1 Lecture 2: 10th Of March 2020

We work with n statistical units, each unit has been observed according to p variables (features). For example: $\underline{x}_1 = (x_{11}, \dots, x_{1p})^T \in \mathbb{R}^p$ is the first statistical unit, while $\underline{x}_n = (x_{n1}, \dots, x_{np})^T \in \mathbb{R}^p$ is the $n-$th statistical unit.

If we combine the statistical units we get the data matrix (data frame): $\mathbb{X} = \begin{bmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \vdots & \vdots \\ x_{n1} & \dots & x_{np} \end{bmatrix}$ in which each column is a feature and each row is a statistical unit.

The data cloud lives in an euclidean space of p dimension. We will assume often that $n >> p$ otherwise we will see strange thing happen!

We often have a special variable y observed with the other features which captures what we want to explain or predict in terms of the other features: $\mathbb{X} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \mathbb{X}y$

It may happen that y is a categorical variable, in this case we want to use the information coming from $\mathbb{X}$ to say something about membership to a group (e.g: classification problem).

We have a random vector $\underline{x} \in \mathbb{R}^p$ and we have a random variable $y \in \mathbb{R}$ which is a quantity we want to predict. What is the best function $f : \mathbb{R}^p \to \mathbb{R}$ to predict y in terms of $\underline{x}$?

We look for the function f that minimises the squared expected error: $\mathbb{E}[(y - f(\underline{x}))^2]$ Note that we minimise the squared expected error because we don't want the positive error to compensate for the negative error. Otherwise we could use different error measures: such as the absolute values or such as elevating to the third power or taking the square root.

Since we want to work in L^2 we consider the squared error!

Note: If we want to find $\arg\min_k \mathbb{E}[(y-k)^2] = \mathbb{E}[y] \implies$ The constant k which minimise that function is $E[y]$

Note:$\mathbb{E}[y|\underline{x}]$ is the Radon-Nykodym Derivative!

In our case f is not a constant function:

$$\mathbb{E}[(y - f(\underline{x}))^2] = \mathbb{E}[(y - \mathbb{E}[y|\underline{x}] + \mathbb{E}[y|\underline{x}] - f(\underline{x}))^2] = \mathbb{E}[(a+b)^2] =$$

$$= \mathbb{E}[(y - \mathbb{E}[y|\underline{x}])^2] + \mathbb{E}[(\mathbb{E}[y|\underline{x}] - f(\underline{x}))^2] + 2\mathbb{E}[(y - \mathbb{E}[y|\underline{x}])(\mathbb{E}[y|\underline{x}] - f(\underline{x}))]$$

Recall now that given w, z then: $\mathbb{E}[w] = \mathbb{E}[\mathbb{E}[w|z]]$ so in our case we have:

$$2\mathbb{E}[(y - \mathbb{E}[y|\underline{x}])(\mathbb{E}[y|\underline{x}] - f(\underline{x}))] = 2\mathbb{E}[\mathbb{E}[(y - \mathbb{E}[y|\underline{x}])(\mathbb{E}[y|\underline{x}] - f(\underline{x}))|\underline{x}]] = 2\mathbb{E}[(\mathbb{E}[y|\underline{x}] - f(\underline{x}))\mathbb{E}[y - \mathbb{E}[y|\underline{x}]|\underline{x}]] =$$

$$= 2\mathbb{E}[(\mathbb{E}[y|\underline{x}] - f(x\underline{x})0] = 2\mathbb{E}[0] = 0$$

because $\mathbb{E}[y|\underline{x}] - f(\underline{x}))$ is a function of $\underline{x}$ and its a number with respect to $\underline{x}$ because: $\mathbb{E}[y - \mathbb{E}[y|\underline{x}]|\underline{x}] = 0$

Therefore: $\mathbb{E}[(y - f(\underline{x}))^2] = \mathbb{E}[(y - \mathbb{E}[y|\underline{x}])^2] + \mathbb{E}[(\mathbb{E}[y|\underline{x}] - f(\underline{x}))^2]$ We need to find f minimising that expression so: $f(\underline{x}) = \mathbb{E}[y|\underline{x}]$ This is because the first term is not changeable by acting on f

Conclusion: We have our optimal predictor, although there is always a difference between y and the best prediction: $\epsilon = y - f(\underline{x})$ This we cannot capture: its the part of y not reachable with information given by $\underline{x}$ Indeed ϵ is called *residual* (i.e: it's an error)!

In any case our model is given by: $y = f(\underline{x}) + \epsilon$ where $f(\underline{x}) = \mathbb{E}[y|\underline{x}]$

Then: $\mathbb{E}[y] = \mathbb{E}[f(\underline{x}) + \epsilon] = \mathbb{E}[f(\underline{x})] + \mathbb{E}[\epsilon] = \mathbb{E}[\mathbb{E}[y|\underline{x}]] + \mathbb{E}[\epsilon] = \mathbb{E}[y] + \mathbb{E}[\epsilon] \implies \mathbb{E}[\epsilon] = 0$

In order to find $\mathbb{E}[y|\underline{x}]$ we need to know the joint distribution of $y|\underline{x}$ but we don't know it, so we need to use data to estimate $f(\underline{x}) = \mathbb{E}[y|\underline{x}]$

Suppose that $\hat{f}$ is our known estimate for f: how good is it? This is what characterises statistical learning with respect to machine learning! We want to know how good it is when we try to predict some new data that we didn't observed in the training data. Moreover we are also interested in the uncertainty related with this prediction!

Problem: finding $\hat{f}$ the estimate of f via $\mathbb{X}$
Suppose $\underline{x}_0 \in \mathbb{R}^p$ is a new statistical unit, what is a prediction for y_0? What is the **Mean Squared Error (MSE)** we are making? Since we have observed the datum: $\underline{x}_0$ is no longer a random variable, here y_0 is the random variable.

By applying our model found above we have that: $y_0 = f(\underline{x}_0) + \epsilon_0$ Thus:

$$\mathbb{E}_{\mathbb{X}}[(y_0 - \hat{f}(\underline{x}_0))^2] = \mathbb{E}_{\mathbb{X}}[(f(\underline{x}_0) + \epsilon_0 - \hat{f}(\underline{x}_0))^2]$$

$$= \mathbb{E}_{\mathbb{X}}[(f(\underline{x}_0) - \hat{f}(\underline{x}_0)^2] + \mathbb{E}_{\mathbb{X}}[\epsilon_0^2] + 2\mathbb{E}_{\mathbb{X}}[(f(\underline{x}_0) - \hat{f}(\underline{x}_0))(\epsilon_0)] = (f(\underline{x}_0) - \hat{f}(\underline{x}_0))^2 + Var[\epsilon_0] + 0$$

The first term leave the expected value since it is a number and not random variable, moreover the mean of ϵ_0 is null so we have that the second term is the variance of ϵ_0

Moreover: $2\mathbb{E}_{\mathbb{X}}[(f(\underline{x}_0) - \hat{f}(\underline{x}_0))(\epsilon_0)] = 2(f(\underline{x}_0) - \hat{f}(\underline{x}_0))\mathbb{E}_{\mathbb{X}}[\epsilon] = 2(f(\underline{x}_0) - \hat{f}(\underline{x}_0)) \cdot 0 = 0$

Therefore:

$$\mathbb{E}_{\mathbb{X}}[(y_0 - \hat{f}(\underline{x}_0))^2] = (f(\underline{x}_0) - \hat{f}(\underline{x}_0))^2 + Var[\epsilon_0]$$

Note that we can only make the first term as small as possible, but we are not able to capture the second term since it's the irreducible error!

Note: $\underline{x}_0$ is a new point and so it doesn't belong to the train set! Indeed if we predict on the training set presumably we can make a good prediction, so we need to calculate the prediction error on non trained data!

Note: in the above problem the ideal f is also called *regression function.*

Sometimes we have few data points with a specific $\underline{X}$ so we cannot compute $\mathbb{E}[Y|\underline{X}]$ so we can relax the definition and calculate $\hat{f}(\underline{X}) = \text{Average}(Y|\underline{X} \in \mathcal{N}(\underline{X}))$ where $\mathcal{N}(\underline{X})$ is a neighbourhood of $\underline{X}$

The above can only be done if p is small: can we always make p small? We can try with some dimensionality reduction techniques (e.g: PCA)! Indeed when p is large, points are very far apart so we don't have access to a local window: we can take a local neighbourhood of a point in $\mathbb{R}^p$ but it might be empty. This is the *curse of dimensionality!*

Geometric Interpretation of the Curse of Dimensionality:

- Suppose $p = 1$, then consider the sphere of radius 1 and dimension 1 which is $\mathcal{S}^1(1)$ Suppose a random variable has uniform distribution over it: $X \sim U[\mathcal{S}^1(1)]$

  How far should we move from point 0 to capture 10% of the points? Call this radius r, then:
  $0.1 = \frac{\text{Length}(\mathcal{S}^1(r))}{\text{Length}(\mathcal{S}^1(1))} = r \implies r = 0.1$

- Suppose $p = 2$, then consider the sphere of radius 1 and dimension 2 which is $\mathcal{S}^2(1)$ Suppose a random variable has uniform distribution over it: $\underline{X} \sim U[\mathcal{S}^2(1)]$

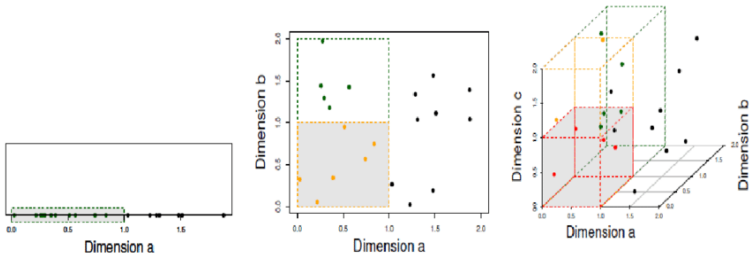
  How far should we move from point 0 to capture 10% of the points? Call this radius r, then:
  $0.1 = \frac{\text{Area}(\mathcal{S}^2(r))}{\text{Area}(\mathcal{S}^2(1))} = r^2 \implies r = \sqrt{0.1} = 0.31$

  Moving from $p = 1$ to $p = 2$ we have that r increased from $r = 0.1$ to $r = 0.31$

- Suppose $p = 100$ then: $r = 0.97$ This means that we need to travel 97% of the radius of universe ($\mathcal{S}^{100}(1)$) to capture 10% of the points!

