Linear Algebra revision

Eigenvalues / eigenvectors / Singular Value Decomposition

- A: real-valued square matrix

$$p \times p$$

We will assume that A has full rank

$$\text{rank}(A) = p$$

[A has p-linearly independent columns (A is invertible)

We say that λ_i is an <u>eigenvalue</u> of A with v_i being the corresponding <u>eigenvector</u> if

$$\boxed{A v_i = \lambda_i v_i}$$

The eigenvalues of A are the roots of its characteristic polynomial

$$\det(A - \lambda I) = 0$$

<u>Example</u>

$$A = \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix}, \quad A - \lambda I = \begin{bmatrix} -\lambda & 1 \\ 2 & 1-\lambda \end{bmatrix}$$

$$\det(A - \lambda I) = \lambda(\lambda - 1) - 2 = \lambda^2 - \lambda - 2$$

$$\lambda_1 = 2 \qquad \lambda_1 = -1$$

$\lambda_1 = 2$

$$A v_1 = \lambda_1 v_1 \rightsquigarrow (A - \lambda I) v_1 = 0$$

$$\begin{bmatrix} -2 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} v_1^{(1)} \\ v_1^{(2)} \end{bmatrix} = 0 \quad \text{---} \quad v_1 = \begin{bmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{bmatrix}$$

Typically we select the eigenvector that has norm 1

$$\|v_1\|_2 = \sqrt{\left(\frac{1}{\sqrt{5}}\right)^2 + \left(\frac{2}{\sqrt{5}}\right)^2} = 1$$

- Because A is invertible it has no zero eigenvalues $(\det(A - \lambda I) = \det(A) \neq 0)$

- The matrices we will be typically working with they will have no repeated eigenvalues.

- We will also assume that all the corresponding eigenvectors are linearly independent (*)

So we have

$$A v_1 = \lambda_1 v_1$$

$$A v_2 = \lambda_2 v_2$$

$$\vdots$$

$$A v_p = \lambda_p v_p$$

$$\underbrace{\begin{bmatrix} Av_1 & Av_2 & \cdots & Av_p \end{bmatrix}}_{p \times p} = \underbrace{\begin{bmatrix} \lambda_1 v_1 & \lambda_2 v_2 & \cdots & \lambda_p v_p \end{bmatrix}}_{p \times p}$$

$$A \underbrace{\begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_p \\ | & | & & | \end{bmatrix}}_{Q} = \underbrace{\begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_p \\ | & | & & | \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} \lambda_1 & 0 & & \\ 0 & \lambda_2 & \cdots & \\ & & & \lambda_p \end{bmatrix}}_{\Lambda}$$

$$\Rightarrow \quad AQ = Q\Lambda$$

$$AQQ^{-1} = Q\Lambda Q^{-1}$$

$$\boxed{A = Q\Lambda Q^{-1}}$$

Because of (*) Q is invertible!

→ diagonalisation of A

Now suppose in addition that A is symmetric

$$(A = A^T)$$

$\Rightarrow$ then the eigenvectors corresponding to different eigenvalues are orthogonal

$$\left.\begin{aligned} Av_1 &= \lambda_1 v_1 \\ Av_2 &= \lambda_2 v_2 \end{aligned}\right\} \Rightarrow v_1 \perp v_2 \quad \left(\begin{aligned} v_1 \cdot v_2 &= 0 \\ v_1^T \cdot v_2 &= 0 \end{aligned}\right)$$

$$\underbrace{\begin{bmatrix} v_1^T \end{bmatrix}}_{1\times p} \underbrace{\begin{bmatrix} | \\ v_2 \\ | \end{bmatrix}}_{p\times 1} \rightarrow \underbrace{\quad}_{1\times 1}$$

these eigenvectors can also be chosen to have norm equal to 1

$$\Rightarrow \begin{cases} v_i^T v_j = 0 & i \neq j \\ v_i^T v_i = \|v_i\|_2^2 = 1 & \end{cases}$$

We call this an orthonormal set of eigenvectors

In matrix form

$$A = Q \Lambda Q^{-1}$$

$$Q = \left[v_1 \,\middle|\, \cdots \,\middle|\, v_p \right], \quad Q^T = \left[\begin{array}{c} v_1^T \\ \hline \vdots \\ \hline v_p^T \end{array} \right]$$

$$Q^T Q = \begin{bmatrix} 1 & & 0 \\ & 1 & \\ 0 & & \ddots \; 1 \end{bmatrix} \quad (\text{check } Q^T Q = I)$$

But this means

$$Q^{-1} = Q^T$$

$$\Rightarrow \quad A = Q \Lambda Q^{-1} = Q \Lambda Q^T$$

$$\boxed{A = Q \Lambda Q^T}$$

spectral decomposition (or Karhunen-Loeve)

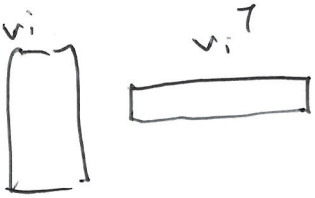
$$A = Q \Lambda Q^T$$

$$\left[v_1 \,\middle|\, \cdots \,\middle|\, v_p \right] \qquad \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_p \end{bmatrix} \qquad \left[\begin{array}{c} v_1^T \\ \hline \vdots \\ \hline v_p^T \end{array} \right]$$

$$A = Q \Lambda Q^T = \sum_{i=1}^{p} \lambda_i \, v_i v_i^T$$

v_i: $p \times 1$, v_i^T: $1 \times p$, $v_i v_i^T$: $p \times p$

(check at home)



Every $v_i v_i^T$ is a ~~rank-1~~ rank-1 matrix

Singular Value Decomposition

Consider an, in general, non-square matrix X of size $n \times p$ (real-valued)

- We assume that X has full-rank:

$$\text{rank}(X) = \min(n, p)$$

The singular value decomposition (SVD) is the following factorization:

$$\underset{n \times p}{X} = U \; D \; V^T$$

- U: $n \times \min(n,p)$
- D: $\min(n,p) \times \min(n,p)$
- V^T: $p \times \min(n,p)$ (V is $\min(n,p) \times p$)

D: is diagonal, contains all the singular values
In this case, these will be the square root of eigenvalues of XX^T.

(XX^T symmetric, all the eigenvalues are positive.)

U: Satisfies $U^T U = I$
The columns of U are eigenvectors of XX^T

V: Satisfies $V^T V = I$
The columns of V are eigenvectors of $X^T X$

Similarly as before we can write X as the sum of rank-1 matrices.

TMP

Important result

If X is square, symmetric, and positive (all the eigenvalues are real and positive)

then SVD coincides with the spectral decomposition.

Derivatives of linear and Quadratic forms

Linear forms

Recall: $f: \mathbb{R} \to \mathbb{R}$, $f(x) = ax$

$$f'(x) = \frac{df}{dx} = a$$

A is a matrix of size $n \times p$

We can naturally define the following function: $f: \mathbb{R}^p \to \mathbb{R}^n$ as

$$f(s) = \underbrace{\underset{n\times p}{A}\ \underset{p\times 1}{s}}_{n\times 1} \in \mathbb{R}^n, \quad s \in \mathbb{R}^p$$

$$s = (s_1, s_2, \dots, s_p)^T$$

$$f(s) = (f_1(s), f_2(s), \dots f_n(s))^T$$

$f_i: \mathbb{R}^p \to \mathbb{R}$ for every $i = 1, \dots, n$

$$\frac{\partial f}{\partial s}(s_0) = \begin{bmatrix} \frac{\partial f_1}{\partial s_1}(s_0) & \frac{\partial f_1}{\partial s_2}(s_0) & \cdots & \frac{\partial f_1}{\partial s_p}(s_0) \\ \frac{\partial f_2}{\partial s_1}(s_0) & \frac{\partial f_2}{\partial s_2}(s_0) & \cdots & \frac{\partial f_2}{\partial s_p}(s_0) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial s_1}(s_0) & \frac{\partial f_n}{\partial s_2}(s_0) & \cdots & \frac{\partial f_n}{\partial s_p}(s_0) \end{bmatrix}_{n\times p}$$

It turns out that for $f(s) = As$ (exercise)

$$\frac{\partial f}{\partial s}(s_0) = A \qquad \text{for every } s_0 \in \mathbb{R}^p$$

Example

$$A = \underset{n \times p}{\underset{1 \times 2}{\begin{bmatrix} 1 & 2 \end{bmatrix}}} \qquad n = 1 \qquad p = 2$$

$$f(s) = As = A \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = s_1 + 2 s_2$$

$$\frac{\partial (As)}{\partial s_1} = 1 \qquad \frac{\partial (As)}{\partial s_2} = 2$$

$$\frac{\partial (As)}{\partial s} = \begin{bmatrix} \frac{\partial (As)}{\partial s_1} & \frac{\partial As}{\partial s_2} \end{bmatrix} = \begin{bmatrix} 1 & 2 \end{bmatrix} = A$$

Quadratic forms

$f: \mathbb{R} \to \mathbb{R}$

$$f(x) = b x^2$$

$$f'(x) = \frac{d f(x)}{dx} = 2 b x$$

Let B be a symmetric $p \times p$ matrix

We can define $f: \mathbb{R}^p \to \mathbb{R}$

$$f(s) = s^T B s$$

$1 \times p \quad p \times p \quad p \times 1$

1×1

$$\frac{\partial f}{\partial s}(s_0) = \left[\frac{\partial f}{\partial s_1}(s_0) \quad \frac{\partial f}{\partial s_2}(s_0) \quad \cdots \quad \frac{\partial f}{\partial s_p}(s_0) \right]$$

It turns out:

$$\frac{\partial f}{\partial s}(s_0) = 2\, s_0^T B$$

$1 \times p \quad p \times p$

$1 \times p$

Note: If B was not symmetric $\frac{\partial f}{\partial s}(s_0) = s_0^T (B + B^T)$

Sometimes for ease of notation we will write this as a column vector:

$$\left[\frac{\partial f}{\partial s}(s_0)\right]^T = \left(2\, s_0^T B\right)^T = 2\, B^T (s_0^T)^T$$
$$= 2 B^T s_0$$
$$= 2 B s_0 \qquad (B = B^T)$$

Example

$$B = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}, \quad Bs = B \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} s_1 + 2s_2 \\ 2s_1 + s_2 \end{bmatrix}$$

$$f(s) = s^T B s = \begin{bmatrix} s_1 & s_2 \end{bmatrix} \begin{bmatrix} s_1 + 2s_2 \\ 2s_1 + s_2 \end{bmatrix}$$

$$= s_1^2 + s_2^2 + 4 s_1 s_2$$

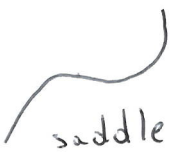
Then according to the definition $\frac{\partial f}{\partial s} = \left(\frac{\partial f}{\partial s_1} \quad \frac{\partial f}{\partial s_2} \right)$

$$= \begin{bmatrix} 2s_1 + 4s_2 & 2s_2 + 4s_1 \end{bmatrix} = 2 \begin{bmatrix} s_1 + 2s_2 & s_2 + 2s_1 \end{bmatrix}$$
$$= 2 \begin{bmatrix} s_1 & s_2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$
$$= 2\, s^T B \qquad \checkmark$$

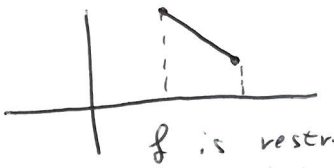
Computing extreme points of a function subject to constraints

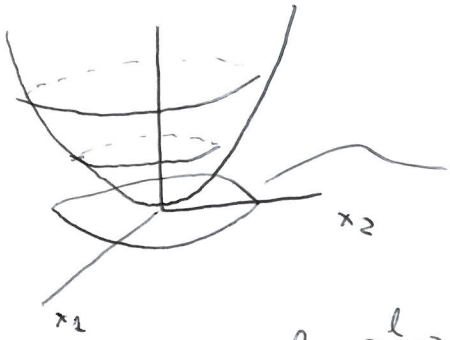
What do you think about the following phrase?

"Let f be a function. In order to find its minimum / maximum I just need to solve $f'(x)=0$"


saddle point


$f(x)=|x|$
non-diffs.


f is restricted to an interval



$f(x_1, x_2) = 2x_1^2 + x_2^2$

constraint: $x_1^2 + x_2^2 = 1$

We have a function $f: \mathbb{R}^l \to \mathbb{R}$. We are interested in computing extreme points (maximum or minimum)

subject to constraints: $h_1(x)=0$

$h_2(x)=0$

$\vdots$

$h_m(x)=0$

Here f and h_i, $i=1,\dots,m$ are differentiable

$$\begin{cases} \max\limits_{x} f(x) \\ \text{s.t.} \quad h_1(x)=0,\ h_2(x)=0,\ \dots,\ h_m(x)=0 \end{cases}$$

This means we are looking for $x^* \in \mathbb{R}^l$ such that $h_1(x^*) = h_2(x^*) = \dots = h_m(x^*) = 0$ and if $\tilde{x}$ is another such point then

$$f(x^*) \geq f(\tilde{x})$$

Example

$$\begin{cases} \max\limits_{x} & f(x) = x_1 + x_2 \\ \text{s.t.} & h(x) = x_1^2 + x_2^2 - 1 = 0 \end{cases} \qquad x = (x_1, x_2)$$

Lagrange multipliers method

a) Form the Lagrangian L

$$L(x) = f(x) - \sum_{j=1}^{m} \lambda_j h_j(x)$$

$f(x)$: target; λ_j: Lagrange multipliers; $h_j(x)$: j-th constraint

In the previous example ($m = 1$)

$$L(x) = L(x_1, x_2) = x_1 + x_2 - \lambda (x_1^2 + x_2^2 - 1)$$

Observation: L is a function of both x and λ ($\lambda_1, \dots \lambda_m$)

$$L(x, \lambda) = L(x_1, \dots, x_\ell, \lambda_1, \dots, \lambda_m)$$

~> Transformation of the original ~~problem~~ constrained problem into an unconstrained one.

b) Construct a system of $\ell + m$ equations.

$$\frac{\partial L}{\partial x} = 0$$

$$\frac{\partial L}{\partial \lambda} = 0$$

→ Derivative of L with respect to every x_i, $i = 1, \dots \ell$. Set every one of them to 0

$$\frac{\partial L}{\partial x} = \left(\frac{\partial L}{\partial x_1}, \frac{\partial L}{\partial x_2}, \dots, \frac{\partial L}{\partial x_\ell} \right) = (0, 0, \dots 0)$$

(ℓ equations)

→ Derivative of L with respect to every λ_i, $i = 1, \dots, m$. Set them to 0.

$$\frac{\partial L}{\partial \lambda} = \left(\frac{\partial L}{\partial \lambda_1}, \frac{\partial L}{\partial \lambda_2}, \dots, \frac{\partial L}{\partial \lambda_m} \right) = (0, 0 \dots 0)$$

(m equations)

c) Solutions of the above system of $\ell + m$ equations (x^*, λ^*) are candidates to solve the original constrained optimisation problem

Exercise

Apply b) to the previous example

$$L(x, \lambda) = x_1 + x_2 - \lambda (x_1^2 + x_2^2 - 1)$$

and you find candidate for extremes, and identify the maximum.

