



# STK4030 - Statistical Learning: Advanced Regression and Classification

Riccardo De Bin

[debin@math.uio.no](mailto:debin@math.uio.no)

## Outline of the lecture

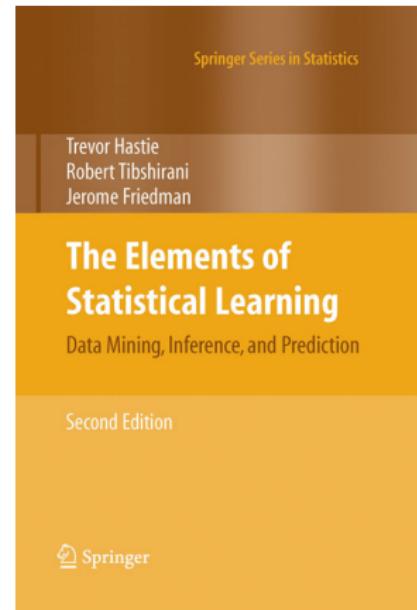
- Introduction
- Overview of supervised learning
  - Variable types and terminology
  - Two simple approaches to prediction: least square and nearest neighbors
  - Statistical decision theory
  - Local methods in high dimensions

## Introduction: Elements of Statistical Learning

This course is based on the book:

**“The Elements of Statistical Learning:  
Data Mining, Inference, and Prediction”**  
by T. Hastie, R. Tibshirani and J.  
Friedman:

- reference book on modern statistical methods;
- free online version,  
<https://web.stanford.edu/~hastie/ElemStatLearn/>.



## Introduction: statistical learning

*“We are drowning in information, but we starved from knowledge”*  
*(J. Naisbitt)*

- nowadays a **huge quantity of data** is continuously collected  
⇒ a lot of **information** is available;
- we struggle with profitably using it;

The **goal of statistical learning** is to “get knowledge” from the data, so that the information can be used for prediction, identification, understanding, . . .

## Introduction: email spam example

Goal: construct an automatic spam detector that block spam.

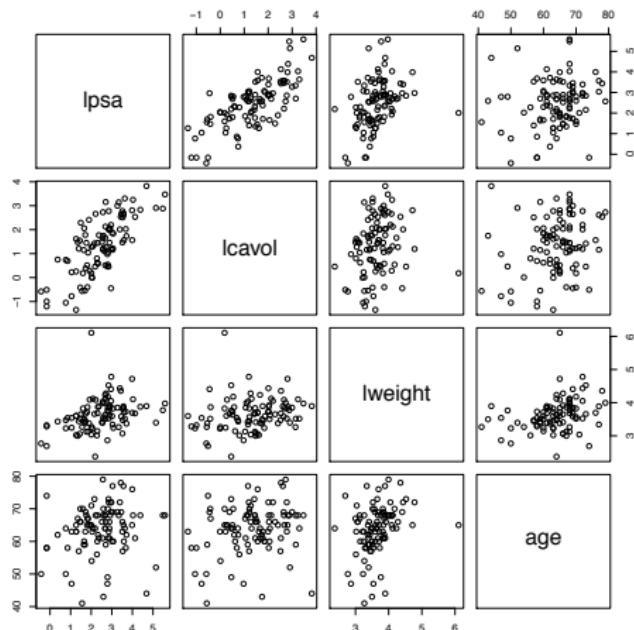
Data: information on 4601 emails, in particular,

- whether was it spam (`spam`) or not (`email`);
- the relative frequencies of 57 of the most common words or punctuation marks.

word	george	you	your	hp	free	hpl	!	...
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	...
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	...

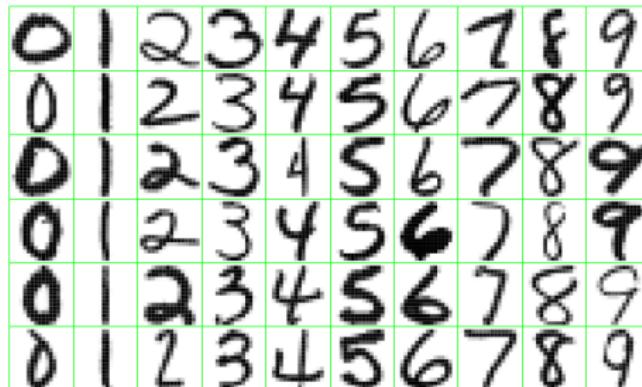
Possible rule: if (%george < 0.6) & (%you > 1.5) then `spam`  
else `email`

## Introduction: prostate cancer example



- data from Stamey et al. (1989);
- goal: predict the level of (log) prostate specific antigen (lpsa) from some clinical measures, such as log cancer volume (lcavol), log prostate weight (lweight), age (age), ...;
- possible rule:  
$$f(X) = 0.32 \text{ lcavol} + 0.15 \text{ lweight} + 0.20 \text{ age}$$

## Introduction: handwritten digit recognition



- data:  $16 \times 16$  matrix of pixel intensities;
- goal: identify the correct digit (0, ..., 9);
- the outcome consists of 10 classes.

FIGURE 1.2. Examples of handwritten digits from U.S. postal envelopes.

## Introduction: other examples

Examples (from the book):

- predict whether a patient, hospitalized due to a heart attack, will have a second heart attack, based on demographic, diet and clinical measurements for that patient;
- predict the price of a stock in 6 months from now, on the basis of company performance measures and economic data;
- identify the numbers in a handwritten ZIP code, from a digitized image;
- estimate the amount of glucose in the blood of a diabetic person, from the infrared absorption spectrum of that persons blood;
- identify the risk factors for prostate cancer, based on clinical and demographic.

## Introduction: framework

In a typical scenario we have:

- an **outcome**  $Y$  (dependent variable, response)
  - ▶ **quantitative** (e.g., stock price, amount of glucose, ...);
  - ▶ **categorical** (e.g., heart attack/no heart attack)

that we want to predict based on

- a set of **features**  $X_1, X_2, \dots, X_p$  (independent variables, predictors)
  - ▶ examples: age, gender, income, ...

In practice,

- we have a **training set**, in which we observe the outcome and some features for a set of observations (e.g., persons);
- we use these data to construct a **learner** (i.e., a rule  $f(X)$ ), which provides a prediction of the outcome ( $\hat{y}$ ) given specific values of the features.

## Introduction: supervised vs unsupervised learning

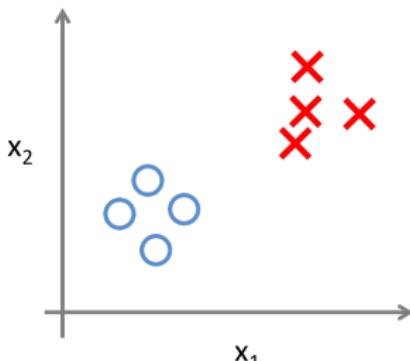
The scenario above is typical of a **supervised learning problem**:

- the outcome is measured in the training data, and it can be used to construct the learner  $f(X)$ ;

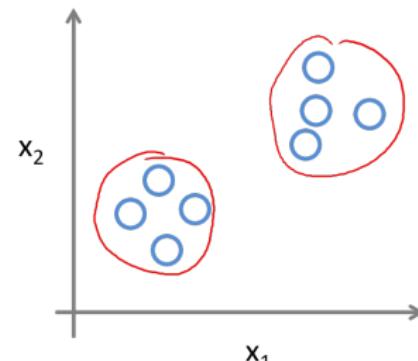
In other cases only the features are measured → **unsupervised learning problems**:

- identification of clusters, data simplification, . . .

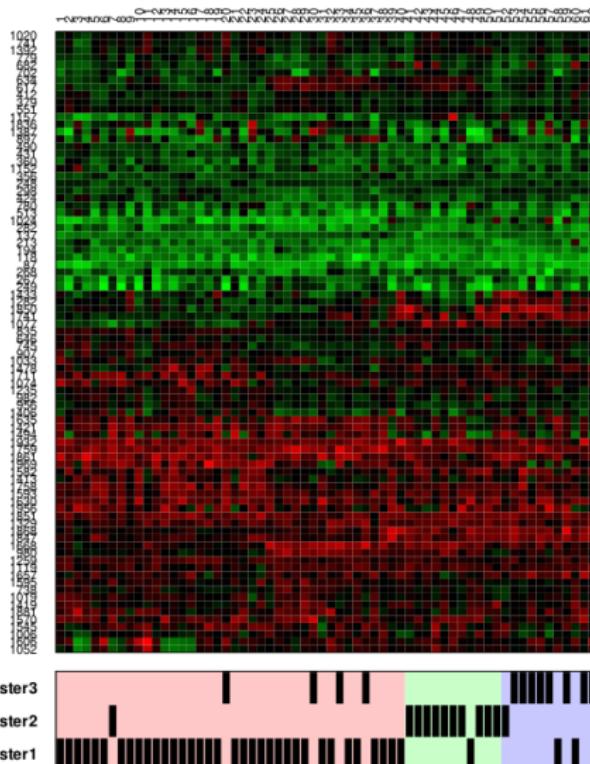
Supervised Learning



Unsupervised Learning



## Introduction: gene expression example



- heatmap from De Bin & Rissolo (2011): 62 obs vs a subset of the original 2000 genes
  - ▶  $p \gg n$  problem;
- goal: group patients with similar genetic information (cluster);
- alternatives (if the outcome was also available):
  - ▶ classify patients with similar disease (classification);
  - ▶ predict the chance of getting a disease for a new patient (regression).

## Introduction: the high dimensional issue

Assume a training set  $\{(x_{i1}, \dots, x_{ip}, y_i), i = 1, \dots, n\}$ , where  $n = 100, p = 2000$ ;

- possible model:  $y_i = \beta_0 + \sum_{j=1}^p \beta_j x_{ij} + \varepsilon_i$ ;
- least squares estimate:  $\hat{\beta} = (X^T X)^{-1} X^T y$ .

Exercise:

- go together in groups of 3-4;
- learn the names of the others in the group;
- discuss problems with the least squares estimate in this case;
- discuss possible ways to proceed;

## Introduction: the high dimensional issue

Major issue:  $X^T X$  is **not invertible**, infinitely many solutions!

Some possible directions:

- **dimension reduction** (reducing  $p$  to be smaller than  $n$ ),
  - ▶ remove variables having low correlation with response;
  - ▶ more formal subset selections;
  - ▶ select a few “best” linear combinations of variables;
- **shrinkage methods** (adding constraint to  $\beta$ ),
  - ▶ ridge regression;
  - ▶ lasso (least absolute shrinkage and selection operator)
  - ▶ elastic net.

## Introduction: course information

- Course: mixture between theoretical and practical;
- evaluation: project (practical) and written exam (theoretical);
- use of computer necessary;
- based on statistical package R:
  - ▶ suggestion: use R Studio ([www.rstudio.com](http://www.rstudio.com)), available at all Linux computers at the Department of Mathematics;
  - ▶ encouragement: follow good R programming practices, for instance consult Google's R Style Guide.

## Variable types and terminology

Variable types: quantitative (numerical), qualitative (categorical).

Naming convention for predicting tasks:

- quantitative response: **regression**;
- qualitative response: **classification**.

We start with the problem of taking two explanatory variables  $X_1$  and  $X_2$  and predicting a binary (two classes) response  $G$ :

- we illustrate two basic approaches:
  - ▶ linear model with least squares estimator;
  - ▶  $k$  nearest neighbors;
- we consider both from a **statistical decision theory** point of view.

Two simple approaches to prediction: linear regression model

## The linear regression model

$$\begin{aligned} Y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon \\ &= X\beta + \varepsilon, \end{aligned} \quad \text{where } X = (\mathbf{1}, x_1, \dots, x_p),$$

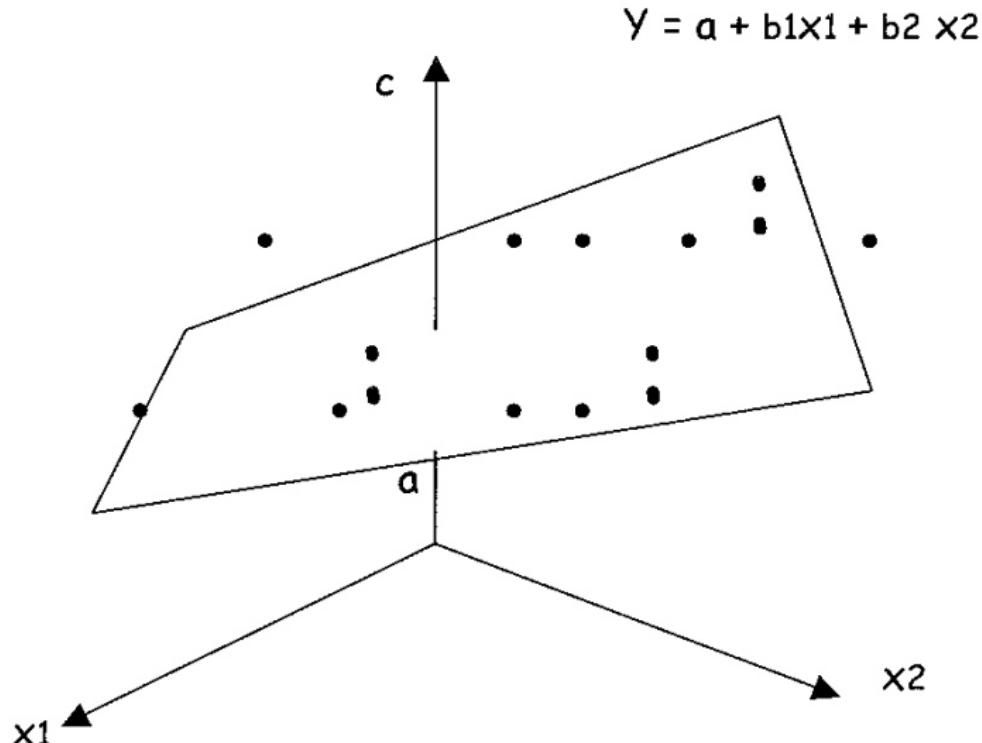
can be used to predict the outcome  $y$  given the values  $x_1, x_2, \dots, x_p$ , namely

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

### Properties:

- easy interpretation;
- easy computations involved;
- theoretical properties available;
- it works well in many situations.

## Two simple approaches to prediction: linear regression model



## Two simple approaches to prediction: least square

How do we **fit** the linear regression model to a training dataset?

- Most popular method: **least square**;
- estimate  $\beta$  by minimizing the **residual sum of squares**

$$\text{RSS}(\beta) = \sum_{i=1}^N (y_i - x_i^T \beta)^2 = (y - X\beta)^T (y - X\beta)$$

where  $X$  is a  $(N \times p)$  matrix and  $y$  a  $N$ -dimensional vector.

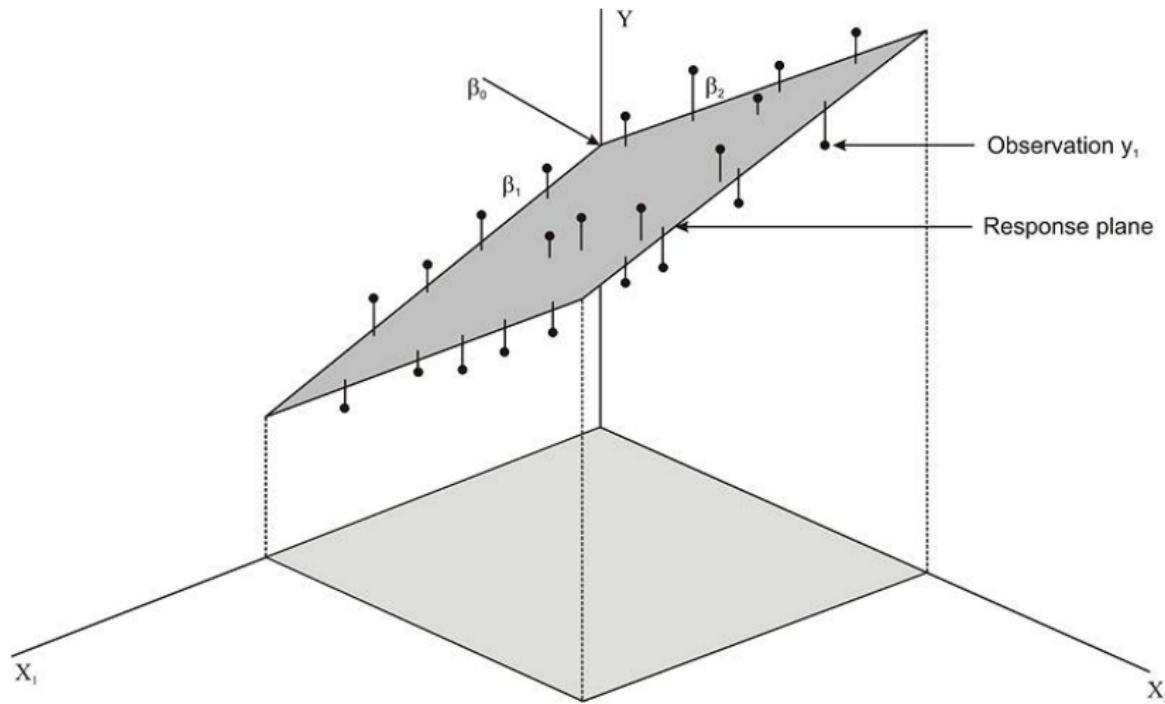
Differentiating w.r.t.  $\beta$ , we obtain the **estimating equation**

$$X^T(y - X\beta) = 0,$$

from which, when  $(X^T X)$  is non-singular, we obtain

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

## Two simple approaches to prediction: least square



Two simple approaches to prediction: least square for binary response

Simulated data with two variables and two classes:

$$Y = \begin{cases} 1 & \text{orange} \\ 0 & \text{blue} \end{cases}$$

If  $Y \in \{0, 1\}$  is treated as a numerical response

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2,$$

a prediction rule

$$\hat{G} = \begin{cases} 1 \text{ (orange)} & \text{if } \hat{Y} > 0.5 \\ 0 \text{ (blue)} & \text{otherwise} \end{cases}$$

gives linear decision boundary  $\{x^T \hat{\beta} = 0.5\}$

- optimal under Gaussian assumptions;
- is it better with nonlinear decision boundary?

## Two simple approaches to prediction: Nearest neighbor methods

A different approach consists in looking at the **closest** (in the input space) **observations** to  $x$  and, based on their output, form  $\hat{Y}(x)$ .

The  **$k$  nearest neighbors prediction** of  $x$  is the mean

$$\hat{Y}(x) = \frac{1}{k} \sum_{i:x_i \in N_k(x)} y_i,$$

where  $N_k(x)$  contains the  $k$  closest points to  $x$ .

- **less assumptions** on  $f(x)$ ;
- we need to decide  $k$ ;
- we need to define a metric (for now, consider the Euclidean distance).

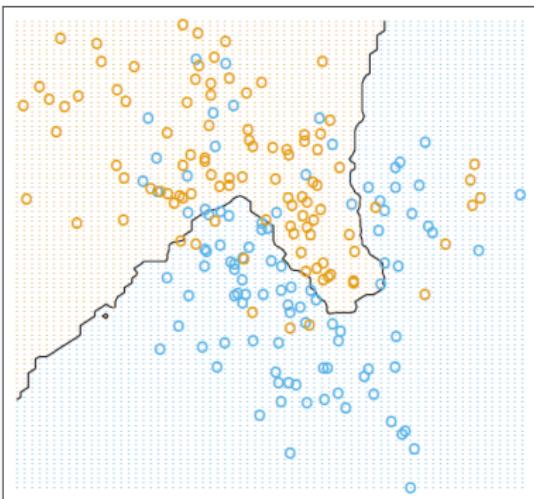
## Two simple approaches to prediction: nearest neighbor methods

Use the same training data  
(simulated) as before:

$$Y = \begin{cases} 1 & \text{orange} \\ 0 & \text{blue} \end{cases}$$

Classify to orange, if there are mostly  
orange points in the neighborhood:

$$\hat{G} = \begin{cases} 1 \text{ (orange)} & \text{if } \hat{Y} > 0.5 \\ 0 \text{ (blue)} & \text{otherwise} \end{cases}$$

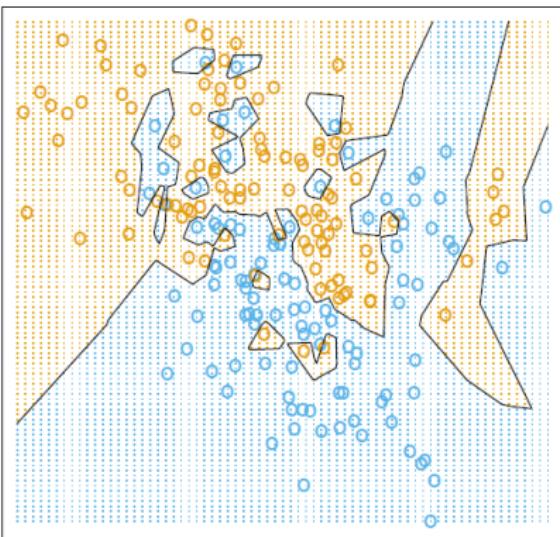


- $k = 15$ ;
- **flexible** decision boundary;
- better performance than the linear regression case:
  - ▶ fewer training observations are missclassified;
  - ▶ is this a good criterion?

## Two simple approaches to prediction: nearest neighbor methods

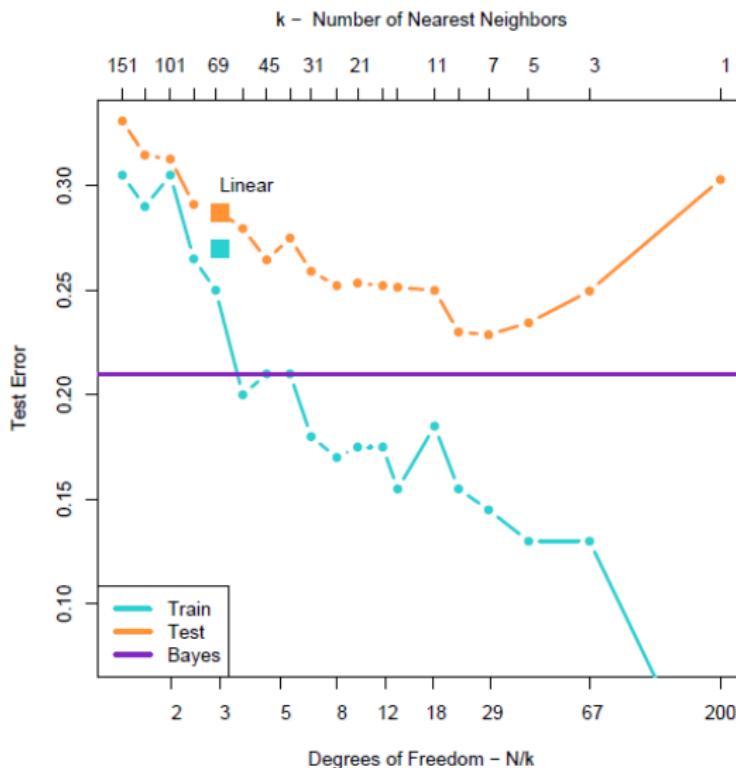
Using the same data as before: Note:

$$Y = \begin{cases} 1 & \text{orange} \\ 0 & \text{blue} \end{cases}$$



- same approach, with  $k = 1$ ;
- no training observations are missclassified!!!
- Is this a good solution?
  - ▶ the learner works greatly on the training set, but what is its prediction ability? (remember this term: **overfitting**);
  - ▶ It would be preferable to evaluate the performance of the methods in an independent set of observations (**test set**);
- bias-variance trade-off.

## Two simple approaches to prediction: how many neighbors in KNN?



## Two simple approaches to prediction: alternatives

Most of the modern techniques are **variants of these two simple procedures:**

- kernel methods that weight data according to distance;
- in high dimension: more weight on some variables;
- local regression models;
- linear models of functions of  $X$ ;
- projection pursuit and neural network.

## Statistical decision theory: theoretical framework

Statistical decision theory gives a mathematical framework for finding the optimal learner.

Let:

- $X \in \mathbb{R}^p$  be a p-dimensional random vector of inputs;
- $Y \in \mathbb{R}$  be a real value random response variable;
- $p(X, Y)$  be their joint distribution;

Our goal is to find a function  $f(X)$  for predicting  $Y$  given  $X$ :

- we need a loss function  $L(Y, f(X))$  for penalizing errors in  $f(X)$  when the truth is  $Y$ ,
  - ▶ example: squared error loss,  $L(Y, f(X)) = (Y - f(X))^2$ .

## Statistical decision theory: expected prediction error

Given  $p(X, Y)$ , it is possible to derive the **expected prediction error** of  $f(X)$ :

$$\text{EPE}(f) = E [L(Y, f(X))] = \int_{x,y} L(y, f(x))p(x, y)dxdy;$$

we have now a criterion for choosing a learner: find  $f$  which **minimizes**  $\text{EPE}(f)$ .

The aforementioned squared error loss,

$$L(Y, f(X)) = (Y - f(X))^2,$$

is by far the most common and convenient loss function. Let us focus on it!

## Statistical decision theory: squared error loss

If  $L(Y, f(X)) = (Y - f(X))^2$ , then

$$\begin{aligned}\text{EPE}(f) &= E_{X,Y}[(Y - f(X))^2] \\ &= E_X E_{Y|X}[(Y - f(X))^2 | X]\end{aligned}$$

It is then sufficient to minimize  $E_{Y|X}[(Y - f(X))^2 | X]$  for each  $X$ :

$$f(x) = \operatorname{argmin}_c E_{Y|X}[(Y - c)^2 | X = x],$$

which leads to

$$f(x) = E[Y | X = x],$$

i.e., the conditional expectation, also known as regression function.

Thus, by average squared error, the best prediction of  $Y$  at any point  $X = x$  is the conditional mean.

## Statistical decision theory: estimation of optimal $f$

In practice,  $f(x)$  must be estimated.

### Linear regression:

- assumes a function linear in its arguments,  $f(x) \approx x^T \beta$ ;
- $\operatorname{argmin}_{\beta} E[Y - X^T \beta]^2 \rightarrow \beta = E[XX^T]^{-1}E[XY]$ ;
- replacing the expectations by averages over the training data leads to  $\hat{\beta}$ .
- Note:
  - ▶ no conditioning on  $X$ ;
  - ▶ we have used our knowledge on the functional relationship to pool over all values of  $X$  (model-based approach);
  - ▶ less rigid functional relationship may be considered, e.g.

$$f(x) \approx \sum_{j=1}^p f(x_j).$$

## Statistical decision theory: estimation of optimal $f$

### $K$ nearest neighbors:

- uses **directly**  $f(x) = E[Y|X = x]$ :
- $\hat{f}(x_i) = \text{Ave}(y_i)$  for observed  $x_i$ 's;
- normally there is **at most** one observation for each point  $x_i$ ;
- uses points in the **neighborhood**,

$$\hat{f}(x) = \text{Ave}(y_i|x_i \in N_k(x))$$

- there are two approximations:
  - ▶ **expectation** is approximated by **averaging** over sample data;
  - ▶ conditioning on a **point** is relaxed to conditioning on a **neighborhood**.