- Suppose $p = 100$ then: $r = 0.99$

In general: $r^p = 0.1$ so r grows exponentially with p We can see it in the following figure:



If p is large:

- We can reduce the dimension of the space in which we embed our data, in a data driven reduction: we want to project our data in a linear sub-space of dimension less than p in a way such that the variability expressed by the data will be conserved. This can be done with Principle Component Analysis.

- We could use a parametric model: $y = f(\underline{x}) + \epsilon$ where $f(\underline{x}) = \beta_0 + \cdots + \beta_p x_p + \epsilon$ Here we just need to estimate the parameters β_i, and to do this we need some domain knowledge about the physics-engineering of the model.

  The above is a linear model: attention to the meaning of *linear*, we mean linear in the parameters, indeed: $f(x) = \beta_0 + b_1 x + b_2 x^2 + b_3 x^3$ is linear!

Polynomial Interpolation is no good for prediction: the model that fit best the data is good for making prediction only in a deterministic model! If there's lots of variability, fitting exactly the data, is the worst that can be done! The larger the model, the more you use the data for fitting, the less information you have about the variability of the prediction! This is the problem of *over-fitting*!

There is also another important problem: the **Bias-Variance Trade-Off**: how good is a model for prediction? Suppose we are given $\underline{x}_0$, then what is y_0?

Suppose we have $\hat{f}$ and data $\mathbb{X}$ then: $\mathbb{E}_{\mathbb{X}}[(y_0 - \hat{f}(\underline{x}_0))^2] = (f(\underline{x}_0) - \hat{f}(\underline{x}_0))^2 + Var[\epsilon_0]$ this is an estimate of the prediction error only given the data!

We now want to evaluate how good is the model: is this model good in general or is this model good only on $\underline{x}_0$?

$$\mathbb{E}[\mathbb{E}_{\mathbb{X}}[(y_0 - \hat{f}(\underline{x}_0))^2]] = \mathbb{E}[(f(\underline{x}_0) - \hat{f}(\underline{x}_0))^2] + \mathbb{E}[Var[\epsilon_0]] = \mathbb{E}[f(\underline{x}_0)] - \mathbb{E}[\hat{f}(\underline{x}_0)] + \mathbb{E}[(\hat{f}(\underline{x}_0) - \hat{f}(\underline{x}_0))^2] + Var[\epsilon_0] =$$

$$= \mathbb{E}[(f(\underline{x}_0) - \mathbb{E}[\hat{f}(\underline{x}_0)])^2] + \mathbb{E}[(\mathbb{E}[\hat{f}(\underline{x}_0)] - \hat{f}(\underline{x}_0))^2] + 2\mathbb{E}\left[\left(f(\underline{x}_0) - \mathbb{E}[\hat{f}(\underline{x}_0)]\right)\left(\mathbb{E}[\hat{f}(\underline{x}_0)] - \hat{f}(\underline{x}_0)\right)\right] + Var[\epsilon_0] =$$

$$= \mathbb{E}[(f(\underline{x}_0) - \mathbb{E}[\hat{f}(\underline{x}_0)])^2] + \mathbb{E}[(\mathbb{E}[\hat{f}(\underline{x}_0)] - \hat{f}(\underline{x}_0))^2] + 0 + Var[\epsilon_0] =$$

Thus we conclude that:

$$\mathbb{E}[(y_0 - \hat{f}(\underline{x}_0))^2] = (f(\underline{x}_0) - \mathbb{E}[\hat{f}(\underline{x}_0)])^2 + Var[\hat{f}(\underline{x}_0)] + Var[\epsilon_0]$$

Where:

- The first term: $(f(\underline{x}_0) - \mathbb{E}[\hat{f}(\underline{x}_0)])^2$ is the bias squared: it measures how far away, is on average, the model from what we want to estimate.
- The second term: $Var[\hat{f}(\underline{x}_0)]$ is the variability (variance) of the model.
- The third term: $Var[\epsilon_0]$ is the irreducible variability.

When fitting a model we want to minimise all the three terms above: the third term is the irreducible term so we can only hope to minimise the first two terms.

The problem is that there is a bias-variance trade-off: usually taking a model and reducing it's bias makes its variance increase, and vice-versa! We will need to cope with it!

Note that for interpolation with a polynomial we have a model with high variance (over-fitting) so that if we change a bit the data then the model changes dramatically. On the other hand the linear regression model has a lot lower variance!

2 Lecture 5: 16th Of March 2020

Suppose we have n statistical units, and p features: we record the data in the matrix (data frame) $\mathbb{X} \in \mathbb{R}^{n\times p}$ where each column corresponds to a variable (feature) and each row to a statistical unit.

Consider the $i-$th row: $\underline{x}_i^T = (x_{11} \dots x_{1p}) \in \mathbb{R}^p$ This gives us the profile of the $i-$th statistical unit!

Look now at the $j-$th column: $\underline{y}_j = (x_{1j}, \dots, x_{nj})^T \in \mathbb{R}^n$ This gives us a sample from the variable x_j

Therefore, there are p vectors, one for each variable in $\mathbb{R}^n$

$\underline{y}_j = (x_{1j}, \dots, x_{nj})^T \in \mathbb{R}^n$ Taking the mean: $\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$ So the sample mean vector is: $\underline{\bar{x}} = (\bar{x}_1, \dots, \bar{x}_p)^T$

The variance of variable x_j is given by: $S_{jj} = \frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2$ Thus the standard deviation of x_j is given by: $\sqrt{S_{jj}}$

Note: when we take S_{jj} to be an estimator of the variance of the population to get unbiased estimator we divide by $n-1$ and not by n!

Definition: The co-variability (co-variance) between two variables, x_k and x_j, is given by:

$$Cov(x_k, x_j) = \frac{1}{n}\sum_{i=1}^{n}(x_{ik} - \bar{x}_k)(x_{ij} - \bar{x}_{ik}) = S_{kj} \text{ for } k, j = 1, \dots, p \text{ Moreover } Cov(x_j, x_j) = S_{jj}$$

We can introduce the co-variance matrix (computed on the sample): $S = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1p} \\ \vdots & \vdots & \vdots & \vdots \\ S_{p1} & \dots & \dots & S_{pp} \end{bmatrix} \in \mathbb{R}^{p\times p}$ this matrix is symmetric obviously.

Definition: The correlation between two variables is given by: $r_{kj} = \frac{S_{kj}}{\sqrt{S_{kk}S_{jj}}} = Cor(x_k, x_j)$ From this we can define the *Correlation Matrix* $r = \begin{bmatrix} r_{11} & \dots & r_{1p} \\ \vdots & \vdots & \vdots \\ \dots & \dots & r_{pp} \end{bmatrix} \in \mathbb{R}^{p\times p}$ which is a symmetric and positive definite matrix..

Note: $r_{kj} \in [-1, 1]$ is a pure number!

When we know the mean and the standard deviation we can use them for visualising intervals: if we have a variable x_1 with mean $\bar{x}_1 = 1.8$ meters and standard deviation $\sqrt{S_{11}} = 0.03$ meters, then we visualise this information in an interval: $[\bar{x}_1 - 3\sqrt{S_{11}}, \bar{x}_1 + 3\sqrt{S_{11}}] = [1.74, 1.86]$

We know that whatever the distribution of x_1, at least 90% of the population is inside the above interval and this comes for **Chebyshev Inequality:**

$$\text{Frequency}\left[\bar{x}_1 - k\sqrt{S_{11}} \le x_1 \le \bar{x}_1 + k\sqrt{S_{11}}\right] \ge 1 - \frac{1}{k^2}$$

A bit of geometry

Let's consider two vectors $\underline{v}, \underline{w} \in \mathbb{R}^n$ then we have an inner product: $< \underline{v}, \underline{w} >= \underline{v}^T\underline{w}$ From this we also have a notion of length: $\|\underline{v}\| = \sqrt{< \underline{v}, \underline{v} >} = \sqrt{\sum v_i^2}$ Moreover: $\cos(\theta) = \frac{< \underline{v}, \underline{w} >}{\|\underline{v}\|\|\underline{w}\|} = \frac{\underline{v}^T\underline{w}}{(\underline{v}^T\underline{v})(\underline{w}^T\underline{w})} = \frac{\sum_i v_i w_i}{\sqrt{\left(\sum_i v_i^2\right)\left(\sum_i w_i^2\right)}}$

Let $\pi_{\underline{v}|\underline{w}}$ be the projection of $\underline{v}$ on $\underline{w}$ This is given by: $\pi_{\underline{v}|\underline{w}} = \pi_{\underline{v}|\mathcal{L}(\underline{w})}$ where $\mathcal{L}(\underline{w}) = \{\underline{z} : \underline{z} = c\underline{w};\ c \in \mathbb{R}\}$ is the linear space identified by $\underline{w}$ Then:

$$\pi_{\underline{v}|\underline{w}} = \|\underline{v}\| cos(\theta) \frac{\underline{w}}{\|\underline{w}\|} = \|\underline{v}\| \frac{\underline{v}^T \underline{w}}{\|\underline{v}\|\|\underline{w}\|} \frac{\underline{w}}{\|\underline{w}\|} = \frac{\underline{v}^T \underline{w} \cdot \underline{w}}{\|\underline{w}\|^2} = \frac{\underline{w}^T \underline{v} \cdot \underline{w}}{\underline{w}^T \underline{w}} = \frac{\underline{w}\underline{w}^T}{\underline{w}^T \underline{w}} \cdot \underline{v}$$

Note that $\underline{w}\underline{w}^T$ is an operator, and its the matrix that projects any vector $\underline{v}$ on the linear space generated by $\underline{w}$. The denominator is just a normalisation number, so that we have orthogonal projector!