(Later on PCA):

$$\begin{cases} f(x_1, x_2) = [x_1 \; x_2]^T A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ \text{s.t.} \quad x_1^2 + x_2^2 = 1 \end{cases} \qquad A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

Review of some statistical concepts

Consider data in a matrix X of size $n \times p$

$n \times p$ → columns are variables

rows are different individuals (sample size)

Large n, p → "Big data"

$n < p$ → "Fat data"

Although entries of X could be categorical ("Yes"/"No") here we consider the simple case in which they are numbers.

Normally, $n >> p$ (much larger), i.e the number of individuals exceeds the number of variables.

Example

X: running times of 10 persons ($n=10$) in 3 different types of ground ($p=3$)

	Hill	Flat	Tarmac
1	19	17	19
2	29	21	25
⋮	⋮	⋮	⋮
10	26	18	29

X : $n \times p$

we have the following concepts

Population

- $X = (X_1, X_2, \dots, X_p)^T \in \mathbb{R}^p$

  (X: random vector)

- $E(X) = (E(X_1), E(X_2), \dots, E(X_p))^T$

  ↑ vector of expectations (population means)

- Covariance between two random variables X_1, X_2

  $Cov(X_1, X_2) = E(X_1 X_2) - E(X_1)E(X_2)$

- Variance-Covariance matrix (symmetric)

  $Cov(X) = \underbrace{E(X X^T)}_{p\times p} - \underbrace{E(X)}_{p\times 1}\cdot\underbrace{E(X^T)}_{1\times p}$

  → Diagonal contains the variances of X_i's

  $Var(X_i) = E(X_i^2) - E(X_i)^2$

  → off-diagonal entries (i,j) are the pairwise covariances

  $Cov(X_i, X_j)$

Sample

- Data X, matrix of size $n \times p$ (typically $n > p$)

$$\begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & & X_{2p} \\ \vdots & \vdots & & \vdots \\ X_{n1} & X_{n2} & & X_{np} \end{bmatrix}$$

$$\left(\frac{1}{n}\sum_{i=1}^n X_{i1}, \frac{1}{n}\sum_{i=1}^n X_{i2}, \dots, \frac{1}{n}\sum_{i=1}^n X_{ip}\right)$$

$\bar{X}$ ← vector of means.

$$\bar{X} = (\bar{X_1}, \bar{X_2}, \dots, \bar{X_p})$$

$$= \frac{1}{n}[1\,1 \dots 1]X$$

$$= \frac{1}{n}(\underbrace{\mathbf{1}^T}_{1\times n} \underbrace{X}_{n\times p})$$

- Sample Variance-Covariance matrix

$$\Sigma = \frac{1}{n-1}\left(X^T X - n\,\bar{X}^T\bar{X}\right)$$

$X^T X$: $p\times n$ · $n\times p$ → $p\times p$; $\bar{X}^T\bar{X}$: $p\times 1$ · $1\times p$ → $p\times p$

→ Diagonal contains the sample variances of each column of X

$$\frac{1}{n-1}\sum_{i=1}^n X_{ij}^2 - \frac{n}{n-1}\bar{X_j}^2$$

$$= \frac{1}{n-1}\sum_{i=1}^n (X_{ij} - \bar{X_j})^2$$

The standard deviation is the square root of the sample variance

↳ off-diagonal entries (i,j) are the covariances between i-th, j-th columns of X

$$\frac{1}{n-1}\sum_{k=1}^{n} X_{ki} X_{kj} - \frac{n}{n-1}\bar{X}_i \bar{X}_j$$

$$= \frac{1}{n-1}\sum_{k=1}^{n} (X_{ki} - \bar{X}_i)(X_{kj} - \bar{X}_j)$$

If C is a matrix such that CX is well-defined

$$E[CX] = C\,E[X]$$

$$Cov(CX) = C\,Cov(X)\,C^T$$

Without loss of generality in our sample data analysis we will assume that every column of X is centered around its mean

This is done by replacing

$$X_{ij} \rightsquigarrow X_{ij} - \bar{X}_j$$

In that case

$$\bar{X} = (0, 0, \ldots, 0)$$

$$\Sigma = \frac{1}{n-1} X^T X$$

this is now the "new X"

Note: This "centering" operation does not change the variance-covariance matrix Σ

(prove it!)

Example $n = 3, p = 2$

"raw" data (before centering)

$$\begin{bmatrix} 1 & 2 \\ 2 & 2 \\ 3 & 8 \end{bmatrix} \qquad \text{means} \quad 2 \quad 4$$

centering $\sim$

$$X = \begin{bmatrix} -1 & -2 \\ 0 & -2 \\ 1 & 4 \end{bmatrix} \qquad \text{mean} \quad 0 \quad 0$$

$$\Sigma = \frac{1}{n-1} X^T X = \frac{1}{3-1} \begin{bmatrix} -1 & 0 & 1 \\ -2 & -2 & 4 \end{bmatrix} \begin{bmatrix} -1 & -2 \\ 0 & -2 \\ 1 & 4 \end{bmatrix}$$

$$= \frac{1}{2} \begin{bmatrix} 2 & 6 \\ 6 & 24 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 3 & 12 \end{bmatrix}$$

Exercise Compute Σ for the "raw" data using

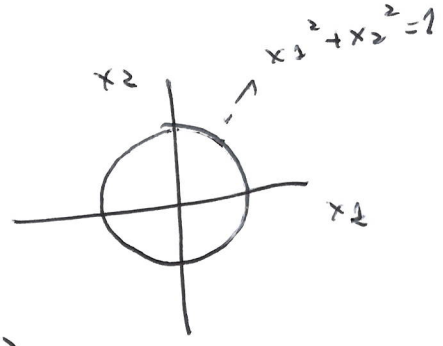
$$\Sigma = \frac{1}{n-1} X^T X - \frac{n}{n-1} \overline{X}^T \overline{X}$$

and check that it coincides with $\uparrow$

Previous exercise

$x = (x_1, x_2)$

$$\begin{cases} \max\limits_{x} & x_1 + x_2 \\ \text{s.t.} & x_1^2 + x_2^2 = 1 \end{cases}$$



$$L(x_1, x_2, \lambda) = x_1 + x_2 - \lambda(x_1^2 + x_2^2 - 1)$$

$$\frac{\partial L}{\partial x_1} = 1 - 2\lambda x_1 \qquad \frac{\partial L}{\partial x_1} = 0 \Rightarrow 1 = 2\lambda x_1 \rightsquigarrow 1 = 4\lambda^2 x_1^2 \quad (1)$$

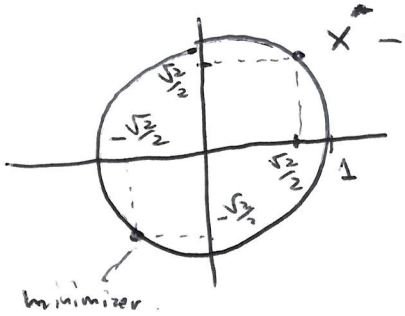
$$\frac{\partial L}{\partial x_2} = 1 - 2\lambda x_2 \qquad \frac{\partial L}{\partial x_2} = 0 \Rightarrow 1 = 2\lambda x_2 \rightsquigarrow 1 = 4\lambda^2 x_2^2 \quad (2)$$

$$\frac{\partial L}{\partial \lambda} = -(x_1^2 + x_2^2 - 1) \qquad \frac{\partial L}{\partial \lambda} = 0 \Rightarrow x_1^2 + x_2^2 = 1 \quad (3)$$

$$(1) + (2) \Rightarrow 2 = 4\lambda^2 \underbrace{(x_1^2 + x_2^2)}_{=1} \Rightarrow \lambda = \pm \frac{\sqrt{2}}{2}$$

if $\lambda = \frac{\sqrt{2}}{2} \Rightarrow x_1 = x_2 = \frac{\sqrt{2}}{2} \rightsquigarrow x_1 + x_2 = \sqrt{2}$ ✓

if $\lambda = -\frac{\sqrt{2}}{2} \Rightarrow x_1 = x_2 = -\frac{\sqrt{2}}{2} \rightsquigarrow x_1 + x_2 = -\sqrt{2}$



Principal Component Analysis (PCA)

Objective: Study projections of the data X that best reproduce its variability

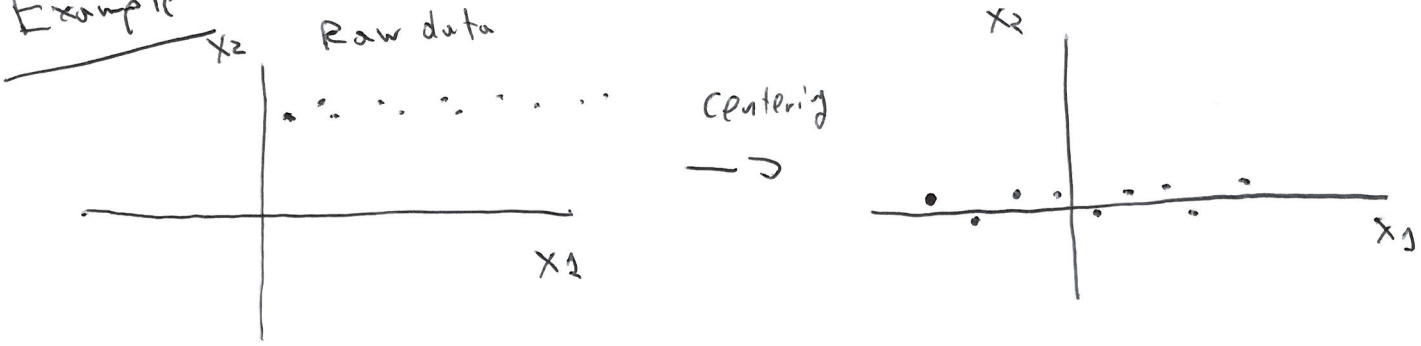
$\Rightarrow$ dimensionality ↓, variability remain as much as original data.

$\Rightarrow$ Useful for: analysis, visualisation, storage

Data X, matrix $n \times p$, and we assume that every column of X has been centered around its mean such that $\bar{X} = (\bar{X_1}, \bar{X_2}, \dots \bar{X_p}) = (0, 0 \dots 0)$

Note: This centering of X does not alter the variance-covariance matrix.

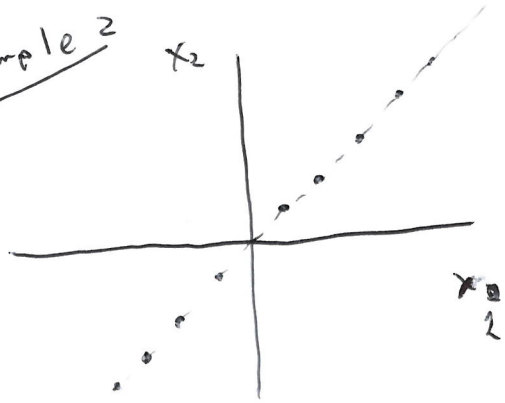
Example 1



$$X: \quad \begin{array}{c|c} X_1 & X_2 \\ \hline 11 & -0.02 \\ 12 & 0.01 \\ 23 & 0.002 \\ -31 & -0.03 \\ \vdots & \vdots \\ 25 & 0.01 \\ -50 & -0.005 \end{array}_{n \times 2} \qquad \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}_{2 \times 1} \qquad \begin{matrix} a_1 = 1 \\ \rightsquigarrow \\ a_2 = 0 \end{matrix} \qquad \begin{bmatrix} 11 \\ 12 \\ 23 \\ -31 \\ \vdots \\ 25 \\ -50 \end{bmatrix}_{n \times 1}$$

Note: $a_1^2 + a_2^2 = 1$

Example 2



$$X: \quad \begin{array}{c|c} x_1 & x_2 \\ \hline 11 & 11 \\ 12 & 12 \\ 23 & 23 \\ -31 & -31 \\ \vdots & \\ 25 & 25 \\ -50 & -50 \end{array} \qquad \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

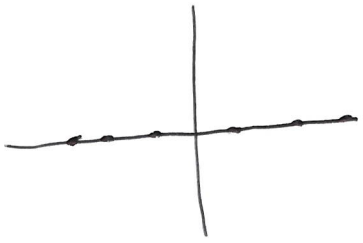
It will turn out that

$\alpha_1 = \frac{\sqrt{2}}{2}$

$\alpha_2 = \frac{\sqrt{2}}{2}$

(we will come back to this)

↓ rotate the data



Aim: Create a linear projection of the data X that has maximal variance

$$z_1 = X \alpha_1$$

z_1: vector of projections, $n \times 1$

X: data matrix, $n \times p$

α_1: vector of constants, size $p \times 1$ (unknown at this stage)

$\|\alpha_1\| = 1$

Check: z_1 has zero mean

We want to find α_1 such that the variance of z_1 is maximised. The sample variance of z_1 is:

$$\frac{1}{n-1} \sum_{i=1}^{n} z_{1i}^2 = \frac{1}{n-1} z_1^T z_1 = \frac{1}{n-1} (X\alpha_1)^T X\alpha_1$$

$$= \frac{1}{n-1} \alpha_1^T X^T X \alpha_1 = \alpha_1^T \left(\frac{1}{n-1} X^T X \right) \alpha_1$$

Σ: variance-covariance matrix

$$= \boxed{\alpha_1^T \Sigma \alpha_1} \leftarrow \text{Quadratic form}$$

$1 \times p$ $\quad p \times p$ $\quad p \times 1$

1×1

PCA

$$z_1 = \underset{n \times p}{X} \cdot \alpha_1 \qquad Var(z_1) = \alpha_1^T \Sigma \alpha_1$$

maximize $\alpha_1^T \Sigma \alpha_1$	subject to $\alpha_1^T \alpha_1 = 1$

$\alpha_1^T \alpha_1 = \|\alpha_1\|_2^2$

$$\sum_{i=1}^{p} (\alpha_{1,i})^2 = 1$$

"maximize the variance of the projection, using projection vector of unit norm"

We turn to the method of Lagrange multipliers in order to solve this problem.

$$L = \underbrace{\alpha_1^T \Sigma \alpha_1}_{\text{"target"}} - \lambda (\alpha_1^T \alpha_1 - 1)$$

- "objective function": sample variance of the projection
- λ: Lagrange multiplier
- "constraint": vector of coefficients has unit norm, $\alpha_1^T \alpha_1 = 1$ re-written as $\alpha_1^T \alpha_1 - 1 = 0$

Total unknowns: p elements for α_1
1 Lagrange multipliers

Taking derivatives of the Lagrangian

[$f(s) = s^T B s$, $\frac{\partial f}{\partial s} = 2Bs$]

$\alpha_1^T \alpha_1 = \alpha_1^T I \alpha_1$

(1) $$\frac{\partial L}{\partial \alpha_1} = 2\Sigma\alpha_1 - 2\lambda\alpha_1 = 2(\Sigma\alpha_1 - \lambda\alpha_1)$$

(2) $$\frac{\partial L}{\partial \lambda} = -(\alpha_1^T \alpha_1 - 1)$$

- Build the system of equations by setting these derivatives to zero

(1) $2(\Sigma \alpha_1 - \lambda \alpha_1) = 0 \Rightarrow \boxed{\Sigma \alpha_1 = \lambda \alpha_1}$

(2) $-(\alpha_1^T \alpha_1 - 1) = 0 \Rightarrow \boxed{\alpha_1^T \alpha_1 = 1}$

This is nothing else than an <u>eigenvalue</u> problem:

λ is an eigenvalue of Σ
α_1 is the corresponding eigenvector of unit norm

} candidates for solution of the maximization problem

Reinserting the above into

$$V(z_1) = \alpha_1^T \Sigma \alpha_1$$
$$= \alpha_1^T (\Sigma \alpha_1) = \alpha_1^T \lambda \alpha_1 = \lambda \underbrace{\alpha_1^T \alpha_1}_{1} = \lambda$$

$\Rightarrow$ In order to maximize the variance of z_1, we take λ to be the <u>largest</u> eigenvalue of Σ, with α_1 the corresponding eigenvector.