## Statistical decision theory: estimation of optimal $f$

- assumption of  $k$  nearest neighbors:  $f(x)$  can be approximated by a **locally constant function**;
- for  $N \rightarrow \infty$ , all  $x_i \in N_k(x) \approx x$ ;
- for  $k \rightarrow \infty$ ,  $\hat{f}(x)$  is getting more stable;
- under mild regularity condition on  $p(X, Y)$ ,

$$\hat{f}(x) \rightarrow E[Y|X = x] \text{ for } N, k \rightarrow \infty \text{ s.t. } k/N \rightarrow 0$$

- is this an universal solution?
  - ▶ small sample size;
  - ▶ curse of dimensionality (see later)

## Statistical decision theory: other loss function

- It is **not necessary** to implement the squared error loss function ( $L_2$  loss function);
- a **valid alternative** is the  $L_1$  loss function:
  - ▶ the solution is the **conditional median**

$$\hat{f}(x) = \text{median}(Y|X = x)$$

- ▶ **more robust estimates** than those obtained with the conditional mean;
- ▶ the  $L_1$  loss function has **discontinuities in its derivatives** → numerical difficulties.

## Statistical decision theory: other loss functions

What happens with a **categorical outcome**  $G$ ?

- similar concept, different loss function;
- $G \in \mathcal{G} = \{1, \dots, K\} \rightarrow \hat{G} \in \mathcal{G} = \{1, \dots, K\}$ ;
- $L(G, \hat{G}) = L_{G, \hat{G}}$  a  **$K \times K$  matrix**, where  $K = |G|$ ;
- each element of the matrix  $l_{ij}$  is the **price to pay to missallocate** category  $g_i$  as  $g_j$ 
  - ▶ all elements on the diagonal are 0;
  - ▶ often non-diagonal elements are 1 (zero-one loss function).

## Statistical decision theory: other loss functions

Mathematically:

$$\begin{aligned} EPE &= E_{G,X}[L(G, \hat{G}(X))] \\ &= E_X \left[ E_{G|X}[L(G, \hat{G}(X))] \right] \\ &= E_X \left[ \sum_{k=1}^K L(g_k, \hat{G}(X)) \Pr(G = g_k | X) \right] \end{aligned}$$

which is sufficient to be minimized pointwise, i.e,

$$\hat{G} = \operatorname{argmin}_{g \in \mathcal{G}} L(g_k, g) \Pr(G = g_k | X = x).$$

When using the 0-1 loss function

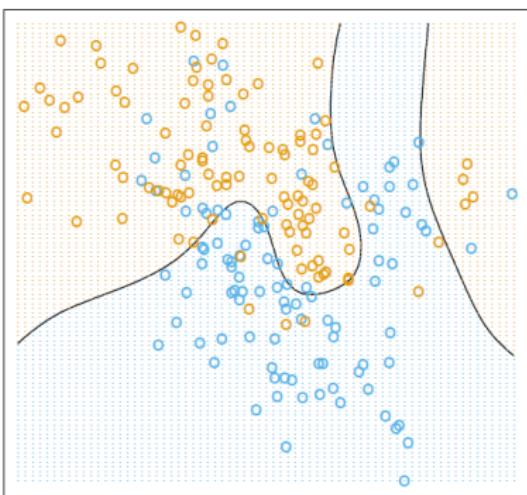
$$\begin{aligned} \hat{G} &= \operatorname{argmin}_{g \in \mathcal{G}} \sum_{k=1}^K \{1 - I(G = g_k)\} \Pr(G = g_k | X = x) \\ &= \operatorname{argmin}_{g \in \mathcal{G}} \{1 - \Pr(G = g_k | X = x)\} \\ &= \operatorname{argmax}_{g \in \mathcal{G}} \Pr(G = g_k | X = x) \end{aligned}$$

## Statistical decision theory: other loss functions

Alternatively,

$$\hat{G}(x) = g_k \text{ if } P(G = g_k | X = x) = \max_{g \in \mathcal{G}} \Pr(G = g | X = x),$$

also known as **Bayes classifier**.



- $k$  nearest neighbor:
  - ▶  $\hat{G}(x) = \text{category with largest frequency in } k \text{ nearest samples};$
  - ▶ approximation of this solution.
- regression:
  - ▶  $E[Y_k | X] = \Pr(G = g_k | X);$
  - ▶ also approximates the Bayes classifier.

## Local methods in high dimensions

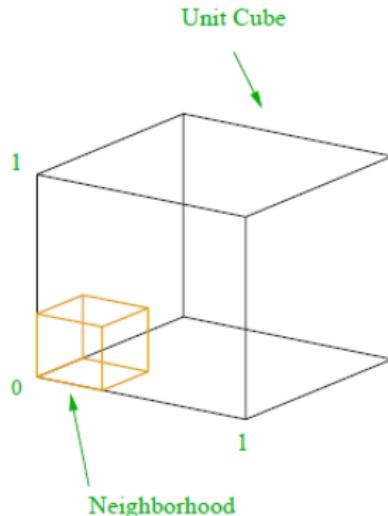
The two (extreme) methods seen so far:

- linear model, **stable but biased**;
- $k$ -nearest neighbor, **less biased but less stable**.

For large set of training data:

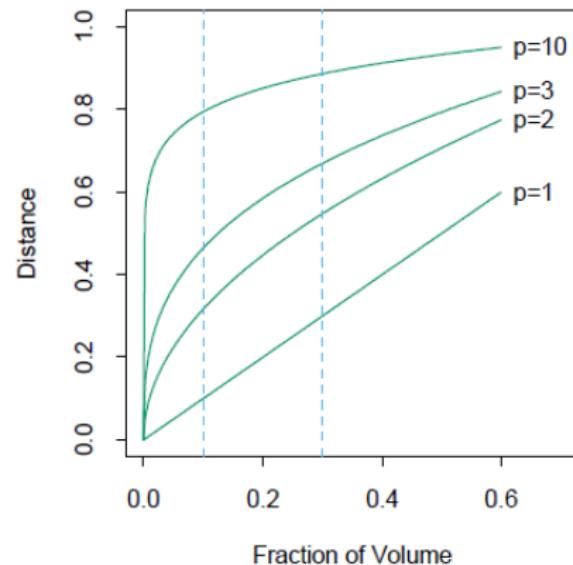
- always possible to use  $k$  nearest neighbors?
- Breaks down in high dimensions → **curse of dimensionality** (Bellman, 1961).

## Local methods in high dimensions: curse of dimensionality



- Assume  $X \sim \text{Unif}[0, 1]^p$ ;
- define  $e_p$  the **expected length size** of a hypercube containing a fraction  $r$  of input points;
- $e_p(r) = r^{1/p}$  ( $e^p = r \Leftrightarrow e = r^{1/p}$ );

## Local methods in high dimensions: curse of dimensionality



- Expected length:  $e_p(r) = r^{1/p}$

$p$	1	2	3	5
$e_p(0.01)$	0.01	0.10	0.22	0.63
$e_p(0.1)$	0.10	0.32	0.46	0.79

## Local methods in high dimensions: curse of dimensionality

Assume  $Y = f(X) = e^{-8||X||^2}$

and use the 1-nearest neighbor to predict  $y_0$  at  $x_0 = 0$ , i.e.

$\hat{y}_0 = y_i$  s.t.  $x_i$  nearest observed

$$\text{MSE}(x_0) =$$

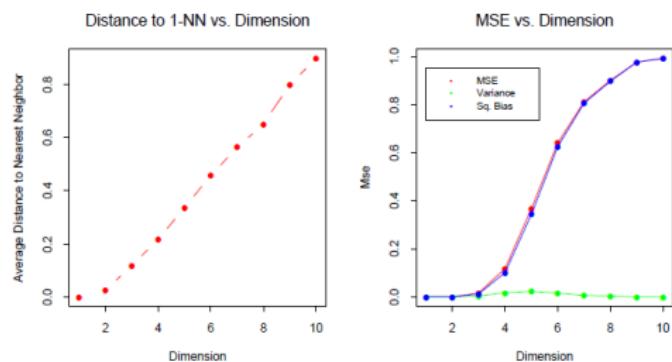
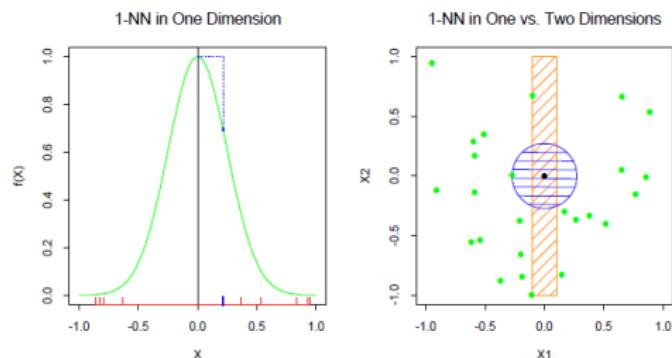
$$= E_{\mathcal{T}}[\hat{y}_0 - f(x_0)]^2$$

$$= E_{\mathcal{T}}[\hat{y}_0 - E_{\mathcal{T}}(\hat{y}_0)]^2$$

$$+ [E_{\mathcal{T}}(\hat{y}_0) - f(x_0)]^2$$

$$= \text{Var}(\hat{y}_0) + \text{Bias}^2(\hat{y}_0)$$

NB: we will see often this bias-variance decomposition!



## Local methods in high dimensions: EPE in the linear model

- Assume now  $Y = X^T \beta + \varepsilon$
- we want to predict  $y_0 = x_0^T \beta + \varepsilon_0$  with  $x_0$  fixed
- $\hat{y}_0 = x_0^T \hat{\beta}$  where  $\hat{\beta} = (X^T X)^{-1} X^T y$

$$\begin{aligned}\text{EPE}(x_0) &= E(y_0 - \hat{y}_0)^2 \\ &= E \left[ (y_0 - E[y_0|x_0] + E[y_0|x_0] - E[\hat{y}_0|x_0] + E[\hat{y}_0|x_0] - \hat{y}_0)^2 \right] \\ &= E(y_0 - E[y_0|x_0])^2 + (E[y_0|x_0] - E[\hat{y}_0|x_0])^2 \\ &\quad + E(\hat{y}_0 - E[\hat{y}_0|x_0])^2 \\ &= \text{Var}(y_0|x_0) + \text{Bias}^2(\hat{y}_0) + \text{Var}(\hat{y}_0)\end{aligned}$$

True and assumed linear model

- Bias=0
- $\text{Var}(\hat{y}_0) = x_0^T E(X^T X)^{-1} x_0 \sigma^2$
- $\text{EPE}(x_0) = \sigma^2 + x_0^T E(X^T X)^{-1} x_0 \sigma^2$

## Local methods in high dimensions: EPE in the linear model

- $\text{EPE}(x_0) = \sigma^2 + x_0^T E(X^T X)^{-1} x_0 \sigma^2$
- If  $x$ 's drawn from a random distribution with  $E(X) = 0$ ,  $X^T X \rightarrow N\text{Cov}(X)$
- Assume also  $x_0$  drawn from same distribution:

$$\begin{aligned} E_{x_0} [\text{EPE}(x_0)] &\approx \sigma^2 + E_{x_0}[x_0^T \text{Cov}(X)^{-1} x_0 N^{-1} \sigma^2] \\ &= \sigma^2 + N^{-1} \sigma^2 \text{trace}[\text{Cov}(X)^{-1} E_{x_0}[x_0 x_0^T]] \\ &= \sigma^2 + N^{-1} \sigma^2 \text{trace}[\text{Cov}(X)^{-1} \text{Cov}(x_0)] \\ &= \sigma^2 + N^{-1} \sigma^2 \text{trace}[I_p] \\ &= \sigma^2 + N^{-1} \sigma^2 p \end{aligned}$$

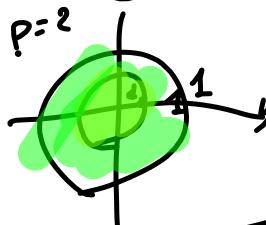
- It increases linearly with  $p$ !

## References |

- BELLMAN, R. (1961). *Adaptive control process: a guided tour*. Princeton University Press, London.
- DE BIN, R. & RISSO, D. (2011). A novel approach to the clustering of microarray data via nonparametric density estimation. *BMC Bioinformatics* **12**, 49.
- STAMEY, T. A., KABALIN, J. N., MCNEAL, J. E., JOHNSTONE, I. M., FREIHA, F., REDWINE, E. A. & YANG, N. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. ii. Radical prostatectomy treated patients. *The Journal of Urology* **141**, 1076–1083.

$$\Pr[\text{all } N \text{ points have distance } \geq d] = \frac{1}{2}$$

$t_i :=$  distance between  $x_i$  and origin

$$\Pr[t_i \geq d] = 1 - \Pr[t_i < d]$$


$$= 1 - \frac{\text{Volume of ball radius } d}{\text{Volume of ball radius } 1}$$

$$\text{Volume of ball with radius } r = \frac{\pi^{P/2}}{(P/2)!} r^P$$

$$P=2 \quad \frac{\pi r^2}{1^2}$$

$$= 1 - \frac{\frac{\pi^{P/2}}{(P/2)!} d^P}{\frac{\pi^{P/2}}{(P/2)!} 1^P} = 1 - d^P$$

$$\Pr[\forall i, t_i \geq d] = \prod_{i=1}^N \Pr[t_i \geq d]$$

" 1/k "

$$= (1 - d^P)^N$$

$$(1 - d^P)^N = \frac{1}{2}$$

$$1 - d^P = \left(\frac{1}{2}\right)^{1/N}$$

$$d^P = 1 - \left(\frac{1}{2}\right)^{1/N}$$

$$d = \left(1 - \left(\frac{1}{2}\right)^{1/N}\right)^{1/P}$$

qed

find  $\hat{f}(x)$  as a useful approximation of  $f(x)$

last week:  $L(y, f(x)) = (y - f(x))^2$  squared loss function

leads to  $f(x) = \underline{\mathbb{E}[Y|X=x]}$

k-nearest neighbour

$$\mathbb{E}[Y|X=x]$$

↑ average      → neighbours

issues

- curse of dimensionality
  - e.g. nearest point gets farer to the point of interest if increasing
- balance between bias-variance

## Statistics vs Machine Learning

Statistical approach:

Starting from the model

$$Y = f(x) + \varepsilon, \quad \mathbb{E}[\varepsilon] = 0, \quad \varepsilon \perp\!\!\!\perp X$$

additive model → approximation of the truth

statist.  $\hat{f}(x)$  approx  $f(x)$

- we do not suppose  $T = f(x)$  (deterministic)

BUT

we add an error term which captures:

- measurement errors;
- effects of non-measured variables;
- ...

Often  $\varepsilon \sim \text{iid}$ ,  $\varepsilon \sim \underline{\mathcal{N}}(0, \sigma^2)$

most natural approach, least square

## Machine learning approach

Assume  $Y = f(x) + \epsilon$

→ start from  $\hat{f}(x)$ , possibly simple  $\hat{f}(x) = c$   
(initialization)

→ evaluate  $\hat{f}(x)$  on our training set

$$\text{e.g. } \sum_{i=1}^n (y_i - \hat{f}(x_i))^2$$

→ modify  $\hat{f}(x)$  to improve the prediction

$$\sum_{i=1}^N (y_i - \hat{f}^{(k+1)}(x_i))^2 < \sum_{i=1}^N (y_i - \hat{f}^{(k)}(x_i))^2$$

→ use training set  $(x_i, y_i)$

→ learning by example

K is the step  
of the algorithm

→ focus is on the learner, that does not  
aim to approximate a true  $f(x)$ ,

statistical / mathematical approach

$(x_i, y_i)$  points of a p+1 dimensional space

$$f(x) : \begin{matrix} X \\ \cap \\ \mathbb{R}^p \end{matrix} \rightarrow \begin{matrix} Y \\ \cap \\ \mathbb{R} \end{matrix}$$

goal: giving an approximation of  $f(x)$   
working b/parts in  $X$  given  $T$

Parametric approach

$$\mathcal{F} = \left\{ p(y, x; \underline{\theta}) , \underline{\theta} \in \Theta \subseteq \mathbb{R}^P \right\}$$

$$Y = f(x; \underline{\theta}) + \varepsilon \quad f_{\underline{\theta}}(x) = f(x; \theta)$$

e.g.:

- linear model:  $f_{\underline{\theta}}(x) = x^T \underline{\beta}$   $\underline{\theta} = \underline{\beta}$

- linear basis expansion:  $f_{\underline{\theta}}(x) = \sum_{k=1}^K h_k(x) \theta_k$

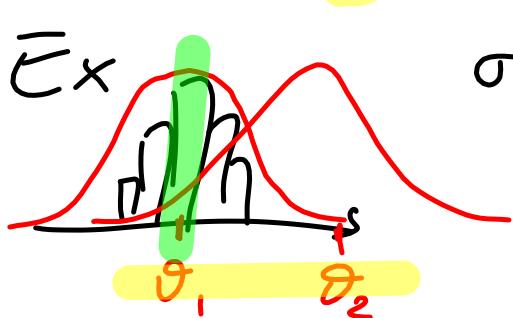
$$h_1(x) = x,$$

$$h_2(x) = x^2$$

$$RSS(\underline{\theta}) = \sum_{i=1}^n (y_i - f_{\underline{\theta}}(x_i))^2$$

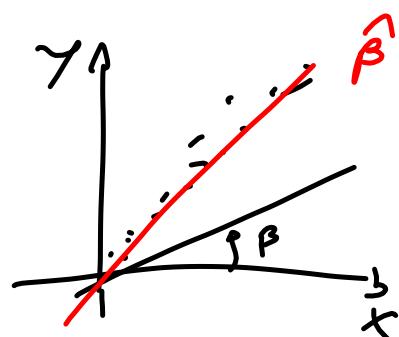
a function of  $\underline{\theta}$

$$\hat{\underline{\theta}} = \underset{\underline{\theta}}{\text{arg min}} RSS(\underline{\theta})$$



$$\sigma^2 = 1$$

$$\hat{\underline{\theta}} = \underline{\theta},$$



Simplest cases

- least squares

more complicated frameworks

- maximum likelihood estimation

## Likelihood estimators

$Y_i \sim P$   $p(y)$  is indexed by  $\theta$

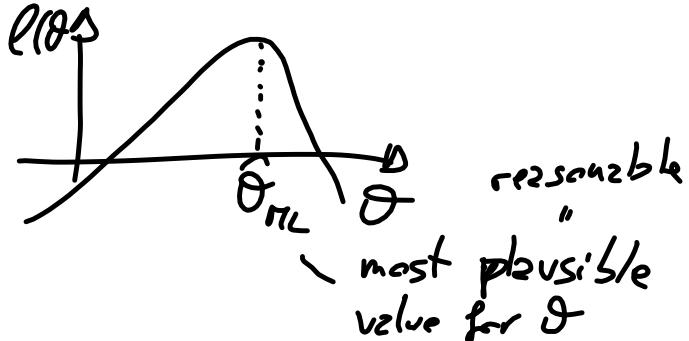
$Y_i$  i.i.d.  $p(y; \theta)$

e.g. Gaussian distributions  $P(y; \underbrace{\mu, \sigma^2}_{\theta})$

$$L(\theta) = \prod_{i=1}^N p(y_i; \theta)$$

$$\ell(\theta) = \sum_{i=1}^n \log p(y_i; \theta)$$

$$\hat{\theta}_{ML} = \underset{\theta}{\operatorname{argmax}} \ell(\theta)$$



Note: when  $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

$$\hat{\theta}_{ML} = \hat{\theta}_{LS}$$

## Restricted estimators

instead of estimating  $\theta$  as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} RSS(\theta)$$

we look for

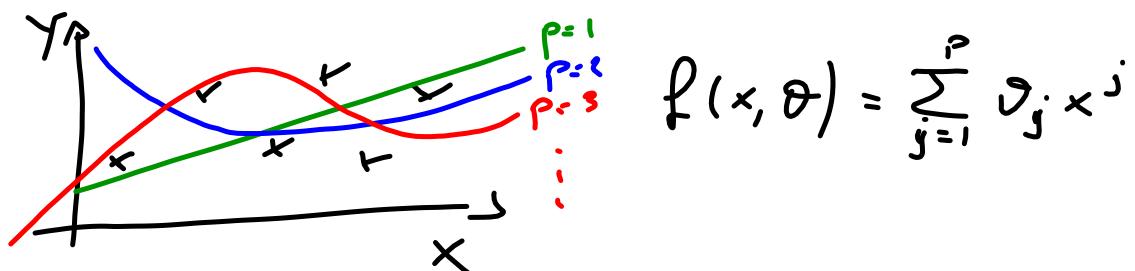
$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} PRSS(\theta)$$

where:  $PRSS(\theta) = RSS(\theta) + \lambda J(\theta)$

$$\text{LASSO} \quad J(\theta) = \sum_{j=1}^p |\theta_j|$$

$$\text{RIDGE} \quad J(\theta) = \sum_{j=1}^p \theta_j^2$$

## Model selection and variance-bias trade-off



given enough parameters  $\theta_j$ , we will always be able to find a line which passes through all points (no bias)

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left\{ RSS(\theta) + \lambda J(\theta) \right\}$$

↑ penalization for the number of parameters

Bias-variance trade off

$$\text{error} = \sigma^2 + \underbrace{\text{variance} + \text{bias}^2}_{\text{MSE}}$$

$$y \sim f(x_0)$$

$$Y = f(x) + \epsilon$$

see figure 2.11

Linear methods for regression

$E[Y|X]$  is linear in inputs

$$E[Y|X] = X^T \beta$$

→ simple

→ often adequate

→ easy to interpret

→ often outperforms fancier methods

↓  
simplicity!

→ sample size is small

→ sparse data

→ low signal-to-noise ratio

linear methods

regression  
(ch 3)

classification  
(ch 4)

### Regression

- consider continuous  $Y$

- linear regression  $f(x) = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} RSS(\beta)$$

$$\sum_{i=1}^n |y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j|^2$$

$$\text{matrix form } (y - X\beta)^T (y - X\beta)$$

$$\frac{\partial RSS(\beta)}{\partial \beta} = -2X^T(y - X\beta)$$

$$\frac{\partial^2 RSS(\beta)}{\partial \beta^2} = 2X^T X$$

$$\frac{\partial RSS(\beta)}{\partial \beta} = 0 \quad X^T(y - X\beta) = 0$$

$$X^T y - X^T X \beta = 0$$

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

$$\frac{\partial^2 RSS(\beta)}{\partial \beta^2} = 2X^T X > 0 \Rightarrow \hat{\beta} \text{ minimum}$$

$$\hat{y} = X \hat{\beta} = \underbrace{X(X^T X)^{-1} X^T}_H y$$

hat matrix  
projection matrix

### Properties

$$\text{Var}(\hat{\beta}) = (X^T X)^{-1} \sigma^2$$

$$\hat{\sigma}^2 = \frac{1}{N-p-1} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Focus on  $\varepsilon \sim N(0, \sigma^2)$   $\varepsilon$ : iid

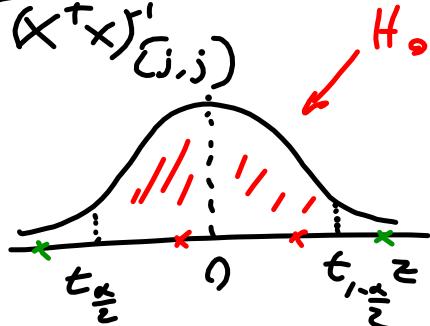
$$\hat{\beta} \sim N(\beta, (X^T X)^{-1} \sigma^2)$$

$$(N-p-1)\hat{\sigma}^2 \sim \sigma^2 \chi_{N-p-1}^2$$

$$H_0: \beta_j = 0 \rightarrow z_j = \frac{\hat{\beta}_j - 0}{sd(\hat{\beta}_j)} = \frac{\hat{\beta}_j}{\hat{\sigma} \sqrt{(x^T x)'_{(j,j)}}}$$

Under  $H_0$ ,  $z_j \sim t_{n-p-1}$

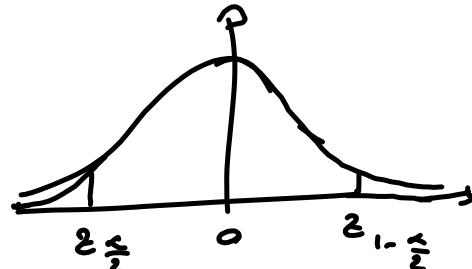
$z_{obs}$



$H_1 \Rightarrow \text{reject } H_0$

when  $\sigma^2$  is known

under  $H_0$ ,  $z_j \stackrel{H_0}{\sim} \mathcal{N}(0; 1)$



$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon$$

is  $x_2$  useful to predict  $y$ ?

$\beta_2 = 0 \rightarrow \epsilon \begin{cases} \text{reject } H_0 & \text{Yes} \\ \text{do NOT reject } H_0 & \text{not Yes} \end{cases}$

Are  $(x_2, x_3)$  useful to predict  $y$ ?