Given $\mathbb{X} = [\underline{y}_1, \ldots, \underline{y}_p]$ where $\underline{y}_j \in \mathbb{R}^n$ sample from x_j Suppose now for example that: $\underline{y}_1 \in \mathbb{R}^n$ is a sample of heights.

In $\mathbb{R}^n$ there is a linear space where there is no statistics, since there is no variability: its the linear space generated by $\underline{1} = (1, \ldots, 1)^T$ which is $\mathcal{L}(\underline{1})$ Then:
if $\underline{v} \in \mathcal{L}(\underline{1})$ then: $\underline{v} = c\underline{1} = (c, \ldots, c)^T$ $c \in \mathbb{R}$ so since there is no variability there is nothing to say!

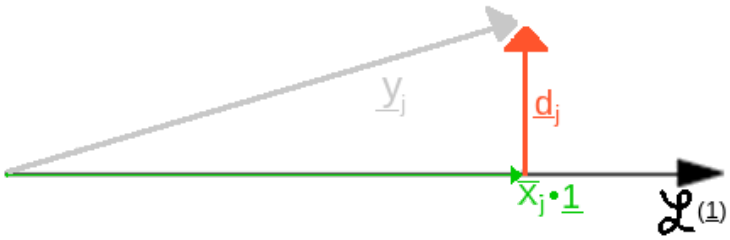
What is the closest approximation of $\underline{y}_1$ where there is no statistic? Geometrically it's the orthogonal projection, which is given by:

$$\pi_{\underline{y}_1|\underline{1}} = \frac{\underline{1}\underline{1}^T}{\underline{1}^T\underline{1}} \cdot \underline{y}_1 = \frac{1}{n}\left(\sum_{i=1}^{n} x_{i1}\right)\underline{1} = \bar{x}_1 \cdot \underline{1} = (\bar{x}_1, \ldots, \bar{x}_1)^T$$

The mean is the best approximation we can get for the vector of observation when we require there is no variability among data!

As it happens with all projection, even here there is the deviation vector $\underline{d}_1$, which is left out: this means that the mean is a poor summary!

$$\pi_{\underline{y}_j|\underline{1}} = \bar{x}_j \cdot \underline{1} \implies \underline{d}_j = \underline{y}_j - \bar{x}_j \cdot \underline{1} = \begin{bmatrix} x_{1j} - \bar{x}_j \\ \vdots \\ x_{nj} - \bar{x}_j \end{bmatrix}$$



We can see that the longer $\underline{d}_j$ the worse the approximation:

$$\|\underline{d}_j\| = \sqrt{\underline{d}_j^T \underline{d}_j} = \sqrt{\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2} = \sqrt{n}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2} = \sqrt{nS_{jj}}$$

This means that the length of the deviation vector is the standard deviation multiplied by $\sqrt{n}$ This is exactly why the standard deviation is talking about variability: it's talking about the information not captured by the mean!

$\underline{y}_j = \bar{x}_j \cdot \underline{1} + \underline{d}_j$ and $\underline{y}_k = \bar{x}_k \cdot \underline{1} + \underline{d}_k$ where: $\bar{x}_j \cdot \underline{1}, \bar{x}_k \cdot \underline{1} \in \mathcal{L}(\underline{1})$ and $\underline{d}_j, \underline{d}_k \in \mathcal{L}^{\perp}(\underline{1})$

Note: the orthogonal linear space has dimension $n - 1$, equal to its degrees of freedom.

Let θ_{jk} be the angle between the deviations $\underline{d}_j$ and $\underline{d}_j$ Then:

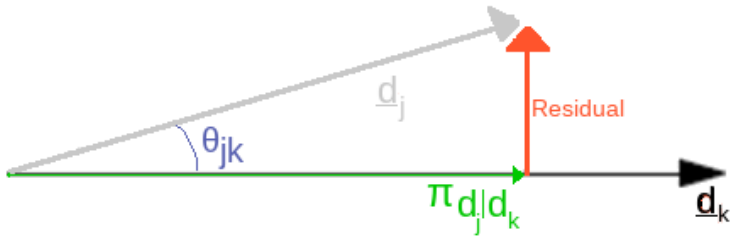
- If $\theta_{jk} = 0 \implies \underline{d}_j = \beta \underline{d}_k$ where $\beta \in \mathbb{R}$ Thus: $\underline{d}_j \in \mathcal{L}(\underline{d}_k)$ thus:

$$\underline{y}_j - \bar{x}_j \cdot \underline{1} = \beta(\underline{y}_k - \bar{x}_k \cdot \underline{1}) \implies \underline{y}_j = \bar{x}_j \cdot \underline{1} + \beta \underline{y}_k - \beta \bar{x}_k \cdot \underline{1}$$

  It means that there is a linear relation between the variable x_j and x_k

- IF $\theta_{jk} = \frac{\pi}{2}$ then there is no component of $\underline{d}_j$ that we can explain through $\underline{d}_k$ So the two deviation vectors have no information related!

What if $\theta_{jk} \in \left[0, \frac{\pi}{2}\right]$? Then there is some information $\pi_{\underline{d}_j|\underline{d}_k}$ about x_j that is contained in $\underline{d}_k$ and some that is left out:



The residual information left out is given by: $\underline{d}_j - \pi_{\underline{d}_j|\underline{d}_k}$ Then we need to compute θ_{jk} thus:

$$\cos(\theta_{jk}) = \frac{\underline{d}_j^T \underline{d}_k}{\|\underline{d}_j\| \|\underline{d}_k\|} = \frac{\sum_{i=1}^{n} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)}{\sqrt{\left(\sum_{i=1}^{n} (x_{ij} - \bar{x}_j^2)\right)\left(\sum_{i=1}^{n} (x_{ik} - \bar{x}_k^2)\right)}} = \frac{S_{jk}}{\sqrt{S_{jj} S_{kk}}} = Corr(x_j, x_k) = r_{jk} \in [-1, 1]$$

No wonder that the correlation is a number in $[-1, 1]$ since it's equal to the cosine! From the above we can see that:

- If $\theta_{jk} = 0 \implies cos(\theta_{jk}) = 1 \implies r_{jk} = 1 \implies \underline{d}_j \in \mathcal{L}(\underline{d}_k)$ This means that there is perfect linear relation between the two variables
- If $\theta_{jk} = \frac{\pi}{2} \implies cos(\theta_{jk}) = 0 \implies r_{jk} = 0 \implies \underline{d}_j \perp \underline{d}_k$
- If $\theta_{jk} \in \left(0, \frac{\pi}{2}\right)$ we have positive or negative correlation

Consider again the data set $\mathbb{X} = \begin{bmatrix} \underline{x}_1^T & \dots & \underline{x}_n^T \end{bmatrix}^T$ where each row $\underline{x}_i \in \mathbb{R}^p$ is a vector talking about a statistical unit. Assume now that $\underline{x}_i$ is the realisation of a random vector $\underline{X}_i$

Suppose than that $\underline{X}_1, \dots, \underline{X}_n \overset{iid}{\sim} \underline{X}$ This means that all the random vectors are independent and identically distributed like the random vector $\underline{X} \in \mathbb{R}^p$

Since $\underline{X} \in \mathbb{R}^p$ then we have that: $\underline{X} = (X_1, \dots, X_p)^T$, where each component X_i is a random variable in $\mathbb{R}$ We need a probability distribution (law) of random variable and we need to summarise it through: its mean, its variance and so on and so forth.

Definition: The Probability Distribution of $\underline{X}$ is defined as: $P_{\underline{X}} : \mathcal{B} \to [0, 1] = \mathbb{P}[\underline{X} \in B] \ \forall B \in \mathcal{B}$ where $\mathcal{B}$ is the Borel $\sigma-$algebra over $\mathbb{R}^p$ and $B \in \mathcal{B}$ is a Borel Set. Then:

- $P_{\underline{X}}(B) = \int_B f_{\underline{X}}(\underline{t}) d\underline{t}$ where $f_{\underline{X}}$ is the density of $\underline{X}$
- $\mathbb{E}[\underline{X}] = (\mathbb{E}[X_1], \dots, \mathbb{E}[X_p])^T = \underline{\mu} = (\mu_1, \dots, \mu_p)^T$ is the expected value (mean) of $\underline{X}$
- $\sigma_{jk} = \mathbb{E}[(X_j - \mu_j)(X_k - \mu_k)]$ for $j, k = 1, \dots, p$ and $\sigma_{jj} = \mathbb{E}[(X_j - \mu_j)^2] = Var(X_j)$

- $\Sigma = [\sigma_{jk}] \in R^{p\times p}$ is the co-variance matrix of $\underline{X}$ Moreover we have that: $\Sigma = \mathbb{E}[(\underline{X} - \underline{\mu})(\underline{X} - \underline{\mu})^T]$ Note that is a symmetric matrix.
- $V = diag(\sigma_{11}, \ldots, \sigma_{pp})$ where $\sigma_{ii} \geq 0$ are the variances. Thus: $V^{1/2} = diga(\sqrt{\sigma_{ii}})$ If: $\sigma_{ii} > 0 \; \forall i = 1, \ldots, p$ we can take the inverse: $V^{-1/2} = diag\left(\frac{1}{\sqrt{\sigma_{ii}}}\right)$
- We can now define the Correlation Matrix: $\rho = V^{-1/2}\Sigma V^{-1/2} \in \mathbb{R}^{p\times p}$ Note that is a symmetric matrix.