(what about $-\alpha_1$? this will also work, so some of the relations we will describe next are alway "up to change of sign")

<u>Summary</u>: The eigenvector α_1 of Σ gives the coefficients, to build the linear projection $X\alpha_1$ of the data, and the corresponding eigenvalue is the <u>sample variance</u> of the projection.

$z_1 \rightarrow$ First principal component

<u>Second and following components</u>

After we have picked z_1, now we define

$$z_2 = X \alpha_2$$

We require again: 1) $V(z_2)$ is maximized as a function of α_2

2) α_2 has unit norm, $\alpha_2^T \alpha_2 = 1$

3) α_2 has to be orthogonal to α_1

Ex: Show that the Lagrangian for the second principal component is

$$L = \alpha_2^T \Sigma \alpha_2 - \lambda_1 (\alpha_2^T \alpha_2 - 1) - \lambda_2 \cdot (\alpha_2^T \alpha_1 - 0)$$

We end up:

$$\begin{cases} \Sigma \alpha_2 = \lambda_2 \alpha_2 \\ \alpha_2^T \alpha_2 = 1 \\ \alpha_2^T \alpha_1 = 0 \end{cases}$$

In order to maximize $V(z_2) = \alpha_2^T \Sigma \alpha_2$, I take the eigenvector α_2 of unit norm, that corresponds to the second largest eigenvalue of Σ

[Recall that $\Sigma = \frac{1}{n-1} X^T X$ is symmetric and if $\lambda_1 > \lambda_2$ are eigenvalues of Σ, with α_1, α_2 are the corresponding eigenvectors $\Rightarrow \alpha_2^T \alpha_1 = 0$]

The pattern is clear:

For maximizing the variance of the projection in a sequential manner.

projection vectors α: eigenvectors of variance-covariance matrix Σ

Variance of the projection: $\alpha^T \Sigma \alpha$ — eigenvalues λ of Σ

$$z_i = X a_i \qquad i = 1, \dots, p$$

Σ is the sample variance-covariance matrix of data X (centered and possibly scaled) and let Λ and A be the matrices of the Spectral decomposition of Σ

(recall $\Sigma = \frac{1}{n-1} X^T X$, symmetric)

$$\Sigma = A \Lambda A^T$$

Here: $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$ with λ_i being the eigenvalues of Σ in decreasing order

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$$

$$A = \left[a_1 \mid a_2 \mid \dots \mid a_p \right]$$

is the matrix with the corresponding eigenvectors a_i as columns.

(In general (in practice), we deal with "well-behaved" matrices, i.e. all eigenvalues are different and non-zero)

PCA terminology

- The eigenvectors a_i are also known as "loadings" (PC loadings)
- The transformed variables $z_1 = X a_1$, $z_2 = X a_2 \dots$ are known "scores" (PC scores)

(5)

We can collect all the PC scores $(z_1, z_2, \dots, z_p)$ as columns in a matrix Z

$$Z = \left[\, z_1 \mid z_2 \mid \dots \mid z_p \,\right]_{n \times p} = \left[\, X_{d_1} \mid X_{d_2} \mid \dots \mid X_{d_2} \,\right]$$

$$= \underbrace{X}_{n \times p} \underbrace{A}_{p \times p}$$

$n \times p$

We can now compute the sample variance covariance matrix of Z

(recall that every column of Z has zero mean (why?))

$$\frac{1}{n-1} Z^T Z = \frac{1}{n-1} (XA)^T (XA)$$

$$= \frac{1}{n-1} A^T X^T X A$$

$$= A^T \left[\frac{1}{n-1} X^T X \right] A$$

$$= A^T \Sigma A \qquad (\Sigma = A \Lambda A^T)$$

$$= A^T A \Lambda A^T A$$

$$= \Lambda \qquad (\text{diagonal matrix } \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_p \end{bmatrix})$$

From the above we observe:

- The sample variance of z_i is the eigenvalue λ_i
- The z_i's (scores) are uncorrelated!
- The total variance of X equals to the total variance of Z

(6) Indeed for the third bullet point:

$$\sum_{i=1}^{p} \mathrm{Var}(X_i) = \mathrm{tr}(\Sigma) = \sum_{i=1}^{p} \lambda_i = \sum_{i=1}^{p} \mathrm{Var}(Z_i)$$

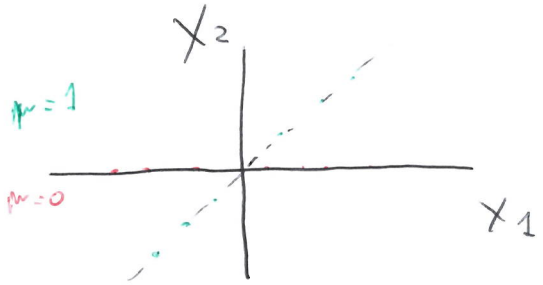
$\mathrm{tr}(\Sigma)$: sum of diagonal entries of Σ (trace of Σ)

General result:

> trace of a matrix = sum of its eigenvalues

Exercise

$$X = \left[X_1 \,\middle|\, \mu X_1 \right]$$



$\Rightarrow$ Do the PCA: find Z, A, Λ

$$A = \left[a_1 \,\middle|\, a_2 \right]$$

if $\mu = 0$ $\quad X = \begin{bmatrix} X_1 & 0 \\ & 0 \\ & \vdots \\ & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} X_1 \end{bmatrix}$

($\begin{bmatrix} 1 \\ 0 \end{bmatrix} = a_1$)

what happens: $\mu = 1$

$\mu = -1$?

PCA continued: Some comments on previous exercises:

Second Principal Component

$$\max \quad a_2^T \Sigma a_2$$
$$\text{s.t} \quad a_2^T a_2 = 1$$
$$a_1^T a_2 = 0$$

$$L = a_2^T \Sigma a_2 - \lambda (a_2^T a_2 - 1) - \mu a_1^T a_2$$

$$\frac{\partial L}{\partial a_2} = 2\Sigma a_2 - 2\lambda a_2 - \mu a_1 \overset{=0}{\Rightarrow} 2\Sigma a_2 - 2\lambda a_2 = \mu a_1$$

$$\frac{\partial L}{\partial \lambda} = -(a_2^T a_2 - 1) \overset{=0}{\Rightarrow} a_2^T a_2 = 1$$

$$\frac{\partial L}{\partial \mu} = -a_1^T a_2 \overset{=0}{\Rightarrow} a_1^T a_2 = 0$$

I claim $\mu = 0$

Multiply the first equation with a_1^T from the left:

$$2 \underbrace{a_1^T \Sigma a_2}_{=} - 2\lambda \underbrace{a_1^T a_2}_{=0} = \mu \underbrace{a_1^T a_1}_{1}$$

$$2 \underbrace{a_2^T \Sigma a_1}_{=} = 2\lambda_1 \underbrace{a_2^T a_1}_{=0}$$

$$\Rightarrow \boxed{\mu = 0}$$

In summary:

$$\boxed{\begin{aligned} \Sigma a_2 &= \lambda a_2 \\ a_2^T a_2 &= 1 \\ a_1^T a_2 &= 0 \end{aligned}}$$

Example

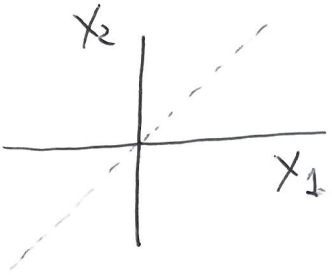
Suppose data matrix X of the following form:

$$X = \begin{bmatrix} 1 & \mu \\ -1 & -\mu \end{bmatrix}$$

$X_1 \quad X_2 = \mu X_1$

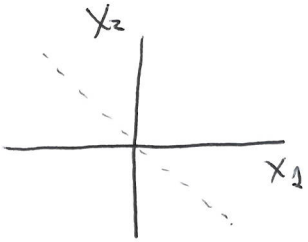
if $\mu = 1$ $\quad X = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$

X_1, X_2 pos. correlated



if $\mu = -1$ $\quad X = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$

X_1, X_2 neg. correlated



if $\mu = 0$ $\quad X = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}$

$$\Sigma = \frac{1}{n-1} X^T X = \begin{bmatrix} 1 & -1 \\ \mu & -\mu \end{bmatrix} \begin{bmatrix} 1 & \mu \\ -1 & -\mu \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 2\mu \\ 2\mu & 2\mu^2 \end{bmatrix} = 2 \begin{bmatrix} 1 & \mu \\ \mu & \mu^2 \end{bmatrix}$$

Total variance of X: $2(1+\mu^2)$

Look at eigenvalues of $\begin{bmatrix} 1 & \mu \\ \mu & \mu^2 \end{bmatrix}$

$$\begin{vmatrix} 1-\lambda & \mu \\ \mu & \mu^2-\lambda \end{vmatrix} = (1-\lambda)(\mu^2-\lambda) - \mu^2 = \mu^2 - \lambda - \lambda\mu^2 + \lambda^2 - \mu^2$$

$$= \lambda(\lambda - 1 - \mu^2)$$

The only positive eigenvalue of Σ is $2(1+\mu^2)$

Eigenvectors:

$$\begin{bmatrix} 2 - 2(1+\mu^2) & 2\mu \\ 2\mu & 2\mu^2 - 2(1+\mu^2) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} -2\mu^2 & 2\mu \\ 2\mu & -2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = 0$$

$$\boxed{a_2 = \mu a_1}$$

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = a_1 \begin{bmatrix} 1 \\ \mu \end{bmatrix}$$

(3) $\mu = 1$

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$$

$$\left(\frac{\sqrt{2}}{2}\right)^2 + \left(\frac{\sqrt{2}}{2}\right)^2 = 1$$

$\mu = -1$

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \end{bmatrix}$$

$\left(\begin{bmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$ work as well $\right)$

$\mu = 0$

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

PCA and SVD

So far we used the spectral decomposition of $\Sigma = A \Lambda A^T$. All the results can be recovered using the SVD of X:

$$\underset{p \times p}{\Sigma} = \underset{p \times p}{A} \underset{p \times p}{\Lambda} \underset{p \times p}{A^T} \qquad \Big| \qquad X = U D V^T$$

U: $n \times \min(n,p)$, D: $\min(n,p) \times \min(n,p)$, V^T: $\min(n,p) \times p$

Let us build Σ using the SVD of X

$$\Sigma = \frac{1}{n-1} X^T X = \frac{1}{n-1} V D \underbrace{U^T U}_{I} D V^T$$

$$= \frac{1}{n-1} V D^2 V^T$$

$$= V \left(\frac{1}{n-1} D^2 \right) V^T$$

We conclude:

a) Eigenvectors of Σ (PC loadings) are the columns of V in SVD of X

$$\boxed{A = V} \qquad [\text{up to a sign}]$$

b) The eigenvalues of Σ are related to D (singular values)

$$\lambda_i = \frac{1}{n-1} d_i^2 \qquad D = \begin{bmatrix} d_1 & & 0 \\ & d_2 & \\ 0 & & \ddots \; d_p \end{bmatrix}$$

Let us finish by computing also the PC scores z_i

$$Z = XA \qquad \text{and now using the SVD}$$

$$Z = XA = UDV^T A \underset{A=V}{=} UD \underbrace{A^T A}_{I} = UD$$

$$\boxed{Z = XA = UD} \qquad (\text{up to a sign})$$

(2)

Interpreting PCA output

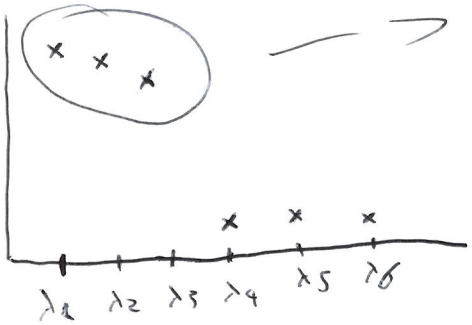
The main objective of PCA is to reduce dimensionality of the data and select only a few principal components, that best represent the variability of the data.

How many components to choose?

Strategies to select the number of components:

Scree plot

Plot eigenvalues (sample variances of PC scores) in a decreasing order of magnitude, look where the plot has a sharp decrease/ becomes flat.
=> Select number of components according to this.



Select 3 components

(easy case)

(hard case)

(3) Tabulate eigenvalues and cumulative sum → scale it to be a percentage.

	PC1	PC2	PC3
Variance	52.2	15.11	1.98
Proportion of variance	0.75	0.21	0.02
Cumulative proportion	0.75	0.96	1.00

→ Set a threshold: of how much variability you want to explain and then take the minimum number of components that explain this variability

Example: if threshold 80% = 0.8

=> take PC1, PC2

if threshold 99% = 0.99

=> take PC1, PC2, PC3

Summarize

The proportion of variance explained by the first k- components

$$\frac{\lambda_1 + \lambda_2 + \dots + \lambda_k}{\lambda_1 + \lambda_2 + \dots + \lambda_p}$$

→ variance of first k- components

→ total variance of all components.

For a fixed threshold pick the smallest value of k such that the above proportion exceeds this threshold.

(4)

Interpreting PCA as weighted averages and contrasts

Look at PC loadings (eigenvectors) and we inspect their signs and magnitudes

These are used to interpret PCA as weighted averages and/or as contrasts

	PC1	PC2
Hill	0.64	−0.17
Flat	−0.09	−0.98
Tarmac	0.75	0.02 ~> too small

PC2 is essentially a weighted average of "Hill" "~~Tarmac~~" flat

↓ weighted average of "Hill" and "Tarmac" (same signs) compared against "Flat" (different signs)

Remember: These interpretations should be the same when all signs are reversed

$$\left[\max_{\|a_1\|_2 = 1} a_1^T \Sigma a_1 \qquad a_1 \to -a_1\right]$$

~~Stat~~ **should we scale the data or not?**

A crucial feature of PCA is that analyses and all conclusions differ between data that have been scaled and data have not been scaled.

(3)

Common practice is to analyse scaled variables (centered and scaled) only if the units of measurements are the same for all the variables.

Note in that case, the proportion of variability explained by the first k- components is

$$\frac{\sum_{i=1}^{k} \lambda_i}{p}$$

- Sample variance - covariance matrix for the scaled data

$$R = S^{-1/2} \Sigma S^{-1/2}$$ → S: diagonal matrix whose diagonal is equal to Σ

$$R = S^{-1/2} \Sigma S^{-1/2}$$

$$= S^{-1/2} A \Lambda A^T S^{-1/2}$$ multiply $S^{1/2} A$ ⟹

$$R S^{1/2} A = S^{-1/2} A \Lambda$$

Compare $\Sigma A = A \Lambda$

↝ R & Σ have different eigenvalues / eigenvectors.

Biplots

A biplot is a plot that simultaneously shows the PC scores and the PC loadings.

PC scores: we usually plot the first two in a scatterplot.