$H_0: \beta_2, \beta_3 = 0$

$$F = \frac{(RSS_0 - RSS_1) / (p_1 - p_0)}{RSS_1 / (n - p_1 - 1)}$$

# Today

- 1) 3 exercises
  - 2) Gauss-Markov theorem  
orthogonalization
  - 3) Model selection
  - 4) Shrinkage methods (ch 3.4)
- 

Ex 2.7

$\checkmark$  drawn  
N pairs  $(x_i, y_i)$  iid from

$$x_i \sim h(x)$$

$$y_i = f(x_i) + \varepsilon_i$$

$$\varepsilon_i \sim N(0, \sigma^2)$$

Estimator for  $f$  linear in the  $y_i$ :

$$\hat{f}(x_0) = \sum_{i=1}^N \ell_i(x_0; X) y_i$$

where weights  $\ell_i(x_0; X) \perp\!\!\!\perp y_i$ , but depend on  $X$ , the entire training sequence of  $x_i$ .

$$\begin{aligned} a) \hat{f}_{LR}(x_0) &= X_0^T \hat{\beta} = \underbrace{x_0^T (X^T X)^{-1} X^T}_{\hat{\beta}} y_i \\ &\Rightarrow \ell_i(x_0, X) = [x_0^T (X^T X)^{-1} X^T]_{[1]} \end{aligned}$$

$$\hat{f}_{KNN}(x_0) = \text{Ave} \{ y_i \mid x_i \in N_k(x_0) \}$$

where  $N_k(x_0)$  is the set of the  $k$  nearest neighbors

$$\Rightarrow \ell_i(x_0, X) = \begin{cases} 1/k & \text{if } x_i \in N_k(x_0) \\ 0 & \text{otherwise} \end{cases}$$

b) Decompose the conditional mean-squared error

$$E_{Y|X} \left[ (f(x_0) - \hat{f}(x_0))^2 \right]$$

student fixed

$$E_{Y|X} \left[ (f(x_0) - E_{Y|X}[\hat{f}(x_0)])^2 + E_{Y|X}[\hat{f}(x_0)] - \hat{f}(x_0))^2 \right]$$

$$= E_{Y|X} \left[ (f(x_0) - E_{Y|X}[\hat{f}(x_0)])^2 + (E_{Y|X}[\hat{f}(x_0)] - \hat{f}(x_0))^2 \right.$$

$$\left. + 2 (f(x_0) - E_{Y|X}[\hat{f}(x_0)]) (E_{Y|X}[\hat{f}(x_0)] - \hat{f}(x_0)) \right]$$

$$= \underbrace{(f(x_0) - E_{Y|X}[\hat{f}(x_0)])^2}_{\text{bias}^2} + \underbrace{\text{Var}_{Y|X}(\hat{f}(x_0))}_{\text{variance}} +$$

$$+ 2 (f(x_0) - E_{Y|X}[\hat{f}(x_0)]) E_{Y|X} \left[ \underbrace{E_{Y|X}[\hat{f}(x_0)] - \hat{f}(x_0)}_{\text{error}} \right]$$

= bias<sup>2</sup> + variance

c) unconditional

$$E_{Y|X} \left[ (f(x_0) - \hat{f}(x_0))^2 \right]$$

$$E_X \left[ E_{Y|X} \left[ (f(x_0) - \hat{f}(x_0))^2 \right] \right]$$

d) conditional:

$$(f(x_0) - E_{Y|X}[\hat{f}(x_0)])^2 + \text{Var}_{Y|X}(\hat{f}(x_0))$$

$$(f(x_0) - E_{Y|X} \left[ \sum_i \ell_i(x_0; X) g_i \right])^2 + \text{Var}_{Y|X} \left[ \sum_i \ell_i(x_0; X) g_i \right]$$

$$(f(x_0) - \sum_i \ell_i(x_0; X) f(x_0))^2 + \sum_i \ell_i(x_0; X)^2 \sigma^2$$

unconditional

$$E_X \left[ (f(x_0) - E_{Y|X} \left[ \sum_i \ell_i(x_0; X) g_i \right])^2 \right] = E_X \left[ \text{Var}_{Y|X} \left[ \sum_i \ell_i(x_0; X) g_i \right] \right]$$

Ex 3.1

$$F = \frac{(RSS_0 - RSS_1) / (P_0 - P_1)}{RSS_1 / (N - P_1 - 1)}$$

when we are testing only one parameter  $F \approx z^2$

$$\rightarrow P_1 = P, P_0 = P - 1$$

$$F = \frac{(RSS_0 - RSS_1)}{RSS_1 / (N - P - 1)} \sim F_{1, N-P-1}$$

$$z = \frac{\hat{\beta}_j}{\hat{\sigma}[x'x]_{jj}} \sim t_{N-P-1}$$

$$(t_{N-P-1})^2 \stackrel{d}{=} F_{1, N-P-1} \Rightarrow z^2 \approx F$$


---

Gauss-Markov theorem

$\hat{\beta}_{LS}$  is BLUE

B est  $\leftarrow$  smallest variance

L inear  $\leftarrow \hat{\theta} = \alpha^\top \hat{\beta} \leftrightarrow \theta = \alpha^\top \beta$

U nbiased  $\leftarrow E[\hat{\theta}] = \theta$

E stimator

consider  $X$  fixed

$$\begin{aligned} E[\hat{\theta}] &= E[\alpha^\top (X^\top X)^{-1} X^\top y] \\ &= \alpha^\top (X^\top X)^{-1} X^\top X \underbrace{\beta}_{I} \\ &= \alpha^\top \beta \end{aligned}$$

$$\text{Th: } \text{Var}(\hat{\theta}) \leq \text{Var}(\tilde{\theta})$$

$\rightarrow$  ex 3.3(a)

$$\text{MSE}(\hat{\theta}) = E[(\hat{\theta} - \theta)^2]$$

$$= E[(\hat{\theta} - E[\hat{\theta}] + E[\hat{\theta}] - \theta)^2]$$

$$= \text{Var}(\hat{\theta}) + (E[\hat{\theta}] - \theta)^2$$

smallest with LS

$\parallel$   
0

•  $MSE(\hat{f})$  is strongly related with the prediction error

Let consider  $y_0 = f(x_0) + \varepsilon$        $\varepsilon \sim N(0, \sigma^2)$

$$E[(y_0 - \hat{f}(x_0))^2] = E[y_0^2 + \hat{f}(x_0)^2 - 2y_0\hat{f}(x_0)]$$

$$= E[y_0^2] + E[\hat{f}(x_0)^2] - 2E[y_0\hat{f}(x_0)]$$

$$= Var(y_0) + E[y_0^2] + Var(\hat{f}(x_0)) + E[\hat{f}(x_0)]^2$$

$$- 2f(x_0)E[\hat{f}(x_0)] \quad \overbrace{(E[\hat{f}(x_0)] - y_0)^2}$$

$$= \sigma^2 + Var(\hat{f}(x_0)) + E[\hat{f}(x_0)]^2 - 2f(x_0)E[\hat{f}(x_0)] + f(x_0)^2$$

$$= \sigma^2 + Var(\hat{f}(x_0)) + \text{bias}^2 \quad \overbrace{MSE}$$

$$E[y_0] = E[f(x_0) + \varepsilon] = E[f(x_0)] + E[\varepsilon]$$

$$\stackrel{!}{=} f(x_0) + 0$$

$$E[y_0^2] = E[(f(x_0) + \varepsilon)^2]$$

$$\stackrel{!}{=} Var(y_0) + E[y_0]^2$$

$$\stackrel{!}{=} \sigma^2 + f(x_0)^2$$

Suppose  $Y = X\beta + \varepsilon$

*Univariable*  
if  $Y = X_1\beta_1 + \varepsilon$

$$\hat{\beta} = \frac{\sum x_i y_i}{\sum x_i^2} = \frac{\langle x, y \rangle}{\langle x, x \rangle}$$

*Multivariable*  
if  $Y = X_1\beta_1 + X_2\beta_2 + \varepsilon$

only if  $X$  is orthogonal ,  $\hat{\beta}_j = \frac{\langle x_j, y \rangle}{\langle x_j, x_j \rangle}$

alternative formula for variance of  $\hat{\beta}_j$ :

$$\text{Var}(\hat{\beta}_j) = \frac{\sigma^2}{\langle e_p, e_p \rangle} = \frac{\sigma^2}{\|e_p\|^2}$$

→ variables may share the same information useful for explaining/predicting  $y$

→ the estimates may be unstable due to high variance

↓  
model selection

$$z_{\text{score}} = \frac{\hat{\beta}}{\text{sd}(\hat{\beta})} \rightarrow \text{smaller}$$

$\rightarrow$  larger

## Model Selection

- prediction accuracy
- interpretability
- portability

- best subset technique

$x = (x_1, x_2, x_3)$

models : 0 variables  $y = \beta_0 + \varepsilon$

1 variable  $y = \beta_0 + \beta_1 x_1 + \varepsilon$

$$y = \beta_0 + \beta_2 x_2 + \varepsilon$$

$$y = \beta_0 + \beta_3 x_3 + \varepsilon$$

2 variables  $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$$

$\binom{3}{2} = \frac{3 \cdot 2}{2} = 3 \rightarrow y = \beta_0 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$

$\rightarrow \binom{8}{2} = \frac{8 \cdot 7}{2} = 28$  picture

- Stepwise techniques

- forward selection

start :  $y = \beta_0 + \varepsilon$   $x_1$ ,  $x_2$ ,  $x_3$

1<sup>st</sup> step :  $y = \beta_0 + \beta_2 x_2 + \varepsilon$   $x_1$ ,  $x_3$

2<sup>nd</sup> step :  $y = \beta_0 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$

$\hat{\beta}_3$  together with  $\hat{\beta}_0$  and  $\hat{\beta}_2$

- backward elimination

start : full model

p > n case  
backward  
elimination is

$$y = \beta_0 + \underline{\beta_1 x_1} + \underline{\beta_2 x_2} + \underline{\beta_3 x_3} + \varepsilon \quad \text{not possible}$$

1<sup>st</sup> step

$$y = \beta_0 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$$

:

:

- stepwise selection  
step back " "

$$\text{RSS}(\boldsymbol{\beta}) + J(\boldsymbol{\beta}) \\ + \underline{C_p}$$

---

stagewise regression

$$y = \beta_0 + \beta_1 x_2 + \varepsilon$$

$$y = \beta_0 + \beta_1 x_1 + \underline{\beta_2 x_2} + \varepsilon$$

stepwise: the estimate  
of  $\beta$  take into account  
all  $x$

stagewise: introduce  
2 new  $\beta$ , its estimate  
is only based on  $x_j$

Ex 3.3 (a)

$$\hat{\theta}_{LS} = \alpha^T \hat{\beta} = \alpha^T (X^T X)^{-1} X^T y \quad E[\alpha^T \hat{\beta}] = \alpha^T \beta$$

$$\tilde{\theta} = c^T y \quad \text{unbiased} \quad c^T = \alpha^T (X^T X)^{-1} X^T + \delta^T$$

$$E[\tilde{\theta}] = E[c^T y]$$

$$\begin{aligned} &= E[\alpha^T (X^T X)^{-1} X^T y + \delta^T y] \\ &= \alpha^T (X^T X)^{-1} X^T X \beta + \delta^T X \beta \\ &= \underline{\alpha^T \beta} + \underline{\delta^T X \beta} \Rightarrow \underline{\delta^T X} = 0 \end{aligned}$$

$$\text{Var}(\tilde{\theta}) = \text{Var}(c^T y)$$

$$\begin{aligned} &= c^T \sigma^2 c \\ &= \sigma^2 (\alpha^T (X^T X)^{-1} X^T + \delta^T) (\alpha^T (X^T X)^{-1} X^T + \delta^T)^T \\ &= \sigma^2 (\alpha^T (X^T X)^{-1} X^T + \delta^T) (X (X^T X)^{-1} \alpha + \delta) \\ &= \sigma^2 \left( \alpha^T (X^T X)^{-1} X^T X (X^T X)^{-1} \alpha + \alpha^T (X^T X)^{-1} X^T \delta + \right. \\ &\quad \left. + \delta^T X (X^T X)^{-1} \alpha + \delta^T \delta \right) \\ &= \sigma^2 \alpha^T (X^T X)^{-1} \alpha + \sigma^2 \delta^T \delta \\ &\quad \parallel \quad \text{VI} \\ &= \text{Var}(\hat{\theta}_{LS}) \quad 0 \end{aligned}$$

## Variable Selection

IDEA: remove irrelevant variables from the model

↑  
not useful to predict/explain the response

- less variance (despite small increase of \$S\_{res}\$)
- better interpretability
- better portability

When a variable is considered irrelevant

- p-value of a test  $> \alpha$ , usually  $\alpha=0.05$
- its inclusion increases a' information criterion

## INFORMATION CRITERIA

IDEA: instead of  $\hat{\theta} = \arg \min_{\theta} L(\theta)$ ,

We find  $\hat{\theta}_{IC} = \arg \min_{\theta} \{L(\theta) + \lambda J(\theta)\}$

$$\hat{\theta}_{IC} = \underline{X} \underline{\beta}$$

$$-2\ell(\theta)$$

$$\sum_{j=1}^p \mathbf{1}[\theta_j \neq 0]$$

GOAL: penalize larger models

$$L(\theta) = -2\ell(\theta)$$

$$J(\theta) = \sum_{j=1}^p \mathbf{1}[\theta_j \neq 0] \quad \# \text{variables in the model}$$

$$\lambda = \begin{cases} 2 & AIC \quad \text{Akaike information criterion} \\ \log(n) & BIC \quad \text{Bayesian} \end{cases}$$

$n$  is the sample size

NB: using AIC for model selection is like using  $\alpha=0.157$

if two explanatory variables are strongly correlated  $\rightarrow$  collinearity

extreme case: variables linearly dependent  
 $\rightarrow$  super-collinearity

in the case of super-collinearity  $(X^T X)^{-1}$  is not invertible  
 (not full rank)

Häerl & Kennard (1970)  $X^T X \rightarrow X^T X + \lambda I_p$

$$I_p = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & 0 & \cdots & 1 \end{pmatrix}$$

$$\hat{\beta}_{\text{ridge}}^{(\lambda)} = (X^T X + \lambda I_p)^{-1} X^T y \quad \lambda \in [0; \infty)$$

when  $\lambda \in (0; \infty)$   $(X^T X + \lambda I_p)^{-1}$  exists

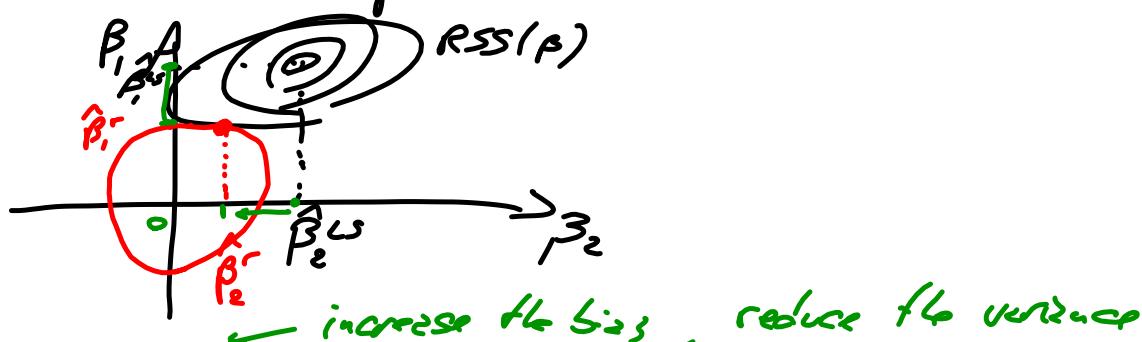
## RIDGE REGRESSION AS A SHRINKAGE METHOD

- $\hat{\beta}_{\text{ridge}}^{(\lambda)} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\}$

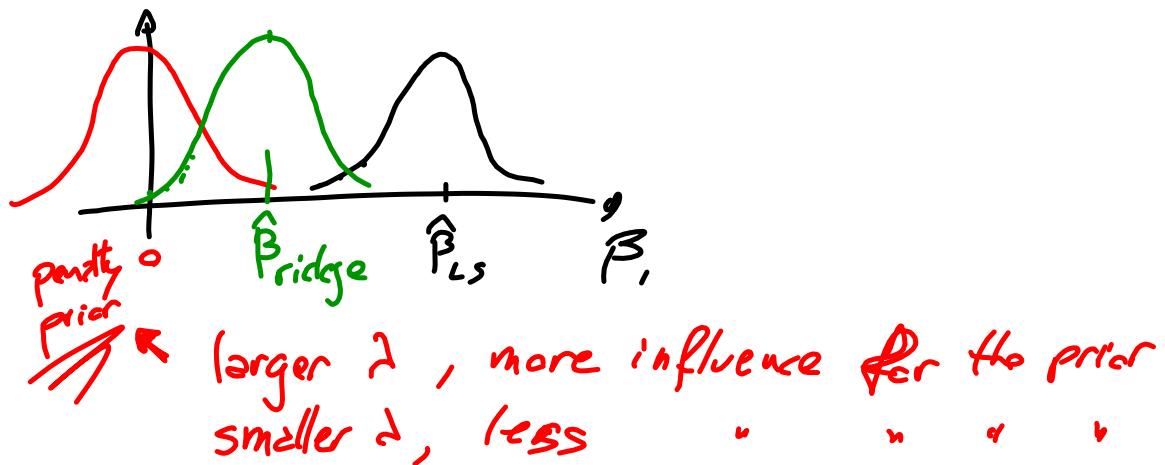
alternative formulation

- $\hat{\beta}_{\text{ridge}}^{(\lambda)} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 \right\}$   
 subject to  $\sum_{j=1}^p \beta_j^2 \leq t$

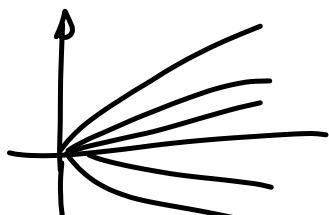
- one-to-one correspondence between  $\lambda$  and  $t$



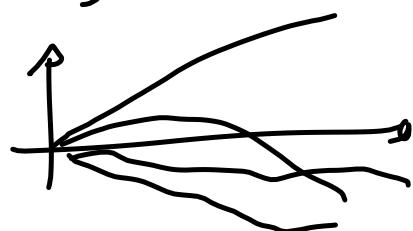
- from a Bayesian point of view, ridge estimator is the posterior mean/mode



if  $X_s$  are uncorrelated



if  $X_s$  are correlated



**IMPORTANT!** We need to standardize our explanatory variables before applying the ridge regression

$$E[X_j] = 0$$

$$\text{Var}[X_j] = 1$$


---

Expected value of ridge estimator

$$\begin{aligned}
 E[\hat{\beta}_{\text{ridge}}^{(\lambda)}] &= E[(X^T X + \lambda I_p)^{-1} X^T y] \\
 &= E[(I_p + \lambda (X^T X)^{-1})^{-1} (X^T X)^{-1} X^T y] \\
 &= (I_p + \lambda (X^T X)^{-1})^{-1} E[\hat{\beta}_{LS}] \\
 &= W_\lambda \beta \quad E[\hat{\beta}_{\text{ridge}}^{(\lambda)}] \neq \beta
 \end{aligned}$$

$$\lambda \rightarrow 0 \quad E[\hat{\beta}_{\text{ridge}}^{(\lambda)}] = \beta$$

$$\lambda \rightarrow \infty \quad E[\hat{\beta}_{\text{ridge}}^{(\lambda)}] = \sigma_{p+1} \leftarrow \begin{array}{l} \text{without intercept} \\ \beta_0 = 0 \end{array}$$

$$\text{important} \quad \lambda_a > \lambda_s \quad \Rightarrow |\hat{\beta}_j(\lambda_a)| < |\hat{\beta}_j(\lambda_s)|$$

due to correlation

Variance of the ridge estimator

$$\begin{aligned}\text{Var}(\hat{\beta}_{\text{ridge}}^{(s)}) &= \text{Var}(W_s \hat{\beta}_{LS}) \\ &= W_s \text{Var}(\hat{\beta}_{LS}) W_s^T \\ &= \sigma^2 W_s (X^T X)^{-1} W_s^T\end{aligned}$$

$$\begin{aligned}\text{Var}(\hat{\beta}_{LS}) - \text{Var}(\hat{\beta}_{\text{ridge}}^{(s)}) &= \sigma^2 \left[ (X^T X)^{-1} - W_s (X^T X)^{-1} W_s^T \right] \\ &= \sigma^2 W_s \left[ (\underline{I} + \lambda (X^T X)^{-1}) \underline{(X^T X)^{-1} (\underline{I} + \lambda (X^T X)^{-1})} - \underline{(X^T X)^{-1}} \right] W_s^T \\ &= \sigma^2 W_s \left[ \cancel{(X^T X)^{-1}} + \lambda (X^T X)^{-2} + \lambda (X^T X)^{-2} + \lambda^2 (X^T X)^{-3} - \cancel{(X^T X)^{-1}} \right] W_s^T \\ &\stackrel{!}{=} \sigma^2 W_s \left[ 2\lambda (X^T X)^{-2} + \lambda^2 (X^T X)^{-3} \right] W_s^T > 0 \\ \Rightarrow \text{Var}(\hat{\beta}_{\text{ridge}}^{(s)}) &\leq \text{Var}(\hat{\beta}_{LS})\end{aligned}$$

## Degrees of freedom

$$\hat{Y}_{LS} = \underbrace{\mathbf{X}(X^T X)^{-1} X^T g}_H \quad df = \text{tr}(H) = p$$

$$\hat{Y}_{RIDGE} = \underbrace{\mathbf{X}(X^T X + \lambda I)^{-1} X^T g}_{H_\lambda} \quad \begin{matrix} \text{effective} \\ df = \text{tr}(H_\lambda) \end{matrix}$$

$$\text{tr}(H_\lambda) = \sum_{j=1}^p \frac{d_j^{-2}}{d_j^{-2} + \lambda} \quad \begin{matrix} \lambda = 0 \rightarrow df = p \\ \lambda \rightarrow \infty \rightarrow df \rightarrow 0 \end{matrix}$$

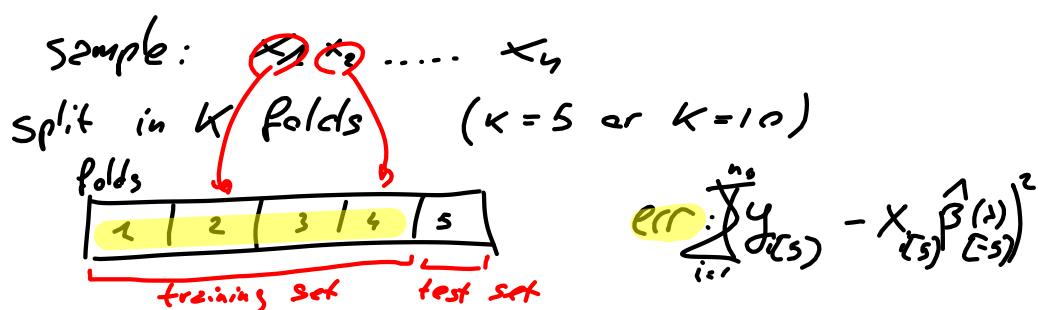
$$AIC : L(\beta) + 2J(\theta)$$

$$\text{least square} : -2\ell(\beta) + 2p$$

$$\text{ridge} : -2\ell(\beta) + 2 \sum \frac{d_j^{-2}}{d_j^{-2} + \lambda}$$

$AIC(\lambda)$  we can find  $\hat{\lambda}$  es  $\arg \min_{\lambda} AIC(\lambda)$

usually  $\lambda$  is computed by cross-validation



train the model on  $K-1$  folds  $\rightarrow \hat{\beta}_{(5)}$   
evaluate its performance on the remaining fold  $\text{err}_5$

- repeat the procedure for  $K$  times, always using a different fold as test set

$\hat{\beta}$  computed by using all observation not belonging to fold 5

$\text{err}_1$ : train set  $\{2, 3, 4, 5\}$   
test set  $\{1\}$

$\text{err}_2$ : train set  $\{1, 3, 4, 5\}$   
test set  $\{2\}$

:

$\text{err}_5$

$$\sum_{K=1}^5 \text{err}_K(\lambda) \quad \lambda \text{ that minimizes}$$

More about shrinkage

$$\underset{n \times p}{X} = UDV^T$$

$U = n \times n$  orthogonal matrix, whose columns span the column space of  $X$

$V = p \times n$  orthogonal matrix, whose columns span the row space of  $X$

$D = n \times n$  diagonal matrix,  $d_{ii}$  are singular values of  $X$ :

$$U^T U = I_n = VV^T$$

LS estimator  $\hat{\beta}_{LS}$

$$\begin{aligned}\hat{\beta}_{LS} &= (X^T X)^{-1} X^T y \\ &= (V D U^T U D V^T)^{-1} V D U^T y \\ &= (V D^2 V^T)^{-1} V D U^T y \\ &= V D^2 V^T V D U^T y \\ &= V D^2 D U^T y\end{aligned}$$

$$\begin{aligned}\hat{y}_{LS} &= X \hat{\beta}_{LS} \\ &= U D V^T V D^{-2} D U^T y \\ &= U D^2 D^{-2} U^T y \\ &= U U^T y\end{aligned}$$

$$\sum_{j=1}^p \frac{d_j}{d_j^2}$$

$$\begin{aligned}\hat{\beta}_{Ridge}^{(\lambda)} &= (X^T X + \lambda I_p)^{-1} X^T y \\ &= (V D U^T U D V^T + \lambda I_p)^{-1} V D U^T y \\ &= (V D^2 V^T + \lambda V V^T) V D U^T y \\ &= V (D^2 + \lambda I_p)^{-1} V^T V D U^T y \\ &= V (D^2 + \lambda I_p)^{-1} D U^T y\end{aligned}$$

$$\begin{aligned}\hat{y}_{Ridge} &= X \hat{\beta}_{Ridge} \\ &= U D V^T V (D^2 + \lambda I_p)^{-1} D U^T y \\ &= U D^2 (D^2 + \lambda I_p)^{-1} U^T y\end{aligned}$$

$$\sum_{j=1}^p \frac{d_j}{d_j^2 + \lambda}$$

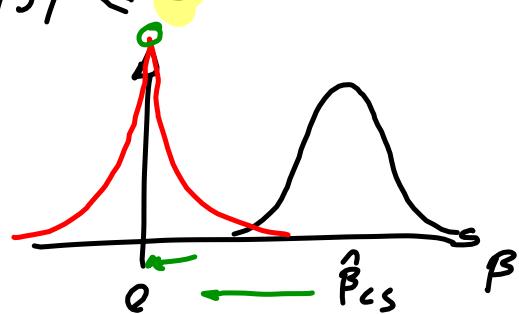
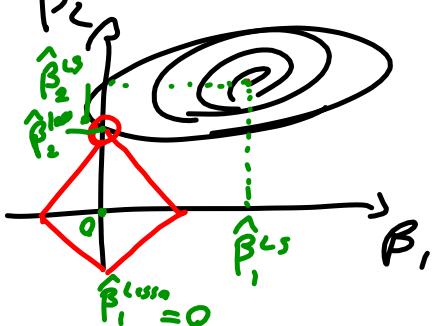
LASSO

$$\hat{\beta}_{\text{LASSO}} = \arg \min_{\beta} \left\{ \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}$$

or, equivalently

$$\hat{\beta}_{\text{LASSO}} = \arg \min_{\beta} \left\{ \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 \right\}$$

subject to  $\sum_{j=1}^p |\beta_j| \leq t$



$$\hat{\beta}_{\text{pen}} = \arg \min_{\beta} \left\{ \sum_i (y_i - \beta_0 - \sum_j x_{ij} \beta_j)^2 + \lambda \sum_j |\beta_j|^q \right\}$$

$q=1 \rightarrow \text{LASSO}$

$q=2 \rightarrow \text{RIDGE REGRESSION}$

$1 < q < 2 \approx \text{average between Lasso and ridge penalties}$

ELASTIC NET

$$\text{penalty : } \lambda \left( \alpha \sum_{j=1}^p \beta_j^2 + (1-\alpha) \sum_j |\beta_j| \right)$$



# STK4030 - Statistical Learning: Advanced Regression and Classification

Riccardo De Bin

[debin@math.uio.no](mailto:debin@math.uio.no)

Procedure	Logic
BE-only	Estimate full model on $x_1, \dots, x_k$ . Repeat: while the least significant term has $P \geq \alpha_2$ , remove it and re-estimate the model
BE	Estimate full model on $x_1, \dots, x_k$ . If the least significant term has $P \geq \alpha_2$ , remove it and re-estimate; otherwise stop. Again: if the least significant term has $P \geq \alpha_2$ , remove it and re-estimate; otherwise stop Repeat <ul style="list-style-type: none"><li>• if most significant excluded term has <math>P &lt; \alpha_1</math>, add it and re-estimate;</li><li>• if least significant included term has <math>P \geq \alpha_2</math>, remove it and re-estimate;</li></ul> until neither action is possible.
FS-only	Estimate null model. Repeat: while the most significant excluded term has $P < \alpha_1$ , add it and re-estimate.
FS	Estimate null model. If the most significant excluded term has $P < \alpha_1$ , add it and re-estimate; otherwise stop. Again: if the most significant excluded term has $P < \alpha_1$ , add it and re-estimate; otherwise stop. Repeat <ul style="list-style-type: none"><li>• if least significant included term has <math>P \geq \alpha_2</math>, remove it and re-estimate;</li><li>• if most significant included term has <math>P &lt; \alpha_1</math>, add it and re-estimate;</li></ul> until neither action is possible.