We now want to work with linear combinations of the components of a random vector: Given $\underline{c} \in \mathbb{R}^p$, we have that:

- $\underline{c}^T\underline{X} = c_1X_1 + \cdots + c_pX_p \in \mathbb{R}$ is a random variable
- $\mathbb{E}[\underline{c}^T\underline{X}] = c_1\mathbb{E}[X_1] + \cdots + c_p\mathbb{E}[X_p] = c_1\mu_1 + \cdots + c_p\mu_p = \underline{c}^T\underline{\mu}$
- $Var(c_1X_1 + c_2X_2) = c_1^2Var(X_1) + c_2^2Var(X_2) + 2c_1c_2Cov(X_1, X_2)$ indeed the variance is a quadratic operator. Moreover: $Var(\underline{c}^T\underline{X}) = \underline{c}^T\Sigma\underline{c}$

We now want to take many linear combinations: $C = (\underline{c}_1^T, \ldots, \underline{c}_k^T)^T$ is a $k \times p$ matrix of constant, with $\underline{c}_i \in \mathbb{R}^p$ Then:

- $C\underline{X} = \left[\underline{c}_1^T\underline{X} \quad \ldots \quad \underline{c}_k^T\underline{X}\right]^T$
- $\mathbb{E}[C\underline{X}] = C\underline{\mu}$
- $Cov(C\underline{X}) = C\Sigma C^T$

In the particular case in which: $C = \underline{c}^T = (c_1, \ldots, c_p) \in 1 \times p$ We have: $Cov(C\underline{X}) = Var(\underline{c}^T)\underline{X}) = C\Sigma C' = \underline{c}^T\Sigma\underline{c}$ which is the same formula we got above!

The data set is: $\mathbb{X}$ which is an $n \times p$ matrix, and the random vector $\underline{X}$ is the model that generated all the rows of the data set. Can we use the data to estimate some characteristics of the model (e.g: its mean and its variance)?

Note: The model is made up of the entire (infinite) population, whereas the data is a sample of size n! Inference: can we use the sample to say something about entire population?

An estimator for $\underline{\mu} = (\mu_1, \ldots, \mu_p)^T$ is given by the sample mean: $(\bar{x}_1, \ldots, \bar{x}_p)^T$ that is computed from the data set. Why is it an estimator? Well $\underline{\bar{x}} = (\bar{x}_1, \ldots, \bar{x}_p)^T$ is the realisation of another random entity which is $\overline{\underline{X}} = \frac{1}{n}\sum_{i=1}^{n}\underline{X}_i$ where $\underline{X}_i \overset{iid}{\sim} \underline{X}$ We don't have $\underline{X}_i$ but we only observe, in the data set, one of its realisations which is: $\underline{x}_i$

Note: $\overline{\underline{X}}$ is the **estimator** (i.e: it's an algorithm) and $\underline{\bar{x}}$ is the **estimate** (i.e: its the output of the algorithm). Thus: the **estimator** is a random variable: we want to know if an **estimator** is good and not if its **estimate** is good!

How good is the estimator $\bar{\underline{X}}$ for generating $\underline{\bar{x}}$? We have the following result:
Proposition: If $\underline{X}_1, \ldots, \underline{X}_n \overset{iid}{\sim} \underline{X}$ and if $\mathbb{E}[\underline{X}] = \underline{\mu}$ and $Cov(X) = \Sigma$ Then:

1) $\mathbb{E}[\bar{\underline{X}}] = \underline{\mu}$ so that the estimator is unbiased: on average it does the right thing. Note that looking at unbiased estimator is not sufficient!
2) $Cov(\bar{\underline{X}}) = \frac{1}{n}\Sigma$ is the variability of the estimator!

Proof of the Above Proposition:

$$\textbf{1)} \quad \mathbb{E}[\bar{\underline{X}}] = \mathbb{E}\left[\left[\frac{1}{n}\sum_{i=1}^{n}X_{i1} \quad \ldots \quad \frac{1}{n}\sum_{i=1}^{n}X_{ip}\right]^T\right] = (\mu_1, \ldots, \mu_p)^T$$

2) $Cov(\underline{\bar{X}}) = \mathbb{E}[(\underline{\bar{X}} - \underline{\mu})(\underline{\bar{X}} - \underline{\mu})^T] = \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n}(\underline{X}_i - \underline{\mu})\right)\left(\frac{1}{n}\sum_{i=1}^{n}(\underline{X}_i - \underline{\mu})\right)^T\right] =$

$$= \frac{1}{n^2}\mathbb{E}\left[\sum_{i=1,k=1}^{n}(\underline{X}_i - \underline{\mu})(\underline{X}_k - \underline{\mu})^T\right] = \frac{1}{n^2}\sum_{i=1,k=1}^{n}\mathbb{E}[(\underline{X}_i - \underline{\mu})(\underline{X}_k - \underline{\mu})^T] = \frac{1}{n^2}n\Sigma = \frac{1}{n}\Sigma$$

This last two passages are due to the fact that:

$$\underline{\bar{X}} = \frac{1}{n}\sum_{i=1}^{n}\underline{X}_i$$

And moreover:

$$\begin{cases} \text{if } i = k \text{ Then: } \mathbb{E}[(\underline{X}_i - \underline{\mu})(\underline{X}_i - \underline{\mu})^T] = \Sigma \\ \\ \text{if } i \neq k \text{ Then: } \mathbb{E}[(\underline{X}_i - \underline{\mu})(\underline{X}_k - \underline{\mu})^T] = \mathbb{E}[\underline{X}_i - \underline{\mu}] \cdot \mathbb{E}[(\underline{X}_k - \underline{\mu})^T] = 0 \cdot \mathbb{E}[(\underline{X}_k - \underline{\mu})^T] = 0 \text{ because } (\underline{X}_i - \underline{\mu}) \perp\!\!\!\perp (\underline{X}_k - \underline{\mu})^T \end{cases}$$

Note: $S = \frac{1}{n}\sum_{i=1}^{n}(\underline{X}_i - \underline{\bar{X}})(\underline{X}_i - \underline{\bar{X}})^T$, which is a $p \times p$ symmetric matrix, is an estimator of Σ It can be proved that: $\mathbb{E}[S] = \frac{n-1}{n}\Sigma$

Therefore to get an unbiased estimator of Σ we use: $\frac{1}{n-1}\sum_{i=1}^{n}(\underline{X}_i - \underline{\bar{X}})(\underline{X}_i - \underline{\bar{X}})^T$

Note that the realisation of S is the sample co-variance (co-variability) matrix: $[S_{ij}]$ with $i, j = 1, \ldots, p$

3 Lecture 6: 17th Of March 2020

Consider the data matrix $\mathbb{X} = \begin{bmatrix} \underline{x}_1^T & \dots & \underline{x}_p^T \end{bmatrix}^T = \begin{bmatrix} \underline{y}_1 & \dots & \underline{y}_p \end{bmatrix}$ with $\underline{x}_i \in \mathbb{R}^p$ which is the realisation of a random vector $\underline{X}_i$ with values in $\mathbb{R}^p$

We assume that $\underline{X}_1, \dots, \underline{X}_n \overset{iid}{\sim} \underline{X}$ with: $\mathbb{E}[\underline{X}] = \underline{\mu}$ and $\Sigma = Cov(\underline{X})$ Where $\Sigma, \underline{\mu}$ are unknown.

Proposition:

- $\bar{\underline{X}}$ is an unbiased estimator of $\underline{\mu}$
- $Cov(\bar{\underline{X}}) = \frac{1}{n}\Sigma$
- $\mathbb{E}[S] = \frac{n-1}{n}\Sigma$ so S is a biased estimator

Corollary: $\mathbb{E}\left[\frac{n}{n-1}S\right] = \Sigma$ so: $\frac{n}{n-1}S$ is an unbiased estimator for Σ

Definition: $\frac{n}{n-1}S = \frac{n}{n-1}\frac{1}{n}\sum_{i=1}^{n}(\underline{X}_i - \bar{\underline{X}})(\underline{X}_i - \bar{\underline{X}})^T = \frac{1}{n-1}\sum_{i=1}^{n}(\underline{X}_i - \bar{\underline{X}})(\underline{X}_i - \bar{\underline{X}})^T$

From now on we set: $S = \frac{1}{n-1}\sum_{i=1}^{n}(\underline{X}_i - \bar{\underline{X}})(\underline{X}_i - \bar{\underline{X}})^T$ Indeed we are not interested in the co-variance of the sample, but in the estimator for the co-variance of the population!

When we will need to refer to the co-variance of the sample we will use the symbol S_n and moreover when n is large the difference is negligible!

Consider now the Deviation Vectors: $\underline{d}_j = \underline{y}_j - \pi_{\underline{y}_j|\underline{1}} = \underline{y}_j - \frac{\underline{1}\underline{1}^T}{\underline{1}^T\underline{1}}\underline{y}_j = \left(I - \frac{\underline{1}\underline{1}^T}{\underline{1}^T\underline{1}}\right)\underline{y}_j$ where the multiplier of $\underline{y}_j$ is the orthogonal projector operator on $\mathcal{L}(\underline{1})^{\perp}$ and its a matrix.