PC loadings: arrows, typically as many as variables.

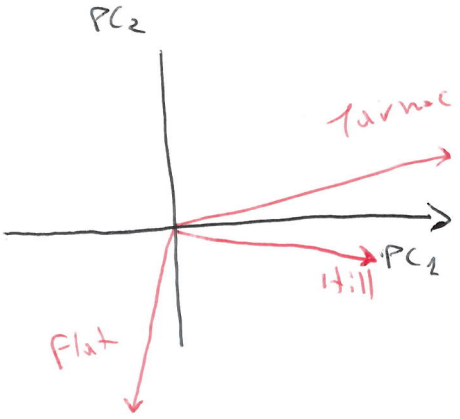
(6) It is used to associate observations with variables in the first two principal components

We also interpret the loadings in a graphical way.

Elements of a biplot

Scatterplot of PC scores

Scatterplot of PC loadings



	PC1	PC2
Hill	0.04	-0.17
Flat	-0.09	-0.98
Farmc	0.75	-0.02

Typically in scaled data, there is no clear dominance of any variable ~> arrows have approximately the same length.

~ ~

PCA is often the first step in reducing complexity of a dataset, by keeping linear combinations of data that maximize the variance.

This method is typically used in conjuction with other methods of machine learning:

Example: - do PCA ~> clustering

\ do PCA ~> regression

150224
(1400)

3 - Clustering

The aim of clustering is to detect groups "clusters" in the individuals that compose the data.

Data matrix $\underline{X} = \begin{pmatrix} & \\ & \\ & \end{pmatrix}$ ← rows are individuals, ↑ columns are variables, entries are numbers, $n \times p$

"a vector of p numbers"

Cluster - group of individuals

We will look at < agglomerative clustering
K-means (medoids) method

3.1 Distances and distance matrix

Assume that we have data on $n=5$ individuals and $p=2$ variables

individual labels (index)

$$\underline{X} = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 4 \\ 3 & 6 \\ 6 & 2 \\ 0 & 5 \\ 1 & 1 \end{pmatrix}$$

We will consider distances between points/individuals that are a metric.

Metric: The distance d_{ij} between individuals i and j satisfies the following properties: non-negative, symmetric, satisfies the triangle inequality and takes value zero between an individual and itself.

- Non-negative $d_{ij} \geq 0$ For all i,j
- Symmetric $d_{ij} = d_{ji}$ for all i,j
- Triangle inequality $d_{ij} + d_{jk} \geq d_{ik}$ for all i,j,k
- Zero distance with itself $d_{ii} = 0$ For all i

Let us consider the 'Manhattan' distance between points in the example

$$d_{12} = |0-3| + |4-6| = 5$$
$$d_{13} = |0-6| + |4-2| = 8$$
$$d_{14} = |0-0| + |4-5| = 1$$
$$d_{15} = |0-1| + |4-1| = 4$$

Exercise. Check/compute $d_{23}, d_{24}, d_{25}, d_{34}, d_{35}, d_{45}$ as well as $d_{11}, \ldots$ $d_{21}, d_{31}, \ldots$

Distance matrix: collection of distances d_{ij} in a $n \times n$ matrix

Notation $\underline{D} = (d_{ij})$

$$\underline{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 5 & 8 & 1 & 4 \\ 5 & 0 & 7 & 4 & 7 \\ 8 & 7 & 0 & 9 & 6 \\ 1 & 4 & 9 & 0 & 5 \\ 4 & 7 & 6 & 5 & 0 \end{pmatrix} \end{array}$$

can also write

$$\underline{D} = \begin{pmatrix} - & 5 & 8 & 1 & 4 \\ 5 & - & 7 & 4 & 7 \\ 8 & 7 & - & 9 & 6 \\ 1 & 4 & 9 & - & 5 \\ 4 & 7 & 6 & 5 & - \end{pmatrix}$$

We consider the following distances between individuals/points $\underline{x}_i = (x_{i1}\ x_{i2}\ \cdots\ x_{ip})$ and $\underline{x}_j = (x_{j1}\ x_{j2}\ \cdots\ x_{jp})$

→ Manhattan $d_{ij} = \sum_{l=1}^{p} |x_{il} - x_{jl}| = \|\underline{x}_i - \underline{x}_j\|_1$ L_1 norm

→ Euclidean $d_{ij} = \sqrt{\sum_{l=1}^{p} |x_{il} - x_{jl}|^2} = \|\underline{x}_i - \underline{x}_j\|_2$ L_2 norm

→ Minkowski $d_{ij} = \left(\sum_{l=1}^{p} |x_{il} - x_{jl}|^m \right)^{1/m} = \|\underline{x}_i - \underline{x}_j\|_m$ L_m norm (Lp norm in books)

($m=1$ gives Manhattan, $m=2$ gives Euclidean)

→ Mahalanobis $d_{ij} = \sqrt{(\underline{x}_i - \underline{x}_j)\, \Sigma^{-1} (\underline{x}_i - \underline{x}_j)^T}$ ($1 \times p$, $p \times p$, $p \times 1$) — $p \times p$ covariance matrix between columns

→ Absolute correlation $d_{ij} = \sqrt{1 - |\rho_{ij}|}$ ← correlation between points

3.2 Agglomerative clustering

This is a sequential procedure to cluster 'n' individuals.

a) Initially, each individual is its own cluster

b) At every step, an updated table of dissimilarities between clusters is considered and minimized to merge an individual to a cluster.

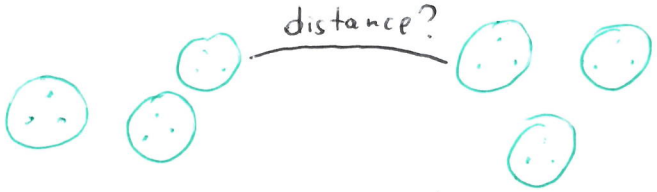
c) At the end of the procedure, a single cluster is achieved.

Agglomerative clustering

Two ingredients to be specified: **distance** & **linkage**

- distance → dissimilarity between points
- linkage → dissimilarity between sets of points (clusters)

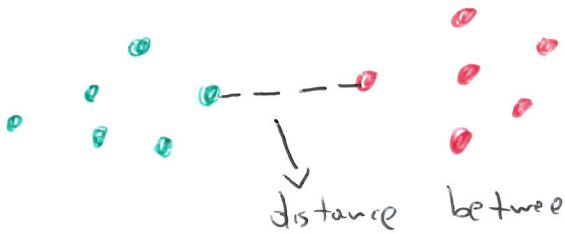
Dissimilarities: "distance" between clusters that contain more than one point



The way these dissimilarities are computed is dictated by the linkage

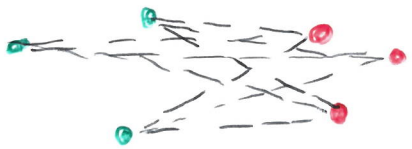
Single linkage

The dissimilarity between clusters is the distance between the closest 2 points in each cluster ("nearest neighbour")
(this is also know as the



Average linkage

The dissimilarity between clusters is the average of distances between points in each cluster ("average neighbour")



(2) Complete linkage

The dissimilarity between clusters is the distance between the two points in each cluster that are farthest apart ("farthest neighbour")

To perform agglomerative clustering we select a
- distance (e.g Euclidean, Manhattan, ...)
- linkage (single, average, complete)

Example Using Manhattan
single linkage

$$X = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 4 \\ 3 & 6 \\ 6 & 2 \\ 0 & 5 \\ 1 & 1 \end{pmatrix}$$

$$D = \begin{matrix} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{pmatrix} - & 5 & 8 & 1 & 4 \\ 5 & - & 7 & 4 & 7 \\ 8 & 7 & - & 9 & 6 \\ 1 & 4 & 9 & - & 5 \\ 4 & 7 & 6 & 5 & - \end{pmatrix} \end{matrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix}$$

a) Initially, every individual is its own cluster. At this stage the distance matrix is the dissimilarity matrix

$$\{1, 2, 3, 4, 5\}$$

b) Merge clusters 1 and 4, at distance 1

$$\Rightarrow \{14, 2, 3, 5\}$$

Update dissimilarity matrix

$$\begin{matrix} \begin{matrix} 14 & 2 & 3 & 5 \end{matrix} \\ \begin{pmatrix} - & 4 & 8 & 4 \\ & - & 7 & 7 \\ & & - & 6 \\ & & & - \end{pmatrix} \end{matrix} \begin{matrix} 14 \\ 2 \\ 3 \\ 5 \end{matrix}$$

(3) Compute dissimilarity between 14 & 2, $d_{14,2}$

We have $d_{1,2} = 5$, $d_{4,2} = 4$

"Single" $d_{14,2} = \min(d_{1,2}, d_{4,2}) = \min(5,4) = 4$

"Average" $d_{14,2} = \frac{d_{1,2} + d_{4,2}}{2} = 4.5$

"complete" $d_{14,2} = \max(d_{1,2}, d_{4,2}) = \max(5,4) = 5$

~> Clusters 14 & 2 are merged at distance 4

=> { 124, 3, 5 }

Update dissimilarity matrix:

	124	3	5
124	-	7	4
3		-	6
5			-

$d_{124,3} = \min(8,7,9) = 7$

$d_{124,5} = 4$

=> Clusters 124 & 5 are merged at distance 4

=> { 1245, 3 }

Update dissimilarity matrix

	1245	3
1245	-	6
3		-

$d_{1245,3} = \min(8,7,9,6) = 6$

c) All clusters are merged at distance 6

{ 1 2 3 4 5 }

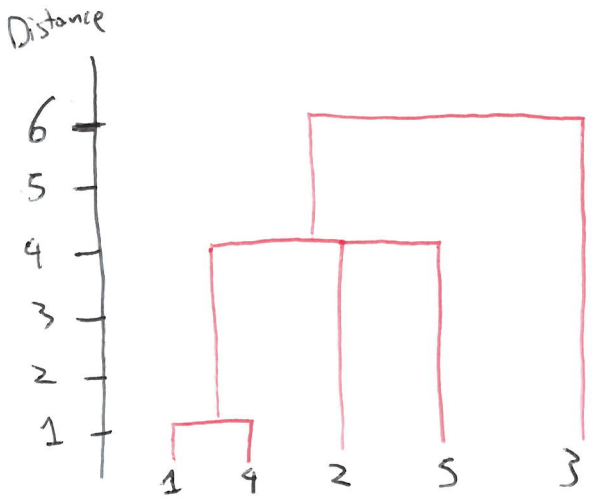
Dendrograms

"Dendro" ~> "tree" in Greek

Example

(single, Manhattan)

	Clusters formed
1	{14, 2, 3, 5}
4	{124, 3, 5}
4	{1245, 3}
6	{12345}



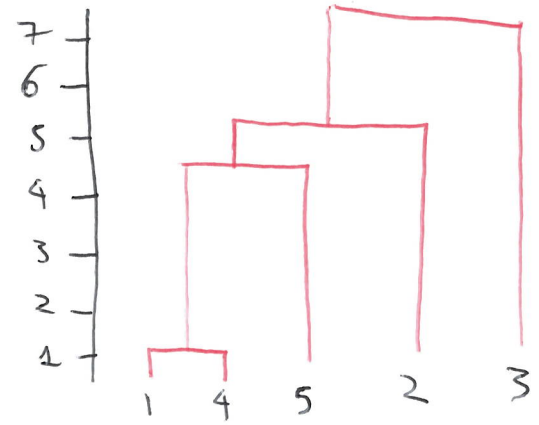
The clustering process is summarized in a diagram called dendrogram. In this diagram, individual points are represented as "leaves". As we move upwards the diagram, clusters are formed and merged in a gradual manner joining the main part ("trunk") of the tree. This joining happens at heights determined by the dissimilarity at every step.

Ordering of leaves in the dendrogram may not correspond to data order.

Another example (try to replicate!)

(average, Manhattan)

Distance	Clusters formed
1	{14, 2, 3, 5}
4.5	{145, 2, 3}
5.3	{1245, 3}
7.5	{12345}



* $d_{14,5} = \frac{d_{1,5} + d_{4,5}}{2} = 4.5$

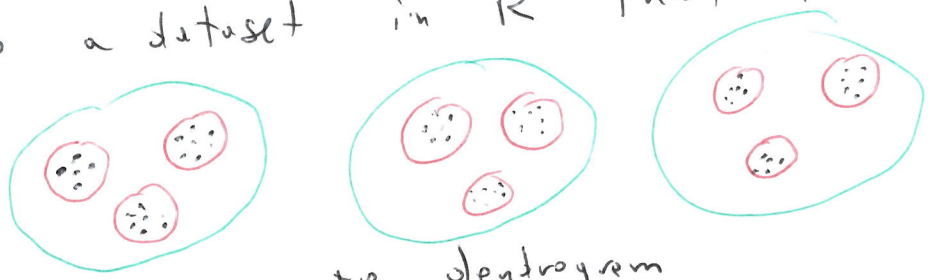
(5) Height of Jumps

The height of jumps in the dedrogram suggests where to "cut" to obtain "meaningfull" groups of individuals.

Big jumps in the dendrogram indicates clusters that are separate (far away from each other) and should not be merged

Do the following for practice

Create a dataset in R that looks like:



=> Examine the dendrogram

- Results depend on the choice of metric and linkage. Different choices should be explored, to examine the clustering from different angles
- Recall that clustering is a problem of unsupervised learning and thus the true number of clusters is not available.

R-demo: $ac: agglomerative coefficient

Measure the dissimilarity of a point to the first cluster it joins, divided by the dissimilarity of the final merging -> average accross points

In R: 1 - ^

-> High values (close to 1) indicates well-formed clusters

-> Low values (close to 0) indicates less-well-formed clusters.

Method of k-means

This clustering methodology works on a different premise than agglomerative clustering. In a sense it is more automatic as the user only has to specify the number of clusters k.

An algorithm seeks to minimize a measure of closedness to clusters, by changing allocation of individuals to clusters.

Example. $k=2$

We would like to define a map

$$\{\text{all possible } k\text{-cluster configurations}\} \longrightarrow \mathbb{R}^+$$

such that

→ this type of cluster has small value

→ this type of cluster has high value.

Once we have defined such a map, then we choose the clustering configuration that minimises this map

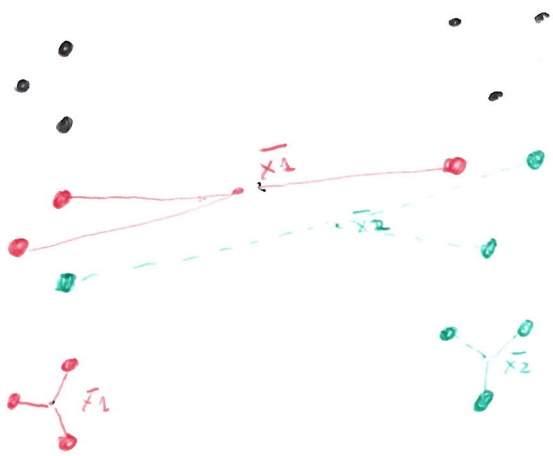
(2) The function to minimise is the following:

$$ESS = \sum_{k=1}^{K} \sum_{i \text{ s.t. } c(i)=k} \| x_i - \bar{x}_k \|_2^2$$

- ESS → error sum of squares
- $\sum_{k=1}^{K}$ → sum over all clusters
- $\sum_{i \text{ s.t. } c(i)=k}$ → sum over all points in cluster k
- x_i → i-th row of matrix X (belongs to the cluster k because of $c(i)=k$)

$\bar{x}_k$ is a centroid for the cluster k. For instance, it can be the mean (vector of means per column/variable)

Example



→ these are my data X

→ ESS is large! obviously this not a "good" clustering

→ ESS is small! obviously this a "good" clustering

ESS essentially sums up all the distances between all the points and the centres (means) of the clusters they belong to.

