}_k^{-1}\boldsymbol{x} + \boldsymbol{x}^\top\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\log\det\boldsymbol{\Sigma}_k + \log\pi_k. \end{aligned} \tag{4.15}$$

- Now discriminants depend *quadratically* on observation $\boldsymbol{x}$.

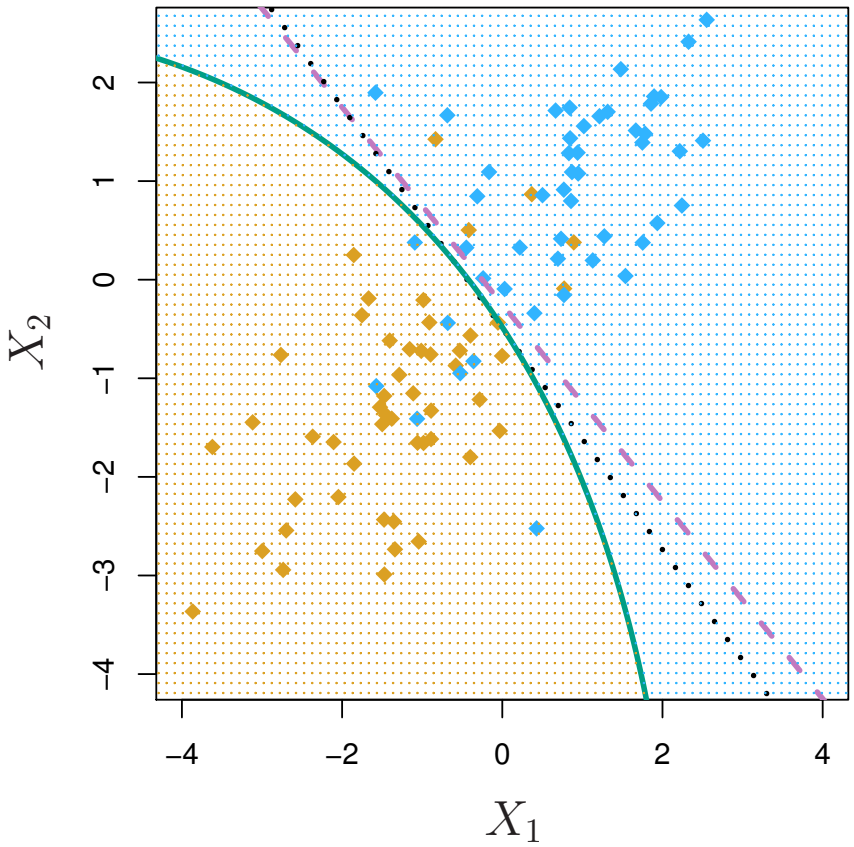
Linear Discriminant Analysis

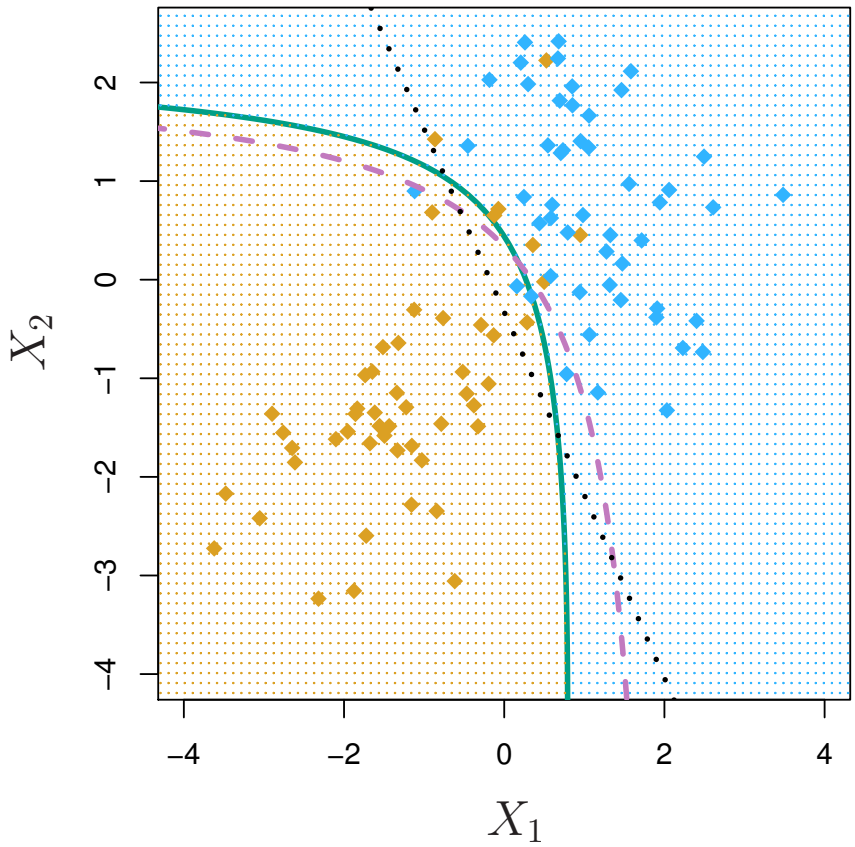
Quadratic discriminant analysis

- Requires estimation of π_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$.
- Possible advantage of QDA over LDA: bias-variance trade-off.
- LDA estimates single covariance matrix: $p(p+1)/2$ parameters.
  QDA estimates K covariance matrices: $Kp(p+1)/2$ parameters.
- For 50 predictors this amounts to $K \cdot 1275$ parameters.
- LDA: larger bias, use for few training observations;
  QDA: larger variance, use for many training observations or when common covariance matrix known to be false.

Linear Discriminant Analysis

Example: LDA vs. QDA





Two-class problem, decision boundaries: Bayes (purple dashed), LDA (black dotted) and QDA (green solid). Shading: QDA classification. Left:$\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2$. Right:$\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2$

Contents

LDA vs. logistic regression: consider $p = 1$, $K = 2$.

- $p_1(x)$, $p_2(x) = 1 - p_1(x)$: probability x belongs to class 1, 2, respectively.
- log-odds for LDA:

$$\log \frac{p_1(x)}{1 - p_1(x)} = \log \frac{p_1(x)}{p_2(x)} = c_0 + c_1 x,$$

  c_0, c_1 functions of μ_1, μ_2, σ^2.
- log-odds for logistic regression:

$$\log \frac{p_1(x)}{1 - p_1(x)} = \beta_0 + \beta_1 x.$$

- Both linear in x, hence produce linear decision boundaries.
- β_0, β_1 via MLE, c_0, c_1 from estimation of mean, variance of Gaussians.
- Same relation between LDA and logistic regression holds for $p > 1$.
- LDA and logistic regression can give differing results if assumptions on Gaussian distribution not met, in this case logistic regression superior.

KNN

- Prediction for observation $X = x$ based on K training observations closest to x. Class selected based on majority of neighbors.
- Non-parametric, no assumptions on shape of decision boundary, hence expected to be superior to LDA and logistic regression when decision boundary highly nonlinear.
- KNN, however, gives no information on relative importance of predictor variables.

QDA

- Compromise between non-parametric KNN and linear LDA/logistic regression.
- Less flexible than KNN.
- Makes some assumptions on decision boundary shape, can perform better for limited observation numbers.

A Comparison of Classification Methods

Comparison scenarios

Six scenarios for comparison: 3 linear, 3 nonlinear decision boundaries.
100 random training data sets each.
For KNN used $K = 1$ and value chosen by *cross validation* (later).

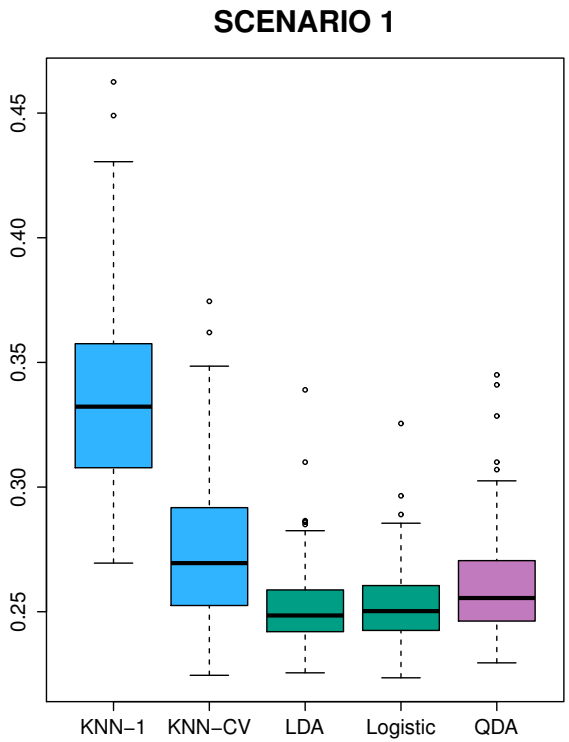
Scenario 1 20 training observations in each of 2 classes; in each class: uncorrelated Gaussian with separate means. LDA performs well, KNN high variance not offset by reduction in bias. QDA worse than LDA since classifier more flexible than necessary. Logistic regression: only slightly worse than LDA (linear decision boundary).

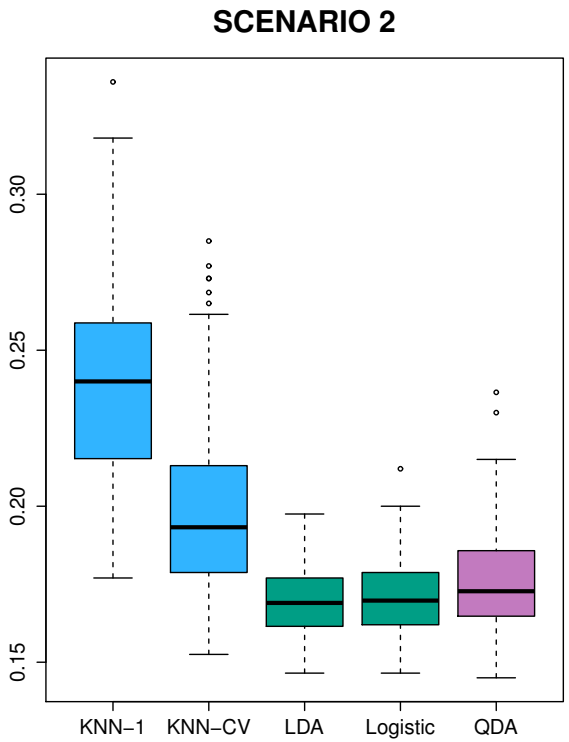
Scenario 2 Same as Scenario 1 except that within each class the 2 predictors had correlation -0.5. Little change.

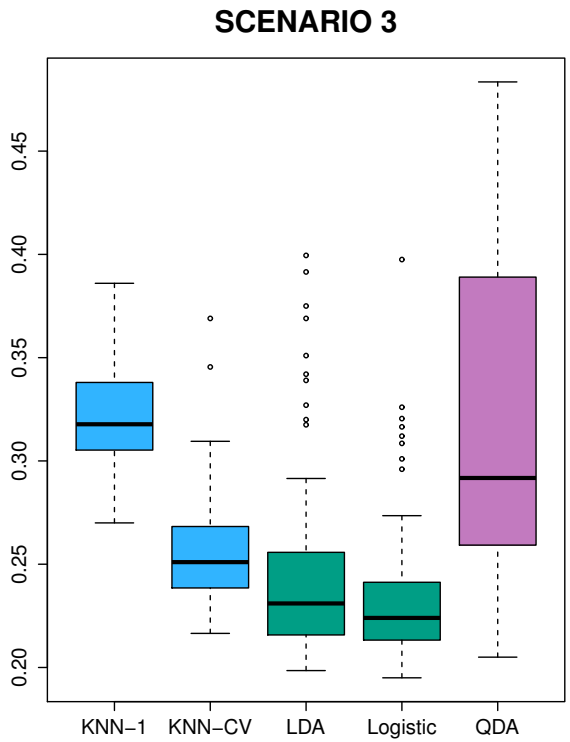
Scenario 3 X_1, X_2 from t-distribution (heavier tails than Gaussian), 50 observations per class. Decision boundary still linear, so assumptions of logistic regression satisfied, but those of LDA violated. Logistic regression outperforms LDA. QDA deteriorates considerably due to non-normality.

A Comparison of Classification Methods

Comparison scenarios







A Comparison of Classification Methods

Comparison scenarios

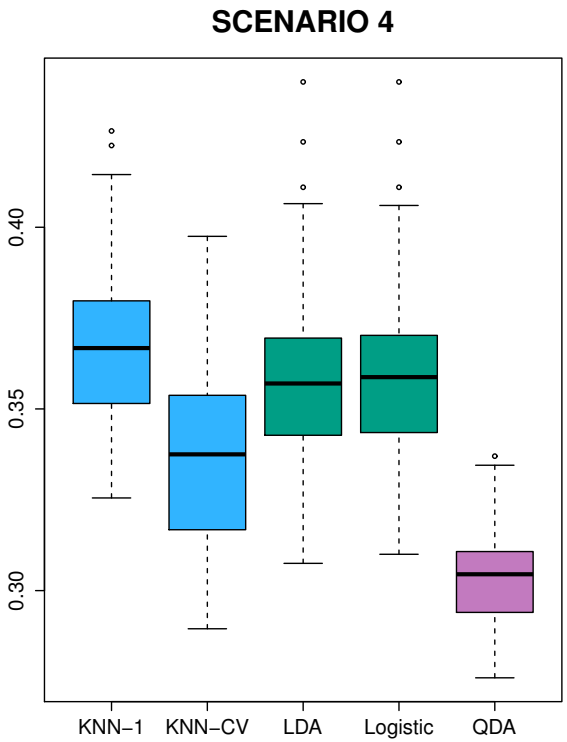
Scenario 4 Normal distribution, correlation 0.5 in the first, -0.5 in second class. Corresponds to QDA assumptions, quadratic decision boundaries. QDA outperforms all other methods.

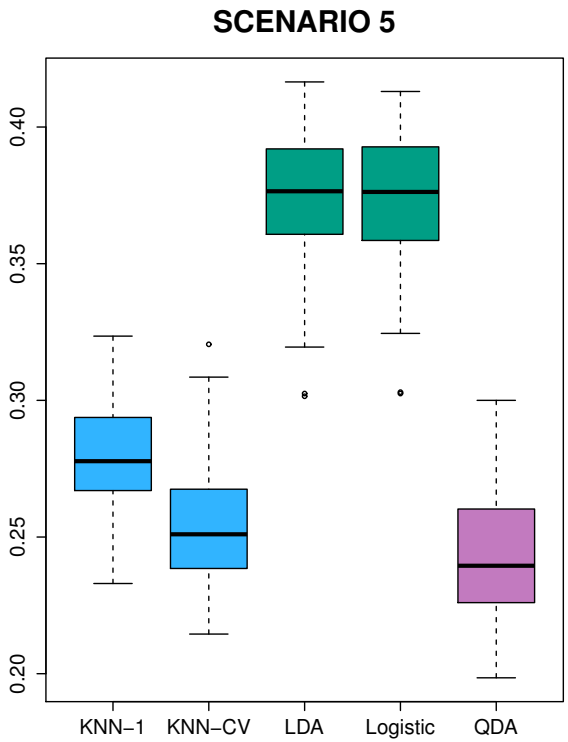
Scenario 5 In each class observations generated by normals with uncorrelated predictors, responses sampled from logistic function using X_1^2, X_2^2 and $X_1 X_2$ as predictors. Quadratic decision boundary. QDA best followed closely by KNN-CV. Linear methods perform poorly.

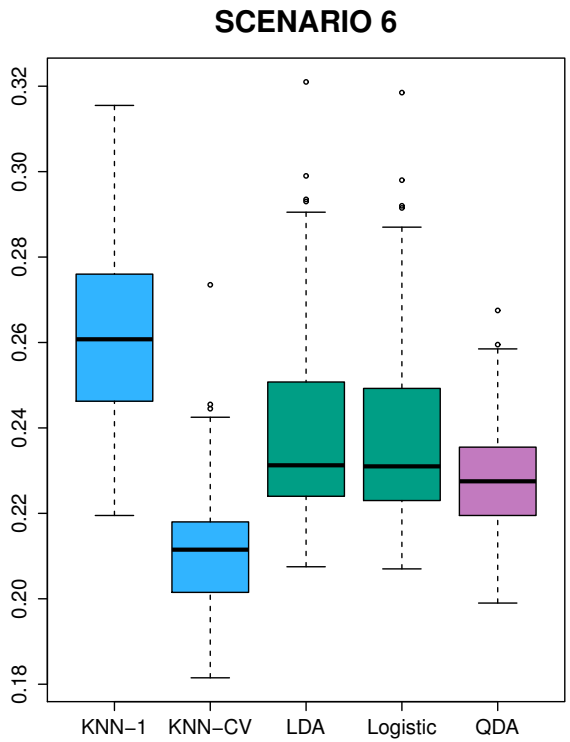
Scenario 6 Same as in 5, except now responses sampled from a more non-linear expression. Now even QDA can no longer correctly model complex decision boundary. QDA better than the linear methods, but more flexible KNN-CV gave best results. But: KNN with $K = 1$ gives worst results.

A Comparison of Classification Methods

Comparison scenarios







A Comparison of Classification Methods

Comparison scenarios

Summary:

- No method superior in all situations.
- Linear decision boundaries: LDA/logistic regression will perform well.
- Moderately nonlinear decision boundaries: QDA can be better.
- More highly nonlinear decision boundaries: high-variance method such as KNN may have advantages, but correct choice of smoothness (flexibility) parameter can be crucial.
- Next chapter: methods for finding the right amount of smoothing.

Contents

Resampling Methods

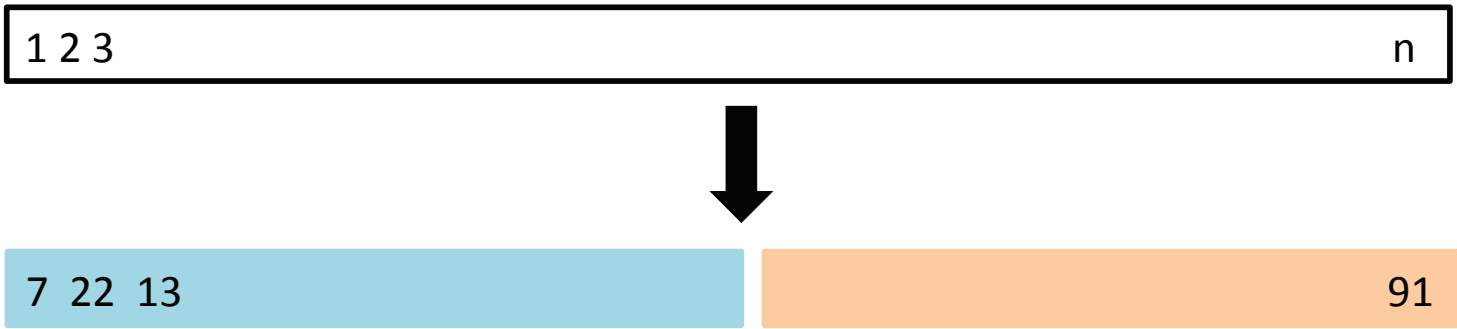
- **Resampling methods** refers to a set of statistical tools which involve refitting a model on different subsets of a given data set in order to assess the variability of the resulting models.
- These methods are computationally more demanding, but now feasible due to increased compute resources.
- Resampling is useful for **model assessment**, i.e., the process of evaluating a model's performance, as well as **model selection**, i.e., the process of selecting the proper level of model flexibility.
- In this chapter we introduce the resampling methods **cross validation** and the **bootstrap**.

Contents

Resampling Methods

Validation set approach

- Chapter 2: training set error vs. test set error.
- Training set error easily calculated, but usually overoptimisrtically low.
- Predictive value of model rests on low test set error.
- **Validation set approach**: divide available observations into *training set* and **validation set** or *hold-out set* and use latter as test set data.

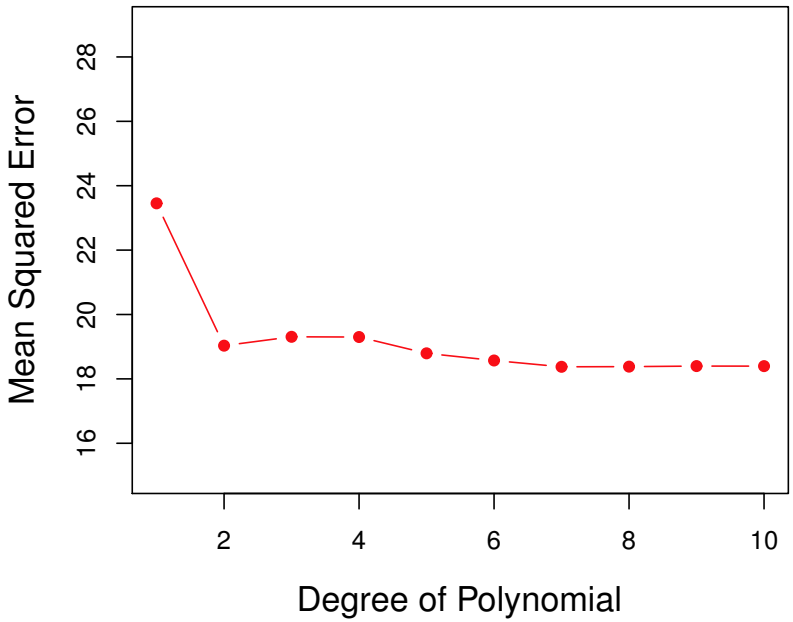


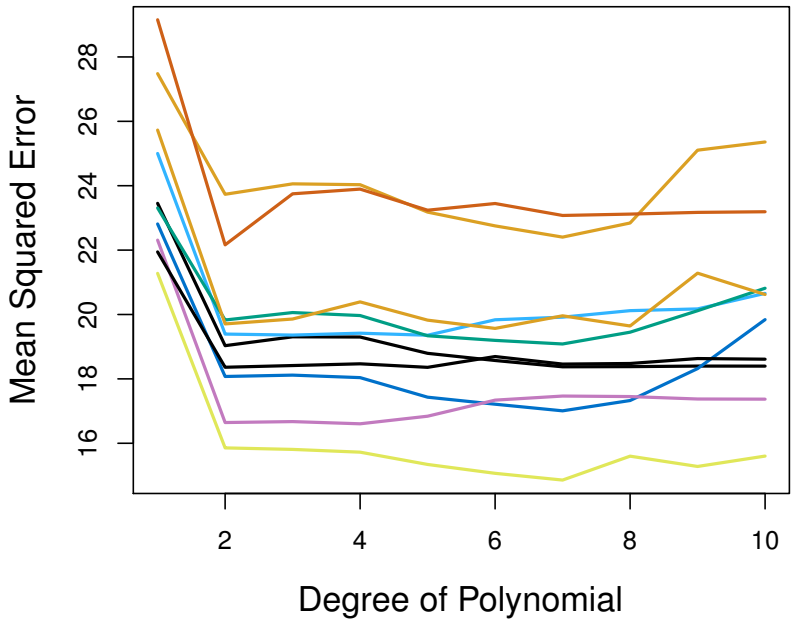
Validation set approach schematic: n observations randomly split into training set (beige) and validation set (blue).

Resampling Methods

Validation set approach

- Recall `Auto` data set (Chapter 3): model predicting `mpg` using `horsepower` and $\texttt{horsepower}^2$ better than linear model.
- Q: would model using higher order polynomial terms yield better (training) results?
- Validation set approach: partition the 392 observations into two sets of 196 each, use as training and validation sets, compute test MSE for various polynomial regression models. Compare different random partitions.



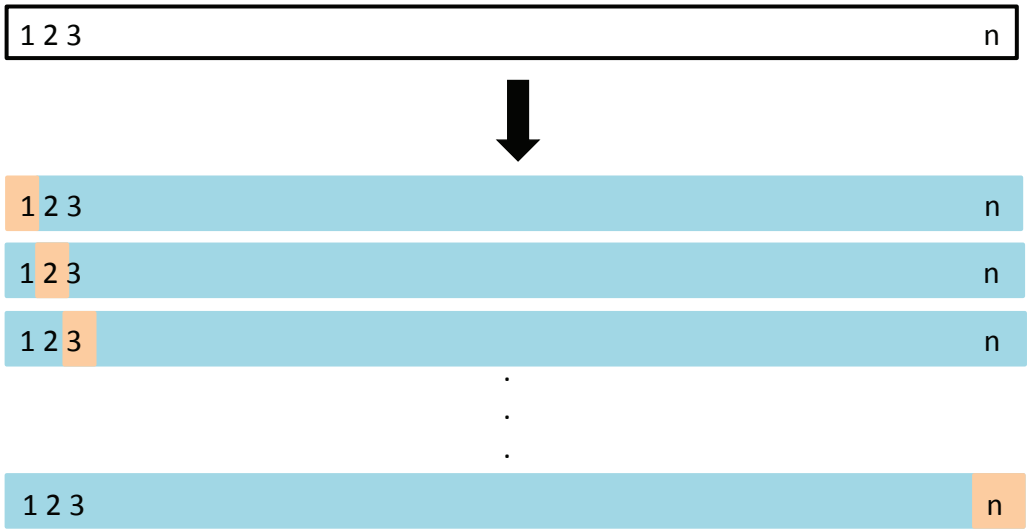


- All 10 partitions agree: adding quadratic term leads to lower validation set MSE, no benefit for higher degree terms.
- Different validation set MSE sequence for each partition.

Two principal shortcomings of validation set approach:

1. High variability of validation set MSE with changing partitions.
2. Valuable data not used to fit model, we expect this results in *overestimating* the test error rate.

- **Leave-one-out cross-validation** (LOOCV): for n observations, use n one-element validation sets, fit model using $n-1$-element training sets.

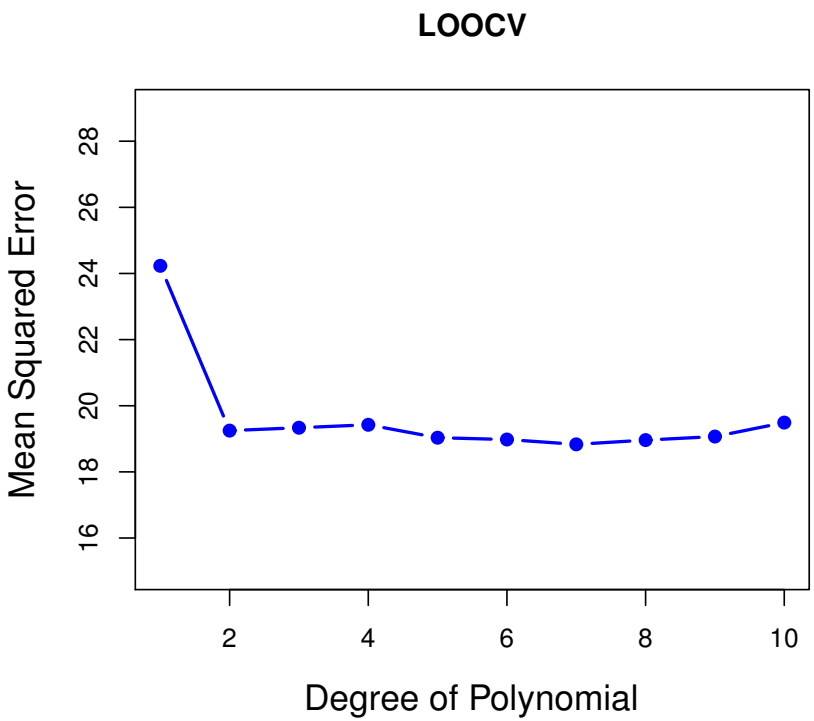


- MSE_i, $i = 1, \ldots, n$: test MSE when validation set consists of i-th observation.
- LOOCV estimate:

$$\mathrm{CV}_{(n)} := \frac{1}{n} \sum_{i=1}^{n} \mathrm{MSE}_i \,.$$

Advantages of LOOCV:

1. Less bias, since each fit uses nearly all observations, less overestimation of test error rate.
2. Well-defined approach, no arbitrariness in partitioning the data as in validation set approach.



LOOCV error curve for `Auto` data set: predicting `mpg` as a polynomial function of `horsepower` for varying polynomial degrees.

- LOOCV requires n fits of $n-1$ observations rather than one for for n observations. Potentially expensive for large n.
- Magic formula:

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1-h_i}\right)^2, \qquad h_i = \frac{1}{n} + \frac{(x_i-\overline{x})^2}{\sum_{j=1}^{n}(x_j-\overline{x})^2}. \tag{5.1}$$

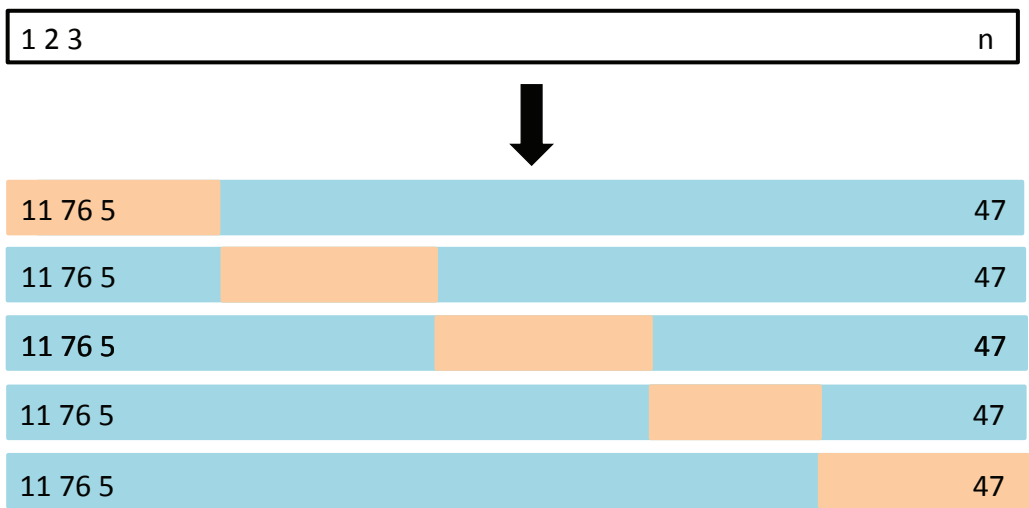
  $h_i \in (1/n, 1)$ is the leverage statistic of observation i as defined in (3.31).
- CV estimate is weighted MSE.
- Makes LOOCV cost same as single fit!
- LOOCV widely applicable (logistic regression, LDA, ...), but (5.1) does not hold in general.

Resampling Methods

k-fold cross validation

- Alternative to LOOCV: **k-fold CV**.
- Randomly partition observations into k groups or **folds**, $\approx$ equal in size.
- Use first fold as validation set and fit using remaining observations.
- Mean-squared error MSE_1 computed using first fold.
- Repeat $k-1$ more times with remaining folds, to obtain $\mathsf{MSE}_2, \ldots, \mathsf{MSE}_k$, and set

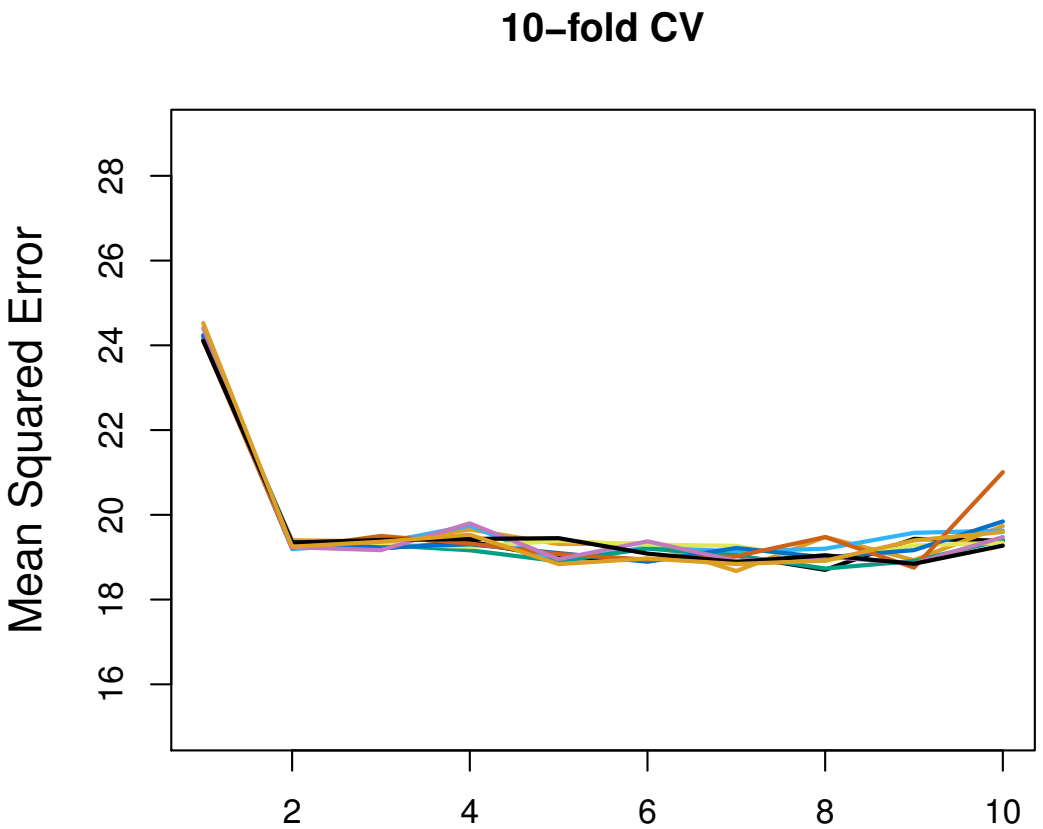
$$\mathsf{CV}_{(k)} := \frac{1}{k} \sum_{i=1}^{k} \mathsf{MSE}_i \,. \tag{5.2}$$



Resampling Methods

k-fold cross validation

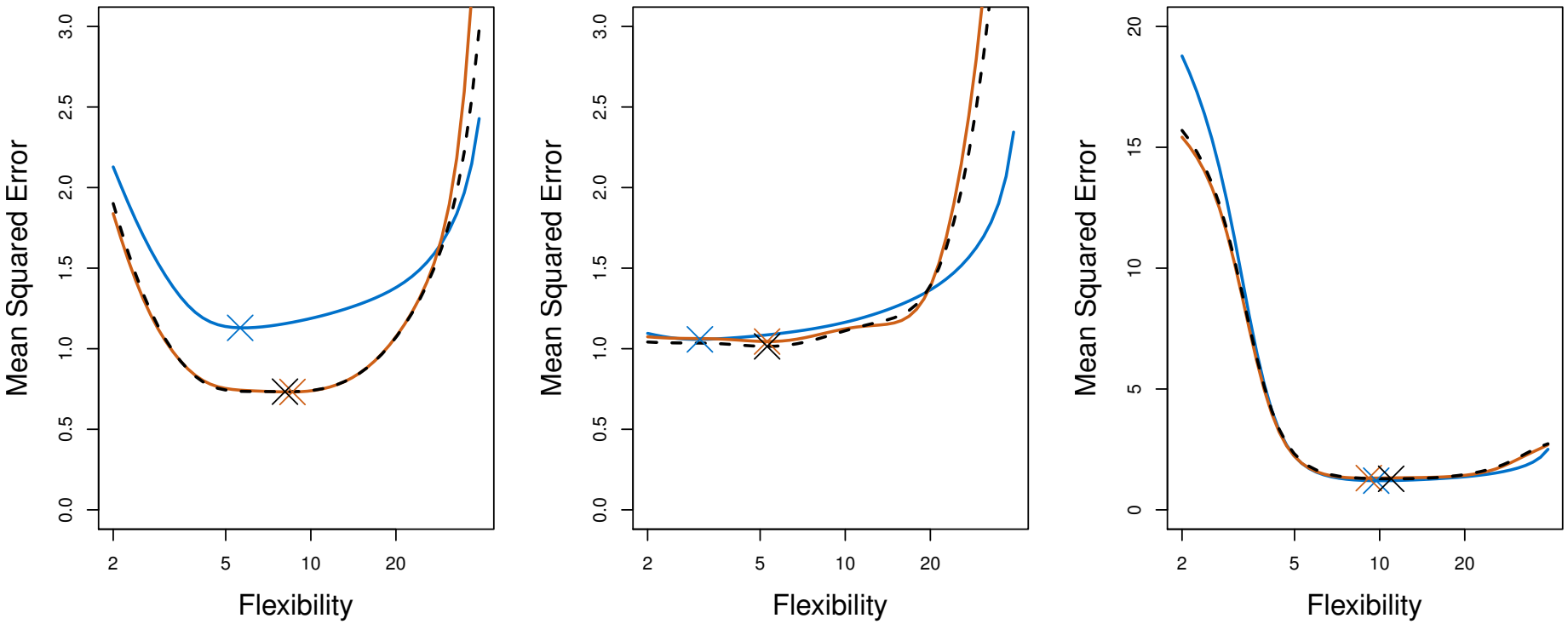
- LOOCV special case of k-fold CV with $k = n$.
- $k = 5$ or $k = 10$ commonly used.
- Appeal: computationally cheaper when magic formula cannot be used.



Nine 10-fold CV estimates for `Auto` data set, each resulting from a different random partition into 10 folds.
Some variability visible, much less than for validation set approach.

Resampling Methods

CV applied to example from Chapter 2



CV estimates for smoothing splines applied to simulated data sets from Chapter 2: LOOCV (black dashed), 10-fold CV (orange solid) beside true test MSE (blue). Crosses denote minimum of each curve.

- Besides its computational advantage, k-fold CV often gives more accurate test MSE estimates than LOOCV.
- Bias reduction: LOOCV gives approximately unbiased estimates, since it uses $n-1$ observations to fit. Validation set approach: most bias, since fewest observations used. k-fold CV: intermediate, as $(k-1)n/k$ observations in each training set.
- Variance: LOOCV has higher variance than k-fold CV with $k < n$.
- Reason: LOOCV gives average of n fitted models, each trained on nearly identical set of models, hence outputs highly correlated.
- For k-fold CV with $k < n$, average outputs of k fitted models whose outputs are less correlated (since overlap between training sets smaller).
- Mean of many highly correlated quantities has higher variance than mean of many quantities which are not as highly correlated, test error estimate resulting from LOOCV tends to have higher variance than test error estimate resulting from k-fold CV.

- For classification (Y qualitative) replace MSE by *number of misclassification* and set

$$\mathrm{CV}_{(n)} := \frac{1}{n}\sum_{i=1}^{n} \mathrm{Err}_i \qquad \mathrm{Err}_i := \mathbb{1}_{\{y_i \neq \hat{y}_i\}}. \tag{5.3}$$

  k-fold CV and validation set error rates defined analogously.

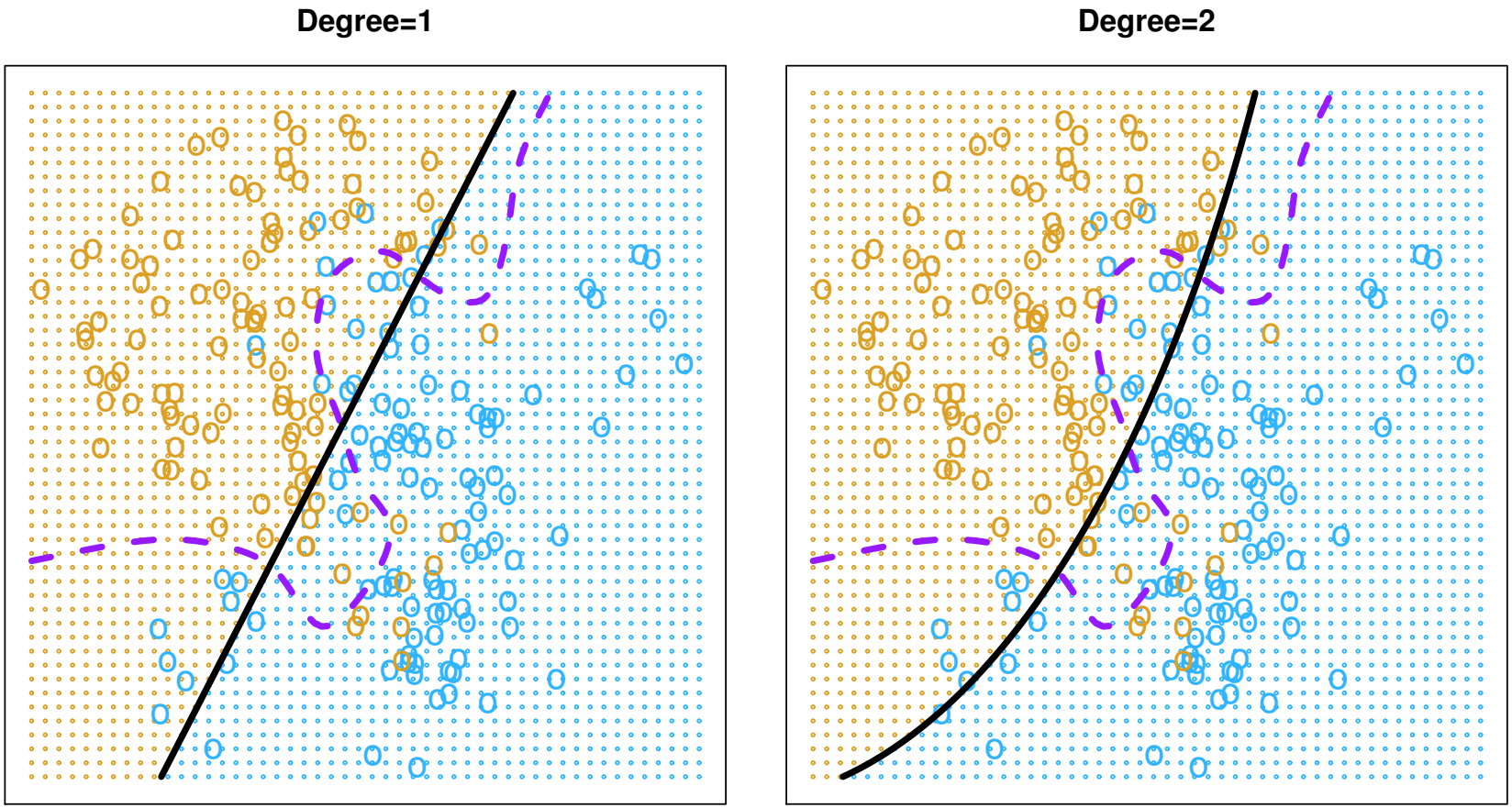
- Can use CV e.g. to perform logistic regression.
- As in linear regression setting, can use polynomial functions in predictor variables:

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2. \tag{5.4}$$

- Consider classification problem from Chapter 2 (Slide 62)

Resampling Methods

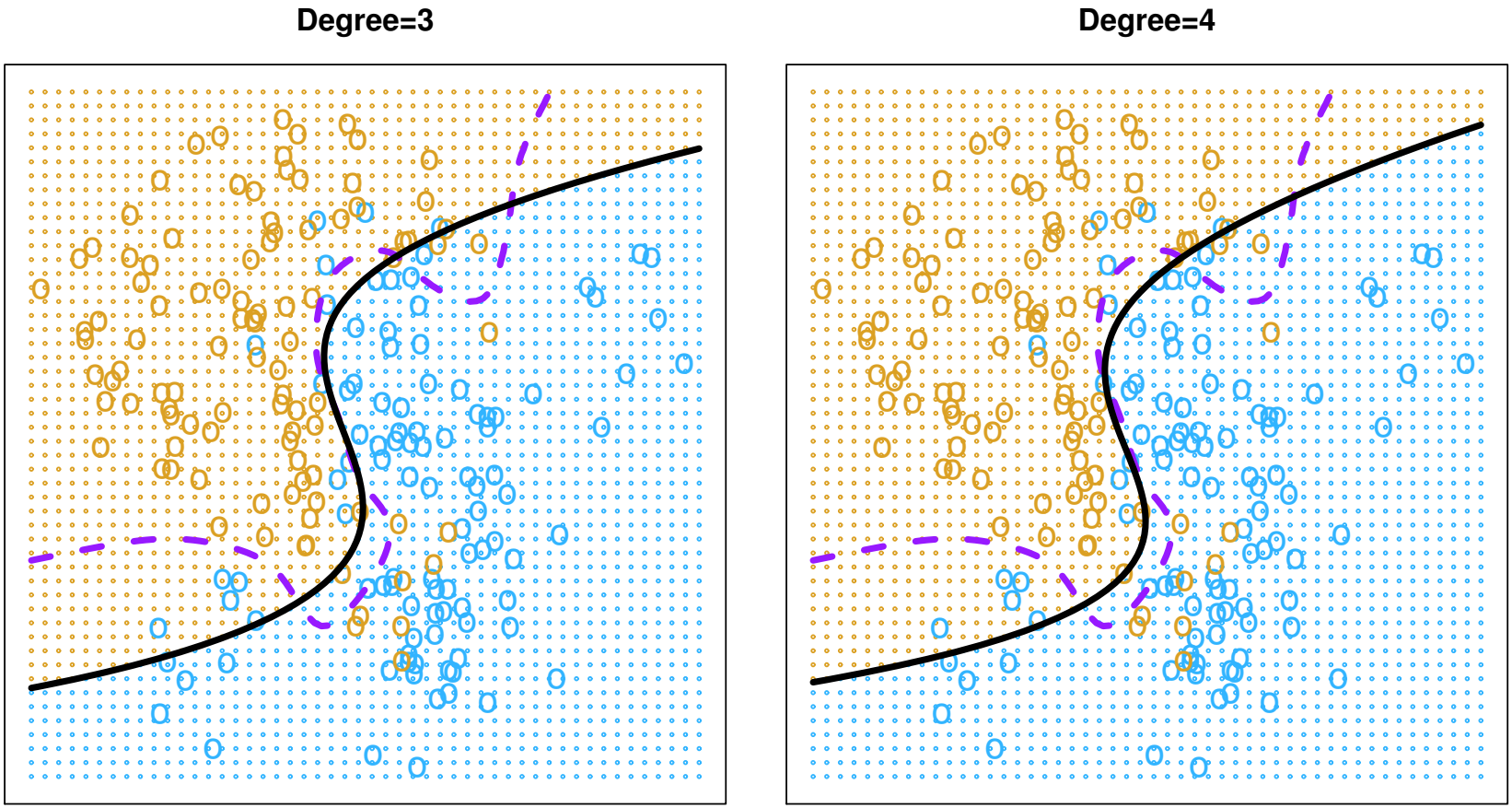
CV in classification setting



Logistic regression fit of 2D classification problem from Slide 62: Bayes decision boundary (purple dashed) and estimated decision boundary (solid black). Left: linear fit. right: quadratic fit. Bayes error rate: 0.133. (True) test error rates: 0.201 and 0.197.

Resampling Methods

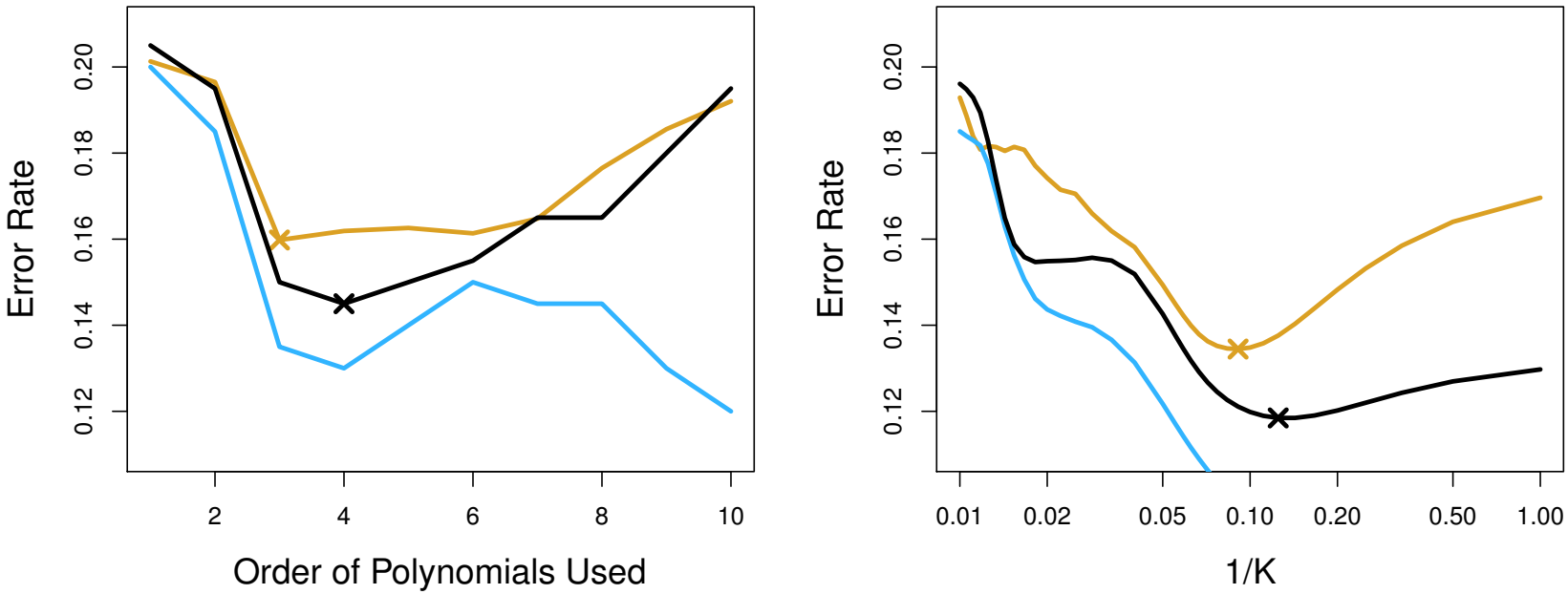
CV in classification setting



Same problem, same legend. Logistic regression now using cubic and quartic fits.

Bayes error rate: 0.133. (True) test error rates now : 0.160 and 0.162.

- In practice neither Bayes decision boundary, Bayes error rate nor true test error rate available, but CV offers way to choose among previous 4 models.



Same problem, same models. Black: 10-fold CV error rates from fitting 10 logistic regression models using polynomial functions of the predictor variables up to degree 10. Brown: true test errors, blue: training set errors. Right: KNN classifier with varying K (now denoting # nearest neighbors).

Observations:

- Training error decreases (roughly) with model flexibility.
- Test set error displays typical U-shape.
- 10-fold CV estimate provides good approximation of test error rates.
- Minimal for degree 4, matches true minimum well.
- Similar observations for KNN.
- Obvious: training set error not useful for model selection.

Contents

Resampling Methods

The bootstrap

- The **bootstrap** is a widely applicable and powerful statistical tool for quantifying the uncertainty associated with an estimate or statistical learning method.
- Example: linear regression coefficients (although simpler alternatives here).
- Nice introduction: [Efron, 2013s][6]

[6] A 250-Year Argument: Belief, Behavior and the Bootstrap. Bull. AMS 50(1) 2013 pp. 129–146.

- Goal: invest fixed sum of money in 2 financial assets with random returns X and Y.
- Invest fraction α in X, remaining $1-\alpha$ in Y.
- Choose α to minimize total risk (variance) of investment, i.e., minimize $\mathbf{Var}(\alpha X + (1-\alpha)Y)$.
- Can show: risk-minimizing value given by

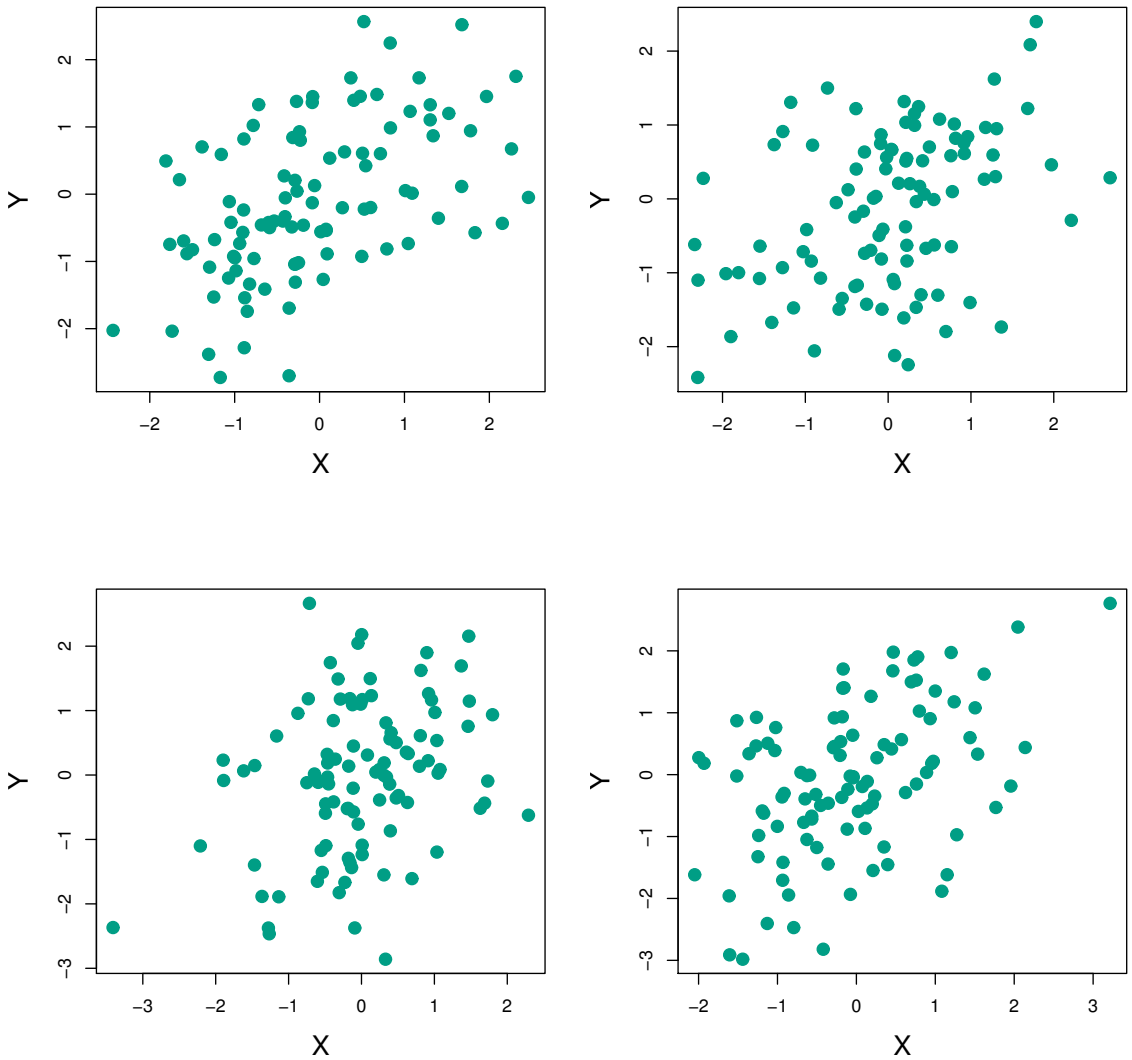
$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}, \tag{5.5}$$

  where $\sigma_X^2 = \mathbf{Var}\, X$, $\sigma_Y^2 = \mathbf{Var}\, Y$, $\sigma_{XY} = \mathbf{Cov}(X, Y)$.
- Latter quantities unknown in practice, use estimates $\hat{\sigma}_X^2$ $\hat{\sigma}_Y^2$ $\hat{\sigma}_{XY}$ and estimate risk minimizing ratio as

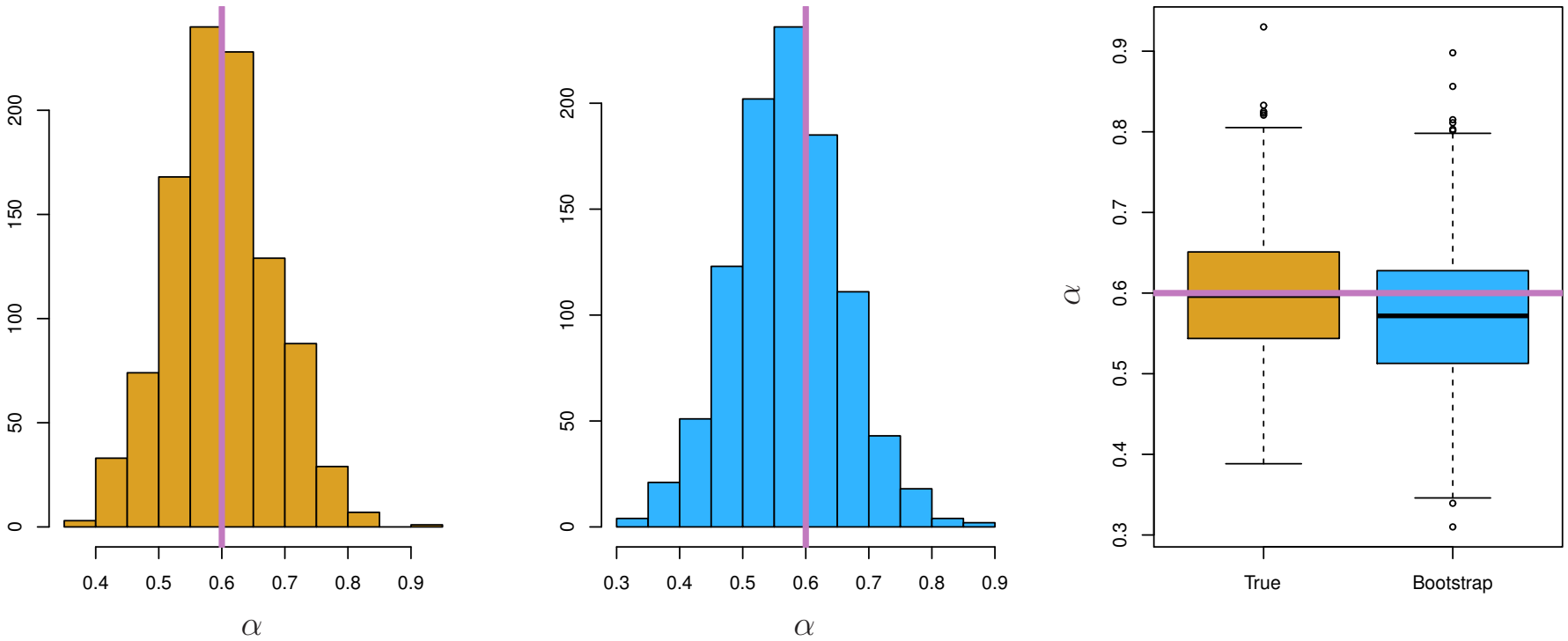
$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\sigma_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}. \tag{5.6}$$

Resampling Methods

The bootstrap: investment problem



Each panel: 100 simulated returns X and Y.
Lexicographically, estimates for α are 0.576, 0.532, 0.657 and 0.651.