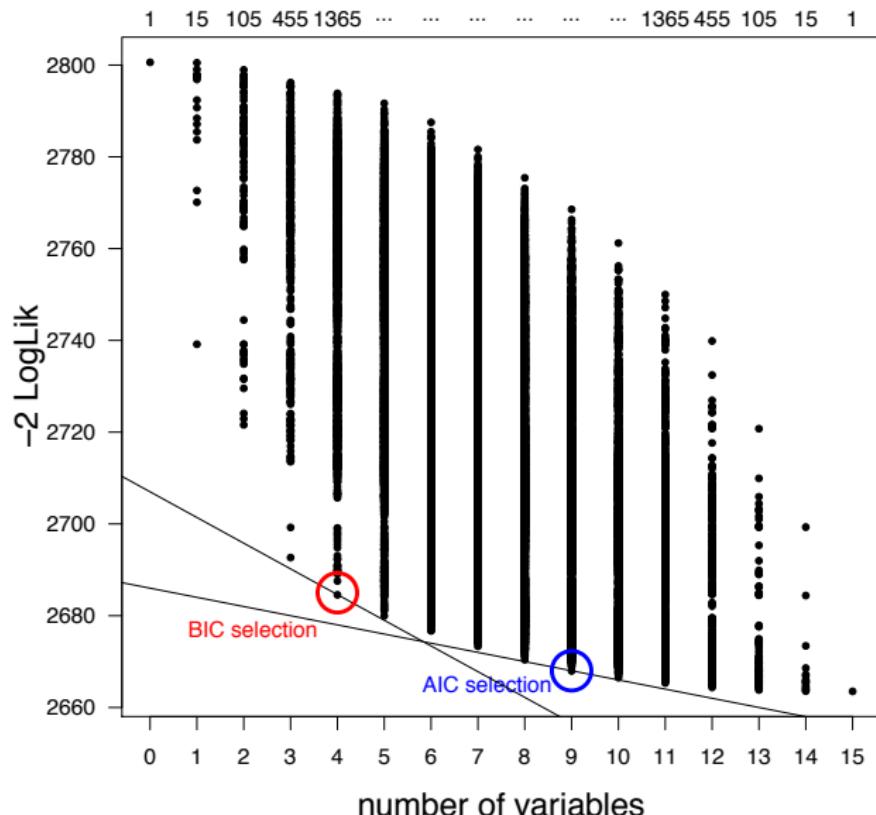
from: Royston & Sauerbrei (2008)

**Table 2.2** Myeloma study (65 patients, 48 events). Results of applying variable selection strategies.<sup>a</sup>

Variable	Nominal significance level, $\alpha$			All-subsets			
	0.05			0.157			AIC
	Full	BE	FS	Full	BE	FS	
$x_1$	*	✓	✓	*	✓	✓	✓
$x_2$			✓			✓	
$x_3$	*	✓		*	✓		✓
$x_4$		✓		*	✓		✓
$x_5$							
$x_6$		✓		*	✓		✓
$x_7$	*	✓		*	✓		✓
$x_8$				*	✓		✓
$x_9$							
$x_{10}$							
$x_{11}$							
$x_{12}$	*	✓		*	✓		✓
$x_{13}$	*	✓		*	✓		✓
$x_{14}$				*			
$x_{15}$							
$x_{16}$							

<sup>a</sup> BE(0.01) and FS(0.01) selected only one variable,  $x_1$ ; \* denotes that a variable is significant at the relevant  $\alpha$ -level in the full model.

from:  
Royston &  
Sauerbrei  
(2008)



**Table 2.3** Educational body-fat data. Full model and that selected by BE(0.05). Final three columns give details of the full model excluding  $x_6$ .

Variable	Full model			BE(0.05)			Full model excl. $x_6$		
	$\hat{\beta}$	SE	$\hat{\beta}/SE$	$\hat{\beta}$	SE	$\hat{\beta}/SE$	$\hat{\beta}$	SE	$\hat{\beta}/SE$
$x_1$	0.074	0.032	2.31	0.056	0.024	2.35	0.211	0.034	6.20
$x_2$	-0.019	0.067	-0.28				0.227	0.074	3.08
$x_3$	-0.249	0.191	-1.30	-0.322	0.121	-2.65	-0.915	0.212	-4.32
$x_4$	-0.394	0.234	-1.68				-0.378	0.278	-1.36
$x_5$	-0.119	0.108	-1.10				0.150	0.124	1.21
$x_6$	0.901	0.091	9.90	0.774	0.033	23.26	-	-	-
$x_7$	-0.146	0.144	-1.02				0.163	0.166	0.98
$x_8$	0.178	0.146	1.22				0.231	0.173	1.33
$x_9$	-0.041	0.245	-0.17				-0.095	0.291	-0.33
$x_{10}$	0.185	0.220	0.85				-0.053	0.259	-0.21
$x_{11}$	0.178	0.170	1.04				-0.066	0.200	-0.33
$x_{12}$	0.277	0.207	1.34				0.058	0.244	0.24
$x_{13}$	-1.830	0.529	-3.46	-1.943	0.406	-4.78	-2.692	0.620	-4.34

from: Royston & Sauerbrei (2008)

```
library(BioconductorManager)
library(breastCancerVDX)

# ids of genes FLOT1
idFLOT1 <- which(fData(vdx)[,5] == 10211)

# ids of ERBB2
idERBB2 <- which(fData(vdx)[,5] == 2064)

# get expression levels of probes mapping to FLOT genes
X <- t(exprs(vdx)[idFLOT1,])
X <- sweep(X, 2, colMeans(X))

# get expression levels of probes mapping to ERBB2 genes
Y <- t(exprs(vdx)[idERBB2,])
Y <- sweep(Y, 2, colMeans(Y))

# regression analysis
summary(lm(formula = Y[,1] ~ X[,1] + X[,2] + X[,3] + X[,4]))
```

from: Van der Wieringen (2015)

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.0000	0.0633	0.0000	1.0000
X[, 1]	0.1641	0.0616	2.6637	0.0081 **
X[, 2]	0.3203	0.3773	0.8490	0.3965
X[, 3]	0.0393	0.2974	0.1321	0.8949
X[, 4]	0.1117	0.0773	1.4444	0.1496

---

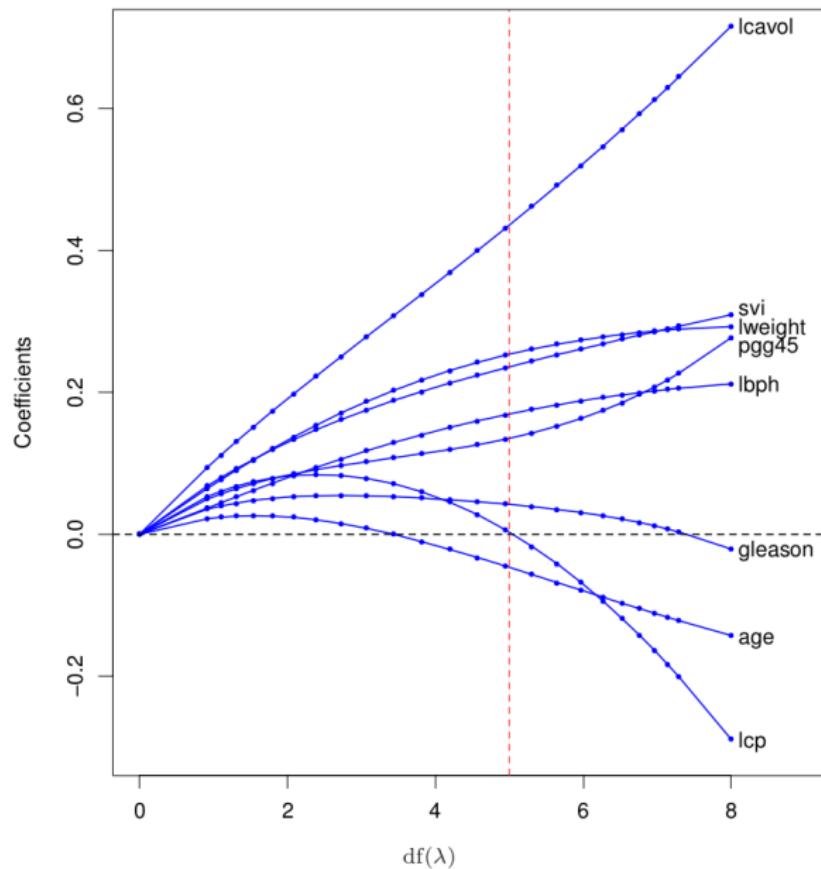
Signif. codes: 0 \*\*\* 0.001 \*\* 0.01 \* 0.05 . 0.1 1

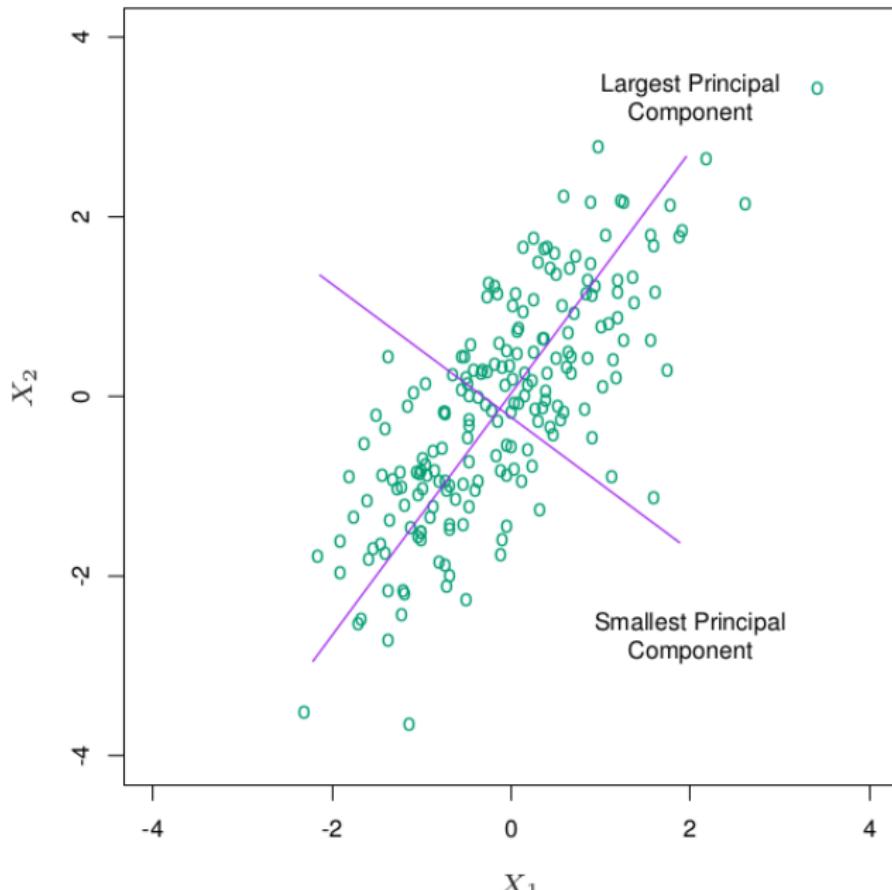
Residual standard error: 1.175 on 339 degrees of freedom

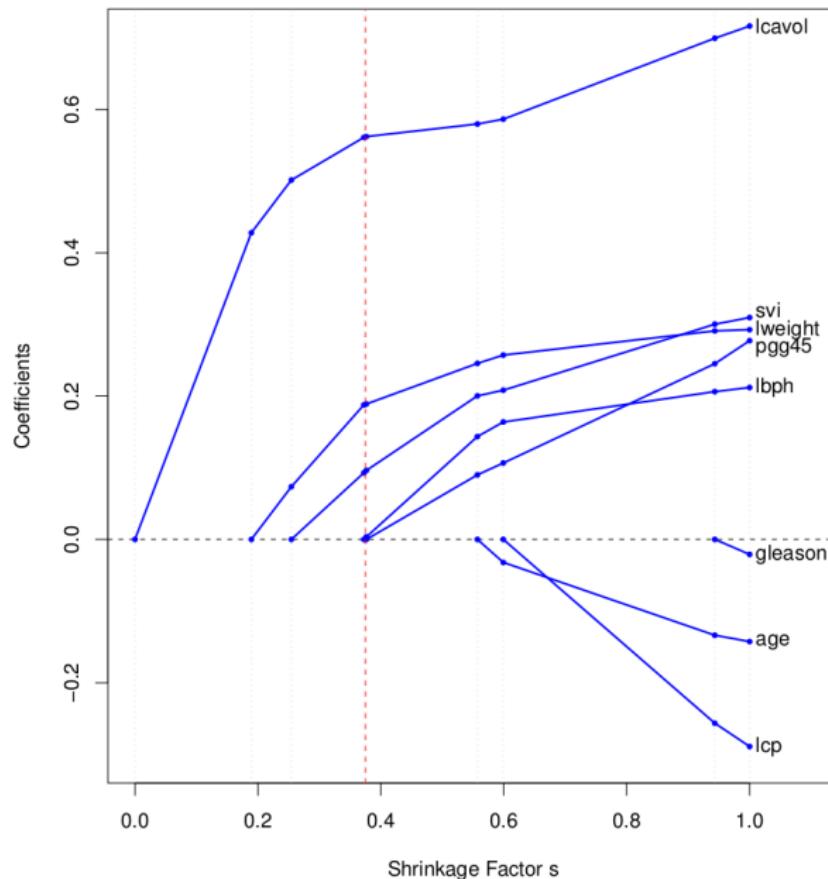
Multiple R-squared: 0.04834, Adjusted R-squared: 0.03711

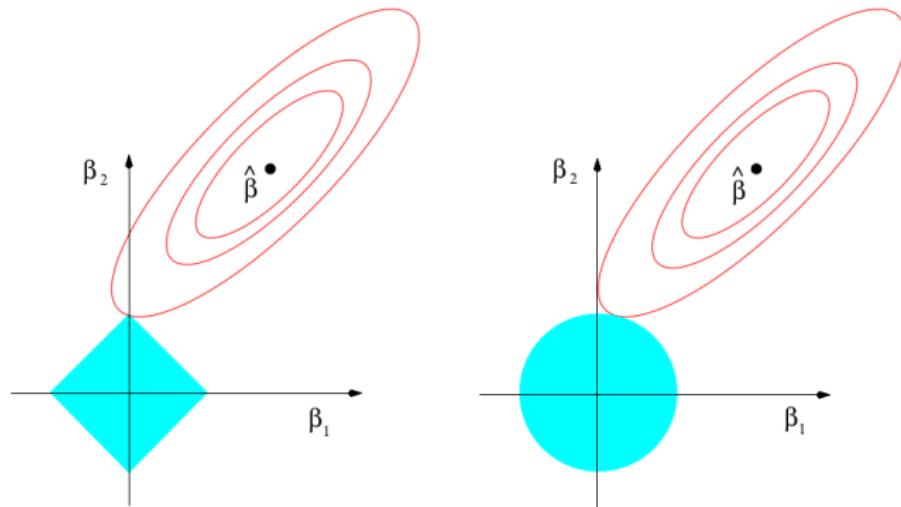
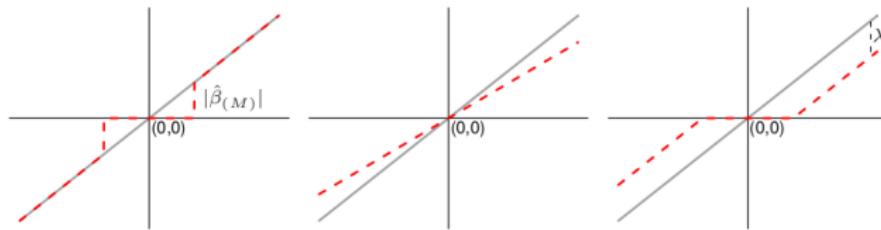
F-statistic: 4.305 on 4 and 339 DF, p-value: 0.002072

from: Van der Wieringen (2015)









Exercise 3.6

$$P(\beta | y) \propto p(y | \beta) p(\beta)$$

where

$$p(y | \beta) \hookrightarrow N(X\beta, \sigma^2 I)$$

$$p(\beta) \hookrightarrow N(0, \tau^2 I)$$

$$\frac{1}{\sqrt{\beta^\top \beta}} \text{ const}$$

$$P(\beta | y) \propto \text{const.} \exp \left\{ -\frac{1}{2\sigma^2} (y - X\beta)^\top (y - X\beta) \right\} \exp \left\{ -\frac{1}{2\tau^2} \beta^\top \beta \right\}$$

$\propto \exp \left\{ (y - X\beta)^\top (y - X\beta) + \frac{\sigma^2}{\tau^2} \beta^\top \beta \right\}$

Exercise 3.10

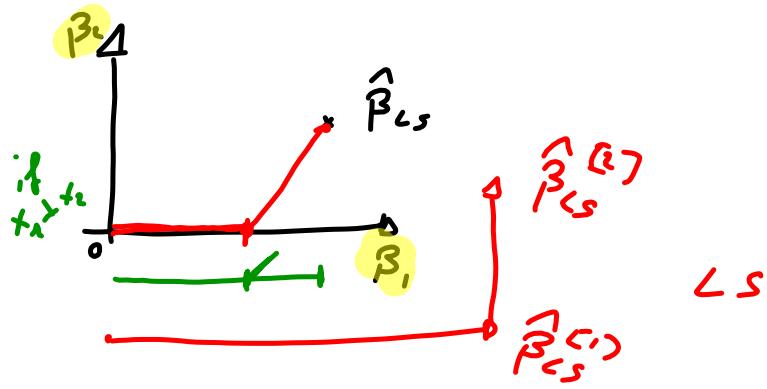
$$F = \frac{(RSS_0 - RSS_1) / (\underbrace{p_1 - p_0}_{=1})}{RSS_1 / (n - p_1 - 1)}$$

→ find the  $\beta_j$  s.t.  $\beta_j = 0$  lead to the smallest  $RSS_0 - RSS_1$

$$\text{we know (ex 3.1)} , F_{1, n-p_1-1} \stackrel{d}{=} z_j^2$$

⇒ the  $\beta_j$  which, when set equal to 0, increases the least the RSS is that with the smallest  $z_j^2$ .

# Least angle regression



we update  $\hat{\beta}_1$  until  
 $\text{cor}(x_1, r) < \text{cor}(x_2, r)$   
 then  
 we update both  $(\hat{\beta}_1, \hat{\beta}_2)$

$$r = y - \bar{y} - \sum_{j=1}^p x_j \hat{\beta}_j$$

- start with  $r = y - \bar{y}$
- check the largest correlation between  $x_j$  and  $r$
- update the regression coefficient of  $x_j$  ( $\hat{\beta}_j$ ) until  $\langle x_j, r \rangle$

- first we update  $\hat{\beta}_1$

Suppose that  $\hat{\beta}_j$  are ordered by importance, i.e., effect of  $x_i$  is larger than effect of  $x_j$ ,  $i < j$

- we reach a point where we add also  $\hat{\beta}_2$  to our solution
- we are updating  $\{\hat{\beta}_1, \hat{\beta}_2\}$

- we reach a point where the update is related to  $\{\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3\}$

:

once a coefficient enters in the set of active regression coefficients, it stays

## FORWARD REGRESSION

1<sup>st</sup> step  $\hat{y} = \hat{\beta}_0 + \hat{\beta}_{LS}^{(1)} x_1$

## LAR

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1^{(1)} x_1$$

- Lasso can be seen as a special case of a modification of LAR, for which once the estimate of a regression coeff. reach 0, it is excluded by the set of active regression coefficient

## METHODS USING DERIVED INPUT DIRECTIONS

- Principal component regression
- Partial least squares

### Principal component regression

IDEA: inputs have different variabilities in different directions  
 → directions with largest variability provide more information

↓ principal components

linear combinations of  $X$  based on directions of largest variability

$$z_m = X v_m$$

$v_m$  = eigenvectors  
 $z_1$  = direction with the largest variability

$z_2$  = " " the 2nd largest variability s.t.  $e_1 \perp e_2$

:

$z_p$  = " " the least variability  $e_p \perp z_1, \dots, z_{p-1}$

### Model

$$y = \theta_0 + \sum_{m=1}^M \theta_m z_m + \varepsilon$$

$$\hat{\theta}_0 = \bar{y}$$

$$\hat{\theta}_m = \frac{\langle z_m, y \rangle}{\langle z_m, z_m \rangle}$$

### NOTE

principal component analysis is scale dependent



IMPORTANT to first standardize  $X$

for each  $x_j$

$$x_{ij}^{st} = \frac{(x_{ij} - \bar{x}_j)}{sd(x_j)}$$

$$\begin{aligned}
 \hat{y} &= \hat{\theta}_0 + \sum_{m=1}^M \hat{\theta}_m z_m & z_m &= X v_m \\
 &= \hat{\theta}_0 + \sum_{m=1}^M \hat{\theta}_m X v_m \\
 &= \hat{\theta}_0 + X \sum_{m=1}^M \hat{\theta}_m v_m \\
 &= \hat{\theta}_0 + X \hat{\beta}_{PCR}
 \end{aligned}$$

- idea: use  $M < p$  principal components, to exclude directions with less information

if  $M = p$ , principal component regression provides the least square estimates, because all  $p$  principal component span the space of  $X$

we obtain results similar to ridge regression

$$\hat{\beta}_{\text{ridge}} = \sum_{j=1}^p u_j \frac{d_j^2}{d_j^2 + \lambda} u_j^T y \quad \hat{\beta}_{LS} = \sum_{j=1}^p u_j u_j^T y$$

$$\hat{\beta}_{PCR} = \sum_{j=1}^M u_j \underbrace{\mathbb{1}_{[j \leq M]}}_{\text{red box}} u_j^T y = \sum_{j=1}^M u_j u_j^T y$$

## Partial least squares

- in the construction of principal components, we do not take into account  $y$

↓  
- in the construction of derived input directions, we also consider  $y$

as for principal component regression it is important to first standardize  $X$

1<sup>st</sup> step:  $\hat{\varphi}_{1j} = \frac{\langle x_j, y \rangle}{\langle x_j, x_j \rangle}$  →  $\underline{z}_1 = \sum_{j=1}^p \hat{\varphi}_{1j} x_j$   
 $\hat{\theta}_1 = \frac{\langle \underline{z}_1, y \rangle}{\langle \underline{z}_1, \underline{z}_1 \rangle}$

$$\underline{x}_j^{(2)} = x_j - \frac{\langle \underline{z}_1, x_j \rangle}{\langle \underline{z}_1, \underline{z}_1 \rangle} \quad j = 1, \dots, p$$

$$\hat{\varphi}_{2j} = \frac{\langle \underline{x}_j^{(2)}, y \rangle}{\langle \underline{x}_j^{(2)}, \underline{x}_j^{(2)} \rangle}$$

$$\underline{z}_2 = \sum_{j=1}^p \hat{\varphi}_{2j} \underline{x}_j^{(2)}$$

$$\hat{\theta}_2 = \frac{\langle \underline{z}_2, y \rangle}{\langle \underline{z}_2, \underline{z}_2 \rangle}$$

### Differences:

PCR: derived input directions are <sup>the</sup> principal components of  $X$ , constructed by looking at the variability within  $X$

PLS: directions take into consideration both the variability and the correlation with  $y$

### Mathematically

PCR  $\max_{\alpha} \text{Var}(X\alpha) \quad \text{s.t. } \|\alpha\| = 1$

$$\alpha^T S V e = 0 \quad l = 1, \dots, m$$

sample covariance matrix

PLS  $\max_{\alpha} \text{Cor}^2(y, X\alpha) \text{Var}(X\alpha)$

uncorrelated directions

In practice, the s.t.  $\|\alpha\| = 1$

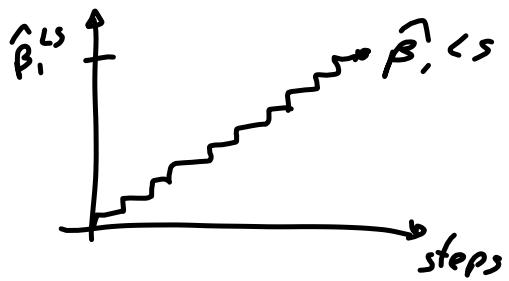
variance component

tends to dominate the

correlation one → similar results!

$$\alpha^T S \hat{\varphi}_l = 0 \quad l = 1, \dots, m-1$$

## Incremental forward stagewise regression



similar to LAR, but each update involves only one parameter each time

$$\hat{\beta}_j^{(m)} = \hat{\beta}_j^{(m-1)} + \underbrace{\delta}_{\text{step}}$$

→ we will go back when we will talk about boosting

### → grouped LASSO

different construction of penalty to take into account structures in the data

- dummy variables related to the same categorical variable
- working with genetic data, group genes belonging to the same pathway

$$\hat{\beta}_{gL} = \arg \min_{\beta} \left\{ \sum_{i=1}^n (y_i - \hat{\beta}_0 - \sum_{e=1}^E x_e \beta_e) + \lambda \sum_{e=1}^E \sqrt{p_e} \|\beta_e\|_2 \right\}$$

group size

$\|\cdot\|_2$  = Euclidean distance

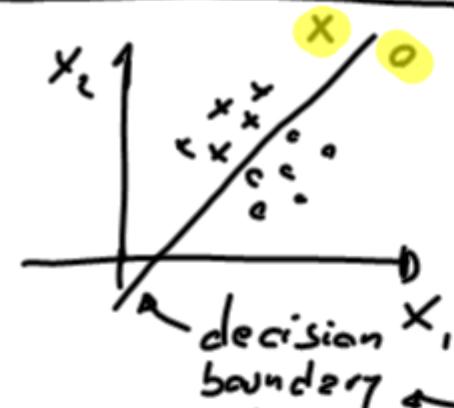
$\|\cdot\|_2 = 0 \Leftrightarrow \text{all components are } 0$

- adaptive LASSO
  - relaxed LASSO
- } → 2-step procedure

- SCAD                   $L(\beta) + J(\beta)$

$$\frac{d J(\beta)}{d \beta} = \lambda \operatorname{sign}(\beta) \left[ \mathbb{1}[|\beta| \leq \lambda] + \frac{(\alpha^2 - |\beta|)}{(\alpha - 1)\lambda} \mathbb{1}[|\beta| > \lambda] \right]$$

# Linear models for classification



until now: classification based on linear regression

$$\forall k \quad \hat{f}_k(x) = \hat{\beta}_{k0} + \hat{\beta}_k^T x$$

decision boundary between two classes  $k$  and  $\ell$

$$x: \hat{f}_k(x) = \hat{f}_\ell(x)$$

discriminant functions  $\delta_k(x)$

it is  
 a member of  
 a class of  
 methods, namely  
 methods based on  
 discriminant functions

$$f_A(x) = \Pr[G=A \mid X=x]$$

$$f_B(x) = \Pr[G=B \mid X=x]$$

$$f_C(x) = \Pr[G=C \mid X=x]$$

posterior probabilities

If  $\delta_k(x)$  or  $\Pr[G=k \mid X=x]$  are linear in  $x \rightarrow$  linear decision boundaries

Actually, we only need monotone transformation of  $\delta_k(x)$  or  $\Pr[G=k | X=x]$  to be linear

Example

$$(i) \hat{\delta}_k(x) = \hat{\beta}_{k0} + \hat{\beta}_{k1}x_1 + \hat{\beta}_{k2}x_2 + \hat{\beta}_{k3}\underline{x_1} + \hat{\beta}_{k4}\underline{x_2}$$

the relationship is linear in the augmented space, but the decision boundaries are quadratic in the original space

(ii) when there are two classes

$$\Pr[G=1 | X=x] = \frac{\exp(\beta_0 + \beta^T x)}{1 + \exp(\beta_0 + \beta^T x)}$$

logit transformation

$$\Pr[G=2 | X=x] = \frac{1}{1 + \exp(\beta_0 + \beta^T x)}$$

$$\log \frac{P}{1-P}$$

$$\log \frac{\Pr[G=1 | X=x]}{\Pr[G=2 | X=x]} = \beta_0 + \beta^T x$$

Linear regression of an Indicator Matrix

- codify each of the classes 1, ..., K with an indicator variable

$$\text{ex. } x^{3 \times 3} \Rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = Y \quad \begin{array}{l} 1^{\text{st}} \text{ obs. is of class 1} \\ 2^{\text{nd}} \text{ and } 3^{\text{rd}} \text{ .. 2} \\ 4^{\text{th}} \text{ .. 5}^{\text{th}} \text{ .. 3} \end{array}$$

$$\rightarrow \text{linear regression: } \hat{Y} = X (X^T X)^{-1} X^T Y$$

$$\text{new observation } x_{\text{new}} \quad \hat{y} = \hat{f}(x_{\text{new}}) = \underbrace{\exp}_{1 \times p} \underbrace{\hat{\beta}}_{p \times K}$$

$$\left( \hat{f}_1(x_{\text{new}}) \quad \hat{f}_2(x_{\text{new}}) \quad \hat{f}_3(x_{\text{new}}) \right)$$

$$\hat{G}(x_{\text{new}}) = \operatorname{argmax}_K \hat{f}_K(x)$$

Why does it work

$$E[Y_k | X=x] = \Pr[G=k | X=x]$$

Are  $\hat{f}_k(x)$  reasonable estimates of  $\Pr[G=k | X=x]$ ?