Apart from the mean, it could be the case that $\bar{x}_k$ is one of the points in cluster k such that the inner sum in ESS is minimised. In that case $\bar{x}_k$ is called the medoid and the method is called k-medoids.

k-means ~> R kmeans

k-medoids ~> R pam

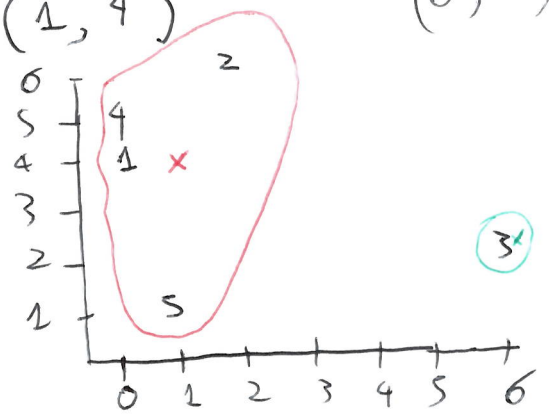
(3)

Example

Recall synthetic data: $X = \begin{pmatrix} 0 & 4 \\ 3 & 6 \\ 6 & 2 \\ 0 & 5 \\ 1 & 1 \end{pmatrix}$ $n = 5$ individuals.

Output k-means, $k = 2$

	Cluster 1	Cluster 2
Members	$\{1, 2, 4, 5\}$	$\{3\}$
center	$(1, 4)$	$(6, 2)$



(1)

k-means

In summary, k-means (k-medoids) is a procedure that splits the dataset into k clusters. This is done by minimising ESS and if we were to do this "by hand" we would have to use an exhaustive approach

Example

$$X = \begin{pmatrix} 0 & 4 \\ 3 & 6 \\ 6 & 2 \\ 0 & 5 \\ 1 & 1 \end{pmatrix}$$

$n = 5$ observations, $k = 2$ clusters.

- Number of ways of splitting these 5 observations into 2 clusters of 3 and 2 individuals

$$\binom{5}{2} = \binom{5}{3} = 10 \quad \text{different ways}$$

- Number of ways of doing this splitting into 2 clusters of 4 and 1 individuals.

$$\binom{5}{1} = \binom{5}{4} = 5 \quad \text{different ways}$$

Clusters	ESS	
	k-medoid	k-means
123, 45	55	39.5
124, 35	37	21
⋮	⋮	⋮
12, 345	56	38
1, 2345	69	34.75
1245, 3	24	20
1235, 4	48	35.75

② Example · ESS value for cluster $\{1,2,4,5\}, \{3\}$ k-means, $k=2$

x_1 (0, 4)
x_2 (3, 6)
x_3 (6, 2) → cluster 2 with center $\bar{x}_2 = (6, 2)$
x_4 (0, 5)
x_5 (1, 1)

→ cluster 1 with center $\bar{x}_1 = \left(\frac{0+3+0+1}{4}, \frac{4+6+5+1}{4}\right) = (1, 4)$

$$ESS = \underbrace{\sum_{i=1,2,4,5} \|x_i - \bar{x}_1\|_2^2}_{\text{sum over elements in 1st cluster}} + \underbrace{\sum_{i=3} \|x_i - \bar{x}_2\|_2^2}_{\text{sum over elements in 2nd cluster}}$$

$= 0$ since $\|x_3 - \bar{x}_2\|_2^2 = 0$

$\|x_1 - \bar{x}_1\|_2^2 = (0-1)^2 + (4-4)^2 = 1$

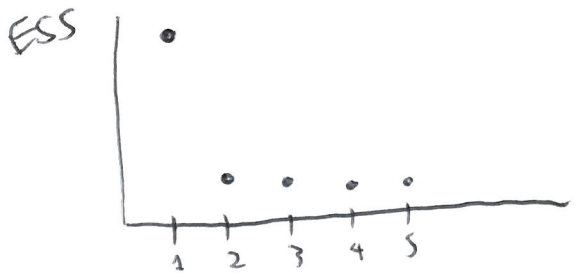
$\|x_2 - \bar{x}_1\|_2^2 = 8$, $\|x_4 - \bar{x}_1\|_2^2 = 2$ $\|x_5 - \bar{x}_1\|_2^2 = 9$ sum $\approx$ 20

This exhaustive approach is too expensive and the number of possible clusters grows exponentially with n. In practice ESS is minimised numerically using

kmeans, pam

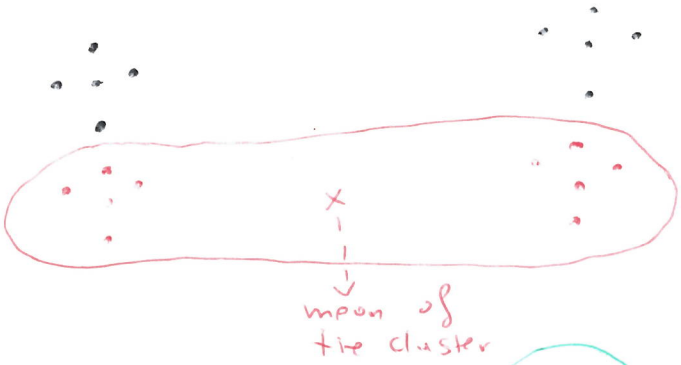
How many k ?

The quantity ESS is used as a tool to determine a "meaningful" number of clusters.
This is a subjective tool and there is no 100% guarantee that the numbers of clusters that we get from the following procedure is the "correct"



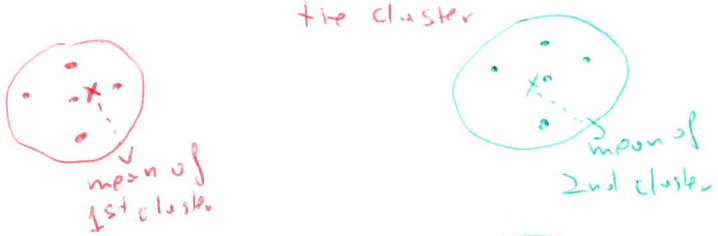
In this easy scenario, where we have a big jump from $k=1$ to $k=2$, we choose $k=2$ as the best option

(3) Example



$k = 1$

- ESS value will be large



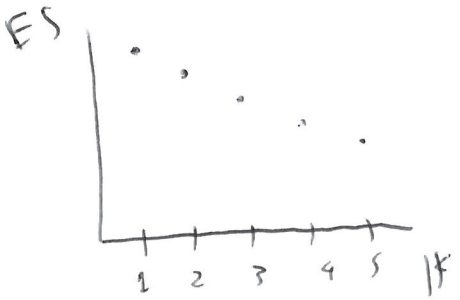
$k = 2$

- ESS value will be much smaller! It will make a big jump

$k = 3$

- ESS will be maybe a little bit smaller but it will not make a big jump

More difficult scenario



Difficult to determine a good value for k

That could happen when we have a dataset which is not so well clustered

Summary:

This kind of plot can be used to select potential number of clusters by looking at the decrease of ESS, while keeping k as small as possible

[Recall: $k = n \Rightarrow ESS = 0$]

④

Silhouette plot

This is a tool that calculates and shows the **silhouette value** $s(i)$ for every individual i in the data.

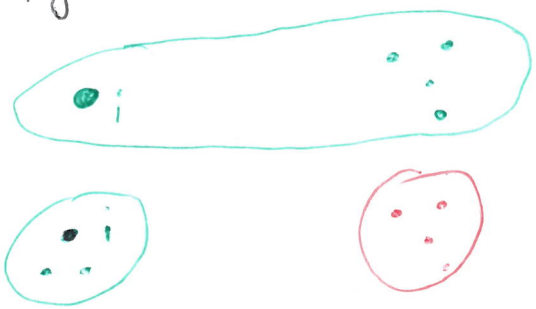
This quantity is usually between 0 and 1 but sometimes it can also take negative values up to -1

Values of $s(i)$	Interpretation
close to 1	→ individual i is well-clustered
close to 0	→ individual i is probably between 2 or more clusters
close to -1 / negative	→ individual i is most likely in the wrong cluster.

This $s(i)$ is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))}$$

$a(i)$: average dissimilarity (distance) between i and all other points of the cluster to which i belongs
(if i is alone, in its own cluster: $a(i) = 0$)



$a(i)$ is large

$a(i)$ is small

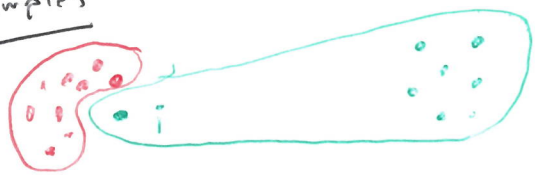
(5)

$d(i, C)$ = average dissimilarity between i and all observations in cluster C

(C: cluster; ~~its nearest~~)

$b(i) = \min_{C} d(i, C)$ → dissimilarity between i and its nearest cluster

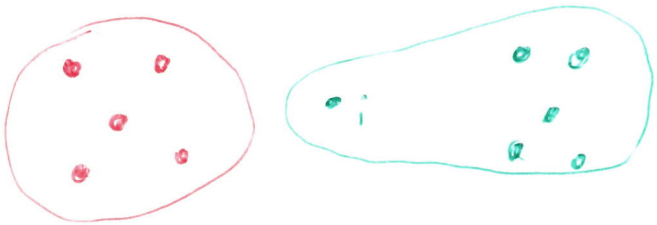
Examples



$a(i)$ is large

$b(i)$ is small (close to zero)

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \simeq \frac{-a(i)}{a(i)} = -1$$



Here $a(i) \simeq b(i)$

$s(i) \simeq 0$

$a(i)$ is small

$b(i)$ is large

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \simeq \frac{b(i)}{b(i)} = 1$$

Clustering combined with PCA

We can use PCA as a pre-processing step to reduce dimensionality of the data and then perform clustering to the PC scores.

That is instead of using the data matrix X we use the first columns of $Z = XA$ of PCA

Advantages: reduce dimensionality and thus computational time before clustering. Better visualisation.

Disadvantages: interpretation of PC scores may not be the same as the one for the raw data. Moreover, there is not always guarantee that we will obtain a better clustering like that.

Supervised Learning

classification regression

In supervised learning we have the variable X (predictors features) but also a response variable Y

target / output

Essentially for every $x=(x_1, \dots, x_p)$ there is a value y associated to it

$$x \longmapsto y$$

The target is to "learn" this map given some data

Data looks like this:

$$\left[\; X \;\middle|\; Y \;\right]$$

X: input variables, $n \times p$ matrix

Y: output variable, matrix of size $n \times 1$

$n \times (p+1)$

In classification, Y takes values from a discrete set (labels). For our purposes, we will consider binary classification: 0 or 1

In regression, Y takes values from a potential infinite set (real values)

On the other hand, the variables X typically are real numbers.

Aim of supervised learning

Give some data (set of input – output variables)

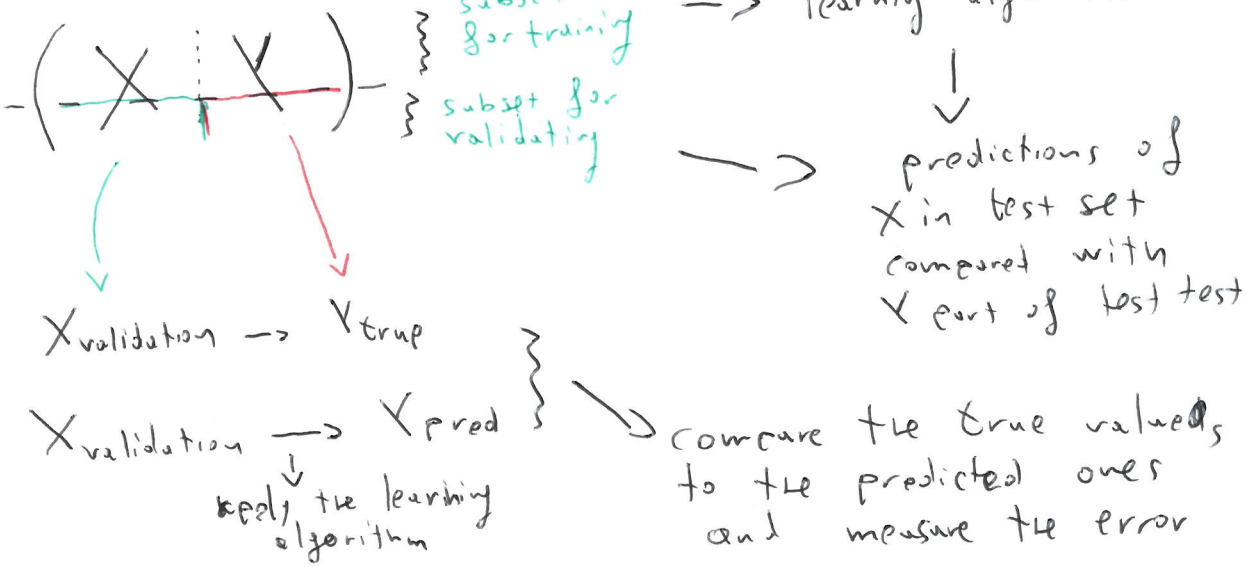
the aim is to create a model / algorithm such that it can predict correctly the output values for new values of X

Usually, we determine the predictive accuracy of an algorithm by measuring its accuracy for a data set that has not been used.

In practice:

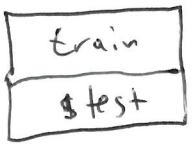
1) We train (build) the algorithm / predictor using some part of the data (training set)
2) We validate (evaluate) the algorithm on some different part of the data (validation / test set)

Data Partition



There are three main strategies for splitting the data

1) Divide the data into a training set and a test set

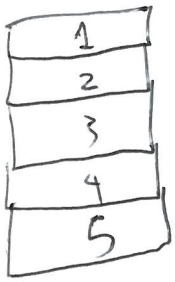


Partitions could be 80/20, 70/30, 50/50

train ↓ test

2) K-fold cross validation
Split the data into k parts (k folds), and use all folds but one to train and the remaining to test. Exhaust for all combinations.

Example 5-fold CV



Train	Test	Error
1,2,3,4	5	error1
1,2,3,5	4	error2
1,2,4,5	3	error3
1,3,4,5	2	error4
2,3,4,5	1	error5

↓

average ~~error~~ over all errors.

3) n-fold cross validation.
Also known as leave-one-out

In each of n instances, one observation is left out of the training set and it is used to evaluate the predicts every time.

The measure of accuracy is usually the average error over all instances.

9 The partition of data is usually done at random but to make results reproducible, we set a value for the random seed

R command set.seed( )

Machine learning vs Statistics

In machine learning, the focus is in comparing models with respect to their prediction capabilities.