Quantification of estimate for α: repeated process of simulating 100 (X, Y)-observations and estimating α using (5.6) 1000 times. Left: histogram of $\{\hat{\alpha}_j\}_{j=1}^{1000}$. ($\sigma_X^2 = 1, \sigma_Y^2 = 1.25, \sigma_{XY} = 0.5, \Rightarrow \alpha = 0.6$, solid vertical line). Center: bootstrap histogram. Right: boxplots of original data and bootstrap data sets.

Mean over all estimates:

$$\overline{\alpha} = \frac{1}{1000} \sum 1000\hat{\alpha}_i = 0.5996 \approx 0.6 = \alpha.$$

Standard deviation:

$$\sqrt{\frac{1}{1000-1} \sum_{i=1}^{1000} (\hat{\alpha}_i - \overline{\alpha})^2} = 0.083,$$

hence $\mathsf{SE}(\hat{\alpha}) \approx 0.083$.
We thus expect $\hat{\alpha}$ to deviate from α by 0.08 on average.

Bootstrap estimate: $\mathsf{SE}(\hat{\alpha}) = 0.087$.

Bootstrap approach:

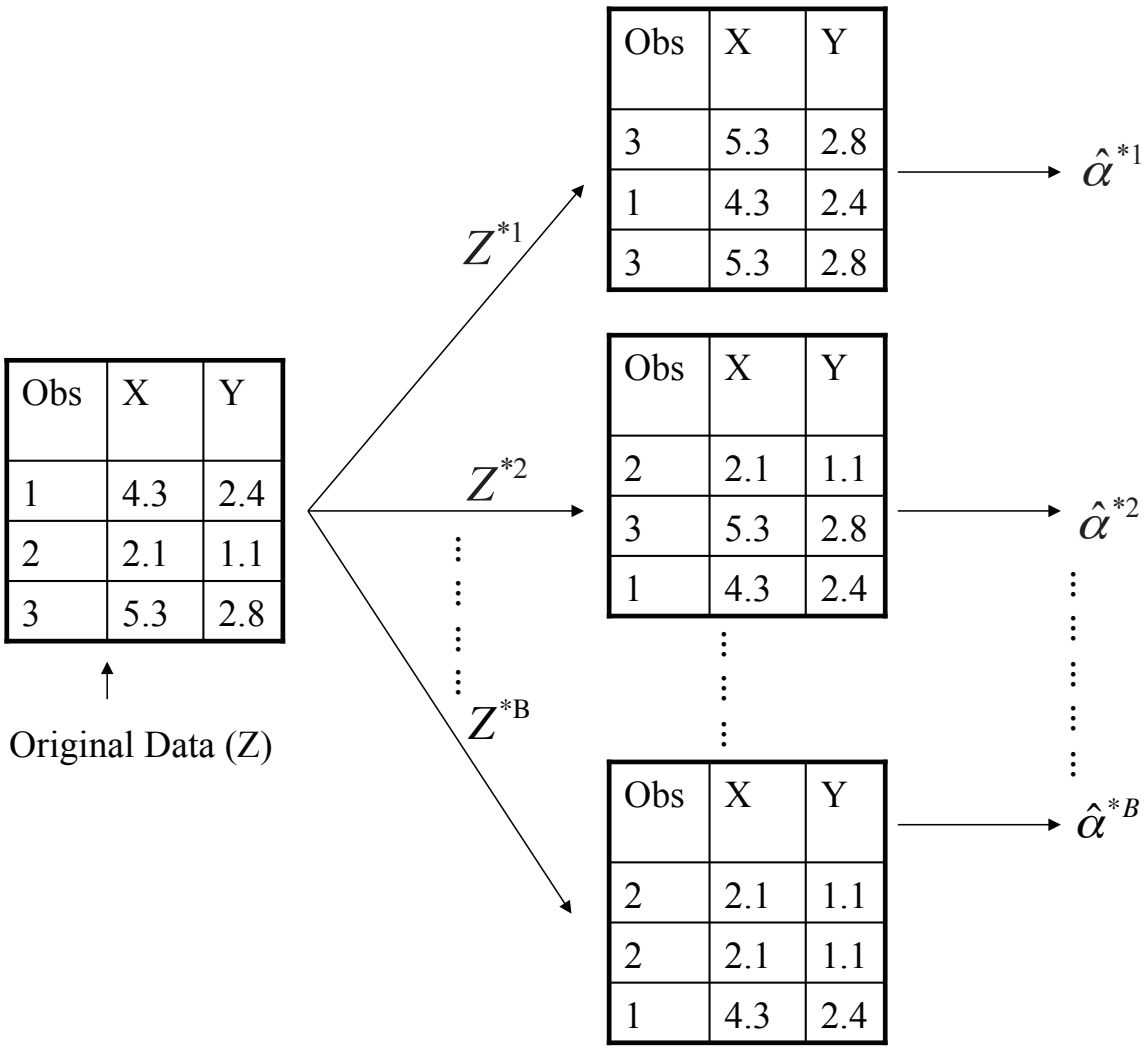
- In general, can't generate multiple instances of given data.
- Bootstrap: use computer to emulate generation of new sample data sets.
- Use these to assess variability of associated estimates.
- Sampling proceeds from *original data set*.
- Sampling proceeds with *replacement*, all components of an observation treated as a unit.
- For $i = 1, \ldots, B$, generate i-th **bootstrap data set** Z^{*i}, each with estimate $\hat{\alpha}^{*i}$.
- Can estimate standard error of these estimates by

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}\left(\hat{\alpha}^{*i} - \frac{1}{B}\sum_{j=1}^{B}\hat{\alpha}^{*j}\right)^2} \tag{5.7}$$

- Example for data set Z containing $n = 3$ elements:

Resampling Methods

The bootstrap



Contents

Linear Model Selection and Regularization

Chapter overview

- Alternative fitting procedures to least squares (LS) for standard linear model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon \tag{6.1}$$

  to improve **prediction accurary** and **model interpretability**.
- Prediction accuracy: for approximately linear (true) model, LS has low bias and, if $n \gg p$, also low variance. More variability if $n \gtrsim p$, no unique minimizer if $n < p$.
  Idea: constraining or **shrinking** estimated coefficients reduces variability in these cases at negligible increase in bias, improving prediction accuracy.
- Model interpretability: some predictor variables may be irrelevant for response; LS will not remove these, hence consider other methods for **feature selection** or **variable selection** to exclude irrelevant variables from multiple regression model (by producing zero coefficients for these).

We consider three classes of fitting alternatives to LS:

- **Subset selection**: Find subset of initial p predictor variables which are relevant, fit model using LS for reduced set of variables.
- **Shrinkage**: fit all p variables, shrink coefficients towards zero relative to LS estimate. Shrinkage (also known as **regularization**) reduces variance, some coefficients shrunk to zero, can be viewed as variable selection.
- **Dimension reduction**: project p predictors into subspace of dimension $M < p$, i.e., construct M linearly independent *pseudo-variables* which depend linearly on original p predictor variables. Use these as new predictors for LS fit.
- Same concepts apply to other methods (e.g. classification).

Contents

Idea: Perform separate LS regression for *all possible subsets* of given p predictor variables.

Algorithm 1: Best subset selection.

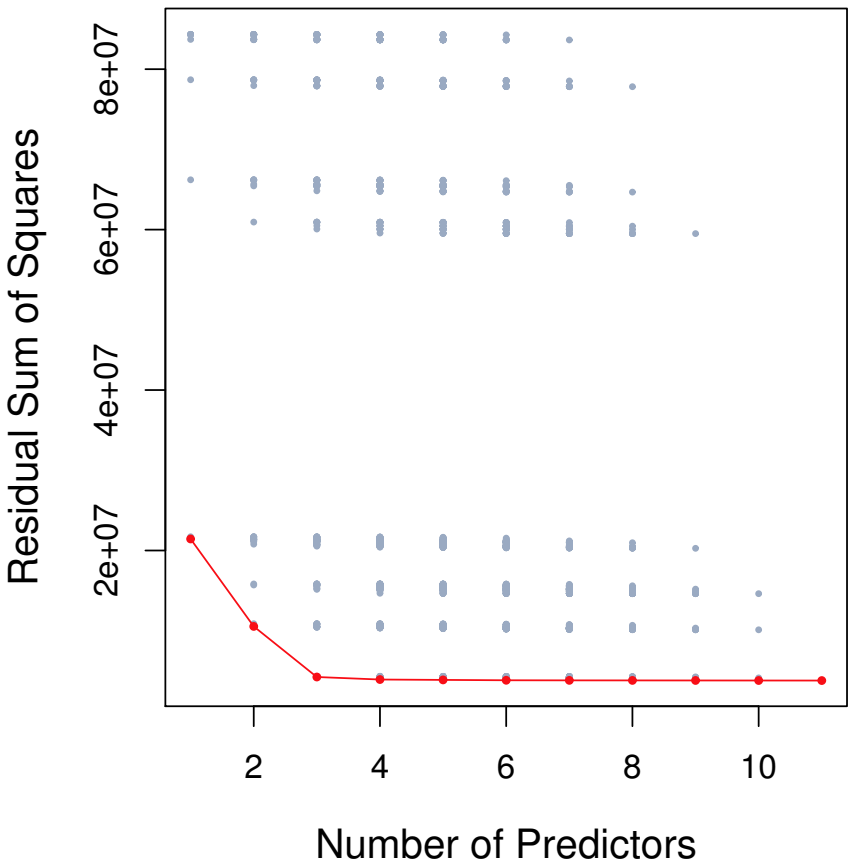
---

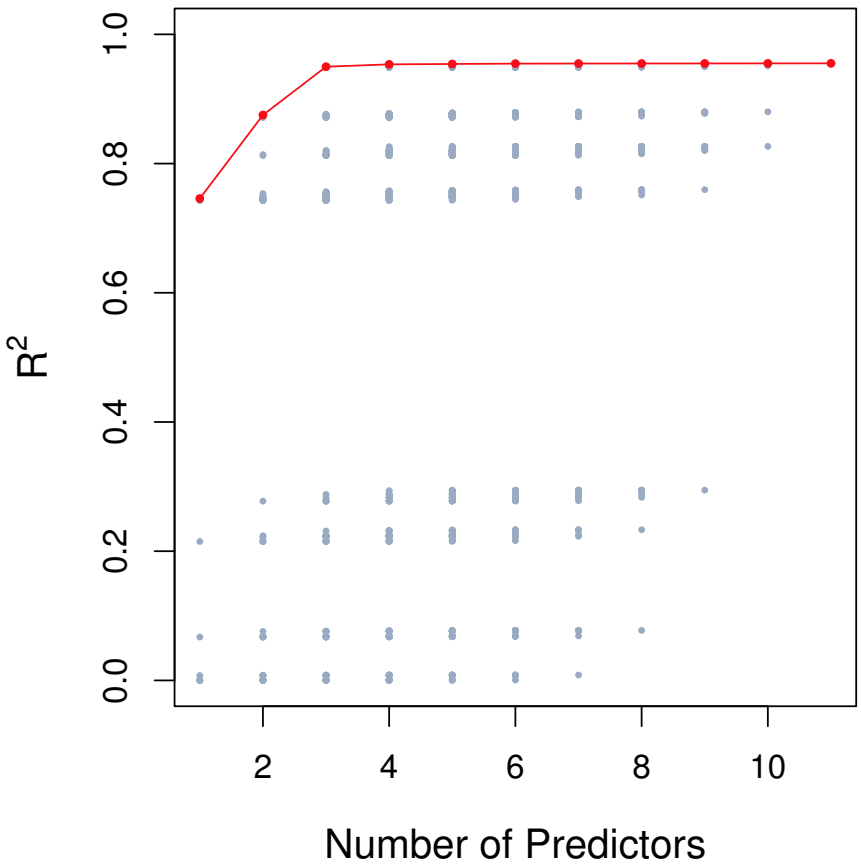
1. Set $\mathscr{M}_0$ to be the **null model**, i.e., containing only constant term β_0.
2. **for** $k = 1, 2, \ldots, p$
   a. Fit all $\binom{p}{k}$ models containing exactly k predictors.
   b. Pick best (smallest RSS, i.e., largest R^2) among these, call it $\mathscr{M}_k$.
3. Select single best model among $\mathscr{M}_0, \ldots, \mathscr{M}_p$ using model selection criterion (later).

- Step 2 reduces # model candidates from 2^p to $p + 1$.
- Models in Step 3 display monotone decreasing RSS (increasing R^2) as # variables increases.
- Want low test error rather than low training error.

Linear Model Selection and Regularization

Best subset selection





Best subset selection for `Credit` data set: 10 predictors (three-valued variable `ethnicity` coded using two dummy variables selected separately).

Red line indicates model with smallest RSS (largest R^2).

- Can apply to classification problems using **deviance** in place of RSS ($-2 \cdot$ maximized log-likelihood).
- Best subset selection simple, but # regression fits to compare grows exponentially with p (e.g. 1024 for $p = 10$, over 1 million for $p = 20$).
- Also, statistical problems for large p: the larger the search space, the higher the chance of finding models performing well on training set, but badly for test set.

Idea: Add predictors to model one at a time, at each step adding variable leading to greatest additional improvement.

Algorithm 2: Forward stepwise selection.

---

1. Set $\mathscr{M}_0$ to be the **null model**, i.e., containing only constant term β_0.
2. **for** $k = 0, 1, \ldots, p-1$
   a. Consider all $p-k$ models augmenting $\mathscr{M}_k$ by one additional predictor.
   b. Pick best (smallest RSS, i.e., largest R^2) among these, call it $\mathscr{M}_{k+1}$.
3. Select single best model among $\mathscr{M}_0, \ldots, \mathscr{M}_p$ using model selection criterion (later).

- Rather than 2^p models considered by best subset selection, forward stepwise selection requires only $1 + p(p+1)/2$ LS fits.
  E.g. $p = 20$: 1,048,576 models for best subset selection, 211 models for fowrard stepwise selection.

- Forward stepwise selection not guaranted to find best model out of 2^p possible. E.g. for $p = 3$, best single-variable model could consist of X_1, while best two-variable model consists of X_2, X_3.
- First 4 selected models for best subset selection and forward stepwise selection on `Credit` data set:

# variables	Best subset	Forward stepwise
1	`rating`	`rating`
2	`rating`, `income`	`rating`, `income`
3	`rating`, `income`, `student`	`rating`, `income`, `student`
4	`cards`, `income`	`rating`, `income`
	`student`, `limit`	`student`, `limit`

- Can use forward stepwise selection in high-dimensional case when $n < p$. However, can only construct submodels $\mathcal{M}_0, \ldots, \mathcal{M}_{n-1}$, since LS can uniquely fit at most $n - 1$ variables.

Idea: Begin with full LS model, successively remove least useful predictor.

Algorithm 3: Backward stepwise selection.

1. Set $\mathscr{M}_p$ to be the **full model**, containing all p predictors.
2. **for** $k = p, p-1, \ldots, 1$
   a. Consider all k models containing all but one of the predictors in $\mathscr{M}_k$.
   b. Pick best (smallest RSS, i.e., largest R^2) among these k models, call it $\mathscr{M}_{k-1}$.
3. Select single best model among $\mathscr{M}_0, \ldots, \mathscr{M}_p$ using model selection criterion (later).

- Again only $1 + p(p+1)/2$ model fits.
- No guarantee of finding best model.
- Requires $n > p$.
- **Hybrid** approaches possible, where addition step followed by removal step.

Linear Model Selection and Regularization

Optimal model selection

- In best subset selection, forward selection and backward selection, need to choose best among models containing different # variables.
- RSS and R^2 measures will always select model with all p variables.
- Goal: select best model with respect to *test* error.
- Two basic approaches:
  1. Indirectly estimate test error by making an adjustment to training error to account for bias due to overfitting.
  2. Directly estimate test error using either validation set approach or cross-validation approach.

- Training set MSE generally underestimates test MSE (recall $\mathsf{MSE} = \mathsf{RSS}/n$)
- For LS regression: coefficients determined by minimization of RSS.
- Therefore training error decreases as variables added to model; not so for test error.
- For fitted LS model containing d predictors, C_p **estimate** defined by

$$C_p := \frac{1}{n}(\mathsf{RSS} + 2d\,\hat{\sigma}^2), \tag{6.2}$$

where $\hat{\sigma}^2$ is an estimate of $\mathbf{Var}\,\varepsilon$, typically computed using full model.
Adds penalty term $2d\,\hat{\sigma}^2$ to training RSS to compensate for underestimating test error.
Can show: C_p unbiased estimate of test MSE if $\hat{\sigma}^2$ unbiased estimate of σ^2.
Hence C_p small for models with small test MSE.

- **Akaike information criterion** (AIC) defined for models fit by maximum likelihood.
- For standard linear model (6.1) with Gaussian noise maximum likelihood fit coincides with LS fit.
- In this case

$$\mathsf{AIC} = \frac{1}{n\,\hat{\sigma}^2}(\mathsf{RSS} + 2d\,\hat{\sigma}^2)$$

  (have omitted additive constant).
- Hence, for LS models C_p and AIC proportional.

- **Bayesian information criterion** (BIC), derived from Bayes point of view, is given by (up to irrelevant constants)

$$\mathrm{BIC} = \frac{1}{n\,\hat{\sigma}^2}(\mathrm{RSS} + d\,\hat{\sigma}^2 \log n) \tag{6.3}$$

- Also tends to be small for models with small test error.
- Replaces $2d\hat{\sigma}^2$ used by C_p with $d\hat{\sigma}^2 \log n$, hence places heavier penalty on models with many variables, results in selection of smaller models than C_p.

- Recall $R^2 = 1 - \mathsf{TSS}\,/\,\mathsf{RSS}$,
  $\mathsf{TSS} = \sum(y_i - \bar{y})^2$ total sum of squares for response.
- R^2 increases as variables added to LS model.
- For LS model with d variables, **adjusted** R^2 statistic given by

$$\text{Adjusted } R^2 := 1 - \frac{\mathsf{RSS}\,/(n-d-1)}{\mathsf{TSS}\,/(n-1)} = 1 - \frac{\mathsf{RSS}}{\mathsf{TSS}} \cdot \frac{n-1}{n-d-1}. \qquad (6.4)$$

- Unlike C_p, AIC and BIC, where small value indicates model with low test error, here a *large* value of the adjusted R^2 statistic indicates a model with a small test error.
- Maximizing adjusted R^2 equivalent to minimizing $\mathsf{RSS}\,/(n-d-1)$.
- Intuition: once all relevant variables have been included, adding additional noise variables will only lead to small decrease in RSS.
- Compared to R^2, adjusted R^2 pays a price for adding irrelevant variables.

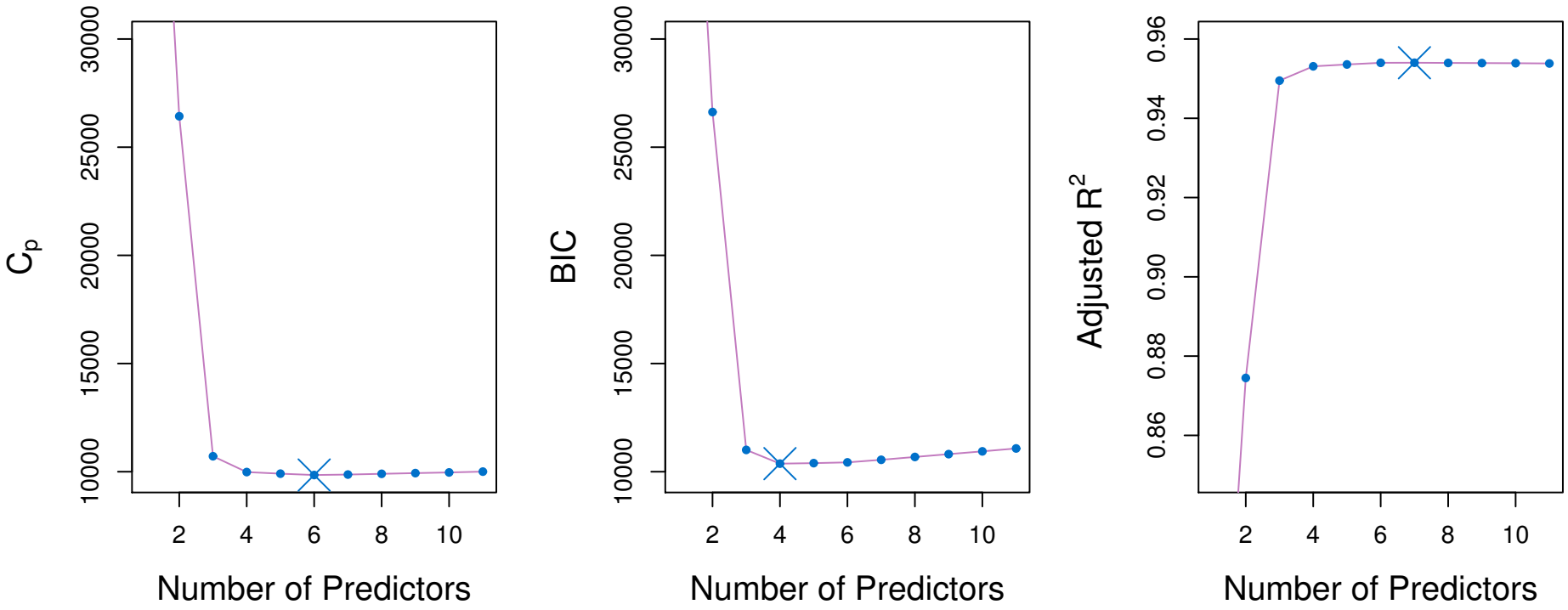
Linear Model Selection and Regularization

C_p, AIC, BIC, adjusted R^2

- Rigorous justifications of C_p, AIC, BIC rely on asymptotic arguments (large n limit).
- Adjusted R^2 popular, intuitive, but not as well motivated statistically.
- All measures simple to use and compute.
- Modified formulas for more general models.

Linear Model Selection and Regularization

C_p, AIC, BIC, adjusted R^2

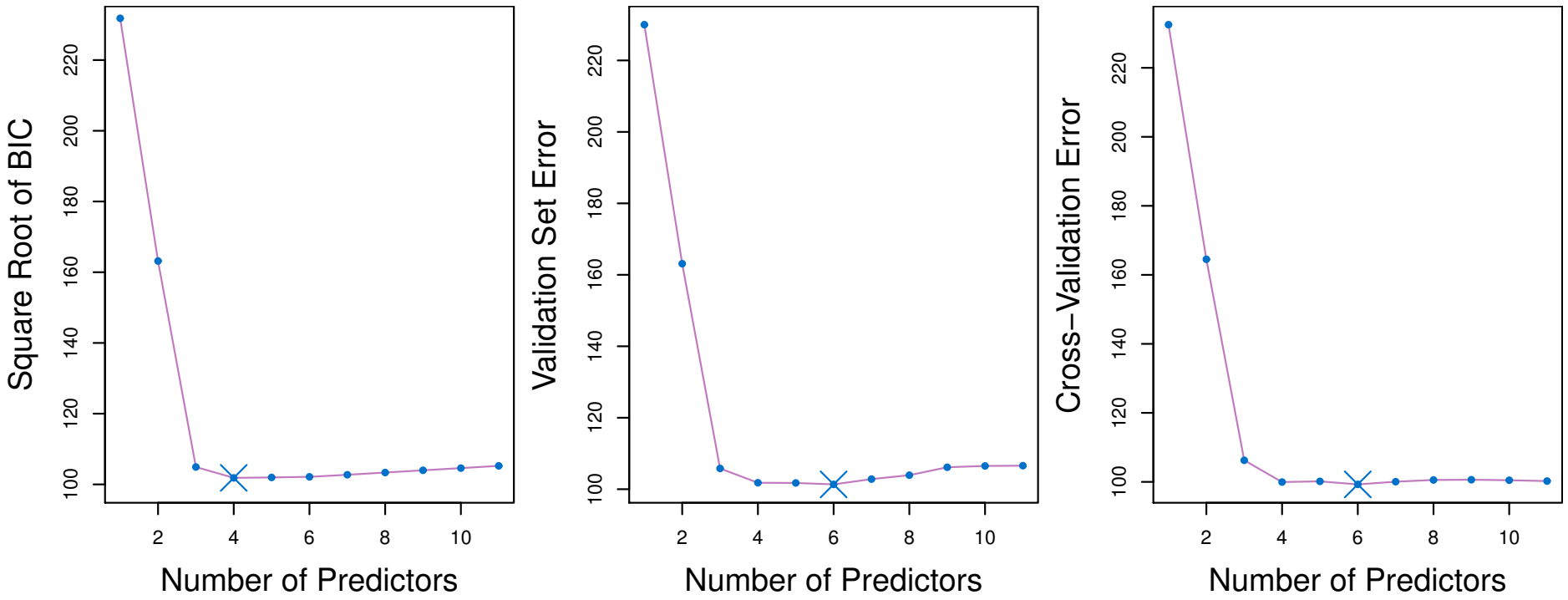


C_p, **BIC** and adjusted R^2 for best models of each size for `Credit` data set (red curve in previous plot).

- Can apply validation and cross-validation to each model and select that with lowest estimate.
- Advantage over C_p, AIC, BIC, adjusted R^2: direct estimate of test error, fewer assumptions about underlying model.
- More widely useable, e.g., when noise variance estimates difficult to obtain.
- CV initially less popular than C_p, AIC, BIC, adjusted R^2 due to computational cost; this is less and less an issue.
- Apply to `Credit` data set: display BIC, validation set errors, cross-validation errors as function of $d = \#$ variables in model.
  Validation: randomly choose 3/4 of observations as training set, remainder as validation set.
  Cross-validation using $k = 10$ folds.

Linear Model Selection and Regularization

Cross-validation



`Credit` data: 3 model error estimates for best model containing 1 to 11 predictors. Both validation set and CV result in 6-variable models. All approaches agree: not much difference in test error for 4, 5, 6-variable models.

Left: $\sqrt{\text{BIC}}$; center: validation set errors; right: CV errors.

- Observation: all 3 error estimates quite flat from 4 variables onward.
- Error estimate-minimizing model likely to change for different partitions of observations or different choice of CV folds.
- **One-standard-error rule**: calculate standard error of estimated test MSE for each model size, then select smallest model for which estimated test error is within one standard error of lowest point on curve.
  Rationale: if several models appear equally good, may as well choose simplest.
  Here: rule leads to 3-variable model.

Contents

Linear Model Selection and Regularization

Shrinkage

- **Inverse problems**: branch of applied mathematics for solving problems where solution extremely sensitive to data and/or solution not unique (e.g.: X-ray tomography, image deblurring).
- Prevalent strategy: instead of original problem, solve *nearby* problem with better stability properties: **regularization**.
- In LS methods: modify objective function by minimizing different norm or adding **penalty term**, thus imposing "a priori information" on the coefficients.
- In statistics, particularly in LS regression, regularization is known as **shrinkage**, as certain coefficients are "shrunk" in magnitude relative to their values under LS estimation.
- Here we introduce two popular shrinkage techniques: **ridge regression** and the **LASSO**.

Least-squares fitting determines coefficients $\beta_0, \ldots, \beta_p$ by minimizing

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_j \right)^2 = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2.$$

In **ridge regression**, one minimizes instead the objective function

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \mathrm{RSS} + \lambda \|\tilde{\boldsymbol{\beta}}\|_2^2, \tag{6.5}$$

where λ is a tuning parameter to be suitably chosen and $\tilde{\boldsymbol{\beta}} := (\beta_1, \ldots, \beta_p)^\top \in \mathbb{R}^p$. From now on $\boldsymbol{\beta} \in \mathbb{R}^p$ and tilde omitted.

In the inverse problems community, this general approach is known as **Tikhonov regularization** and λ is called the **regularization parameter**.

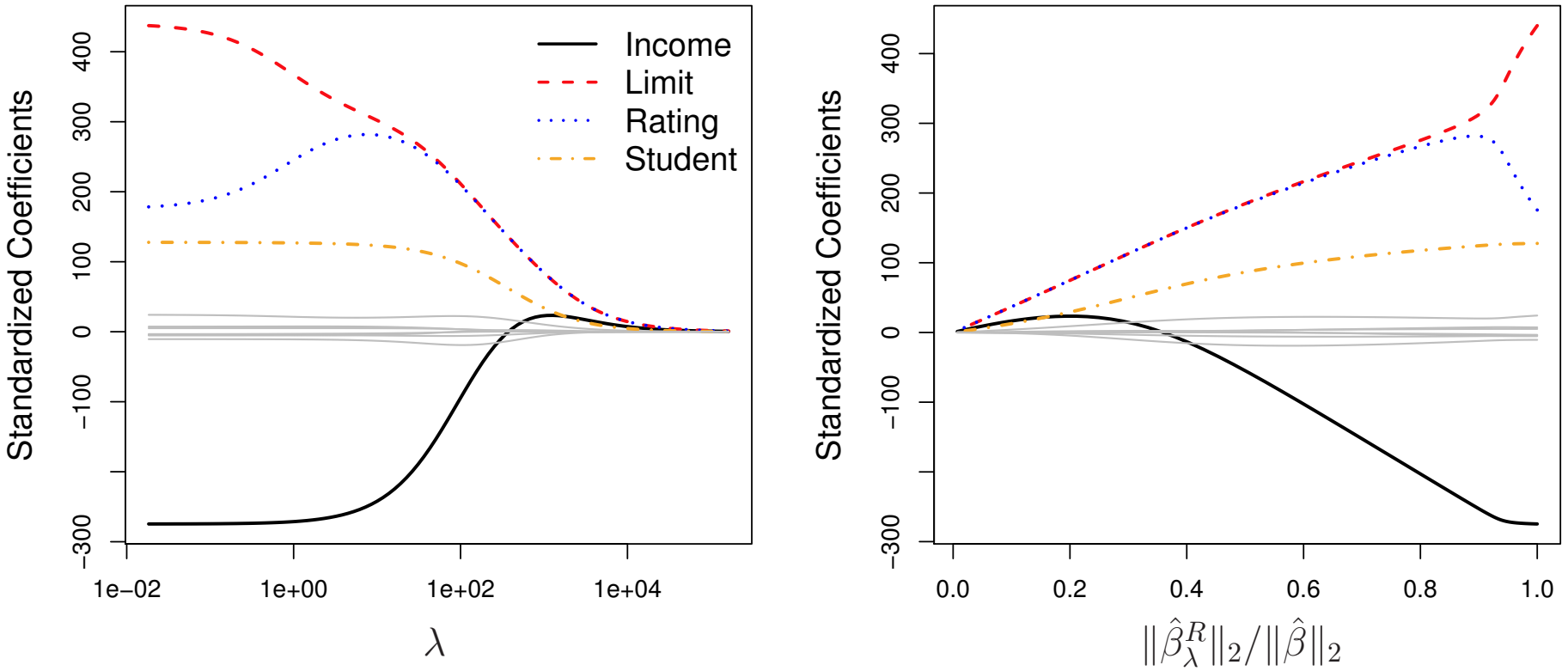
- Tuning λ constitutes tradeoff between two objectives: minimizing RSS (good fit to data) and minimizing **shrinkage penalty** $\lambda\|\boldsymbol{\beta}\|_2^2$, which shrinks $\beta_1, \ldots, \beta_p$ to zero.
- $\lambda = 0$: recover standard LS estimate.
- $\lambda \to \infty$: $\boldsymbol{\beta} \to \mathbf{0}$.
- Different estimate for each value of λ, choice critical.
- Intercept omitted from shrinkage: this is just the mean value of response when all predictor variables are zero.
  Under assumption that all columns of data matrix $\boldsymbol{X}$ have been **centered** to have mean zero, then

$$\hat{\beta}_0 = \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i.$$

- In the following, for the standard linear model, we tacitly assume $\boldsymbol{X}$ to be centered, the coefficient β_0 to be set to its optimal value $\overline{y}$ and the coefficient vector to be estimated to consist of the components $\beta_1, \ldots, \beta_p$.

Linear Model Selection and Regularization

Ridge regression



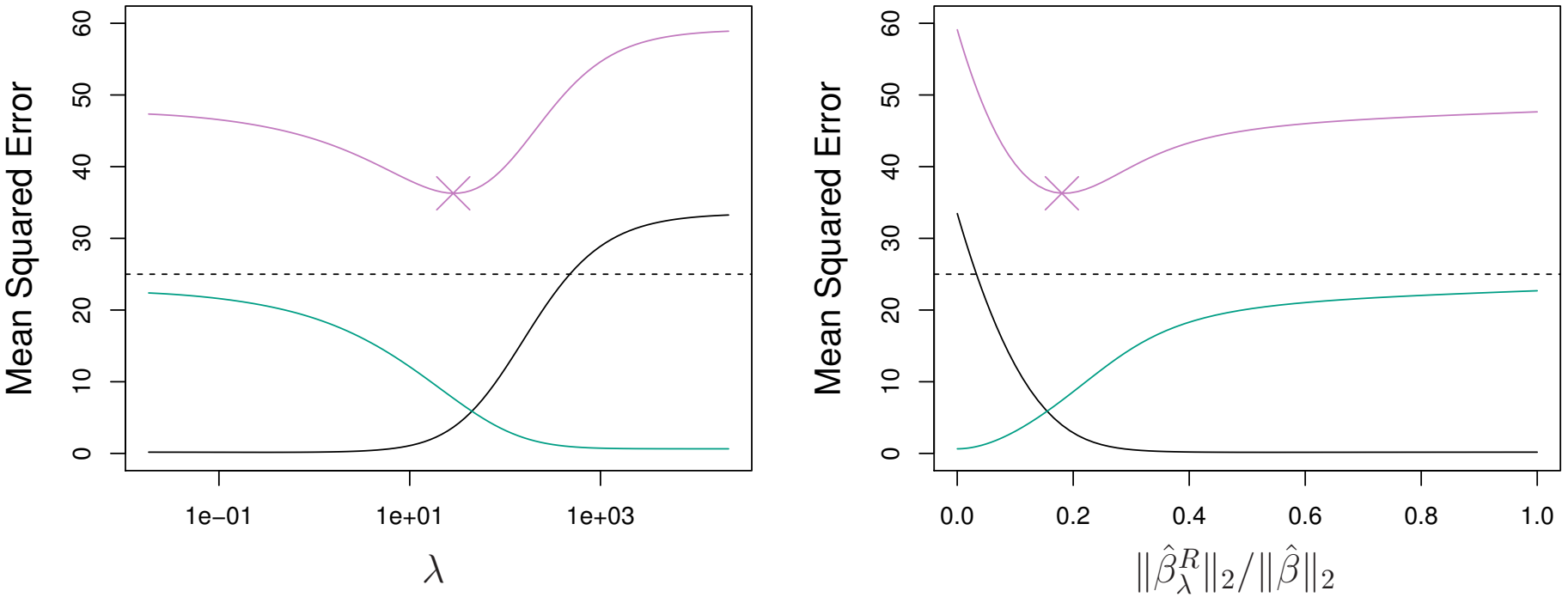
Ridge regression applied to `Credit` data set: values of coefficients of the 10 predictor variables against λ. Lines for largest coefficients `income`, `limit`, `rating` and `student` displayed in distinct colors. Right: x-axis is $\|\hat{\boldsymbol{\beta}}^R_\lambda\|_2/\|\hat{\boldsymbol{\beta}}\|_2$ in place of λ.
Predictor variables standardized before carrying out ridge regression.

- For LS estimation of standard linear model, rescaling a predictor variable $X_j \leftarrow cX_j$ simply results in reciprocal rescaling of estimate as $\hat{\beta}_j \leftarrow \hat{\beta}_j/c$. Consequence: $\hat{\beta}_j X_j$, hence data fit, remains the same. This property is called **scale equivariance**.
- This is no longer the case for ridge regression: value of $\hat{\beta}^R_{j,\lambda} X_j$ depends on λ as well as the scaling of X_j (possibly even the scaling of other predictors).
- Therefore, best to **standardize** predictor variables by transformation

$$x_{i,j} \leftarrow \tilde{x}_{i,j} := \frac{x_{i,j}}{s_j}, \qquad s_j := \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{i,j} - \overline{x}_j)^2}. \tag{6.6}$$

  Denominator s_j estimates variance of j-th predictor.
- Standardized predictor observations have unit variance estimate.

Bias-variance tradeoff: as λ increases, model flexibility decreased, reducing variance, increasing bias.



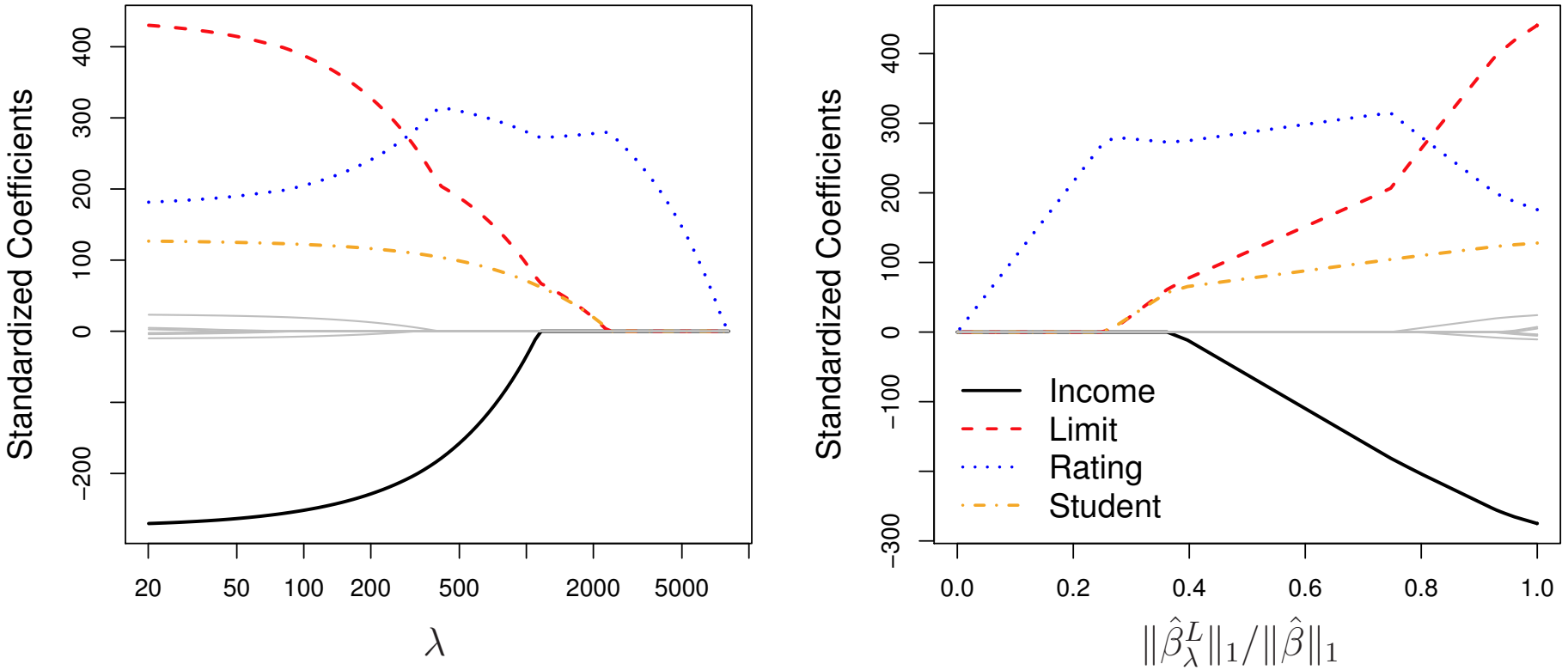
Simulated data, $p = 45$ predictors, $n = 50$ observations. Test MSE (purple), squared bias (black) and variance (green) of ridge regression predictions. Cross: minimal MSE. Dashed line: minimal possible MSE.

- In general: for almost linear (true) model, LS estimate has low bias, but possibly high variance, particularly when p large relative to n.
- For $p > n$ LS fit not unique, but ridge regression still works, trading off slight bias for much reduced variance.
- Computational advantage over best subset selection: ridge regression for many values of λ can be computed at cost of essentially one LS fit, compared to comparing 2^p models.

- Disadvantage of ridge regression: will generally include all p predictors in the model, in contrast with subset selection methods.
- OK for prediction, challenging for interpretation.
- Example: `Credit` data set; most important variables are `income`, `limit`, `rating` and `student`. Model including just these desirable, ridge regression will generally include all 10 predictors.
- **LASSO** (least absolute shrinkage and selection operator): choose coefficients β_j to minimize

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{i,j}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS}+\lambda\|\boldsymbol{\beta}\|_1. \qquad (6.7)$$

- ℓ^2-penalty in ridge regression replaced by ℓ^1-penalty, $\|\boldsymbol{\beta}\|_1 = |\beta_1|+\cdots+|\beta_p|$.
- ℓ^1-penalty: for λ sufficiently large, results in some estimates $\hat{\beta}^L_{j,\lambda}$ being exactly zero, effecting an implicit **variable selection**, yielding in **sparse models**, which are easier to interpret.



LASSO applied to `Credit` data set. Note difference to ridge regression for intermediate values of λ: as λ increases, coefficients are successively set to zero, thereby removed from model.

Can show: ridge regression and LASSO estimates solve **constrained minimization problems**

$$\hat{\boldsymbol{\beta}}_\lambda^L = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \quad \text{subject to } \|\boldsymbol{\beta}\|_1 \leq s \qquad (6.8)$$

and

$$\hat{\boldsymbol{\beta}}_\lambda^R = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \quad \text{subject to } \|\boldsymbol{\beta}\|_2^2 \leq s, \qquad (6.9)$$

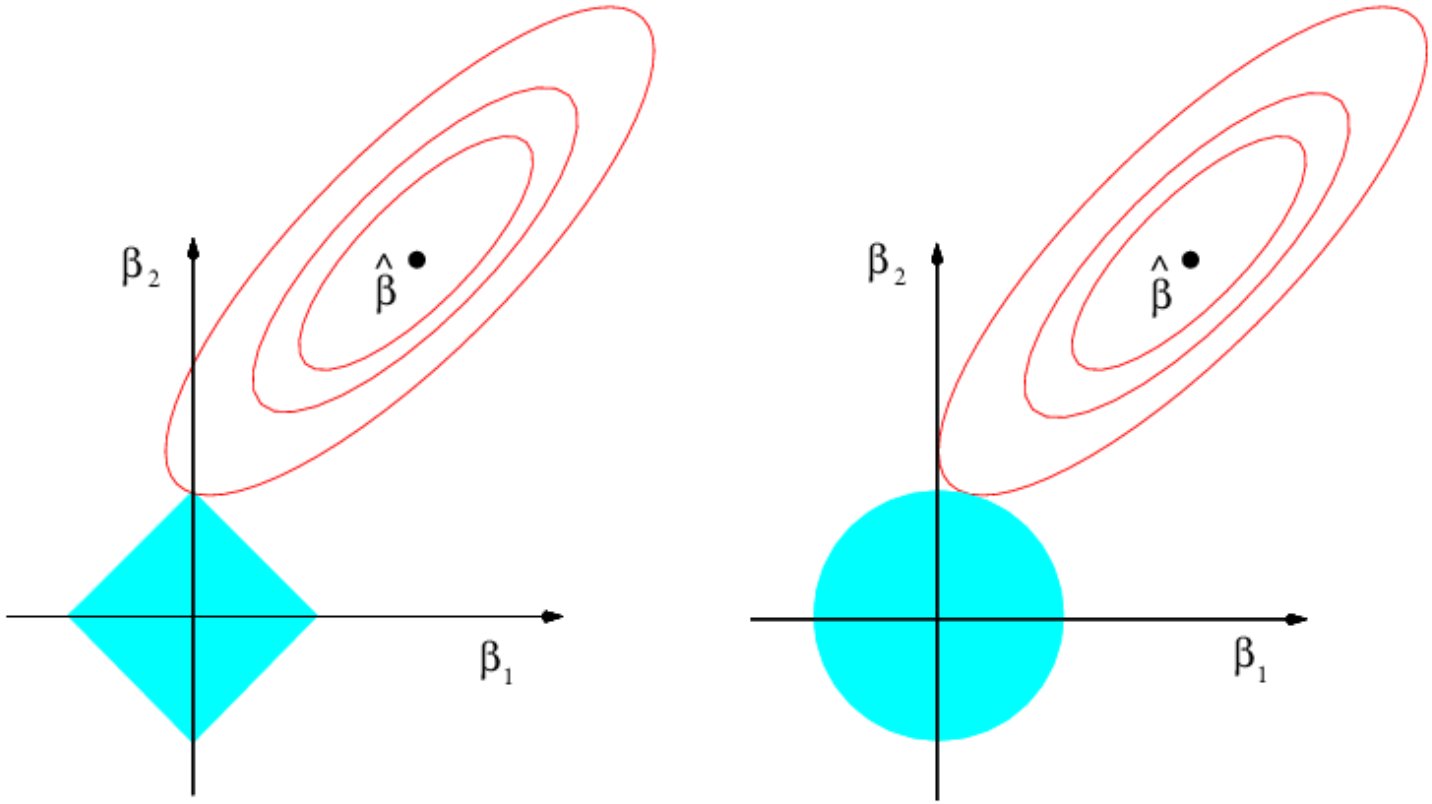
respectively.
In other words: for each value of λ, there is a corresponding value of s, such that both problems give the same estimates.

- Consider constrained minimization problem

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \text{ subject to } \sum_{j=1}^{p} \mathbb{1}_{\{\beta_j \neq 0\}} \leq s \quad (6.10)$$

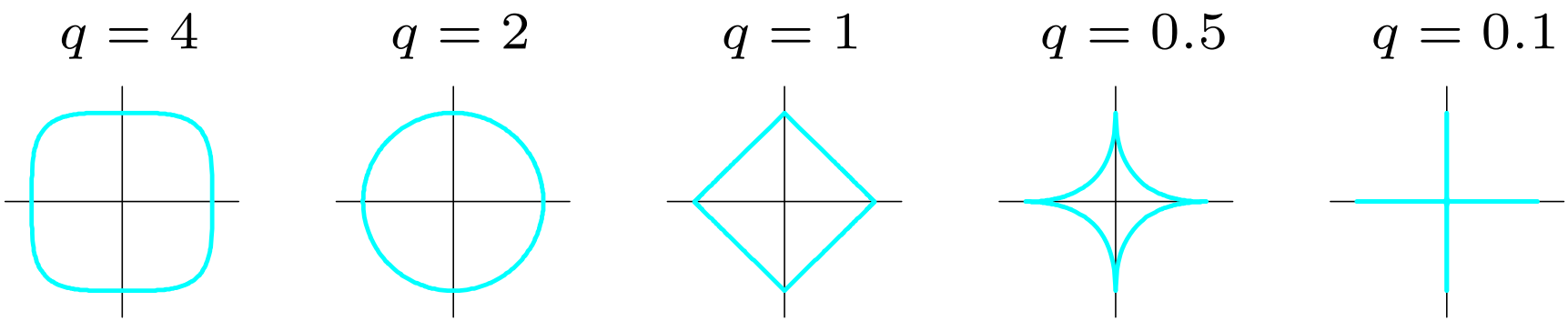
- Minimizes RSS subject to constraint that no more than s coefficients are nonzero.
- This is equivalent to best subset selection.
- Computationally infeasible for large p, since it involves considering all $\binom{p}{s}$ models containing s predictors.
- Hence ridge regression / LASSO computationally feasible alternatives to best subset selection replacing intractable form of budget in (6.10).

- Formulations (6.8) and (6.9) key to understanding variable selection property of LASSO:



Red: RSS contours, blue: constraints $|\beta_1| + |\beta_2| \leq s$ (left) and $\beta_1^2 + \beta_2^2 = s$ (right).

- Unit spheres of $\sum_{j=1}^{p} |\beta_j|^q$ for $q < 2$ progressively sharper (no longer a norm for $q < 1$).

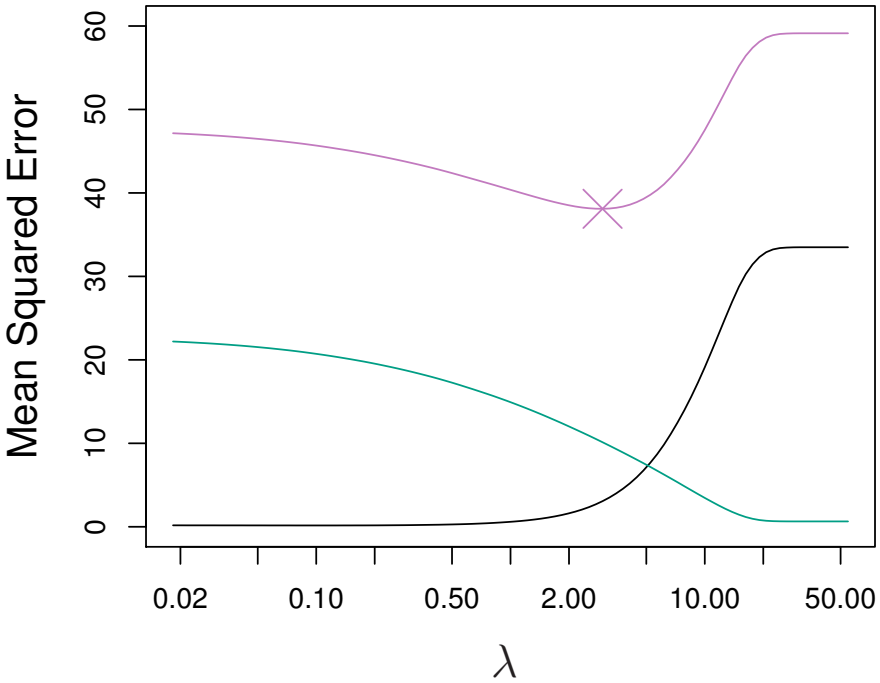


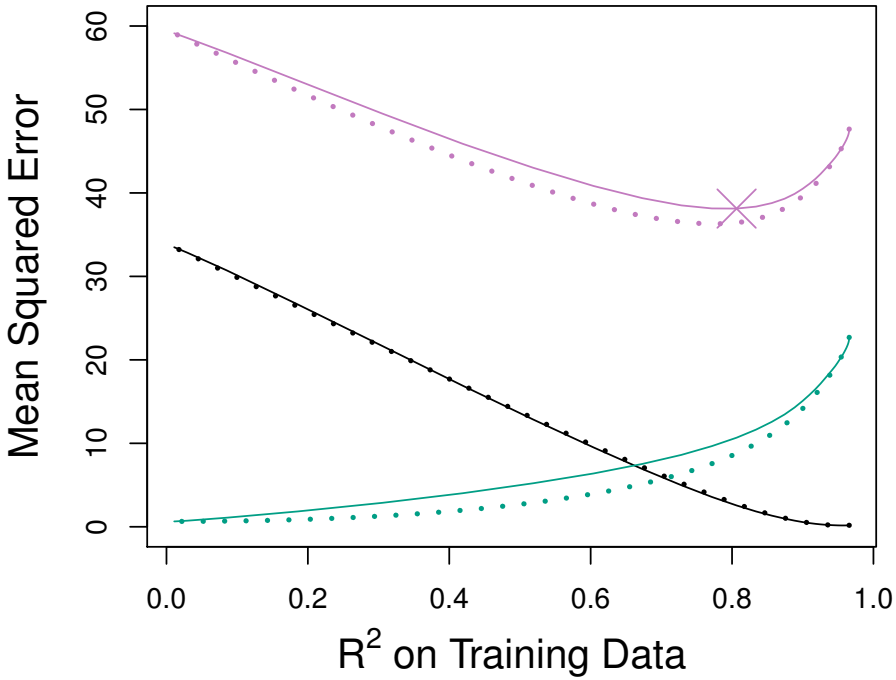
- Limiting case: $q = 0$ counts # nonzero components.

Linear Model Selection and Regularization

Comparison of ridge regression with LASSO

Simulated data using all $p = 45$ predictors: ($\beta_j \neq 0 \; \forall j$ in true model)





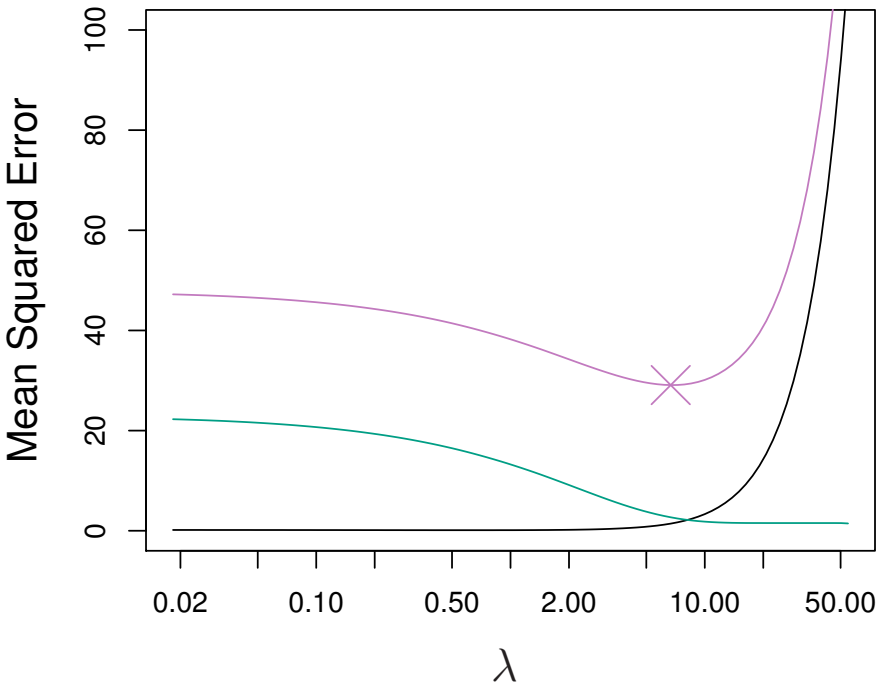
Left: Test MSE (purple), squared bias (black) and variance (green) of LASSO for different values of λ.

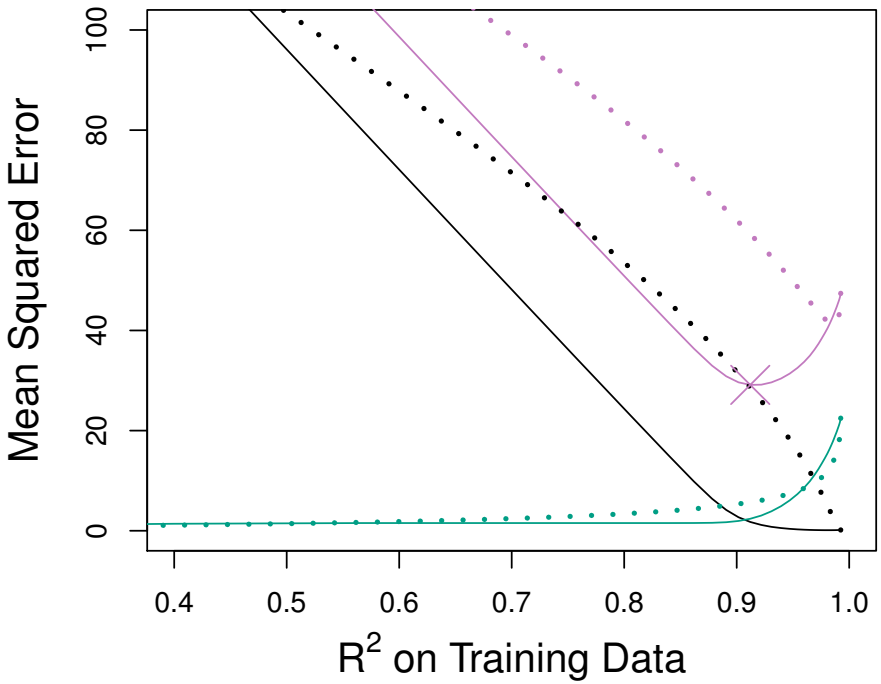
Right: Comparison of test MSE (purple), squared bias (black) and variance (green) against training R^2; dotted lines denote corresponding quantities for ridge regression.

Linear Model Selection and Regularization

Comparison of ridge regression with LASSO

Simulated data using only 2 out of $p = 45$ predictors: (only two $\beta_j \neq 0$ in true model)





Left: Test MSE (purple), squared bias (black) and variance (green) of LASSO for different values of λ.

Right: Comparison of test MSE (purple), squared bias (black) and variance (green) against training R^2; dotted lines denote corresponding quantities for ridge regression.