$d = [\underline{d}_1 \dots \underline{d}_p]$ is the deviation $n \times p$ matrix. Moreover: $d = \left(I - \frac{\underline{1}\underline{1}^T}{\underline{1}^T\underline{1}}\right)\mathbb{X}$ Thus:

$$S = \begin{bmatrix} s_{11} & \dots & s_{1p} \\ \dots & \dots & \dots \\ s_{p1} & \dots & s_{pp} \end{bmatrix} = \frac{1}{n-1}\begin{bmatrix} \sum_i (X_{i1} - \bar{X}_1)^2 & \dots & \sum_i (X_{i1} - \bar{X}_1)(X_{ip} - \bar{X}_p) \\ \vdots & \vdots & \vdots \\ \sum_i (X_{ip} - \bar{X}_p)(X_{i1} - \bar{X}_1) & \dots & \sum_i (X_{ip} - \bar{X}_p)^2 \end{bmatrix} =$$

$$= \frac{1}{n-1}\begin{bmatrix} \underline{d}_1^T\underline{d}_1 & \underline{d}_1^T\underline{d}_2 & \dots & \underline{d}_1^T\underline{d}_p \\ \vdots & \vdots & \vdots & \vdots \\ \underline{d}_p^T\underline{d}_1 & \dots & \dots & \underline{d}_p^T\underline{d}_p \end{bmatrix} = \frac{1}{n-1}d^T d = \frac{1}{n-1}\mathbb{X}^T\left(I - \frac{\underline{1}\underline{1}^T}{\underline{1}^T\underline{1}}\right)^T\left(I - \frac{\underline{1}\underline{1}^T}{\underline{1}^T\underline{1}}\right)\mathbb{X} = \frac{1}{n-1}\mathbb{X}^T\left(I - \frac{\underline{1}\underline{1}^T}{\underline{1}^T\underline{1}}\right)\mathbb{X}$$

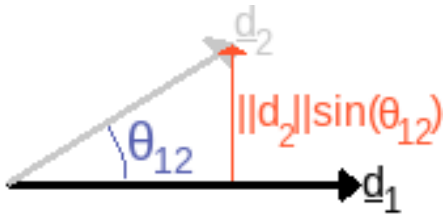
The last equality is due to the fact that $\left(I - \frac{\underline{1}\underline{1}^T}{\underline{1}^T\underline{1}}\right)$ is an orthogonal projection, and it's a self-adjoint (symmetric) and idem-potent operator!

Variability in a multivariate sense

Definition: The **generalised variance** is given by $det(S)$ and the **total variance** is given by $Tr(S)$

Why the above numbers are intuitive idea about variance in a multi-variate setting? Well consider the case in which $p = 2$ Then:

$$S = \frac{1}{n-1} d^T d = \frac{1}{n-1} \begin{bmatrix} \underline{d}_1^T \underline{d}_1 & \underline{d}_1^T \underline{d}_2 \\ \underline{d}_2^T \underline{d}_1 & \underline{d}_2^T \underline{d}_2 \end{bmatrix} = \frac{1}{n-1} \begin{bmatrix} \|\underline{d}_1\|^2 & \|\underline{d}_1\|\|\underline{d}_2\| cos(\theta_{12}) \\ \|\underline{d}_1\|\|\underline{d}_2\| cos(\theta_{12}) & \|\underline{d}_2\|^2 \end{bmatrix}$$



In this case the generalised variance is:

$$det(S) = \frac{1}{(n-1)^2}(\|\underline{d}_1\|^2\|\underline{d}_2\|^2 - \|\underline{d}_1\|^2\|\underline{d}_2\|^2 cos^2(\theta_{12})) = \frac{1}{(n-1)^2}\|\underline{d}_1\|^2\|\underline{d}_2\|^2 sin^2(\theta_{12})$$

which is proportional to the square area of the parallelogram generated by $\underline{d}_1$ and $\underline{d}_2$

We can see that as $det(S)$ increase, so does the area of the parallelogram: we can either increase in length of the sides (i.e: we increase the variability of the variables), or we can increase the angle (i.e: we increase the dependence among variable).

Thus if $det(S) = 0$ it means that there is one variable dependent on another variable, so that the information that variable is giving is useless, indeed if $det(S) = 0$ then either the angle is zero or the length of the sides is zero

In this case the total variance is: $Tr(S) = \frac{1}{n-1}(\|d_1\|^2 + \|d_2\|^2)$ which is proportional to the sum of the marginal variability!

Conclusion: in the case in which p is generic, we have that $det(S)$ is proportional to the volume squared of parallelotope generated by $\underline{d}_i$ and $Tr(S)$ is the sum of $\|d_i\|^2$

Proposition:
$det(S) = 0 \iff \underline{d}_1, \dots, \underline{d}_p$ are linearly dependent, that is: $\exists c \neq 0$ such that: $\underline{d} \cdot \underline{c} = 0 \implies c_1\underline{d}_1 + \dots + c_p\underline{d}_p = 0$

Poof of the Proposition Above:

$\Longleftarrow$ Assume that $\underline{d}_1, \dots, \underline{d}_p$ are linearly dependent, that is: $\exists \underline{c} \in \mathbb{R}^p$ such that: $\underline{d} \cdot \underline{c} = \underline{0}$ Then:
$S = \frac{1}{n-1} d^T d \implies S\underline{c} = \frac{1}{n-1} d^T d\underline{c} = 0$ because of the linear dependence. Therefore: The columns of S are linearly dependent, so that $det(S) = 0$

$\Longrightarrow$ Assume that $det(S) = 0$ then there must be $\underline{c} \neq 0$ such that; $S\underline{c} = \underline{0}$ Thus:
$\underline{c}^T S\underline{c} = 0 \implies \frac{1}{n-1}\underline{c}^T d^T d\underline{c} = 0 \implies \frac{1}{n-1}\|d\underline{c}\|^2 = 0 \implies d\underline{c} = 0$ But this means that $\underline{d}_1, \dots, \underline{d}_p$ are linearly dependent vectors!

Important: Thanks to the above proposition we have that if $det(S) = 0$ then $\underline{d}_1, \dots, \underline{d}_p$ are linearly dependent so $\exists \underline{c} \neq \underline{0} : c_1 d_1 + \dots + c_p d_p$ Then without loss of generality assume $c_1 \neq 0 \implies \underline{d}_1 = -\sum_{i=2}^{p} \frac{c_i}{c_1}\underline{d}_i \implies \underline{y}_1 = \bar{x}_1 \cdot \underline{1} - \sum_{i=2}^{p} \frac{c_i}{c_1}(\underline{y}_i - \bar{x}_i \underline{1})$

This means that there is no new information in the variable 1: once we know variables from 2 trough p we can perfectly predict the variable 1. Remember that if we don't have enough data (i.e: $n > p$) then $det(S) = 0$ and we will be

in this situation!

Proposition: $\mathbb{X}$ is an $n \times p$ matrix. If $p \geq n$ then we are guaranteed that $det(S) = 0$

Proof: $d = [\underline{d}_1, \ldots, \underline{d}_p]$ where $\underline{d}_i \in \mathbb{R}^n$ But the deviation vectors aren't free to range over the entire space since: $\underline{d}_i \in \mathcal{L}^{\perp}(\underline{1})$

We know that: $dim(\mathcal{L}^{\perp}(\underline{1})) = n - 1$ so we can have at most $n - 1$ (i.e: degrees of freedom) linearly independent vectors: if $p \geq n$ we are considering p vectors in a space of dimension $n - 1$, so necessarily we have that $\underline{d}_1, \ldots, \underline{d}_p$ are linearly dependent and therefore $det(S) = 0$

Therefore a necessary condition for having $det(S) > 0$ is that $n \geq p + 1$

Since S is a $p \times p$ symmetric matrix. and $s_{ij} \in \mathbb{R}$ then the Spectral Decomposition Theorem holds and thus: $\exists \lambda_1, \ldots, \lambda_p \in \mathbb{R}$ and $\underline{e}_1, \ldots, \underline{e}_p \in \mathbb{R}^p$ such that $\underline{e}_i$ form an ortho-normal system of $\mathbb{R}^p$, that is: $\underline{e}_i^T \underline{e}_j = \begin{cases} 0 \text{ if } i \neq j \\ 1 \text{ Otherwise} \end{cases}$

So: $S = \sum_{i=1}^{p} \lambda_i \underline{e}_i \underline{e}_i^T$ where $\underline{e}_i \underline{e}_i^T$ is the projection on the linear space spanned by $\underline{e}_i$ Moreover: $(\lambda_i, \underline{e}_i)$ is the couple of eigenvalue and eigen-vector, so: $S_i \underline{e} = \lambda_i \underline{e}_i$ for $i = 1, \ldots, p$

Setting now $P = [\underline{e}_1, \ldots, \underline{e}_p]$ and $\Lambda = diag(\lambda_1, \ldots, \lambda_p)$ then: $S = P \Lambda P^T$ Moreover: $det(S) = \prod_{i=1}^{p} \lambda_i$ is the generalised variance and $tr(S) = \sum_{i=1}^{p} \lambda_i$ is the total variance.

Proposition: S is positive semi-definite. Moreover: If $det(S) \neq 0 \implies S$ is positive definite.

Proof of the Proposition Above: We need to prove that $\underline{c}^T S \underline{c} \geq 0 \; \forall \underline{c} \in \mathbb{R}^p$ then:

$$\underline{c}^T S \underline{c} = \frac{1}{n-1} \underline{c}^T d' d \underline{c} = \frac{1}{n-1} \|d\underline{c}\|^2 \geq 0$$

Suppose now that $\exists \underline{c} \neq \underline{0}$ such that: $\underline{c}^T S \underline{c} = 0$ then: $\|d\underline{c}\| = 0 \implies d\underline{c} = \underline{0} \implies \underline{d}_1, \ldots, \underline{d}_p$ are linearly dependent, therefore: $det(S) = 0$

So if $det(S) \neq 0$ then such c above cannot exist therefore S is positive definite (i.e: $det(S) > 0$)

Notation: $S = \sum_{i=1}^{p} \lambda_i \underline{e}_i \underline{e}_i^T$ From now on we will conventionally assume that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$

Assume that $det(S) \neq 0$ then $\lambda_i > 0 \; \forall i$ then: $S = \sum_{i=1}^{p} \lambda_i \underline{e}_i \underline{e}_i^T$ We can find its inverse: $S^{-1} = \sum_{i=1}^{p} \frac{1}{\lambda_i} \underline{e}_i \underline{e}_i^T$

The matrix S^{-1} induces a metric on $\mathbb{R}^p$ which is the right metric when we do statistics: if $\underline{x}, \underline{y} \in \mathbb{R}^p$ the distance induced by S is given by: $d^2_{S^{-1}}(\underline{x}, \underline{y}) = (\underline{x} - \underline{y})^T S^{-1} (\underline{x} - \underline{y})$ This is a distance in $\mathbb{R}^p$ and its called **Mahalanobis Distance**.