Yes and no

→ if intercept is included  $\sum_{k=1}^K \hat{f}_k(x) = 1$

→  $\hat{f}_k(x)$  can be  $<0$  or  $>1$  → problems happen when  
the new observation is  
many times it works → outside the training hull,  
despite this issue due to the rigidity of  
linear regression

A bigger problem is the so called masking effect

- only if  $K \geq 3$

As we see in Fig 4.3, when  $K=3$ , it is sufficient to have a quadratic rule.

More generally, if we have  $K$  classes, we need a curve of degree  $K-1$

# LINEAR DISCRIMINANT ANALYSIS

- From the decision theory (Section 2.4) we know that for optimal classification we need to know the class posterior  $\Pr(G=k | X=x)$

Suppose:

- $f_k(x)$  is the density of  $x$  conditional to class  $G=k$   
 $\Pr[X=x | G=k]$
- $\pi_k(x)$  is the prior probability to be in class  $k$ ,  $\Pr[G=k]$

Then

$$\begin{aligned} \Pr[G=k | X=x] &= \frac{\Pr[X=x | G=k] \Pr[G=k]}{\Pr[X=x]} = \sum \Pr[X=x | G=k] \Pr[G=k] \\ &= \frac{f_k(x) \pi_k(x)}{\sum_{e=1}^E f_e(x) \pi_e(x)} \end{aligned}$$

We can choose  $f_k(x)$  and  $\pi_k(x)$  as we prefer...

- when  $f_k(x)$  is from a multivariate Gaussian distribution, then Linear Discriminant Analysis (LDA)  
 Quadratic Discriminant Analysis (QDA)

$$f_k(x) = \frac{1}{(2\pi)^{D/2} |\Sigma_k|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) \right\}$$

In particular, for LDA we suppose  $\Sigma_k = \sum f_k$

We can then compare two classes by the log-ratio

$$\log \frac{\Pr[G=k | X=x]}{\Pr[G=0 | X=x]} = \log \frac{\frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu_k)^T \Sigma^{-1} (x - \mu_k) \right\} \hat{h}_k / D}{\frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu_0)^T \Sigma^{-1} (x - \mu_0) \right\} \hat{h}_0 / D}$$

$$= \log \frac{\hat{h}_k}{\hat{h}_0} - \frac{1}{2} \left( x^T \Sigma^{-1} x - 2x^T \Sigma^{-1} \mu_k + \mu_k^T \Sigma^{-1} \mu_k - x^T \Sigma^{-1} x + 2x^T \Sigma^{-1} \mu_0 - \mu_0^T \Sigma^{-1} \mu_0 \right)$$

$$= \log \frac{\hat{h}_k}{\hat{h}_0} - \frac{1}{2} (\mu_k + \mu_0)^T \Sigma^{-1} (\mu_k - \mu_0) + x^T \Sigma^{-1} (\mu_k - \mu_0)$$

Note

- the decision boundaries are NOT the perpendicular bisectors of the segments joining the centroids (happens only if  $\Sigma = \sigma^2 I$ )
- the linear discriminant function  $\delta_k(x)$  is

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \hat{h}_k$$

$$\rightarrow \hat{G}(x) = \arg \max_k \delta_k(x)$$

Since we do not know the values of the parameters  $\hat{h}_k, \mu_k, \Sigma$ , we need to estimate them

$$\cdot \hat{h}_k = \frac{N_k}{N} = \frac{\# \text{ observations in class } k}{\text{total } \# \text{ of observations}}$$

$$\cdot \hat{\mu}_k = \sum_{g_i=k} x_i / N_k$$

$$\cdot \hat{\Sigma} = \sum_{k=1}^K \sum_{g_i=k} (x_i - \hat{\mu}_k) (x_i - \hat{\mu}_k)^T / (N - K)$$

With two classes, there is a simple correspondence between LDA and classification by linear regression ( $\rightarrow$  Ex. 4.2)

With  $K > 2$ , there are substantial differences

LDA does NOT suffer from the masking effect.

## Exercise 4.2

$x \in \mathbb{R}^p$   $\gamma$  has 2 classes

size  $N_1$  size  $N_2$

target code:  $\frac{N}{N_1}$  target code:  $\frac{N}{N_2}$

(a) LDA classifies to class  $\ell$  if  $\delta_\ell(x) > \delta_i(x)$

$$x^\top \underline{\Sigma^{-1} \hat{\mu}_2} - \frac{1}{2} \underline{\hat{\mu}_2^\top \Sigma^{-1} \hat{\mu}_2} + \underline{\log \frac{N_2}{N}} >$$

$$x^\top \underline{\Sigma^{-1} \hat{\mu}_1} - \frac{1}{2} \underline{\hat{\mu}_1^\top \Sigma^{-1} \hat{\mu}_1} + \underline{\log \frac{N_1}{N}}$$

$$x^\top \underline{\Sigma^{-1} (\hat{\mu}_2 - \hat{\mu}_1)} > \frac{1}{2} (\hat{\mu}_2 + \hat{\mu}_1)^\top \underline{\Sigma^{-1} (\hat{\mu}_2 - \hat{\mu}_1)} - \underline{\log \frac{N_2}{N_1}}$$

(b)  $\min \sum_{i=1}^n (y_i - \beta_0 - x_i^\top \beta)^2$

$\gamma$  = target vector, target code  $\frac{N}{N_1}, \frac{N}{N_2}$

$$\underline{\gamma = t_1 U_1 + t_2 U_2} \quad t_1 = -\frac{N}{N_1}, t_2 = \frac{N}{N_2}$$

$$\underline{1 = U_1 + U_2}$$

$$\begin{matrix} U_1 & U_2 \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{matrix}$$

Note that  $\hat{\mu}_i = \frac{x^\top U_i}{N_i}$   $\Rightarrow \underline{x^\top U_1 = N_1 \hat{\mu}_1}$   
 $\underline{x^\top U_2 = N_2 \hat{\mu}_2}$

Therefore  $\underline{x^\top \gamma} = x^\top (t_1 U_1 + t_2 U_2) = \underline{t_1 N_1 \hat{\mu}_1 + t_2 N_2 \hat{\mu}_2}$

Our goal is to minimize  $(Y - \beta_0 1 - X\beta)^T (Y - \beta_0 1 - X\beta)$

$$\frac{\partial RSS}{\partial \beta_0} \quad \left\{ \begin{array}{l} - (Y - \beta_0 1 - X\beta)^T 1 - 1^T (Y - \beta_0 1 - X\beta) = 0 \\ - (Y - \beta_0 1 - X\beta)^T X - X^T (Y - \beta_0 1 - X\beta) = 0 \end{array} \right.$$

$$\rightarrow \left\{ \begin{array}{l} 2\beta_0 1^T 1 - 2 1^T (Y - X\beta) = 0 \\ 2X^T X\beta - 2X^T Y + 2\beta_0 X^T 1 = 0 \end{array} \right.$$

$$\beta_0 = \frac{1}{N} 1^T (Y - X\beta)$$

$$\rightarrow X^T X \beta - X^T Y + \frac{1}{N} X^T 1 1^T Y - \frac{1}{N} X^T 1 1^T X \beta = 0$$

$$\underbrace{(X^T X - \frac{1}{N} X^T 1 1^T X)}_{LHS} \beta = \underbrace{X^T Y - \frac{1}{N} X^T 1 1^T Y}_{RHS}$$

$$RHS = t_1 N_1 \hat{\mu}_1 + t_2 N_2 \hat{\mu}_2 - \frac{1}{N} (X^T (U_1 + U_2) (U_1 + U_2)^T Y)$$

$$= t_1 N_1 \hat{\mu}_1 + t_2 N_2 \hat{\mu}_2 - \frac{1}{N} (X^T U_1 + X^T U_2) (U_1 + U_2)^T (t_1 U_1 + t_2 U_2)$$

$$= t_1 N_1 \hat{\mu}_1 + t_2 N_2 \hat{\mu}_2 - \frac{1}{N} (N_1 \hat{\mu}_1 + N_2 \hat{\mu}_2) (t_1 U_1^T U_1 + t_1 U_1^T U_2 + t_2 U_2^T U_1 + t_2 U_2^T U_2)$$

$$= t_1 N_1 \hat{\mu}_1 + t_2 N_2 \hat{\mu}_2 - \frac{1}{N} (N_1 \hat{\mu}_1 + N_2 \hat{\mu}_2) (t_1 N_1 + t_2 N_2)$$

$$= t_1 N_1 \hat{\mu}_1 + t_2 N_2 \hat{\mu}_2 - \frac{N_1^2}{N} t_1 \hat{\mu}_1 - \frac{N_1 N_2}{N} \hat{\mu}_1 t_2 - \frac{N_1 N_2}{N} \hat{\mu}_2 t_1 - \frac{N_2^2}{N} t_2 \hat{\mu}_2$$

$$= t_1 \hat{\mu}_1 \left( N_1 - \frac{N_1^2}{N} \right) + t_2 \hat{\mu}_2 \left( N_2 - \frac{N_2^2}{N} \right) - \frac{N_1 N_2}{N} \hat{\mu}_1 t_2 - \frac{N_1 N_2}{N} \hat{\mu}_2 t_1$$

$$= t_1 \hat{\mu}_1 \left( \frac{N_1(N_1 + N_2) - N_1^2}{N} \right) + t_2 \hat{\mu}_2 \left( \frac{N_2(N_1 + N_2) - N_2^2}{N} \right) - \frac{N_1 N_2}{N} \hat{\mu}_1 t_2 - \frac{N_1 N_2}{N} \hat{\mu}_2 t_1$$

$$= \frac{N_1 N_2}{N} (t_1 - t_2) (\hat{\mu}_1 - \hat{\mu}_2)$$

$$LHS = \underline{\underline{X^T X}} - \frac{1}{N} \underline{\underline{X^T 1 1^T X}}$$

$$\Sigma = \sum_{k=1}^K \sum_{g_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T / (N-2)$$

$$(N-2)\Sigma = X^T X - N_1 \hat{\mu}_1 \hat{\mu}_1^T - N_2 \hat{\mu}_2 \hat{\mu}_2^T$$

$$\text{so } \underline{\underline{X^T X}} = (N-2)\Sigma + N_1 \hat{\mu}_1 \hat{\mu}_1^T + N_2 \hat{\mu}_2 \hat{\mu}_2^T$$

$$\frac{1}{N} \underline{\underline{X^T 1 1^T X}} = \frac{1}{N} \underline{\underline{X^T (U_1 + U_2) (U_1 + U_2)^T X}}$$

$$= \frac{1}{N} (N_1 \hat{\mu}_1 + N_2 \hat{\mu}_2) (N_1 \hat{\mu}_1 + N_2 \hat{\mu}_2)^T$$

$$= \frac{N_1^2 \hat{\mu}_1 \hat{\mu}_1^T}{N} + \frac{N_1 N_2}{N} \hat{\mu}_1 \hat{\mu}_2^T + \frac{N_1 N_2}{N} \hat{\mu}_2 \hat{\mu}_1^T + \frac{N_2^2}{N} \hat{\mu}_2 \hat{\mu}_2^T$$

$$= (N-2)\Sigma + N_1 \hat{\mu}_1 \hat{\mu}_1^T + N_2 \hat{\mu}_2 \hat{\mu}_2^T = \frac{N_1^2}{N} \hat{\mu}_1 \hat{\mu}_1^T - \frac{N_1 N_2}{N} \hat{\mu}_1 \hat{\mu}_2^T - \frac{N_1 N_2}{N} \hat{\mu}_2 \hat{\mu}_1^T - \frac{N_2^2}{N} \hat{\mu}_2 \hat{\mu}_2^T$$

$$= (N-2)\Sigma + \left( \frac{N_1(N_1+t_2) - t_1^2}{N} \right) \hat{\mu}_1 \hat{\mu}_1^T + \left( \frac{N_2(N_2+t_1) - t_2^2}{N} \right) \hat{\mu}_2 \hat{\mu}_2^T + \\ - \frac{N_1 N_2}{N} \hat{\mu}_1 \hat{\mu}_2^T - \frac{N_1 N_2}{N} \hat{\mu}_2 \hat{\mu}_1^T$$

$$= (N-2)\Sigma + \frac{N_1 N_2}{N} (\hat{\mu}_2 - \hat{\mu}_1)^T (\hat{\mu}_2 - \hat{\mu}_1) \quad \Sigma_B = \frac{N_1 N_2}{N^2} (\hat{\mu}_2 - \hat{\mu}_1)^T (\hat{\mu}_2 - \hat{\mu}_1)$$

$$= (N-2)\Sigma + N \Sigma_B$$

$$[(N-2)\Sigma + N \Sigma_B] \beta = \frac{N_1 N_2}{N} (t_1 - t_2) (\hat{\mu}_1 - \hat{\mu}_2)$$

$$= \frac{N_1 N_2}{N} \left( -\frac{N}{N_1} - \frac{N}{N_2} \right) (\hat{\mu}_1 - \hat{\mu}_2)$$

$$= \frac{N_1 N_2}{N} \left( -\frac{N(N_1 + N_2)}{N_1 N_2} \right) (\hat{\mu}_1 - \hat{\mu}_2)$$

$$= N (\hat{\mu}_2 - \hat{\mu}_1)$$

(c) Note that  $\sum \beta = (\hat{\mu}_2 - \hat{\mu}_1) \underbrace{\frac{N_1 N_2}{N^2} (\hat{\mu}_2 - \hat{\mu}_1)^T \beta}_{\text{Scalar}}$ , it goes in the direction of  $\sqrt{ }$

Both the LHS and the RHS of

$$[(N-2)\Sigma + N\Sigma_B] \beta = N(\hat{\mu}_2 - \hat{\mu}_1) \text{ go in the direction of } (\hat{\mu}_2 - \hat{\mu}_1), \text{ so}$$

The solution must be proportional to  $\Sigma^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$

$$\lambda \Sigma^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$$

(d)  $t_1$  and  $t_2$  were chosen arbitrarily, so the result holds

$$(e) \hat{\beta}_0 = \frac{1}{N} \underline{1}^T (\underline{Y} - \underline{X}\hat{\beta})$$

$$= \frac{1}{N} (\underline{U}_1 + \underline{U}_2)^T (t_1 \underline{U}_1 + t_2 \underline{U}_2) - \frac{1}{N} (\underline{U}_1 + \underline{U}_2)^T \underline{X} \hat{\beta}$$

$$= \frac{1}{N} (t_1 \underline{U}_1^T \underline{U}_1 + t_2 \underline{U}_2^T \underline{U}_2 + t_1 \underline{U}_2^T \underline{U}_1 + t_2 \underline{U}_1^T \underline{U}_2) - \frac{1}{N} (\underline{U}_1^T \underline{X} + \underline{U}_2^T \underline{X}) \hat{\beta}$$

$$= \frac{1}{N} \left( -\frac{N_1}{N} \cdot N_1 + \frac{N_2}{N} \cdot N_2 \right) - \frac{1}{N} (N_1 \hat{\mu}_1 + N_2 \hat{\mu}_2) \hat{\beta}$$

$$= -\frac{1}{N} (N_1 \hat{\mu}_1 + N_2 \hat{\mu}_2) \hat{\beta}$$

$$\begin{aligned} \hat{f}(x) &= \hat{\beta}_0 + x^T \hat{\beta} \\ &= -\frac{1}{N} (N_1 \hat{\mu}_1 + N_2 \hat{\mu}_2) \hat{\beta} + x^T \hat{\beta} \\ &= \frac{1}{N} (Nx^T - N_1 \hat{\mu}_1 - N_2 \hat{\mu}_2) \hat{\beta} \\ &= \frac{1}{N} (Nx^T - N_1 \hat{\mu}_1 - N_2 \hat{\mu}_2) \lambda \Sigma^{-1} (\hat{\mu}_2 - \hat{\mu}_1) \quad \lambda \in \mathbb{R} \end{aligned}$$

$$\begin{aligned} \hat{f}(x) > 0 \iff Nx^T \Sigma^{-1} (\hat{\mu}_2 - \hat{\mu}_1) > (N_1 \hat{\mu}_1 + N_2 \hat{\mu}_2) \lambda \Sigma^{-1} (\hat{\mu}_2 - \hat{\mu}_1) \\ x^T \Sigma^{-1} (\hat{\mu}_2 - \hat{\mu}_1) > \frac{1}{N} (N_1 \hat{\mu}_1 + N_2 \hat{\mu}_2) \Sigma^{-1} (\hat{\mu}_2 - \hat{\mu}_1) \end{aligned}$$

$$\text{LDA} \cdot x^T \Sigma^{-1} (\hat{\mu}_2 - \hat{\mu}_1) > \frac{1}{2} (\hat{\mu}_1 + \hat{\mu}_2) \Sigma^{-1} (\hat{\mu}_2 - \hat{\mu}_1)$$

The two rules are equal if  $N_1 = N_2 = \frac{N}{2}$

## Linear Discriminant Analysis

$$\delta_K(x) = x^T \mu_K - \frac{1}{2} \mu_K^T \Sigma_K^{-1} \mu_K + \log \pi_K$$

$$\Pr[G=K | X=x] = \frac{f_K(x) \pi_K}{\sum_{k=1}^K f_k(x) \pi_k}$$

$f_k(x)$  Gaussian  $\rightarrow$  LDA, QDA

$$\Sigma_K = \sum \pi_k \rightarrow \text{LDA}$$

When we do NOT assume  $\Sigma_K = \sum \pi_k$ , then QDA

$$\delta_K(x) = -\frac{1}{2} \log |\Sigma_K| - \frac{1}{2} (x - \mu_K)^T \Sigma_K^{-1} (x - \mu_K) + \log \pi_K$$

QDA the quadratic term does not simplify

- estimates for QDA are the same as those for LDA, but
  $\hat{\Sigma}_K = \sum_{j=1}^K (x_j - \hat{\mu}_K)(x_j - \hat{\mu}_K)^T / (N_K - 1)$
- similar techniques, difference in # of parameters to estimate
  - for LDA, # pers =  $(K-1) \times (P-1)$   $\rightarrow$  # parameters to estimate all differences to contrast for each difference  $S_K - \delta_1$
  - for QDA, # pers =  $(K-1) \times \left[ \frac{P(P+3)}{2} + 1 \right]$

Note: both methods perform quite well in a large number of situations

- data support only linear (or quadratic) decision boundaries
- Gaussian models are stable

## Regularized Discriminant Analysis

- idea: create a sort of compromise between LDA and QDA
- we allow differences among the covariance matrices, but we shrink them toward  $\bar{\Sigma}$  & similar to ridge

$$\hat{\Sigma}_K(\alpha) = \underline{\alpha} \hat{\Sigma}_K + (\underline{1}-\alpha) \hat{\Sigma}$$

where  $\alpha \in [0; 1]$  should be chosen (i.e., by cross-validation)  
 $\alpha$  controls the amount of shrinkage

$$\alpha = 0 \rightarrow \text{LDA}$$

$$\alpha = 1 \rightarrow \text{QDA}$$

- Further possibility is to shrink  $\hat{\Sigma}$  toward  $\sigma^2 I$

$$\hat{\Sigma} = \gamma \hat{\Sigma} + (1-\gamma) \sigma^2 I$$

where  $\gamma \in [0; 1]$  has a similar meaning than  $\alpha$

- Combining

$$\hat{\Sigma}_K(\alpha, \gamma) = \alpha \hat{\Sigma}_K + (1-\alpha) [\gamma \hat{\Sigma} + (1-\gamma) \sigma^2 I]$$

we obtain a general family for the covariate matrix, depends which on  $\alpha, \gamma$

## Reduced-rank LDA

Fisher: find the best combination  $\alpha = \alpha X$  such that the between-class variance is maximized relative to the within-class variance

Total variance  $T = \underline{B} + \underline{W}$

$$\begin{aligned}
 T &= \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T \\
 &= \frac{1}{N} \sum_{k=1}^K \sum_{g_i=k} ((x_i - \bar{x}_k) + (\bar{x}_k - \bar{x}))((x_i - \bar{x}_k) + (\bar{x}_k - \bar{x}))^T \\
 &= \frac{1}{N} \left[ \sum_{k=1}^K \sum_{g_i=k} (x_i - \bar{x}_k)(x_i - \bar{x}_k)^T + \sum_{k=1}^K \left[ \sum_{j \neq k}^K (x_i - \bar{x}_j) \right] (\bar{x}_k - \bar{x})^T + \right. \\
 &\quad \left. + \sum_{k=1}^K (\bar{x}_k - \bar{x}) \sum_{g_i=k} (x_i - \bar{x}_k)^T + \sum_{k=1}^K \sum_{g_i=k} (\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^T \right] \\
 &= \underbrace{\frac{1}{N} \sum_{k=1}^K \sum_{g_i=k} (x_i - \bar{x}_k)(x_i - \bar{x}_k)^T}_{\text{Within-class variance}} + \underbrace{\frac{1}{N} \sum_{k=1}^K N_k (\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^T}_{\text{between-class variance}}
 \end{aligned}$$

$x_i$ : from class  $k$   
 $g_i$ : class of  $x_i$

e:  $\alpha^T B \alpha$  is maximized  
 $\alpha^T W \alpha$  is minimized

$$a_1 : \arg \max_{\alpha} \frac{\alpha^T B \alpha}{\alpha^T W \alpha} \quad \text{ans} \quad \max_{\alpha} \alpha^T B \alpha \text{ subject to } \alpha^T W \alpha = 1$$

→ generalized eigenvalue problem,  $a$  corresponds to the largest eigenvalue of  $W^{-1}B$

$$a_2 : a_2 \perp a_1, \quad \arg \max_{\alpha} \frac{\alpha^T B \alpha}{\alpha^T W \alpha}$$

$a_3, \dots$

$a_1, a_2, \dots$  are called:  
 - discriminant coordinates  
 - canonical variates

## Properties

- initially data reduction for visualization
- can be seen as a restricted classification rule

the centroids lie in the  $L$ -dimensional subspace of  $\mathbb{R}^P$

## Logistic regression

- model the posterior probabilities of the  $K$  classes, s.t.
  - linear functions in  $x$
  - sum of them = 1
  - they  $\in [0; 1]$

Logistic regression models

$$\log \frac{\Pr[G=1 | X=x]}{\Pr[G=K | X=x]} = \beta_{10} + \beta_1^T x$$

$$\log \frac{\Pr[G=2 | X=x]}{\Pr[G=K | X=x]} = \beta_{20} + \beta_2^T x$$

:

$$\log \frac{\Pr[G=K-1 | X=x]}{\Pr[G=K | X=x]} = \beta_{K-10} + \beta_{K-1}^T x$$

- specifies  $K-1$  log-odd
- based on the logit transformation

$$\begin{array}{l} K=2 \\ \geq 2 \end{array} \quad \log \frac{P}{1-P} = x\beta \Leftrightarrow P = \frac{e^{x\beta}}{1+e^{x\beta}}$$

$$\Pr[G=1 | X=x] = P = \frac{e^{x\beta^T}}{1+e^{x\beta^T}}$$

$$\Pr[G=2 | X=x] = 1 - \Pr[G=1 | X=x] = 1 - P = \frac{1}{1+e^{x\beta^T}}$$

$$\frac{e^{x\beta^T}}{1+e^{x\beta^T}} + \frac{1}{1+e^{x\beta^T}} = 1$$

$$\begin{aligned} \beta &= \beta_0 \ \beta \\ x &= (1, x) \end{aligned}$$

$$\hat{\beta} = \arg \max_{\beta} L(\beta)$$

$$\begin{aligned} L(\beta) &= \prod_{i=1}^n f(y_i; P(x_i; \beta)) \\ &= \prod_{i=1}^n \cancel{\left( \frac{1}{\rho(x_i; \beta)} \right)^{y_i} \left( 1 - \rho(x_i; \beta) \right)^{1-y_i}} \end{aligned}$$

$$\begin{aligned} l(\beta) &= \sum_{i=1}^n y_i \log \rho(x_i; \beta) + (1-y_i) \log (1-\rho(x_i; \beta)) \\ &= \sum_{i=1}^n \left[ y_i \log \frac{e^{x_i \beta^\top}}{1+e^{x_i \beta^\top}} + \log \left( 1 - \frac{e^{x_i \beta^\top}}{1+e^{x_i \beta^\top}} \right) - y_i \log \left( 1 - \frac{e^{x_i \beta^\top}}{1+e^{x_i \beta^\top}} \right) \right] \\ &= \sum_{i=1}^n \left[ y_i x_i \beta^\top - y_i \log (1+e^{x_i \beta^\top}) + \log \left( \frac{1+e^{x_i \beta^\top} - e^{x_i \beta^\top}}{1+e^{x_i \beta^\top}} \right) + y_i \log \left( \frac{1}{1+e^{x_i \beta^\top}} \right) \right] \\ &= \sum_{i=1}^n y_i x_i \beta^\top - \log (1+e^{x_i \beta^\top}) \end{aligned}$$

$$\begin{aligned} \ell_\beta(\beta) &= \frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^n \left[ y_i x_i - \frac{e^{x_i \beta^\top}}{1+e^{x_i \beta^\top}} x_i \right] = 0 \\ &= \sum_{i=1}^n x_i (y_i - \rho(x_i; \beta)) = 0 \quad \text{system of } p+1 \\ &\quad \text{equations not linear in } \beta \end{aligned}$$

→ to find the solution, we can use the Newton-Raphson algorithm

$$x_{n+1} = x_n - \frac{\ell(x_n)}{\ell'(x_n)} \leftarrow \ell_\beta(\beta)$$

$$\ell_{\beta\beta}(\beta) = \frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^\top}$$

$$\begin{aligned} \ell_{\beta\beta}(\beta) &= - \sum_{i=1}^n \left( x_i \frac{e^{x_i \beta^\top} (1+e^{x_i \beta^\top}) - x_i e^{x_i \beta^\top} e^{x_i \beta^\top}}{(1+e^{x_i \beta^\top})^2} \right) x_i^\top / (1+e^{x_i \beta^\top})^2 \\ &= - \sum_{i=1}^n x_i \frac{e^{x_i \beta^\top} (1+e^{x_i \beta^\top} - e^{x_i \beta^\top})}{(1+e^{x_i \beta^\top})^2} x_i^\top \\ &= - \sum_{i=1}^n x_i \frac{e^{x_i \beta^\top}}{(1+e^{x_i \beta^\top})^2} x_i^\top \end{aligned}$$