Assume data matrix $\boldsymbol{X} = \boldsymbol{I}$ $(p = n)$ and $\overline{y} = 0$.
LS problem reduces to minimizing

$$\sum_{j=1}^{p}(y_j - \beta_j)^2, \qquad \text{hence } \beta_j = y_j,\ j = 1, \ldots, p. \tag{6.11}$$

Ridge regression and lasso estimation result from minimizing

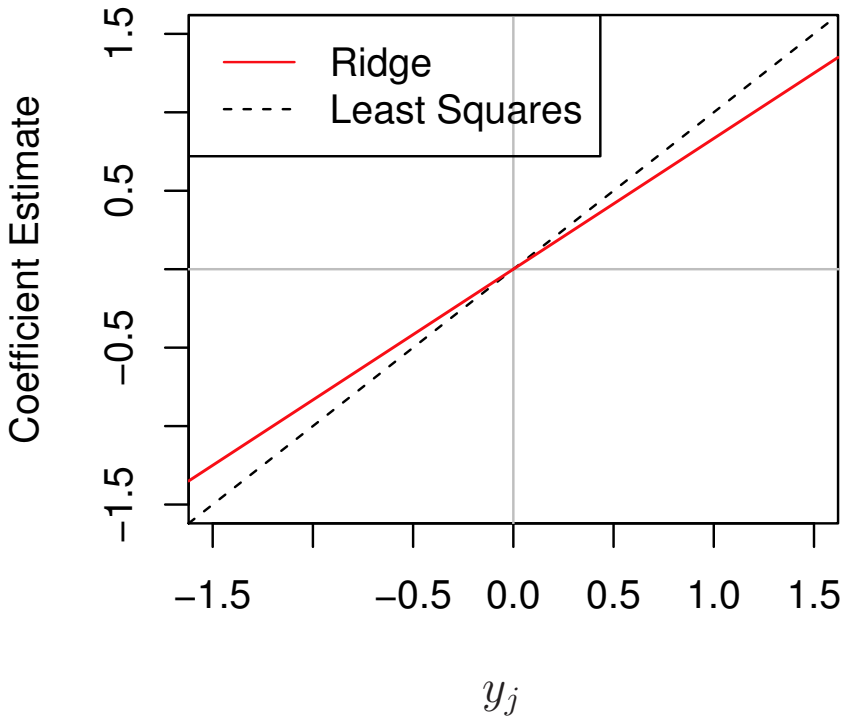
$$\sum_{j=1}^{p}(y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \quad \text{and} \quad \sum_{j=1}^{p}(y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|,$$

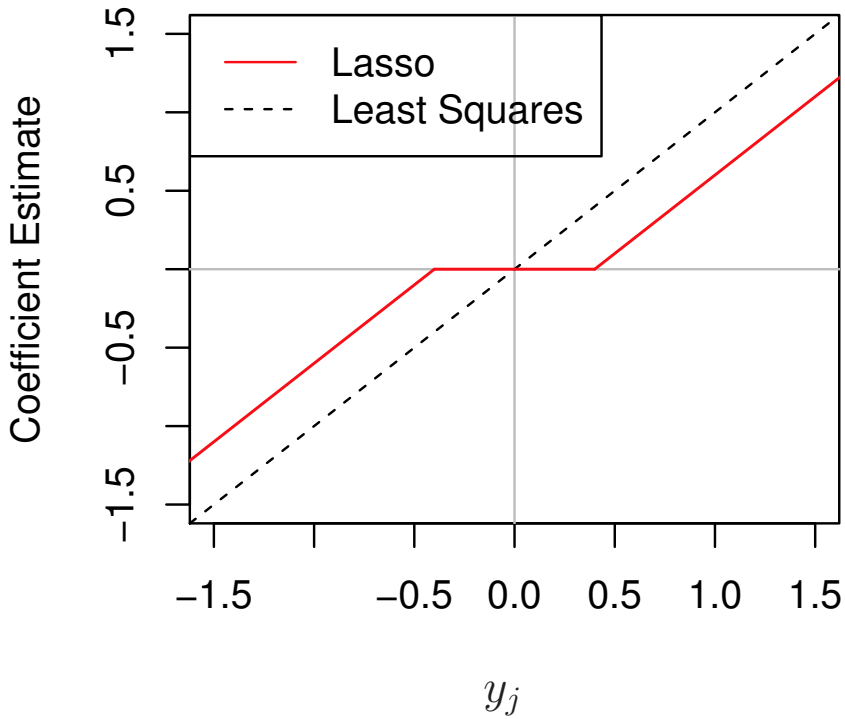
respectively, with solutions

$$\hat{\boldsymbol{\beta}}^R = \frac{y_j}{1+\lambda}, \qquad \hat{\boldsymbol{\beta}}^L = \begin{cases} y_j - \lambda/2 & \text{if } y_j > \lambda/2, \\ y_j + \lambda/2 & \text{if } y_j < -\lambda/2, \\ 0 & \text{if } |y_j| \le \lambda/2. \end{cases}$$

Linear Model Selection and Regularization

Simple special case for ridge regression and the lasso





Ridge regression (left) and lasso (right) estimates for one variable of special case $\boldsymbol{X} = \boldsymbol{I}$ and $p = n$.

General case: more complicated (of course), but basic mechanism still holds:

- Ridge regression: shrinks every dimension roughly by same proportion.
- Lasso: shrinks all components to zero by similar amount, sufficiently small coefficients damped to zero.

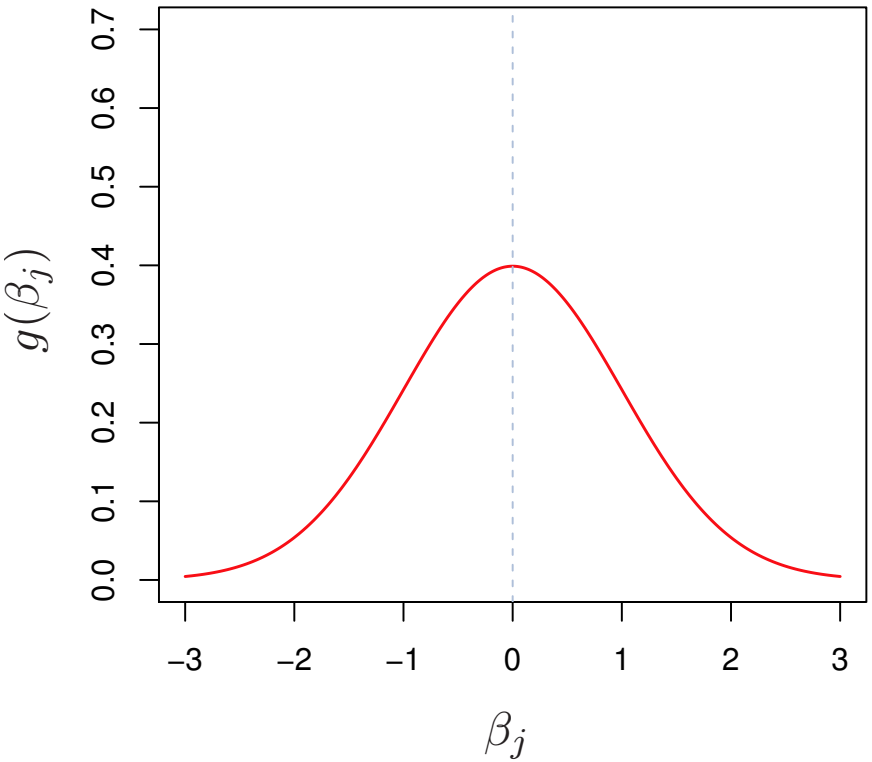
- Assume prior distribution on $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$, with density $p(\boldsymbol{\beta})$.
- Likelihood of data: $f(Y|X, \boldsymbol{\beta})$, $X = (X_1, \ldots, X_p)$.
- Bayes' rule then says (noting X is fixed)

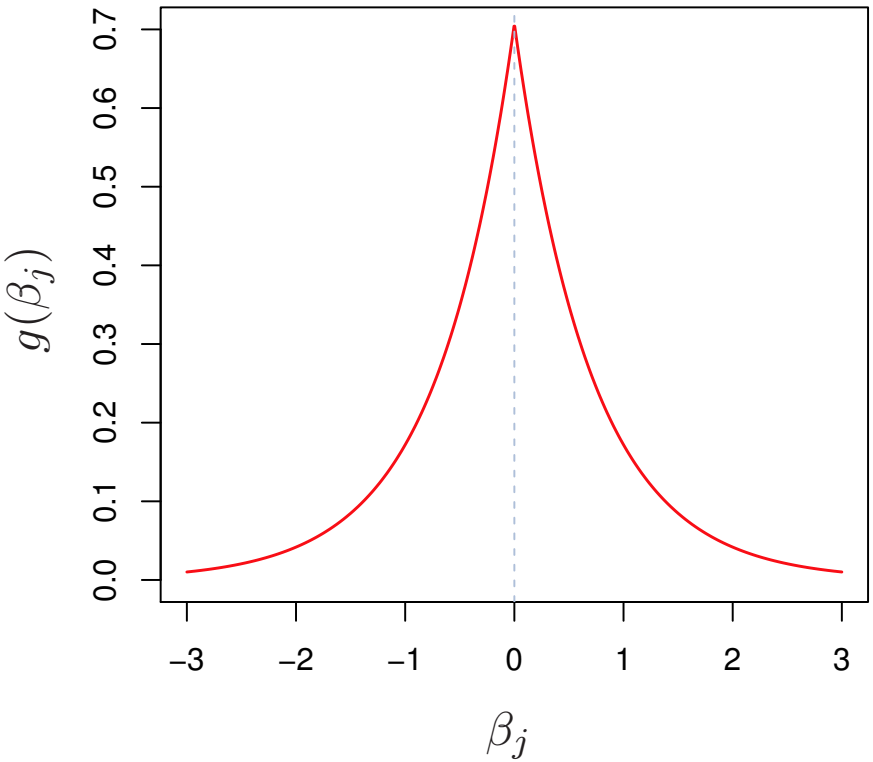
$$p(\boldsymbol{\beta}|X, Y) \propto f(Y|X, \boldsymbol{\beta}) \cdot p(\boldsymbol{\beta}|X) = f(Y|X, \boldsymbol{\beta}) \cdot p(\boldsymbol{\beta}).$$

- Assume standard linear model $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$, with independent Gaussian noise and $p(\boldsymbol{\beta}) = \prod_{j=1}^{p} g(\beta_j)$ for pdf g.
- Ridge regression/lasso results from two special cases for g:
  - g centered Gaussian, λ-dependent variance, then ridge regression estimate is **posterior mode** (and posterior mean) of $\boldsymbol{\beta}$.
  - g centered Laplace distribution with λ-dependent scale parameter, then posterior mode for $\boldsymbol{\beta}$ given by lasso estimate. (Not posterior mean; posterior mean itself not sparse.)

Linear Model Selection and Regularization

Bayesian interpretation for ridge regression and the lasso





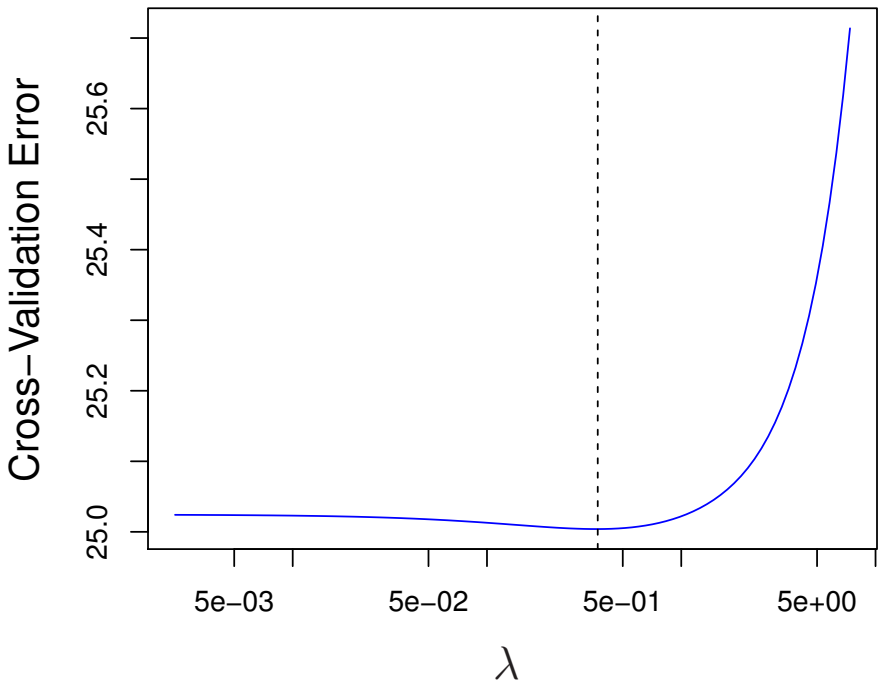
Prior densities for Bayesian interpretation of shrinkage methods.
Left: centered Gaussian prior density, results in posterior distribution with ridge regression solution as posterior mode.

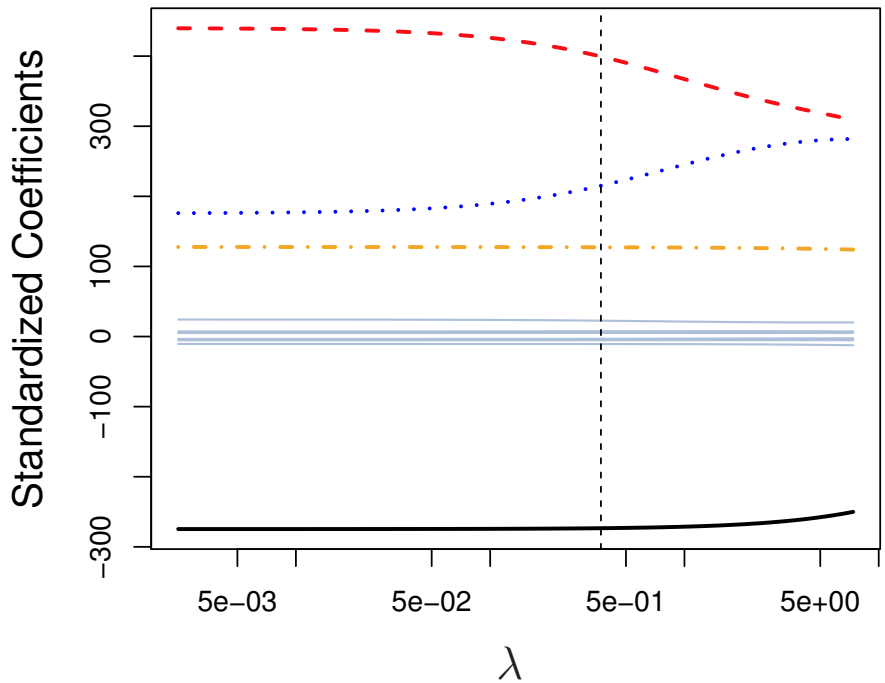
Right: centered Laplace (double-exponential) prior density, results on lasso solution as posterior mode.

Linear Model Selection and Regularization

Selection of λ

- Model selection methods required measure of goodness to compare models.
- Shrinkage methods require selection of shrinkage parameter λ.
- **Cross-validation** approach: fix a grid of λ values; compute cross-validation error for each λ; select λ with smallest error; refit this model with all available observations.



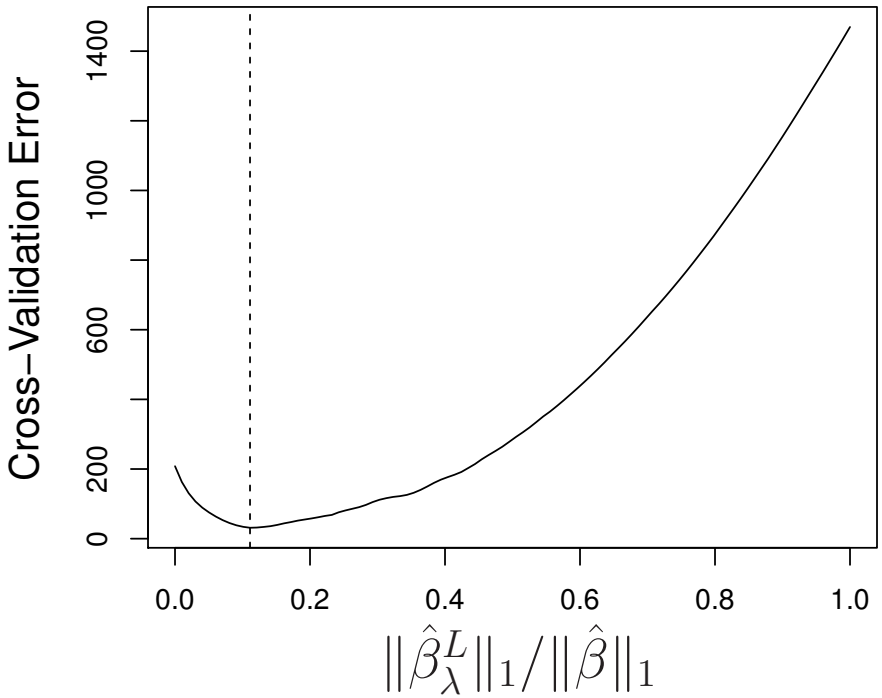


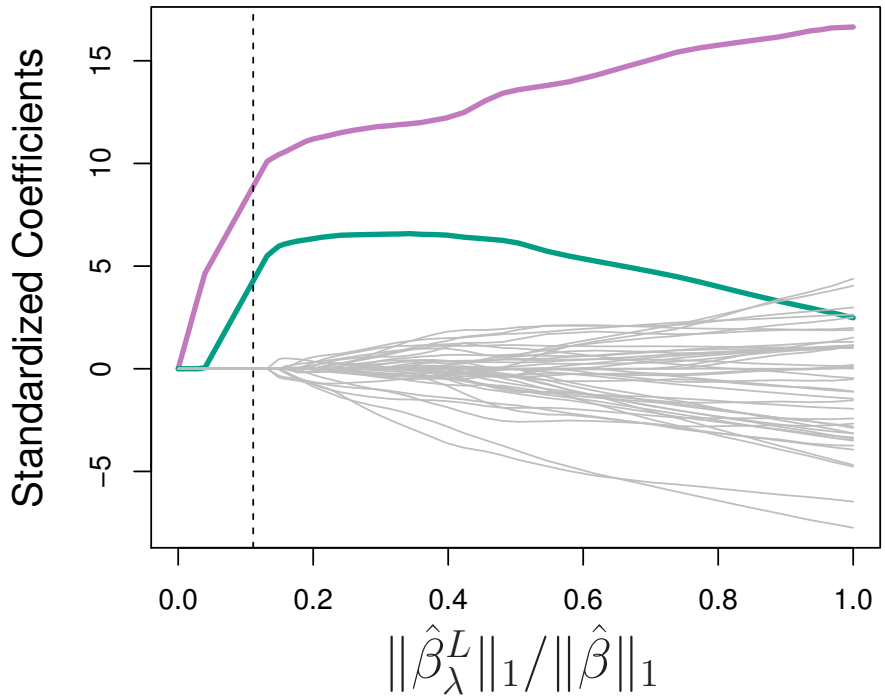
Left: LOOCV errors vs. λ for ridge regression applied to `Credit` data set.

Right: Coefficient estimates vs. λ. Vertical dashed line indicates selected λ.

Linear Model Selection and Regularization

Selection of λ





10-fold CV applied to data set from Slide 281.

Left: CV error. Right: coefficient estimates. Vertical dashed line indicates CV error-minimizing λ. Colored lines represent 2 predictors related to response, grey lines unrelated predictors (**signal** vs. **noise**).

Lasso assigns relevant predictors much larger estimates; CV chooses λ for which irrelevant predictors set to zero. Compare LS estimate (far right).

Contents

- Up to now: control variance by removing predictor variables or shrinking coefficients.
- Now: reduce variance by projecting into subspace of dimension $M < p$.
- Set

$$Z_m := \sum_{j=1}^{M} \phi_{j,m} X_j, \ m = 1, \ldots, M, \qquad \text{i.e.,} \ \ \boldsymbol{Z} = \boldsymbol{X}\boldsymbol{\Phi}, \ \boldsymbol{\Phi} \in \mathbb{R}^{p \times M}. \tag{6.12}$$

- Fit standard linear regression model

$$Y = \theta_0 + \theta_1 Z_1 + \cdots + \theta_M \beta_M + \varepsilon. \tag{6.13}$$

- **Dimension reduction**: fit $M + 1 < p + 1$ coefficients.
- For well-chosen $\boldsymbol{\Phi}$, this reduced-dimension approach can outperform LS.

- Note:

$$\sum_{m=1}^{M} z_{i,m}\theta_m = \sum_{m=1}^{M}\sum_{j=1}^{p} \phi_{j,m}x_{i,j}\theta_m = \sum_{j=1}^{p} x_{i,j}\sum_{m=1}^{M} \phi_{j,m}\theta_m = \sum_{j=1}^{p} x_{i,j}\beta_j$$

  with $\beta_j := \sum_{m=1}^{M} \phi_{jk,m}\theta_m$.

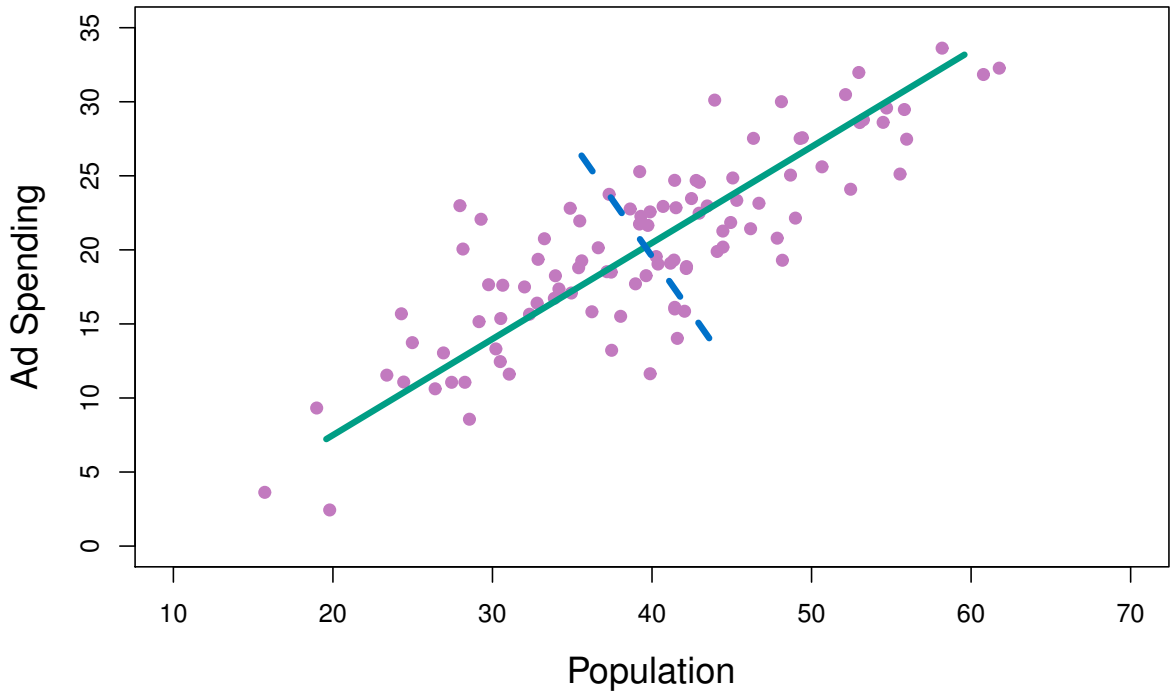
- In matrix terms:

$$\boldsymbol{Z\theta} = \boldsymbol{X\Phi\theta} =: \boldsymbol{X\beta}, \qquad \boldsymbol{\beta} := \boldsymbol{\Phi\theta}$$

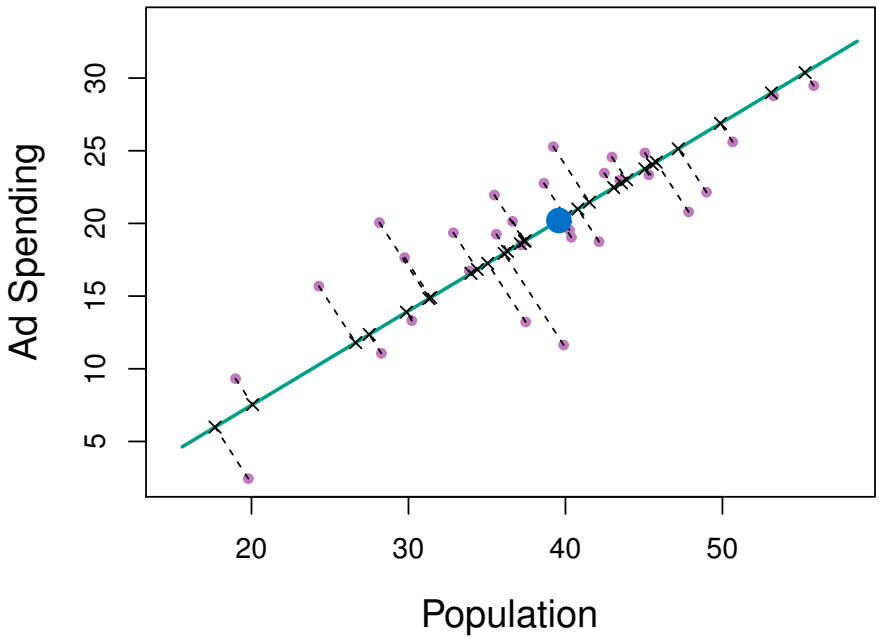
  Hence can view (6.13) as special case of original linear model (6.1).

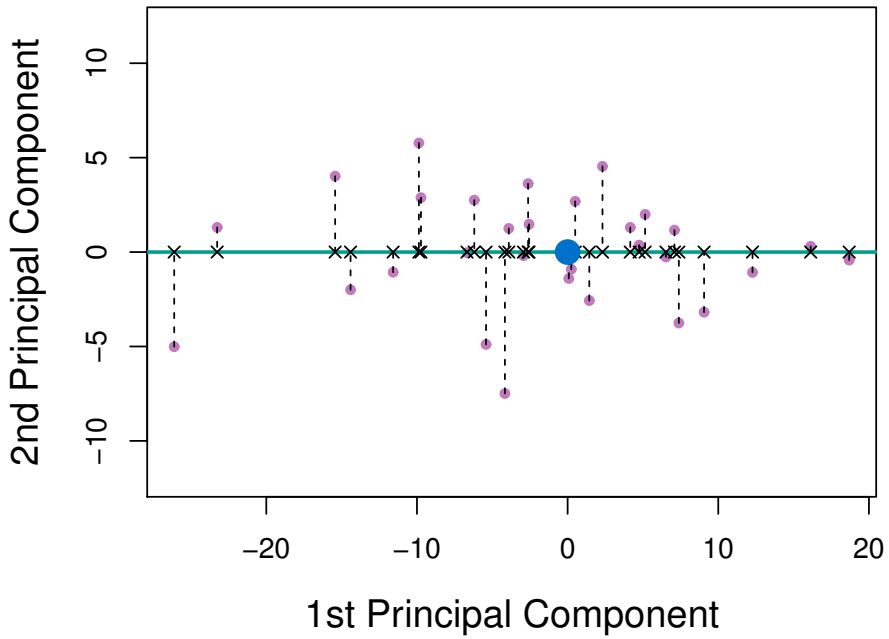
- Dimension reduction constrains $\boldsymbol{\beta}$ by making it a linear function of $M < p$ variables $\{\theta_m\}_{m=1}^{M}$.
- May introduce bias, but when $p \gg n$ this is outweighed by resulting variance reduction.
- Next: 2 ways of choosing $\boldsymbol{\Phi}$.

- **Principal components analysis** (PCA): approach for deriving a low-dimensional feature set from a large set of variables.
- First principal component: direction in $\mathbb{R}^p$ in which observations vary the most.



Population size `pop` vs. ad spending `ad` for 100 cities (purple dots). Green solid line: first principal component; blue dashed line: second principal component.





- Project data on direction (line) along which it varies most.
- For `pop` / `ad` data: $\phi_{1,1} = 0.839$, $\phi_{2,1} = 0.544$, giving

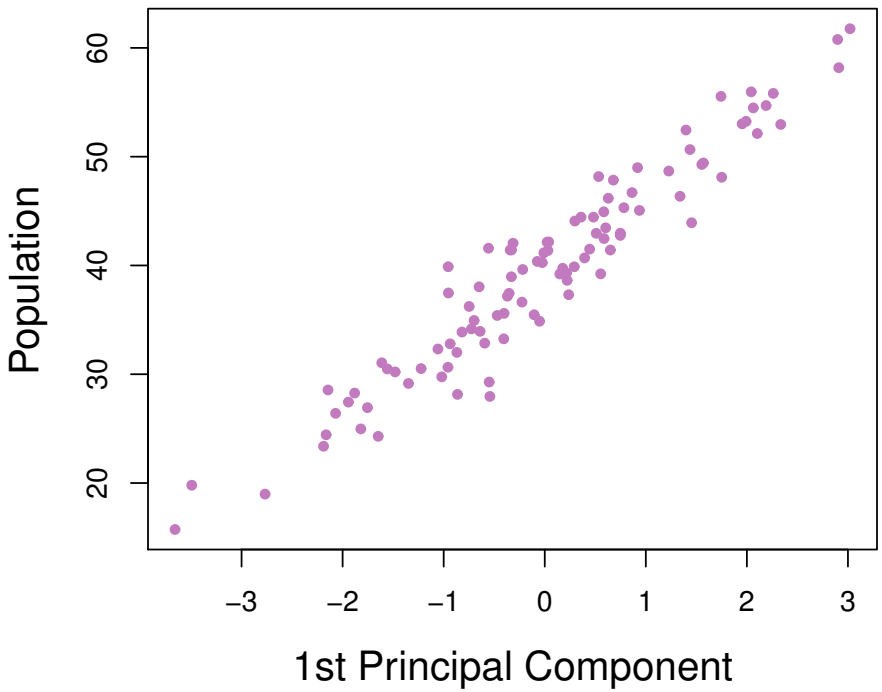
$$Z_1 = 0.839 \times (\texttt{pop} - \overline{\texttt{pop}}) + 0.544 \times (\texttt{ad} - \overline{\texttt{ad}})$$

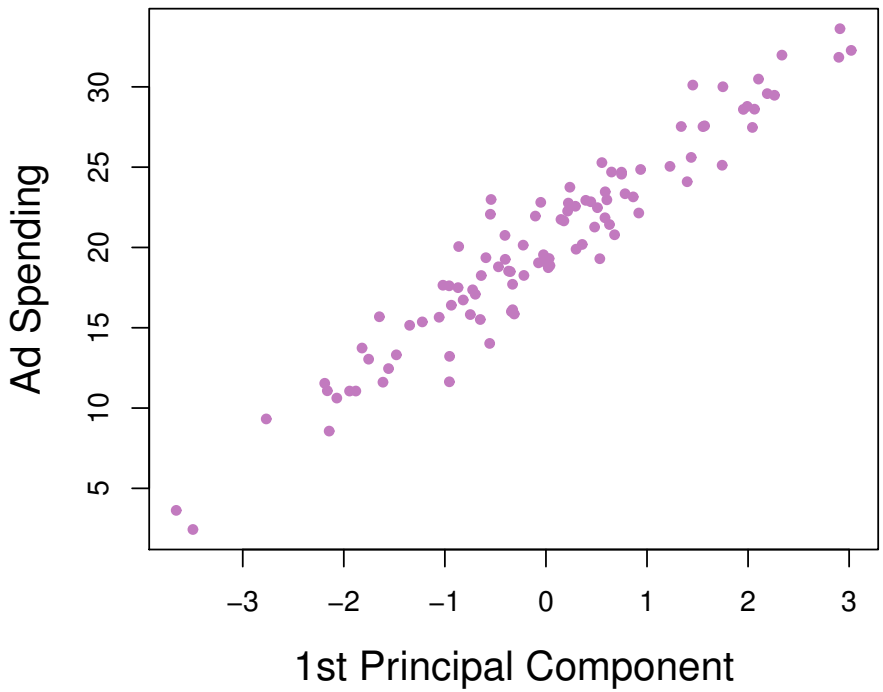
- Out of every (normalized) linear compbination of the (centered) variables, Z_1 has maximal variance.

- Principal component data vector ("**scores**") has same length n, e.g.

$$z_{i,1} = 0.839 \times (\mathtt{pop}_i - \overline{\mathtt{pop}}) + 0.544 \times (\mathtt{ad}_i - \overline{\mathtt{ad}}), \qquad i = 1, \ldots, n.$$

- Alternative interpretation of PCA: 1st principal component vector defines line as close as possible to data in sense of minimizing sum of squared perpendicular distances between each data point and this line.
-



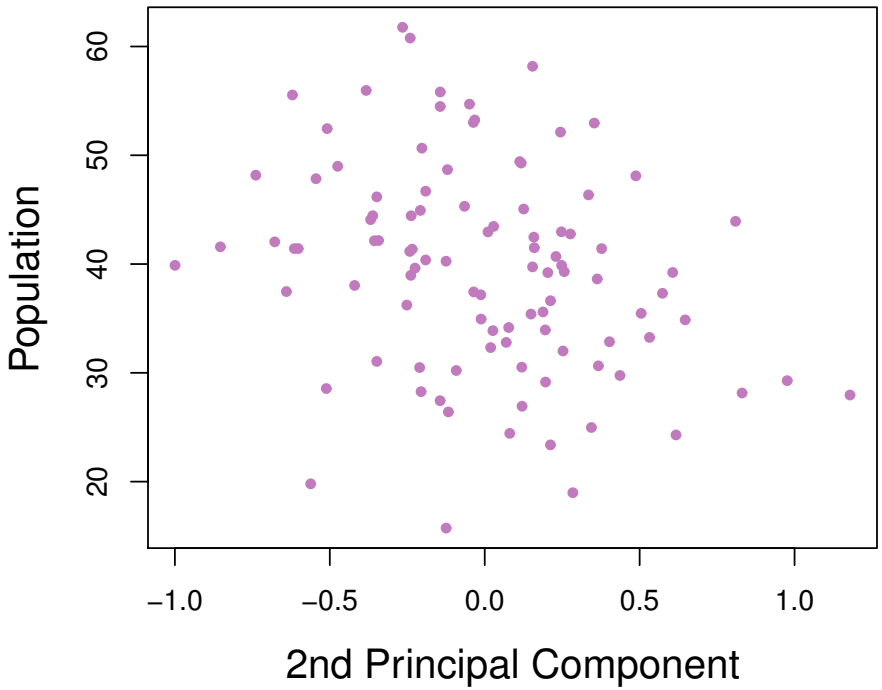


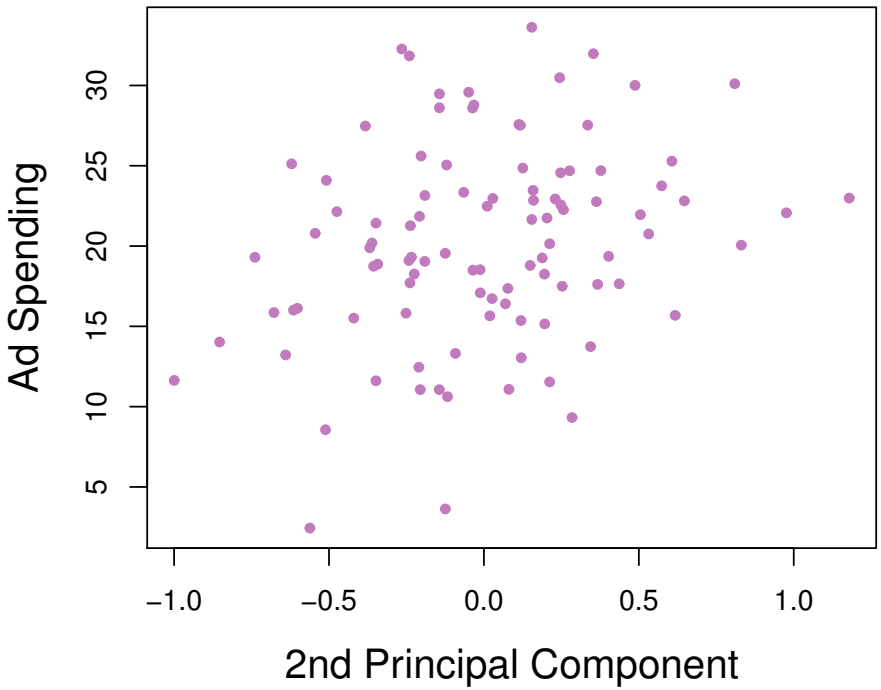
First principal components scores $z_{i,1}$ for `pop` and `ad`. Strong relationship in both cases, i.e., principal component captures most of the information contained in the two predictors.

- Second principal component Z_2: direction of largest variance among all linear combinations of predictor variables which is *orthogonal* to (uncorrelated with) Z_1.
- Here:
$$Z_2 = 0.544 \times (\texttt{pop} - \overline{\texttt{pop}}) - 0.839 \times (\texttt{ad} - \overline{\texttt{ad}}).$$
Since $p = 2$, this covers all of remaining variance.
- Of these, Z_1 contains most of the information, cf. much larger variation in Z_1-coordinate than Z_2-coordinate in right panel of figure on Slide 292.
- Plot on Slide 296 displays $z_{i,2}$ against `pop` and `ad` predictors: much less relationship than with Z_1.
Thus, Z_1 sufficient to explain most of variability in data set.
- For p predictor variables, can construct up to p principal components.

Linear Model Selection and Regularization

Principal components regression

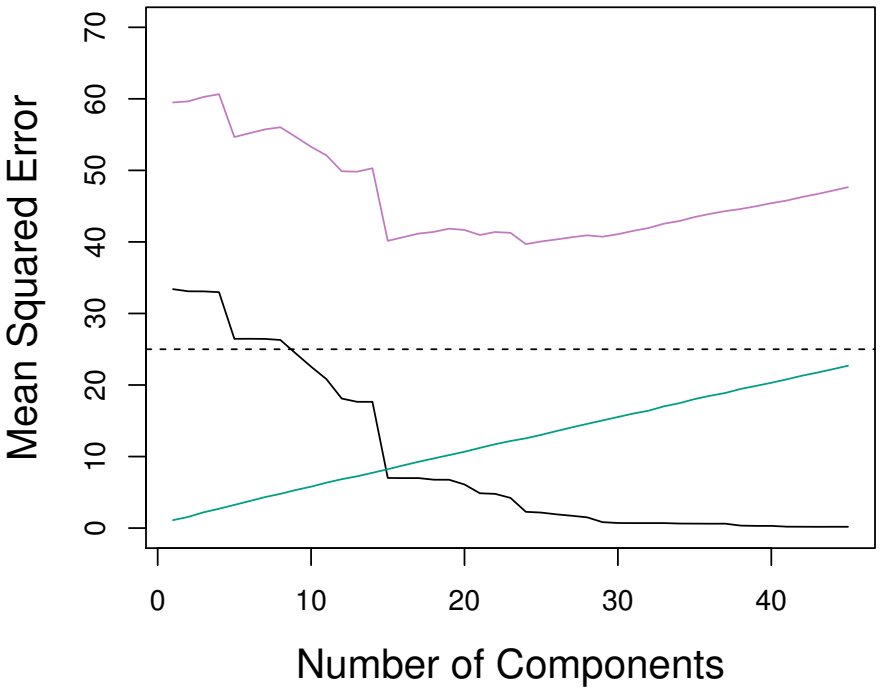


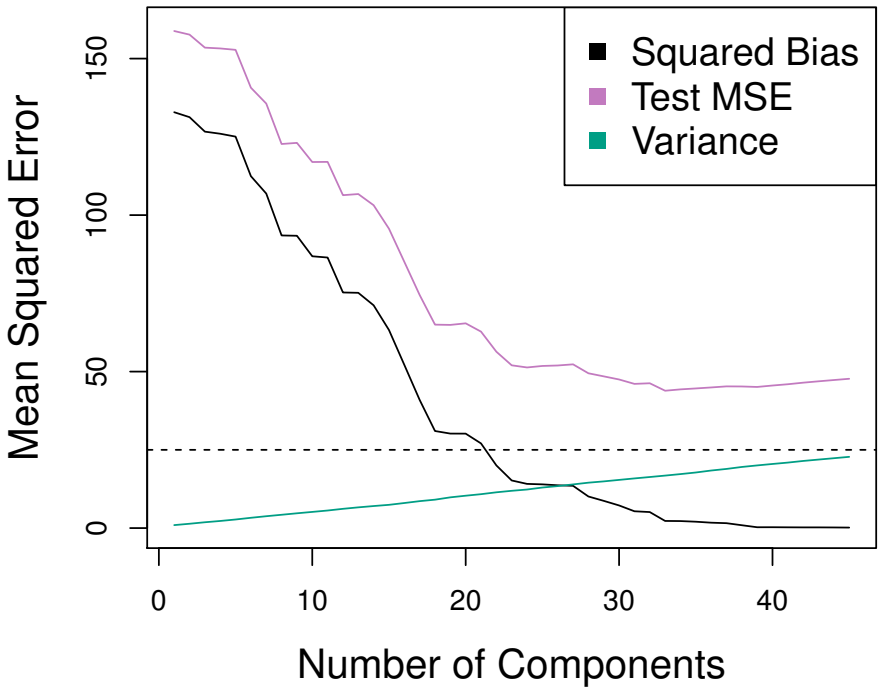


- **Principal components regression** (PCR): construct first M proincipal components $Z_1, \ldots, Z_M$, use these in a linear regression model fit by LS.
- Guiding principle: directions in span of $X_1, \ldots, X_p$ with most variance are the directions associated with response Y.
- Under this assumption, fitting LS model to $Z_1, \ldots, Z_M$ will yield better predictions than fitting $X_1, \ldots, X_p$, since most information related to response Y contained in $Z_1, \ldots, Z_M$, and estimating $M \ll p$ coefficients avoids overfitting.

Linear Model Selection and Regularization

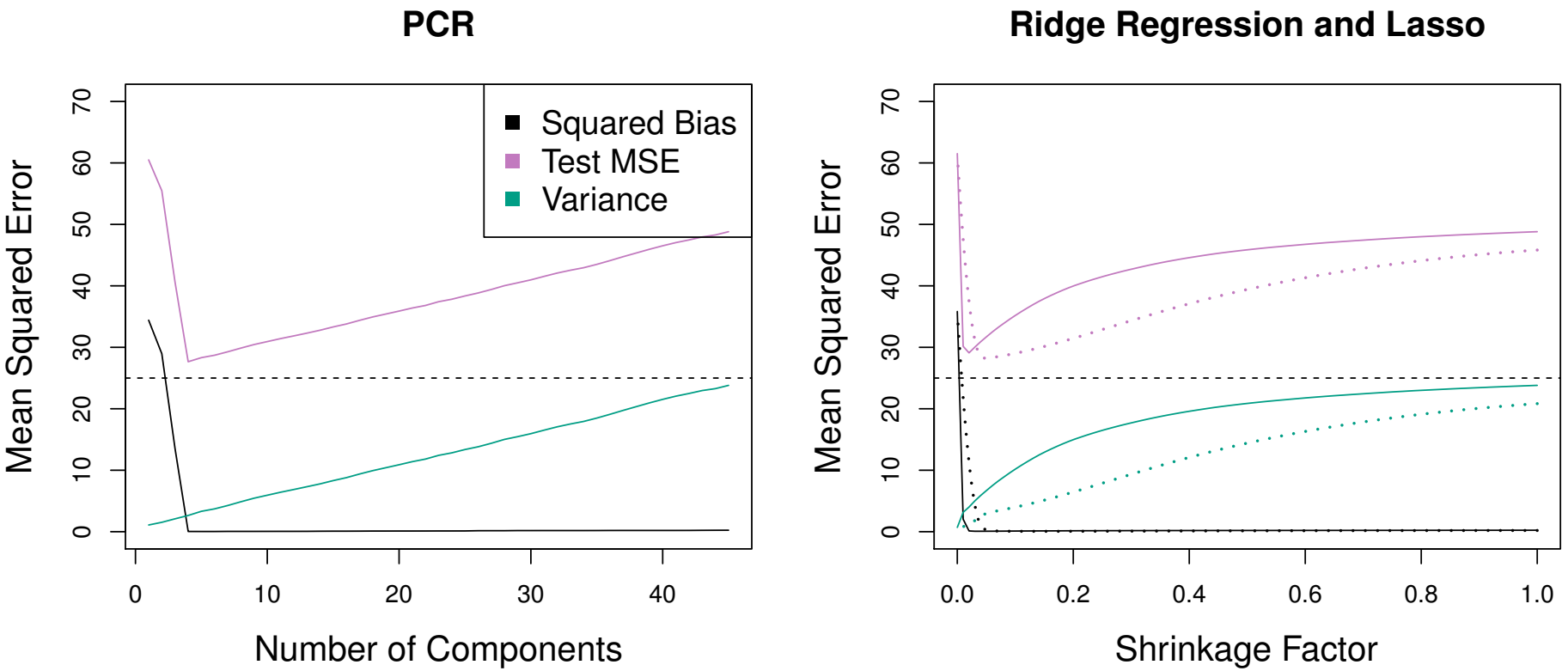
Principal components regression





PCR fits to data sets from Slide 280 (left) and Slide 281 (right): MSE against # proncipal components M. More components reduces bias, increases variance (U-shape). $p = M$ coincides with LS fit of original predictors. Compared with ridge regression and lasso results in figures on Slides 272, 280 and 281, PCR seen to underperform shrinkage.

Worse performance of PCR in previous example due to fact that many principal components needed needed to explain response.



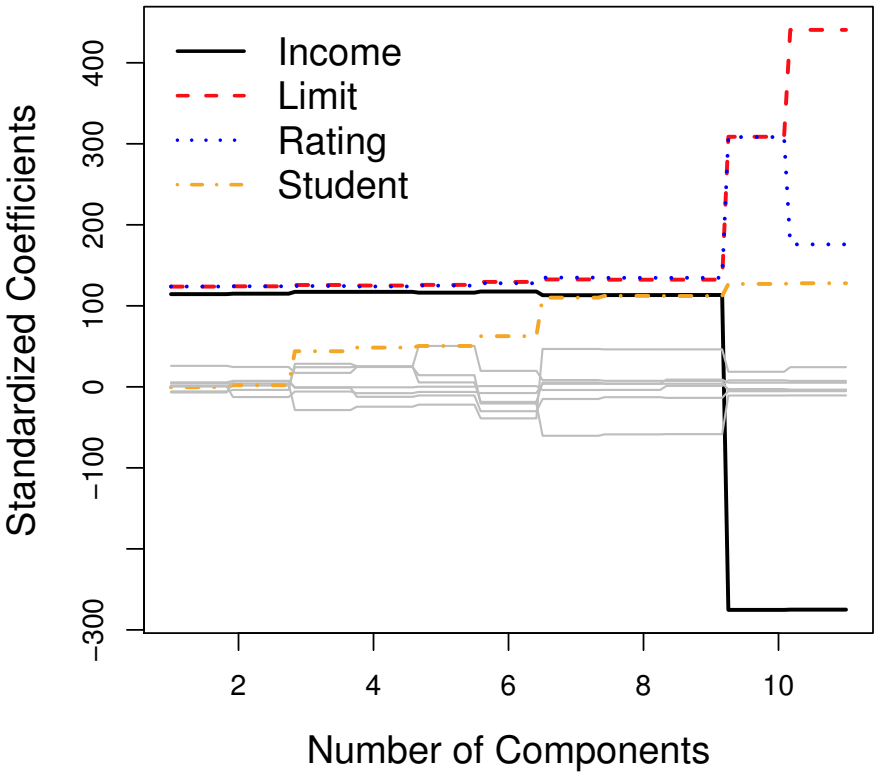
Data generated in such a way that response depends exclusively on first 5 principal components. Left: PCR, MSE has clear minumum at $M = 5$.

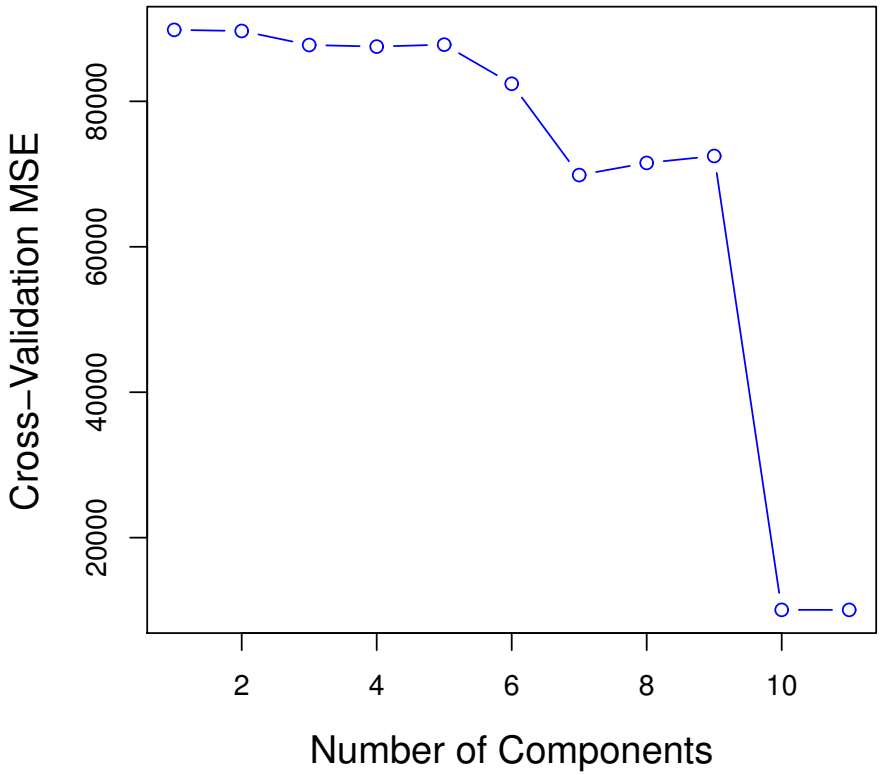
Right: ridge regression (dotted) and lasso (solid) results.

- PCR uses $M < p$ new variables, but these all still depend on original predictors.
- Hence, PCR not a feature selection method.
- In this aspect, PCR closer to ridge regression than lasso.
- Ridge regression can be viewed as a continuous version of PCR.
- # principal components M can be chosen by CV.
- Recommended: first standardize data.

Linear Model Selection and Regularization

Principal components regression



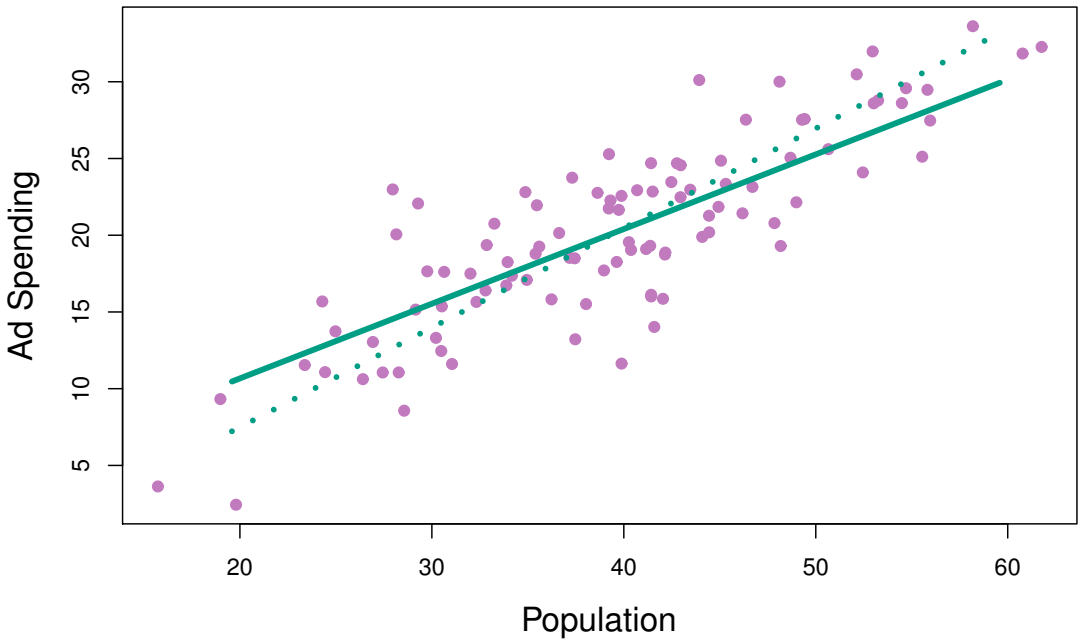


PCR applied to `Credit` data set.

Left: standardized coefficients. Right: CV MSE against M.

Lowest error for 10 components (only one less than full model).

- PCR only looks at predictor variability, not at response.
- In this sense, it is **unsupervised**.
- **Partial least squares** (PLS): **supervised** variant of PCR: find linear combination of predictors containing most variability and best explain response.
- To construct Z_1, set each coefficient in $Z_1 = \sum_{j=1}^{p} \phi_{j,1} X_j$ to coefficient of simple linear regression of Y onto X_j. Results in coefficient proportional to $\mathsf{Cor}(X_j, Y)$.
  This places highest weight on variables most strongly related to response Y.
- To identify Z_2, first adjust all predictors for Z_1 by regressing these on Z_1 and taking residuals. Interpretation: remaining information not explained by first PLS direction. Compute Z_2 using this orthogonalized data just as Z_1 was computed using original data.
- In the same way, compute further PLS directions $Z_3, \ldots, Z_M$.

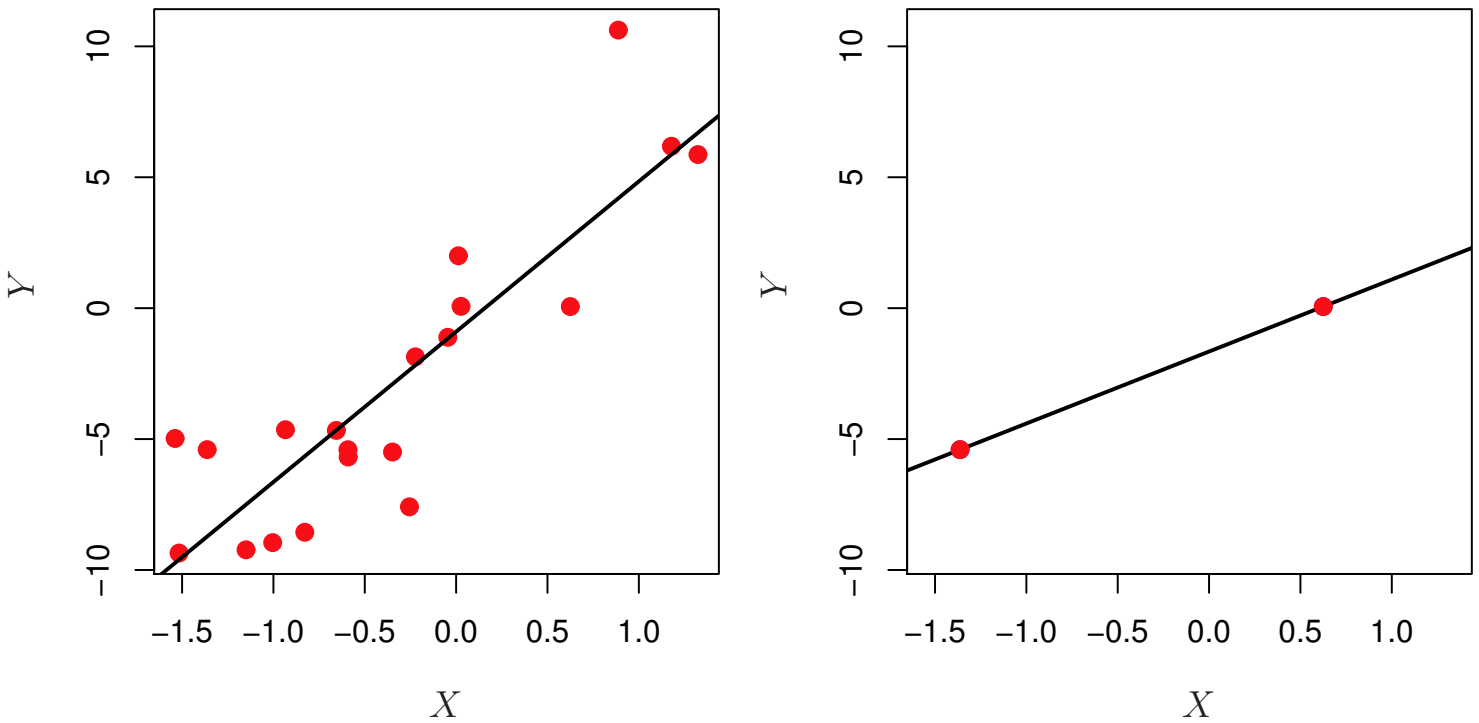


PLS on synthetic data set giving `Sales` data in each of 100 regions as response to two predictors `Population Size` and `Advertising Spending`. Solid line: first PLS direction, dotted: first principal components direction.

Contents

- Most traditional statistical techniques: $n \gg p$ (**low-dimensional setting**).
- Typical example: Predict patient's blood pressure based on age, gender, body mass index (BMI). Three predictors, and typically thousands of patients' data.
- More recently, in many fields such as medicine, finance, marketing, trend towards collecting almost unlimited number of feature measurements (p large), while cost of obtaining sufficienly many samples prohibitive.
- Example: in place of age, gender, BMI, collect measurements of half million **single nucleotide polymorphisms**, i.e., common individual DNA mutations. Results in $p \approx 500,000$, $n \approx 200$.
- Example: 'Bag-of-words' model to understand customers' online shopping patterns, using as features all search terms entered in search engine (binary feature vector). Only few hundred users consented to their data being used. Results in $n \approx 100$, p much larger.