Which neighbourhoods are generated by the **Mahalanobis Distance**?

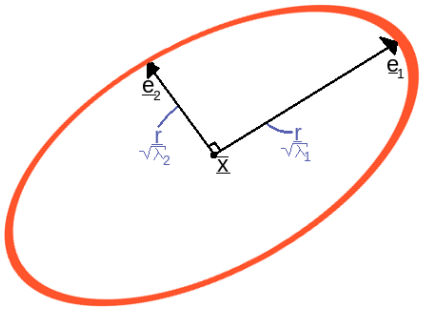
$$\mathcal{E}_{r^2, S^{-1}}(\bar{\underline{x}}) = \{\underline{x} \in \mathbb{R}^p : d_{S^{-1}}(\underline{x}, \bar{\underline{x}})^2 \leq r^2\}$$

Therefore the distance induced by S^{-1} (i.e: the **Mahalanobis Distance**) generates neighbourhoods that are circles in $\mathbb{R}^p$ given that we measure distance between points with the **Mahalanobis Distance**. If we go back to euclidean geometry we

have that:

$$\mathcal{E}_{r^2,S^{-1}}(\underline{\bar{x}}) = \{\underline{x} \in \mathbb{R}^p : d_{S^{-1}}(\underline{x}, \underline{\bar{x}})^2 \leq r^2\} = \{\underline{x} \in \mathbb{R}^p : (\underline{x} - \underline{\bar{x}})^T S^{-1} (\underline{x} - \underline{\bar{x}}) \leq r^2\}$$

Thus: this is the set of points inside an ellipse centred on the baricentre of the data $\underline{\bar{x}}$, and the axis of the ellipse are given by the eigen-vectors of S^{-1} and since: $S^{-1} = \sum_{i=1}^{p} \frac{1}{\lambda_i} \underline{e}_i \underline{e}^T$ then the first axis-length is proportional to $r \frac{1}{\sqrt{\frac{1}{\lambda_1}}} = r\sqrt{\lambda_1}$ and the other axis is proportional to $r \frac{1}{\sqrt{\frac{1}{\lambda_2}}} = r\sqrt{r_2}$ since the ellipse is induced by the quadratic form S^{-1} with eigenvalues $\frac{1}{\lambda i}$

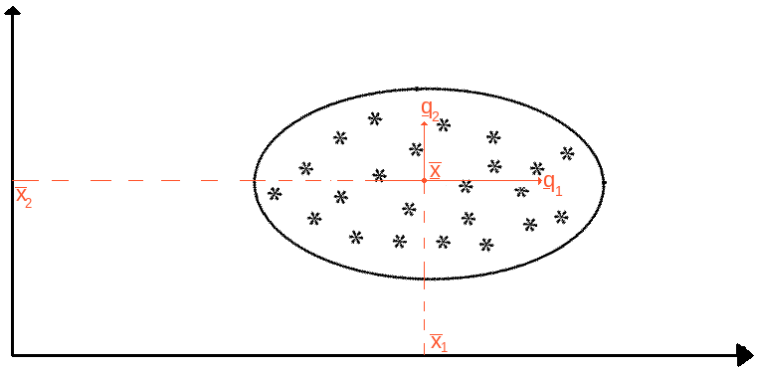


Moreover: $\text{Volume}(\mathcal{E}_{r^2,S^{-1}})(\underline{\bar{x}}) = k_p r^p \sqrt{\prod_{i=1}^{p} \lambda_i} = k_p r^p \sqrt{det(S)}$ where k_p is a constant which depends on p.

Note that if $p = 2$ then $k_p = \pi$

Therefore we can see from the expression of the volume above, the larger the generalised variance the larger the neighbourhood around the mean!

Example: Let $p = 2$ then: $S = diag(s_{11}, s_{22})$ with $s_{11} > s_{22}$ There is no correlation between x_1, x_2



Note: The data cloud above has that shape because the variance along first variable is bigger!

Consider now: $\underline{q}_1 = (q_1, \bar{x}_2)^T$ and $\underline{q}_2 = (\bar{x}_1, q_2)^T$ Suppose they are at the same distance form baricentre then: $d_e(\underline{q}_1, \underline{\bar{x}}) = d_e(\underline{q}_2, \underline{\bar{x}})$ where the subscript e means that we are considering the euclidean distance.

But is it really the case? First of all let's standardise the data, since the variances aren't the same. Then: $d_e(std(\underline{q}_1), std(\bar{x})) = \frac{|q_1 - \bar{x}_1|}{\sqrt{s_{11}}}$ and $d_e(std(\underline{q}_2), std(\underline{\bar{x}})) = \frac{|q_2 - \bar{x}_2|}{\sqrt{s_{22}}}$ But since we supposed that $s_{11} > s_{22}$ then the first distance is smaller than the second distance!

If $\underline{q} = (q_1, q_2)^T$ then: $d_e(std(\underline{q}), std(\bar{\underline{x}}) = \sqrt{\frac{(q_1 - \bar{x}_1)^2}{s_{11}} + \frac{(q_2 - \bar{x}_2)^2}{s_{22}}} = \sqrt{(\underline{q} - \bar{\underline{x}})^T S^{-1} (\underline{q} - \bar{\underline{x}})}$

Conclusion: The **Mahalanobis Distance** is the right one because it considers the variability of the data in calculating the distance between the units!

What happens if we have a more general form of S? The above formula still works!

4 Lecture 7: 19th Of March 2020

Let $p = 2$, suppose that $S = diag(s_{11}, s_{22})$ and consider the neighbourhoods generated by the *Mahalanobis Distance:* $\mathcal{E}_r(\underline{\bar{x}}) = \{\underline{x} \in \mathbb{R}^p : (\underline{x} - \underline{\bar{x}})^T S^{-1}(\underline{x} - \underline{\bar{x}}) \leq r^2\}$ Then we have that: $\text{Area}(\mathcal{E}_r(\underline{\bar{x}})) \propto r^2\sqrt{det(S)} = r^2\sqrt{s_{11}s_{22}}$

We can see that if either $det(S)$, or $tr(S)$, increase then $\text{Area}(\mathcal{E}_r(\underline{\bar{x}}))$ increases, even if we are not modifying the radius r!

Does this happen because $p = 2$ and the matrix S is diagonal? No! Indeed, let $p \geq 1$ and consider any S, with $det(S) > 0$ so that we can invert it and we can define the *Mahalanobis Distance* Then: we can always find a reference system with respect to which the matrix S is diagonal, indeed:
By the Spectral Decomposition Theorem we have that: $S = \sum_{i=1}^{p} \lambda_i \underline{e}_i \underline{e}_i$ with the convention that λ_1 is biggest eigenvalue.
Thus: $S = P\Lambda P^T$ Therefore in the new reference system made up of the eigen-vectors (which are ortho-normal) of S we have that S is diagonal!

Suppose the original system has axis given by: x_i then $\underline{w} = (x_1, \ldots, x_p)^T$ The new system has axis given by: $\underline{e}_i$ then: $\underline{\tilde{w}} = (\underline{e}_1^T \underline{w}, \ldots, \underline{e}_p^T \underline{w})^T = P^T \underline{w}$

Of course: $\|\underline{w}\| = \|\underline{\tilde{w}}\|$ indeed P is just a change of base matrix (it's orthogonal matrix). Thus:

$$\|\underline{w}\|^2 = \underline{w}^T \underline{w} = \underline{\tilde{w}}^T \underline{\tilde{w}} = \underline{w}^T P P^T \underline{w}$$

Thus:

- In the original system we have that:
  - $\mathbb{X} = (\underline{x}_1^T, \ldots, \underline{x}_n^T)^T$
  - $S = \frac{1}{n-1}\mathbb{X}^T \left(I - \frac{\underline{1}\underline{1}^T}{\underline{1}^T\underline{1}}\right)\mathbb{X}$
- In the new system we have that:
  - $\tilde{\mathbb{X}} = ((P^T\underline{x}_1)^T, \ldots, (P^T\underline{x}_n)^T)^T = (\underline{x}_1^T P, \ldots, \underline{x}_n^T P)^T = \mathbb{X}P$
  - $\tilde{S} = \frac{1}{n-1}\tilde{\mathbb{X}}^T \left(I - \frac{\underline{1}\underline{1}^T}{\underline{1}^T\underline{1}}\right)\tilde{\mathbb{X}} = \frac{1}{n-1}P^T\mathbb{X}^T \left(I - \frac{\underline{1}\underline{1}^T}{\underline{1}^T\underline{1}}\right)\mathbb{X}P = P^T S P = P^T P \Lambda P^T P = \Lambda$

    Note that $P^T P = I$ since P is an orthogonal matrix, whose columns are the eigen-vectors!

We can see that in the new system the co-varaince matrix is Λ which is the diagonal matrix made up of the eigenvalues of the original co-variance matrix!

Take home message: there is always a reference system for which the coordinates are uncorrelated and so correlation is not a property of the data, but of the data looked at with particular glasses given by the reference system!