## Newton-Raphson

$$\beta_{\text{new}} = \beta_{\text{old}} - \ell_{\beta\beta}(\beta)^{-1} \ell_\beta(\beta)$$

$$\ell_\beta(\beta) = X^T(y - P)$$

$$\ell_{\beta\beta}(\beta) = -X^T W X$$

$$\frac{e^{xP^\top}}{1+e^{xP^\top}} \cdot \frac{1}{1+e^{xP^\top}} = \frac{e^{xP^\top}}{(1+e^{xP^\top})^2}$$

$$P = \frac{e^{xP^\top}}{1+e^{xP^\top}}$$

where  $W = \begin{bmatrix} & \\ & \end{bmatrix}$

$$P(1-P)$$

$$\beta_{\text{new}} = \beta_{\text{old}} + \frac{\ell_{\beta\beta}(\beta)}{\ell_\beta(\beta)}$$

$\beta$  in  $W$  and  $P$   
are  $\beta_{\text{old}}$

$$= (X^T W X)^{-1} X^T W (X \beta_{\text{old}} + W^{-1}(y - P))$$

$$\approx (X^T W X)^{-1} X^T W \rightarrow \text{weighted least square}$$

- repeat the steps of the Newton-Raphson algorithm until it converges to  $\hat{\beta}$

## $L_1$ regularized logistic regression

The  $L_1$  penalty (LASSO) can be applied to the logistic regression as well

$$\hat{\beta} = \arg_{\beta} \min \left\{ -\ell(\beta) + \lambda \sum_{j=1}^p |\beta_j| \right\}$$

$$= \arg_{\beta} \max \left\{ \ell(\beta) - \lambda \sum_{j=1}^p |\beta_j| \right\}$$

$$= \arg_{\beta} \max \left\{ \sum_{i=1}^n \left[ y_i (\beta_0 + \beta^T x_i) - \log(1 + e^{\beta_0 + \beta^T x_i}) - \lambda \sum_{j=1}^p |\beta_j| \right] \right\}$$

## Exercise:

- try to reproduce Table 3.3 with the data from the South Africa heart disease example

$\hat{\beta}$	LS	BEA( $\alpha=0.05$ )	LASSO	RIDGE	LDA	QDA

- read ch 4.4.5

Logistic regression vs LDA

$$\text{LDA} \quad \log \frac{\Pr[G=0 | X=x]}{\Pr[G=1 | X=x]} = \log \frac{\hat{\pi}_0}{\hat{\pi}_1} + \frac{1}{2} (\mu_0 - \mu_1)^T \Sigma^{-1} (\mu_0 - \mu_1) + x^T \Sigma^{-1} (\mu_0 - \mu_1)$$

$$= \underline{\alpha_{01}} + \underline{\alpha_1^T x}$$

$$= \underline{\beta_{01}} + \underline{\beta_1^T x}$$

similar to the logistic regression

$$\Pr[G=k, X=x] = \Pr(x) \Pr[G=k | X=x]$$

both LDA and logistic regress.

$$\frac{e^{\beta_0 x + \beta_k^T x}}{1 + \sum_{l=1}^K e^{\beta_l x + \beta_k^T x}}$$

LDA also model this part

$$\Pr(x) = \sum_{k=1}^K \hat{\pi}_k \phi(x; \mu_k, \Sigma)$$

Gaussian

## Model assessment and model selection

- ① evaluate the performance (in term of prediction) of a selected model
- ② select the best model's (for prediction)

GOAL of  $\Rightarrow$  prediction model

generalization: a prediction model must be valid in broad generality, and not valid for a specific dataset

Bias, variance, model complexity

Define:

$Y$  = target variable

$X$  = input matrix

$\hat{f}(x)$  prediction rule, that is trained on a training set  $\mathcal{T}$

Error is measured through a loss function

$$L(Y, \hat{f}(x))$$

that should penalize differences between  $Y$  and  $\hat{f}(x)$

typical choice  
for continuous  
outcomes

$$L(Y, \hat{f}(x)) = \begin{cases} (Y - \hat{f}(x))^2 & \text{quadratic loss} \\ |Y - \hat{f}(x)| & \text{absolute loss} \end{cases}$$

Test error (generalization error) is a prediction error computed on an independent sample

$$\text{Err}_{\mathcal{T}} = E[L(Y, \hat{f}(x)) \mid \mathcal{T}]$$

relabel,  $X, Y$

$\mathcal{T}$  is fixed, is the specific training sets on which we derive our prediction rule

In general, we would like to minimize the expected prediction error

$$\text{Err} = E[L(Y, \hat{f}(x))] = E[\text{Err}_{\mathcal{T}}]$$

We do not want a model with the smallest prediction error for the specific training set, but for the general case

Since we have our training set, we are going to estimate  $\text{Err}_\tau$

↳ the training error is NOT a good estimate of  $\text{Err}_\tau$

$$\text{↳ } \bar{\text{err}} = \frac{1}{N} \sum_{i=1}^N L(y_i; f(x_i))$$

We do NOT want to minimize the training error.

We saw that by increasing the model complexity, we can always decrease the training error

**OVERFITTING** ↳ our model is specific for the training set  
 ↳ generalizes poorly

Similar story for categorical outcome

$G$ : target variable → takes  $K$  values in  $\mathcal{G}$

typical loss functions in this case  $L(G, \hat{G}(x)) \cdot \begin{cases} \underline{1}_{(G \neq \hat{G}(x))} \\ -2 \log L(\beta) \end{cases}$

indicator function  
 $\underline{1}_{(G \neq \hat{G}(x))} \begin{cases} 1 & \text{if } G \neq \hat{G}(x) \\ 0 & \text{otherwise} \end{cases}$

0-1 loss

deviance

- 2  $\ell(\beta)$  is a general loss function, it can be used in cases (Binomial, Gamma, Poisson, log-normal, ...)

- the factor -2 is added to make the loss function be equal to the squared loss in the Gaussian case

$$L(\beta) = \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \sum_{i=1}^N (y_i - x_i^\top \beta)^2 \right\}$$

$$\ell(\beta) = -\frac{1}{2} \sum_{i=1}^N (y_i - x_i^\top \beta)^2$$

Sum of squares — -2  $\ell(\beta)$  = Squared loss

In an ideal situation  
 ↳ a lot of data

we can randomly split the observations in three independent sets

training	validation	test
model selection		model assessment

- training set : contains the data on which we fit our models
- validation set : data we use to identify the best model
- test set : data to assess the performance of the selected model

↳ this set must be considered only at the end of the analysis,  
 i.e. only when we have already chosen the best model  
 Must be ignored for model selection  
 - avoid overoptimism

How to split the data in the three sets :

- no general rule
- back suggestion is : 50% training set  
                           25% validation set  
                           25% test set

- it depends:
  - + sample size
  - + on the signal-to-noise ratio
  - + complexity of the model we are considering

## The Bias-Variance Decomposition

$$Y = f(x) + \varepsilon \quad , \quad E[\varepsilon] = 0$$

$\text{Var}[\varepsilon] = \sigma_\varepsilon^2$

$$\begin{aligned} \text{Err}(x_o) &= E[(Y - \hat{f}(x_o))^2 | X = x_o] \\ &= \sigma_\varepsilon^2 + (E[\hat{f}(x)] - f(x_o))^2 + E[(\hat{f}(x_o) - E[\hat{f}(x_o)])^2] \\ &= \underset{\substack{\text{irreducible} \\ \text{error}}}{\sigma_\varepsilon^2} + \text{bias}^2 + \text{variance} \end{aligned}$$

where

- $\sigma_\varepsilon^2$ , the variance of the target around the true mean, so we cannot do anything about that
- bias<sup>2</sup>, the squared difference between the average of our estimates and the true mean
- variance, the expected squared difference between  $\hat{f}(x)$  and its mean

kNN

$$\hat{f}(x_o) = \frac{1}{K} \sum_{x_i \in N_K(x_o)} y_i$$

$$\text{Err}(x_o) = \sigma_\varepsilon^2 + \left[ f(x_o) - \frac{1}{K} \sum_{k=1}^K f(x_{e_k}) \right]^2 + \frac{\sigma_\varepsilon^2}{K}$$

$$E[\hat{f}(x)] = E\left[\frac{1}{K} \sum_{k=1}^K Y_{e_k}\right] = \frac{1}{K} \sum_{k=1}^K E[Y_{e_k}] = \frac{1}{K} \sum_{k=1}^K f(x_{e_k})$$

$$\text{Var}[\hat{f}(x)] = \text{Var}\left[\frac{1}{K} \sum_{k=1}^K Y_{e_k}\right] = \frac{1}{K^2} \sum_{k=1}^K \text{Var}[Y_{e_k}] = \frac{1}{K} \sum_{k=1}^K \sigma_\varepsilon^2 = \frac{K}{K} \sigma_\varepsilon^2 = \frac{\sigma_\varepsilon^2}{K}$$

$$\#K \propto \frac{1}{\text{complexity}}$$

increase K → decrease complexity  
 ↓ more bias  
 ↓ reduce the variance

Similar for linear model  $\hat{f}(x; \beta) = x^T \beta$

- slightly more difficult to derive, it turns out to be

$$\frac{1}{N} \sum_{i=1}^n \text{Err}(x_i) = \sigma_\epsilon^2 + \frac{1}{N} \sum_{i=1}^n [\hat{f}(x_i) - E[\hat{f}(x_i)]]^2 + \frac{P}{N} \sigma_\epsilon^2$$

called the in-sample error

$P = \# \text{ of variables}$

increasing the model complexity,  
we increase the variance  
component of the error

For regularized regression (e.g., lasso, ridge, ...) the form is the same,  
but there is an additional dependence on the tuning (complexity) parameter  $\alpha$

- We can go more into details on the bias component

$$E_{x_*} [\hat{f}(x_*) - E[\hat{f}(x_*)]]^2 = E_{x_*} [\hat{f}(x_*) - x_*^T \hat{\beta}_*]^2 + E_{x_*} [x_*^T \beta - E[x_*^T \beta]]^2$$

$$\hat{\beta}_* = \arg \min_{\beta} E[\hat{f}(x) - x^T \beta]^2$$

$$\text{Average [model bias]}^2 + \text{Avg [Estimation bias]}^2$$

difference between the  
true function and the best fitting  
linear approximation

error between the average  
estimate and the best model

difference between truth and best model

additional bias that we add  
when, e.g., add shrinkage

- we can reduce it only increasing the model  
space (more variables, more complex relationship - interaction -,...)

OLS = 0

## Optimism of the Training Error

Let us start from the definitions of test time:

$$\text{Err}_{\mathcal{T}} = E_{(x_0, y_0)} [L(Y_0, \hat{f}(x_0)) | \mathcal{T}]$$

Test error, where

- $x_0, Y_0$  are new (test) point  $\rightarrow$  random

- $\mathcal{T}$  is fixed  $\mathcal{T} = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$

Averaging over all training sets

$$\text{Err} = E_{\mathcal{T}} [E_{(x_0, y_0)} [L(Y_0, \hat{f}(x_0)) | \mathcal{T}]]$$

gives the expected error

We also saw the training error

$$\bar{\text{err}} = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{f}(x_i))$$

is NOT a good estimate of the test error, because it underestimates it.

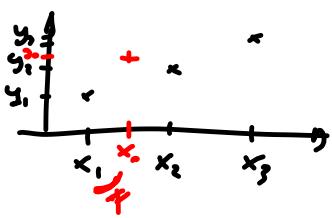
Same data used for both training  $\hat{f}(x)$  and to test its performance  $\rightarrow$  optimistic estimate

The test error can be thought as extra-sample error (the estimation of the error is computed on new points  $x_0 \neq x_i$ )

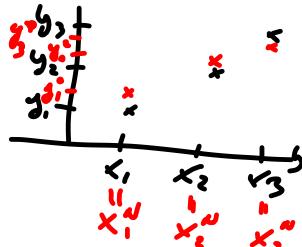
We are going to evaluate the optimism of  $\bar{\text{err}}$  in the in-sample case, i.e. we have new observations in the same points of training set

$$\text{Err}_{in} = \frac{1}{N} \sum_{i=1}^N E_{Y_0} [L(Y_i, \hat{f}(x_i)) | \mathcal{T}]$$

extra-sample



in sample



-only  $Y_0$  is random, new  $y_i$  for  $x_1, \dots, x_N$

**Definition:** we define optimism the difference between  $\text{Err}_{\text{in}}$  and  $\overline{\text{err}}$

$$\text{op} := \text{Err}_{\text{in}} - \overline{\text{err}}$$

$\text{op}$  is generally positive, as  $\overline{\text{err}}$  is computed on training data

**Definition:** average optimism is

$$\omega := E_y[\text{op}]$$

*we are computing the expected value over the training set outcome*

For a reasonable number of loss functions, including 0-1 loss and squared error, it can be shown that

$$\omega = \frac{2}{N} \sum_{i=1}^N \text{Cov}(\hat{y}_i, y_i)$$

where Cov denotes the covariance  $\rightarrow \text{Ex 7.4}$

**Note:**

- optimism depends on how much  $y_i$  affects its own prediction
- the harder we fit the data, the larger the value of  $\text{Cov}(\hat{y}_i, y_i)$   
 $\rightarrow$  the higher the optimism

As a consequence:

$$E_y[\underline{\text{Err}_{\text{in}}}] = E_y[\overline{\text{err}}] + \frac{2}{N} \sum_{i=1}^N \text{Cov}(\hat{y}_i, y_i)$$

When  $\hat{y}_i$  is obtained by a linear fit in the inputs,  $Y = f(x) \cdot \varepsilon$

$$\sum_{i=1}^N \text{Cov}(\hat{y}_i, y_i) = d \sigma_\varepsilon^2 \quad \leftarrow \begin{matrix} \text{effective number} \\ \text{of parameters} \end{matrix}$$

therefore

$$\rightarrow E_y[\underline{\text{Err}_{\text{in}}}] = E_y[\overline{\text{err}}] + 2 \frac{d}{N} \sigma_\varepsilon^2 \quad (*)$$

- optimism increases with the number of inputs;
- decreases " " sample size.

*e.g. in linear regression, the number of covariates*

Methods we will see:

- Cp, AIC and BIC estimate the prediction error by estimating the training error and the optimism (work when estimates are linear in their parameters)
- cross-validation and bootstrap-based procedure try to estimate directly the expected error

**Note:**

- in-sample error is generally NOT of interest (we are mainly interested in new data, including new points in  $X$ )
- to select the best model / tuning the complexity parameter, we are more interested in the relative difference in errors rather than the absolute one.

## Estimates of $\text{Err}_{\text{in}}$

Start from (\*)

$$E_g[\bar{\text{err}}_{\text{in}}] = E_g[\bar{\text{err}}] + 2 \frac{d}{N} \hat{\sigma}_e^2 \quad (*)$$

Write the general form of the in-sample estimate

$$\hat{\bar{\text{err}}}_{\text{in}} = \bar{\text{err}} + \hat{w}$$

- when we have linearity and squared errors, from (\*)

$$C_P = \bar{\text{err}} + 2 \frac{d}{N} \hat{\sigma}_e^2$$

where:

-  $\bar{\text{err}}$  is computed through the squared loss;

-  $d$  is # of parameters

-  $\hat{\sigma}_e^2$  is an estimate of the noise variance

it is computed using the full model, because it has the smallest bias

NB: There are other versions of  $C_P$ , different versions may give different numbers, but they all lead to the same model

Similar idea for AIC (Akaike Information Criterion)

- we start again from (\*), but we want to be more general  
(the error is computed through a likelihood approach)

We are using the asymptotic result ( $N \rightarrow \infty$ )

$$-2 E[\log P_\theta(Y)] \approx -\frac{2}{N} E\left[\sum_{i=1}^N \log P_\theta(y_i)\right] + 2 \frac{d}{N}$$

$P_\theta(Y)$  is the family of densities for  $Y$  (containing the "true" density)

loglik ↴ the maximized log-likelihood  
 $\ell(\hat{\theta})$   
is the mle maximum likelihood estimate

E.g., in the logistic regression:  $AIC = -\frac{2}{N} \log \text{lik} + 2 \frac{d}{N}$

Gaussian regression:  $AIC \propto \text{G}$

AIC ↴  $C_P$  is a special case of AIC

To find the best model, we choose that with the smallest AIC

- straightforward in the simplest cases, we need more attention in more complex situations. In particular, we need to find a reasonable measure for the model complexity

Note: for regularized/penalized regression

$$AIC(\alpha) = \bar{\text{err}}(\alpha) + 2 \times \frac{d(\alpha)}{N} \hat{\sigma}_e^2$$

- usually minimizing AIC is not the best solution to find the value of the tuning parameter  $\alpha$  → cross-validation is better approach

## The effective number of parameters

- generalized the concept of number of parameters, in order to extend the previous approaches to more complex situations

Suppose

$$\mathbf{y} = (y_1, \dots, y_N) \text{ outcome}$$

$$\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_N) \text{ predictions}$$

Linear regression

$$\hat{\mathbf{y}} = \underbrace{\mathbf{x}(\mathbf{x}^T \mathbf{x})^{-1} \mathbf{x}^T}_{S} \mathbf{y}$$

Linear methods       $\hat{\mathbf{y}} = S\mathbf{y}$       Ridge regression

where  $S$  is a  $N \times N$  matrix which:

- depends on  $X$
- independent on  $\mathbf{y}$

$$\hat{\mathbf{y}} = \underbrace{\mathbf{x}(\mathbf{x}^T \mathbf{x} + \lambda \mathbf{I}_p)^{-1} \mathbf{x}^T}_{S} \mathbf{y}$$

Effective number of parameters (or effective degrees of freedom)

$$df(S) = \text{trace}(S)$$

value of  $\lambda$

We should replace  $\text{trace}(S)$  to  $d$  to obtain the correct criterion

If  $\mathbf{y} = f(\mathbf{x}) + \varepsilon$ ,  $\text{Var}(\varepsilon) = \sigma_\varepsilon^2 \rightarrow \sum_{i=1}^n \text{Cov}(\hat{y}_i, y_i) = \text{trace}(S) \sigma_\varepsilon^2$

So  $df(\hat{\mathbf{y}}) = \frac{\sum_{i=1}^n \text{Cov}(\hat{y}_i, y_i)}{\sigma_\varepsilon^2} \rightarrow \text{Ex 7.5}$

## The Bayesian approach and BIC

The BIC is an alternative criterion to AIC,

↳ Bayesian Information Criterion

$$\frac{1}{N} \text{BIC} = -\frac{2}{N} \log \text{lik} + \log N \cdot \frac{c}{N}$$

$$\frac{\text{AIC}}{\text{BIC}} = e^{\frac{2}{N}}$$

Despite they are quite similar, AIC and BIC come from completely different ideas. BIC comes from the Bayesian approach to model selection

$$\Pr(M_m | z) \propto \Pr(M_m) \Pr(z | M_m)$$

$$\propto \Pr(M_m) \int \Pr(z | M_m, \theta_m) \Pr(\theta_m | M_m) d\theta_m$$

To choose between two models, we compare their posterior probabilities

$$\frac{\Pr(M_0 | z)}{\Pr(M_e | z)} = \frac{\Pr(M_m)}{\Pr(M_e)} \cdot \frac{\Pr(z | M_m)}{\Pr(z | M_e)}$$

↑  
in a large number  
of case = 1  
(give the same prior probability to the two model)

Bayes factor  
the choice between the two models is based on the Bayes factor

Approximately

$$\Pr(z | M_m) = \log(z | \hat{\theta}_m, M_m) - \frac{c_m}{2} \log N + O(1)$$

If the loss function is  $-2 \log \Pr(z | \hat{\theta}, M_m)$ , we obtain BIC

- We <sup>may</sup> select the model with smallest BIC

correspond to selecting the model with highest posterior probability

Note that

$$\frac{e^{-\frac{1}{2} \text{BIC}_m}}{\sum_{e=1}^E e^{-\frac{1}{2} \text{BIC}_e}}$$

is the posterior probability of selecting the model m

## AIC vs BIC

- no clear preference
- BIC leads to a sparser Model
- AIC leads to a model with more predictors
- BIC is consistent ( $N \rightarrow \infty$ , the probability of selecting the true model goes to 1)
- For finite sample size, BIC tends to select a model which is too sparse

## Cross-validation

The cross-validation aims to estimate the ~~extra-sample~~ error

$$Err = E[L(Y, \hat{f}(x))]$$

the average test error when  $\hat{f}(x)$  is applied to a new sample

If we had enough data  $\hookrightarrow$  training set  
"test"

$\hookrightarrow$  in general, we have not, so we mimic this split using the limited amount of data we have



- divide the observations in  $K$  folds
- we use, in turn,  $K-1$  folds to train the model (derive  $f^{(k)}(x_i)$ )
- we evaluate the model in the remaining fold

$$CV(\hat{f}) = \sum_{k=1}^K L(y_i, \hat{f}^{(k)}(x_i))$$

- if  $K=2$ , two-fold cross-validation
- if  $K=N$ , leave-one-out cross-validation (LOOCV)  
in this case, each observation is a fold

How do we choose  $K$

- bias-variance trade-off
- smaller the  $K$ , larger bias, smaller variance
- larger the  $K$ , smaller bias, larger variance  
(the extreme case is LOOCV, where we use  $N-1$  observations for training the model  $\rightarrow$  the training sets are really similar to each other)
- usual choices are  $K=5$  or  $K=10$

[Fig 7.8]

$\rightarrow$  1-Err vs  $N$

$\rightarrow$  the classifier is OK until  $\approx N=100$  (then is flat)

• if  $N=200$ ,  $K=5 \rightarrow$  training  $N_t = 160$      $\checkmark$      $160 > 100$

• if  $N=50$ ,  $K=5 \rightarrow$  ..  $N_t = 40$      $\times$      $40 \cancel{>} 100$

Notes:

- CV estimates  $\text{Err}$  and not the  $\text{Err}_2$
- if we want to select a tuning parameter viz CV

$\hat{f}^{-k}(x, \alpha)$  is the model selected using  $\alpha$  and  $k$  folds on the observations which do not belong to the  $k$ -th fold

$$CV(\hat{f}, \alpha) = \frac{1}{n} \sum_{k=1}^{K_n} \sum_{i=1}^{n_k} L(y_i, \hat{f}(x_i, \alpha))$$

$$\hat{\alpha} = \arg \min_{\alpha} CV(\hat{f}, \alpha)$$

Generalized cross-validation

- convenient approximation to the LOOCV, for squared loss function

$$\text{Loocv} \quad \hat{f} = Sy$$

$$N \sum_{k=1}^{n_k} \sum_{i=1}^{n_k} \rightarrow \frac{1}{N} \sum_{i=1}^N (y_i - \hat{f}_{-i}(x_i))^2 = \frac{1}{N} \sum_{i=1}^N \left( \frac{y_i - \hat{f}(x_i)}{1 - S_{ii}} \right)^2$$

where  $S_{ii}$  is the  $i$ th term on the diagonal of  $S$

The generalized cross-validation (GCV)

$$GCV(\hat{f}) = \frac{1}{N} \sum_{i=1}^N \left( \frac{y_i - \hat{f}(x_i)}{1 - \text{trace}(S)/N} \right)^2$$

- computational advantages;
- similarities between AIC and GCV → Ex 7.7

Exercise 7.4

$$E_{\text{err}_{in}} = \frac{1}{N} \sum_{i=1}^N E_{Y_i} \left[ (Y_i^o - \hat{f}(x_i))^2 \right]$$

$$\bar{\text{err}} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{f}(x_i))^2 \quad y = f(x) + \varepsilon.$$

$$E_Y[\sigma_p] = \frac{2}{N} \sum_{i=1}^N \text{Cov}(\hat{g}_i, y)$$

$$\begin{aligned} E_Y[\sigma_p] &= E_Y \left[ \frac{1}{N} \sum_{i=1}^N \left\{ E_{Y_i} \left[ (Y_i^o - \hat{f}(x_i))^2 \right] - (y_i - \hat{f}(x_i))^2 \right\} \right] \\ &= E_Y \left[ \frac{1}{N} \sum_{i=1}^N \left\{ E_{Y_i} [Y_i^{o2}] - 2E_{Y_i} [Y_i^o] \hat{f}(x_i) + \hat{f}(x_i)^2 - y_i^2 + 2E_Y[y_i \hat{f}(x_i)] - \hat{f}(x_i)^2 \right\} \right] \\ &= \frac{1}{N} \sum_{i=1}^N E_Y \left[ E_{Y_i} [Y_i^{o2}] \right] - 2E_Y[Y_i^o] E_Y[\hat{f}(x_i)] - E_Y[y_i^2] + 2E_Y[y_i \hat{f}(x_i)] \end{aligned}$$

$$\bullet E_Y \left[ E_{Y_i} [Y_i^{o2}] - \hat{f}(x_i)^2 + \hat{f}(x_i)^2 \right] = \sigma_\varepsilon^2 + f(x)^2$$

$$\bullet E_Y \left[ y^2 - \hat{f}(x)^2 + \hat{f}(x)^2 \right] = \sigma_\varepsilon^2 + f(x)^2$$

$$\bullet E_{Y_i} [Y_i^o] = f(x) = E_Y[y] \quad \hat{f}(x) = \hat{y}$$

$$\begin{aligned} E_Y[\sigma_p] &= \frac{1}{N} \sum_{i=1}^N \left\{ -2E_Y[y] E_Y[\hat{y}] + 2E_Y[y_i \hat{y}_i] \right\} \\ &= \frac{2}{N} \sum_{i=1}^N \text{Cov}(y_i, \hat{y}_i) \end{aligned}$$

Ex 7.5

For  $\hat{y} = Sy$  show that  $\sum_{i=1}^n \text{Cov}(y_i, \hat{y}_i) = \text{trace}(S)\sigma^2$

$$\begin{aligned}\sum_{i=1}^n \text{Cov}(y_i, \hat{y}_i) &= \text{trace}(\text{Cov}(y, \hat{y})) \\ &= \text{trace}(\text{Cov}(y, Sy)) \\ &= \text{trace}(S \text{Cov}(y, y)) \\ &\stackrel{!}{=} \text{trace}(S \text{Var}(y)) \\ &\stackrel{!}{=} \text{trace}(S\sigma^2) \\ &\stackrel{!}{=} \text{trace}(S)\sigma^2\end{aligned}$$

Cov matrix  $\begin{pmatrix} \text{Cov}(y_1, y_1) & \text{Cov}(y_1, y_2) & \dots & \text{Cov}(y_1, y_n) \\ \text{Cov}(y_2, y_1) & \text{Cov}(y_2, y_2) & & \vdots \\ \vdots & & \ddots & \\ \text{Cov}(y_n, y_1) & \dots & \ddots & \text{Cov}(y_n, y_n) \end{pmatrix}$