- When $p \geq n$ LS cannot (should not) be used, since data will be fit perfectly.
- Example: $p = 1$ $n = 20$ vs. $n = 2$:

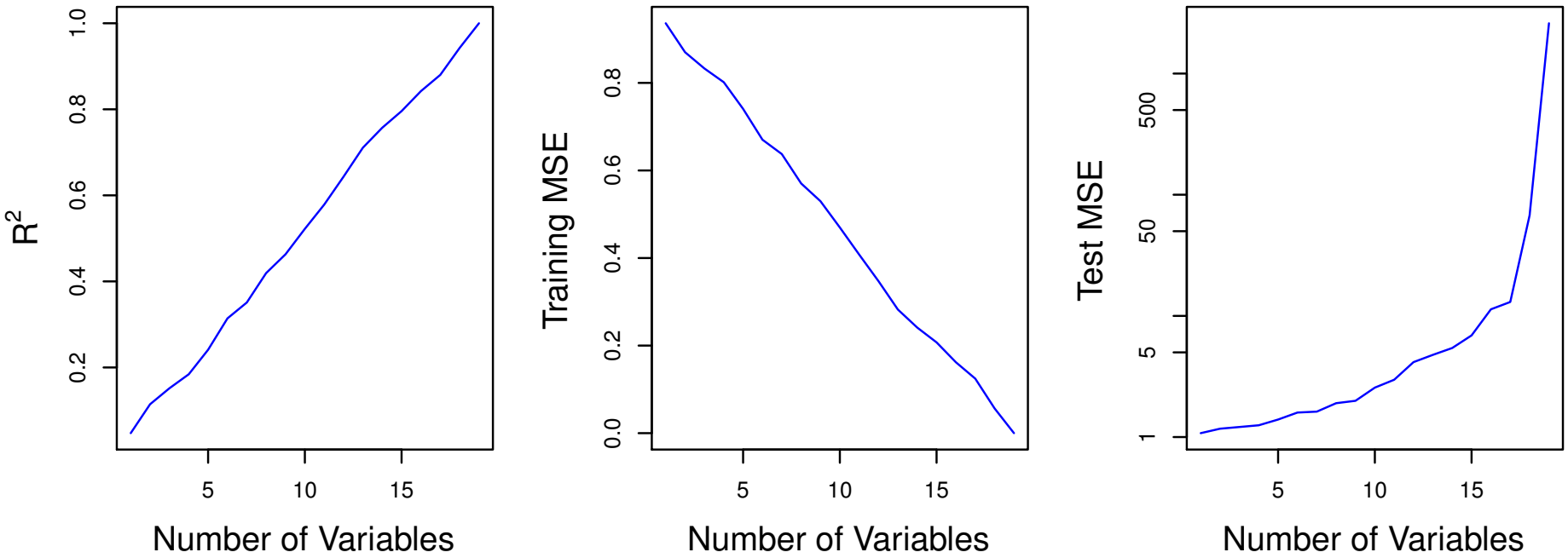


Right model will not generalize well (overfitting), model too flexible..

Linear Model Selection and Regularization

The high-dimensional setting: what goes wrong?

Another example: $n = 20$ observations for $1 \leq p \leq 20$ features, each *completely unrelated* to response.



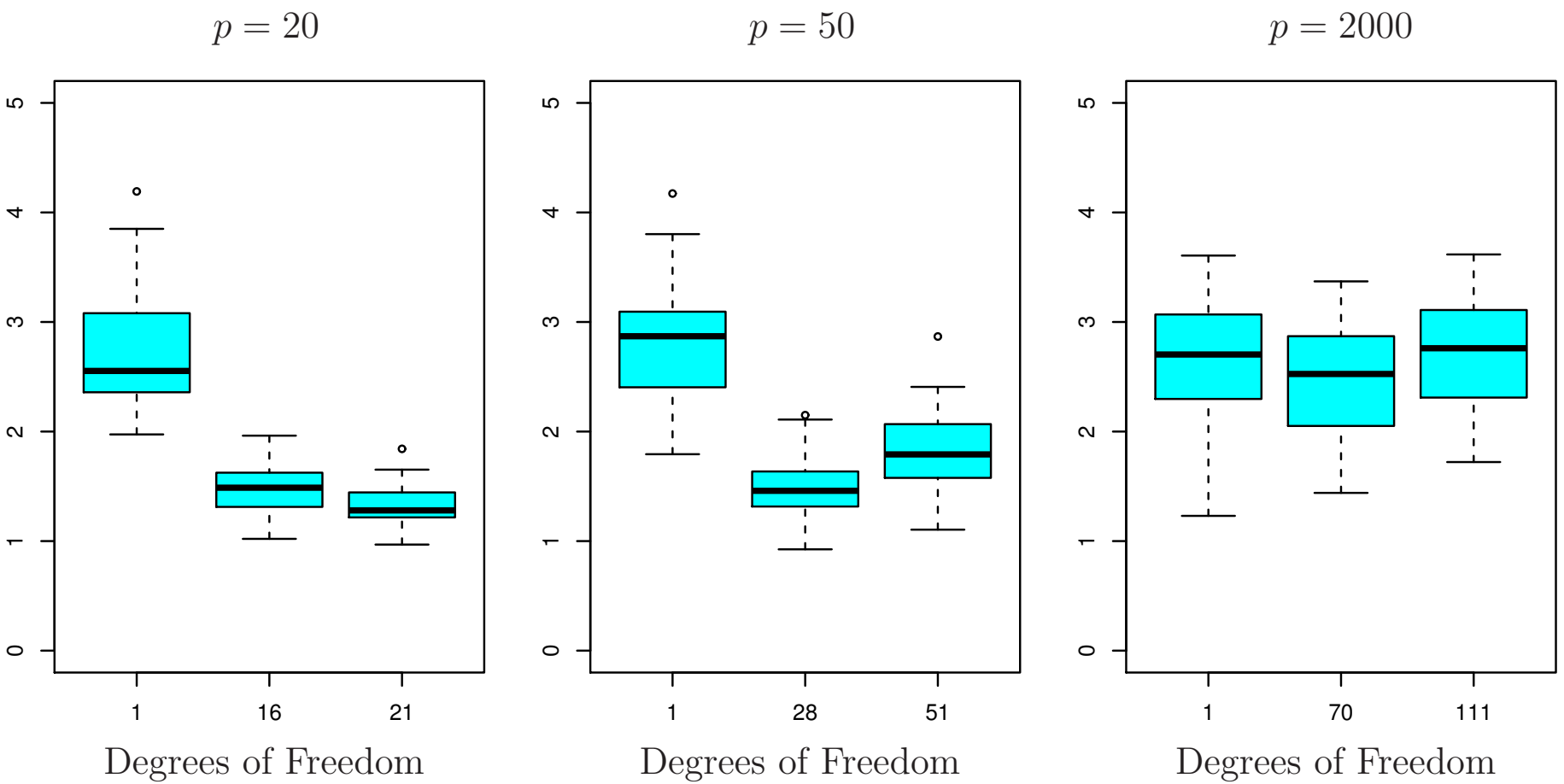
As p increases, $R^2 \to 1$, training MSE $\to 0$ despite no relation of predictors to response. At the same time, test MSE sharply increases as model increasingly flexible.

Casual observer may find large model superior if only first two quantities monitored.

- Model selection techniques based on C_p, AIC, BIC not appropriate for high-dimensional setting, as estimating $\hat{\sigma}^2$ problematic.
- Adjusted R^2 may easily yield value of 1 in high-dimensional setting.
- Less flexible regression models (stepwise selection, shrinkage, PCR) particularly useful in high dimensions. Avoid overfitting by constraining flexibility.
- Next figure: Lasso on $n = 100$ simulated training observations using $p = 20, 50$ and $2,000$ features, of which 20 related to response.
  Then MSE evaluated on independent test set.

Linear Model Selection and Regularization

The high-dimensional setting: what goes wrong?



- For $p = 20$, lowest test MSE for low value of λ. For larger p, best model obtained for larger λ. When $p = 2000$, lasso performs badly for all values of λ.
- Rather than λ, plot shows **degrees of freedom** of model, i.e., # nonzero coefficients of lasso estimate.

Summary:

1. Shrinkage plays key role in high dimensions.
2. Correct value of tuning parameter essential.
3. Test error increases with dimension, unless additional features informative.

- Third observation related to **curse of dimensionality**: quality of model need not increase as features added.
- Compare left and right panel in figure: test MSE almost doubles as p increased from 20 to 2000.
- **Noise features** (not related to response) increase dimension, exacerbate overfitting danger.
- Adding features truly related to response will generally improve model.
- New sensor technology allowing for millions of observations can lead to worse results if features not relevant. Even if relevant, variance incurred by fitting their coefficients may outweigh reduction in bias from additional features.

Linear Model Selection and Regularization

The high-dimensional setting: what goes wrong?

- In high dimensions: **collinearity** problem extreme. (Why?)
- Never know which variables truly predictive, can never obtain best coefficients.
- At best: assign large coefficients to variables correlated with variables truly predictive for response.
- For $p > n$ can easily obtain useless model with zero residual.
- Traditional measures of model quality based on training data often highly misleading in high dimensions.
- Reporting MSE on independent test data particularly important here.

Contents

Optimality of LS Estimate

The Gauss-Markov[7] theorem

Theorem 6.1 (Gauss-Markov theorem)

Given observations $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \varepsilon_i$, $i = 1, \ldots, n$, for which the uncorrelated random noise variables ε_i have mean zero and constant variance $\sigma^2 > 0$, and assuming that the observation vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p \in \mathbb{R}^n$ are linearly independent, then the least squares estimate

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}, \quad \boldsymbol{X} = [\boldsymbol{x}_1 | \cdots | \boldsymbol{x}_p] \in \mathbb{R}^{n \times p}, \quad \boldsymbol{y} \in \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

has minimal variance among all linear unbiased estimators of $\boldsymbol{\beta}$.

[7] C.F. Gauss, 1777–1855, A.A. Markov, 1856–1922

Optimality of LS Estimate

Remarks

- No assumption is made on the distribution of the errors, only on their first two moments.
- The theorem also holds if $\mathbf{Var}\,\varepsilon = \boldsymbol{\Sigma}$ is a (nonsingular) covariance matrix. In this case the best linear unbiased estimator solves the **weighted least squares** problem

$$\|y - X\beta\|_{\boldsymbol{\Sigma}} \to \min_{\beta}, \qquad \|x\|_{\boldsymbol{\Sigma}}^2 = x^\top \boldsymbol{\Sigma}^{-1} x.$$

- The theorem does not say there are no better estimators than LS, only that any better estimators are either nonlinear or biased.
- Examples of biased estimators are ridge regression and the lasso.
- ESL:
  "Most models are distortions of the truth, and hence are biased; picking the right model amounts to creating the right balance between bias and variance.

Theorem 6.2 (Singular value decomposition)

For any matrix $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ of rank r, there exist orthogonal matrices $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{V} \in \mathbb{R}^{p\times p}$ as well as a "diagonal" matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_r & O \\ O & O \end{bmatrix} \in \mathbb{R}^{n\times p} \qquad \text{where} \qquad \boldsymbol{\Sigma}_r = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}$$

and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, such that

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top. \tag{SVD}$$

- The positive numbers $\sigma_1, \ldots, \sigma_r$ are called the **singular values** of $\boldsymbol{A}$.
- The columns of $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n]$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_p]$ are the **left and right singular vectors**, respectively.

Singular Value Decomposition

Properties

1. Representation of $\boldsymbol{A}$ as sum of rank-1 matrices:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r]\, \boldsymbol{\Sigma}_r\, [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r]^\top = \sum_{k=1}^{r} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^\top$$

2. Singular vector mapping properties:

$$\boldsymbol{A}\boldsymbol{v}_k = \begin{cases} \sigma_i \boldsymbol{u}_i & k = 1, 2, \ldots, r, \\ \boldsymbol{0} & k = r+1, \ldots, p \end{cases}$$

and

$$\boldsymbol{A}^\top \boldsymbol{u}_k = \begin{cases} \sigma_k \boldsymbol{v}_k & k = 1, 2, \ldots, r, \\ \boldsymbol{0} & k = r+1, \ldots, n. \end{cases}$$

3.

$\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r\}$	is an ON-basis of	$\mathscr{R}(\boldsymbol{A})$.
$\{\boldsymbol{u}_{r+1}, \ldots, \boldsymbol{u}_m\}$	is an ON-basis of	$\mathscr{N}(\boldsymbol{A}^\top) = \mathscr{R}(\boldsymbol{A})^\perp$.
$\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r\}$	is an ON-basis of	$\mathscr{R}(\boldsymbol{A}^\top) = \mathscr{N}(\boldsymbol{A})^\perp$.
$\{\boldsymbol{v}_{r+1}, \ldots, \boldsymbol{v}_n\}$	is an ON-basis of	$\mathscr{N}(\boldsymbol{A})$.

4. Eigenspaces of $\boldsymbol{A}\boldsymbol{A}^\top$ and $\boldsymbol{A}^\top\boldsymbol{A}$:
   - $\sigma_1^2, \ldots, \sigma_r^2$ are the non-zero eigenvalues of $\boldsymbol{A}^\top\boldsymbol{A}$ and $\boldsymbol{A}\boldsymbol{A}^\top$, respectively:

$$\boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\boldsymbol{V}^\top = \boldsymbol{V}\begin{bmatrix}\boldsymbol{\Sigma}_r^2 & o\\ o & o\end{bmatrix}\boldsymbol{V}^\top,$$

$$\boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^\top\boldsymbol{U}^\top = \boldsymbol{U}\begin{bmatrix}\boldsymbol{\Sigma}_r^2 & o\\ o & o\end{bmatrix}\boldsymbol{U}^\top.$$

   - In particular, the singular values $\sigma_1, \ldots, \sigma_r$ are uniquely determined by $\boldsymbol{A}$.
   - The right singular vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p$ form an ON-basis of $\mathbb{R}^p$ of eigenvectors of $\boldsymbol{A}^\top\boldsymbol{A}$:

$$\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{v}_k = \begin{cases}\sigma_k^2\boldsymbol{v}_k & k = 1, 2, \ldots, r,\\ \boldsymbol{0} & k = r+1, \ldots, p.\end{cases}$$

     The left singular vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ form an ON-basis of $\mathbb{R}^n$ of eigenvectors of $\boldsymbol{A}\boldsymbol{A}^\top$:

$$\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{u}_k = \begin{cases}\sigma_i^2\boldsymbol{u}_k & k = 1, 2, \ldots, r,\\ \boldsymbol{0} & k = r+1, \ldots, n.\end{cases}$$

Singular Value Decomposition

Properties

5. If $\boldsymbol{A} = \boldsymbol{A}^\top \in \mathbb{R}^{n\times n}$ with non-zero eigenvalues

$$\lambda_1, \ldots, \lambda_r, \qquad |\lambda_1| \geq \cdots \geq |\lambda_r| > 0,$$

   then the singular values of $\boldsymbol{A}$ are given by $\sigma_k = |\lambda_k|$.

6. The (p-dimensional) unit sphere is mapped by $\boldsymbol{A}$ to an ellipsoid (in $\mathbb{R}^n$) with center $\mathbf{0}$ and semi-axes $\sigma_k \boldsymbol{u}_k$ ($\sigma_k := 0$ für $k > r$).

7. For $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ there holds $\|A\|_2 = \sigma_1$ and $\|\boldsymbol{A}\|_F = \sqrt{\sigma_1^2 + \cdots + \sigma_r^2}$.
   For $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ invertible, there holds in addition that $\|\boldsymbol{A}^{-1}\|_2 = \sigma_n^{-1}$.

8. Anlalogous statements hold for complex-valued matrices $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H$ ($\boldsymbol{U}, \boldsymbol{V}$ unitary). (In (5) replace $\boldsymbol{A} = \boldsymbol{A}^\top$ by '$\boldsymbol{A}$ normal'.)

Singular Value Decomposition

Best rank-k approximation

- E. Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen. 1. Teil: Entwicklung willkürlicher Funktionen nach Systemen vorgeschriebener. Math. Ann., 63 (1907), pp. 433–476
- C. Eckart, G. Young. The approximation of one matrix by another of lower rank. Psychometrika, 1 (1936), pp. 211–218
- L. Mirsky. Symmetric gauge functions and unitarily invariant norms. Quart. J. Math. Oxford, 11 (1960), pp. 50–59

Theorem 6.3 (Best approximation by matrices of lower rank)

For a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ of rank r with SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ the best approximation problem

$$\min\{\|\boldsymbol{A}-\boldsymbol{B}\|_2 \,:\, \boldsymbol{B}\in\mathbb{R}^{n\times p} \text{ and } \operatorname{rank}(B)\le k\}$$

for $k < r$ is solved by

$$\boldsymbol{A}_k := \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top \qquad \text{with} \qquad \|\boldsymbol{A}-\boldsymbol{A}_k\|_2 = \sigma_{k+1}.$$

- $\boldsymbol{A}_k$ as above is also the closest rank-k matrix to $\boldsymbol{A}$ in the Frobenius-norm $\|\cdot\|_F$, with distance $\|\boldsymbol{A}-\boldsymbol{A}_k\|_F = \sqrt{\sigma_{k+1}^2+\cdots+\sigma_r^2}$.

- Recall the ridge regression estimate $\hat{\boldsymbol{\beta}}_R$ for the LS problem $\boldsymbol{X}\boldsymbol{\beta} \approx \boldsymbol{y}$ with data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ and observation vector $\boldsymbol{y} \in \mathbb{R}^n$: for a given value of the tuning (or regularization) parameter $\lambda \geq 0$ it was defined by

$$\hat{\boldsymbol{\beta}}_R = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, Q_\lambda(\boldsymbol{\beta}), \qquad Q_\lambda(\boldsymbol{\beta}) := \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2.$$

- Rewriting the objective function $Q_\lambda(\boldsymbol{\beta})$ as

$$Q_\lambda(\boldsymbol{\beta}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta} = \begin{bmatrix}\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} \\ \sqrt{\lambda}\boldsymbol{\beta}\end{bmatrix}^\top \begin{bmatrix}\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} \\ \sqrt{\lambda}\boldsymbol{\beta}\end{bmatrix}$$

$$= \left\| \begin{bmatrix}\boldsymbol{y} \\ \boldsymbol{0}\end{bmatrix} - \begin{bmatrix}\boldsymbol{X} \\ \sqrt{\lambda}\boldsymbol{I}\end{bmatrix}\boldsymbol{\beta} \right\|_2^2,$$

we observe that ridge regression can be viewed as a standard LS formulation for the augmented problem

$$\begin{bmatrix}\boldsymbol{X} \\ \sqrt{\lambda}\boldsymbol{I}\end{bmatrix}\boldsymbol{\beta} \approx \begin{bmatrix}\boldsymbol{y} \\ \boldsymbol{0}\end{bmatrix}.$$

- The associated normal equations of ridge regression

$$(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})\boldsymbol{\beta} = \boldsymbol{X}^\top \boldsymbol{y} \tag{6.14}$$

  are obtained from those of original LS problem by adding $\lambda \boldsymbol{I}$ to the coefficient matrix, guaranteeing positive definiteness for $\lambda > 0$.
- Given an SVD $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ of the data matrix $\boldsymbol{X}$ with orthogonal matrices $\boldsymbol{U} = [\boldsymbol{u}_1|\ldots|\boldsymbol{u}_n] \in \mathbb{R}^{n\times n}$, $\boldsymbol{V} = [\boldsymbol{v}_1|\ldots|\boldsymbol{v}_p] \in \mathbb{R}^{p\times p}$ and, and assuming it has full rank $p \le n$, $\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_p \\ \boldsymbol{O} \end{bmatrix}$, $\boldsymbol{\Sigma}_p = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$, $\sigma_1 \ge \cdots \ge \sigma_p > 0$, this implies

$$\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Sigma}^\top \boldsymbol{\Sigma}\boldsymbol{V}^\top, \quad \boldsymbol{\Sigma}^\top \boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2),$$

$$\boldsymbol{X}^\top \boldsymbol{y} = \boldsymbol{V}\boldsymbol{\Sigma}^\top \boldsymbol{U}^\top \boldsymbol{y} = \sum_{j=1}^{p} \sigma_p (\boldsymbol{u}_j^\top \boldsymbol{y}) \boldsymbol{v}_j$$

.

- Inserting these expressions into the normal equations (6.14) yields

$$\boldsymbol{V}(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}+\lambda\boldsymbol{I})\boldsymbol{V}^\top\boldsymbol{\beta}=\boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{U}^\top\boldsymbol{y}$$

or, setting $\boldsymbol{\gamma}:=\boldsymbol{V}^\top\boldsymbol{\beta}$,

$$(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}+\lambda\boldsymbol{I})\boldsymbol{\gamma}=\boldsymbol{\Sigma}^\top\boldsymbol{U}^\top\boldsymbol{y},\qquad\text{giving}\qquad\gamma_j=\frac{\sigma_j}{\sigma_j^2+\lambda}\boldsymbol{u}_j^\top\boldsymbol{y},\quad j=1,\ldots,p,$$

and finally, with $\boldsymbol{\beta}=\boldsymbol{V}\boldsymbol{\gamma}$, the ridge regression estimate

$$\hat{\boldsymbol{\beta}}_R=\sum_{j=1}^{p}\frac{\sigma_j}{\sigma_j^2+\lambda}(\boldsymbol{u}_j^\top\boldsymbol{y})\boldsymbol{v}_j.$$

- Observe that $\hat{\boldsymbol{\beta}}_R$ is obtained from the standard LS estimate $\hat{\boldsymbol{\beta}}=\sum_{j=1}^{p}\frac{\boldsymbol{u}_j^\top\boldsymbol{y}}{\sigma_j}\boldsymbol{v}_j$ by multiplying each coefficient with the **filter factor**

$$\frac{\sigma_j^2}{\sigma_j^2+\lambda},\qquad j=1,\ldots,p.$$

- Given SVD, ridge regression estimates for additional λ essentially for free.

Principal Components

Covariance matrix of a random vector

- Recall: the variance of a random variable X with expectation $\mu := \mathbf{E}[X]$ is given by

$$\sigma^2 = \mathbf{Var}\, X = \mathbf{E}\left[(X-\mu)^2\right].$$

- For a **random vector** $X = (X_1, \ldots, X_p)^\top \in \mathbb{R}^p$ with expectation $\boldsymbol{\mu} := \mathbf{E}[X]$, the **variance** or **covariance matrix** is given by

$$\boldsymbol{C} := \mathbf{Var}\, X = \mathbf{E}\left[(X-\boldsymbol{\mu})(X-\boldsymbol{\mu})^\top\right] = \boldsymbol{C}^\top \in \mathbb{R}^{p\times p},$$

with matrix entries

$$C_{i,j} = \mathbf{E}\left[(X_i-\mu_i)(X_j-\mu_j)\right] = \mathbf{Cov}(X_i, X_j), \quad i,j = 1,\ldots,p.$$

Principal Components

Total variance a random vector

- A scalar measure of the total variance contained in a random vector $X \in \mathbb{R}^p$ is provided by the **trace** of its covariance matrix

$$\operatorname{tr} \boldsymbol{C} = \sum_{j=1}^{p} C_{j,j} = \sum_{j=1}^{p} \mathbf{Cov}(X_j, X_j) = \sum_{j=1}^{p} \mathbf{Var}\, X_j.$$

- Justification:

$$\begin{aligned}\mathbf{E}\left[\|X - \mathbf{E}[X]\|_2^2\right] &= \mathbf{E}\left[\|X - \boldsymbol{\mu}\|_2^2\right] = \mathbf{E}\left[(X-\boldsymbol{\mu})^\top (X-\boldsymbol{\mu})\right] \\ &= \mathbf{E}\left[\sum_{j=1}^{p} (X_j - \mu_j)^2\right] = \sum_{j=1}^{p} \mathbf{E}\left[(X_j-\mu_j)^2\right] = \sum_{j=1}^{p} \mathbf{Var}\, X_j.\end{aligned}$$

- By a well-known result from linear algebra, if $\lambda_j(\boldsymbol{C})$ denotes the j-th eigenvalue (in descending order) of $\boldsymbol{C}$[8], there also holds

$$\operatorname{tr} \boldsymbol{C} = \sum_{j=1}^{p} \lambda_j(\boldsymbol{C}).$$

[8] Note that these are real and positive as $\boldsymbol{C}$ is symmetric and positive-definite.

- Given a spectral decomposition

$$\boldsymbol{C} = \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^\top, \quad \boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}, \quad \boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_p),$$

of $\boldsymbol{C}$ and the fact that the **Frobenius norm** $\|\cdot\|_F$ is unitarily invariant, we also have

$$\operatorname{tr}\boldsymbol{C} = \|\boldsymbol{\Lambda}^{1/2}\|_F^2 = \|\boldsymbol{W}\boldsymbol{\Lambda}^{1/2}\boldsymbol{W}^\top\|_F^2 = \|\boldsymbol{C}^{1/2}\|_F^2.$$

- In view of the fact that $|\lambda_j(\boldsymbol{C})| = \lambda_j(\boldsymbol{C})$ for covariance matrices, the spectral decomposition $\boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^\top$ is also a singular value decomposition.
- Combining with Theorem 6.3, we conclude that for any $k \in \{1, \ldots, p\}$ the matrix

$$\boldsymbol{C}_k = \sum_{j=1}^{k} \lambda_j \boldsymbol{w}_j \boldsymbol{w}_j^\top,$$

where $\boldsymbol{W} = [\boldsymbol{w}_1 | \ldots | \boldsymbol{w}_p]$, is the best approximation of the covariance matrix $\boldsymbol{C}$ in the spectral and Frobenius norms among all matrices of rank $\leq k$.

Principal Components

Linear combinations of random vector components

- Given a random vector $X = (X_1, \ldots, X_p)^\top \in \mathbb{R}^p$ and $\boldsymbol{w}_j$ a normalized eigenvector of its covariance matrix $\boldsymbol{C}$ with associated eigenvalue λ_j, define the scalar random variable $Z_j := \boldsymbol{w}_j^\top X$. Then

$$\begin{aligned}\mathbf{Var}\, Z_j &= \mathbf{E}\left[(\boldsymbol{w}_j^\top X - \mathbf{E}\left[\boldsymbol{w}_j^\top X\right])^2\right] = \mathbf{E}\left[\left(\boldsymbol{w}_j^\top (X-\boldsymbol{\mu})\right)^2\right]. \\ &= \mathbf{E}\left[(\boldsymbol{w}_j^\top (X-\boldsymbol{\mu}))(X-\boldsymbol{\mu})^\top \boldsymbol{w}_j)\right] = \boldsymbol{w}_j^\top \mathbf{E}\left[(X-\boldsymbol{\mu})(X-\boldsymbol{\mu})^\top\right] \boldsymbol{w}_j \\ &= \boldsymbol{w}_j^\top \boldsymbol{C}\boldsymbol{w}_j = \lambda_j.\end{aligned}$$

- More generally, for any linear combination $Z = \boldsymbol{\phi}^\top X$, $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)^\top$, we have

$$\begin{aligned}\mathbf{Var}\, Z &= \mathbf{E}\left[(\boldsymbol{\phi}^\top X - \mathbf{E}\left[\boldsymbol{\phi}^\top X\right])^2\right] = \mathbf{E}\left[(\boldsymbol{\phi}^\top (X-\boldsymbol{\mu}))^2\right] \\ &= \mathbf{E}\left[\left(\sum_{j=1}^p \phi_j (X_j - \mu_j)\right)^2\right] = \sum_{j,k=1}^p \phi_j \phi_k \mathbf{E}\left[(X_j-\mu_j)(X_k-\mu_k)\right] \\ &= \boldsymbol{\phi}^\top \boldsymbol{C}\boldsymbol{\phi}.\end{aligned}$$

Principal Components

Linear combinations of random vector components

- For two general linear combinations $Z_1 = \boldsymbol{\phi}_1^\top X,\ Z_2 = \boldsymbol{\phi}_2^\top X$, we conclude by an analogous calculation that

$$\mathbf{Cov}(Z_1, Z_2) = \boldsymbol{\phi}_2^\top \boldsymbol{C} \boldsymbol{\phi}_1$$

  and therefore that Z_1 and Z_2 are **uncorrelated** if and only if $\boldsymbol{\phi}_2^\top \boldsymbol{C} \boldsymbol{\phi}_1 = 0$, i.e., if the coefficient vectors $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ are orthogonal in the inner product generated by the (symmetric and positive definite) matrix $\boldsymbol{C}$.
- If we seek a change of variables $Z = \boldsymbol{\Phi}^\top X$ with a nonsingular $\boldsymbol{\Phi} \in \mathbb{R}^{p\times p}$ such that the components of Z are uncorrelated with unit variance, then it is necessary that

$$\boldsymbol{I} = \mathbf{E}\left[(Z - \mathbf{E}[Z])(Z - \mathbf{E}[Z])^\top\right] = \mathbf{E}\left[\boldsymbol{\Phi}^\top (X - \boldsymbol{\mu})(X - \boldsymbol{\mu})^\top \boldsymbol{\Phi}\right] = \boldsymbol{\Phi}^\top \boldsymbol{C} \boldsymbol{\Phi}.$$

  The set of all matrices $\boldsymbol{\Phi} \in \mathbb{R}^{p\times p}$ which achieve this is precisely the set of all **congruences** taking $\boldsymbol{C}$ to $\boldsymbol{I}$.

Principal Components

Linear combinations of random vector components

- **Example 1:** given **Cholesky factorization** $C = LL^\top$, choosing $\Phi := L^{-\top}$ gives

$$\Phi^\top C \Phi = L^{-1}(LL^\top)L^{-\top} = I.$$

- **Example 2:** given **spectral decomposition** $C = W\Lambda W^\top$, choosing $\Phi := W\Lambda^{-1/2}$ gives

$$\Phi^\top C \Phi = \Lambda^{-1/2} W^\top (W\Lambda W^\top) W\Lambda^{-1/2} = I.$$

- **Example 3:** given **square-root-free Cholesky factorization** $C = LDL^\top$, where L is lower triangular with a unit diagonal and D is diagonal, choosing $\Phi := L^{-\top}$ gives

$$\Phi^\top C \Phi = L^{-1}(LDL^\top)L^{-\top} = D.$$

- **Example 4:** given **spectral decomposition** $C = W\Lambda W^\top$, choosing $\Phi := W$ gives

$$\Phi^\top C \Phi = W^\top (W\Lambda W^\top) W = \Lambda.$$

For a square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ the expression $\dfrac{\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$, $\quad \boldsymbol{0} \neq \boldsymbol{x} \in \mathbb{R}^n$, is called a **Rayleigh quotient**.

Theorem 6.4 (Fischer, 1905; Courant, 1920)

Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and $k \in \{1, 2, \ldots, n\}$. Then

$$\lambda_k = \min_{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{n-k} \in \mathbb{R}^n} \; \max_{\substack{0\neq \boldsymbol{x}\in\mathbb{R}^n \\ \boldsymbol{x}\perp \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{n-k}}} \frac{\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}, \tag{6.15}$$

$$\lambda_k = \max_{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{k-1} \in \mathbb{R}^n} \; \min_{\substack{0\neq \boldsymbol{x}\in\mathbb{R}^n \\ \boldsymbol{x}\perp \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{k-1}}} \frac{\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \tag{6.16}$$

- The extremal values of the Rayleigh quotient are attained when $\boldsymbol{x}$ is an eigenvector associated with λ_1 or λ_n, respectively.

Consequences of Theorem 6.4:

- Linear combination $\boldsymbol{\phi}^\top X$ where $\|\boldsymbol{\phi}\|_2 = 1$ with maximal variance obtained for $\boldsymbol{\phi} = \boldsymbol{\phi}_1 = \boldsymbol{w}_1$. This is the first principal component.
- Linear combination $\boldsymbol{\phi}^\top X$ where $\|\boldsymbol{\phi}\|_2 = 1$ with maximal variance subject to $\boldsymbol{\phi} \perp \boldsymbol{w}_1$ obtained for $\boldsymbol{\phi} = \boldsymbol{\phi}_2 = \boldsymbol{w}_2$ (second principal component).
- Linear combination $\boldsymbol{\phi}^\top X$ where $\|\boldsymbol{\phi}\|_2 = 1$ with maximal variance subject to $\boldsymbol{\phi} \perp \boldsymbol{w}_1, \ldots, \boldsymbol{w}_{j-1}$ obtained for $\boldsymbol{\phi} = \boldsymbol{\phi}_j = \boldsymbol{w}_j$ (j-th principal component).
- The change of variables afforded by replacing the original random variables $X_1, \ldots, X_p$ by the principal components $Z = \boldsymbol{W}^\top X$ is the (unscaled) congruence obtained from the spectral decomposition.
  The total variance contained in Z is given by

$$\mathbf{E}\left[\|Z - \mathbf{E}[Z]\|_2^2\right] = \sum_{j=1}^{p} \mathbf{Var}\, Z_j = \sum_{j=1}^{p} \lambda_j = \operatorname{tr} \sum_{j=1}^{p} \lambda_j \boldsymbol{w}_j \boldsymbol{w}_j^\top = \operatorname{tr} \boldsymbol{C},$$

  which coincides with the total variance contained in X.

Principal Components

PCR

- Performing regression of a data vector $\boldsymbol{y}$ on $M < p$ principal components results in **principal components regression** (PCR).
- The total variance contained in random vector $(Z_1, \ldots, Z_M)^\top$ is

$$\mathbf{E}\left[\|Z - \mathbf{E}[Z]\|_2^2\right] = \sum_{j=1}^{M} \mathbf{Var}\, Z_j = \sum_{j=1}^{M} \lambda_j = \operatorname{tr} \sum_{j=1}^{M} \lambda_j \boldsymbol{w}_j \boldsymbol{w}_j^\top = \operatorname{tr} \boldsymbol{C}_k.$$

- The fraction of neglected variance in PCR using M principal components is

$$\frac{\sum_{j=k+1}^{p} \lambda_j}{\sum_{j=1}^{p} \lambda_j} = 1 - \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{p} \lambda_j}.$$

- The covariance matrix $\boldsymbol{C}$ and expectation vector $\boldsymbol{\mu}$ are theoretical constructs and typically unavailable hence estimated from data.
- As usual, we denote the data matrix (design matrix) by

$$\boldsymbol{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,p} \\ \vdots & & \vdots \\ x_{n,1} & \cdots & x_{n,p} \end{bmatrix} = [\boldsymbol{x}_1 | \cdots | \boldsymbol{x}_p] \in \mathbb{R}^{n \times p},$$

  each column corresponding to one of p predictor variables (features) and each row to one of n observations (samples, realizations).
- We denote the vector of **sample means** by $\overline{\boldsymbol{X}} := \frac{1}{n}\boldsymbol{e}^\top \boldsymbol{X} = [\overline{x_1}, \ldots, \overline{x_p}]$ and obtain the **centered data matrix** as

$$\tilde{\boldsymbol{X}} := [\boldsymbol{x}_1 - \overline{x_1}\boldsymbol{e} | \cdots | \boldsymbol{x}_p - \overline{x_p}\boldsymbol{e}] = \boldsymbol{X} - \boldsymbol{e}\overline{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{e}\frac{1}{n}\boldsymbol{e}^\top \boldsymbol{X} = (\boldsymbol{I} - \frac{1}{n}\boldsymbol{e}\boldsymbol{e}^\top)\boldsymbol{X}.$$

- Finally, the **unbiased sample covariance matrix** is

$$\boldsymbol{S}_n := \frac{1}{n-1}\tilde{\boldsymbol{X}}^\top \tilde{\boldsymbol{X}} = \frac{1}{n-1}\boldsymbol{X}(\boldsymbol{I} - \frac{1}{n}\boldsymbol{e}\boldsymbol{e}^\top)^2\boldsymbol{X} = \frac{1}{n-1}\boldsymbol{X}(\boldsymbol{I} - \frac{1}{n}\boldsymbol{e}\boldsymbol{e}^\top)\boldsymbol{X}.$$

Principal Components

Data

- In practice the sample covariance matrix $\boldsymbol{S}_n$ takes the place of the covariance matrix $\boldsymbol{C}$.
- For PCA/PCR, one can compute a spectral decompositon of $\boldsymbol{S}_n$.
- Alternatively, given an SVD $\tilde{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, a spectral decomposition of $\boldsymbol{S}_n$ is obtained as

$$\boldsymbol{S}_n = \frac{1}{n-1}\tilde{\boldsymbol{X}}^\top\tilde{\boldsymbol{X}} = \frac{1}{n-1}\boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\boldsymbol{V}^\top.$$

- The SVD approach is generally numerically stabler, in particular if $\tilde{\boldsymbol{X}}$ is ill-conditioned. The spectral decomposition may be cheaper, as $\tilde{\boldsymbol{X}}^\top\tilde{\boldsymbol{X}}$ is smaller than $\tilde{\boldsymbol{X}}$.

Contents

Nonlinear Regression Models

Chapter overview

- Despite the benefits of simplicity and interpretability of the standard linear model for regression, it will suffer from large bias if the model generating the data depends nonlinearly on the predictors.
- In this chapter we explore methods which make the linear regression model more flexible by using **linear combinations of nonlinear functions**, specifically
  1. polynomial and piecewise polynomial functions,
  2. piecewise constant functions,
  3. piecewise piecewise polynomial functions with penalty terms and
  4. generalized additive model functions

  of the predictors.

Contents

- For univariate models, **polynomial regression** replaces the simple linear regression model

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

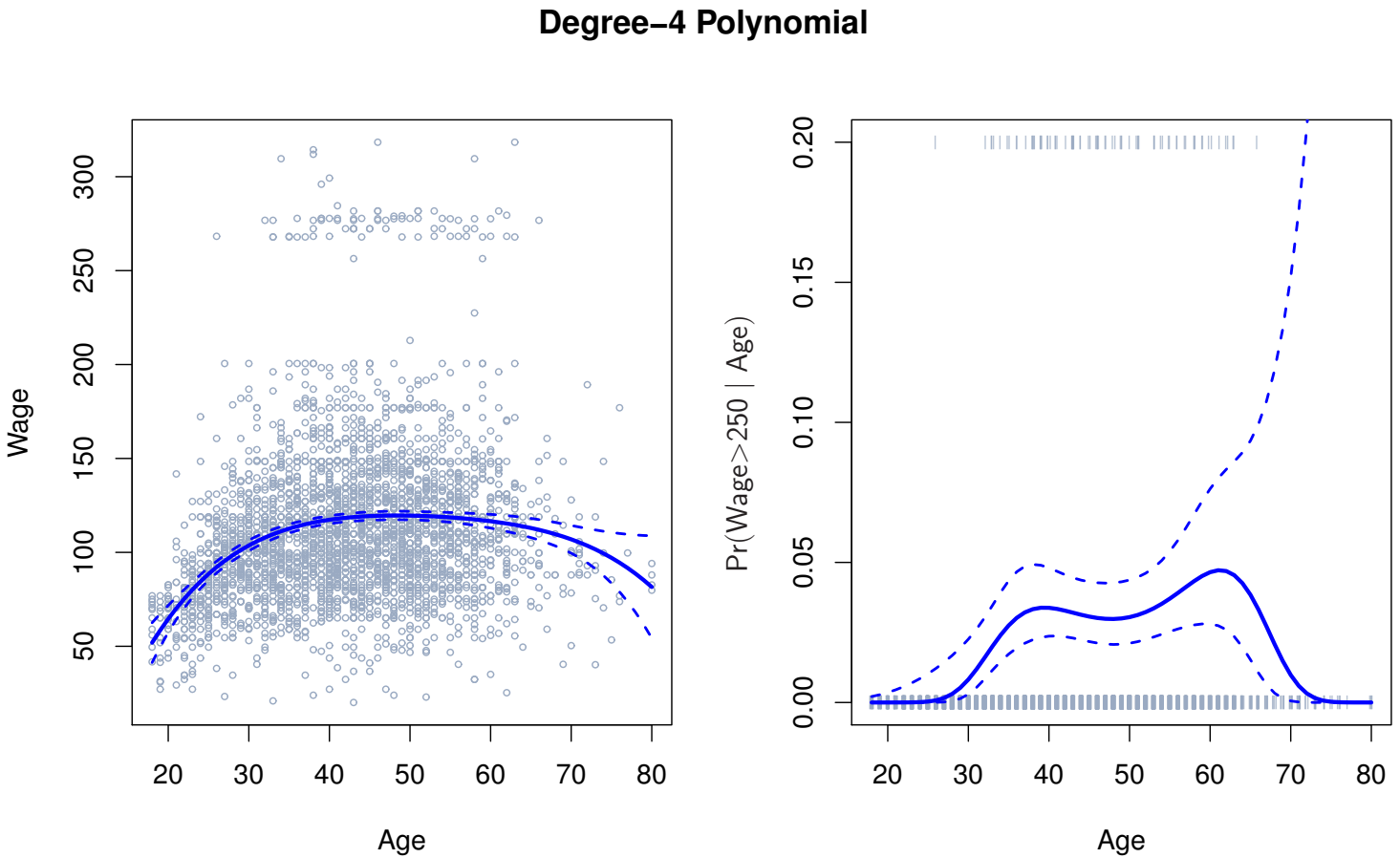
with a polynomial of degree $d > 1$ in the predictor variable

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_d X^d + \varepsilon.$$

- High degree polynomials are often difficult to handle due to their oscillatory behavior and their unboundedness for large arguments, so that degrees higher than 4 can become problematic if employed naively.
- **Example:** `Wage` data set: income and demographic information for males who reside in the central Atlantic region of the United States.
Fit response `wage` [in $ 1000] to predictor `age` by LS using a polynomial of degree $d = 4$.

Nonlinear Regression Models

Polynomial Regression



Left: Polynomial ($d = 4$) LS fit of `wage` against `age` (solid blue) with 95% confidence interval (blue dashed). Right: Model of event $\{$`wage` $> 250\}$ using logistic regression with $d = 4$, fitted posterior probability (solid blue) with 95% confidence interval (blue dashed).

Left panel in previous figure:

- Given fit of particular age value x_0

$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \hat{\beta}_3 x_0^3 + \hat{\beta}_4 x_0^4,$$

  use variance/covariance estimates of $\hat{\beta}_j$ to estimate variance of $\hat{f}(x_0)$.
- If $\hat{\boldsymbol{C}} \in \mathbb{R}^{5\times 5}$ is the estimated covariance matrix of the β_j, then

$$\mathbf{Var}\, \hat{f}(x_0) = \boldsymbol{\ell}_0^\top \hat{\boldsymbol{C}} \boldsymbol{\ell}_0, \qquad \text{where} \qquad \boldsymbol{\ell}_0 = (1, x_0, x_0^2, \ldots, x_0^4)^\top.$$

- Estimated **pointwise standard error** of $\hat{f}(x_0)$ is the square root of this variance.
- Repeat calculation for all x_0, plotting $\pm 2\times$ standard error (corresponds to $\approx$ 95% confidence interval for normally distributed errors) yields dashed lines.

Right panel in previous figure:

- Observations seem to fall into 2 classes: *high earners* ($>$ \$250K) and *low earners*; treat `wage` as binary response variable with these two groups.
- Using logistic regression, can predict this binary response using polynomial functions of predictor `age`.
- This corresponds to fitting

$$\mathbf{P}(y_i > 250|x_i) = \frac{\exp(\beta_0 + \beta_1 x_i + \cdots + \beta_d x_i^d)}{1 + \exp(\beta_0 + \beta_1 x_i + \cdots + \beta_d x_i^d)}.$$

- Gray marks in figure denote ages of high and low earners.
- Solid blue: fitted probabilities of being high/low earner given `age`, dashed blue gives 95% confidence interval (very wide).
- Only 79 high earners of $n = 3000$ observations, results in high variance of coefficients and therefore wide confidence intervals.

Contents

Idea:

- Polynomials are globally defined on the domain of the predictor(s) X.
- To model more locally varied response behavior, divide domain of X into subdomains and use different response model on each.
- Simplest case: different constant function on each subinterval.
- Amounts to converting a continuous variable into an **unordered categorical variable**.

- Introduce "cut points" $c_1 < c_2 < \cdots < c_K$ in range of X, construct $K+1$ new (dummy) variables with indicator function $\mathbb{1}(\cdot)$

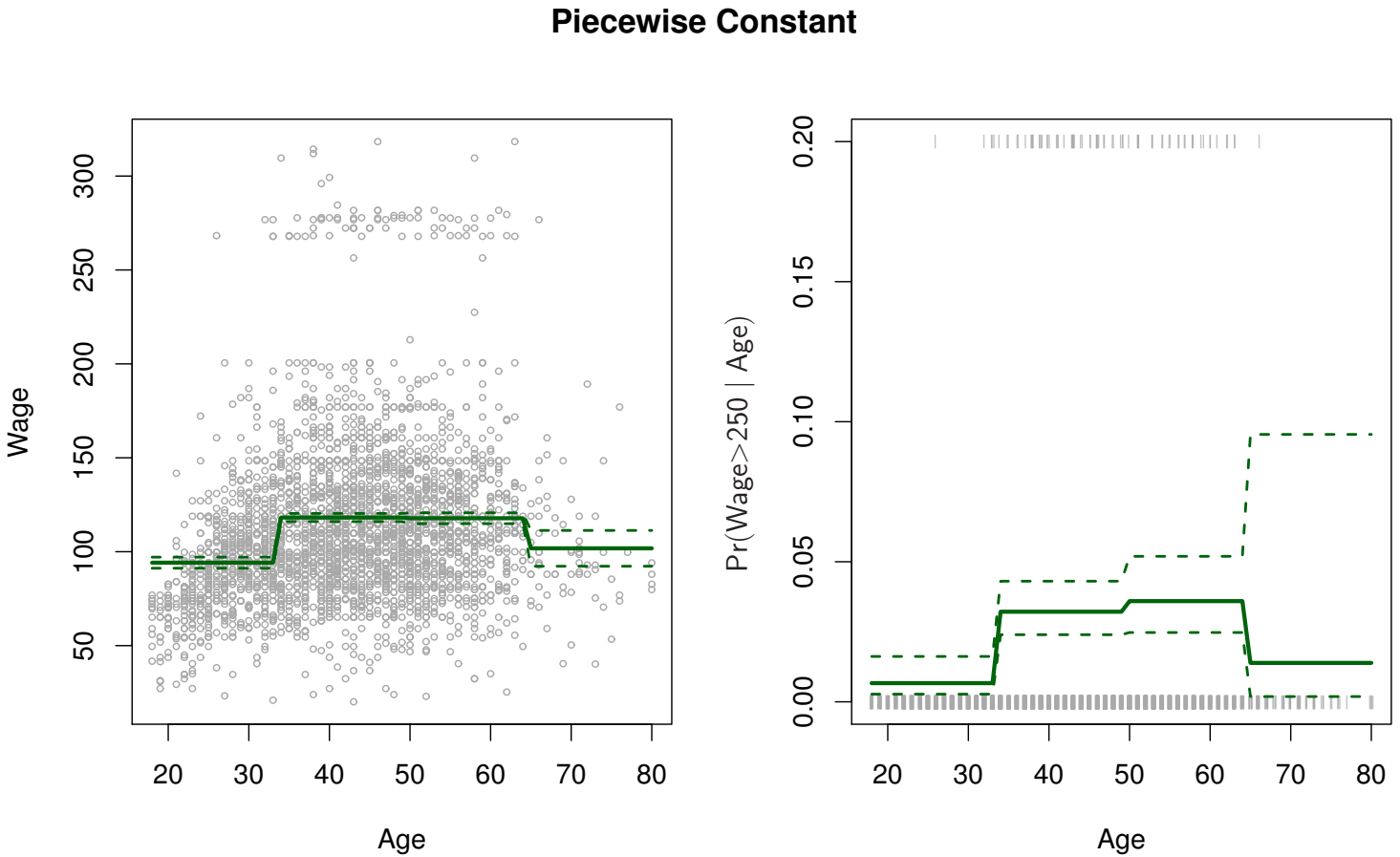
$$\begin{aligned} C_0(X) &= \mathbb{1}(X < c_1), \\ C_1(X) &= \mathbb{1}(c_1 \le X < c_2), \\ &\vdots \\ C_{K-1}(X) &= \mathbb{1}(c_{K-2} \le X < c_{K-1}), \\ C_K(X) &= \mathbb{1}(c_K \le X). \end{aligned} \tag{7.1}$$

- Since events exhaustive and mutually exclusive we have $\sum_{k=0}^{K} C_k(X) \equiv 1$.
- Now fit LS model using $C_1(X), \ldots, C_K(X)$ as predictors[9]:

$$y_i = \beta_0 + \beta_1 C_1(X) + \beta_2 C_2(X) + \cdots + \beta_K C_K(X) + \varepsilon_i.$$

$\beta_j (j > 0)$: average increase in response for $X \in [c_j, c_{j+1})$ relative to $X < c_1$.

[9] Omit C_0 as this is redundant with the intercept.



Left: piecewise constant fit of `wage` against `age` (solid) with 95% confidence band (dashed). Right: modeling event $\{$`wage` $> 250\}$ using logistic regression (solid) with 95% confidence band (dashed).

Previous figure:

- Left: capturing response behavior requires choosing the cut points appropriately. Increasing trend of `wage` with `age` clearly missed in first bin.
- Right: logistic regression fits

$$\mathbf{P}(y_i > 250|x_i) = \frac{\exp(\beta_0 + \beta_1 C_1(X) + \cdots + \beta_K C_K(X)}{1 + \exp(\beta_0 + \beta_1 C_1(X) + \cdots + \beta_K C_K(X)}$$

to predict probability of being high earner given `age`.

Piecewise constant approximation popular in biostatistics and epidemiology, where bins often correspond to 5-year age groups.

Nonlinear Regression Models

General regression functions

- Polynomial, piecewise constant regression examples of **basis function approach**, where linear combination of transformations $\{b_k(X)\}_{k=1}^K$ of predictor variables used for fitting:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_K b_K(x_i) + \varepsilon_i$$

- Basis functions b_k chosen a priori. Examples:

$$b_k(x_i) = \begin{cases} x_i^k & \text{polynomial regression,} \\ \mathbb{1}(c_k \leq x_i < c_{k+1}) & \text{piecewise constant regression.} \end{cases}$$

- Model still linear in the coefficients, hence all inferential methods of linear LS still applicable (standard errors for coefficient estimates, F-statistics for model significance etc.).
- Many possible choices: wavelets, Fourier modes, splines, etc.