This is contrast to statistics, where models are fitted to data and then significance tests for the terms of the models (predictive capability is not the main goal)

(1400)

4-Classification

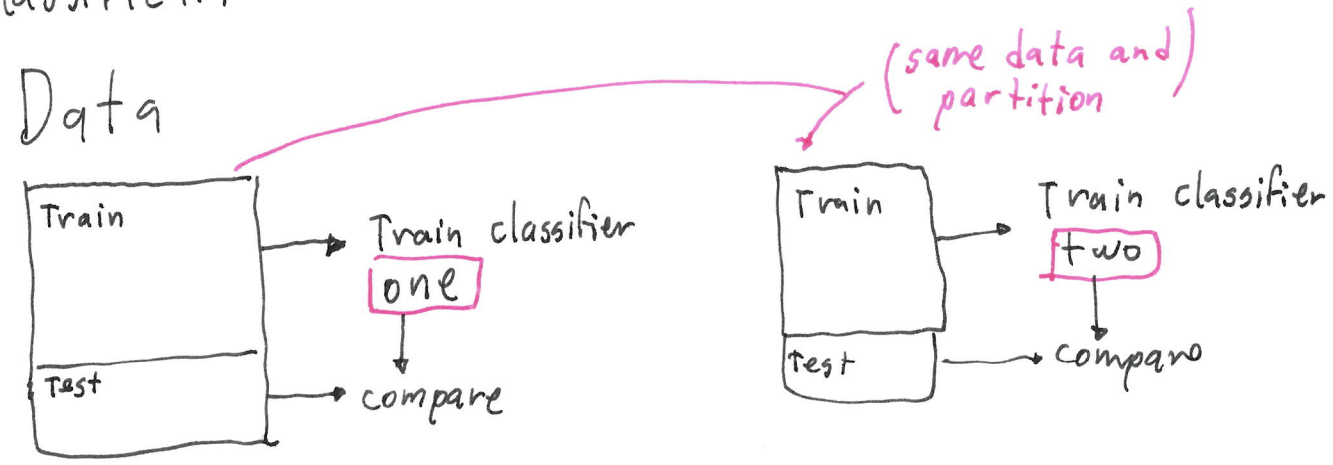
In classification, outputs take values from a discrete set. This is a set of labels and we are interested in predicting labels.
We will concentrate on the simplest case with only two labels.

Labels < 0 "-" healthy individual — normal transaction — person has not defaulted
1 "+" ill individual — Fraudulent transaction — person has defaulted

ill individual: detecting disease
Fraudulent transaction: detecting fraud in debit card payment
person has defaulted: assessing default on loans

4.1 Performance of methods

Using available data, the machine learning task of classification involves comparison of classifiers.



Example with two classifiers.

As an example of comparisons, consider the following data.

In automatic fraud detection

Y_{true}	$Y_{Model\ 1}$	
1	0	"Algorithm "Model 1" concluded no fraud yet it was." False negative
0	1	"Algorithm flagged a fraud yet there was none" False positive
0	1	
1	1	← True positive

Misclassification

Define the following concepts

True positive (TP) is a positive classified as a positive.

False positive (FP) is a negative classified as a positive.

False negative (FN) is a positive classified as a negative.

True negative (TN) is a negative classified as a negative.

We use the acronyms to denote counts of data

TP, FP, FN, TN.

$P = TP + FN$ ← number of positives

$N = TN + FP$ ← number of negatives

(1500)

Confusion matrix - a table that summarizes the different cases (correct classifications, misclassifications)

Preferred

True class	Predicted class 0	Predicted class 1	
0	TN	FP	N
1	FN	TP	P

A different way of showing

same table same information

True	Predicted 1	Predicted 0	
1	TP	FN	P
0	FP	TN	N

Example with $P = 3, N = 2$
($P + N = 5$ observations)

True	Pred. 0	Pred. 1	
0	0	2	2
1	1	2	3

$TP = 2$ ← correct
$FP = 2$ } Misclassified
$FN = 1$ }
$TN = 0$ ← correct

Pred

True	1	0	
1	2	1	3
0	2	0	2

$TPR = 2/3 = 0.66$

$FPR = 2/2 = 1$

Accuracy $= \frac{2}{5} = 0.4$

Error rate $= \frac{3}{5} = 0.6$

Some measures of classifiers' performance: * Important 3

* True positive rate (TPR) $TPR = TP/P$ (aka Hit rate, recall or sensitivity)
  proportion of positives correctly classified as positives.

* False positive rate (FPR) $FPR = FP/N$ (equal to 1 - specificity)
  proportion of negatives wrongly classified as positives

False negative rate (FNR) $FNR = \frac{FN}{P} = 1 - TPR$
proportion of positives wrongly classified as negatives.

True negative rate (TNR) $TNR = \frac{TN}{N} = 1 - FPR$ (aka specificity)
proportion of true negatives correctly classified as negatives

Precision $= \frac{TP}{TP+FP}$
proportion of cases classified as positive that are really positive (aka positive predictive value)

Accuracy $= \frac{TP+TN}{P+N}$
proportion of instances correctly classified.

Error rate $= \frac{FP+FN}{P+N}$
proportion of instanes classified incorrectly

F1 score $= \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$
combination of precision and recall

Example. We have P+N=8 observations available for testing two models (classifiers) that have been trained already.

Y_{true}	Y_1	Y_2
1	1	1
1	1	1
0	1	1
1	1	0
0	0	1
0	0	1
1	0	0
0	0	1
0	1	1

M1

T. \ P.	0	1	
0	3	2	5
1	1	2	3

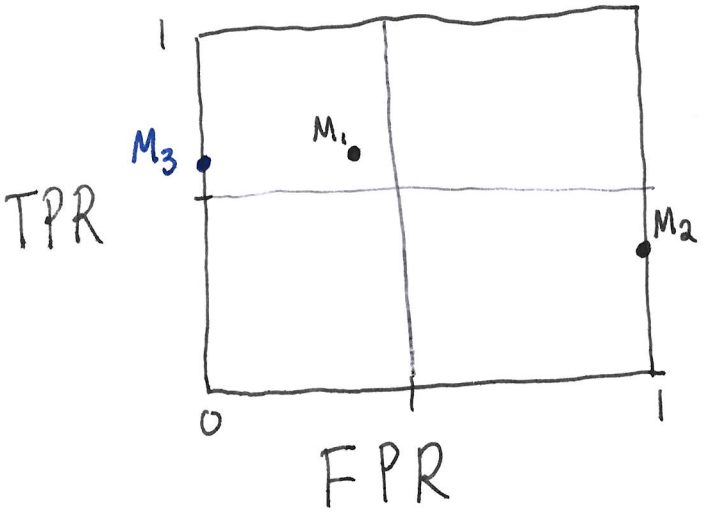
TPR = 0.67
FPR = 0.4

M2

T. \ P.	0	1	
0	0	5	5
1	2	1	3

TPR = 0.33
FPR = 1

We compare classifiers by plotting TPR vs. FPR
This is known as the Receiver Operating characteristic (ROC) graph.



Model	TPR	FPR
1	0.67	0.4
2	0.33	1

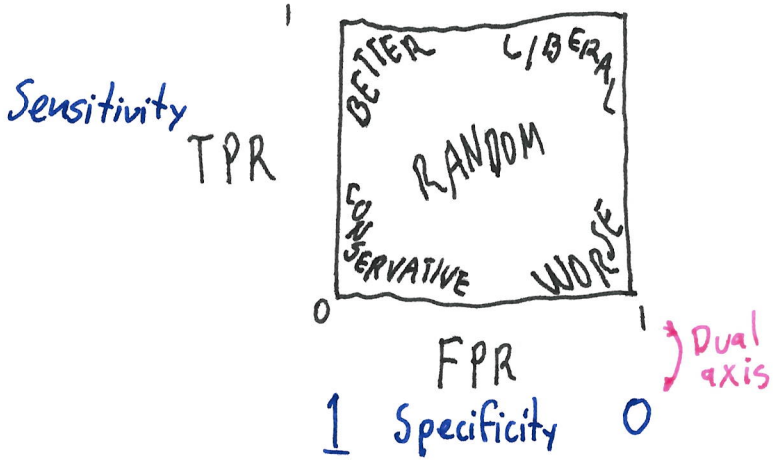
"Model 1 is a better classifier than Model 2"

140324 Model 3 reverses predictions Y_2

Model	TPR	FPR
3	0.67	0

140324
(1400)

The ROC graph has regions with clear interpretation.
The adjectives apply to classifiers.



Note that flipping predicted values $0 \to 1$, $1 \to 0$ will improve a bad classifier (worsen a good classifier)

Confusion matrices for some of these extreme cases

Ideal	Worst	Liberal	Conservative	Random
N O / O P	O N / P O	O N / O P	N O / P O	N/2 N/2 / P/2 P/2
TPR=1	0	1	0	1/2
FPR=0	1	1	0	1/2

4.2 The ROC curve and AUC

Classifiers are points in ROC graph and we compare them according to the position in the graph
- closeness to top left corner FPR=0, TPR=1
- relative positions and importance of 'positive/negative'.

There are cases where we can tune the classifier through some parameter. Then the plot of points in the ROC graph depends on such parameter and we have the ROC curve.

Example (logistic model) with linear predictor $\beta_0 + \beta_1 x$

Training data

x	y
1	0
-7	0
-3	0
5	1
13	1
9	0

and we have $\hat{\beta}_0 = -2.0948$, $\hat{\beta}_1 = 0.2862$

so we can predict $\Pr(Y=1 \mid x) = \dfrac{e^{\hat{\beta}_0 + \hat{\beta}_1 x}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x}} = p(x)$

Test data — "set a threshold and produce labels"

x	y	p(x)	0	0.05	0.2	0.3	1
3	0	0.2251	1 ← FP	1	1	0	0
15	1	0.9	1 ✓	1	1	1	0
-1	1	0.0846	1 ✓	1	0	0	0
-5	0	0.0286	1 ← FP	0	0	0	0
11	1	0.7414	1 ✓	1	1	1	0
7	1	0.4771	1 ✓	1	1	1	0

	0	1
0	0	2
1	0	4

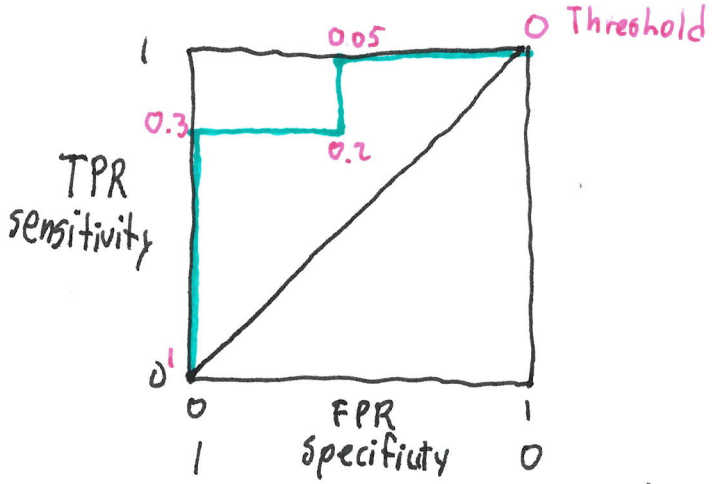
	0	1
0	1	1
1	1	3

	0	1
0	2	0
1	4	0

Threshold	TPR	FPR
0	1	1
0.05	1	0.5
0.2	0.75	0.5
0.3	0.75	0
1	0	0

AUC = 0.875

Using thresholds and values TPR, FPR, we build the ROC curve.



190324 (1400) The area under the (ROC) curve AUC is a useful quantity to compare the performance of classifiers. The closer AUC gets to one, the better the performane. AUC is also known as the c-statistic.

Models for classification

We'll do a survey of models for this task. The emphasis is on the comparison between classifiers.

4.3 Linear classifier

The simplest model for the vector of labels $\underline{Y}$ treats it as if it was continuous. This is a regression with expectation

$$E(Y) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

β_0: intercept; $\beta_1 x_1 + \ldots + \beta_p x_p$: terms for the p explanatory variables

Function lm in R

Using fitted coefficients $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ and test data, we obtain predictions $\hat{y}$. A simple rule is to allocate label "1" if $\hat{y} > 0.5$ and label "0" otherwise.

Advantages
- Simple and intuitive
- Well known and understood

Disadvantages
- Estimates $\hat{y}$ may lie outside $[0,1]$ and be far away from labels "0", "1". (*)
- Difficult to use with more than two labels.

4.4 Logistic classifier

Logistic regression is an established methodology which is part of generalized linear models (GLM). For example techniques for

- standard linear regression
- count data regression
- binary data regression ←

} are all GLMs.

The logistic classifier avoids the inconvenience (*) modelling the probability $p(\underline{x}) = Pr(Y=1|\underline{x})$ in such a way that $p(\underline{x}) \in [0,1]$.

We use a distribution (Bernoulli) to model data and a special parameterization (logistic link) to model the probabilities. The Bernoulli probabilities $p_i = p(\underline{x}_i)$ obbey

$$p_i = p(\underline{x}_i) = \frac{e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p}}$$

← data is $Y_i \sim Ber(p_i)$

Inverting the logistic puts the log-odds as a linear function of covariates

multiple logistic regression

$$\log \frac{p(\underline{x}_i)}{1 - p(\underline{x}_i)} = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

To train the logistic model, we use maximum likelihood estimation (outside the scope of MTH6101).

In R use function glm with option family = "binomial"

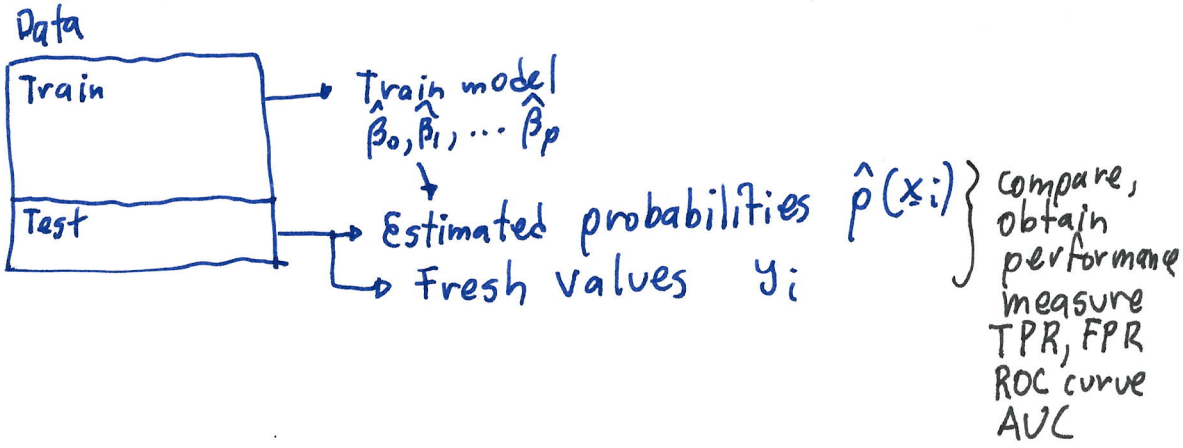
(By default, the link to "binomial" is logistic)

After training, we have estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots \hat{\beta}_p$ and compute probabilities with fresh data

$$\hat{p}(\underline{x}_i) = \left(e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p}\right) / \left(1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p}\right)$$

↑ fresh data point

Together with (fresh) response values "true observations" we use these probabilities to describe the performance of the classifier (and compare).



Example - Default data, logistic classifier. We have three classifiers for response 'default'.