$\sum_{i,j} m_{ij} = \text{trace } M$

Exercise 7.7

Use the approximation:  $\frac{1}{(1-x)^2} \approx 1 + 2x$

to show similarities between  $C_p/AIC$  and  $GCV$

$$C_p = \bar{\text{err}} + 2 \frac{d}{N} \hat{\sigma}_e^2 \quad AIC = -\frac{2}{N} \log \text{lik} + 2 \frac{d}{N}$$

$$GCV = \frac{1}{N} \sum_{i=1}^n \left( \frac{y_i - \hat{f}(x_i)}{1 - \text{trace}(S)/N} \right)^2$$

$$C_p = \frac{1}{N} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2 + \frac{2d}{N} \hat{\sigma}_e^2 \quad \text{trace}(S) = \frac{\sum \text{Cov}(y, \hat{y})}{\sigma^2}$$

$$\begin{aligned}GCV &= \frac{1}{N} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2 \cdot \left( \frac{1}{1 - \frac{\text{trace}(S)}{N}} \right)^2 \\ &= \frac{1}{N} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2 \left( 1 + 2 \cdot \frac{\text{trace}(S)}{N} \right) \\ &= \frac{1}{N} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2 + \frac{2}{N} \text{trace}(S) \frac{\sum (y_i - \hat{f}(x_i))^2}{N} = \bar{\text{err}} + \frac{2}{N} \text{trace}(S) \hat{\sigma}_e^2\end{aligned}$$

Gaussian regression

$$AIC \propto -\frac{2}{N} \log \left( \exp \left\{ -\frac{1}{2\hat{\sigma}_e^2} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2 \right\} \right) + \frac{2d}{N}$$

$$= + \frac{1}{N} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2 + \frac{2d}{N} \quad \text{trace}(S)$$

$$\propto \frac{1}{N} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2 + \frac{2d}{N} \hat{\sigma}_e^2$$

## Bootstrap methods

- what is bootstrap
- how to use bootstrap for error estimation  $\widehat{E[Err_T]}$

IDEA: generate bootstrap sample from the empirical distribution computed on original sample  
 → by resampling with replacement from the original sample

- suppose  $\mathcal{T} = \{(x_1, y_1), \dots, (x_n, y_n)\}$
- by resampling,  $\mathcal{T}_1^* = \{(x_1^{*1}, y_1^{*1}), \dots, (x_n^{*n}, y_n^{*n})\}$
- repeat for  $B$  large,  $\mathcal{T}_2^*, \mathcal{T}_3^*, \dots, \mathcal{T}_B^*$

Based on the generated bootstrap sample (which mimic new experiments) we can estimate any aspect of the distribution of a map

### Example

original sample  $\mathcal{T} = \{z_1, z_2, z_3, z_4\} = \{1, 3, 4, 6\}$

generate  $B$  bootstrap sample  
 by resampling with replacement from  $\mathcal{T}$

$$\mathcal{T}_1^* = \{z_1^*, z_2^*, z_3^*, z_4^*\}$$

⋮

$$\mathcal{T}_3^* = \{1, 1, 1, 1\}$$

$$(x, y) \quad \text{Cov}(x, y)$$

$$\begin{aligned} \mathcal{T} & (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \\ & (4, 3), (1, 3), \dots, (4, 2) \end{aligned}$$

$$\mathcal{T}_1^* = \{(1, 3), (4, 2), \dots, (4, 2)\}$$

## Bootstrap approach for prediction error estimation

### WRONG APPROACH

- estimate our  $\hat{f}(x)$  from each bootstrap sample
- evaluate how well  $\hat{f}_s^*(x)$  estimate  $y$

$$\hat{E}_{\text{err}}^{\text{wrong}} = \frac{1}{B} \sum_{s=1}^B \left( \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{f}_s^*(x_i)) \right)$$

⚠: training and test set are not independent

$$\begin{aligned} E[\hat{E}_{\text{err}}^{\text{wrong}}] &= 0.184, \quad 1\text{NN}, \quad Y \perp X \\ &\left( \begin{array}{l} \bullet y_i \in \mathcal{T}_s^* \rightarrow \text{error} = 0 \\ \bullet y_i \notin \mathcal{T}_s^* \rightarrow \text{error} = 0.5 \end{array} \right) \\ &= 0.5 \times \underbrace{\Pr[y_i \notin \mathcal{T}_s^*]}_{0.368} + 0 \times \underbrace{\Pr[y_i \in \mathcal{T}_s^*]}_{0} \end{aligned}$$

$\Pr[\text{observation } i \notin \text{bootstrap sample } s]$

$$\Pr[\mathcal{T}_{s[i]}^* \neq y_i] = \frac{N-1}{N} \Rightarrow \Pr[y_i \notin \mathcal{T}_s^*] = \left(\frac{N-1}{N}\right)^N$$

same for all positions

$$= e^{-1} \approx 0.368$$

$$E[\hat{E}_{\text{err}}^{\text{wrong}}] \Big|_{\substack{X \perp Y \\ 1\text{NN}}} \approx 0.5 \times e^{-1} = 0.184$$

An important fact

$$\Pr[\text{observation } i \text{ belongs to a bootstrap sample } s] = 1 - e^{-1}$$

≈ 0.632

approximation of  $1 - \left(\frac{N-1}{N}\right)^N$

## CORRECT APPROACH

$$\mathcal{T} = \{z_1, \dots, z_N\}$$

$$\mathcal{T}_s^* = \{z_1^*, \dots, z_{N_s^*}^*\} \text{ resampling with replacement}$$

→ there are original observations which are included more than once  
 ⇒ there are original observations which are not included at all

$\hat{\omega}_{\text{test}}$  These can be used as a test set as they are not used in the training process.

$$\hat{\text{Err}}^{(1)} = \frac{1}{N} \sum_{i=1}^N \frac{1}{|C_i^*|} \sum_{j \in C_i^*} L(y_j, \hat{f}_s^*(x_j))$$

where  $|C_i^*|$  is the number of bootstrap samples that do not contain  $i$

ISSUES

→ the average number of unique observations in the training set is  $0.632N$  → not so far from  $0.5N$ , that is the value related to 2-fold CV

→ similar issues of training-set-size bias than 2-fold CV  
 → result in a small overestimation of the error

To solve the issue, the .632 estimator has been developed

$$\hat{\text{Err}}^{(.632)} = 0.368 \bar{\text{err}} + 0.632 \hat{\text{Err}}^{(1)}$$

In general, it works well, but in some case it fails, like in our  $X \perp Y$

$$\bar{\text{err}} = 0 \rightarrow \hat{\text{Err}}^{(.632)} = 0.632 \hat{\text{Err}}^{(1)}$$

Further improvements ".632+ estimator"

- based on the quantity  $\hat{\gamma}$ , the no-information-error rate  
 error that we obtain when inputs and class label are independent  
 $\hat{\gamma}$  is computed by permuting  $x$  and  $y$  separately, we compute the prediction error for each combination of  $y_i$  and  $x_j$

$$\hat{\gamma} = \frac{1}{N} \sum_{i=1}^N \frac{1}{N} \sum_{j=1}^N L(y_i, \hat{f}(x_j))$$

-  $\hat{\gamma}$  is used to compute the overfitting rate

$$\hat{R} = \frac{\hat{\text{Err}}^{(1)} - \bar{\text{err}}}{\hat{\gamma} - \bar{\text{err}}} \quad 0 \leq R \leq 1$$

$\hat{R}$  no overfitting

- finally

$$\hat{\text{Err}}^{(.632+)} = (1 - \hat{R}) \bar{\text{err}} + \hat{R} \hat{\text{Err}}^{(1)}$$

$$\text{where } \hat{R} = \frac{0.632}{1 - 0.368 \hat{R}}$$

N.B.: bootstrap sample has the same size of the original sample

## Generalized Additive Models

- extensions of the (generalized) linear model

Linear model

- powerful tool

- can be used in several cases (regression, classification, ...)

Main limitations

- it suppose linear effects, often not true in reality

( $\hat{\beta}$  is the increments in  $y$  when the corresponding  $x$  increases by 1)

In the context of regression, the (generalized) additive model has the form

$$E[Y|X_1, \dots, X_p] = \alpha + f_1(x_1) + \dots + f_p(x_p)$$

where

$Y$  is the outcome

$X_j$  are the predictors

$f_j$  is a function which describe the effect of  $X_j$

- we already saw that we can use  $f_j(x_j) = x_j^2$ ,  $f_j(x_j) = \log x_j$

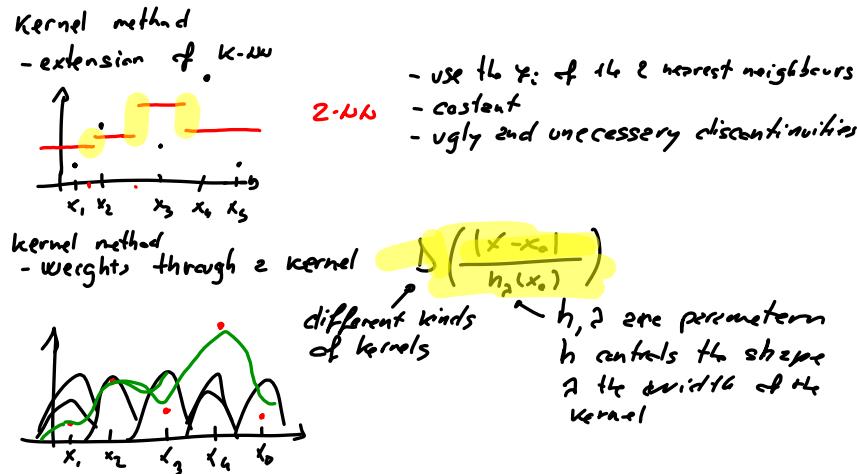
- we can be more general, and use a nonparametric function (splines, kernel, ...)

Splines  $\rightarrow$  § 5.2

kernel  $\rightarrow$  § 6.1, § 6.2

- cubic splines  $\rightarrow$  bottom right of fig 5.2

- natural splines  $\rightarrow$  since the estimation outside the observations' range can be dangerous, the line is forced to be linear outside the range

**GAM**

for functional effect, we can

$$E[Y|X] = \alpha + \beta_1 f_1(x_1) + \beta_2 f_2(x_2) + \dots + \beta_p f_p(x_p)$$

$\downarrow \text{reg}(x_i)$

→ the least square estimator approach is usable

$$E[Y|X] = \alpha + f_1(x_1) + \dots + f_p(x_p)$$

→ backfitting algorithm

**Generalized Additive Model**

↳ extending the GLM (STK 3100)

GLM  $g(\mu(x)) = \alpha + \beta^T x$  extending the linear model to all exponential families sampling models

$\downarrow \text{link function}$

e.g. logistic model

$$g(\cdot) = \text{logit} \quad \mu(x) = P[Y=1|x=x]$$

$$\log\left(\frac{\mu(x)}{1-\mu(x)}\right) = \alpha + \beta_1 x_1 + \dots + \beta_p x_p$$

GAM  $g(\mu(x)) = \alpha + \sum_{j=1}^p f_j(x_j)$

additive logistic regression :  $\log\left(\frac{\mu(x)}{1-\mu(x)}\right) = \alpha + \sum_{j=1}^p f_j(x_j)$

**Advantages of GAM:**

- flexibility, due to  $f$  (we can capture non-linear effects)
- interpretability, due to the additivity (not so different from the usual interpretation of GLM)

Note: not all effect need to be non-linear / linear

- semi parametric model  $g(\mu(x)) = \underbrace{x^T \beta}_{\text{parametric}} + \underbrace{f(z)}_{\text{non-parametric}}$

e.g. semi parametric model: Cox model

$$\lambda(t) = \underbrace{\lambda_0(t)}_{\text{non-parametric}} \underbrace{\exp(x^T \beta)}_{\text{parametric part}}$$

### 8.2.2 Maximum likelihood inference

Todz

- Boosting (Adaboost): the first popular boosting algorithm)
- statistical interpretation of boosting (L2 Boosting)
- likelihood-based boosting (together with model-based boosting (gradient boosting), statistical boosting)

$Y_i$  iid with density  $p(y_i; \theta)$

$$\text{e.g. } Y_i \sim N(\mu, \sigma^2) \rightarrow p(y_i; \underbrace{\mu, \sigma^2}_{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y_i - \mu)^2\right\}$$

$$L(\theta; y) = \prod_{i=1}^n p(y_i; \theta) \quad \text{likelihood}$$

$$l(\theta; y) = \sum_{i=1}^n \log p(y_i; \theta) \quad \text{log-likelihood}$$

$$\hat{\theta} = \arg \max_{\theta} l(\theta; y) = \arg \max_{\theta} L(\theta; y)$$

$$l_\theta(\theta; y) = \frac{\partial l(\theta; y)}{\partial \theta} \quad \text{score function}$$

$$l_{\theta\theta}(\theta; y) = \frac{\partial^2 l(\theta; y)}{\partial \theta \partial \theta^\top} \quad \begin{array}{l} \xrightarrow{\quad} j(\theta) = -l_{\theta\theta}(\theta; y) \\ \swarrow s(\theta) = E_\theta[j(\theta)] \end{array} \quad \begin{array}{l} \text{observed} \\ \text{information} \end{array}$$

If  $\theta_0$  is the true parameter

$$\hat{\theta} \xrightarrow{\text{asym}} N(\theta_0; j(\theta_0)^{-1})$$

Estimation

$$\hat{\theta} \sim N(\hat{\theta}; j(\hat{\theta})^{-1}) \quad \text{or} \quad \hat{\theta} \sim N(\hat{\theta}; i(\theta)^{-1})$$

$$\text{confidence intervals} \quad \hat{\theta} \pm z_{1-\alpha/2} \sqrt{j'(\hat{\theta})}$$

## 10 Boosting

Leo Breiman: "[Boosting is] the best off-the-shelf classifier in the world"

- originally developed for classification;
- translated into the statistical world and use for all purposes (regression, ...)
- extended in its use;
- interpretable (Gini)

Starting challenge:

"Can a committee of blackheads somehow arrive at a highly reasoned decision, despite the weak judgement of the individual members?"

goal: obtain a good classifier

- apply "weak estimators", in the context of classification classifiers which lead to a solution only slightly better than a random choice

idea: repeatedly apply a weak estimator to modifications of the data

gives more weight to the missclassified observations

AdeBoost

Consider a two class classification problem

$$Y_i \in \{-1, 1\}$$

$X_i$ : the vector of inputs

AdaBoost algorithm

① initialize, weights are  $(\frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}) = w^{[0]}$

② for  $m$  from 1 to  $m$ -step ( $M$ )

a) fit the weak estimator  $\hat{g}(x)$  to the weighted data;

b) compute the weighted in-sample missclassification rate

$$\text{err}^{[m]} = \frac{\sum_{i=1}^N w_i^{[m-1]} \mathbb{1}[y_i \neq \hat{g}^{[m]}(x_i)]}{\sum_{i=1}^N w_i^{[m-1]}}$$

c) compute  $\alpha_m = \log \left( \frac{1 - \text{err}^{[m]}}{\text{err}^{[m]}} \right)$

$\alpha_m$  is used to weight the contribution of the  $\hat{g}^{[m]}(x)$  to the final estimate (classification)

d) update the weights

$$\tilde{w}_i = w^{[m-1]} \exp \left\{ \alpha_m \underbrace{\mathbb{1}[y_i \neq \hat{g}^{[m]}(x_i)]}_{\text{reweight only missclassified observations}} \right\} \quad i = 1, \dots, N$$

OK

$$w_i^{[m]} = \frac{\tilde{w}_i}{\sum_{i=1}^N \tilde{w}_i}$$

③ compute final result

$$\hat{f}_{\text{AdaBoost}} = \text{sgn} \left( \sum_{m=1}^{m-\text{stop}} \alpha^{[m]} \hat{g}^{[m]}(x) \right)$$

### Example

$$\text{step } \{0\} \quad w = \left( \frac{1}{10}, \frac{1}{10}, \dots, \frac{1}{10} \right)$$

$$\text{step (i)} \quad \text{err} = \frac{\sum_{i=1}^n \frac{1}{10} \mathbf{1}[y_i \neq \hat{g}_i]}{\sum_{i=1}^n \frac{1}{10}} = \frac{\frac{3}{10}}{\frac{10}{10}} = 0.3$$

$$\alpha_1 = \log \frac{1 - err}{err} = \log 0.7 + \log 0.3 \approx 0.86$$

$$\tilde{w} = \left( \underbrace{\exp(0.84) \frac{1}{10}, 0.23, 0.23, 0.1, \dots, 0.1}_{\text{miss classifier in the first iteration}} \right)$$

$\psi^{(1)} \approx (0.17, 0.17, 0.17, 0.07, \dots, 0.07)$

## Statistical view of Boosting

- functional gradient descent/d algorithm  
forward stagewise (additive) modelling

→ Sep Adaboost is an iterative procedure to minimize a loss-function, in particular an exponential loss-function

$$L(y, f(x)) = \exp\{-y f(x)\}$$

Consider a generic step  $m$

- the current classifier is  $\hat{f}^{(m-1)} = \sum_{k=1}^{m-1} \alpha_k \hat{g}^{(k)}(x)$  1...m-1 given in looking  
(if the algorithm had stopped at the m-1 iteration)
  - the goal is to find  $(\alpha_m, g_m) = \arg \min_{\alpha_m g_m} \sum_{i=1}^n \exp \left\{ -y_i \sum_{k=1}^m \alpha_k \hat{g}^{(k)}(x_i) \right\}$

$$\alpha_m, g_m = \arg \min_{\alpha, g} \sum_{i=1}^n \exp \left\{ -y_i \left( \underbrace{\sum_{k=1}^{m-1} \alpha_k g_k}_{\notin \mathcal{L}^{m-1}} + \hat{g}(x_i) \right) \right\}$$

where  $w_i = \exp\{-g f^{(m-1)}\}$ , which do not depend neither on  $m$  nor on  $g$  (given from the previous iteration)  $g \in \{-1, 1\}$

- two step procedure: first we minimize with respect to  $g$   $\in \{1, 1\}$

$$\begin{aligned}
 & \text{argmin}_g \sum_{i=1}^n w_i^{(i-1)} \exp(-y_i g) \\
 & = \text{argmin}_g \left\{ \underbrace{\sum_{g=y}^{(i-1)} w_i e^{-\alpha}}_{\frac{e^{-\alpha} \sum w_i}{g-y}} + \underbrace{\sum_{g \neq y}^{(i-1)} w_i e^{\alpha}}_{\frac{(e^\alpha - e^{-\alpha}) \sum w_i}{g-y}} \right\} \\
 & = \text{argmin}_g \left\{ e^{-\alpha} \sum_{i=1}^{(i-1)} w_i + (e^\alpha - e^{-\alpha}) \sum_{g \neq y} w_i \right\} \\
 & \text{argmin} \left\{ \underbrace{\sum_{g=y}^{(i-1)} w_i e^{-\alpha}}_{\frac{\sum w_i e^{-\alpha}}{g-y}} + \underbrace{\sum_{g \neq y}^{(i-1)} w_i e^{-\alpha}}_{\frac{\sum w_i e^{-\alpha}}{g-y}} - \underbrace{\sum_{g \neq y}^{(i-1)} w_i e^{-\alpha}}_{\frac{\sum w_i e^{-\alpha}}{g-y}} + \underbrace{\sum_{g \neq y}^{(i-1)} w_i e^{-\alpha}}_{\frac{\sum w_i e^{-\alpha}}{g-y}} \right\} \\
 & = \text{argmin}_g \left\{ e^{-\alpha} \sum_{i=1}^{(i-1)} w_i + (e^\alpha - e^{-\alpha}) \sum_{i=1}^{(i-1)} w_i \right\} \mathbf{1}_{[g \neq y_i]} \\
 & g = \text{argmin}_g \left\{ \sum_{i=1}^n w_i \mathbf{1}_{[g_i \neq y_i]} \right\}
 \end{aligned}$$

$$\begin{aligned}
 &= \arg \min_{\alpha, g} \sum_{i=1}^n w_i^{(m-1)} \exp \{-g \alpha g\} \\
 &\quad \arg \min_{\alpha} \sum_{g=y}^{m-1} w_i^{(m-1)} \exp \{-g \alpha g\} \quad \frac{\partial L}{\partial \alpha} = \sum \\
 &\quad y=g \rightarrow \sum_{g=y}^{m-1} w_i^{(m-1)} \exp \{-\alpha\} \\
 &\quad y \neq g \rightarrow \sum_{g \neq y}^{m-1} w_i^{(m-1)} \exp \{\alpha\} \\
 &\frac{\partial L}{\partial \alpha} = - \sum_{g=y}^{m-1} w_i^{(m-1)} \exp \{-\alpha\} + \sum_{g \neq y}^{m-1} w_i^{(m-1)} \exp \{\alpha\} = 0 \\
 &\text{Multiply both terms for } e^\alpha \\
 &\quad = - \sum_{g=y}^{m-1} w_i^{(m-1)} + \sum_{g \neq y}^{m-1} w_i^{(m-1)} \exp \{2\alpha\} = 0 \\
 &e^{2\alpha} \sum_{g \neq y}^{m-1} w_i^{(m-1)} = \sum_{g=y}^{m-1} w_i^{(m-1)} \\
 &e^{2\alpha} = \frac{\sum_{i=1}^n w_i^{(m-1)} - \sum_{g \neq y}^{m-1} w_i^{(m-1)}}{\sum_{g \neq y}^{m-1} w_i^{(m-1)} + \sum_{g=y}^{m-1} w_i^{(m-1)}} \\
 &\alpha = \frac{1}{2} \log \left( \frac{1 - \text{err}}{\text{err}} \right) \\
 \text{where } \text{err} &= \frac{\sum_{g \neq y}^{m-1} w_i^{(m-1)}}{\sum_{i=1}^n w_i^{(m-1)}} \underline{1(g \neq y)}
 \end{aligned}$$

- $\hat{g}^{(m)}$  minimizer of the weighted missclassification
- $\alpha = \frac{1}{2} \log \left( \frac{1 - \text{err}}{\text{err}} \right)$

Our general classifier is updated as

$$\hat{f}^{(t+1)} = \hat{f}^{(t)} + \alpha_m \hat{g}^{(m)}$$

which causes the weights of the next iteration to be

$$w_i^{(t+1)} = w_i^{(t)} \cdot \exp \left\{ -\alpha y_i \hat{g}_i^{(m)} \right\}$$

Since  $\hat{g}_i^{(m)} = \sum_{g_j \neq y_i} 1 + \sum_{g_j = y_i} (+1) - \sum_{g_j \neq y_i} 1$

$$= 2 \sum_{g_j \neq y_i} 1 - 1$$

$$\begin{aligned} w_i^{(t+1)} &= w_i^{(t)} \cdot \exp \left\{ \alpha \left( 2 \sum_{g_j \neq y_i} 1 - 1 \right) \right\} \\ &= w_i^{(t)} \exp \left\{ 2\alpha \sum_{g_j \neq y_i} 1 - \alpha \right\} \\ &= w_i^{(t)} e^{-\alpha} e^{2\alpha \sum_{g_j \neq y_i} 1} \end{aligned}$$

constant for each observation  
→ can be ignored

$2\alpha$  is  $\alpha$  in the algorithm

→ AdaBoost minimizes the exponential loss criterion by a forward stagewise procedure

Note:

- this statistical view allows us to interpret the results of the procedure.

In particular, it can be shown that the minimizer of the exponential loss

$$f^*(x) = \arg \min_{f(x)} E_{Y|x} [e^{-y f(x)}] = \frac{1}{2} \log \frac{\Pr[Y=1|x]}{\Pr[Y=-1|x]}$$

$$\text{alternatively: } \Pr[Y=1|x] = \frac{1}{1+e^{-f^*(x)}}$$

$\frac{1}{2}$  the log-odds for  $\Pr[Y=1|x]$  → Ex 10.2

- other loss-functions lead to the same minimizer, for example the negative log-likelihood

$$\hat{\pi} = \Pr[Y=1|x] = \frac{e^{f(x)}}{e^{f(x)} + e^{-f(x)}} = \frac{1}{1+e^{-2f(x)}}$$

$$\cdot y' = \frac{y+1}{2} \in \{0;1\}$$

then

$$l(\hat{\pi}) = y' \log \hat{\pi} + (1-y') \log (1-\hat{\pi})$$

$$\Rightarrow -l(\hat{\pi}) = \log \left( 1 + e^{-2y' f(x)} \right)$$



## Exercise 10.2

$$f^*(x) = \arg \min_{f(x)} E_{Y|x} [e^{-Y f(x)}]$$

$$\frac{\partial}{\partial f(x)} E_{Y|x} [e^{-Y f(x)}] = E_{Y|x} [-Y e^{-Y f(x)}]$$

$$E_{Y|x} [-Y e^{-Y f(x)}] = 0 \quad Y = \begin{cases} -1 & \Pr[Y = -1|x] \\ 1 & \Pr[Y = 1|x] \end{cases}$$

$$+ (1) e^{-(+1)f(x)} \Pr[Y = -1|x] - (1) e^{-(-1)f(x)} \Pr[Y = 1|x] = 0$$

$$e^{f(x)} e^{f(x)} \Pr[Y = -1|x] = e^{-f(x)} \Pr[Y = 1|x] \quad \text{multiply both sides by } e^{R(x)}$$

$$e^{2f(x)} \Pr[Y = -1|x] = \Pr[Y = 1|x]$$

$$e^{2f(x)} = \frac{\Pr[Y = 1|x]}{\Pr[Y = -1|x]}$$

$$f(x) = \frac{1}{2} \log \left( \frac{\Pr[Y = 1|x]}{\Pr[Y = -1|x]} \right)$$

## Exercise 10.4

$$X = (x_1, \dots, x_{10}) \quad x_j \sim N(0; 1) \quad x_j \text{ iid} \quad N = 2000$$

$$Y_i = \begin{cases} 1 & \text{if } \sum_{j=1}^{10} x_{ij}^2 > \chi_{10}^2(0.5) \\ -1 & \text{if } \sum_{j=1}^{10} x_{ij}^2 \leq \chi_{10}^2(0.5) \end{cases}$$

$\rightarrow$  obs  $\left\{ \begin{array}{c} 1 \\ 2 \\ \vdots \\ N \end{array} \right\} \quad \begin{array}{ccccccccc} \bar{x}_1 & \bar{x}_2 & \dots & \bar{x}_{10} & \text{apply } (\underline{x^2}, \underline{1}, \underline{\text{sum}}) \\ \underbrace{\bar{x}_{11} & \bar{x}_{12} & \dots & \bar{x}_{10} } & & & & \end{array}$

Statistical boosting { gradient boosting  
likelihood-based boosting

From the previous lecture:

AdeBoost : classifier

- weak estimator

- loss function

- iteratively apply a weak estimator to modifications of the data in order to minimize a loss function

↳ AdeBoost : weight more missclassified observations

AdeBoost : - stump  
- tree  
- ...

for classification  $\rightarrow$  AdeBoost<sup>t</sup>  
 $e^{-Y_t w_t}$

from classification to regression

- loss function : RSS LS estimator

- weak estimator :  $\hat{y} = \underline{\nu(X^T X)^{-1} X^T y}$   $\nu \rightarrow \alpha$

penalty parameter  $0 < \nu < 1$  weak estimator

↳ makes our LS "weak" default = 0.1

- modification of the data :  $y \rightarrow \nu$  residuals

focusing on the nat explaining part of the variation outcome

$L_2$  Boost algorithm for linear regression

① Initialization: initialize the regression coefficient estimate  $\hat{\beta}^{[0]} = (0, \dots, 0)$

(first modification of the data:  $v = y - X\beta = y - 0 = y$ )

② for  $m$  from 1 to  $m\_stop$

a) fit the weak estimator to the modification of the data

$$\hat{b}^{[m]} = \nu (X^T X)^{-1} X^T v$$

b) update the estimate  $\hat{\beta}^{[m]} = \hat{\beta}^{[m-1]} + \hat{b}^{[m]}$

c) modify the data:  $v = y - X^T \hat{\beta}^{[m]}$

③ final estimate

$$\hat{\beta}_{L_2\text{Boost}} = \sum_{m=1}^{m_{\text{stop}}} \hat{b}^{[m]} = \hat{\beta}^{[m_{\text{stop}}]}$$

Note:

$$m \rightarrow \infty, \hat{\beta}_{L_2\text{Boost}} \rightarrow \hat{\beta}_{\text{OLS}}$$

need of an early stop (find the "right"  $m_{\text{stop}}$ ) in order to not overfit (to find the best balance between bias and variance for the prediction error)

-  $m\_stop$  is the crucial tuning parameter

if it is too small: too much bias (our model does not explain the outcome variation)

if it is too big: too much variance (we overfit the data)

Complexity parameter

boosting has a second tuning parameter,  $\nu$  (is not so important, because smaller values  $\Rightarrow$  more steps (iterations))  
larger values  $\rightarrow$  less steps

$X$  must be centered  $E[X_j] = 0$   
(it is ok to standardize)

$L_2$  Boost algorithm in general

- the goal is to minimize the loss function. At each step we want to identify the direction of the greatest decrease of the loss function

and negative gradient :

$$-\frac{\partial L(y, f(x))}{\partial f(x)}$$

$$\text{e.g. } \frac{\partial \sum_{i=1}^n (y_i - x_i^\top \beta)^2}{\partial x^\top \beta} = \cancel{\frac{n}{2}} \underbrace{\sum_{i=1}^n (y_i - x_i^\top \hat{\beta})}_{\text{residuals}}$$

$$\frac{1}{n} \exp\left\{-\frac{1}{2}(y - x^\top \beta)\right\}$$

In general:

① Initialization :  $\hat{f}(x) = 0$  or  $\hat{f}(x) = \bar{y}$

② for  $m$  from 1 to  $m$  stop

(a) derive  $b = -\frac{\partial L(y, \hat{f}(x))}{\partial \hat{f}(x)}$

↑ RSS lin. reg.