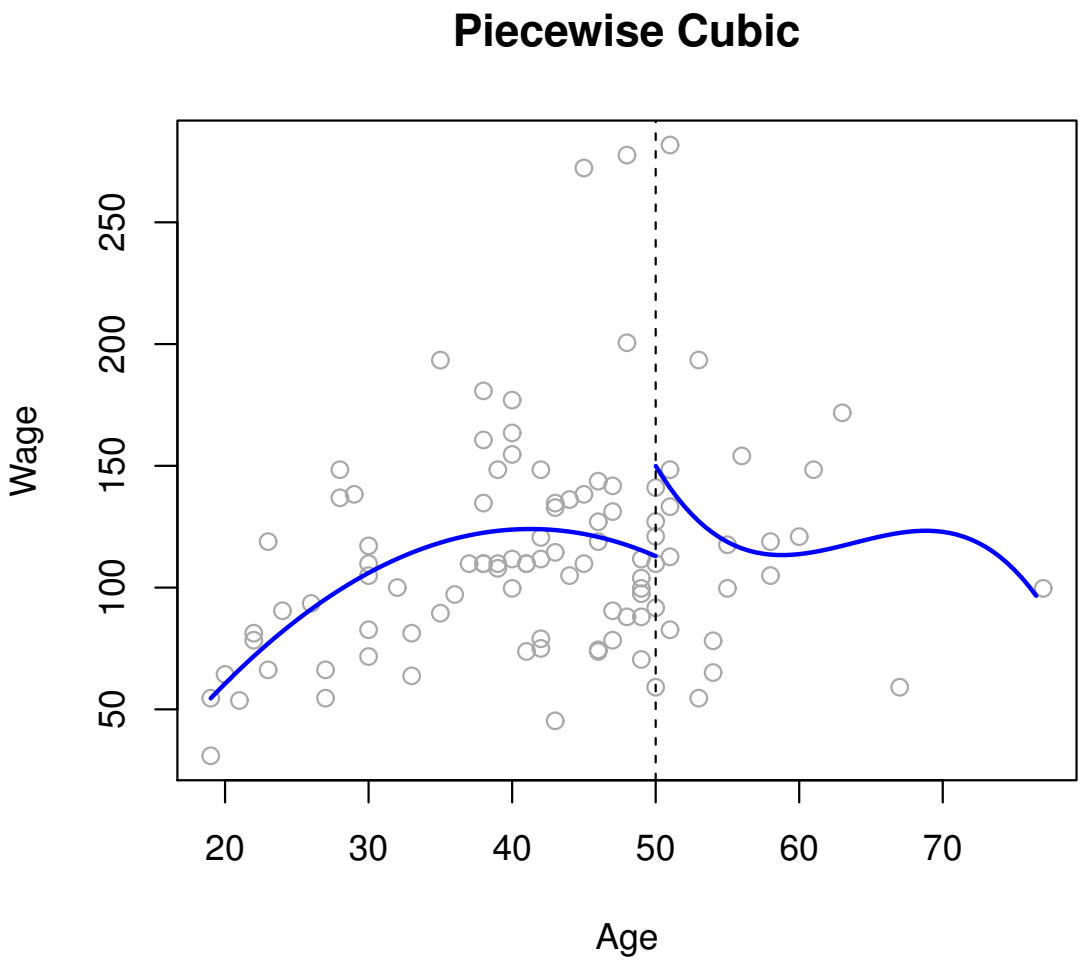
Contents

- As in piecewise constant modes, introduce partition of X domain into sub-intervals.
- Fit a different low-degree polynomial in each subinterval.
- E.g. cubic:

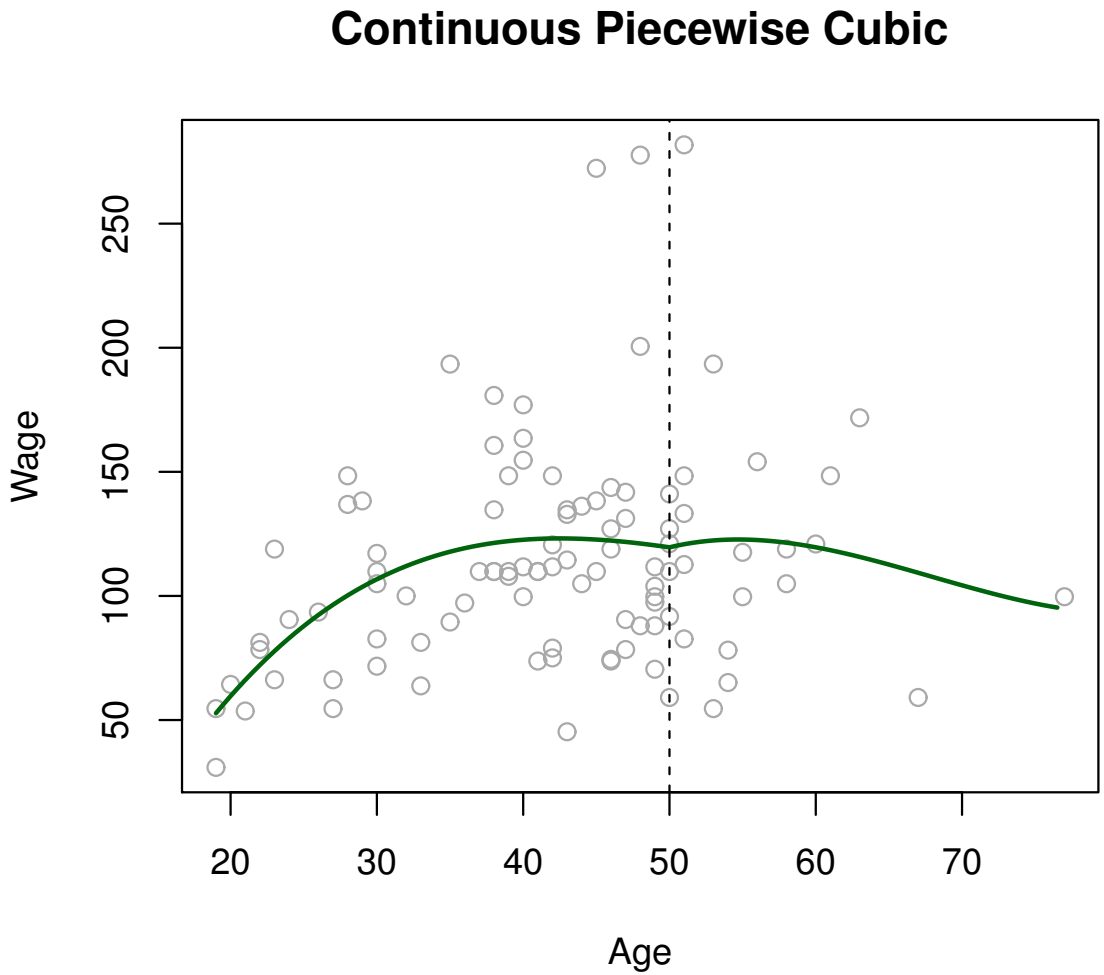
$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \varepsilon_i, \tag{7.2}$$

  with separate coefficients $\beta_0, \beta_1, \beta_2, \beta_3$ in each subinterval.
- Spline termionology: cut points called **knots**.
- Piecewise cubic with single knot at $X = c$:

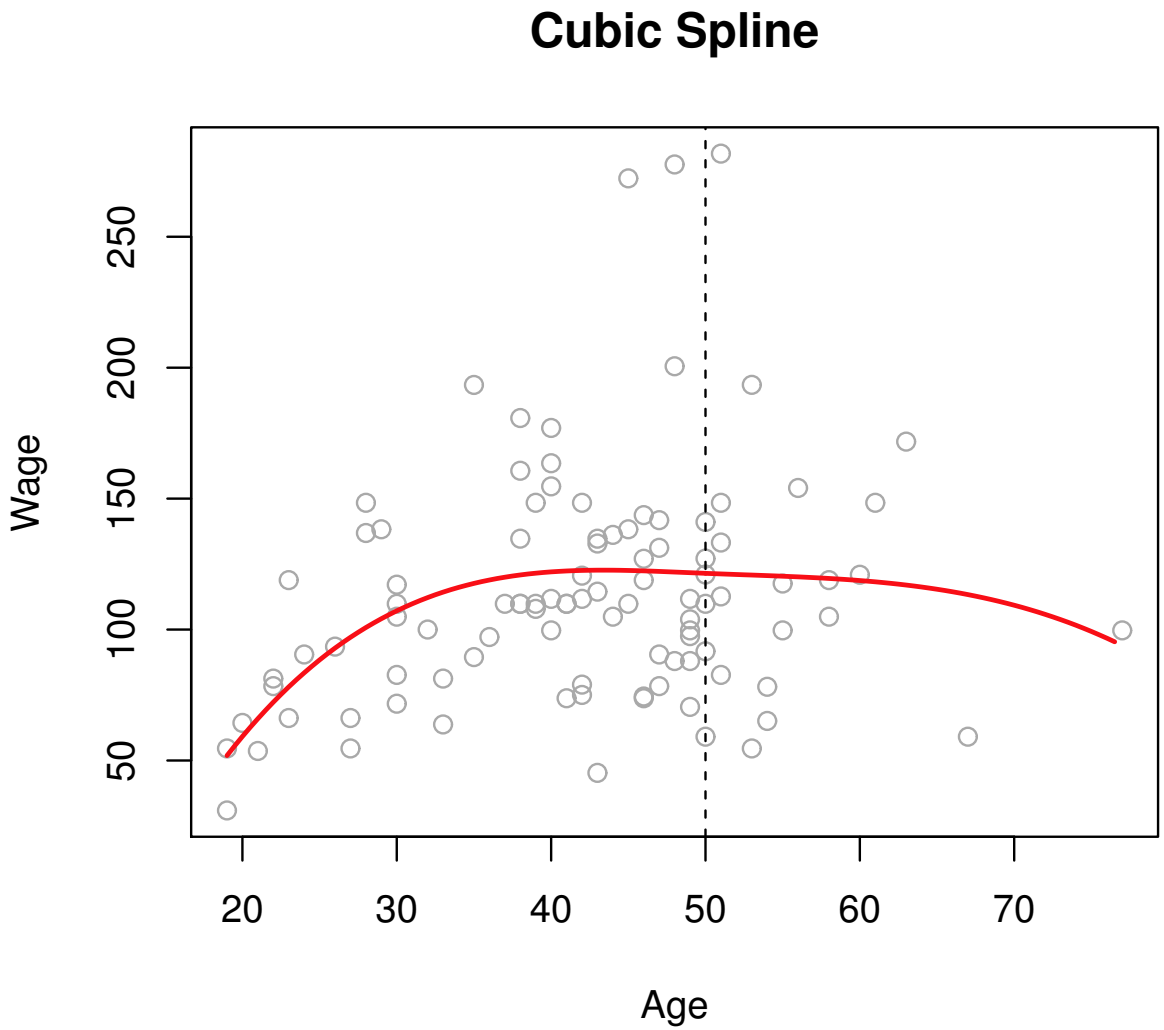
$$y_i = \begin{cases} \beta_{0,1} + \beta_{1,1} x_i + \beta_{2,1} x_i^2 + \beta_{3,1} x_i^3 + \varepsilon_i, & \text{if } x_i < c, \\ \beta_{0,2} + \beta_{1,2} x_i + \beta_{2,2} x_i^2 + \beta_{3,2} x_i^3 + \varepsilon_i, & \text{if } x_i \geq c. \end{cases}$$



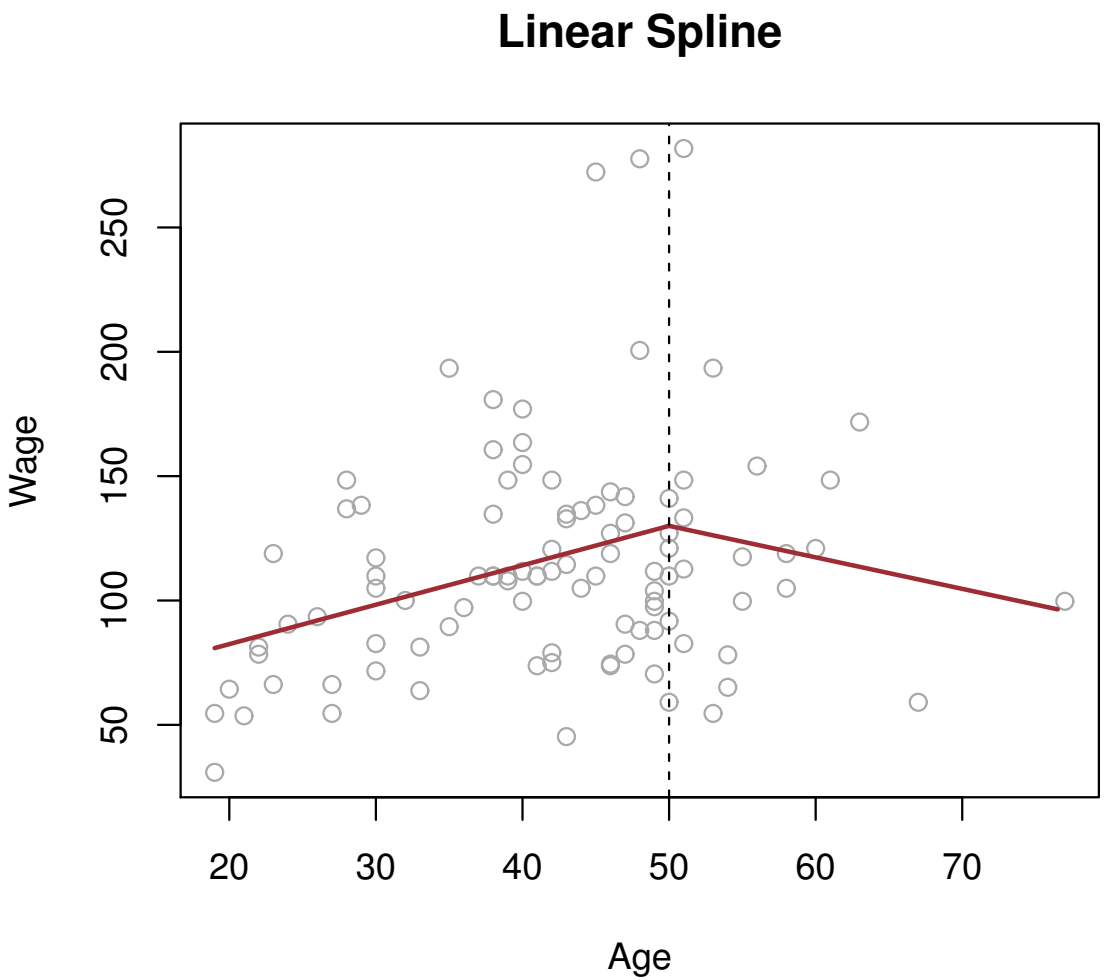
A piecewise cubic fit of `wage` against `age` for the `Wage` data set. Note the discontinuity at the (single) knot $c = 50$. Model has $8 = 2 \times 4$ degrees of freedom.



A piecewise cubic fit of the same data, now with the added constraint that the two polynomials should agree at the knot. This still leaves a 'kink' at the knot, i.e., a discontinuity of the first derivative.



A piecewise cubic fit of the same data, now with the added constraint that the two polynomials as well as their first derivatives should agree at the knot.



A piecewise linear fit of the same data with continuity constraint.

- Cubic spline with K knots: $4 + K$ degrees of freedom.
- General definition of (univariate) spline: piecewise polynomial of degree d with continuity of derivatives of orders $0, 1, 2 \ldots, d-1$.
- Cubic spline model with K knots can be modeled as

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \varepsilon_i$$

  using appropriate basis functions.
- One possible basis (cubic case): start off with monomials x, x^2, x^3, then add for each knot ξ one **truncated monomial**

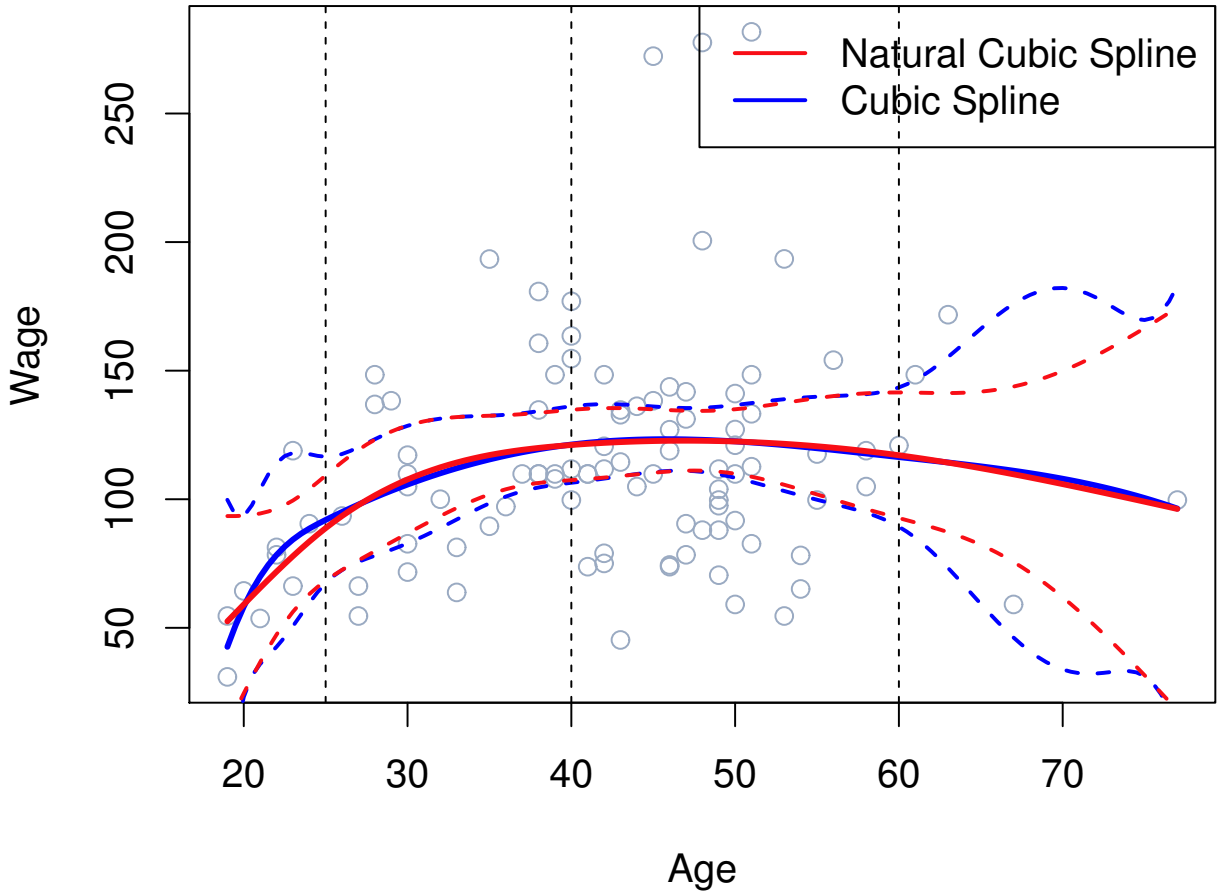
$$h(x, \xi) := (x - \xi)_+^3 := \begin{cases} (x-\xi)^3 & \text{if } x > \xi, \\ 0 & \text{otherwise.} \end{cases}$$

- Adding single basis function $h(x, \xi)$ to model (7.2) will introduce discontinuity only in third derivative at $x = \xi$.

Nonlinear Regression Models

LS regression with splines

To fit LS regression model with cubic splines using K knots $\{\xi_k\}_{k=1}^K$, use $K+3$ predictor variables $X, X^2, X^3, h(X,\xi_1), \ldots, h(X,\xi_K)$.

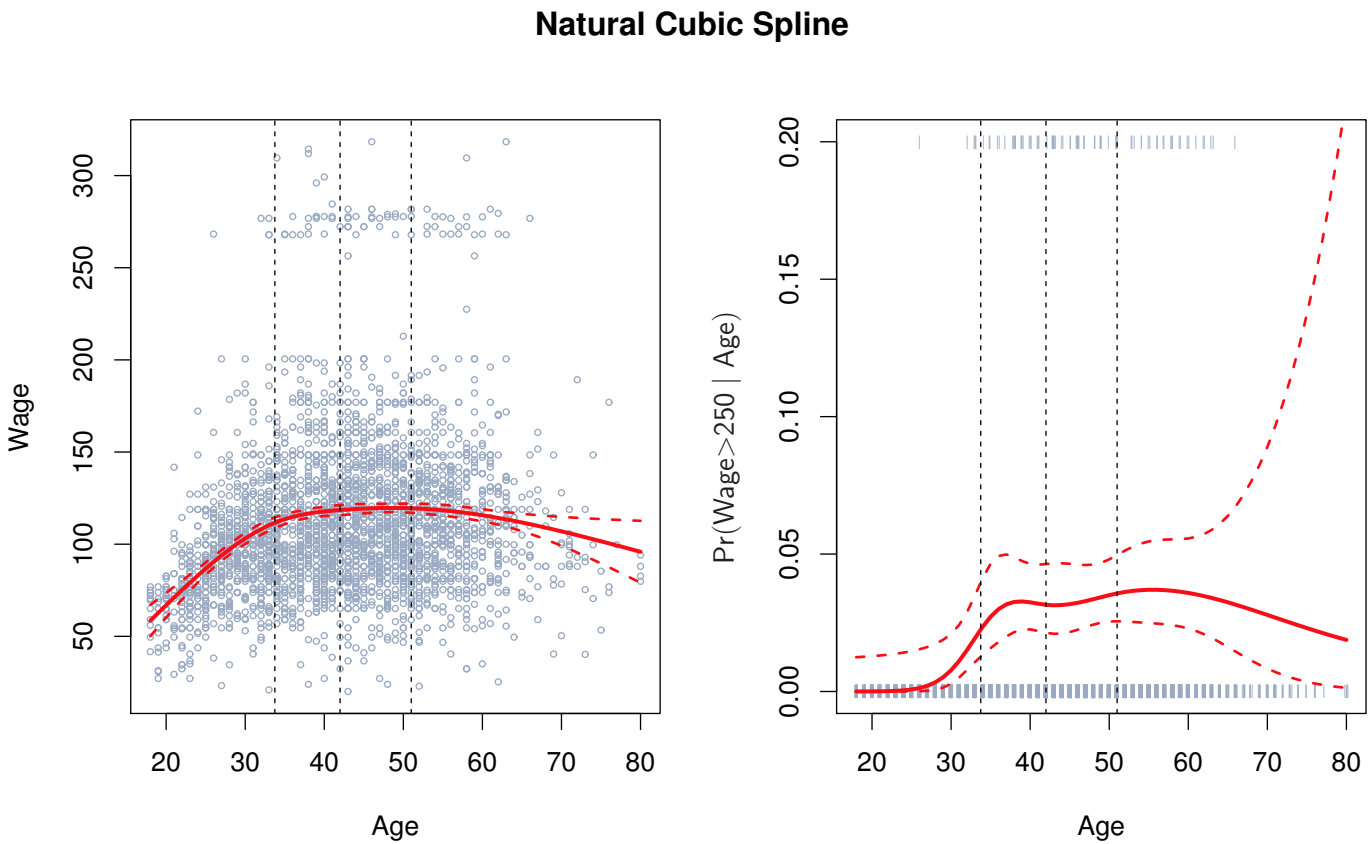


Cubic and natural (linear beyond boundary knots) spline fit using 3 knots to fit a subset of the `Wage` data. Note the large variance near the endpoints.

Nonlinear Regression Models

Choice of spline knots

- Spline most flexible near knots, place these where most variability expected.
- Common practice: space knots uniformly, choose # degrees of freedom, have software place knots at uniform quantiles.

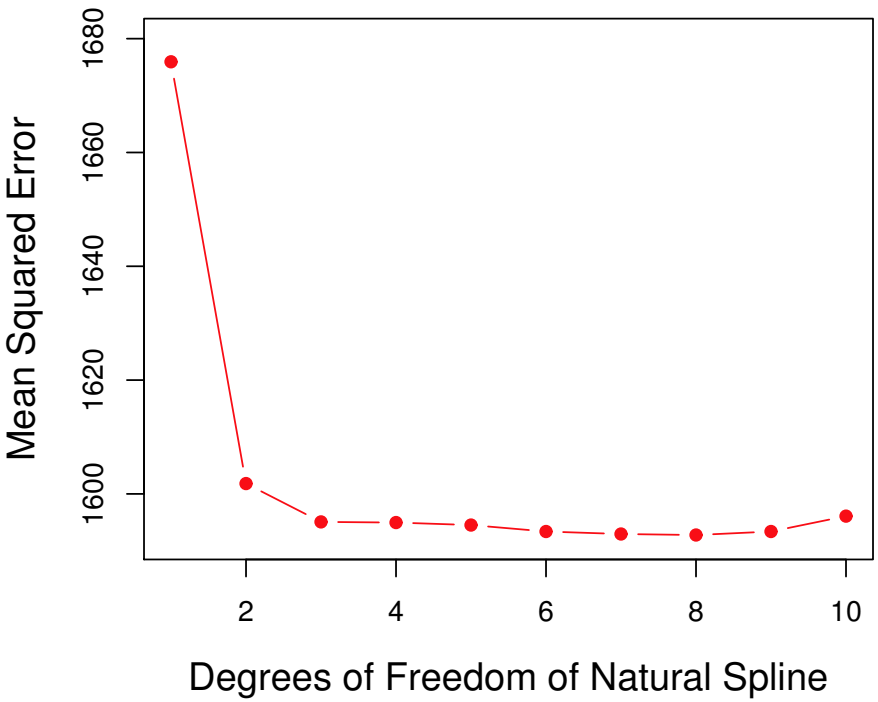


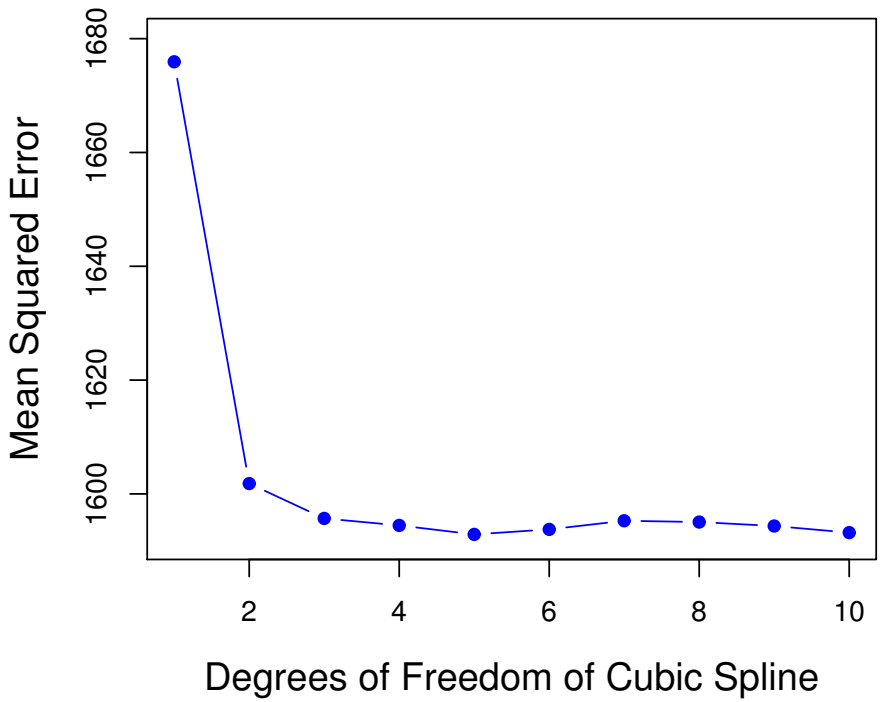
Previous figure:

- Fit natural cubic spline to `Wage` data. Three knots, chosen automatically at 25th, 50th and 75th percentiles of `age`.
- Requested 4 DOF, leading to 3 interior knots. Actually: 5 knots including 2 boundary knots. Corresponds to $9 = 5 + 4$ DOF for cubic spline. Two natural constraints at boundary knots to enforce linearity, leaving $5 = 9 - 4$ DOF. One DOF absorbed in intercept, leaves 4 DOF.
- Right panel: Logistic regression modeling binary event $\{\texttt{wage} > 250\}$. Shown: fitted posterior probability.
- Choosing # knots: trial and error or cross-validation.

Nonlinear Regression Models

Choice of spline knots





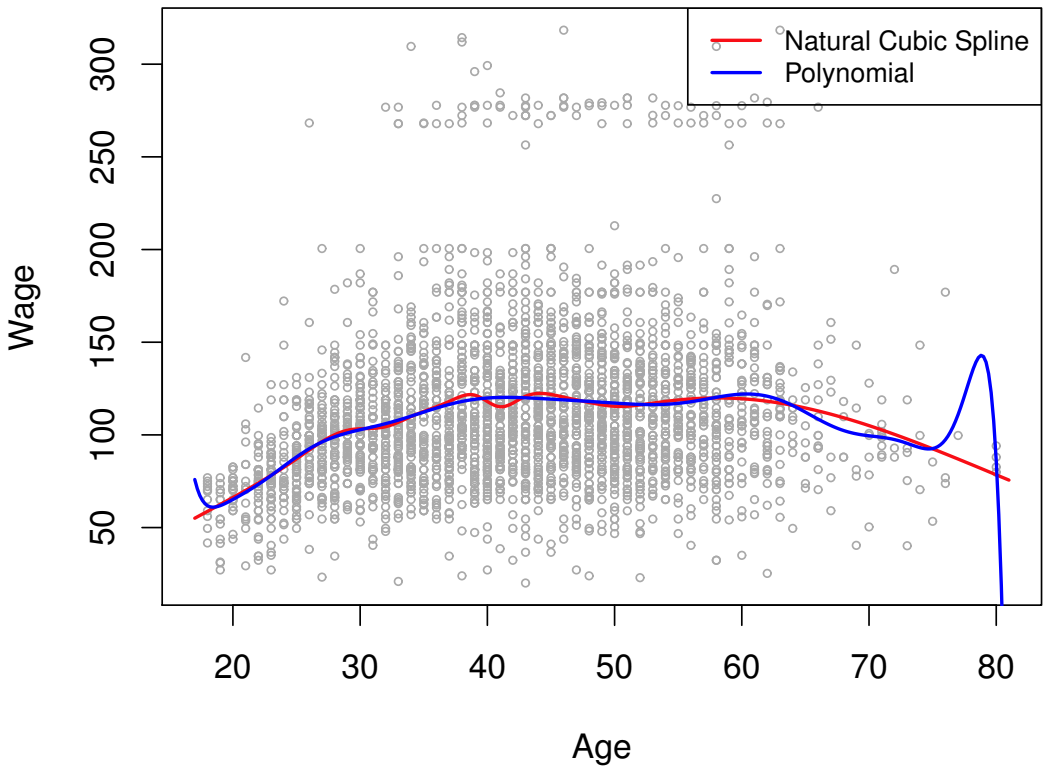
Ten-fold CV MSE for selecting DOF when fitting splines to `Wage` data.

Clear result: 1-DOF not adequate.

Nonlinear Regression Models

Comparison with polynomials

- Spline regression often superior to polynomial.
- More stable as flexibility comes from variation of coefficients of low-degree polynomials and knot placement.



For `Wage` data: comparison of natural cubic spline with 15 DOF to polynomial of degree 15. Latter shows spurious variation near endpoints.

Contents

- Fitting data with smooth function g: want small RSS $= \sum_{i=1}^{n}(y_i - g(x_i))^2$.
- With no constraints on g, can always attain RSS $= 0$ by interpolating the data, leading to overfitting.
- Ensure smoothness by adding penalty term: minimize

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 \,\mathrm{d}t \tag{7.3}$$

  with tuning parameter λ controlling weight assigned to smoothness.
- Limiting values: $\lambda = 0$ corresponds to no smoothing, leading to interpolation for sufficiently many DOF; $\lambda \to \infty$ tends to linear LS fit.
- λ controls bias-variance tradeoff of smoothing spline.
- Can show: minimizer of (7.3) is natural cubic spline with knots at $x_1, \ldots, x_n$. Not the natural cubic spline of the basis function approach, but a **shrunken** version, degree of shrinkage controlled by λ.

- Smoothing spline: natural cubic spline with knots at $x_1, \ldots, x_n$, i.e., n DOF.
- Can show: as $\lambda \to \infty$, **effective degrees of freedom** df_λ decrease from n to 2.
- Smoothing spline has nominally n DOF, these are heavily constrained, i.e., they are "shrunk" by higher weighting of the penalty term.
- Measure of flexibility of smoothing splines: df_λ.
- Mapping from observation vector $\boldsymbol{y} \in \mathbb{R}^n$ to vector $\hat{\boldsymbol{g}}_\lambda$ of n coefficients defining the smoothing spline with penalty parameter λ is linear, i.e.,

$$\boldsymbol{S}_\lambda \hat{\boldsymbol{g}}_\lambda = \boldsymbol{y}, \qquad \boldsymbol{S}_\lambda \in \mathbb{R}^{n\times n}.$$

Effective DOF defined by

$$df_\lambda := \operatorname{tr} \boldsymbol{S}_\lambda = \sum_{i=1}^{n} [\boldsymbol{S}_\lambda]_{i,i}.$$

Nonlinear Regression Models

Smoothing splines: choosing λ

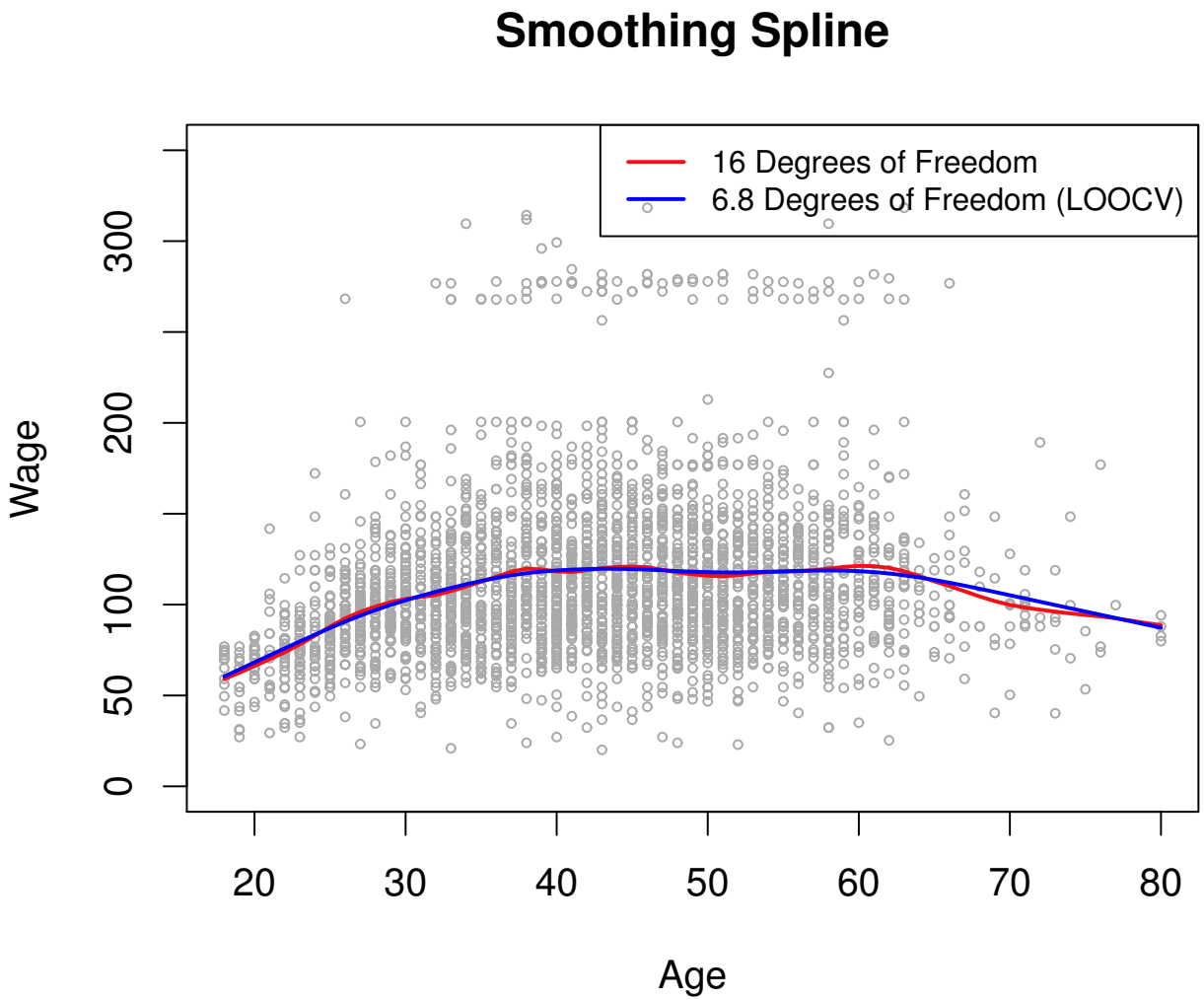
- For smoothing splines no need to choose knot number and locations, each predictor observation x_i is a knot.
- Remaining problem is choice of smoothing parameter λ.
- Obvious option: choose λ to minimize CV estimates of RSS.
- For smoothing splines LOOCV error can be computed at nearly the cost of single fit:

$$\mathrm{RSS}_{CV}(\lambda) = \sum_{i=1}^{n}(y_i - \hat{g}_\lambda^{(-i)}(x_i))^2 = \sum_{i=1}^{n}\left[\frac{y_i - \hat{g}_\lambda(x_i)}{1 - [\boldsymbol{S}_\lambda]_{i,i}}\right]^2,$$

$\hat{g}_\lambda^{(-i)}(x_i)$: value of smoothing spline fitted with all but i-th observation,

$\hat{g}_\lambda(x_i)$: value of smoothing spline using all observations.

- Similar "magic formula" in (5.1) for LS regression.



Smoothing spline fit to `Wage` data. Red: specified 16 effective DOF.
Blue: λ determined by LOOCV, resulting in $df_\lambda = 6.8$.

Contents

Nonlinear Regression Models

Generalized additive models

- Up to now: single predictor X, extensions of simple linear regression.
- Here: consider extensions of multiple linear regression of response Y on predictors $X_1, \ldots, X_p$.
- Framework: **generalized additive models** (GAMs).
- Allow nonlinear functions of X_j while maintaining additivity.
- Can be applied with quantitative and qualitative responses.

- Extend standard multiple linear regression model

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_p x_{i,p} + \varepsilon_i$$

to

$$y_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{i,j}) + \varepsilon_i.$$

- Additive: separate f_j for each X_j, then add.
- Example: Consider natural splines and task of fitting model

$$\texttt{wage} = \beta_0 + f_1(\texttt{year}) + f_2(\texttt{age}) + f_3(\texttt{education}) + \varepsilon \qquad (7.4)$$

from `Wage` data set, with quantitative variables `year`, `age` and qualitative variable `education` $\in$ {`<HS`, `HS`, `<Coll`, `Coll`, `>Coll`}. Fit f_1, f_2 using natural splines, f_3 using separate constant for each value (dummy variable approach).

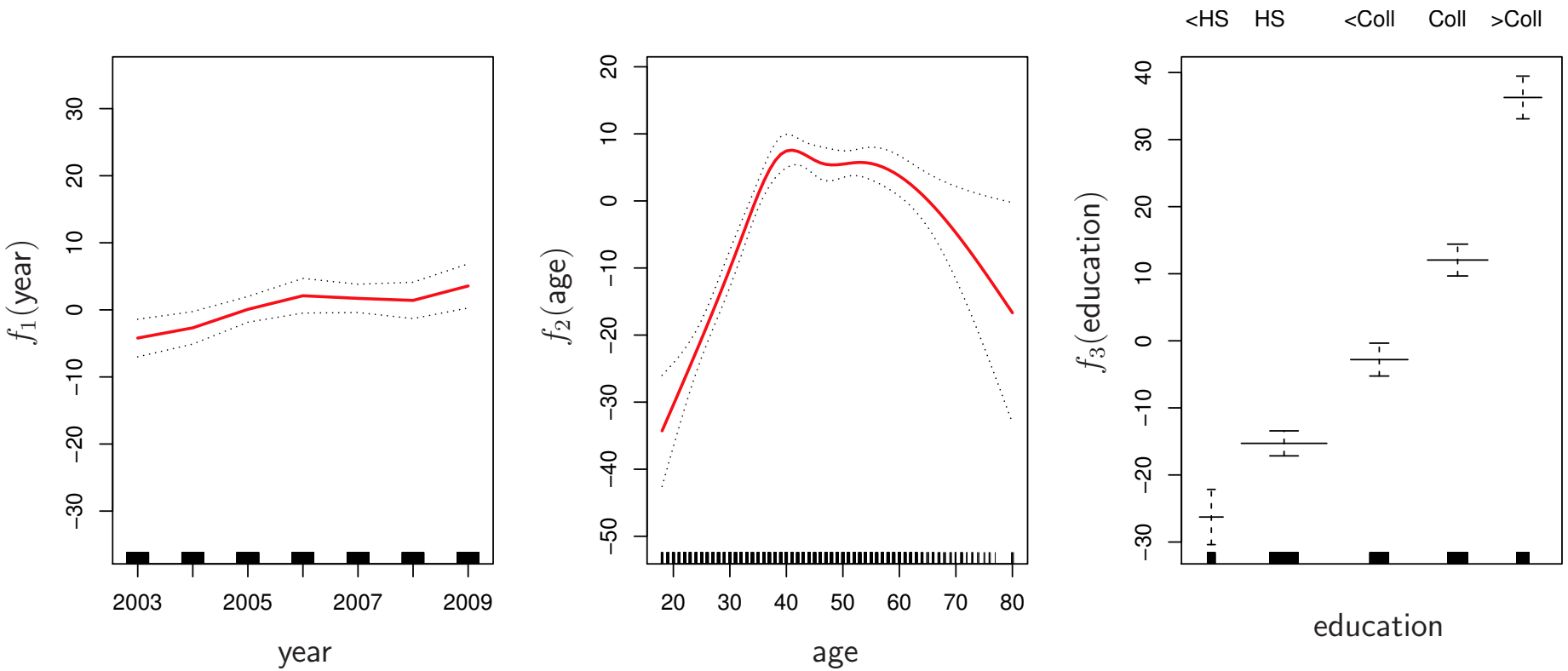
Nonlinear Regression Models

GAMs for regression

- Fit entire model (7.4) using LS, expand each function in natural spline basis or dummy variables, resulting in single large regression matrix.

Nonlinear Regression Models

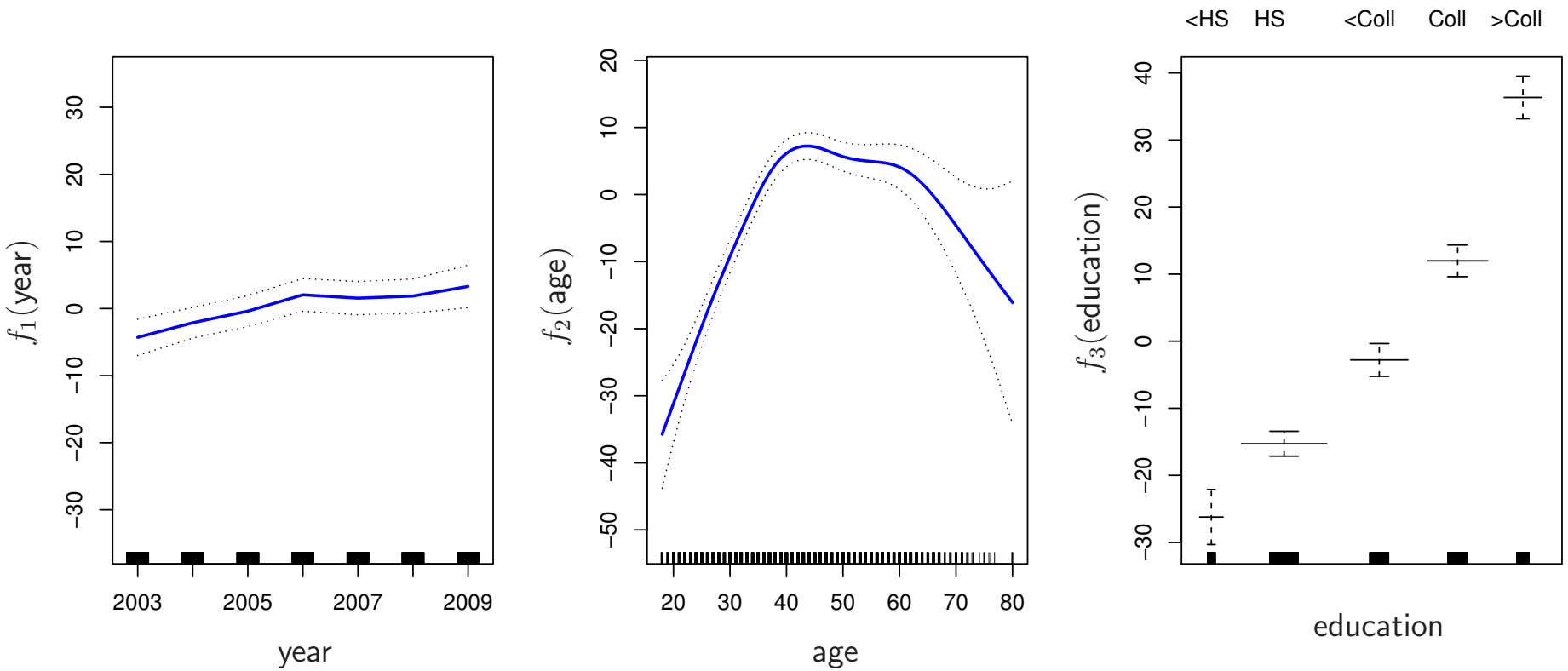
GAMs for regression



Relationship of each feature and response (`wage`). f_1 and f_2 are natural splines in `year` and `age` with 4 and 5 DOF, respectively. f_3 is a step function fit to qualitative predictor `education`.

Nonlinear Regression Models

GAMs for regression



Same as before except f_1 and f_2 smoothing splines with 4 and 5 DOF, respectively. Fit of smoothinmg splines more difficult than for natural splines, standard software solves an optimization problem via an algorithm known as **backfitting**.

Nonlinear Regression Models

GAMs: benefits and shortcomings

+ GAMs allow fitting nonlinear f_j to each X_j in order to capture nonlinear dependencies.
+ Potentially more accurate predictions of response Y.
+ Model still additive, effect of each X_j can be examoned separately, useful for inference.
+ Smoothness of each f_j can be summarized via (effective) DOF.
- Additivity is a restriction, interactions can be missed. Can add interaction terms manually by adding predictors $X_j \times X_k$ or low degree interaction functions $f_{j,k}(X_j, X_k)$.

GAMs are a useful compromise between linear and fully nonparametric methods such as random forests and boosting (later).

Contents

Tree-Based Methods

Chapter overview

- In previous chapter, considered *piecewise* approximation for univariate models (piecewise constant, splines, etc.).
- Here: piecewise constant *multivariate approximation*.
- Much greater variety of possible domain partitions.
- **Recursive binary partitioning**: efficient representation using **binary trees**.
- Can be used for regression and classification.
- Refinements: **bagging**, **random forests**, **boosting**.

Contents

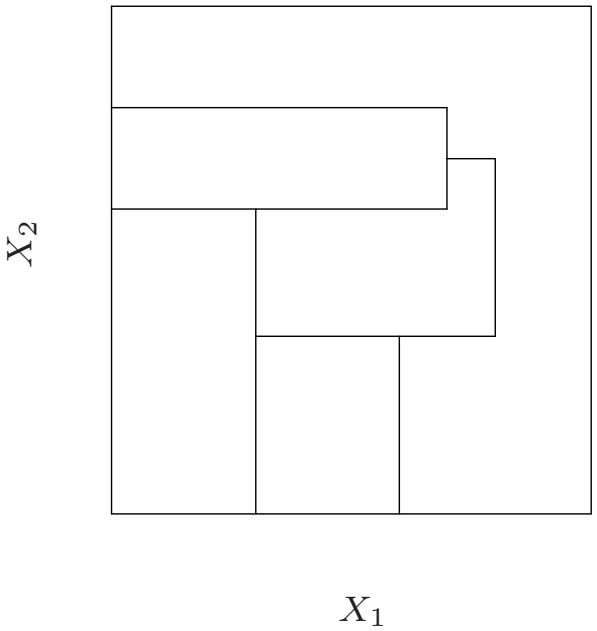
Tree-Based Methods

Basic idea

- Consider bivariate model

$$Y = f(X_1, X_2), \quad X_i \in [0, 1].$$

- Divide feature space into axis-aligned rectangles.
- Within each rectangle, predict Y as the mean of the observations it contains.

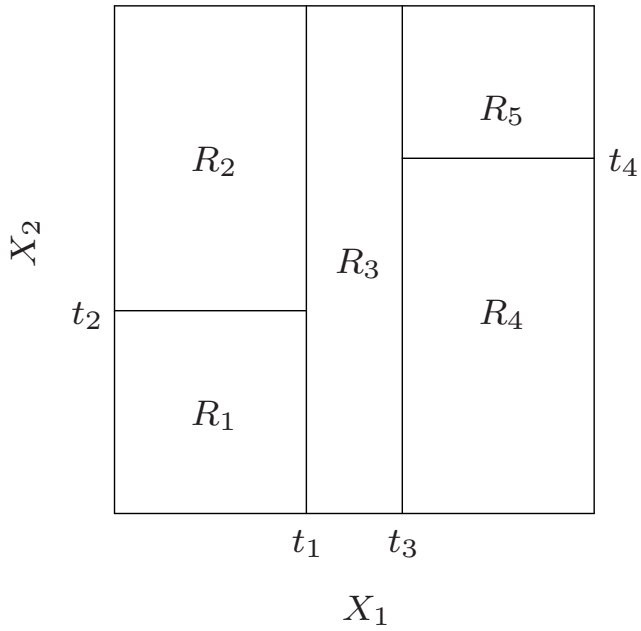


Tree-Based Methods

Basic idea

- Simpler structure: construct rectangles by **recursive binary partitioning**.
- Predicting $Y = \hat{y}_{R_m}$ in region R_m yields piecewise constant model

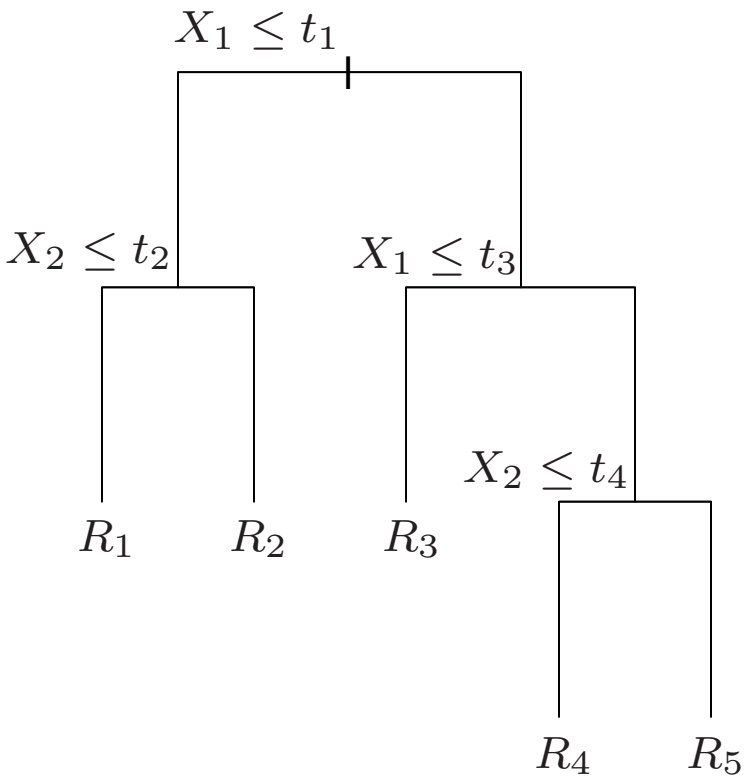
$$Y = \hat{f}(X) = \sum_{m=1}^{5} \hat{y}_{R_m} \mathbb{1}(X \in R_m)$$

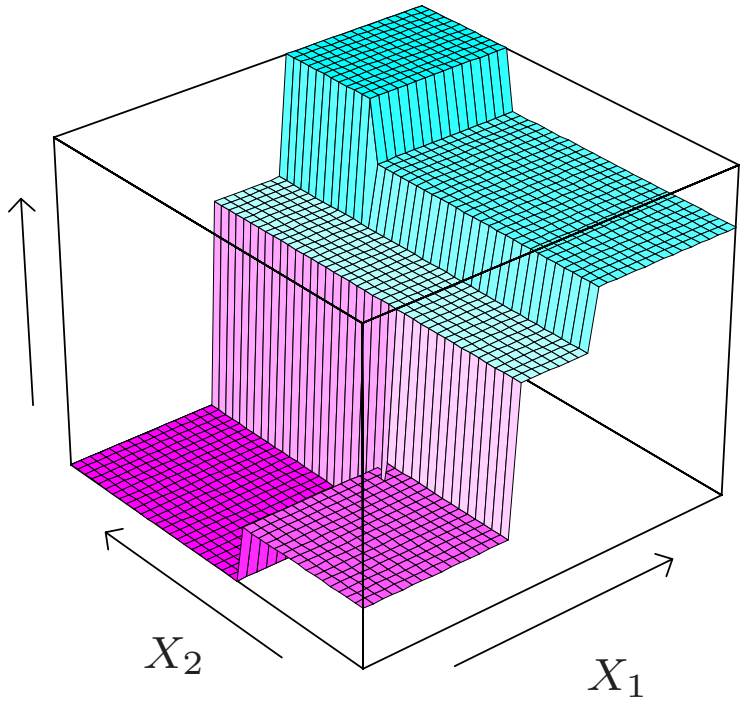


- Partitioning sequence:
  - First split at $X = t_1$.
  - Next, split region $X_1 \leq t_1$ at $X_2 = t_2$ and region $X_1 > t_1$ at $X_1 = t_3$.
  - Finally: region $X_1 > t_3$ is split at $X_2 = t_4$.

Tree-Based Methods

Basic idea





Recursive binary partition more conveniently represented by a binary tree; regions appear as **leaves**, internal nodes are the splits.

Equivalent representation as piecewise constant function.

Tree-Based Methods

Hitters example

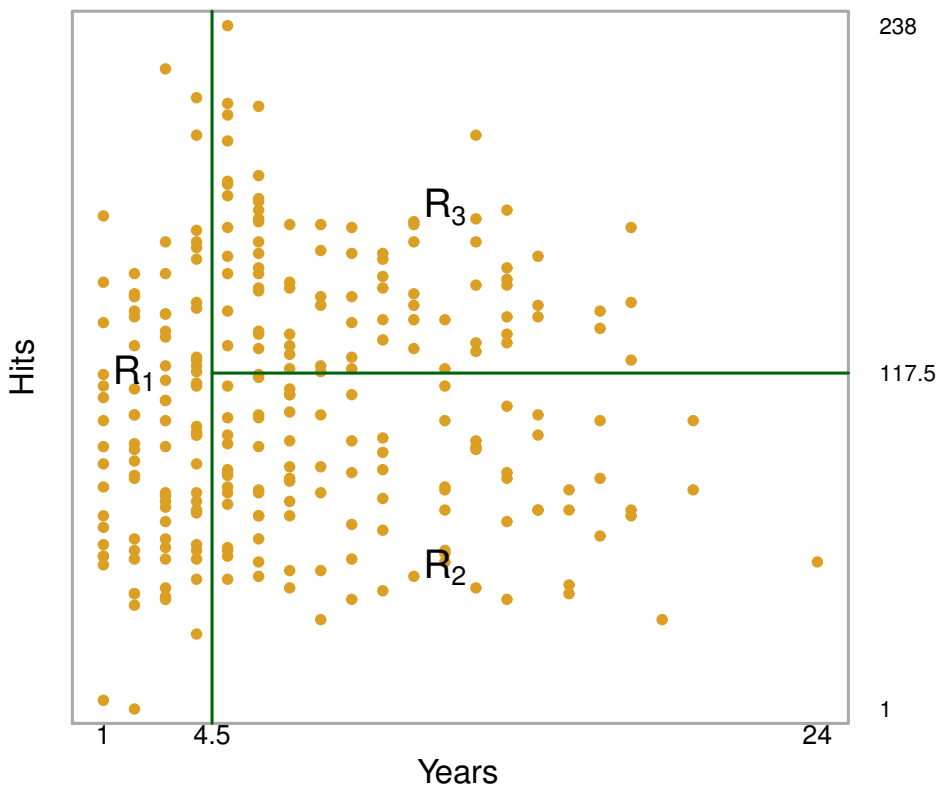
- `Hitters` data set: predict baseball player's `Salary` based on

  `Years`: # years played in major leagues
  `Hits` : # hits made in previous year

- First remove observations missing `Salary` values.
- Log-transform `Salary` values [k$] to make distribution more bell-shaped.

Tree-Based Methods

Hitters example



Observations of `Years` and `Hits` with partitioning arising from two splits.

- First split yields
  $R_1 = \{X : \texttt{Years} < 4.5\}$.
- Second split at $\texttt{Hits} = 117.5$ yields
  $R_2 = \{X : \texttt{Years} \geq 4.5, \texttt{Hits} < 117.5\}$
  and
  $R_3 = \{X : \texttt{Years} \geq 4.5, \texttt{Hits} \geq 117.5\}$
- Predicted `Salary` in these regions:
  $R_1 : \$1000 \times \mathrm{e}^{5.107} = \$165,174,$
  $R_2 : \$1000 \times \mathrm{e}^{5.999} = \$402,838,$
  $R_3 : \$1000 \times \mathrm{e}^{6.740} = \$845,346.$

Tree-Based Methods

Hitters example

Regression tree resulting from these splits.

- Left **branch** contains R_1, right branch R_2 and R_3.
- Tree has two **internal nodes**, three **terminal nodes** (leaves).
- Number in each leaf gives mean value of log(`Salary`) for corresponding region.
- Interpretation: `Years` is most important factor in determining `Salary` (less experienced players earn less); `Hits` important `Salary`-relevant feature only among experienced players.
- Advantages of tree: easily interpretable, nice graphical representation.
- Known as **CART**: classification and regression tree.

- **Goal:** Partition feature space into high-dimensional rectangles $\{R_m\}_{m=1}^M$ in such a way that

$$\text{RSS} = \sum_{m=1}^{M} \sum_{i \in R_m} (y_i - \hat{y}_{R_m})^2,$$

  is minimized. ($\hat{y}_{R_m}$: mean of the response observations contained in R_m.)
- Comparing all possible partitions computationally infeasible, hence use **top-down**, **greedy** approach.
- Top-down refers to starting with the entire feature space and recursively splitting regions (**recursive binary splitting**).
- **Greedy** approach refers to determining the locally best split without looking ahead and possibly choosing a split leading to a better tree in some future step.

- To construct first split, consider splitting along variable X_j at splitting point $X_j = s$ and define half-spaces

$$R_1(j,s) := \{X : X_j \leq s\}, \qquad R_2(j,s) := \{X : X_j > s\}. \tag{8.1}$$

- Seek splitting variable index j and splitting point s which minimize

$$\min_{\hat{y}_{R_1}} \sum_{x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \min_{\hat{y}_{R_2}} \sum_{x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2. \tag{8.2}$$

- For fixed j, s, the two minimizing values of $\hat{y}_{R_1}$ and $\hat{y}_{R_2}$ are clearly the sample means of the response observations in R_1 and R_2, respectively.
- For each j, the optimal splitting point s can be found very quickly; with best split (j, s), partition data into the resulting two subregions and continue splitting recursively.

Tree-Based Methods

Tree pruning

- Can continue recursive binary splitting until, e.g., cardinality of all leaves fall below given value.
- If tree too complex, danger of overfitting: smaller tree (fewer splits) will have lower variance. Hence, alternative stopping criterion could be minimal reduction of RSS for each split.
  However, split with small RSS reduction may enable larger reduction in subsequent splits.
- Better strategy: grow very large tree T_0, then **prune** it back to obtain a **subtree**.
- Can compare different subtrees using **cross-validation**, but comparing all possible subtrees is infeasible.

- **Cost complexity pruning** (a.k.a. *weakest link pruning*): consider sequence of trees indexed by tuning parameter $\alpha \geq 0$.
- To each $\alpha \geq 0$ there corresponds a subtree $T \subset T_0$ which minimizes

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|, \tag{8.3}$$

  where $|T|$ denotes the number of leaves of tree T. Tuning parameter α controls trade-off between fit to training data and tree complexity.
- $\alpha = 0$ corresponds to T_0. For $\alpha > 0$ (8.3) minimized by smaller tree T_α (can show this is unique).
- To find T_α use **weakest link pruning**: successively collapse internal node producing smallest per-node increase in $\sum_{m,i}(y_i - \hat{y}_{R_m})^2$, continue until single-node tree reached. Can show: this tree sequence must contain T_α.
- Select α using validation set or cross-validation.

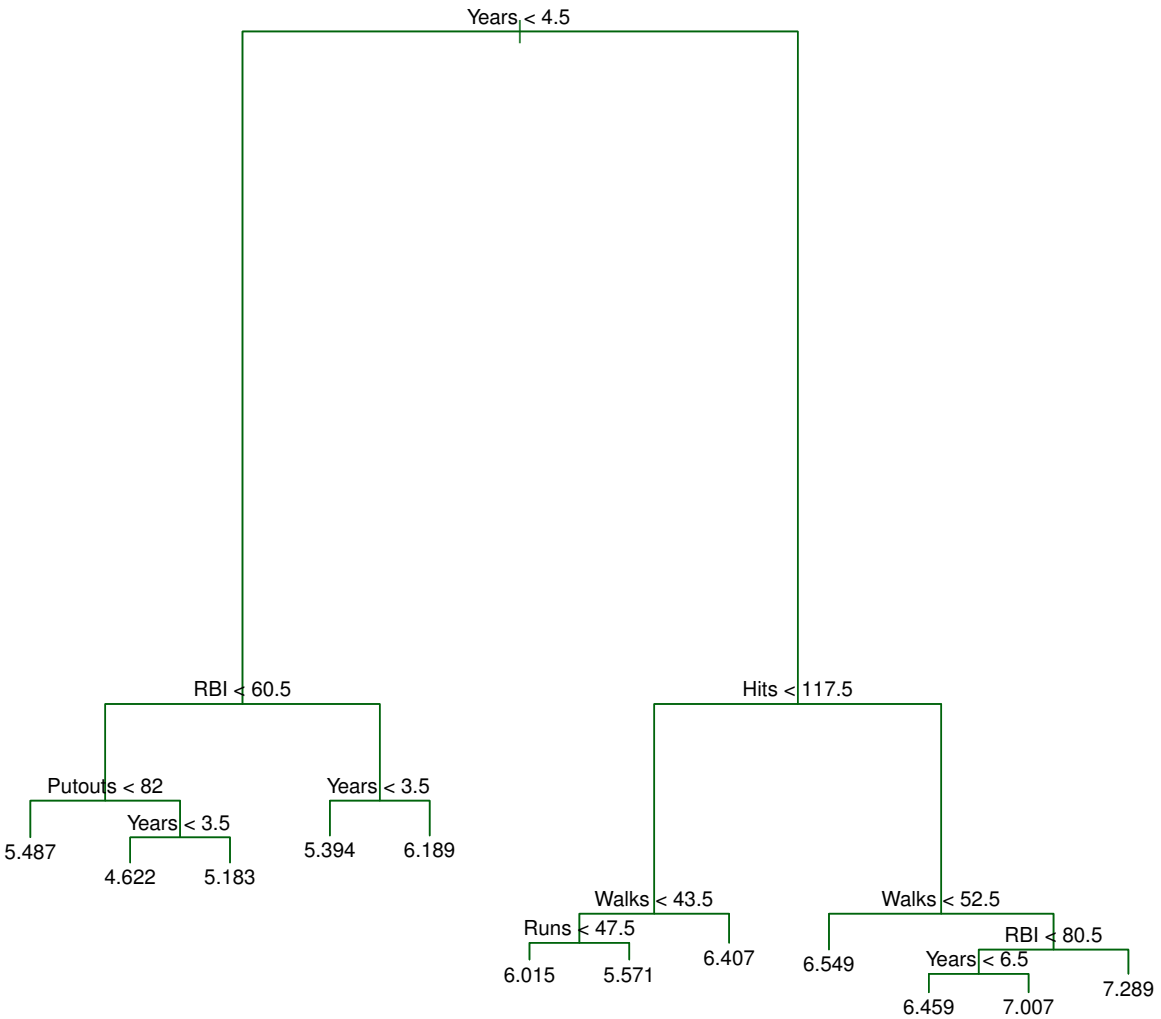
Algorithm 4: Regression tree.