Indeed the generalised variance in the original system is $det(S)$ and the generalised variance in new system is $det(\tilde{S}) = det(\Lambda)$ So they are the same, indeed the determinant of the a matrix is always the product of its eigenvalues! Moreover the same holds for the total variance!

We can see thus that the properties of the data set are indeed given by the total variance and the generalised variance, since they are invariant with respect to the reference system!

Let $\underline{X}$ be a random vector in $\mathbb{R}^p$ and suppose we have data that is iid realisation of this vector. Suppose $\mathbb{E}[\underline{X}] = \underline{\mu}$ and $Cov[\underline{X}] = \Sigma$

Let $\underline{a} \in \mathbb{R}^p$ then $\underline{a}^T\underline{X} = a_1X_1 + \cdots + a_pX_p$ We want to find the $\underline{a}$ such that $Var[\underline{a}^T\underline{X}]$ is maximum. Is this problem well posed? No:
For example suppose the maximum variability is reached by $\underline{a}^T\underline{X} = 10X_1$ Now consider: $10\underline{a}$ then:
$Var(10\underline{a}^T\underline{X}) = 100Var(\underline{a}^T\underline{X})$ We get something even larger so $\underline{a}^TX$ couldn't possibly be the maximum!

A well posed problem is the following: find $\underline{a}$ such that $\|\underline{a}\| = 1$ and $Var(\underline{a}^TX)$ is maximum. Then:

$$\max_{\underline{a}\in\mathbb{R}^p:\|\underline{a}\|=1} Var(\underline{a}^T\underline{X}) = \max_{\underline{a}\in\mathbb{R}^p:\|\underline{a}\|=1} \underline{a}^T\Sigma\underline{a} = \max_{\underline{a}\in\mathbb{R}^p} \frac{\underline{a}^T\Sigma\underline{a}}{\underline{a}^T\underline{a}}$$

Lemma: Let B be a $p \times p$ positive semi-definite matrix and assume that $B = \sum_{i=1}^{p} \lambda_i \underline{e}_i\underline{e}_i^T$ is its spectral decomposition. Then:

1) $\max_{\underline{x}\in\mathbb{R}^p} \frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \lambda_1$ so that: $\arg\max_{\underline{x}\in\mathbb{R}^p} \frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \underline{e}_1$

2) $\max_{\underline{x}\in\mathbb{R}^p:\underline{x}\perp\underline{e}_1} \frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \lambda_2$ so that: $\arg\max_{\underline{x}\in\mathbb{R}:\underline{x}\perp\underline{e}_1} \frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \underline{e}_2$

i) ...

p) $\max_{\underline{x}\in\mathbb{R}^p:\underline{x}\perp\underline{e}_1,\ldots,\underline{x}\perp\underline{e}_{p-1}} \frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \lambda_p$ so that: $\arg\max_{\underline{x}\in\mathbb{R}^p:\underline{x}\perp\underline{e}_1,\ldots,\underline{x}\perp\underline{e}_{p-1}} \frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \underline{e}_p$

Proof of the Above Lemma: $B = \sum_{i=1}^{p} \lambda_i\underline{e}_i\underline{e}_i^T = P\Lambda P^T$

Consider a generic $\underline{x} \in \mathbb{R}^p$ then: $\frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \frac{\underline{x}^TP\Lambda P^T\underline{x}}{\underline{x}^TPP^T\underline{x}} \underset{\underline{y}=P^T\underline{x}}{=} \frac{\underline{y}^T\Lambda\underline{y}}{\underline{y}^T\underline{y}} = \sum_{i=1}^{p} \lambda_i\underline{y}_i^2 \frac{1}{\sum_{i=1}^{p}\underline{y}_i^2} \leq \lambda_1$ since λ_1 is the maximum eigenvalue!

Now consider a special $\underline{x}$ namely: $\underline{x} = \underline{e}_1$ then: $\frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \frac{\underline{e}_1^TB\underline{e}_1}{\underline{e}_1^T\underline{e}_1} = \underline{e}_1^TB\underline{e}_1 = \underline{e}_1^T\lambda_1\underline{e}_1 = \lambda_1$

But λ_1 is the maximum eigenvalue and thus the above expression is always less than or equal to λ_1 thus:

$$\max_{\underline{x}\in\mathbb{R}^p} \frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \lambda_1 \quad \text{so that:} \quad \arg\max_{\underline{x}\in\mathbb{R}^p} \frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \underline{e}_1$$

Now we need to prove **2)**
$\frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \frac{\underline{x}^TP\Lambda P^T\underline{x}}{\underline{x}^TPP^T\underline{x}} = \star$ We need to impose that: $\underline{x} \perp \underline{e}_1 \implies \underline{y} = P^T\underline{x} = (\underline{e}_1^T, \ldots, \underline{e}_p^T)^T\underline{x} = (0, \underline{e}_2^T\underline{x}, \ldots, \underline{e}_p^T\underline{x},)^T$

Thus: $\star = \frac{\underline{y}^T\Lambda\underline{y}}{\underline{y}^T\underline{y}} = \sum_{i=2}^{p} \lambda_i\underline{y}_i^2 \left(\sum_{i=2}^{p}\underline{y}_i^2\right)^{-1} \leq \lambda_2$ The sum starts from $i = 2$ since the first component of $\underline{y}$ is zero, since $\underline{x} \perp \underline{e}_1$

Therefore: $\frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} \leq \lambda_2$ if $\underline{x} \perp \underline{e}_1$ Now if $\underline{x} = \underline{e}_2$ then we reach λ_2 so that:

$$\max_{\underline{x}\in\mathbb{R}^p:\underline{x}\perp\underline{e}_1} \frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \lambda_2 \text{ so that: } \arg\max_{\underline{x}\in\mathbb{R}:\underline{x}\perp\underline{e}_1} \frac{\underline{x}^TB\underline{x}}{\underline{x}^T\underline{x}} = \underline{e}_2$$

We keep on proceeding this way, until we find are at the point **p)**

$$\frac{\underline{x}^T B\underline{x}}{\underline{x}^T\underline{x}} = \frac{\underline{y}^T \Lambda \underline{y}}{\underline{y}^T\underline{y}} = \sum_{i=1}^{p} \lambda_i y_i^2 \left(\sum_{i=1}^{p} y_i^2\right)^{-1} \geq \lambda_p \implies \lambda_p \leq \frac{\underline{x}^T B\underline{x}}{\underline{x}^T\underline{x}} \leq \lambda_1 \; \forall \underline{x} \in \mathbb{R}^p$$

Now if we take: $\underline{x} = \underline{e}_p$ we get that $\frac{\underline{x}^T B\underline{x}}{\underline{x}^T\underline{x}} = \lambda_p$ so the last eigen-vector finds the minimum! And this concludes the proof!

We can now use the above **Lemma:**

$$\max_{\underline{a}\in\mathbb{R}^p:\|\underline{a}\|=1} Var(\underline{a}^T\underline{X}) = \max_{\underline{a}\in\mathbb{R}^p} \frac{\underline{a}^T\Sigma\underline{a}}{\underline{a}^T\underline{a}} = \star$$

From the above **Lemma**, since Σ is positive semi-definite we have that: $\star = \lambda_1$ and $\arg \max_{\underline{a}\in\mathbb{R}^p:\|\underline{a}\|=1} Var(\underline{a}^T\underline{X}) = \underline{e}_1$ given that $\Sigma = \sum_{i=1}^{p} \lambda_i \underline{e}_i \underline{e}_i^T$ is the spectral decomposition of Σ

Definition: $y_1 = \underline{e}_1^T\underline{X}$ is called **first principal component** (PC1), and along this direction we have maximal variability.

Note: sometimes we centre on the baricentre (mean) $y_1 = \underline{e}_1^T(\underline{X} - \underline{\mu})$ For the co-variance nothing changes!

The second principal component must be uncorrelated with the first principal component, thus we have the following problem:

$$\max_{\underline{a}\in\mathbb{R}^p:\|\underline{a}\|=1 \text{ and } Cov(\underline{a}^T\underline{X},\underline{e}_1^T\underline{X})=0} Var(\underline{a}^T\underline{X}) = \max_{\underline{a}\in\mathbb{R}^p: Cov(\underline{a}^T\underline{X},\underline{e}_1^T\underline{X})=0} \frac{\underline{a}^T\Sigma\underline{a}}{\underline{a}^T\underline{a}} = \star\star\star$$

Now suppose that $C = (\underline{a}^T, \underline{b}^T)^T$ is a $2 \times p$ matrix. Then:

$$Cov(C\underline{X}) = C\Sigma C^T = \begin{bmatrix} \underline{a}^T\Sigma\underline{a} & \underline{a}^T\Sigma\underline{b} \\ \underline{b}^T\Sigma\underline{a} & \underline{b}^T\Sigma\underline{b} \end{bmatrix}$$

If C was a $k \times p$ matrix then we would $Cov(C\underline{X}) = C\Sigma C^T$ thus in this case we have that: $Cov(\underline{a}^T\underline{X}, \underline{b}^T\underline{X})\underline{a}^T\Sigma\underline{b} \implies Cov(\underline{a}^T\underline{X}, \underline{e}_1^T\underline{X}) = \underline{a}^T\Sigma\underline{e}_1 = \lambda_1\underline{a}^T\underline{e}_1 = 0 \iff \underline{a} \perp \underline{e}_1$

Therefore:

$$\star\star\star = \max_{\underline{a}\in\mathbb{R}^p:\underline{a}\perp\underline{e}_1} \frac{\underline{a}^T\Sigma\underline{a}}{\underline{a}^T\underline{a}} = \lambda_2 \text{ and } \arg\max_{\underline{a}\in\mathbb{R}^p:\underline{a}\perp\underline{e}_1} \frac{\underline{a}^T\Sigma\underline{a}}{\underline{a}^T\underline{a}} = e_2$$

Therefore the second principal component is given by: $y_2 = \underline{e}_2^T\underline{X}$

Note: sometimes we centre on the baricentre (mean) $y_2 = \underline{e}_2^T(\underline{X} - \underline{\mu})$ For the co-variance nothing changes!