1: 'default' → $\beta_0 + \beta_1$ 'balance'

2: 'default' → $\beta_0 + \beta_1$ 'income'

3: 'default' → $\beta_0 + \beta_1$ 'student'

Data has $n = 10{,}000$ observations, split 80/20 (5 folds)

4 folds to train (8000 obs.), 1 fold to validate\test (2000 obs.)

Classifier	AUC	
1	0.961	← Good classifier
2	0.5232	} Like random classifiers
3	0.4895	

4.5 K-nearest neighbors classifier (KNN classifier)

A new observation is classified (labeled) according to the labels of the K nearest neighbors to it.

labels are unknown this is part of testing data

labels are known this is part of training data

R function Knn

For this classifier, the training and validation steps are performed in a single run; there are no parameters estimates produced (no ROC curve).

The value of K chosen is usually odd so that there will be no ties among nearest neighbors. In the case of K even (or more than two labels), the label is allocated at random.

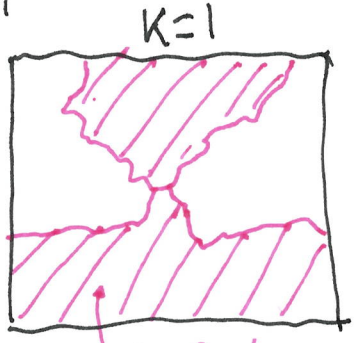
Example KNN, default data, same preparation 80/20 as earlier. We consider K=1, 2, 3, 4, 5.

210324 (1400) Analysis performed with standardized variables.

K	1	2	3	4	5	
TPR	0.3	0.316	0.416	0.3	0.383	← as high as possible
FPR	0.024	0.019	0.009	0.009	0.004	← as low as possible

↓ suggested classifier

The KNN classifier is simple to explain and can produce complex regions for classification. It requires the covariates to be continuous, though.



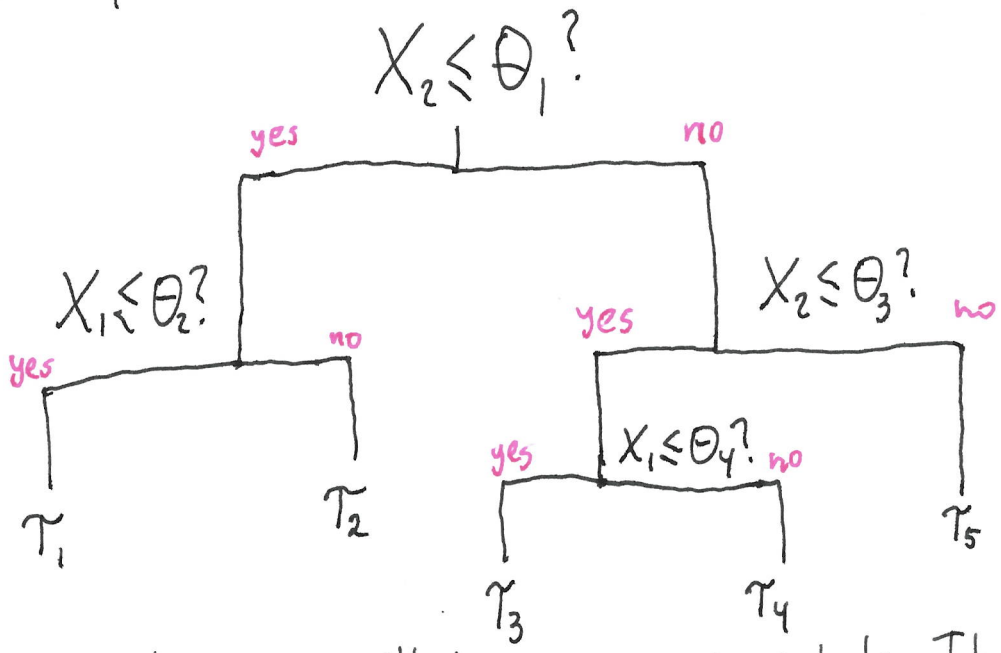
← See page 83 notes

4.6 Classification tree

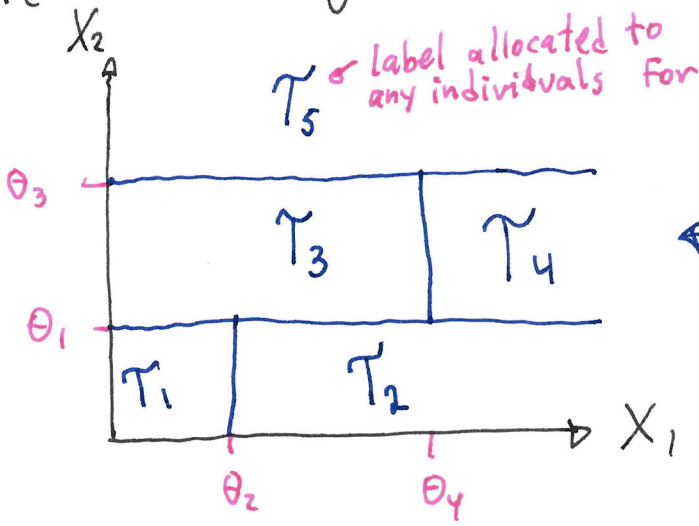
A classification tree is a structure in which 'leaves' represent class labels and 'branches' represent conjunctions of variables that lead to those class labels.

Example of tree with variables X_1, X_2

Here $\tau_1, \tau_2, \ldots$ are terminal nodes "leaves" i.e. labels



For us, leaves will be our 0/1 labels. They are associated with regions in the space of covariates.



← Partitioning explained and induced by the tree

R function tree

Advantages

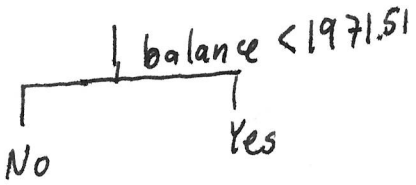
- Simple to understand and interpret
- Able to handle categorical and numerical data
- Simple to analyze, requiring basic data preparation
- Can be validated with data
- No distributional assumptions, simple in terms of its statistics
- Mirrors human decision-making more than other approaches

Limitations

- The problem of training a tree is known to be hard (computationally costly). Algorithms use greedy search and there is no guarantee to reach the optimal tree for a ~~given~~ given data set.
- The method of fitting trees to data can be very non-robust. A small change in data may have huge implications in the resulting tree and predictions.
- It may overfit the data and be overly complicated.

260324 (1400) Example 'Default' data set. Same partition of data as earlier, For validation P=60, N=1940.

Tree (after pruning and removing redundancies)



Confusion matrix

Tr. \ Pr.	0	1
0	1934	6
1	43	17

$FPR = \frac{6}{1940} = 0.003$

$TPR = \frac{17}{60} = 0.283$

4.7 Linear discriminant analysis (LDA)

We use a linear combination of variables to separate two (or more) classes of objects.

LDA assumes $p(\underline{x}|y=0)$ and $p(\underline{x}|y=1)$ are both normally distributed $(\underline{\mu}_0, \underline{\Sigma}_0)$ $(\underline{\mu}_1, \underline{\Sigma}_1)$

conditional distrbs of $\underline{x}$ given y

$\underline{\mu}$: vector of means

$\underline{\Sigma}$: variance covariance matrix

The Bayes optimal solution predicts points as being from the second class ($y=1$, positive) if the log-likelihood ratios are bigger than a threshold T

$$\underbrace{(\underline{x}-\underline{\mu}_0)^T\underline{\Sigma}_0^{-1}(\underline{x}-\underline{\mu}_0)+\log|\underline{\Sigma}_0|}_{\text{log of multivariate normal density } (\underline{\mu}_0,\underline{\Sigma}_0)} - \underbrace{(\underline{x}-\underline{\mu}_1)^T\underline{\Sigma}_1^{-1}(\underline{x}-\underline{\mu}_1)-\log|\underline{\Sigma}_1|}_{\text{log of multivariate normal } (\underline{\mu}_1,\underline{\Sigma}_1)} > T$$

If matrices $\underline{\Sigma}_0, \underline{\Sigma}_1$ are different, this is quadratic discriminant analysis.

In LDA, we assume that the covariance matrices are identical $\underline{\Sigma}=\underline{\Sigma}_0=\underline{\Sigma}_1$ and that $\underline{\Sigma}$ is full rank. After cancelling terms, the decision criterion above is a threshold on the dot product

R function `lda`

$$\underline{w}\cdot\underline{x} > c$$

(1500)

with $\underline{w}=\underline{\Sigma}^{-1}(\underline{\mu}_1-\underline{\mu}_0)$ and $c=\frac{1}{2}\left(T-\underline{\mu}_0^T\underline{\Sigma}^{-1}\underline{\mu}_0+\underline{\mu}_1^T\underline{\Sigma}^{-1}\underline{\mu}_1\right)$.

In other words, observation $\underline{x}$ belongs to a certain class depending on its location regarding a hyperplane perpendicular to $\underline{w}$. The location of the hyperplane is defined by the constant c.

Example 'Default' data, same partition 4:1 as before (80/20)

$P=60$ and $N=1940$

Confusion matrix

Tr. \ Pr.	0	1
0	1937	3
1	44	16

$FPR=\frac{3}{1940}=0.0015$

$TPR=\frac{16}{60}=0.266$

$AUC=0.9638$

Other topics in classification: quadratic discriminant, neural networks' classifier, Bayesian "naive" classifier; more than two labels...

5 - Lasso and regularization (penalized likelihood techniques)

The aim of penalization techniques is to improve over the performance of traditional estimators (regression).

Brief review of regression.

The model is $\underline{Y} = \underline{X}\underline{\beta} + \underline{\varepsilon}$; usually $\underline{Y}$ has been centered around its mean and the columns of $\underline{X}$ been centered as well (possibly scaled), so that the model has no intercept.

((In what follows, we will use the singular value decomposition $X = \underline{U}\underline{D}\underline{V}^T$.))

To estimate parameters, we consider the usual quadratic criterion

$$SS_\varepsilon = \|\underline{Y} - \underline{X}\underline{\beta}\|_2^2.$$

($\underline{Y}$: observed; $\underline{X}\underline{\beta}$: predicted)

The standard estimate $\hat{\beta}$ is the least squares estimate

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

$$= (\underline{V}\underline{D}\underbrace{\underline{U}^T\underline{U}}_{I}\underline{D}\underline{V}^T)^{-1}\underline{V}\underline{D}\underline{U}^T\underline{Y} = (\underline{V}\underline{D}^2\underline{V}^T)^{-1}\underline{V}\underline{D}\underline{U}^T\underline{Y} = \underline{V}\underline{D}^{-2}\underbrace{\underline{V}^T\underline{V}}_{I}\underline{D}\underline{U}^T\underline{Y}$$

($\underline{X}^T$, $\underline{X}$, $\underline{X}^T$)

$$= \underline{V}\underline{D}^{-1}\underline{U}^T\underline{Y}.$$

Consider the predicted values $\hat{Y} = X\hat{\beta}$ that we now develop

$$\hat{Y} = X\hat{\beta} = UDV^T VD^{-1}U^T Y = UU^T Y$$

($V^T V = I$, $DD^{-1} = I$, $X = UDV^T$, $\hat{\beta} = VD^{-1}U^T Y$, $UU^T Y$ = coordinates of Y with respect to orthonormal basis U)

$$H = X(X^TX)^{-1}X^T = UU^T$$

$$H^2 = UU^TUU^T = UU^T \quad (U^TU = I)$$

The predictions $\hat{Y}$ can be written as $\hat{Y} = \sum_{j=1}^{p} u_j u_j^T Y$ (columns of U)

and we note that U^TY are the coordinates of Y with respect to orthonormal basis U.

280324
(1400)

Regularization

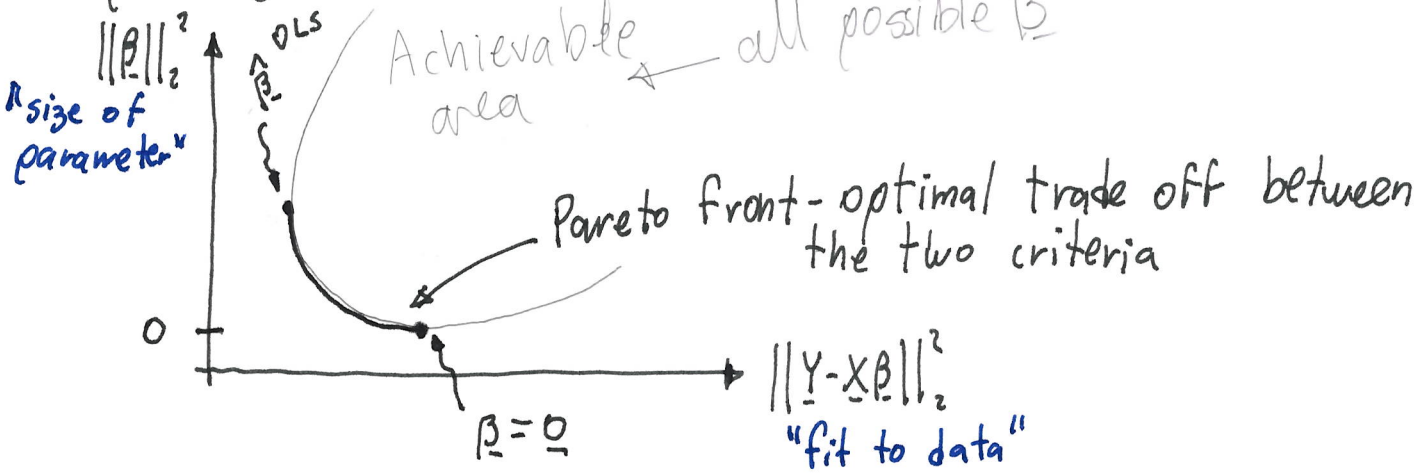
lm R functions	$\|\|Y - X\beta\|\|_2^2$ ← OLS
ridge (own)	$\|\|Y - X\beta\|\|_2^2 + \lambda\|\|\beta\|\|_2^2$ ← Ridge (L_2 penalization)
lars	$\frac{1}{2}\|\|Y - X\beta\|\|_2^2 + \lambda\|\|\beta\|\|_1$ ← Lasso (L_1 penalization)
glmnet	$\|\|Y - X\beta\|\|_2^2 + \lambda[\alpha\|\|\beta\|\|_1 + (1-\alpha)\|\|\beta\|\|_2^2]$ ← Elastic net (Mixture of L_1, L_2)

5.1 Ridge regression

We want to find β that achieves a small value of $\|\|Y - X\beta\|\|_2^2$ in such a way that β

is not too large. We measure the size of $\underline{\beta}$ using square euclidean norm $||\underline{\beta}||_2^2$

We have a dual objective problem in optimization theory.



This dual problem can be scalarized to have a single objective function. After simplification, the objective function is

$$R = \underbrace{||\underline{Y}-\underline{X}\underline{\beta}||_2^2}_{SS_E} + \lambda \underbrace{||\underline{\beta}||_2^2}_{\text{sq. Eucl. norm of } \underline{\beta}} \quad \lambda: \text{parameter}, \quad \sum_{i=1}^{p} \beta_i^2$$

to be minimized for (fixed) positive λ. This is a regularized (L_2 regularized) least squares problem.