1. Use recursive binary splitting to grow tree T_0 on the training data, stopping when each leaf contains fewer than some minimum number of observations.
2. Apply cost complexity pruning to T_0 to obtain sequence of best subtrees, as a function of α.
3. Use K-fold cross-validation to choose α: divide training observations into K folds. For each $k = 1, \ldots, K$:
   i. Repeat steps 1 and 2 on all but k-th fold of training data.
   ii. Evaluate test MSE on left out k-th fold, as function of α.

   Average MSE for each value of α, choode α minimizing average MSE.
4. Return subtree from Step 2 corresponding to minimizing α.

Tree-Based Methods

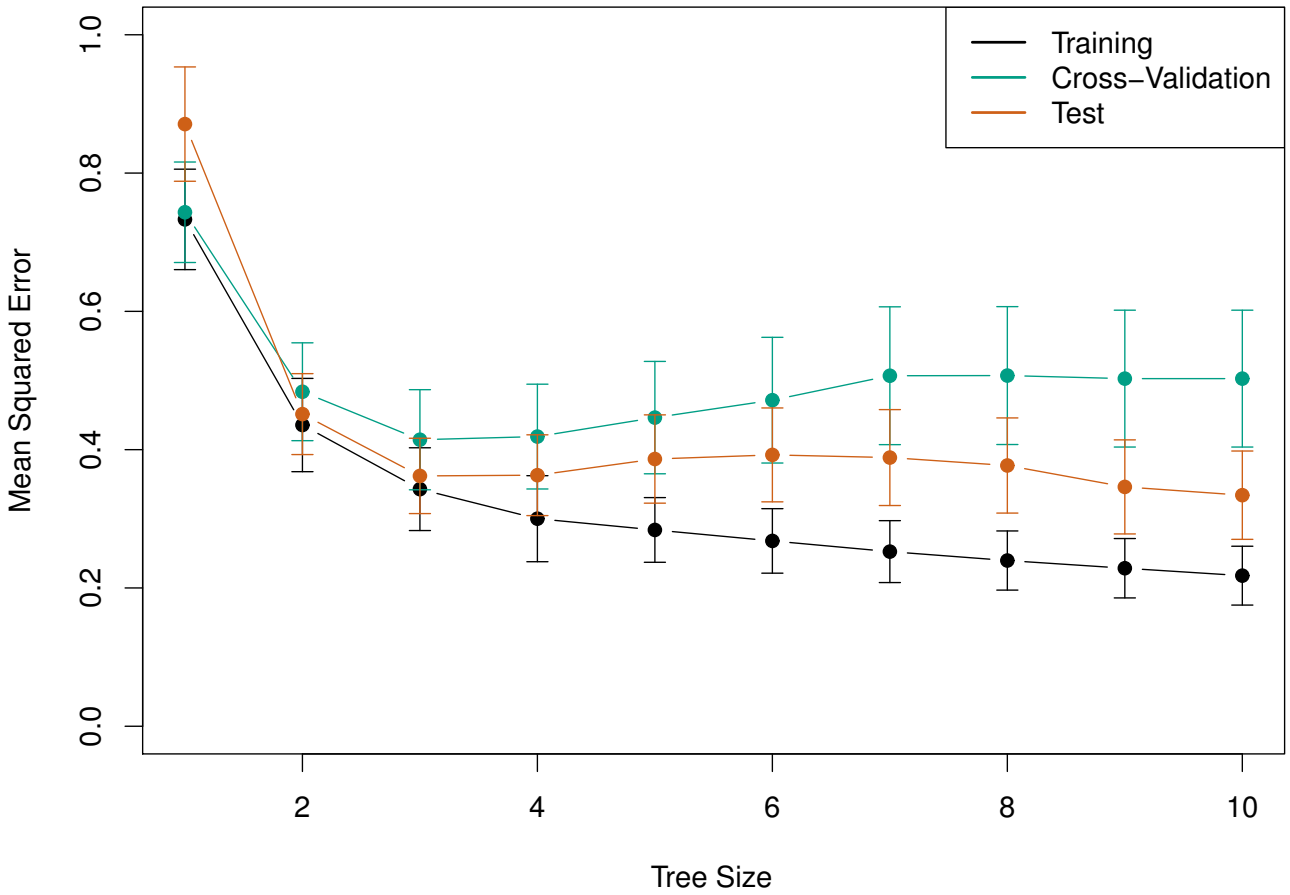
Hitters example revisited



- `Hitters` data set using nine features.
- Randomly divide data set into 132 training and 131 test observations.
- Grow tree on training data.
- Vary α to obtain subtrees T_α with different numbers of leaves.
- Perform 6−fold CV to estimate MSE of T_α as function of α.
- Unpruned tree shown on left.

Tree-Based Methods

Hitters example revisited



Training, CV and test MSEs for regression tree of `Hitters` data set as a function of α with bands indicating ± 1 standard error. CV MSE somewhat pessimistic, but reasonable estimate of test MSE.

- Tree-based piecewise constant prediction model for qualitative response.
- In place of mean value, predict in R_m the *most commonly occurring* response observation there (**majority vote**).
- Grow classification tree using recursive binary splitting.
- In place of RSS, can use **classification error rate** E to determine optimal splits. This is simply the fraction of training observations not belonging to the most commonly occurring class, i.e.

$$E = 1 - \max_k \hat{p}_{m,k},$$

  $\hat{p}_{m,k}$: proportion of training observations in R_m from k-th class.
- Classification error rate E not sensitive enough for tree-growing. Two other popular measures preferable:

- The **Gini index** is defined by

$$G = \sum_{k=1}^{K} \hat{p}_{m,k}(1 - \hat{p}_{m,k}) \tag{8.4}$$

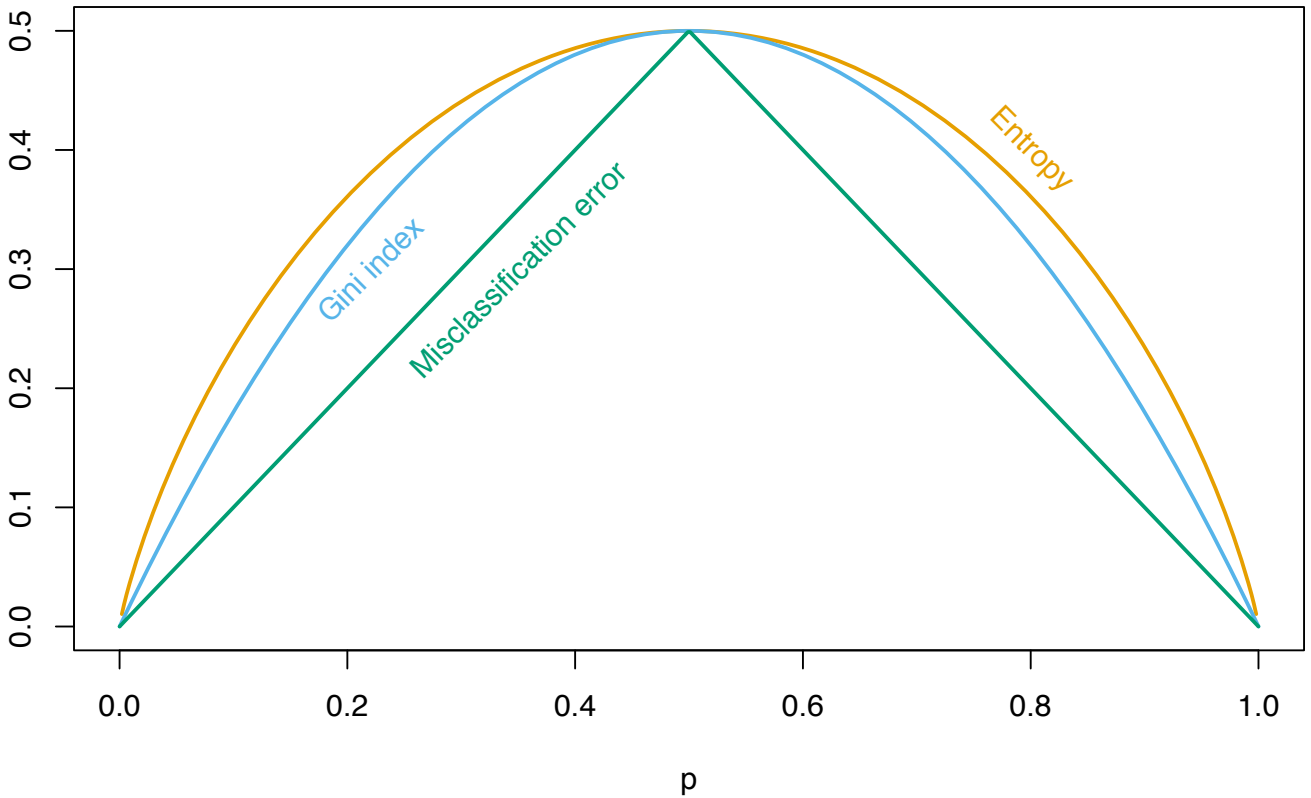
  and represents a measure of total variance across the K classes. Small if all $\hat{p}_{m,k}$ close to zero or one; indication of node *purity*, i.e., small value indicates node contains predominantly observations from a single class.
- **Entropy** (or **deviance**) is defined by

$$D = -\sum_{k=1}^{K} \hat{p}_{m,k} \log \hat{p}_{m,k}. \tag{8.5}$$

  Note $\hat{p}_{m,k} \log \hat{p}_{m,k} \leq 0$ since $\hat{p}_{m,k} \in [0, 1]$.
  As for G, D small if $\hat{p}_{m,k}$ close to zero or one for all k.
- Any of E, G or D can be used to build the tree, but pruning should be done using E to maximize prediction accuracy of final pruned tree.

Tree-Based Methods

Gini index and entropy



Node purity measures for two-class classification as a function of proportion p in class 2. Entropy has been scaled to pass through $(0.5, 0.5)$. For two classes, if p denotes the proportion in class 2, the three measures are $1 - \max(p, 1-p)$, $2p(1-p)$ and $-p\log p - (1-p)\log(1-p)$.

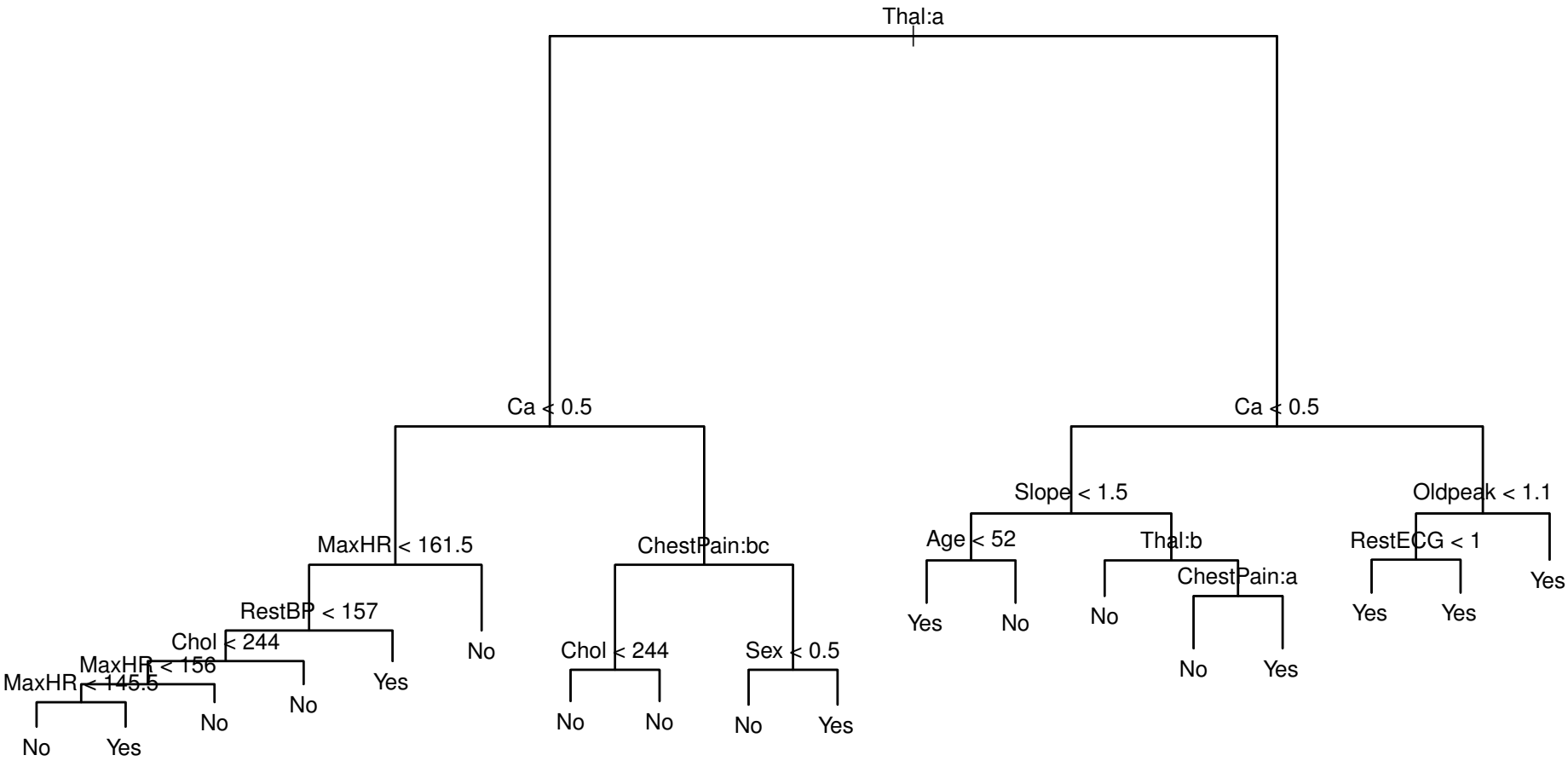
Tree-Based Methods

Heart example

- `Heart` data set: binary response `HD` for 303 patients who presented with chest pain.
- Response `Yes` indicates presence of heart disease (based on angiographic test), `No` indicates absence of heart disease.
- 13 predictors including `Age`, `Sex`, `Chol` (cholesterol measurement), and further heart and lung function measurements.
- Cross-validation results in tree with six leaves.

Tree-Based Methods

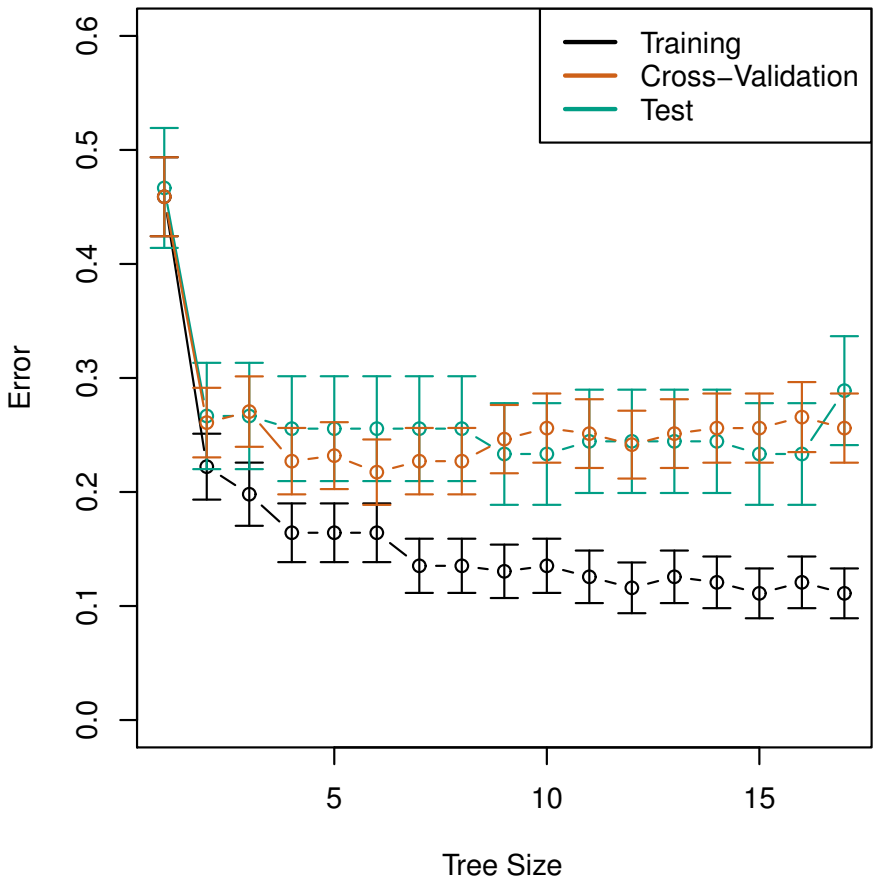
Heart example

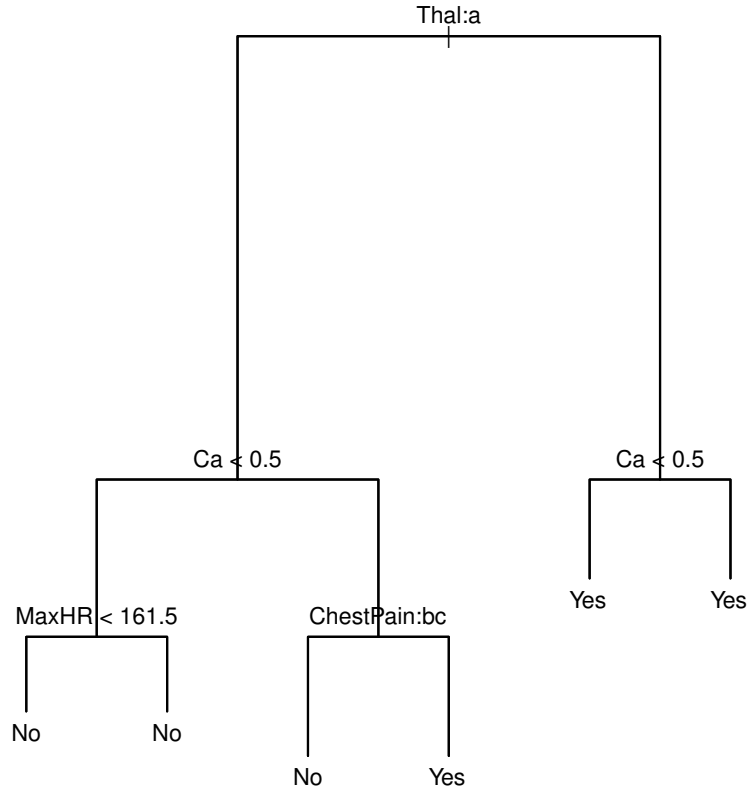


`Heart` data set: unpruned tree.

Tree-Based Methods

Heart example





`Heart` data set. Left: training, CV and test MSE for different sizes of pruned tree. Right: pruned tree corresponding to minimal CV MSE.

Tree-Based Methods

Heart example, qualitative predictors

- `Heart` data set contains a number of *qualitative* predictor variables such as `Sex`, `Thal` (Thallium stress test) and `ChestPain`.
- Splitting along one of these variables: assign some of the qualitative values to one branch, remaining values to other branch.
- In previous image: some internal nodes split quantitative variables.
- Top internal node splits `Thal`. Text `Thal:a` indicates left branch consists of observations with first value of `Thal` (normal), right consists of remaining values (fixed or reversible defects).
- Text `ChestPain:bc` on third split on left indicates left branch contains observations with second and third values of `ChestPain` variable (whose possible values are *typical angina*, *atypical angina*, *non-anginal pain* and *asymptomatic*).

Tree-Based Methods

Heart example, leaves with identical values

- Some leaves in `Heart` classification tree have the same predicion values.
- Example: split `RestECG` < 1 near bottom right of unpruned tree, both subregions predict response value `Yes`. Why perform split in the first place?
- Split made to increase *node purity*.
- All 9 observations in right branch have leaf response value `Yes`. In left branch, 7/11 have response value `Yes`.
- Importance of node purity: given test observation belonging to region on right branch, then response certaionly `Yes`. For test observation on left branch, pobably `Yes`, but with much less certainty.
- Even though `RestECG` < 1 does not reduce classification error, it improves the Gini index and entropy, which are more sensitive to node purity.

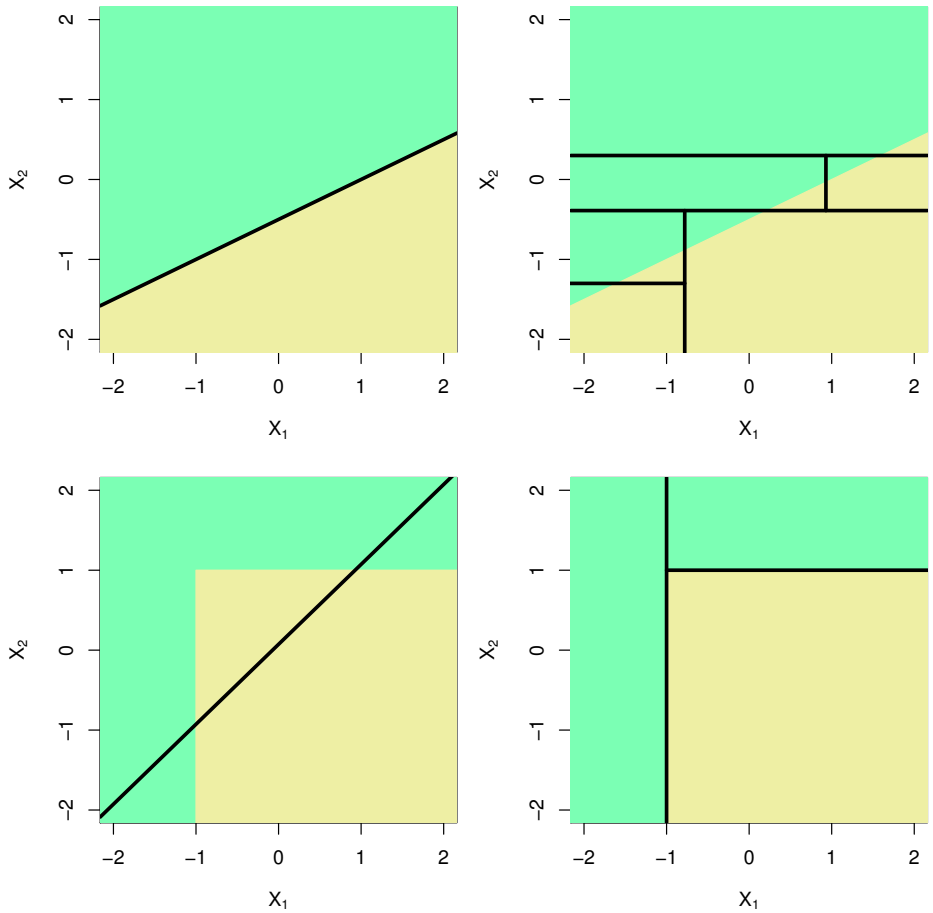
- Prediction model of linear regression vs. regression tree

$$f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p, \qquad f(X) = \sum_{m=1}^{M} \hat{\boldsymbol{y}}_{R_m} \mathbb{1}_{\{X \in R_m\}}$$

  with regression coefficients $\{\beta_j\}_{j=0}^{p}$ and partition of feature space into rectangular regions R_m.
- If feature-response relation close to linear, linear regression model likely superior.
  Otherwise, tree-based models may outperform linear regression.
- Relative performances can be assessed by estimating test MSE via CV or validation set approach.
- Other considerations may also be relevant in comparison, such as interpretability or visualization.

Tree-Based Methods

Trees vs. linear models: example



Top row: 2D classification example with linear decision boundary (shaded regions), linear regression model superior. Bottom row: nonlinear)(piecewise constant) decision boundary captured perfectly by tree-based method,

Tree-Based Methods

Advantages and shortcomings of trees

+ Easy to explain (more so than linear regression).
+ Some argue decision trees more closely mimick human decision-making than linear regression/classification techniques (also widely used outside of statistical learning).
+ Trees, particularly small ones, easily displayed graphically, easily interpreted by non-experts.
+ Can handle qualitative predictors without introducing dummy variables.
- Prediction accuracy generally not as good as classical regression and classification techniques.
- Robustness (stability w.r.t. small data changes) often lacking.

Some of these disadvantages addressed by bagging, random forests, boosting.

Contents

- Recall the **bootstrap** approach introduced in Chapter 5 for randomly generating subsamples of a set of observations for estimating statistical quantities without collecting additional data.
- Here we revisit the bootstrap to show how it can be used as a **variance-reduction technique** for any statistical learning method.
- This is particularly relevant for decision trees, which tend to possess high variance.
- **Bagging**: **b**ootstrap **agg**regation.
- Variance can be reduced by averaging observations: for $\{X_k\}_{k=1}^N$ i.i.d. RV with variance σ^2, variance of $\overline{X} = (X_1 + \cdots + X_N)/N$ is σ^2/N.
- **Idea:** Collect N training sets, construct prediction model $\hat{f}_k$ for each, and average these to aggregate model

$$\hat{f}_{\text{avg}}(x) = \frac{1}{N}\sum_{k=1}^{N} \hat{f}_k(x).$$

- Collecting N data sets generally infeasible.
- Bootstrap: randomly select (with replacement) N_b sets of samples from (single) original data set.
  For each resampled data set, construct prediction model $\hat{f}_k^*$, $k = 1, \ldots, N_b$.
  Form bootstrap aggregate model

$$\hat{f}_{\text{bag}}(x) = \frac{1}{N_b} \sum_{k=1}^{N_b} \hat{f}_k^*(x).$$

  This is called **bagging**.
- For (regression) decision trees: grow (unpruned) tree for each resampled data set and average them.
- For classification trees: replace average with **majority vote**, i.e., for each new predictor observation, have aggregate model predict that class occurring most commonly across all decision trees $\hat{f}_k^*$.

Tree-Based Methods

Out-of-bag error estimation

- Error of bagged model can be estimated without CV or validation sets by exploiting that we are using bootstrapped subsets of fixed observation set.
- Can show: on average, each bagged tree uses of 2/3 of the observations.
- Remaining third: "**out-of-bag**" (OOB) observations.
- For i-th observation: predict response using all trees for which it was OOB. Take average (regression) or majority vote (classification) to obtain aggregated prediction for all n observations, compare with response observation, i.e., MSE (regression) or classification error (classification), to obtain error estimate.
- Can show: for N_b sufficiently large, OOB error estimate virtually equivalent to LOOCV error estimate.
  This is a great benefit when performing bagging on large data sets, where CV would be computationally burdensome.

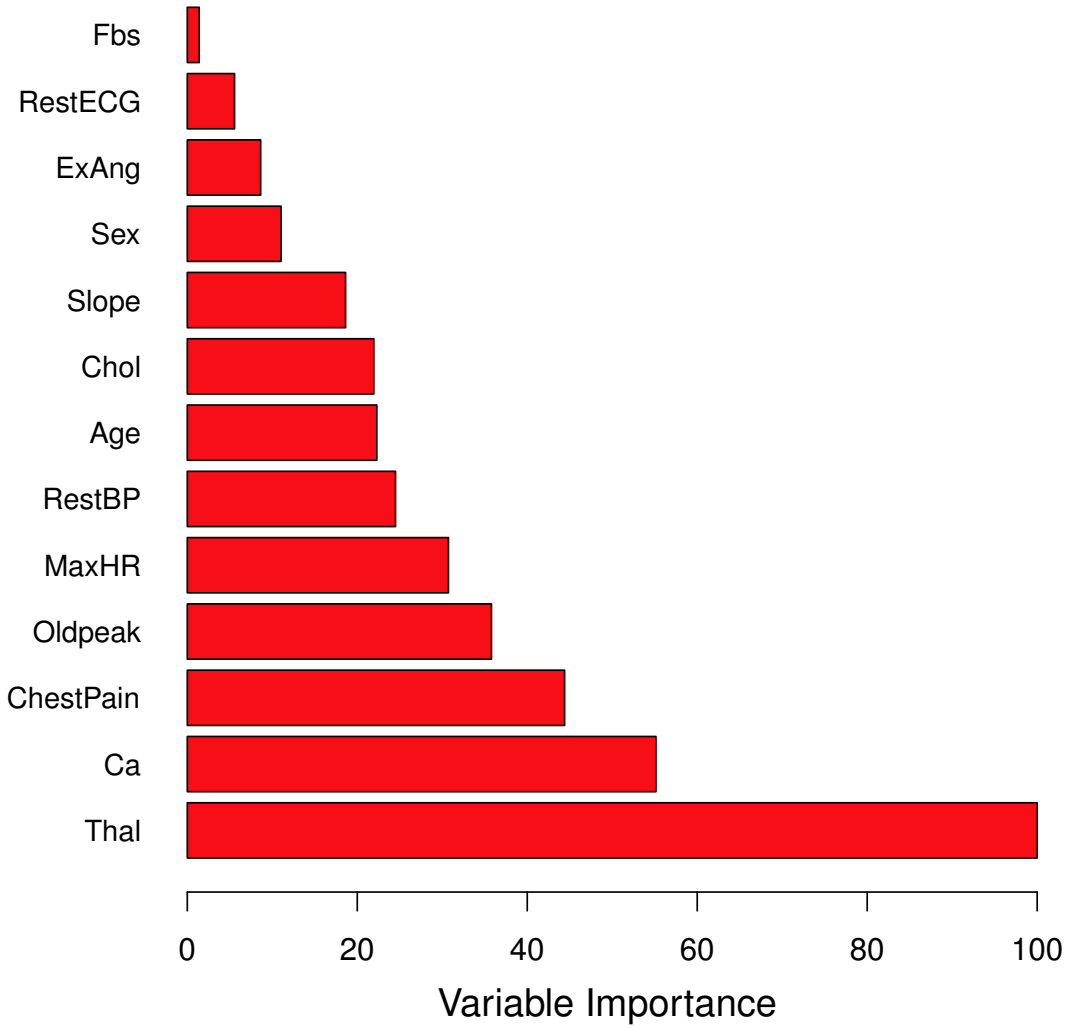
Tree-Based Methods

Measuring variable importance

- Drawback of bagging: no single decision tree for interpretation. Increased predicion accuracy at the expense of interpretability.
- Crucial interpretation element: which predictor variables most important?
- For regression trees, overall summary of importance of each predictor can be obtained by recording total amount RSS decreases when split performed along this variable, then averaging over all trees.
- For classification trees: record totel Gini index reduction due to splits along each predictor per tree, then average over all trees.
- Large value indicates important predictor variable.

Tree-Based Methods

Measuring variable importance: Heart data set



`Heart` data set. Variable importance (relative to maximum) in terms of mean decrease in Gini index.

- Improve over bagging by "**decorrelating**" trees.
- Build decision trees on bootstrapped samples as in bagging.
- When choosing next predictor variable to split, restrict selection to $m < p$ randomly chosen variables instead of full set of p predictors.
  New set of m splitting candidates chosen at each splitting step.
- Common choice: $m \approx \sqrt{p}$. Smaller m called for in case of many correlated predictors.
  $m = p$ recovers bagging.
- **Rationale:** strongly dominant variables will be used in splitting for majority of bagged trees, leading to similarity among these trees, and thereby strong correlations.
  This limits the variance reduction from averaging.
- On average, $(p-m)/p$ splits will not even consider a given strong predictor.
- This mechanism results in a decorrelation of the trees, making their average less variable, hence more reliable.

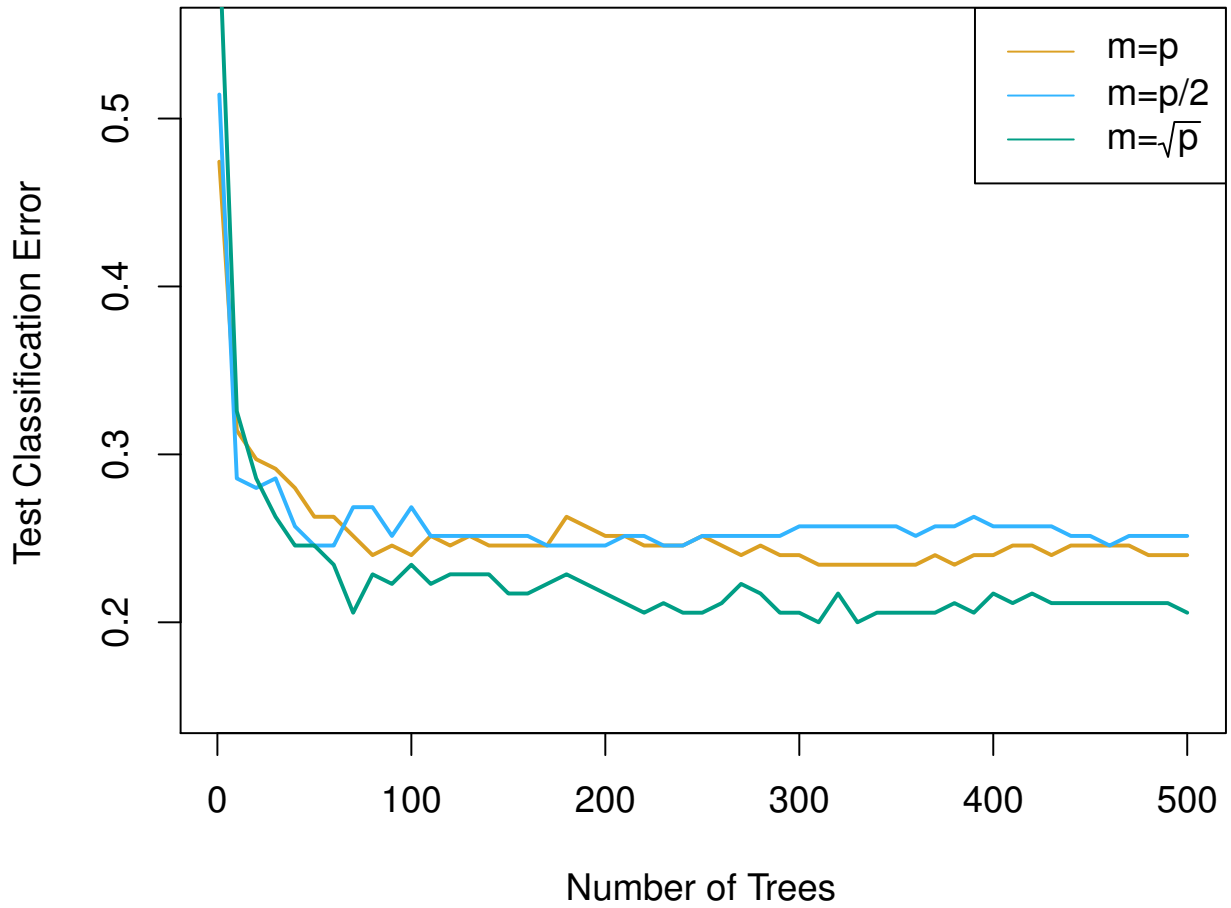
Tree-Based Methods

Random forests: gene expression example

- High-dimensional biological data set: expression measurements for 4,718 genes on tissue samples from 349 patients.
- Human genome contains $\approx 20,000$ genes.
- Individual genes have varying levels of **expression** (activity) in different body cells, tissue or biological conditions.
- Here: each patient sample assigned to one of 15 classes (normal or one of 14 cancer types).
- Goal: predict cancer type using random forests based on 500 genes with largest variance in training set.
- Random division into training and test set.
- Random forests applied for 3 different values of m.

Tree-Based Methods

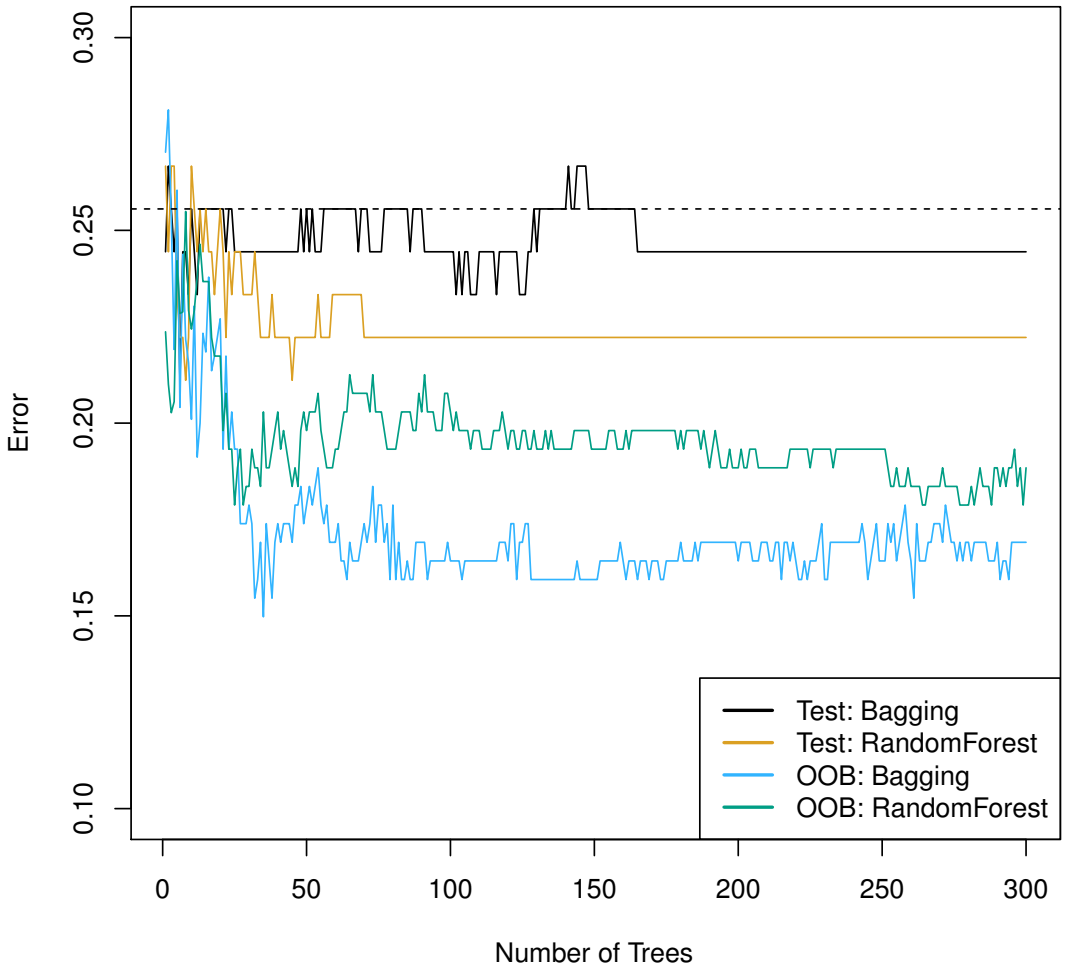
Random forests: gene expression example



Random forests for 15-class gene expression with $p = 500$: test error against # trees. Single tree has classification error rate of 45.7%. Null rate (always assign to dominant class) is 75.4%. As for bagging, no danger of overfitting as # trees increases.

Tree-Based Methods

Bagging vs. random forests: Heart data set



Test error against number N_b of bootstrapped data sets. Random forests used $m = \sqrt{p}$. Dashed line: error of single classification tree. Solid green/blue: OOB errors considerably lower.

- **Boosting:** general approach for improving predictions of statistical learning methods, here in context of decision trees.
- Basic approach as in bagging, but trees grown **sequentially** using information from previously generated trees.
- No bootstrap sampling; instead, each tree fit to a modified version of original data set.
- Prodedure: begin with tree fit to original data.
- Fit next tree to **residuals** of first model in place of observation responses. Then add this tree to the first, as a **model correction**.
- Each tree can be small (few leaves) determined by parameter d in the algorithm.
  By fitting small trees to residuals, $\hat{f}$ is slowly improved in areas where it previously didn't perform well.
- Boosting for classification trees slightly more complicated.

Tree-Based Methods

Tuning parameters in boosting

1. # trees N_b. Overfitting is possible with boosting, although it sets in slowly. Selection using CV.
2. Shrinkage parameter $\lambda > 0$ (small) determining learning rate.
   Typical values 10^{-2} or 10^{-3}.
   Very small λ can require very large N_b for good prediction.
3. # splits d per tree, controls complexity of boosted ensemble.
   Can also use $d = 1$ ("stump") with single split, leads to an **additive model**.
   Since d splits can involve at most d variables, it controls the **interaction order** of the boosted model.

Algorithm 5: Boosting for regression trees.

1. Set $\hat{f}(x) \equiv 0$ and $r_i = y_i$, $i = 1, \ldots, n$ (entire training set).
2. **for** $k = 1$ **to** N_n **do**
   Fit a tree $\hat{f}_k$ with d splits $(d+1)$ leaves to training data (X, r)
   Update $\hat{f}_k$ by adding damped version of new tree:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}_k(x)$$

   Update residuals

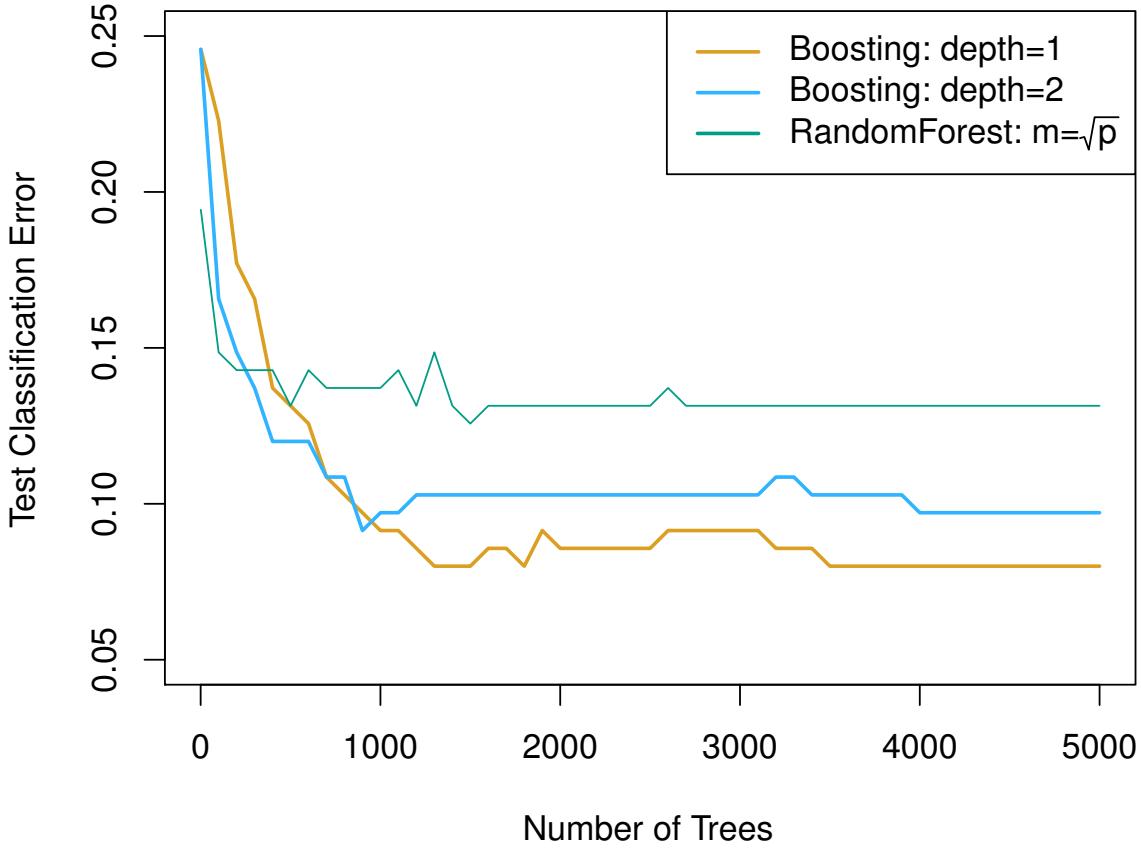
$$r_i \leftarrow r_i - \lambda \hat{f}_k(x_i), \quad i = 1, \ldots, n.$$

3. Output boosted model

$$\hat{f}(x) = \sum_{k=1}^{N_b} \lambda \hat{f}_k(x).$$

Tree-Based Methods

Gene expression example revisited



Boosting and random forests for gene expression example: test error against # trees using $\lambda = 0.01$ for boosted models. Depth-1 trees slightly outperform depth-2 trees, both outperform random forest, but difference is within standard error. Single tree has error rate 24%.

Contents

Contents

Support Vector Machines

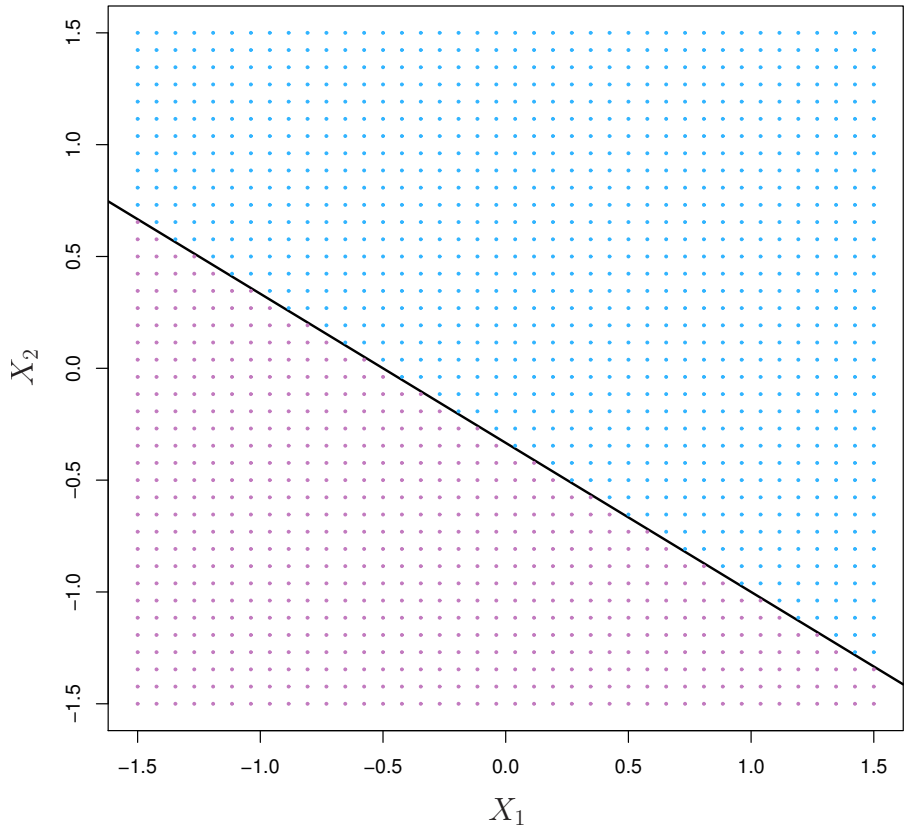
Hyperplanes for classification

- Assume a (true) bivariate classification model has a linear decision boundary.
- The points **on** the decision boundary are characterized by an equation of the form

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0.$$

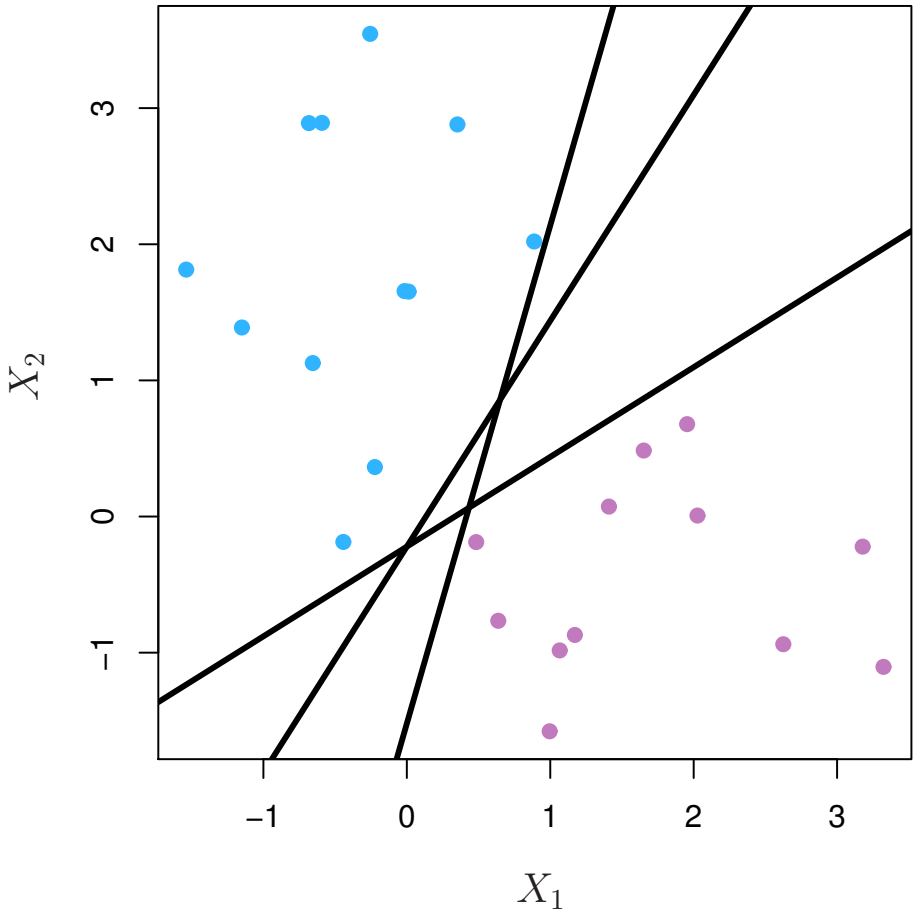
- The points on either side are characterized by

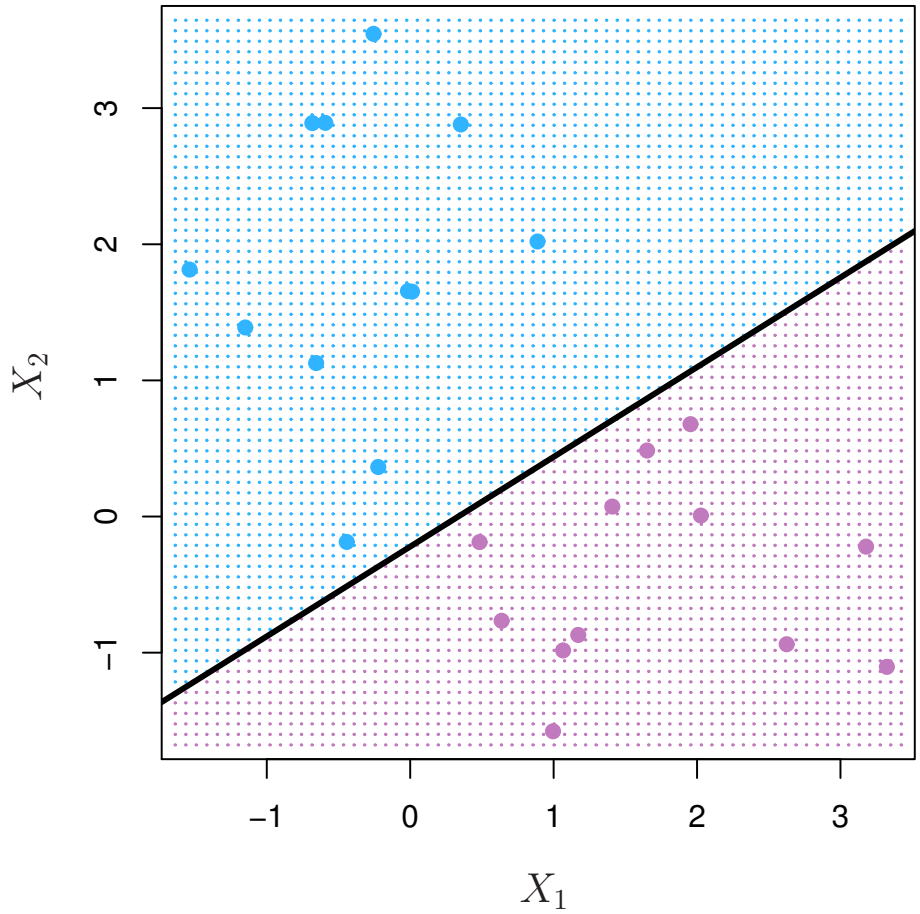
$$\beta_0+\beta_1 X_1+\beta_2 X_2 \begin{cases} > 0 & \text{one side,} \\ < 0 & \text{other side.} \end{cases}$$



Support Vector Machines

Hyperplanes for classification





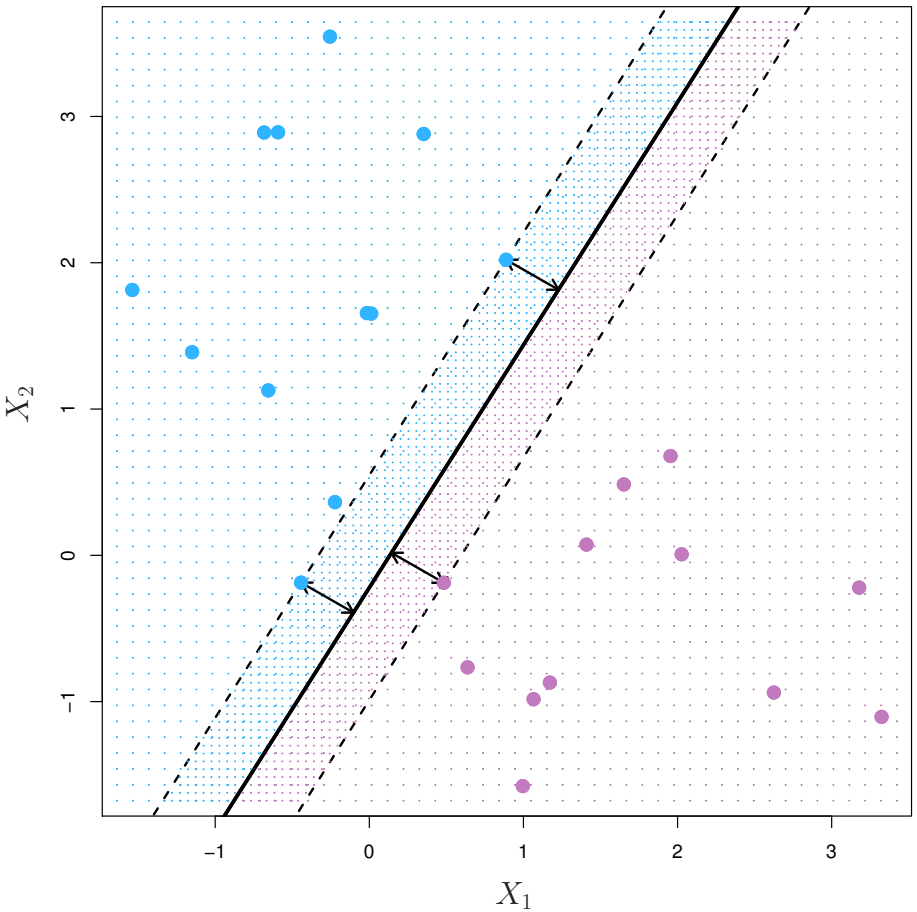
Left: Given a 2D data set of binary observations that can be split by a line, the line is in general not unique.

Right: Once a line has been fixed, this defines an associated classification model.

Support Vector Machines

Maximal margin classifier

- Compare with: KNN (Chapter 2), logistic regression, LDA, QDA (Chapter 4), decision trees (Chapter 8).
- Given binary observations, which linear decision boundary to choose?
- Separating line farthest from all training observations: **maximal margin hyperplane** or **optimal separating hyperplane**.
- Minimal distance from this line to closest observation: **margin**.



- Margin: half-width of largest slab separating the two observation set classes.
- Associated classification model: **maximal margin classifier**.

Support Vector Machines

Maximal margin classifier

- Large margin instills confidence in the classifier.
- In previous example: 3 points at marginal distance to separating hyperplane (there will always be at least 2, why?) These observations are called the **support vectors** of the maximal margin hyperplane, as moving these would move the latter.
- "Although the maximal margin classifier is often successful, it can also lead to overfitting when p is large."

Support Vector Machines

Constructing the maximal margin classifier

- **Given:** training set $\{(\boldsymbol{x}_j, y_j)\}_{j=1}^n$, $\boldsymbol{x}_j \in \mathbb{R}^p$, $y_j \in \{-1, 1\}$.
- **Goal:** determine coefficients $\beta_0, \beta_1, \ldots, \beta_p$ such that

$$\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p} \begin{cases} > 0 & \text{if } y_i = 1, \\ < 0 & \text{if } y_i = -1, \end{cases} \qquad i = 1, \ldots, n,$$

or, equivalently, $y_i(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p}) > 0$, $i = 1, \ldots, n$, such that the margin between the resulting hyperplane and the data set is maximal.