(b) fit our weak estimator :  $\hat{g} = g(v, x, 0)$

↑ lin. reg. residuals

(c) update the estimate  $\hat{f}^{[m]} = \hat{f}^{[m-1]} + \hat{g}$

③ Finalization  $\hat{f}_{\text{boost}} = \hat{f}^{[\text{last}]} \quad [\text{stop}]$

GAM :  $g = \sum_{j=1}^P f_j(x_j)$

$L_2$  Boost for High Dimensional Data

- one of the advantages of boosting is that we can handle HAD
- component wise version of boosting

## Componentwise Boosting

- Linear regression model

① Initialization  $\hat{\beta}_j^{(0)} = 0 \quad j = 1, \dots, p$   
 $(v = y - X\hat{\beta}^{(0)}) = y$

② For  $m$  from 1 to  $m$ -step

- compute possible updates for each dimension of the regression coefficient vector separately  
 (fit a **weak** estimator on each dimension of  $X$ )

$$\hat{b}_j = \frac{\sum_{i=1}^n x_i^{(j)} v_i}{\sum_{i=1}^n x_i^{(j)2}} \quad j = 1, \dots, p$$

- select the best update among the  $p$  possibilities

$$j^* : \arg \min_j \sum_{i=1}^n (v_i - x_i^{(j)} b_j)^2$$

- update the  $j^*$ -th regression coefficient

$$\hat{\beta}^{(m)} = \hat{\beta}^{(m-1)} + (0, \dots, 0, \hat{b}_{j^*}, 0, \dots, 0)$$

- modify the data  $v = y - X\hat{\beta}^{(m)}$

Fit a GAM in R with Boosting

$$y = f_1(x_1) + f_2(x_2) + \dots + f_p(x_p)$$

splines      tree      → function `mboost` of the package `mboost`

$$\mu(y) = X^\top \beta \rightarrow \text{glmboost}$$

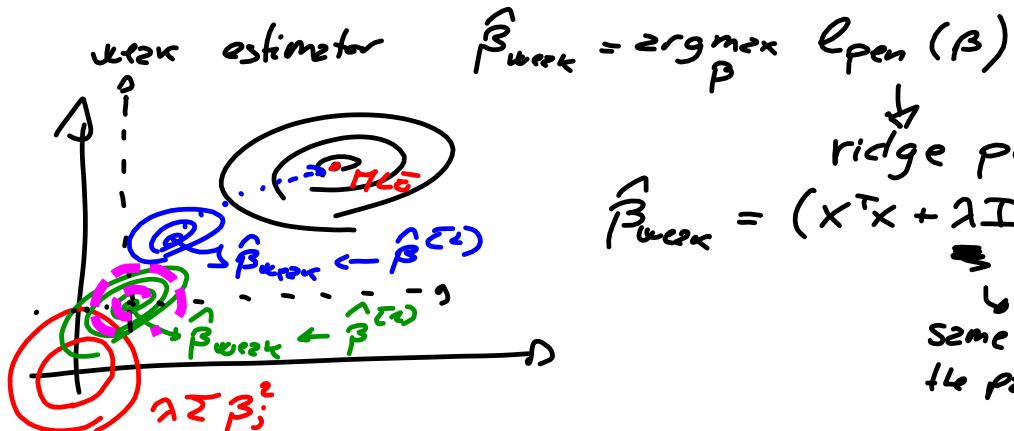
R Gaussian regression,  
logistic regression

:

## Likelihood-based boosting

- fully statistical approach vs likelihood-based
- weak estimator:

standard estimator  $\hat{\beta}_{MLE} = \arg \max_{\beta} \ell(\beta)$



weak estimator  $\hat{\beta}_{weak} = \arg \max_{\beta} \ell_{pen}(\beta)$

ridge penalty

$$\hat{\beta}_{Ridge} = (X^T X + \lambda I)^{-1} X^T y$$

same meaning of the parameter  $\lambda$

## Boosting Ridge (Gaussian regression)

- initialize  $\hat{\beta}^{[1]} = (0, \dots, 0) \rightarrow v = y - X\hat{\beta}^{[1]}$
- fit the weak estimator  $\hat{b} = (X^T X + \lambda I)^{-1} X^T y$

$$\frac{\partial \ell_{pen}(\beta)}{\partial \beta} = 0 \quad \ell_{pen}(y - X\beta)(y - X\beta)^T + \lambda \beta \beta^T = 0$$

$\frac{\partial \ell_{pen}}{\partial \beta} : -X^T(y - X\beta) + \lambda \beta = 0$

$-X^T y + X^T X \beta + \lambda \beta = 0$

$\beta(X^T X + \lambda I) = X^T y$

$\hat{b} = \hat{\beta} = (X^T X + \lambda I)^{-1} X^T y$

*Kernel of 2 Gaussian log-likelihood*

$$\exp \left\{ -\frac{1}{2} (y - X\beta)(y - X\beta)^T \right\}$$

(b)  $\hat{\beta}^{[n]} = \hat{\beta}^{[n-1]} + \hat{b}$

- (c) modification of the data  
(add an offset in the log-likelihood)

$$\frac{1}{2} (y - X\hat{\beta}^{[n]})^T - X\beta (y - X\hat{\beta}^{[n]}) - X\beta)^T$$

3  $\hat{\beta}_{\text{Boost}} = \sum_{n=1}^{m, \text{step}} \hat{b}^{[n]}$

weak estimator is the ridge estimator  
 Gaussian regression  $\rightarrow$  likelihood-based boosting  $\rightarrow$  some model when we choose  $\psi$  and  $\lambda$  in the right way  
 gradient boosting

### Exercise

$$\hat{y} = X \hat{\beta}_{\text{boost}} = \sum_{m=1}^{m_{\text{step}}} S(I - S)^m y = I - (I - S)^{m_{\text{step}}+1} y$$

where  $S = X b$

using  $b = (X^T X + \lambda I)^{-1} X^T y$  Boosting with ridge estimator

Show through SVD that Boosting ridge and ridge regression provide different shrinkage effect

ridge regression :  $\frac{d_j^2}{d_j^2 + \lambda}$

Boosting Ridge :  $(1 - (1 - \frac{d_j^2}{d_j^2 + \lambda}))^{m_{\text{step}}+1})$

## Likelihood-based boosting

- Loss function
- weak estimator
- modification of the data

### $L_2$ Boost

$$\hat{y}_j = \frac{\partial L(y, f(x))}{\partial f(x)}$$

penalized version of OLS

$$\hat{y}(X^T X)^{-1} X^T u$$

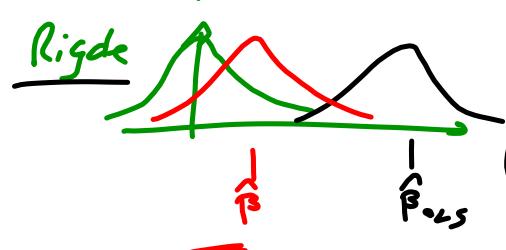
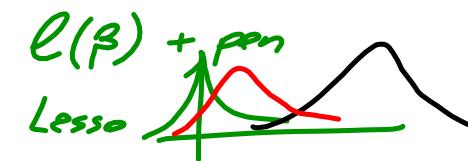
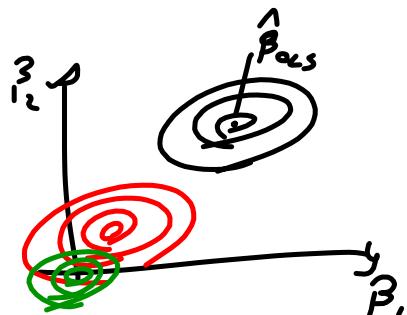
$u \propto \hat{L}_2$  OLS minimizer  
 $\hat{L}_2$  loss

$\hat{L}_2$  is negative log-likelihood

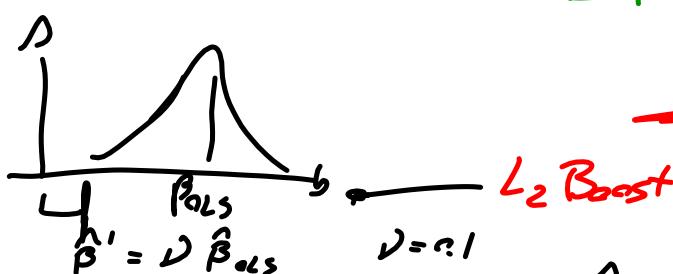
parametric version

$$L(y, f(x, \beta)) := \text{negative log-likelihood}$$

weak estimator  $\rightarrow$  instead of minimizing the negative log-likelihood, we minimize = penalized version of it



$$\hat{\beta}_{Weak} = (X^T X + \lambda I)^{-1} X^T u$$



$$\begin{aligned}\hat{\beta}_{Boost}^{(1)} &= \hat{\beta}^{(0)} + \hat{\beta}^{(1)} \\ \hat{\beta}_{Boost}^{(2)} &= \hat{\beta}^{(0)} + \hat{\beta}^{(1)} + \hat{\beta}^{(2)}\end{aligned}$$

## Algorithm

1) Initialization  $\hat{\beta} = \underline{0}$   $v = y$

2) for  $m$  from 1 to  $m_{\text{steps}}$   
 $\ell_{\text{pen}}(0 + \beta; 1)$

$$\underbrace{\frac{\partial \ell_{\text{pen}}(0 + \beta; 1)}{\partial \beta} = 0}_{\text{Ridge penalty}} \quad (X^T X + I \lambda)^{-1} X^T v$$

3)  $\hat{b}^{(m)} = (X^T X + \lambda I)^{-1} X^T v$

$$\ell_{\text{pen}}(\sum \hat{b}^{(m)} + \beta; 1)$$

Gaussian case  $\ell_{\text{pen}}(\sum b^{(m)} + \beta; 1) =$

$$(y - \sum \hat{b}^{(m)} X - X\beta)^T (y - \sum \hat{b}^{(m)} X - X\beta) + \lambda \beta \beta^T$$

## Exercise

$$\hat{f}(x) = \underline{\underline{\hat{\beta}}}$$

$$\hat{f}_b^{(0)} = X(X^T X + \lambda I)^{-1} X^T y - S y$$

$$\hat{f}_{\text{boost}}^{(1)} = X(X^T X + \lambda I)^{-1} X^T u$$

$$\hat{f}_{\text{boost}}^{(1)} = y - X \hat{\beta}^{(0)}$$

$$y - X(X^T X + \lambda I)^{-1} X^T y$$

$$= X(X^T X + \lambda I)^{-1} X^T (y - X(X^T X + \lambda I)^{-1} X^T y)$$

$$= S(I - S)y$$

$$\hat{f}_{\text{boost}}^{(2)} = X(X^T X + \lambda I)^{-1} X^T u$$

$$= X(X^T X + \lambda I)^{-1} X^T (y - \underline{X \hat{b}^{(0)}} - \underline{X \hat{b}^{(1)}})$$

$$= X(X^T X + \lambda I)^{-1} X^T (y - \underline{X(X^T X + \lambda I)^{-1} (y - X(X^T X + \lambda I)^{-1} X^T y)})$$

$$= X(X^T X + \lambda I)^{-1} X^T y$$

$$= S(y - S(I - S)y - S y)$$

$$= S(I - S + S^2 - S) y = \underline{S(I - S)^2} y$$

⋮

$$\hat{f}_b^{(m)} = S(I - S)^m y$$

m-th improvement

$$\hat{\beta}_{\text{boost}}^{(m-\text{step})} = \sum_{j=0}^{m-\text{step}} \hat{b}^{(j)}$$

$$\hat{y} = X \hat{\beta}^{(m-\text{step})} = \sum_{j=0}^{m-\text{step}} S(I - S)^j y = \underbrace{(I - (I - S)^{m-\text{step}+1})}_{H_m} y$$

$$\hat{y} = H_m y \text{ where } H = I - (I - S)^{m\_stop+1}$$

$$= \underline{I} - \underline{(I - X(X^T X + \lambda I)^{-1} X^+)^{m\_stop+1}}$$

use Singular Value Decomposition

$$X = UDV^T \text{ where } U \text{ spans the column space of } X, U^T U = I$$

$V$  has  $n$  rows,  $V^T V = VV^T = I$

$D$  diagonal matrix with singular values  
 $d_1 \geq d_2 \geq \dots \geq d_p$

$$\begin{aligned} H_m &= I - (I - UDV^T (UDU^T UDV^T + \lambda I)^{-1} VDU^T)^{m\_stop+1} \\ &= I - (I - UDV^T (D^2 + \lambda I)^{-1} VDU^T)^{m\_stop+1} \\ &= I - (I - U D^2 (D^2 + \lambda I)^{-1} U^T)^{m\_stop} \\ &\vdots \\ &= U (I - (I - \tilde{D})^{m\_stop+1}) U^T \quad \tilde{D} = (D^2 + \lambda I)^{-1} D^2 \\ &\quad \tilde{d}_j^2 = \frac{d_j^2}{d_j^2 + \lambda I} \end{aligned}$$

Ridge estimator (from lecture 4, notes page 8)

$$\text{ridge } X\hat{\beta}_{\text{ridge}} = U D^2 (D^2 + \lambda I)^{-1} U^T$$

$$\text{ridge-boosting } X\hat{\beta}_{\text{boost}} = U (I - (I - D^2 (D^2 + \lambda I)^{-1})^{m\_stop+1}) U^T$$

$$\text{no } \gamma, \lambda, m\_stop \text{ s.t. } D^2 (D^2 + \lambda I)^{-1} = I - (I - D^2 (D^2 + \lambda I)^{-1})^{m\_stop+1}$$

$$m\_stop = 0.$$

apply only to first time  
the weak estimator (ridge)

$$\begin{aligned} D^2 (D^2 + \lambda I)^{-1} &= I - (I - D^2 (D^2 + \lambda I)^{-1})^{m\_stop+1} \\ &= I - I + D^2 (D^2 + \lambda I)^{-1} \\ &= \lambda \end{aligned}$$

$L_2$  Boost is an algorithm that goes under the umbrella of gradient boosting  
 + likelihood-based boosting }  $\rightarrow$  statistical boosting

$L_2$  Boost : • Bühlmann & Yu (2003)

Boosting with the  $L_2$  loss: regression and classification

• Bühlmann & Hothorn (2007)

Boosting algorithms: regularization, prediction and model fitting

likelihood-based boosting : • Tutz & Binder (2006, 2007)

Boosting ridge regression

CAT with implicit variable selection  
 by likelihood-based boosting

• general on statistical boosting: Mller et al. (2004)

The evolution of boosting algorithms

## Advance regression and classification

### Classification :

- KNN
- linear regression for classification
- LDA, QDA
- logistic regression
- AdaBoost
- $L_2$  Boost for classification (gradient boosting)

KNN  
logistic  
regression

### Regression :

- linear regression and the OLS estimator
- penalized regression methods
  - LASSO ( $L_1$  penalty:  $\lambda \sum_{j=1}^p |\beta_j|$ )
  - RIDGE ( $L_2$  penalty:  $\lambda \sum_{j=1}^p \beta_j^2$ )
  - Elastic-net (combination between  $L_1$  and  $L_2$  penalties)
  - Boosting { $L_2$  Boost  
likelihood-based boosting}
  - LAR
- methods which use derived inputs
  - PCR
  - PLS

Central concept : bias-variance trade-off

OLS  $\rightarrow$  have 0 bias, minimize the variance

(Gauss-Markov theorem: OLS is BLUE)

e i g n t r n s g +

We saw approaches which accept an increase in bias to have a (larger) decrease in variance

$$E[(Y_o - \hat{f}(x_o))^2] = E[Y_o^2 - 2Y_o \hat{f}(x_o) + \hat{f}(x_o)^2]$$

$$\begin{aligned} &= E[Y_o^2] - E[\hat{f}(x_o)]^2 + E[\hat{f}(x_o)^2] - E[\hat{f}(x_o)]^2 + E[\hat{f}(x_o)]^2 \\ &\quad - 2(E[Y_o \hat{f}(x_o)]) - E[Y_o]E[\hat{f}(x_o)] + E[Y_o]E[\hat{f}(x_o)] \end{aligned}$$

$\text{cov}(Y_o, \hat{f}(x_o)) = 0$

$$= \sigma^2 + \text{Var}(\hat{f}(x_o)) + (E[\hat{f}(x_o)] - \hat{f}(x_o))^2$$

$\stackrel{\text{= irreducible error}}{=} \text{variance} + \text{bias}^2$

$$Y = f(x_o) + \varepsilon \quad f(x_o) \perp \varepsilon$$

- prediction error
  - training error (underestimating the prediction error)
  - test error (computed on independent data)

- comparing methods based on prediction error
  - training error + estimate optimiser of the training error
    - AIC
    - BIC
  - try to estimate the test error
    - Cross-validation
    - Bootstrap approaches (0.632 and 0.632+ bootstrap)
- model selection
  - backward elimination ... 2nd combinations ↗ stepwise selection
  - forward selection ↗ stepback selection
  - best subset Selection
  - stopping criteria (AIC, BIC, significance level ↗ F-test)

## ~~splines~~

- problems with high-dimensional data
  - classical approaches (OLS) do not work
    - { - add constraints (LASSO - RIDGE - ...)
    - stagewise approaches (boosting)
      - || update only one dimension per time
- independence between training and test set
 

training	validation	test	independent
----------	------------	------	-------------

fit models ↑                      choose the model ↑                      evaluate ←

  - CV:  $k-1$  fold and the remaining fold are totally independent  
training/test split

- when we apply a method which require standardization, we first standardize the training set (or the set which temporarily acts as a training set) without involving the observations in the test set. Then the observations on the test set will be "standardized" using means and standard deviations computed on the training set.

Exam 2015

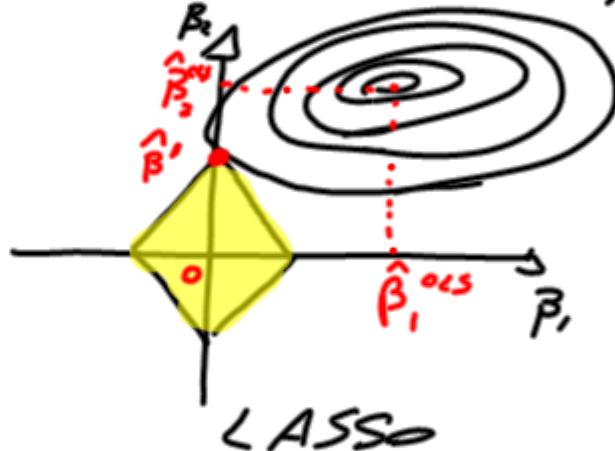
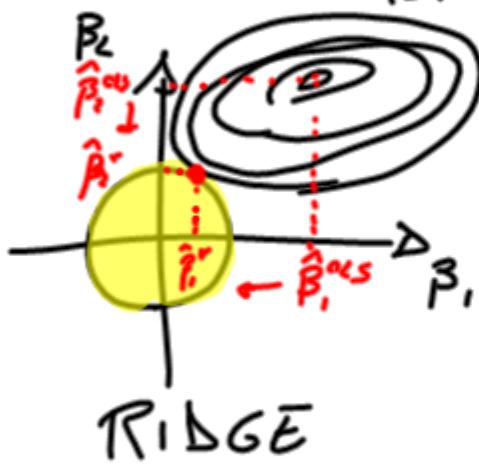
Exercise 1:  
a) 1: Lasso

2: ridge regression

$$\text{PRSS}_{\lambda}^{\text{Lasso}} = \sum_{i=1}^N (y_i - \beta_1 x_1 - \beta_2 x_2)^2 + \lambda(|\beta_1| + |\beta_2|)$$

$$\text{PRSS}_{\lambda}^{\text{ridge}} = \sum_{i=1}^N (y_i - \beta_1 x_1 - \beta_2 x_2)^2 + \lambda(\beta_1^2 + \beta_2^2)$$

b)



Exercise 2

a)  $\hat{\beta}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y$

$\hat{\beta}_{\text{OLS}} = (X^T X)^{-1} X^T y$

$$\begin{aligned}\hat{\beta}_{\text{ridge}} &= \underbrace{(X^T X + \lambda I)^{-1} (X^T X)}_A \underbrace{(X^T X)^{-1} X^T y}_{\hat{\beta}_{\text{OLS}}} \\ &= A \hat{\beta}_{\text{OLS}}\end{aligned}$$

$$b) N \rightarrow \infty, X^T X \approx N \Sigma$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$\hat{\beta}_{\text{ridge}} = A \hat{\beta}_{\text{OLS}} = (X^T X + \lambda I)^{-1} X^T \hat{\beta}_{\text{OLS}}$$

$$\underset{N \text{ large}}{\approx} (N \Sigma + \lambda I)^{-1} N \Sigma \hat{\beta}_{\text{OLS}} \quad \Sigma = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$= (\Sigma + \frac{\lambda I}{N})^{-1} \Sigma \hat{\beta}_{\text{OLS}}$$

$$= \begin{pmatrix} 1 + \frac{\lambda}{N} & 1 \\ 1 & 1 + \frac{\lambda}{N} \end{pmatrix}^{-1} \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \begin{pmatrix} \hat{\beta}_{\text{OLS}}^{(1)} \\ \hat{\beta}_{\text{OLS}}^{(2)} \end{pmatrix}$$

$$= \frac{1}{(1 + \frac{\lambda}{N})^2 - \rho^2} \begin{pmatrix} 1 + \frac{\lambda}{N} & -\rho \\ -\rho & 1 + \frac{\lambda}{N} \end{pmatrix} \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \begin{pmatrix} \hat{\beta}_{\text{OLS}}^{(1)} \\ \hat{\beta}_{\text{OLS}}^{(2)} \end{pmatrix}$$

$$= \frac{1}{(1 + \frac{\lambda}{N})^2 - \rho^2} \begin{pmatrix} 1 + \frac{\lambda}{N} - \rho^2 & \lambda + \frac{\lambda}{N} \rho - \lambda \\ \frac{\lambda}{N} \rho & 1 + \frac{\lambda}{N} - \rho^2 \end{pmatrix} \begin{pmatrix} \hat{\beta}_{\text{OLS}}^{(1)} \\ \hat{\beta}_{\text{OLS}}^{(2)} \end{pmatrix}$$

$$\hat{\beta}_{\text{ridge}}^{(1)} = \frac{1}{(1 + \frac{\lambda}{N})^2 - \rho^2} \left[ \left( 1 + \frac{\lambda}{N} - \rho^2 \right) \hat{\beta}_{\text{OLS}}^{(1)} + \frac{\lambda}{N} \rho \hat{\beta}_{\text{OLS}}^{(2)} \right]$$

$$\hat{\beta}_{\text{ridge}}^{(2)} = \frac{1}{(1 + \frac{\lambda}{N})^2 - \rho^2} \left[ \frac{\lambda}{N} \hat{\beta}_{\text{OLS}}^{(1)} + \left( 1 + \frac{\lambda}{N} - \rho^2 \right) \hat{\beta}_{\text{OLS}}^{(2)} \right]$$

c) Lasso part : exploit variable selection property of LASSO (i.e., remove irrelevant predictors)

Ridge part : better handle of correlation structures in  $X$

### Exercise 3

a) K-fold cross-validation



training set  
validation set

1) for each value of  $\lambda$ ,  $\kappa = 1$

$$\hat{\beta}_{\text{less}, \kappa}(\lambda) \quad \begin{array}{c} \lambda - K \\ \hline \end{array} \quad \begin{array}{c} \text{training} \\ \text{validation} \end{array}$$

$$\underline{\text{PECV}_n(\lambda)} = L\left(y^{\text{tr}}, \hat{x}^{\lambda - K} \underline{\hat{\beta}_{\text{less}, \kappa}(\lambda)}\right)$$

2)  $\kappa = 2$

:

$$\text{final } \text{PECV}_{\text{tot}}(\lambda) = \sum_{\kappa=1}^K \text{PECV}_{\kappa}(\lambda)$$

$$\lambda^* = \underset{\lambda}{\operatorname{arg\,min}} \{ \text{PECV}_{\text{tot}}(\lambda) \}$$

b) 2-fold CV : more bias, less variance

$\text{Loocv}$  : less bias, more variance

2 additional aspects : Loocv computationally more intense  
Loocv deterministic

## Exercise 4

a) Boosting :  
    "repeatedly apply a weak estimation to  
    modifications of the data to minimize  
    a loss function"  
-iterative procedure

Bagging : bootstrap samples  $x_i^* = (x_{i1}^*, x_{i2}^*, \dots, x_{in}^*)$   
 $\hat{\beta}_b$  on  $x_i^*$   
 $\hat{\beta}_{\text{BAGGING}} = \frac{1}{B} \sum_{b=1}^B \hat{\beta}_b$

- b) not in the program this year
- c) point a) apply the weak estimator on the reweighted observations
- point d)  $\sum_{m=1}^{n_{\text{step}}} \alpha_m \mathbf{1}[y_i \neq G_m(x_i)]$
- point e)  $\sum_{m=1}^{n_{\text{step}}} \alpha_m G_m(x)$