Minimizing R.

$$R = (\underline{Y}-\underline{X}\underline{\beta})^T(\underline{Y}-\underline{X}\underline{\beta}) + \lambda \underline{\beta}^T\underline{\beta}$$

$$= \underline{Y}^T\underline{Y} - 2\underline{\beta}^T\underline{X}^T\underline{Y} + \underbrace{\underline{\beta}^T\underline{X}^T\underline{X}\underline{\beta} + \lambda \underline{\beta}^T\underline{\beta}}_{\underline{\beta}^T(\underline{X}^T\underline{X}+\lambda\underline{I})\underline{\beta}}$$

$$\frac{\partial R}{\partial \underline{\beta}} = -2\underline{X}^T\underline{Y} + 2\underline{X}^T\underline{X}\underline{\beta} + 2\lambda\underline{I}\underline{\beta}$$

The estimate that minimizes R is the following closed form:

$$\hat{\beta}^R = \hat{\beta}^R(\lambda) = (X^TX + \lambda I)^{-1} X^T Y$$

0 20424 (1400)

$$= V(D^2 + \lambda I)^{-1} D U^T Y \quad \leftarrow \text{using s.v.d. of } X = UDV^T$$

Diagonal matrix with entries $\frac{d_1}{d_1^2+\lambda}, \ldots, \frac{d_p}{d_p^2+\lambda}$

For $\lambda \to 0$, $\hat{\beta}^R \to \hat{\beta}$ ← ordinary least squares

$\lambda \to \infty$, $\hat{\beta}^R \to 0$ ← in the limit, ridge estimate shrinks to zero

When we compute predictions, we have

$$\hat{Y}^R = X\hat{\beta}^R = \underbrace{U D(D^2+\lambda I)^{-1} D}_{\text{diagonal}} U^T Y = \sum_{j=1}^{p} u_j \frac{d_j^2}{d_j^2+\lambda} u_j^T Y$$

Hat matrix

Columns of U

Factor of shrinkage of ridge regression compared with regression predictions $\sum_{j=1}^{p} u_j u_j^T Y$

Advantages of the ridge estimator $\hat{\beta}^R$

As we increase λ, moving along the ridge, shrinkage of coefficients leads to a reduction in the variance of predictions, at the expense of increasing bias.

Ridge is simpler than subset selection and may predict better yet it is not a variable selection technique.

A key problem in ridge regression is the selection of λ for the final model. This can be done by cross-validation.

The <u>ridge trace</u> is the plot of $\hat{\beta}^R = \hat{\beta}^R(\lambda)$ against λ. This is done for a range of values of λ, and each entry of $\hat{\beta}^R$ is plotted. Alternatively, the horizontal axis can be <u>percentage</u> of <u>shrinkage</u>

$$\left|\left|\hat{\beta}^R(\lambda)\right|\right|_2^2 \Big/ \left|\left|\hat{\beta}\right|\right|_2^2$$

sq. Eucl. norm of ridge estimate $\hat{\beta}^R$

s.s.e.n. of O.L.S. $\hat{\beta}$

and another way of plotting uses the "effective degrees of freedom" $\mathrm{tr}(H_\lambda)$ in the horizontal axis.

We have $\mathrm{tr}(H_\lambda) = \mathrm{tr}\left(\underline{U}\,\underline{D}\left(\underline{D}^2+\lambda\underline{I}\right)^{-1}\underline{D}\,\underline{U}^T\right)$ ← $\mathrm{tr}(AB)=\mathrm{tr}(BA)$ "circular property of the trace"

(1500)

$$= \mathrm{tr}\left(\underline{U}^T\underline{U}\,\underline{D}\left(\underline{D}^2+\lambda\underline{I}\right)^{-1}\underline{D}\right)$$

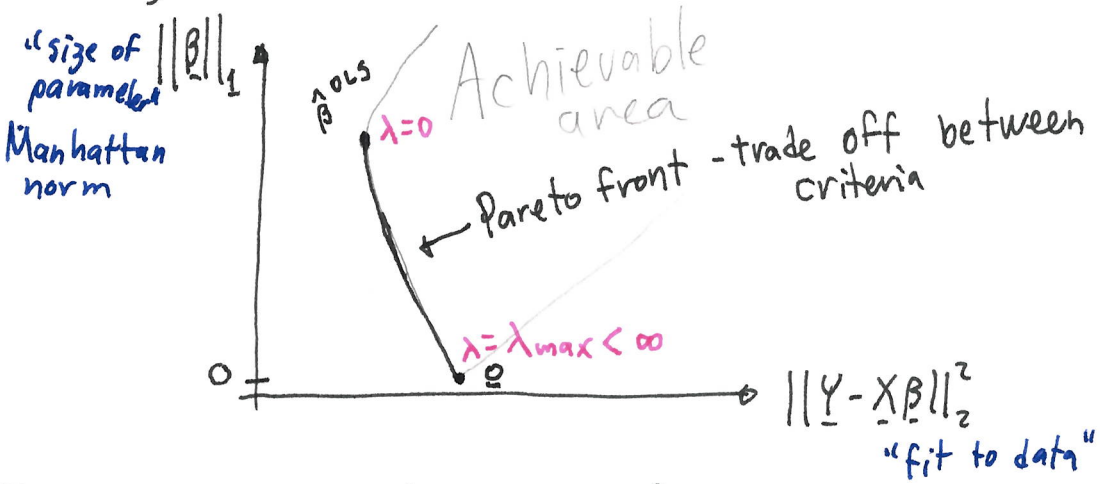
$$= \mathrm{tr}\left(\underline{D}\left(\underline{D}^2+\lambda\underline{I}\right)^{-1}\underline{D}\right) = \sum_{j=1}^{p}\frac{d_j^2}{d_j^2+\lambda}$$

$$0 < \mathrm{tr}(H_\lambda) \le p$$

5.2 Lasso

A drawback of $\hat{\beta}^R$ is that it generally involves all variables in the sense that its entries are not zero for finite λ. Lasso (least absolute shrinkage and selection operator) overcomes this disadvantage using L_1 penalization.

Using a similar dual-objective diagram...



The objective function for the lasso is an L_1 penalization of the sum of squares of the error

$$L = \frac{1}{2}||Y-X\beta||_2^2 + \lambda||\beta||_1$$

$\frac{1}{2}$: needed for technical reason

$||Y-X\beta||_2^2$: sum of squares of the error

λ: parameter $\lambda \geq 0$

$||\beta||_1$: Manhattan norm $\sum_{i=1}^{p}|\beta_i|$

Example with data (one explanatory variable, $n=7$)

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$ ← before centering

X Y

$$\frac{1}{7}\begin{pmatrix} -4 & -1 \\ -4 & -1 \\ -4 & -1 \\ 3 & -1 \\ 3 & -1 \\ 3 & 6 \\ 3 & -1 \end{pmatrix}$$ ← after centering

$$X^TX = \frac{12}{7}$$

$$X^TY = \frac{3}{7}$$

$$Y^TY = \frac{6}{7}$$

OLS $\hat{\beta} = (X^TX)^{-1}X^TY = \frac{7}{12}\cdot\frac{3}{7} = \frac{3}{12} = \frac{1}{4} = 0.25$

Ridge $\hat{\beta}^R = (X^TX+\lambda I)^{-1}X^TY = \left(\frac{12}{7}+\lambda\right)^{-1}\cdot\frac{3}{7} = \frac{3}{12}\left(\frac{12}{12+7\lambda}\right)$

Lasso $\hat{\beta}^L = \frac{3}{12}\left(1-\frac{7}{3}\lambda\right)$

→ when $\lambda = \frac{3}{7}$ we have $\hat{\beta}^L = 0$

Let us compute the lasso estimate

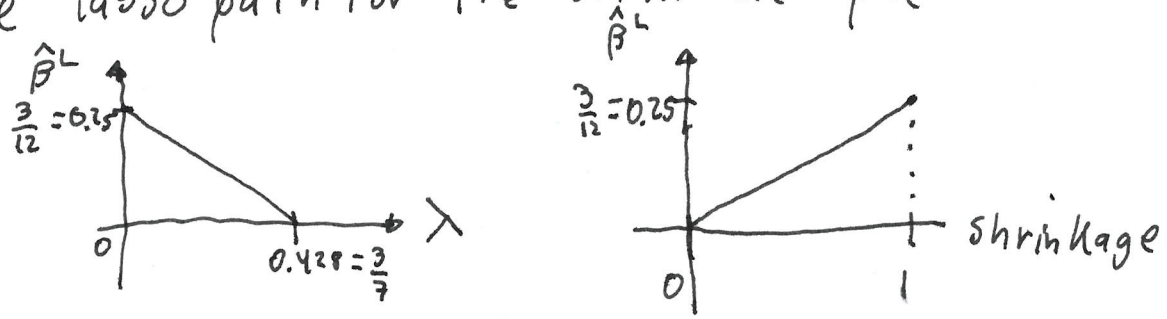
$L = \frac{1}{2} Y^T Y - \beta^T X^T Y + \frac{1}{2}\beta^T X^T X \beta + \lambda \|\beta\|_1$ which for this single parameter case becomes

$L = \frac{6}{14} - \frac{3}{7}\beta + \frac{12}{14}\beta^2 + \lambda\beta$ ← β equals $|\beta|$ in the case $\beta > 0$. We start here as $\hat{\beta} = 0.25 > 0$.

$\frac{\partial L}{\partial \beta} = -\frac{3}{7} + 2\cdot\frac{12}{14}\beta + \lambda$

$= -\frac{3}{7} + \frac{12}{7}\beta + \lambda$ ~~~ (equate to zero and solve for β)

The lasso path for the small example



040224
(1400)

Contrary to ridge, there is no general closed formula for the lasso estimate $\hat{\beta}^L = \hat{\beta}^L(\lambda)$ that minimizes criterion L.

Lasso path is the collection of $\hat{\beta}^L$ as λ takes values in $[0, \infty)$. However $\hat{\beta}^L$ shrinks to zero for a finite λ. It consists of piecewise linear trajectories, which change direction at breakpoints.

Lasso has a dual function as estimation procedure and as variable selection method. Models from lasso are simpler to interpret than those from ridge and are sparser (have less non-zero coefficients).

Example (notes) Credit card data. Six explanatory variables, data split 75/25.

R function lars

As is common practice, data is centered (and scaled) so no intercept in the analysis.

$\hat{\beta}$ beta from lars

λ	Income	Limit	Rating	Cards	Age	Education	Shrinkage $\|\|\hat{\beta}^\lambda\|\|_1 / \max \|\|\hat{\beta}\|\|_1$
259.79	—	—	—	—	—	—	0/2.01 = 0
85.35	—	—	0.59	—	—	—	0.59/2.01 = 0.29
37.40	—	0.11	0.64	—	—	—	0.75/2.01 = 0.37
14.09	-0.37	0.35	0.76	—	—	—	1.5/2.01 = 0.74
7.86	-0.47	0.42	0.80	—	-0.02	—	1.71/2.01 = 0.85
3.12	-0.54	0.58	0.71	0.02	-0.03	—	1.98/2.01 = 0.93
0	-0.59	0.70	0.63	0.03	-0.04	-0.01	2.01/2.01 = 1

λ values

Rows correspond to breakpoints, indexed by either λ or by shrinkage

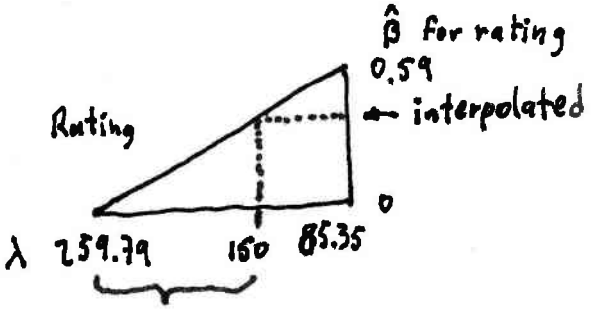
Each column has estimates at every breakpoint

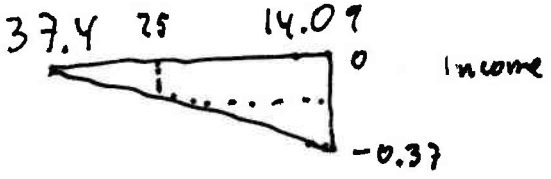
Top row ← all coefficients shrink to zero

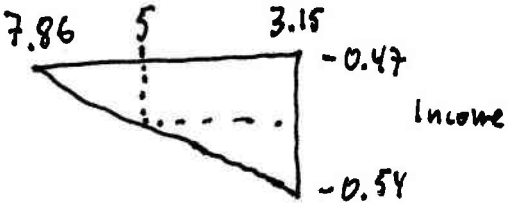
Bottom row ← O.L.S.

090424 (1400) Let us interpolate and compute $\hat{\beta}^L(150)$

$$\hat{\beta}^L(150) = (0, 0, 0.3711, 0, 0, 0)^T$$







$$\frac{|150 - 259.79|}{|85.35 - 259.79|} = \frac{109.79}{174.4331} = 0.629$$

Elastic net

An extension of the L_1 penalization of lasso and the squared L_2 penalization of ridge regression are elastic nets. The penalization is a combination of L_1 and squared L_2, controlled with an additional parameter α:

$$E = ||\underline{Y} - \underline{X}\underline{\beta}||_2^2 + \lambda\left[\alpha||\underline{\beta}||_1 + (1-\alpha)||\underline{\beta}||_2^2\right]$$

(SS_E; parameter; controls the mixture)

R function glmnet

$\alpha = 1 \Rightarrow E = L$ lasso

$\alpha = 0 \Rightarrow E = R$ ridge

Exercise. For the small example of 020424, show that minimizing E leads to the path

$$\hat{\beta}^E(\lambda,\alpha) = \frac{6/7 - \lambda\alpha}{24/7 + \lambda(1-\alpha)}$$

The elastic net path is not anymore a series of piecewise linear trajectories as with the lasso. The trajectories are continuous, ridge-like but with the benefit that some coefficients shrink to zero, and that the shrinkage of all is achieved for finite λ. (Although the glmnet does not show) the elastic net path has a series of breakpoints similar to lasso, where the path changes direction.

(1600)

5.3 Penalized likelihood

So far we have considered the standard least squares criterion $||\underline{Y} - \underline{X}\underline{\beta}||_2^2$ on top of which we added a penalty based on the size of the parameter vector ← L_1 lasso, L_2^2 ridge, L_1, L_2^2 elastic net.

} minimization

We could start from a different point, that is, ~~if~~ to consider <u>likelihood</u> as the basis for estimation, and on top of the likelihood, add a penalty of the type elastic net.

If the likelihood is denoted as $\ell(\underline{\beta})$, ← log-likelihood
↑ parameter vector

the penalized version is

maximized

$$\ell_p(\underline{\beta}) = \underbrace{\ell(\underline{\beta})}_{\text{likelihood}} - \lambda \underbrace{g(\underline{\beta}, \alpha)}_{\text{penalty by the size of } \underline{\beta}},$$

↑ penalized

R function glmnet

Where we use $g(\underline{\beta}, \alpha) = \alpha \|\beta\|_1 + (1-\alpha)\|\underline{\beta}\|_2^2$ ← Elastic net penalty.

Example. (as in the lab) Classification of glass, of 7 types of glass we are interested in type = 7. There are 9 variables available. We use glmnet with "binomial" distribution and $\alpha = 1/10$ (close to ridge penalization). K=3 folds are used.