- Formally:

maximize M as a function of $\beta_0, \beta_1, \ldots, \beta_p, M$ such that

$$\sum_{j=1}^{p} \beta_j^2 = 1, \tag{9.1a}$$

$$y_i(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p}) \geq M, \quad i = 1, \ldots, n. \tag{9.1b}$$

Support Vector Machines

Constructing the maximal margin classifier

- (9.1) is a constrained optimization problem.
- Constraint (9.1b) ensures all observations on correct side of separating hyperplane if $M > 0$.
- Constraint (9.1a) not strictly necessary, merely scales M to coincide with margin.
- For details on solving the optimization problem see the reading list or a class on constrained optimization.
- Optimal separating hyperplane produces function $\hat{f}(\boldsymbol{x}) = \beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}, \boldsymbol{\beta} := (\beta_1, \ldots, \beta_p)^\top$ from which the classifier is derived as

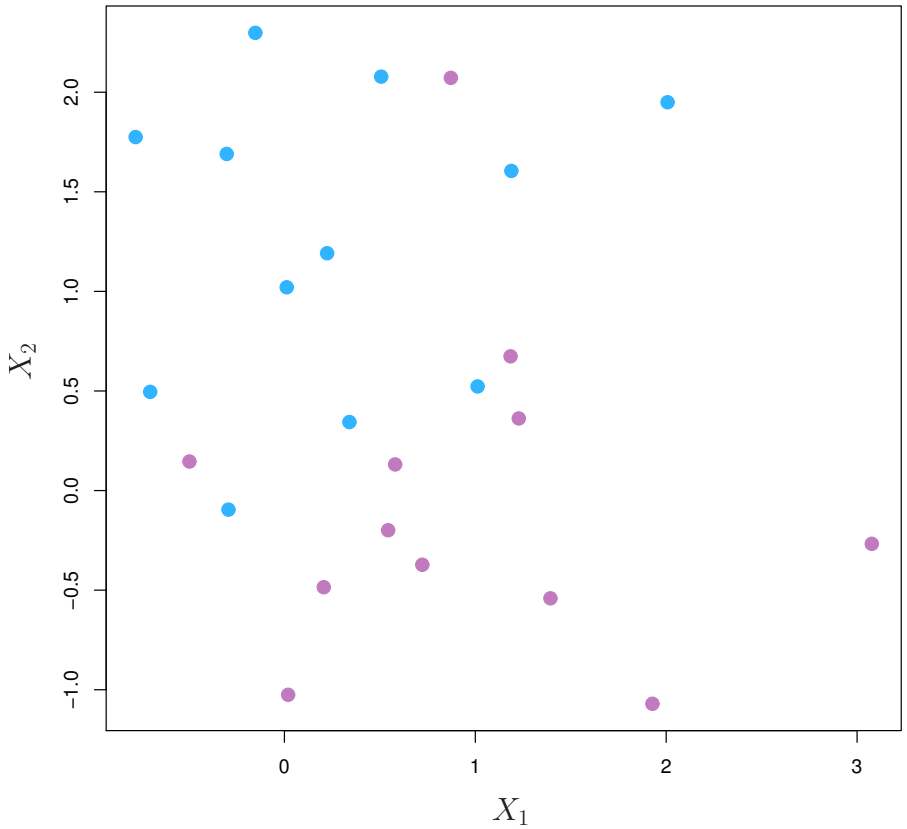
$$\hat{F}(\boldsymbol{x}) := \operatorname{sign} \hat{f}(\boldsymbol{x}).$$

- All training observations outside margin, not necessarily so for test observations. Intuition: large margin on training data will result in good separation of test data.

Support Vector Machines

Non-separable case

- A maximal margin classifier (or even a separating hyperplane) need not exist for a given data set.
- In this case optimization problem (9.1) has no solution with $M > 0$.
- Will extend concept of separating hyperplane using so-called **soft margin** to almost separate the classes.
- This leads to **support vector classifier**.

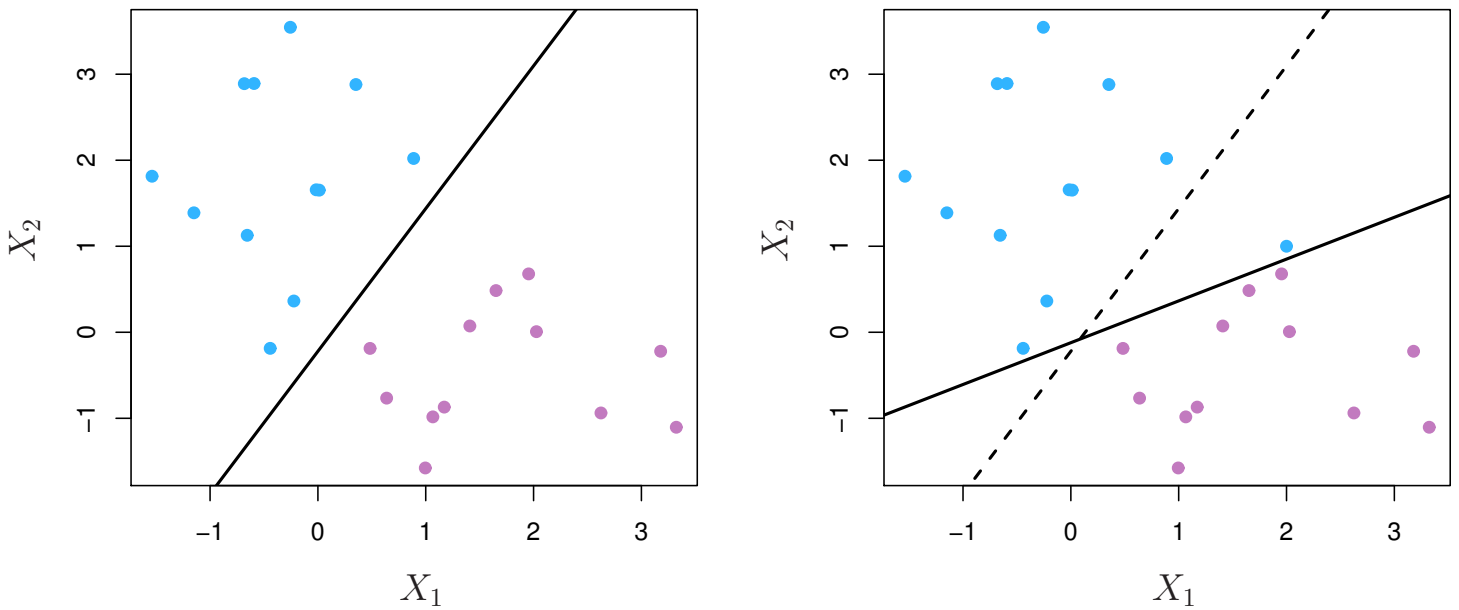


Contents

Support Vector Classifiers

Shortcomings of exact hyperplane classifiers

- Even if separating hyperplane exists, associated classifier possibly undesirable.
- Example: single additional blue point dramatically changes maximal margin hyperplane, leading to tiny margin.

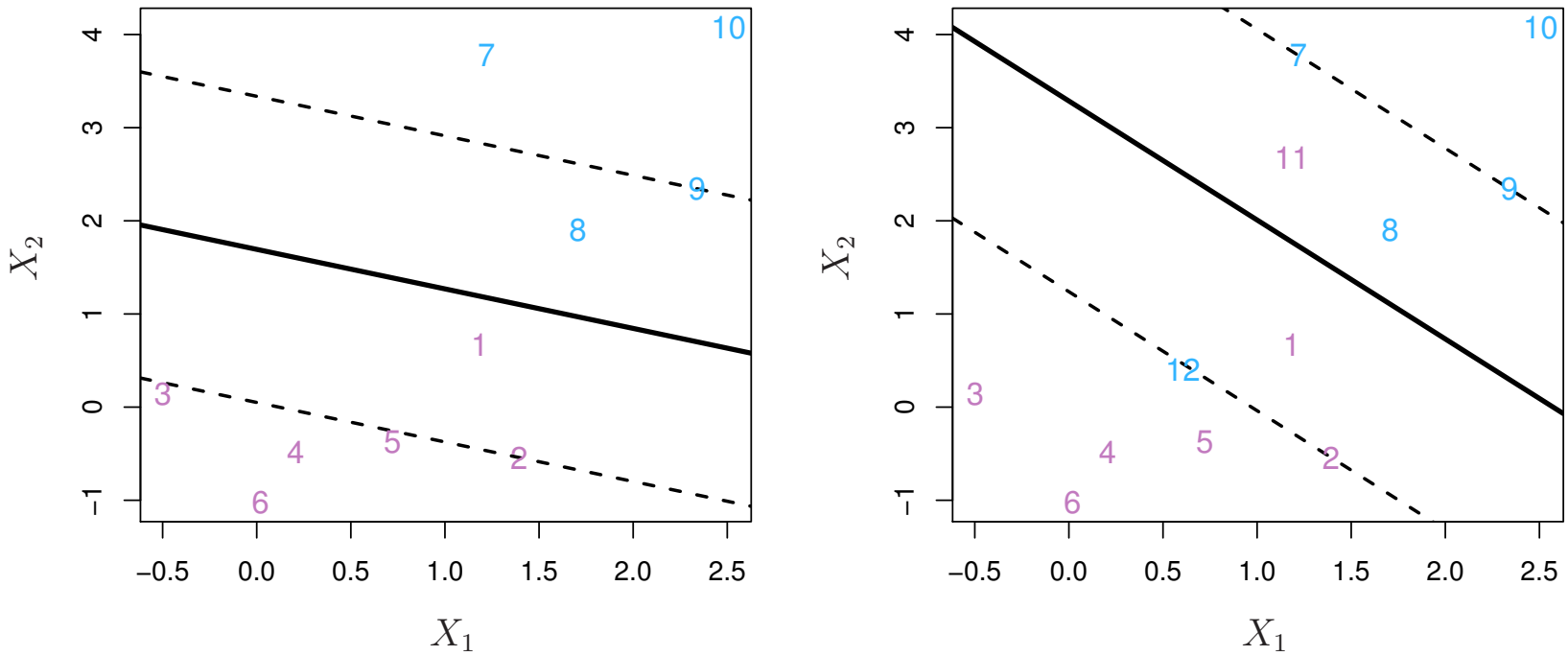


- Reasonable to accept hyperplane classifier which does **not** perfectly separate training data in the interest of greater robustness to individual observations and better classification of **most** training observations.

Support Vector Classifiers

Support vector classifiers

- **Support vector classifier** (**soft margin classifier**): allow some observations to be on wrong side of margin or hyperplane. (Soft margin: can be violated by some training observations).
- Example: hyperplane (solid), margins (dashed) and observations.



Observations 11 (violet) and 12 (blue) are added in right panel.

- Support vector classifier solves following optimization problem:

$$\begin{aligned}
&\text{maximize } M \text{ as a function of } \beta_0, \beta_1, \ldots, \beta_p, M, \epsilon_1, \ldots, \epsilon_n \\
&\text{subject to} \\
&\sum_{j=1}^{p} \beta_j^2 = 1, && (9.2a) \\
&y_i(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p}) \geq M(1 - \epsilon_i), && (9.2b) \\
&\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C, \qquad i = 1, \ldots, n, && (9.2c)
\end{aligned}$$

  where $C \geq 0$ is a tuning parameter.
  $\epsilon_1, \ldots, \epsilon_n$ are known as **slack variables**, allowing some observations to be on the wrong side of the margin or hyperplane.
- Classify test observation $\boldsymbol{x}$ based on sign of $\hat{f}(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$.

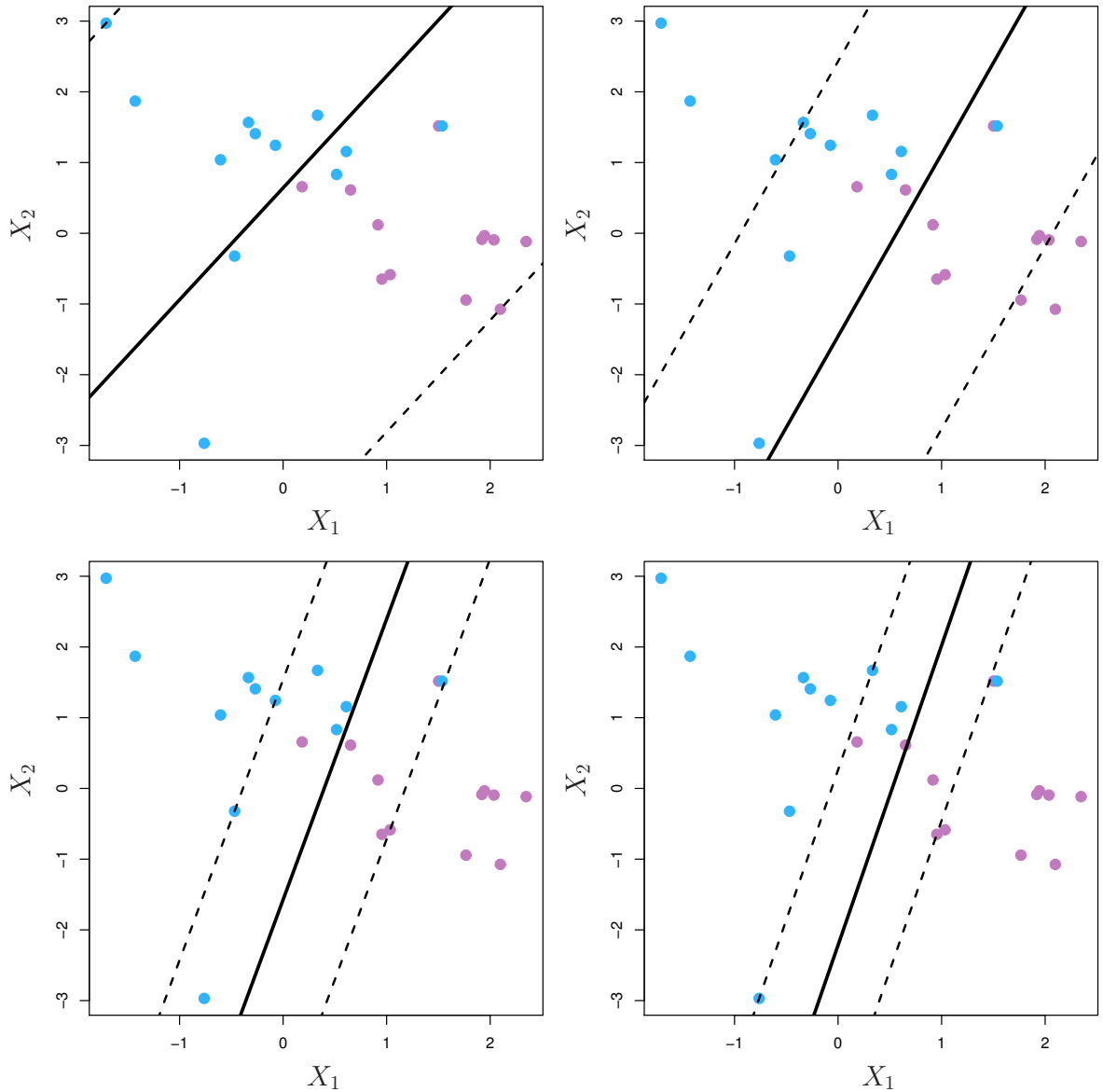
Support Vector Classifiers

Remarks on optimization problem

- Slack variable ϵ_i indicates location of $\boldsymbol{x}_i$ relative to margin:
  $\epsilon_i = 0$: $\boldsymbol{x}_i$ on correct side of margin,
  $\epsilon_i > 0$: $\boldsymbol{x}_i$ on wrong side of margin,
  $\epsilon_i > 1$: $\boldsymbol{x}_i$ on wrong side of hyperplane.
- C: bounds total permitted violation, budget for amount margin can be violated. $C = 0$ implies $\epsilon_i = 0$ for all i, i.e., recover maximal margin hyperplane optimization.
  C controls bias-variance trade-off.
- Example on next slide: Support vector classifier fit using 4 different values of C (value decreases from top left to bottom right).

Support Vector Classifiers

Example



Support Vector Classifiers

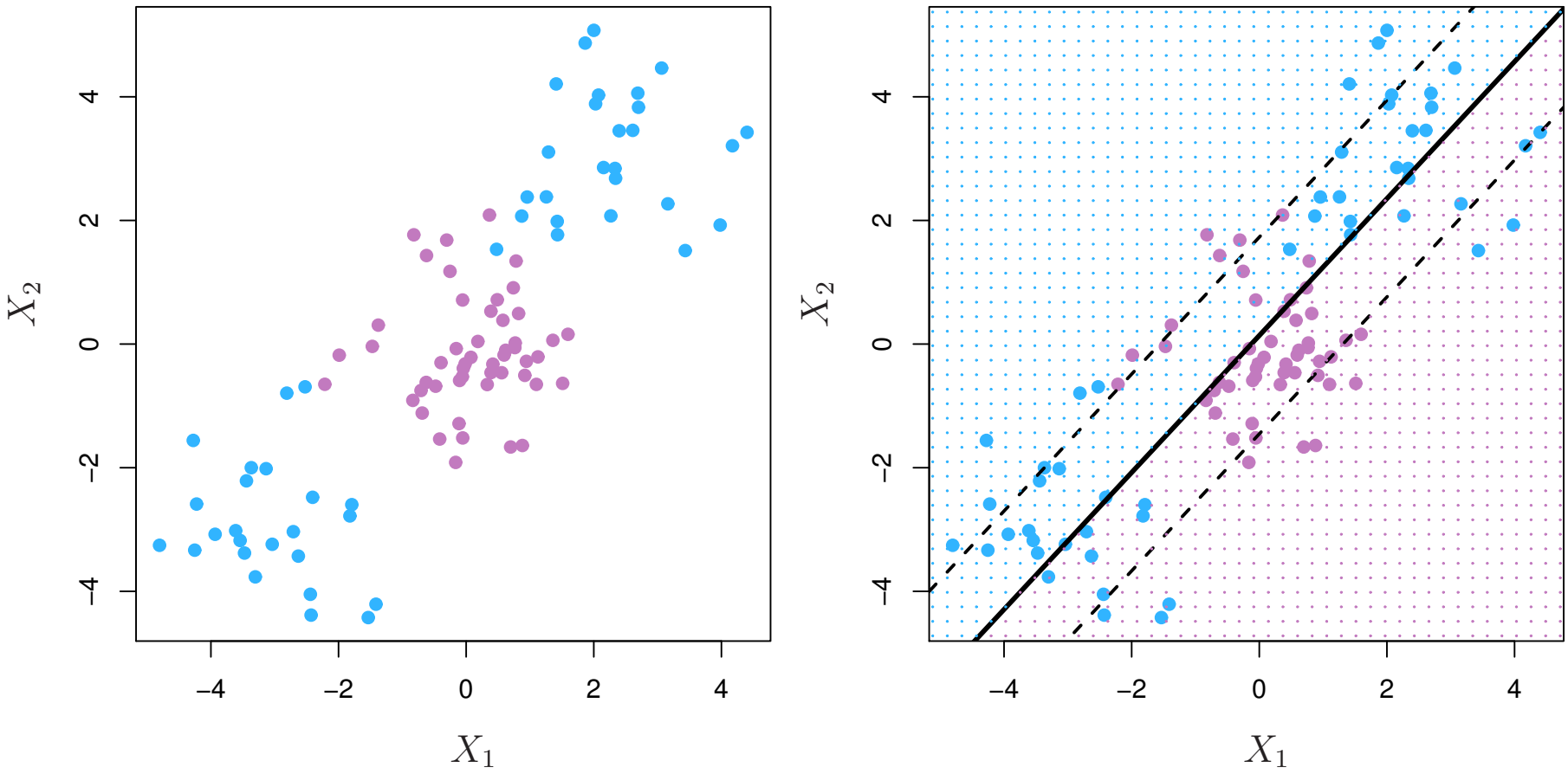
Support vectors

- Only observations on or violating margin will affect hyperplane/classifier.
- Observations on correct side orf margin have no effect on support vector classifier: changing their position without violating the margin would not change the classifier.
- The remaining observations (on or in violation of margin) are called **support vectors**.
- C large: margin wide, many observations violate margin, many support vectors, many observations involved determining hyperplane, classifier has low variance, potentially high bias (top left, pevious figure)
- C small: fewer support vectors, lower bias, higher variance (bottom right, pevious figure).
- Compare LDA classifier: depends on mean of **all** observations within each class as well as within-class covariance matrix computed using all observations.
- By contrast: logistic regression, unlike LDA, has low sensitivity to observations far from decision boundary. (Will see later: logistic regression and support vector classifier closely related.)

Contents

Support Vector Machines

Nonlinear decision boundaries



Some data sets require decision boundaries which are curved, hence a support vector classifier will perform poorly.

Support Vector Machines

Nonlinear decision boundaries

- In Chapter 7: extended model flexibility by adding nonlinear terms.
- Analogously: in addition to features $X_1, \ldots, X_p$, add $X_1^2, \ldots X_p^2$. Then (9.2) becomes

maximize M as a function of $\beta_0, \beta_{1,1}, \beta_{1,2} \ldots, \beta_{p,1}, \beta_{p,2}, M, \epsilon_1, \ldots, \epsilon_n$

subject to

$$\sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{j,k}^2 = 1, \tag{9.3a}$$

$$y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_{j,1} x_{i,j} + \sum_{j=1}^{p} \beta_{j,2} x_{i,j}^2 \right) \geq M(1 - \epsilon_i), \tag{9.3b}$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C, \qquad i = 1, \ldots, n, \tag{9.3c}$$

- Could also add interaction terms $X_j X_k$ in this way.

Support Vector Machines

New idea

- Enlarging the feature space in this way quickly makes the computations unmanageable.
- The key computational ingredient involves using **kernels**.
- Details of solving optimization problem (9.2) for support vector classifier involve **inner product** of observations

$$\langle \boldsymbol{x}_i, \boldsymbol{x}_{i'} \rangle = \sum_{j=1}^{p} x_{i,j} x_{i',j}$$

rather than observations themselves.

- Can show that linear support vector classifier has representation

$$\hat{f}(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle \tag{9.4}$$

in terms of n parameters $\{\alpha_i\}_{i=1}^{n}$ (one per training observation).

Support Vector Machines

New idea

- Estimating parameters $\beta_0, \alpha_1, \ldots, \alpha_n$ involves all $n(n-1)/2$ pairwise inner products of training observations.
- It turns out that in (9.4) α_i is only nonzero if $\boldsymbol{x}_i$ is a support point.
- Denoting the index set of the support points by $\mathscr{S}$, (9.4) becomes

$$\hat{f}(\boldsymbol{x}) = \beta_0 + \sum_{i \in \mathscr{S}} \alpha_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle. \tag{9.5}$$

- This typically involves far fewer points than (9.4).
- Expanding the inner products in (9.5) establishes the relationship between the parameters α_i and the original coefficients β_j.
- **Summary:** computation and evaluation of linear classifier rests on evaluating innter products of point in feature space.
- Replace inner products $\langle \boldsymbol{x}, \boldsymbol{x}_i \rangle$ with different function $K(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$ of $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$ referred to as a **kernel function** or simply **kernel**.

Support Vector Machines

Kernels

- Kernels quantify the degree of similarity or strength of relationship between two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$ of feature space.
- Could simply take inner product

$$K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \sum_{j=1}^{p} x_{i,j} x_{i',j},$$

(**linear kernel**), recovering support vector classifier. Here similarity of observations quantified using standard correlation.

- Alternatively,

$$K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \left(1 + \sum_{j=1}^{p} x_{i,j} x_{i',j}\right)^d$$

known as **polynomial kernel** of degree $d \in \mathbb{N}$. Leads to nonlinear decision boundary for support classifier, involves higher order terms in original features.

Support Vector Machines

Kernels

- Combination of support vector classifier with nonlinear kernel referred to as **support vector machine** (SVM).
- In this case the model has the form

$$\hat{f}(\boldsymbol{x}) = \beta_0 + \sum_{i \in \mathscr{S}} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i). \tag{9.6}$$

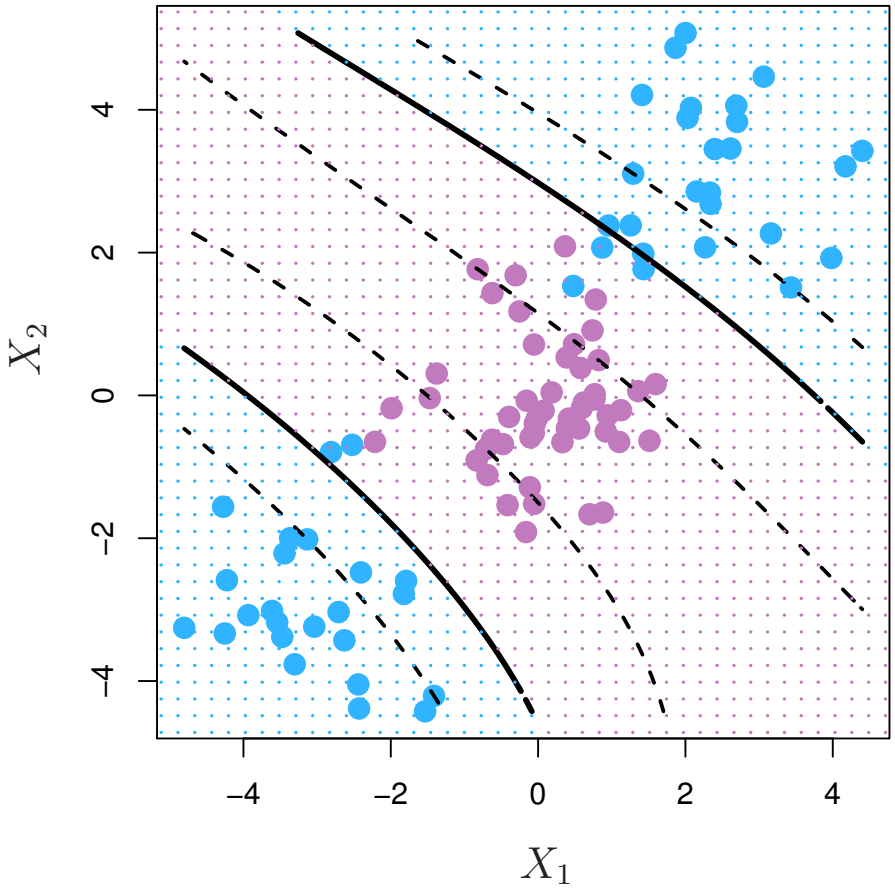
- Next picture: same data set as previous picture, SVM classifier using polynomial kernel (left panel). Right panel shows SVM using **radial kernel**

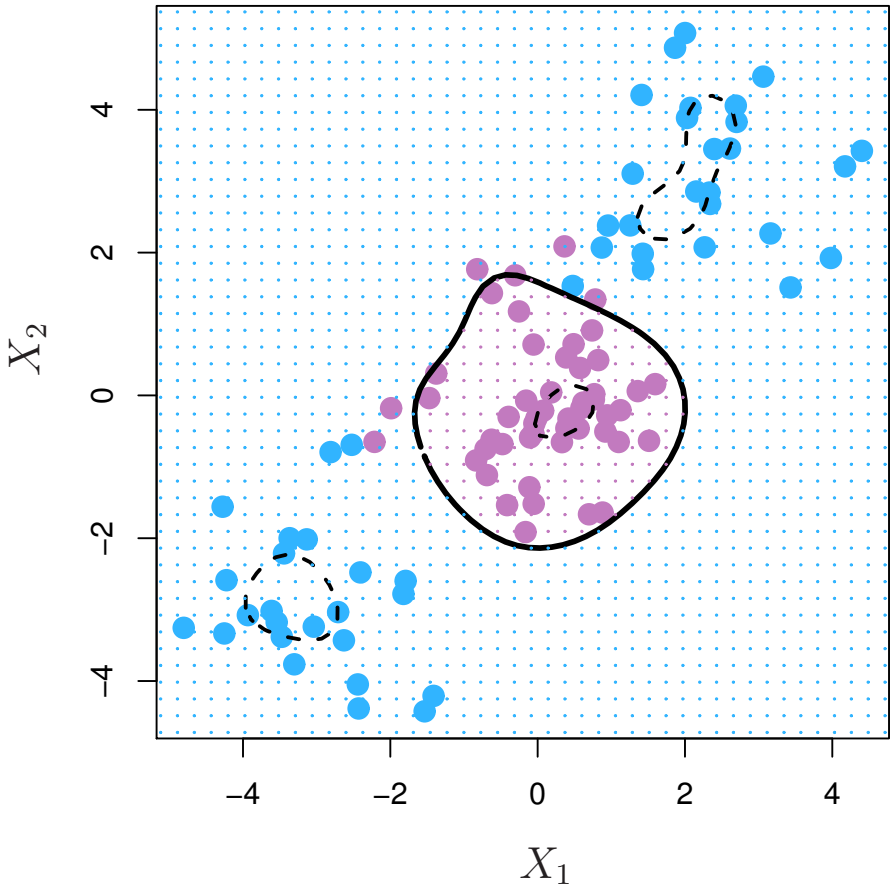
$$K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \exp\left(-\gamma \sum_{j=1}^{p} (x_{i,j} - x_{i',j})^2\right) = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|_2^2), \tag{9.7}$$

$\gamma > 0$ a parameter.

Support Vector Machines

Kernels





Support Vector Machines

Kernels

- If test observation $\boldsymbol{x} \in \mathbb{R}^p$ far from training observation $\boldsymbol{x}_i$ in the sense that $\|\boldsymbol{x} - \boldsymbol{x}_i\|_2$ large, then $K(\boldsymbol{x}, \boldsymbol{x}_i) = \exp(-\gamma\|\boldsymbol{x} - \boldsymbol{x}_i\|_2^2)$ will be very small.
- Therefore in (9.6) observation $\boldsymbol{x}_i$ will have almost no influence on $\hat{f}(\boldsymbol{x})$.
- Recall: class label prediction depends on sign of $\hat{f}(\boldsymbol{x})$.
- Therefore, observations far away from $\boldsymbol{x}$ have little influence in class prediction for $\boldsymbol{x}$. (Radial kernel has very **local** behavior.)
- Advantage of using kernel over simply enlarging feature space with nonlinear expressions in predictor variables: explicit enlarging of feature space avoided, leading to great computational savings (particularly if feature space infinite-dimensional).

Contents

Support Vector Machines

SVMs with more than two classes

- Up to now: maximal margin/support vector classifiers/machines for binary classification.
- Concept rests on separating hyperplane idea, does not readily generalize to $K > 2$ classes.
- Of many proposals for this problem, two most popular are **one-versus-one** and **one-versus-all** approaches.

Support Vector Machines

One-versus-one classification

- For $K > 2$ classes, there are $\binom{K}{2} = K(K-1)/2$ possible class pairs.
- For each pair, construct separate SVM to compare just these two classes, e.g., for pair (k_1, k_2), code k_1 as $+1$ and k_2 as -1.
- To each test observation, apply all $K(K-1)/2$ SVMs, record how often observation assigned to each class.
- Finally, classify observation as belonging to class to which it was assigned most often.

Support Vector Machines

One-versus-all classification

- For $K > 2$ classes, fit K SVMs, each comparing one class with remaining $K - 1$ classes.
- Denote by $\beta_{0,k}, \beta_{1,k}, \ldots, \beta_{p,k}$ the coefficients resulting from fitting SVM to determine membership in k-th class, coded as $+1$, against membership in one of the $K - 1$ remaining classes, coded as -1.
- For test observation $\boldsymbol{x} = (x_1, \ldots, x_p)^\top$, assign to class k for which

$$\beta_0 + \beta_{1,k} x_1 + \cdots + \beta_{p,k} x_p$$

is largest, indicating high level of confidence that $\boldsymbol{x}$ belongs to class k.

Contents

Support Vector Machines

Relationship to logistic regression

- SVMs introduced in mid-1990s.
- Immediately successful due to performance, marketing, novelty of approach.
- Approach notably different from previously established classification methods such as logistic regression, LDA.
- Introduced kernel idea for expanding feature space and allowing nonlinear decision boundaries.
- In the meantime, deep connections between SVMs and classical statistical learning methods have been discovered.

Support Vector Machines

Relationship to logistic regression

- Can show: optimization problem (9.3) for fitting support vector classifier $f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$ can be reformulated as

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\arg\min} \left\{ \sum_{i=1}^{n} \max\{0, 1 - y_i f(x_i)\} + \lambda \sum_{j=1}^{p} \beta_j^2 \} \right\}, \tag{9.8}$$

  with $\lambda \geq 0$ a tuning parameter.
- λ large: $\beta_q 1, \ldots \beta_p$ small, more margin violations tolerated, low variance, high bias.
- λ small: fewer violations tolerated high variance, low bias.
  Corresponds to low value of C in (9.3c).
- Term $\lambda \sum_{j=1}^{p} \beta_j^2$ is exactly the ridge regression penalty term, provides regularization, controls bias-variance trade-off for support vector classifier.

Support Vector Machines

Relationship to logistic regression

- Formulation (9.8) has form "Loss + Penalty" (data misfit + regularization)

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\arg\min}\ \left(L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta})\right) \qquad \begin{array}{l} L: \text{loss function} \\ P: \text{ penalty function} \\ (\boldsymbol{x},\boldsymbol{y}): \textit{data}. \end{array} \tag{9.9}$$

  Loss function measures degree to which model fits data.
  Penalty function weighted by regularization parameter λ.

- Ridge regression and the lasso both employ loss function

$$L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\beta}) = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j}\right)^2,$$

  with $P(\boldsymbol{\beta}) = \sum_{j=1}^{p}\beta_j^2$ for ridge regression and $\sum_{j=1}^{p}|\beta_j|$ for the lasso.

Support Vector Machines

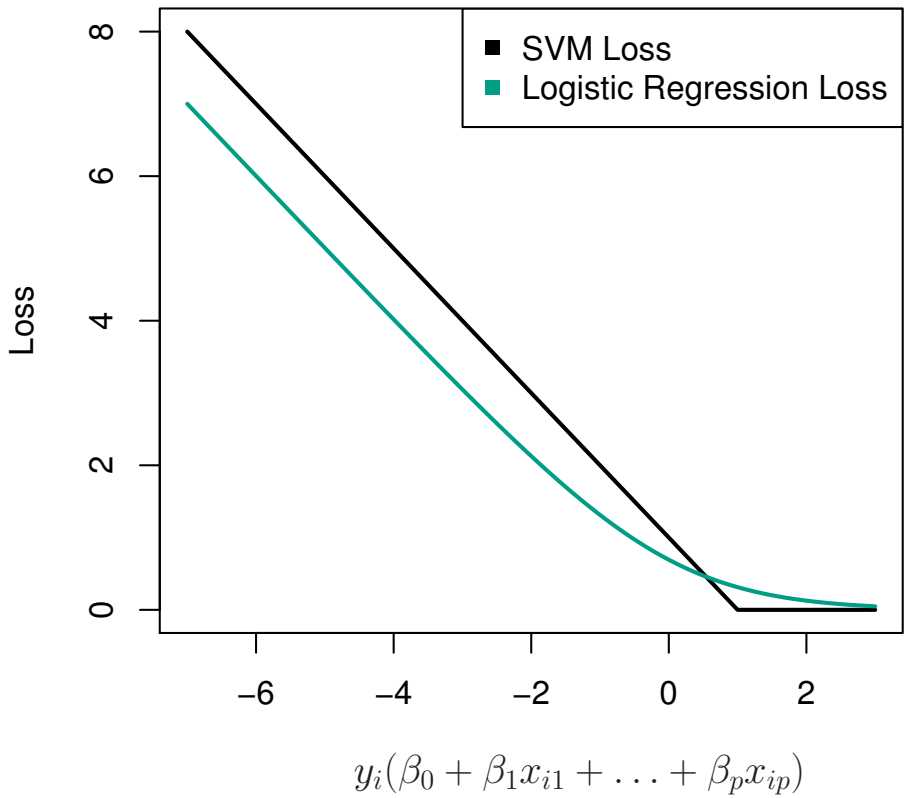
Relationship to logistic regression

- In (9.9), loss function has form

$$L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta}) = \max\{0, 1 - y_i(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p}\}$$

  known as **hinge loss**.

- Hinge loss function closely related to logistic regression loss function.



Support Vector Machines

Relationship to logistic regression

- In support classifier, only support vectors determine classifier, not observations on correct side of margin.
- Reason: loss function zero whenever $y_i(\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p}) \geq 1$. Here margin cooresponds to value 1 and $\sum \beta_j^2$ determines margin width.
- Logistic regression loss function dies not vanish anywhere, but very small away from decision boundary.
- SVM's behave better for well-separated classes, logistic regression preferred when more overlap present.
- Importance of parameter C not initially realized.
- Nonlinear kernels possible also in logistic regression, but not as common in practice.
- Extension of SVM to regression: **support vector regression**. Uses different loss function than LS regression, where only residuals above positive threshold contribute (extension of margin concept from classification to regression).

Contents

Unsupervised Learning

Introduction

- **Supervised learning**: n observations $\{(\boldsymbol{x}_i, y_i)_{i=1}^n\}$, each consisting of feature vector $\boldsymbol{x}_i \in \mathbb{R}^p$ and a response observation y_i.
- Construct prediction model $\hat{f}$ such that $y_i \approx \hat{f}(\boldsymbol{x}_i)$ in order to predict $y = \hat{f}(\boldsymbol{x})$ for values $\boldsymbol{x}$ not among data set.
- **Unsupervised learning:** only feature observations available, no response data.
- Prediction not possible.
- Instead: statistical techniques for "discovering interesting things" about observations $\{\boldsymbol{x}_i\}_{i=1}^n$.
- Informative visualization of the data.
- Indentification of subgroups in the data/variables.
- Here: **principal components analysis** (PCA) and **clustering**.

Unsupervised Learning

Challenges

- For supervised learning tasks, e.g., binary classification, large selection of well developed algorithms (logistic regression, LDA, classification trees, SVMs) as well as assessment techniques (CV, validation set, ...).
- Unsupervised learning more subjective.
- No clear goal of analysis (such as response prediction).
- Often performed as part of **exploratory data analysis**.
- Results harder to assess (by very nature).
- Examples:
  - finding patterns in gene expression data for cancer patients;
  - identifying subgroups of customers of online shopping platform which display similar behavior/interest;
  - determining which content a search engine should display to which individuals.

Contents

Unsupervised Learning

Principal components analysis

- Many correlated feature/predictor variables $X_1, \ldots, X_p$.
- Form new predictor variables Z_m (components) as linear combinations of original variables.
- Construct Z_m to be uncorrelated, ordered by decreasing variance.
- **Ideal situation:** first few $M < p$ components (principal components) explain large part of total variance of original variables. In this case data set well explained by restriction to principal components.
- Have used this idea for principal components regression (Chapter 6). There, used principal components as new (fewer) predictor variables.
- PCA: process by which principal components derived; also a technique for data visualization.
- Unsupervised, since applies only to feature/predictor variables.

- To visualize p-variate data using bivariate scatterplots, $\binom{p}{2} = p(p-1)/2$ pairs to examine.
- Besides effort involved, individual scatterplots not necessarily that informative, containing only small fraction of information carried by complete data.
- Ideal: find low (1, 2 or 3)-dimensional representation of data containing all (most) relevant information.
- **First principal component**: linear combination

$$Z_1 = \phi_{1,1}X_1 + \cdots + \phi_{p,1}X_p, \qquad \sum_{j=1}^{p} \phi_{j,1}^2 = 1, \tag{10.1}$$

of original feature variables X_j with normalized coefficients ("**loadings**") with maximal variance.
Loading vector $\boldsymbol{\phi}_1 := (\phi_{1,1}, \ldots, \phi_{p,1})^\top$.

- Given data set

$$\boldsymbol{X} \in \mathbb{R}^{n\times p}, \quad \text{i.e., } n \text{ samples of } p \text{ features } X_1, \ldots, X_p,$$

- Each column $\boldsymbol{x}_j = (x_{1,j}, \ldots, x_{n,j})^\top \in \mathbb{R}^n$, $j = 1, \ldots, p$, contains n samples (observations) of j-th feature.
- Each row $\tilde{\boldsymbol{x}}_i^\top = (x_{i,1}, \ldots, x_{i,p}) \in \mathbb{R}^p$, $i = 1, \ldots, n$, contains one sample of p features.
- Here information synonymous with variance, hence assume centered columns, i.e.,

$$\boldsymbol{e}^\top \boldsymbol{x}_j = 0, \quad j = 1, \ldots, p, \qquad \boldsymbol{e} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n,$$

hence sample mean of each column is zero.

- Loadings $\{\phi_{j,1}\}_{j=1}^p$ for first principal component determined as (normalized) coefficients in linear combination

$$z_1 = \phi_{1,1}\boldsymbol{x}_1 + \cdots + \phi_{p,1}\boldsymbol{x}_p = \boldsymbol{X}\boldsymbol{\phi}_1$$

  such that z_1 has largest sample variance (mean remains zero).
- In other words, loadings $\{\phi_{j,1}\}_{j=1}^p$ solve optimization problem

$$\max\left\{\frac{1}{n}\sum_{i=1}^n\left(\sum_{j=1}^p \phi_{j,1}x_{i,j}\right)^2 : \sum_{j=1}^p \phi_{j,1}^2 = 1\right\} \tag{10.2}$$

- In other words, loading vector $\boldsymbol{\phi}_1$ solves optimization problem

$$\max_{\|\boldsymbol{\phi}\|_2=1} \|\boldsymbol{X}\boldsymbol{\phi}\|_2^2 = \max_{\|\boldsymbol{\phi}\|_2=1} \boldsymbol{\phi}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\phi}.$$

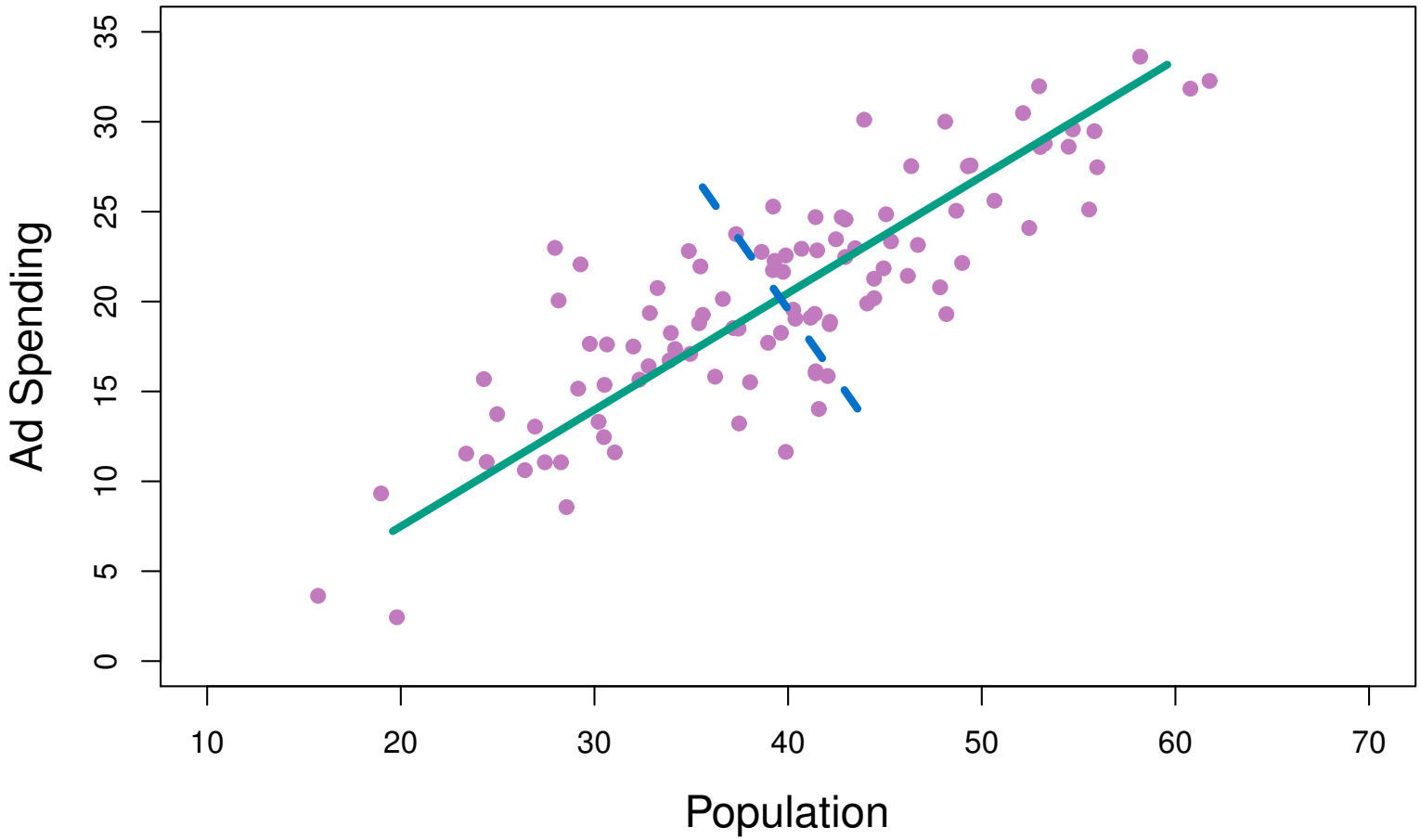
- In other words (Courant-Fischer max-min principle), $\boldsymbol{\phi}_1$ is a normalized eigenvector associated with largest eigenvalue of $\boldsymbol{X}^\top\boldsymbol{X}$.

- Equivalent characterization: $\boldsymbol{\phi}_1$ is a right singular vector associated with the largest singular values of (centered) data matrix $\boldsymbol{X}$.
- Components $z_{1,1}, \ldots, z_{n,1}$ of $\boldsymbol{z}_1$ referred to as **scores** of first principal component.
- **Geometric interpretation:** loading vector $\boldsymbol{\phi}_1$ defines direction in feature space along which data varies the most.
  "Projection of data points $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n$ (rows of $\boldsymbol{X}$) in this direction yield principal component scores $\boldsymbol{z}_1$."
  This is simply the dual interpretation of the matrix-vector product $\boldsymbol{z}_1 = \boldsymbol{X}\boldsymbol{\phi}_1$: rather than as a linear combination of the columns $\{\boldsymbol{x}_j\}_{j=1}^p \subset \mathbb{R}^n$ of $\boldsymbol{X}$, it is viewed as the vector of inner products of $\boldsymbol{\phi}_1$ with the rows $\{\tilde{\boldsymbol{x}}_i\}_{i=1}^n \subset \mathbb{R}^{1\times p}$ of $\boldsymbol{X}$:

$$\boldsymbol{z}_1 = \boldsymbol{X}\boldsymbol{\phi}_1 = \begin{bmatrix} \tilde{\boldsymbol{x}}_1^\top \boldsymbol{\phi}_1 \\ \vdots \\ \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\phi}_1 \end{bmatrix}.$$

Unsupervised Learning

Computing the first principal component



First principal component loading vector in advertising data set (green). Here $p = 2$ and observation data can be viewed along with principal component vectors.

- Second principal component Z_2: linear combination of $X_1, \ldots, X_p$ with largest variance subject to condition that it is uncorrelated with Z_1.
- Scores

$$\boldsymbol{z}_2 = \phi_{1,2}\boldsymbol{x}_1 + \cdots + \phi_{p,2}\boldsymbol{x}_p = \boldsymbol{X}\boldsymbol{\phi}_2$$

  with second principal components loading vector $\boldsymbol{\phi}_2 = (\phi_{1,2}, \ldots, \phi_{p,2})^\top$.
- Uncorrelatedness equivalent with orthogonality in Euclidean inner product.
- Hence $\boldsymbol{\phi}_2$ is normalized eigenvector associated with second-largest eingenvalue of $\boldsymbol{X}^\top\boldsymbol{X}$, or normalized right singular vector associated with second-largest singular value of $\boldsymbol{X}$.
- Previous figure: $p = 2$, only one possibility for $\boldsymbol{\phi}_2$ (dashed blue line).
- Remaining components Z_m defined analogously: linear combination of $X_1, \ldots, X_p$ with maximal variance uncorrelated with $Z_1, \ldots, Z_{m-1}$ (Euclidean orthogonality of recombined sample vectors).
- There are at most $\min\{n-1, p\}$ principal components.

- Denoting the SVD of the centered data matrix as $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ gives $\boldsymbol{X}^\top\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\boldsymbol{V}^\top$.
- The eigenvalues of $\boldsymbol{X}^\top\boldsymbol{X}$ in descending order are displayed on the diagonal of $\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} = \mathsf{diag}(\sigma_1^2, \ldots, \sigma_p^2)$.
- The total variance in the data represented by $\boldsymbol{X}$ is given by $\|\boldsymbol{X}\|_F^2 = \sigma_1^2 + \cdots + \sigma_p^2$.
- The principal component loading vectors $\{\boldsymbol{\phi}_j\}_{j=1}^p$ are given by the normalized eigenvectors of $\boldsymbol{X}^\top\boldsymbol{X}$ or, equivalently, the right singular vectors of $\boldsymbol{X}$, i.e.,

$$\boldsymbol{\phi}_j = \boldsymbol{v}_j, \qquad j = 1, \ldots, \min\{n-1, p\}.$$

- For the scores z_m, we have

$$z_m = \boldsymbol{X}\boldsymbol{\phi}_m = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\boldsymbol{v}_m = \sigma_m\boldsymbol{u}_m, \qquad m = 1, \ldots, \min\{n-1, p\}.$$

Unsupervised Learning

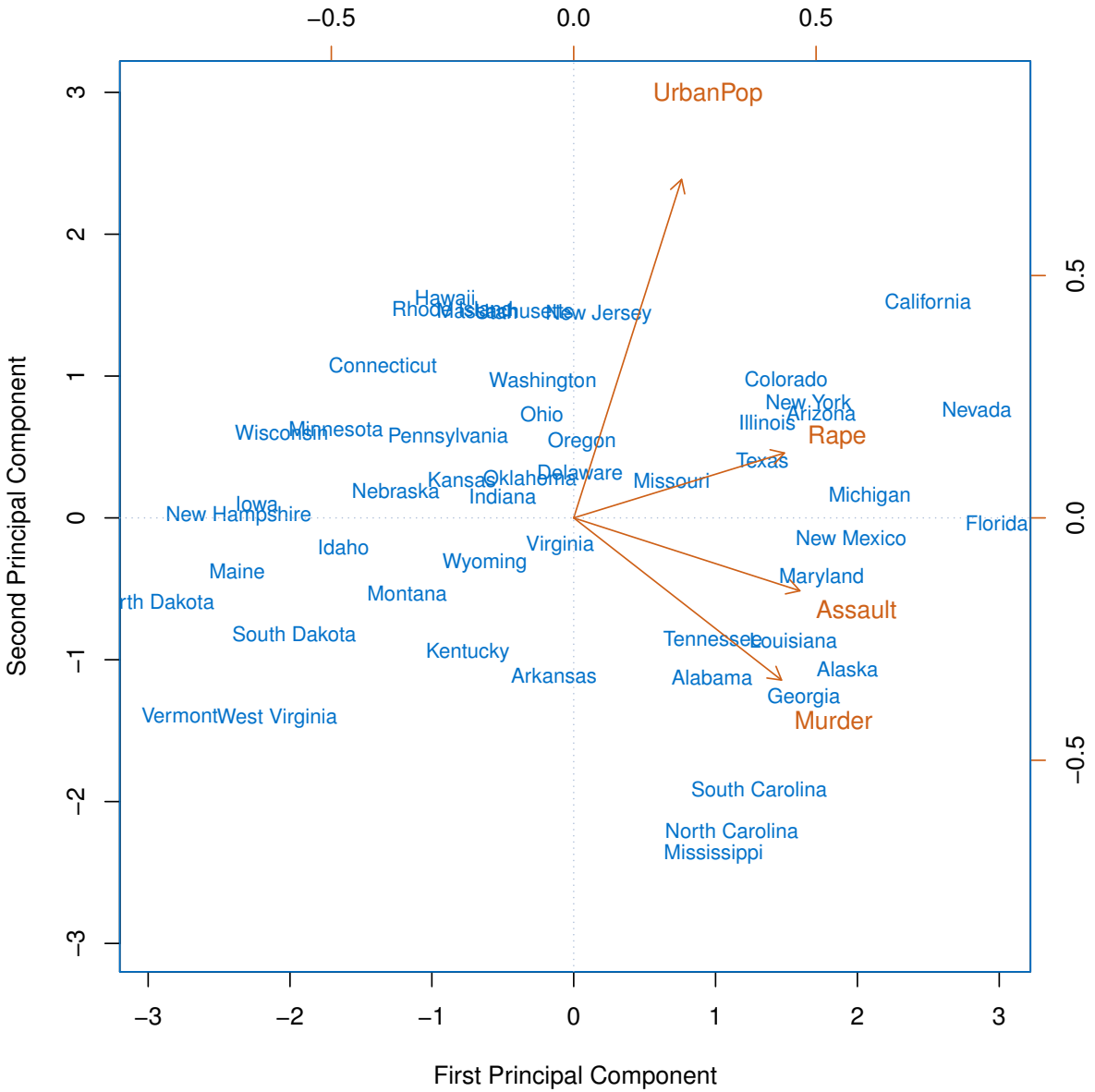
Example: `USArrests` data set

- `USArrests` data set: arrests per 100,000 residents of each of the 50 states of the USA for each of the crimes `Assault`, `Murder` and `Rape`.
- Also records `UrbanPop`, percentage of each state's population living in urban areas.
- Number of samples = length of PC score vector $n = 50$.
- Dimension of feature space = length of PC loading vectors $p = 4$.
- PCA performed after standardizing data matrix (column mean zero, standard deviation one).
- PC loading vectors

	PC1	PC2
`Murder`	0.5358995	-0.4181809
`Assault`	0.5831836	-0.1879856
`Rape`	0.5434321	0.1673186
`UrbanPop`	0.2781909	0.8728062

Unsupervised Learning

Example: `USArrests` data set



- **Biplot** of data in space of first two principal components.
- Blue state names: score in first 2 PC.
- Orange arrows: first two PC loading vectors (axes on right and top).
- Biplot: displays both PC scores and PC loadings.

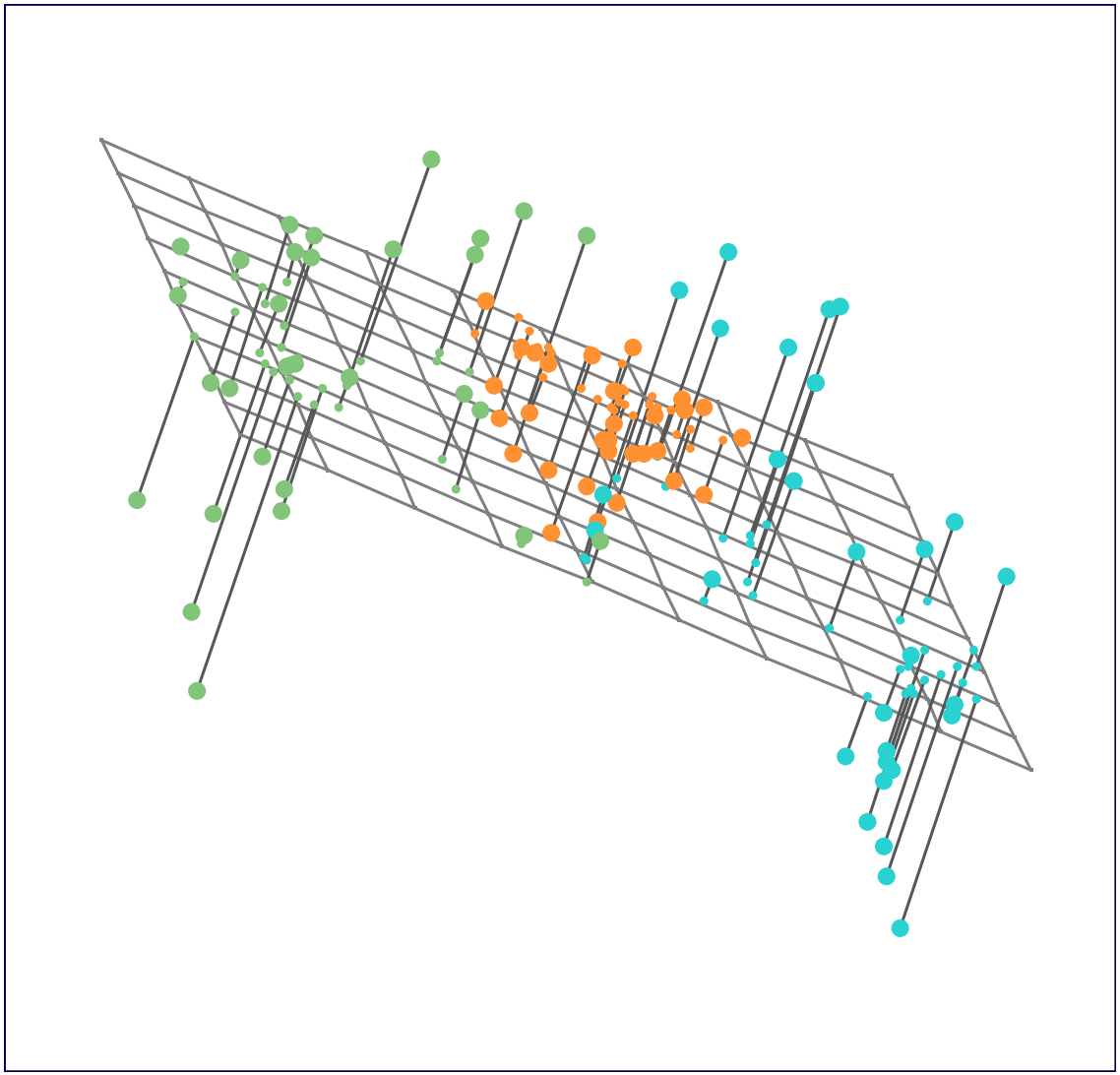
Unsupervised Learning

Example: `USArrests` data set: Interpretation of figure

- First loading vector places approximately equal weight on `Assault`, `Murder` and `Rape`, much less weight on `UrbanPop`.
- Hence first PC roughly corresponds to measure of overall rate of serious violent crime.
- Second loading vector has more weight on `UrbanPop`, much less on remaining three features, hence roughly corresponds to level of urbanization of each state.
- Overall, crime-related variables close to each other in space spanned by first two PC, `UrbanPop` far from these: indicates crime-related variables highly correlated, weakly correlated with `UrbanPop`.
- State differences in first PC: states with high score in first component tend to have high crime rates (e.g. California, Nevada, Florida); those with negative first PC scores tend to have low crime rates (e.g. North Dakota).
- State differences in 2nd PC: High score in 2nd PC (e.g. California) indicates high level uf urbanisation, low score low level (e.g. Mississippi).
- States close to origin?

Unsupervised Learning

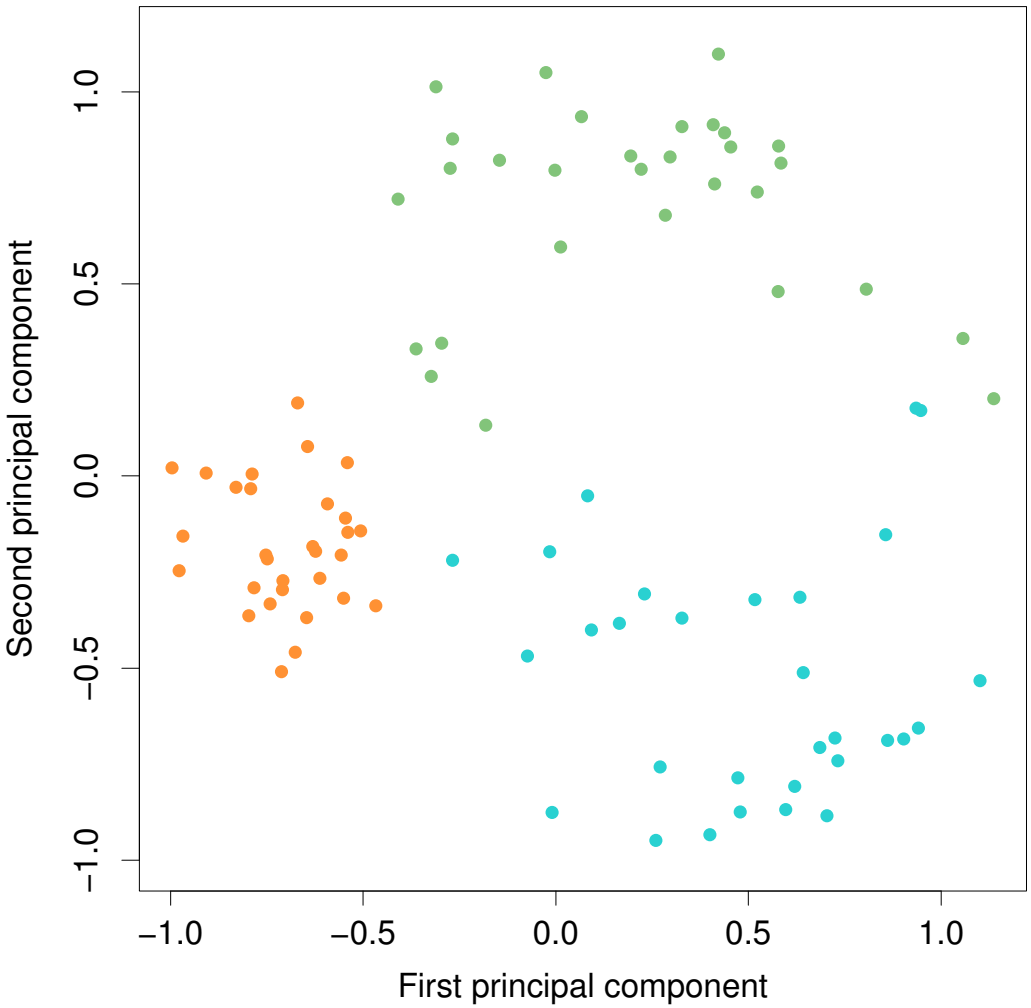
PCA: another interpretation



- First two PC loading vectors of a 3D data set along with observations.
- Span a plane along which observations have highest variance.
- Alternative interpretation: PC provide low-dimensional surfaces that are **closest** to the observations.

Unsupervised Learning

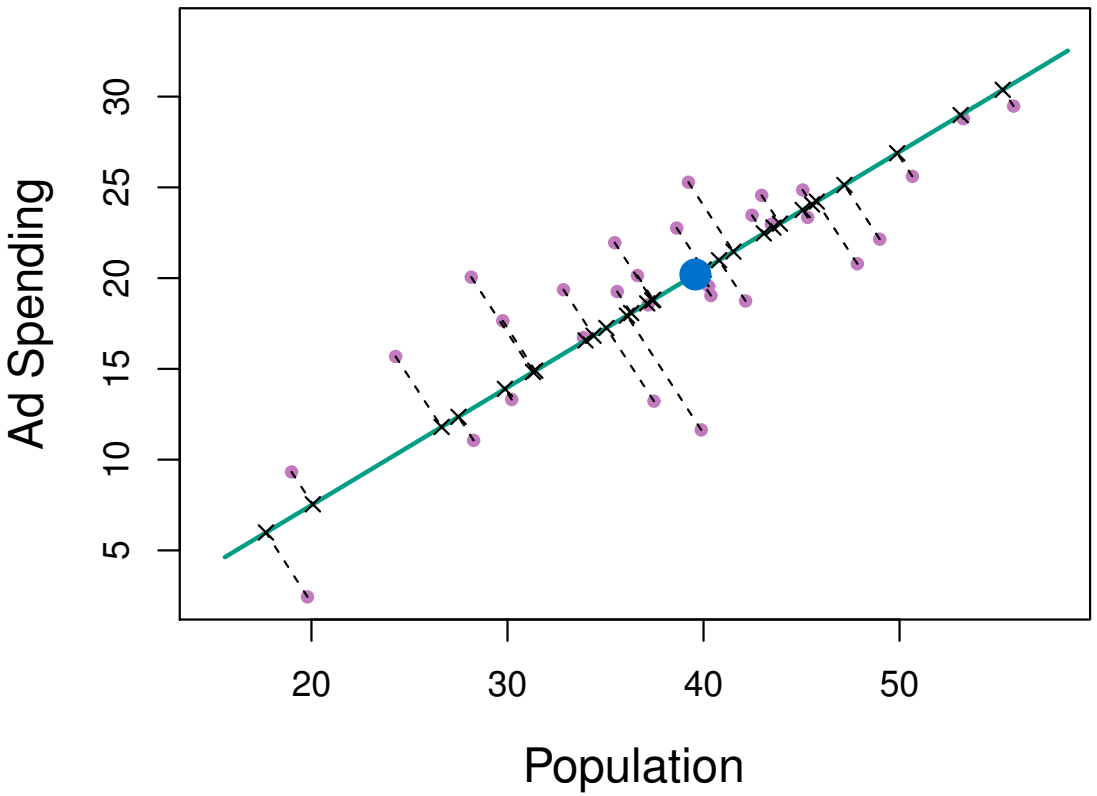
PCA: another interpretation



- First two PC loading vectors of a 3D data set along with observations.
- Span a plane along which observations have highest variance.
- Alternative interpretation: PC provide low-dimensional surfaces that are **closest** to the observations.
- Projection of observations to closest plane: variance is maximized.

Unsupervised Learning

PCA: another interpretation



- Example from Chapter 6 (ad spending vs. population)
- 1st PC loading vector: line in $\mathbb{R}^p$ closest to observations (in Euclidean distance).
- Dashed lines: distance between each observation and first PC loading vector.
- In this sense: good summary of the data.

Summary: the first M principal components and associated score vectors together yield a best approximation of the observational data:

$$x_{i,j} \approx \sum_{m=1}^{M} z_{i,m}\,\phi_{j,m}. \tag{10.3}$$

Explanation: writing all $n \times p$ equations (10.3) in matrix form yields

$$\boldsymbol{X} \approx \sum_{m=1}^{M} \boldsymbol{z}_m \boldsymbol{\phi}_m^\top = \sum_{m=1}^{M} \sigma_m \boldsymbol{u}_m \boldsymbol{v}_m^\top,$$

which is simply the singular value expansion of $\boldsymbol{X}$ truncated after m terms. Recalling the best approximation property of the truncated SVD in the spectral and Frobenius norms explains the nearness of the expression (10.3) to the data.

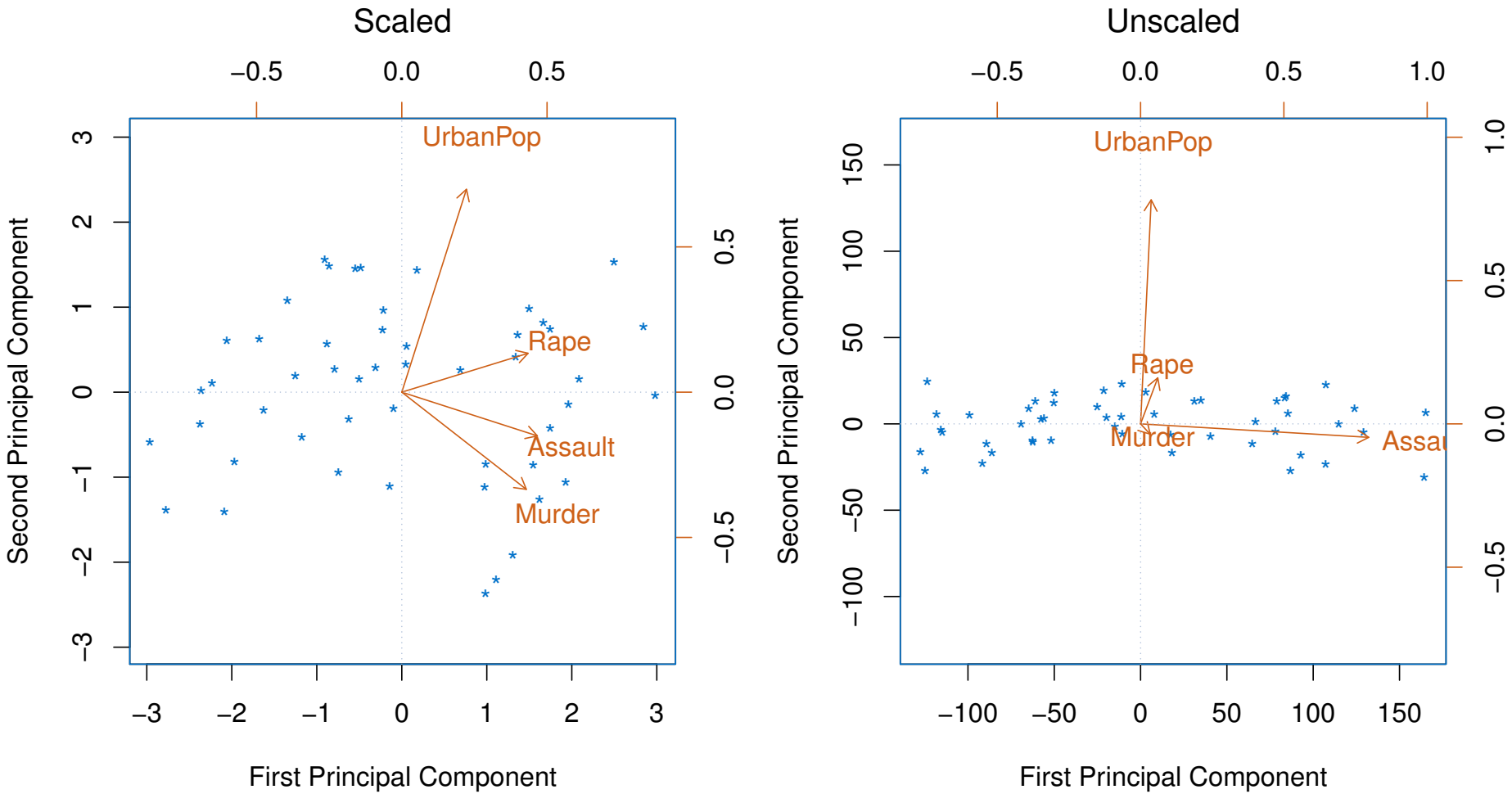
Unsupervised Learning

PCA: scaling

- Data matrix centered before applying PCA.
- Individual scaling of the predictor variables (columns) will affect the outcome of PCA.
- Contrast with linear regression, where rescaling of a variable compensated by associated coefficient.
- In `USArrests` example, each variable was rescaled to have standard deviation one.
- Reason: variables have different units (`Murder`, `Rape`, and `Assault` in # occurrences / 100,000 people, `UrbanPop` in percentage living in urban areas.
  Also: variances 18.97, 87.73, 6945.16 and 209.5, respectively, display large variation.
  Hence without scaling, first PC loading vector will have very large weight on `Assault`.
- Scaling is recommended, but doing so should be deliberate.

Unsupervised Learning

PCA: scaling



Biplots for PCA applied to `USArrests` data set.

Left: all variables scaled to have standard deviation one.

Right: PCA performed on unscaled variables.

Unsupervised Learning

PCA: uniqueness

- Singular vectors, normalized eigenvectors unique up to sign.
  Hence same holds for principal components.
- Different software packages will yield same PC loading vectors up to sign.
- Sign flipping harmless, as PC represent directions in Euclidean space.
- Note that flipping sign in $\boldsymbol{\phi}_m$ in (10.3) will result in sign flip in $\boldsymbol{z}_m$, leaving product unchanged.

- How much information is lost by replacing original data with PC approximation (projecting observations on first $M < p$ principal componants)?
- More precisely: how much of the variance of the original data is missing in the PC approximation? What is the **portion of variance explained** (PVE)?
- Define total variance in (centered) $\boldsymbol{X}$ by

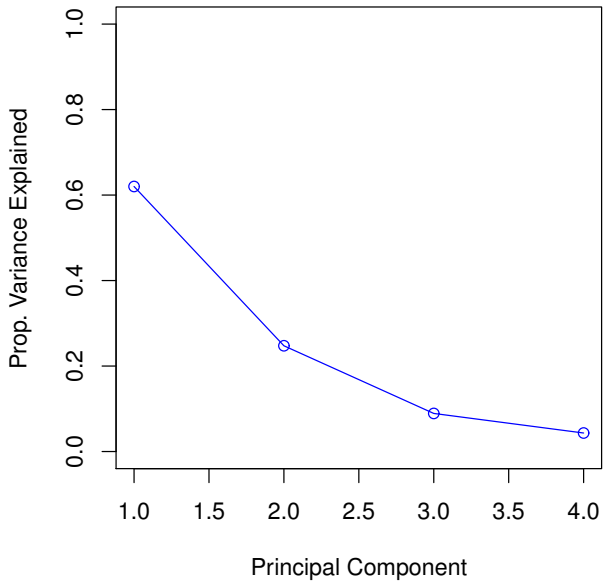
$$\sum_{j=1}^{p} \mathbf{Var}\, X_j := \sum_{j=1}^{n} \frac{1}{n} \sum_{i=1}^{n} x_{i,j}^2 = \frac{1}{n} \|\boldsymbol{X}\|_F^2.$$

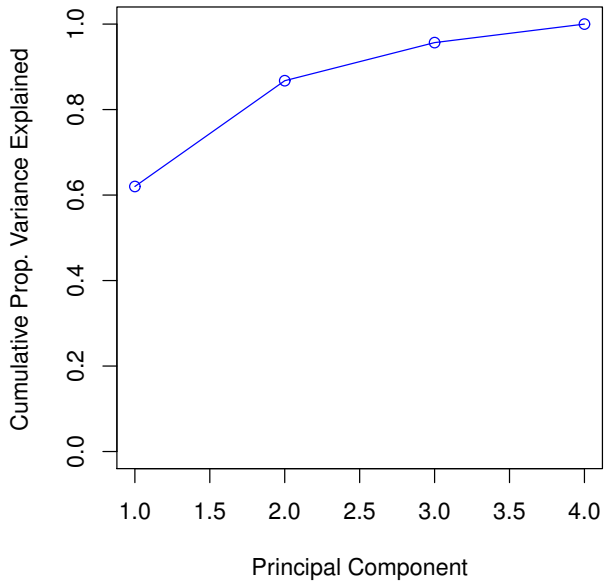
- Variance explained by m-th PC:

$$\frac{1}{n}\|\boldsymbol{z}_m\|_2^2 = \frac{1}{n}\sum_{i=1}^{n} z_{i,m}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p} \phi_{j,m} x_{i,j}\right)^2 = \|\boldsymbol{X}\boldsymbol{\phi}_m\|_2^2.$$

- Hence PVE of m-th PC given by $\|\boldsymbol{X}\boldsymbol{\phi}_m\|_2^2 / \|\boldsymbol{X}\|_F^2$.

- In `USArrests` data set: first PC explains 62% of total variance, 2nd explains 24.7%. Hence first two explain $\approx$ 87%, remaining two only 13%.
- Therefore, the biplot gives an accurate summary of the data (using just 2 dimensions).
- **Scree plots**: display PVE of each PC as well as cumulative PVE





Unsupervised Learning

PCA: sufficient number of principal components

- Can choose M between 1 and $\mathbf{min}\{p, n-1\}$.
- Ideal: smallest M conveying good understanding of data.
- Scree plot can provide guidance: fix M at "**elbows**", i.e., where proportion of variance explained has a noticeable drop.
  In previous example, elbow after $M = 2$ could be argued.
- Such visual analysis is heuristic, subjective and ad-hoc, but there is no general answer for determining how many PCs is enough (exploratory data analysis).
- In supervised learning, M is a tuning parameter, which can be determined by CV or similar validation technique.

Unsupervised Learning

PCA: further uses for PC

- Supervised learning: new features, smaller in number than original.
- **Low-rank** approximation of $\boldsymbol{X}$ obtained by truncating SVD after $M < p$ terms often better than full $\boldsymbol{X}$ due to noise reduction (e.g. latent semantic indexing).
- Signal of a data set often contained in first few PC, rest can be noise.

Contents

Unsupervised Learning

Clustering methods

- Broad set of techniques for finding **clusters** or **subgroups** in a data set.
- Partition data into distinct subsets of similar observations, where notion of similarity is problem-dependent.
- Unsupervised problem of finding structure in data set.
- Clustering and PCA seek to simplify data via small number of summaries, but via different mechanisms
  - PCA seeks low-dimensional representation of observations explaining good fraction of their variance.
  - Clustering seeks homogeneous subgroups among observations.
- **Example:** given marketing measurements (median household income, occupation, distance from nearest urban area, etc.) for large population, perform *market segmentation* to identify subgroups of people more receptive to a particular form of advertising or more likely to buy a particular product (cluster people in a data set)
- Here: 2 approaches, **K-means clustering**, **hierarchocal clustering**.

- Partition data into $K \in \mathbb{N}$ disjoint clusters.
- Upon fixing K, algorithm assigns each observation to one of K clusters.
- Let $\{C_k\}_{k=1}^K$ denote sets containing indices of n observations in cluster k such that

$$\bigcup_{k=1}^{K} C_k = \{1, \ldots, n\},$$

$$C_k \cap C_\ell = \emptyset \text{ for } k \neq \ell, \qquad k, \ell = 1, \ldots, K,$$

- A good clustering is one for which *within-cluster variation* is small.
- With $W(C_k)$ denoting a measure of amount by which observations in cluster k differ, K-means clustering tries to determine

$$\operatorname*{arg\,min}_{C_1,\ldots,C_K} \sum_{k=1}^{K} W(C_k).$$

- Common measure for in-cluster-variation: **squared Euclidean distance**

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{i,j} - x_{i',j})^2,$$

  $|C_k|$ denoting the cardinality of C_k.

- The cluster optimization problem thus becomes

$$\underset{C_1,\ldots,C_K}{\arg\min} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{i,j} - x_{i',j})^2 \right\} \tag{10.4}$$

- The number of possible clusterings of n observations in to K clusters grows [10] like K^n. There are, however, simple heuristics for finding good approximations of the solution.

[10] These are known as the *Stirling numbers of the second kind*, $S(n, K) \sim k^n/k!$ as $n \to \infty$.

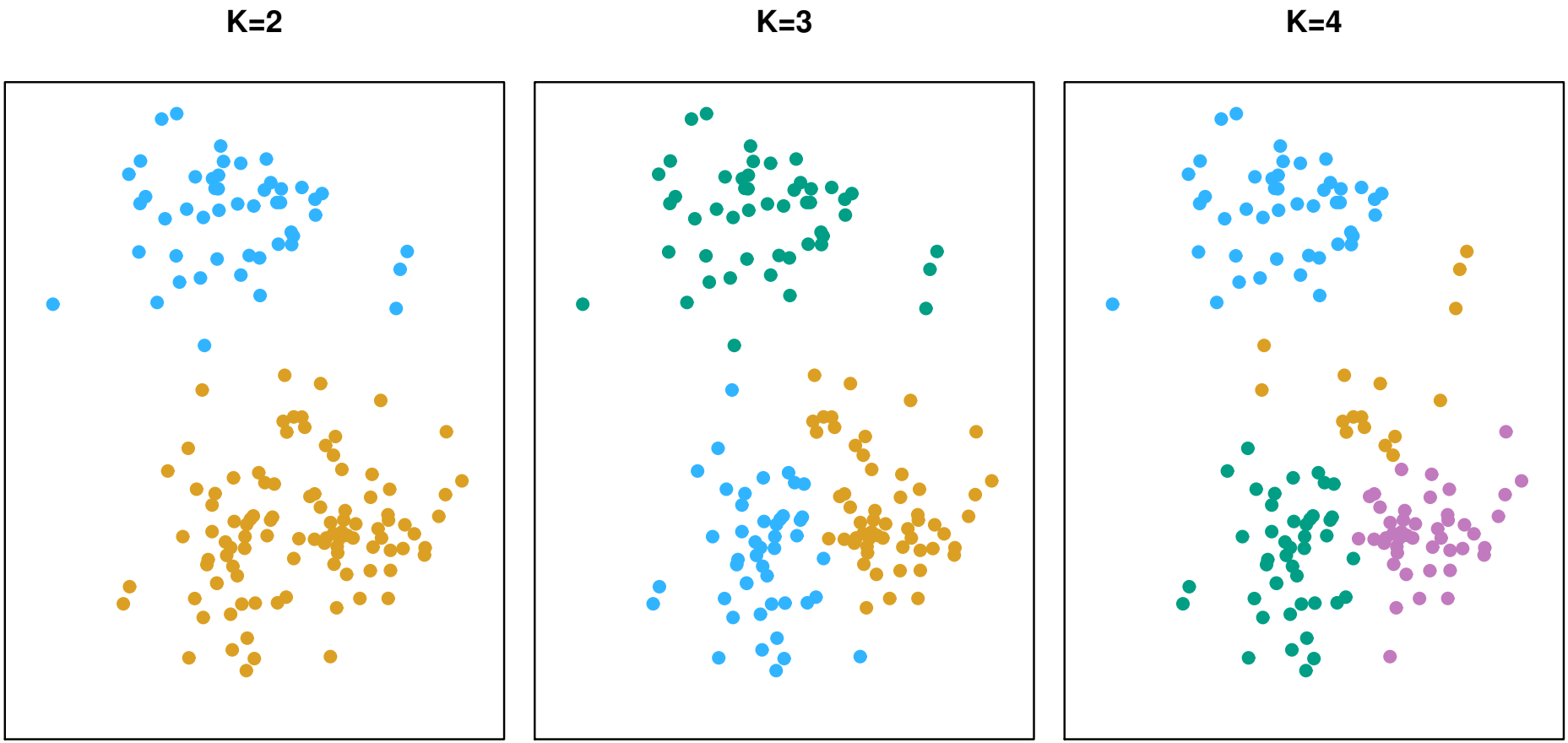
Algorithm 6: K-means clustering.

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:
   a. For each of the K clusters, compute the cluster **centroid**. The k-th cluster centroid is the vector of the p feature means for the observations in the kth cluster.
   b. Assign each observation to the cluster whose centroid is closest (where closest is defined by Euclidean distance).

The name of the algorithm is due to the computation of the centroids in step (2a), which are computed as the mean across all observations currently assigned to each cluster.

Unsupervised Learning

K-means clustering



Simulated data in 2D, $n = 150$. Results of applying K-means clustering with $K = 2, 3, 4$.

- Algorithm 6 guaranteed to decrease the value of the objective (10.4) in each step.
- Introducing the **cluster means**

$$\overline{x}_{k,j} := \frac{1}{|C_k|} \sum_{j=1}^{p} x_{i,j},$$

there holds

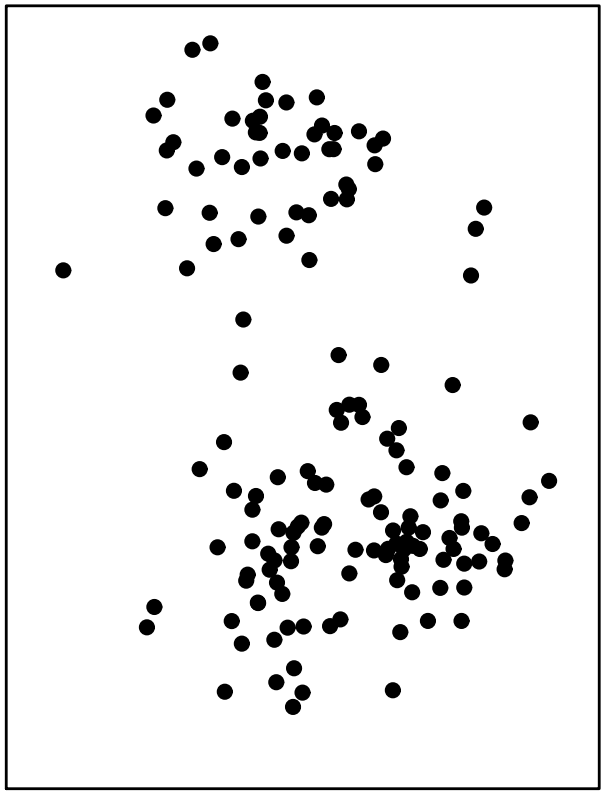
$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{i,j} - x_{i',j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{i,j} - \overline{x}_{k,j})^2.$$

- In Step (2a), cluster means for each feature are the constants that minimizing the sum-of-squares deviations.
- In step (2b), reallocating the observations within the clusters can only decrease the objective.
- As Algorithm 6 is run, objective improves until it no longer changes, ending in a **local optimum**.

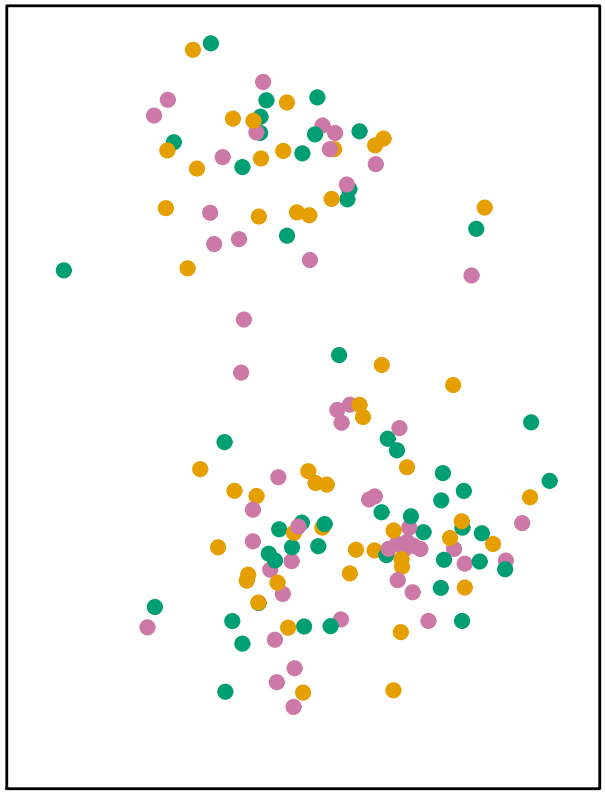
Unsupervised Learning

K-means clustering

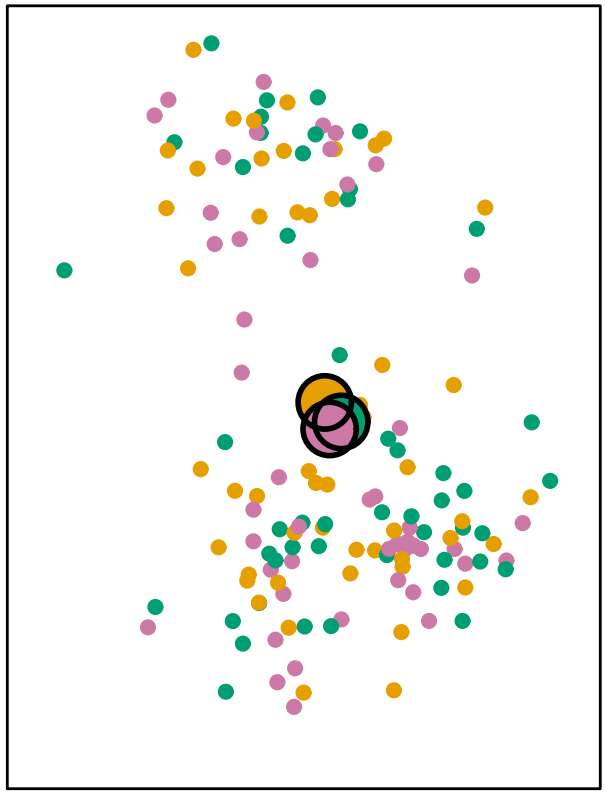
Data



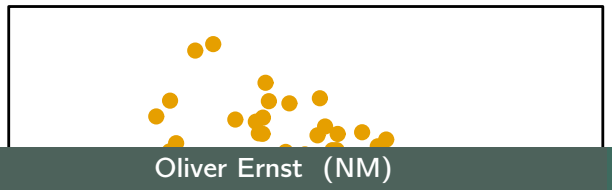
Step 1



Iteration 1, Step 2a



Iteration 1, Step 2b



Iteration 2, Step 2a

Final Results

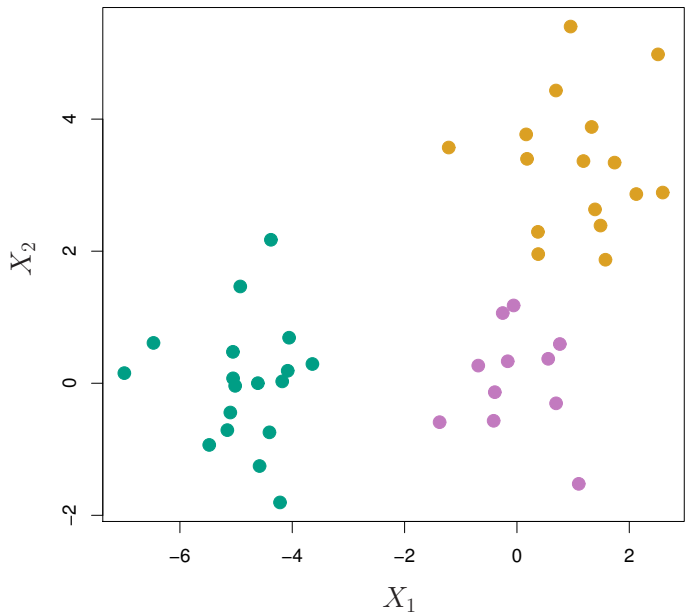
Unsupervised Learning

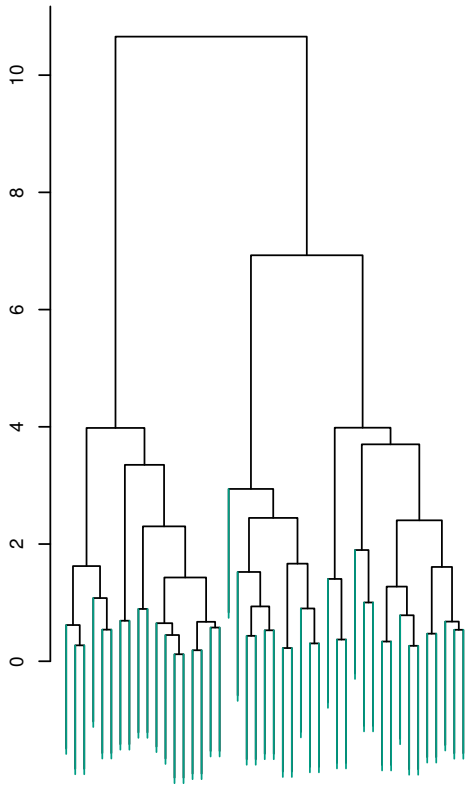
K-means clustering, dealing with local minima

Since result of K-means typically only local minimum, advisable to run multiple times using different random initial clusterings and pick the outcome with smallest objective.

Here K-means with $K = 3$ was run on the data in the previous toy example with different random initializations. Three outcomes achieved the same (suboptimal) objective value.

- Alternative to K-means algorithm, does not require K to be specified in advance.
- Results in tree-based cluster representation called a **dendrogram**.
- Here: **bottom-up** or **agglomerative clustering**.



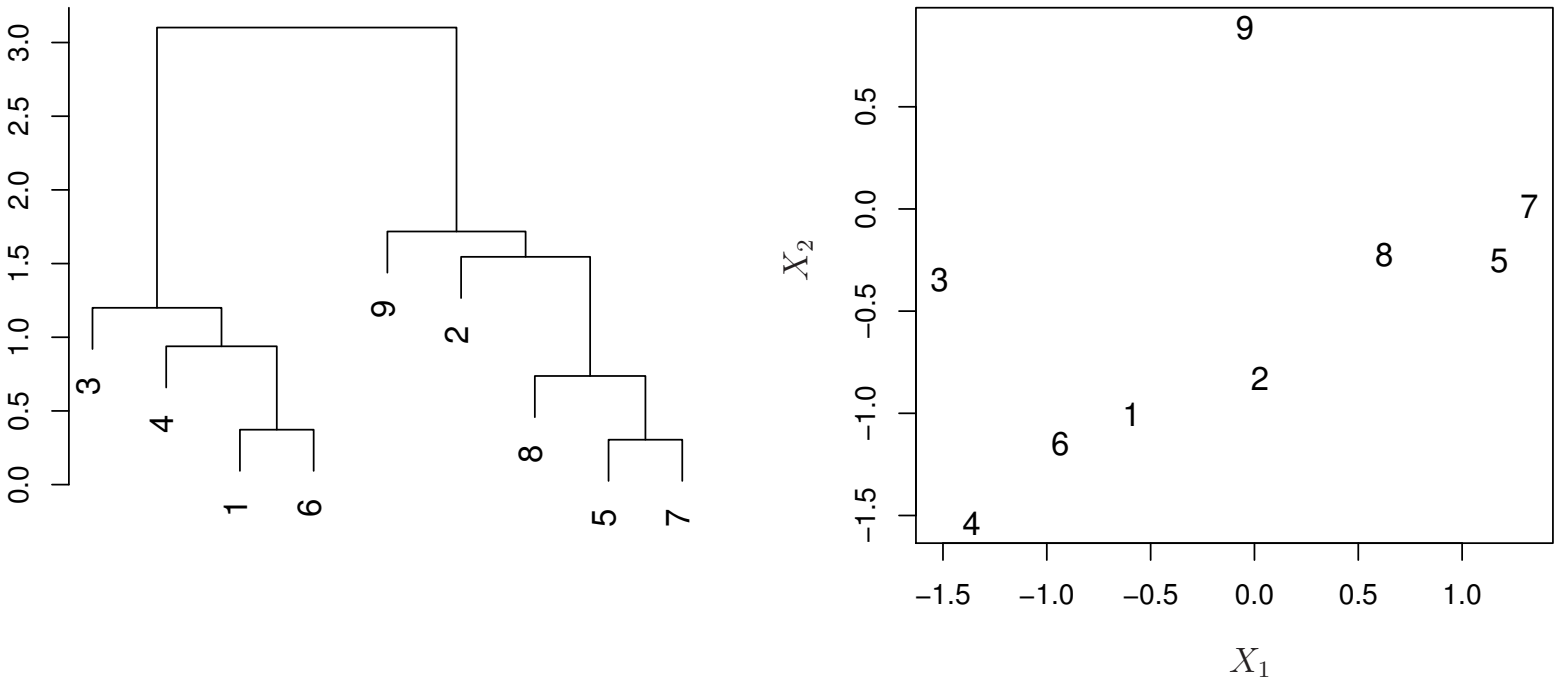


Simulated data, 45 observations, 3 classes, hierarchical clustering results in dendrogram on the right.

- Each leaf corresponds to one of the original 45 observations.
- Moving up the tree, some leaves begin to fuse into branches, reflecting similarity of the leaves.
- Advancing further up, branches fuse with leaves or other branches.
- Earlier fusion (bottom-up) indicates stronger similarity of (groups of) observations.
- More precisely: for any pair of observations, the distance (from bottom) to where their subtrees are first joined is a measure of their non-similarity.

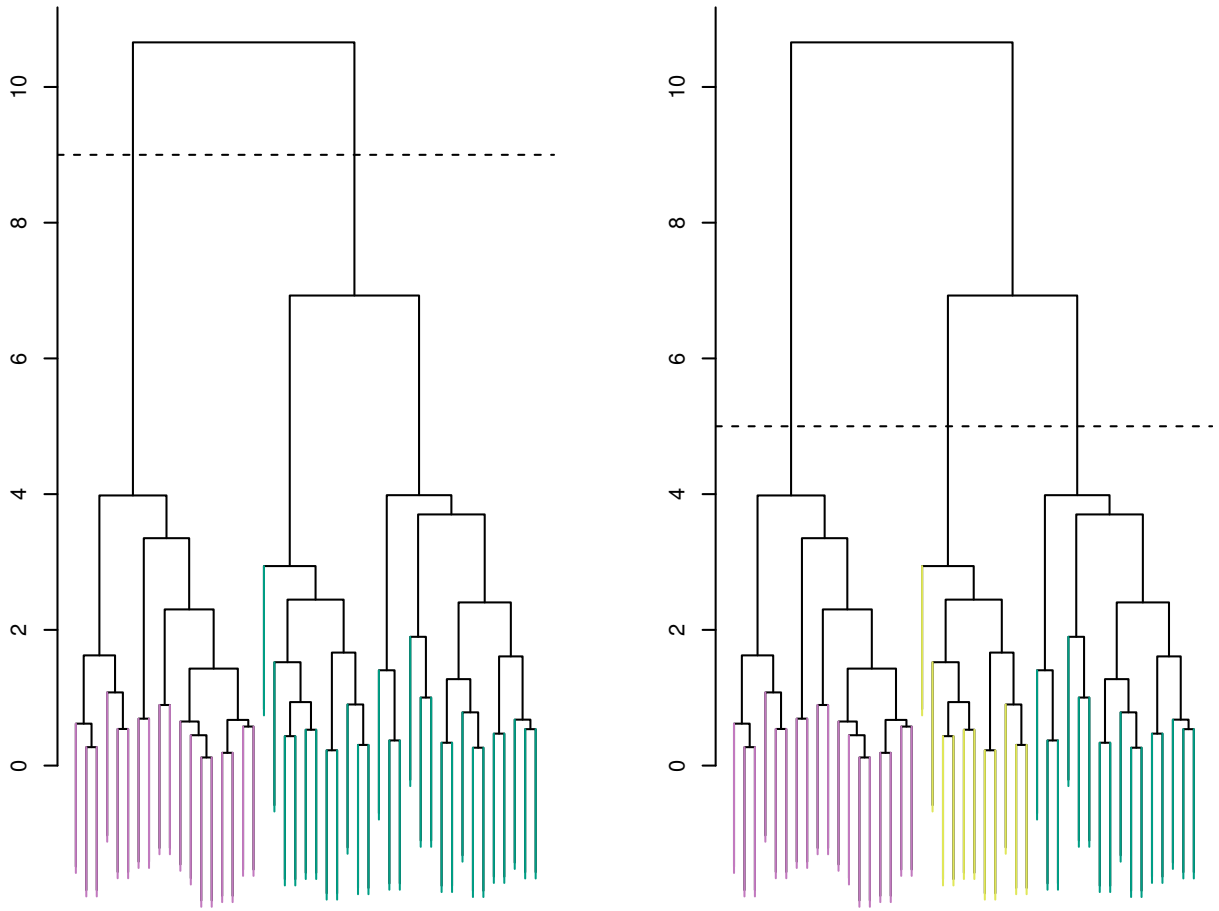
Unsupervised Learning

Hierarchical clustering, interpreting a dendrogram



Left: dendrogram of 9 observations of two-dimensional data. Right: Original data.
1 and 6 as well as 5 and 7 very similar; 9 no more similar to 2 than to 8, 5 and 7, even though 9 and 2 close horizontally in dendrogram; 2, 8, 5, 7 all fuse with 9 at same height, ≈ 1.8.

Cutting a dendrogram horizontally, the distinct sets of observations beneath the cut can be interpreted as clusters.



On the left, cutting dendrogram at height of 9 yields 2 clusters.

On the right, cutting at height 5 yields 3 clusters.

Further cuts can be made at different heights yielding clusters of size between 1 (no cut) and n (cut at height 0).

Height plays same role as K in K-means clustering.

Unsupervised Learning

Hierarchical clustering, identifying clusters from a dendrogram

- Single dendrogram yields any number of clusters.
- Cut usually chosen by inspection.
- **Hierarchical** refers to the fact that clusters from different heights in the same dendrogram are nested. However, nested structure not always realistic. (Group split 50-50 among males and females, and equally split among 3 nationalities.) In such situations K-means may yield better results.

- Introduce measure of dissimilarity between observation pairs. e.g. Euclidean distance.
- Start at bottom, each observation treated as its own cluster.
- Two most similar clusters fused, yielding $n-1$ clusters.
- Next fusion yields $n-2$ clusters.
- Proceed until single cluster remains.

Algorithm 7: Hierarchical clustering.

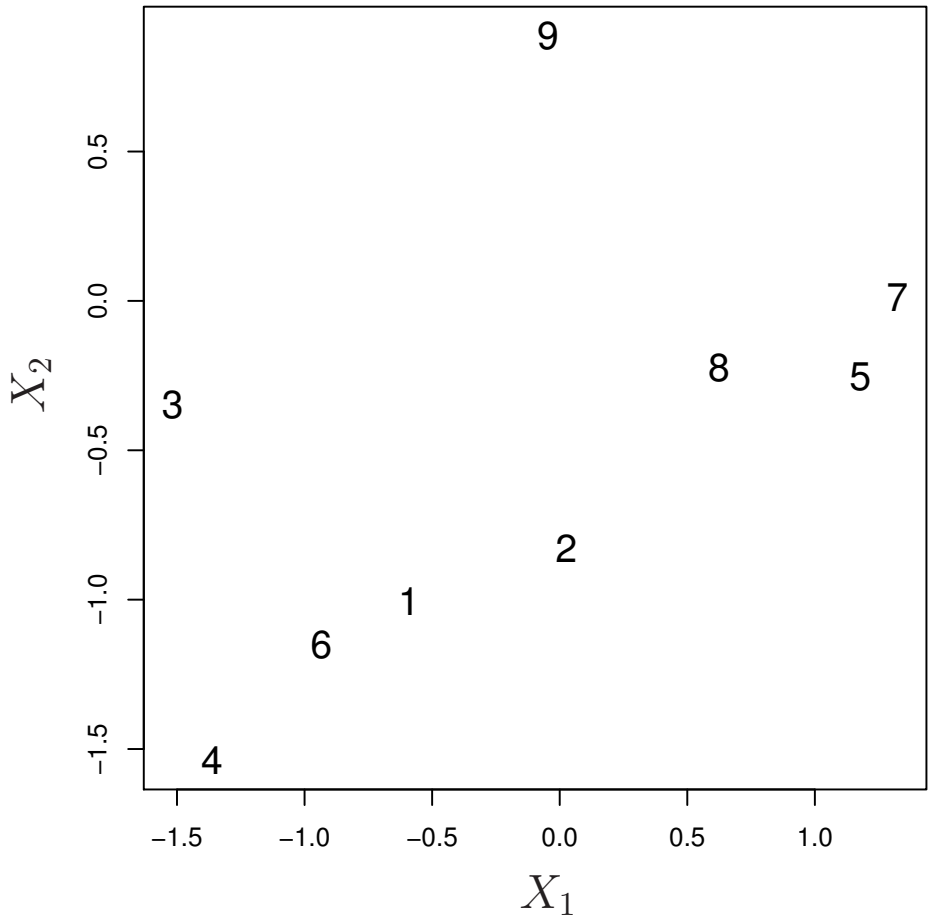
1. Begin with n observations and a measure of all $n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.
2. For $i = n, n-1, \ldots, 2$:
   a. Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.
   b. Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

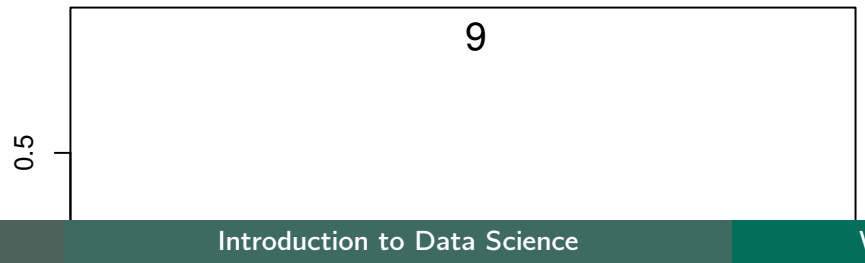
- How is distance measure between groups of observations defined?
- Different notions of **linkage** possible

Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

Unsupervised Learning

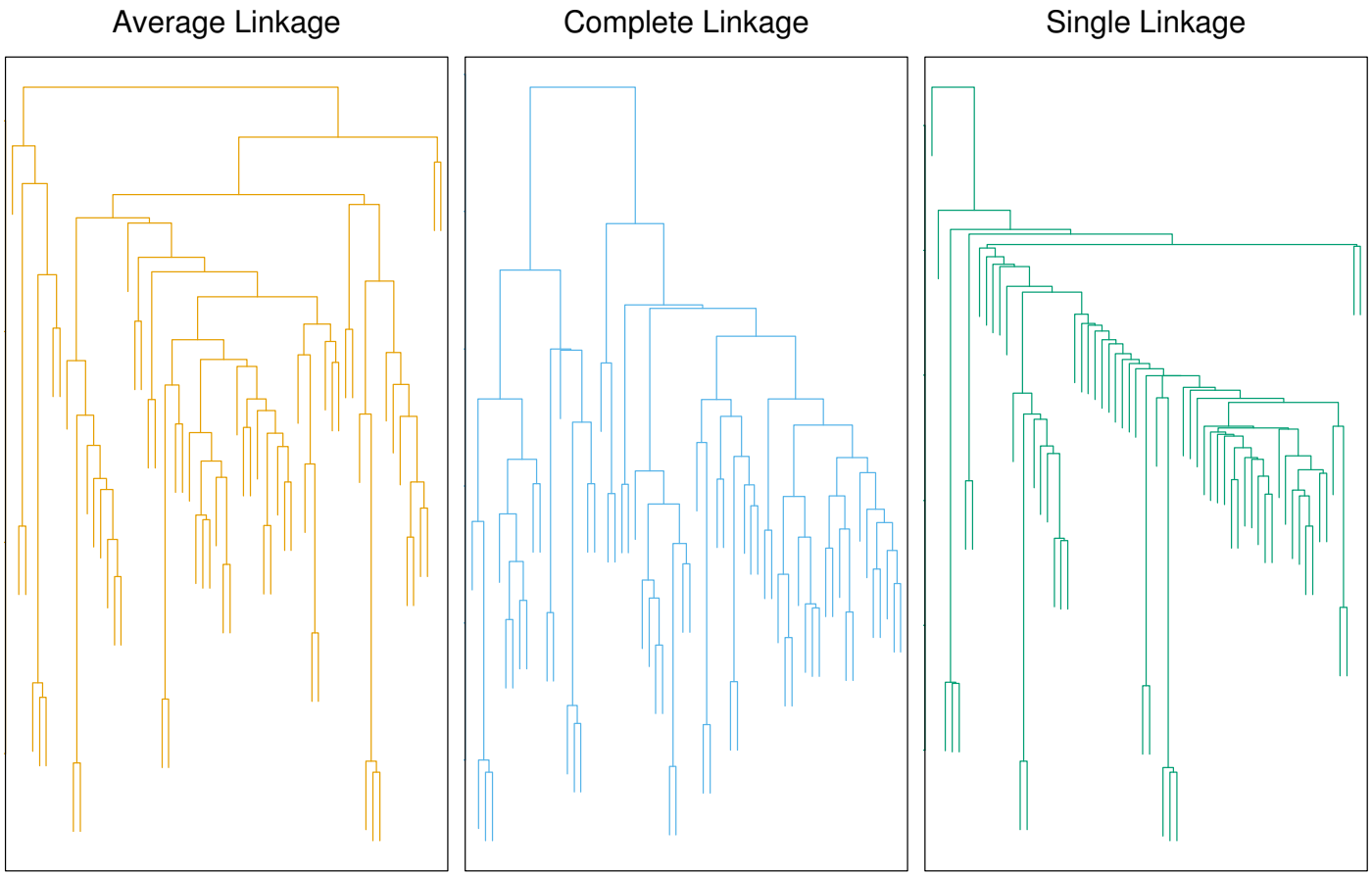
Hierarchical clustering algorithm: linkage





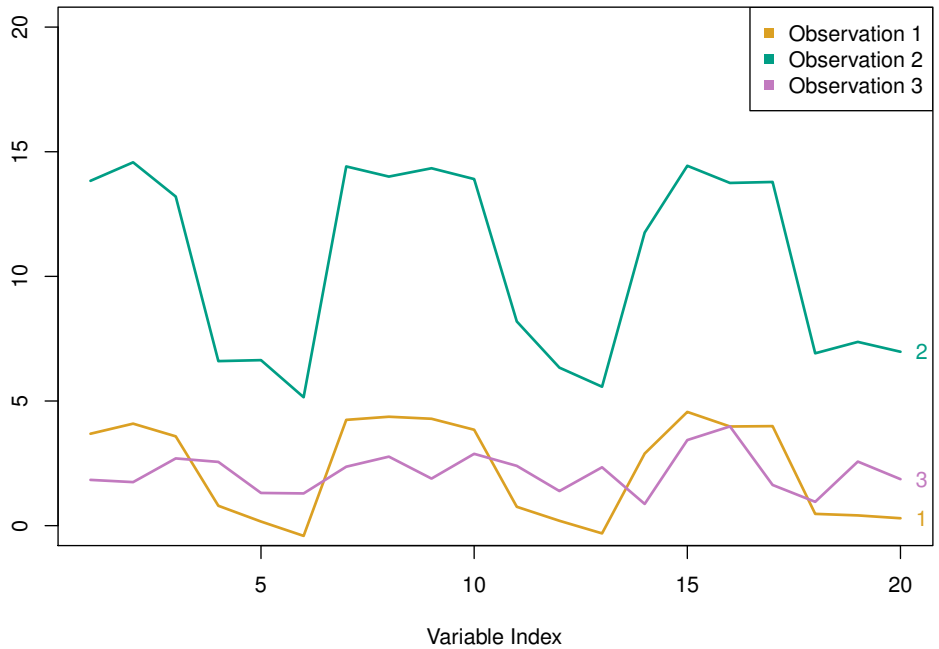
Unsupervised Learning

Hierarchical clustering algorithm: linkage



Dendrogram resulting from hierarchical clustering algorithm using average, complete and single linkage applied to the same data set. Average and complete linkage tend to produce more balanced dendrograms.

- Alternative to Euclidean distance: **correlation-based distance**, which considers two observations similar if their features are highly correlated.
- This may be true even if their Euclidean distance is large.



3 observations of 20 variables each. 1 and 3 have similar values (small Euclidean distance) but are weakly correlated. 1 and 2 have a large Euclidean distance but are closely correlated.

Example: Online retailer clustering customers

- Objective: cluster shoppers based on their past shopping histories; identify subgroups of similar shoppers so each group can be shown items/ads of shared interest.
- Data as matrix: rows shoppers, columns items for sale, entries # times shopper has purchased item.
- In Euclidean distance, shoppers who have purchased very few items would be close (may not be desirable).
- In correlation-based distance, shoppers with similar preferences (e.g. who bought items A and B but never C and D) would be close, even if some have purchased in higher volume than others.
- Here correlation-based distance probably better.

Unsupervised Learning

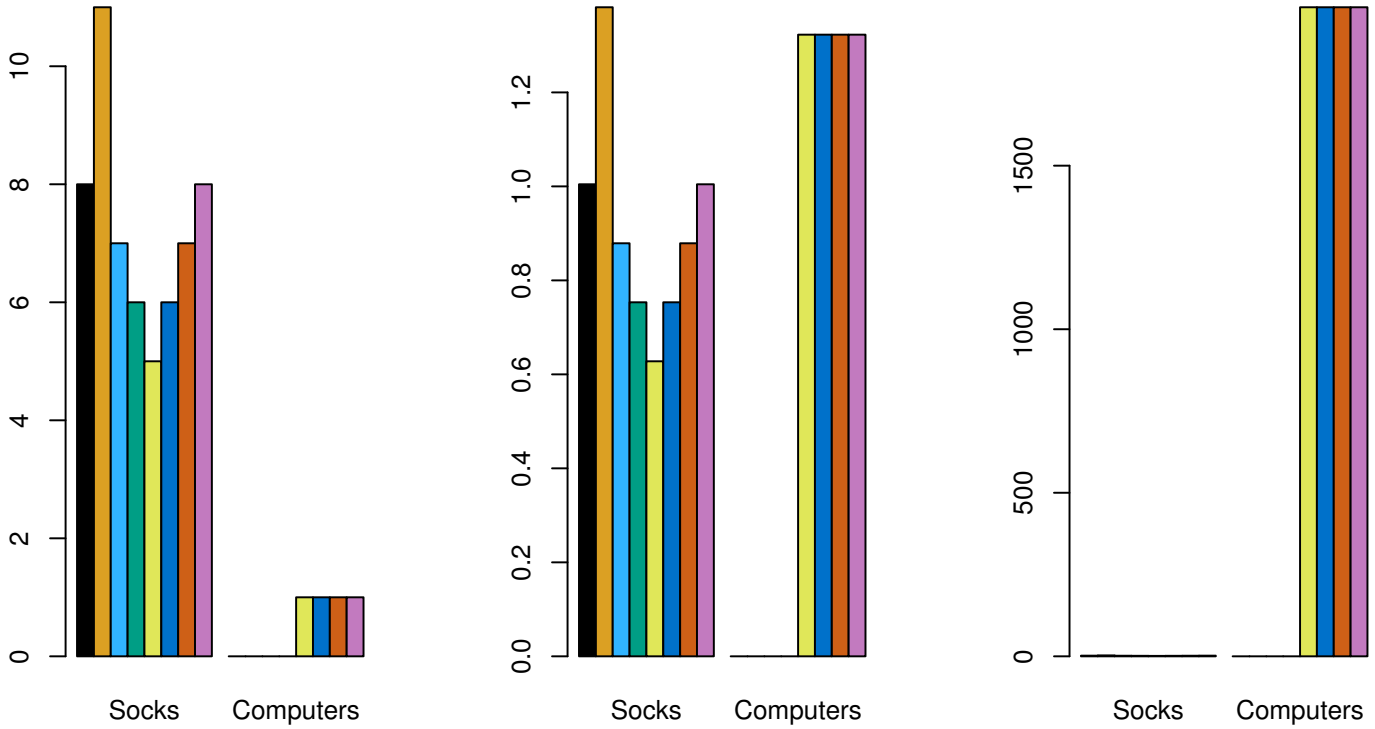
Hierarchical clustering: scaling issues

Scale data to standard deviation one before applying dissimilarity measure?

- Online store again: some items likely purchased more often than others (socks vs. computers).
- High-frequency purchases tend to have stronger effect on inter-shopper dissimilarity.
- Scaling to unit standard deviation before computing inter-observation dissimilarity gives each variable equal importance.
- Also advisable when observation features measured in different scales/units.
- Applies to K-means clustering as well.

Unsupervised Learning

Hierarchical clustering: scaling issues



Online retailer selling only socks and computers. Left: # socks/computers purchased by 8 customers (distinguished by color). In Euclidean-based distance of raw data, computer purchases have little or no effect (less informative, computers have higher margins). Center: each variable scaled by its standard deviation. Right: same data, with y-axis showing amount spent on each item.

Decisions to make a priori

- Standardize observations/features before measuring similarity? (Centering, scaling)
- For hierarchical clustering:
  - Choice of dissimilarity measure?
  - Choice of linkage?
  - Choice of dendrogram cutting height?
- For K-means: choice of K?

Validating obtained clusters

- Have we found meaningful subgroups or only clustered the noise?
- Some proposals for assigning p-values to clusters given in [Hastie et al., 2009]

Further issues

- Sometimes assigning all observations to clusters may be inappropriate.
- Example: most observations belong to small number of (unknown) subgroups. A few observations very different from rest. This presence of outliers which shouldn't be in any cluster can heavily distort the clustering outcome.
- This issue addressed by **mixture models** (soft version of K-means clustering), described in [Hastie et al., 2009].
- Non-robustness to data perturbations: perform clustering on n observations, repeat after randomly removing observations. Often result will strongly differ.
- **Recommendations:** Perform clustering repeatedly with different parameter choices and look for patterns which consistently emerge. Also cluster subsets to obtain sense of robustness. View results not as absolute truth, but as starting point for further investigation.