Conclusion:

$$\max_{\underline{a}\in\mathbb{R}^p:\|\underline{a}\|=1, Cov(\underline{a}^T\underline{X},\underline{e}_i^T\underline{X})=0 \;\; i=1,\ldots j-1} Var(\underline{a}^T\underline{X}) = \lambda_j \text{ and } \arg \max_{\underline{a}\in\mathbb{R}^p:\|\underline{a}\|=1, Cov(\underline{a}^T\underline{X},\underline{e}_i^T\underline{X})=0 \;\; i=1,\ldots j-1} Var(\underline{a}^T\underline{X}) = e_j$$

We define the $j-$th principal component as: $y_j = \underline{e}_j^T\underline{X}$

Note: sometimes we centre on the baricentre (mean) $y_j = \underline{e}_j^T(\underline{X} - \underline{\mu})$ For the co-variance nothing changes!

Therefore: $\underline{Y} = (y_1, \ldots, y_p)^T$ is the vector of principal components and it is given by: $\underline{y} = P^T\underline{X}$

Proposition:

- $\mathbb{E}[\underline{Y}] = \mathbb{E}[P^T\underline{X}] = P^T\mathbb{E}[\underline{X}] = P^T\underline{\mu}$ Therefore If $\underline{Y} = P^T(\underline{X} - \underline{\mu})$ then: $\mathbb{E}[\underline{Y}] = 0$
- $Cov[\underline{Y}] = Cov(P^T\underline{X}) = P^T\Sigma P = P^T P\Lambda P^T P = \Lambda$ Therefore:

$$Cov(y_i, y_j) = \begin{cases} 0 \text{ if } i \neq j \\ \lambda_i = Var(y_i) \text{ if } i = j \end{cases} \qquad \forall i, j = 1, \ldots, p$$

  This means that the principal components are all uncorrelated!

We have an order among the principal components: the first is the one with largest variability (since its related from the biggest eigenvalue), the second one is the one with the second largest variability, and so on and so forth.

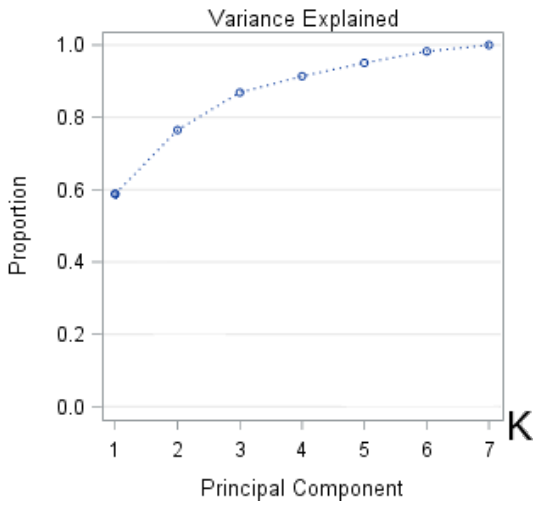
PCA provides us with a nice tool for dimensionality reduction: we can capture most of the total variability with the first few principal components, so we can forget about the last few components, and project our data set in the space generated by the more meaningful principal components! This provides a reduction of the dimensionality without losing (too much) information about the variability!

Since the principal components identify the dimension along which there is maximum variability, we can rank individuals along the maximum variability!

Observation: we don't lose any variability by looking at the data with the glasses given by the eigen-vectors! Indeed:

- The generalised variance of $\underline{Y}$ is: $det(\Lambda) = \prod_{i=1}^{p} \lambda_i = det(\Sigma)$ which is the generalised variance of $\underline{X}$ Thus we lose nothing in terms of generalised variance!
- The total variance of $\underline{Y}$ is: $tr(\Lambda) = \sum_{i=1}^{p} \lambda_i = tr(\Sigma)$ which is the total varianc of $\underline{X}$ Thus we lose nothing in terms of total variance!

Note that $Var(y_1) = \lambda_1$ so y_1 is capturing (i.e: explaining) $\lambda_1 \left(\sum_{i=1}^{p} \lambda_i\right)^{-1}$ proportion of total variability! So with y_1, y_2 we capture: $(\lambda_1 + \lambda_2)\left(\sum_{i=1}^{p} \lambda_i\right)^{-1}$ of the total variability.



In the figure above k is the number of principal components (PC) considered. Note that this is called *scree plot*.

Most of the time the above curve is more dramatic: there is an elbow, so when we find such a curve we stop at the elbow and consider those principal components to represent our data.

Indeed after an elbow we just have a small marginal gain, that keeps on getting smaller!

For example in the figure above we could either pick $k = 2, 3$ so that we would then project our data set into the space spanned by the first two (or three) principal components!

We can either stop at an elbow, or when can take as many principal components are needed to explain so much percentage (that we need to choose) of the total variability!

Many times the meaning of a principal components it's hard to explain indeed it's the linear combination of many variables. It doesn't have to make sense to us, since we moved to a very weird reference system, but it makes sense to the data!

5 Lecture 9: 23rd Of March 2020

Recall that $y_i = \underline{e}_i^T \underline{X}$ is the $i-$th principal component. Then: $y_i = e_{1i}X_1 + e_{2i}X_2 + \cdots + e_{pi}X_p$ The components of the eigen-vectors in this linear combination are called **loading (or weights)**

The main problem of PCA is interpreting the meaning of the above weighted average: sometimes we won't find a meaning, but if we do we get a lot of insight on the phenomenon.

We need to look at the correlation between X_k and Y_i to find some meaning!

Proposition: $Corr(Y_i, X_k) = e_{ki}\frac{\sqrt{\lambda_i}}{\sqrt{\sigma_{kk}}}$ for $i, k = 1, \ldots, p$ So if we fix the variances, what is meaningful is the loading e_{ki}

Proof: $Corr(Y_i, X_k) = Cov(Y_i, X_k)\frac{1}{\sqrt{\lambda_i \sigma_{kk}}}$ But:

$Cov(Y_i, X_k) = Cov(\underline{e}_i^T \underline{X}, \underline{u}_k^T \underline{X}) = \underline{e}_i^T \Sigma \underline{u}_k = u_k' \Sigma e_i = \lambda_i u_k' e_i = \lambda_i e_{ki}$ where $\underline{u}_k = (0, \ldots, 0, \underbrace{1}_{k-th \text{ position}}, 0, \ldots, 0)^T$

Then: $Corr(Y_i, X_k) = \lambda_i e_{ki}\frac{1}{\sqrt{\lambda_i \sigma_{kk}}} = \sqrt{\lambda_i}\frac{1}{\sqrt{\sigma_{kk}}}e_{ki}$ where e_{ki} is the $k-$th component of the $i-$th eigen-vector. Thus we have thesis!

Note: To perform PCA, given $\underline{X}$ we standardise: $\underline{Z} = V^{-1/2}(\underline{X} - \underline{\mu}) = \left(\frac{x_1 - \mu_1}{\sqrt{\sigma_{11}}}, \ldots, \right)^T$ where $V = diag(\sigma_{ii})$ and $\Sigma = [\sigma_{ij}]$ then: $\mathbb{E}[\underline{Z}] = 0$ and $Cov(\underline{Z}) = V^{-1/2}\Sigma V^{-1/2} = \rho$ which is the correlation matrix (i.e: co-variance matrix of standardised variables), which made of pure numbers!

Now that we got rid of the variability due to the different type of unit of measure, we can compute the PCA, using the right unit of measure: the standard deviation!

Note: $\rho = \sum_{i=1}^{p} \lambda_i \underline{e}_i \underline{e}_i^T$ where $\lambda_i, \underline{e}_i$ are the eigenvalues and eigen-vectors of the correlation matrix! Well these are not simply related to the ones of the co-variance matrix: there is no general rule to relate them.

So we standardise data $\underline{X}$ and we get $\underline{Z}$, now we apply PCA: $Y_i = \underline{e}_i^T \underline{Z} = \underline{e}_i^T V^{-1/2}(\underline{X} - \underline{\mu})$ note that:

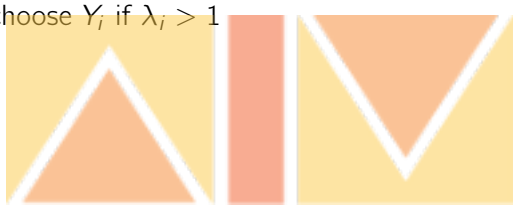
1) $\sum_{i=1}^{p} Var(Y_i) = Tr(\rho) = p = \sum_{i=1}^{p} Var(z_i)$ since the correlation matrix ρ has all ones along the main diagonal.

2) $Corr(Y_i, Z_k) = e_{ki}\frac{\sqrt{\lambda_i}}{Var(z_k)} = e_{ki}\sqrt{\lambda_i}$ where e_{ki} its the $k-$th component of the $i-$th eigen-vector of the correlation matrix.

3) Proportion of total variability explained by the first k components is given by: $\frac{1}{p}\sum_{i=1}^{k} \lambda_i$

4) The average value for the eigenvalues of the correlation matrix is: $\bar{\lambda} = \frac{1}{p}\sum_{i=1}^{p} \lambda_i = Tr(\rho)\frac{1}{p} = 1$

   This motivate one rule of thumb for selecting principal components: we choose those associated to eigenvalues that are greater than 1, so that they are greater than the average! So we choose Y_i if $\lambda_i > 1$



Example: Suppose that $\Sigma = \begin{bmatrix} 1 & 4 \\ 4 & 100 \end{bmatrix}$ is the co-variance matrix of $\underline{X} = (X_1, X_2)^T$ Then: $\rho = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix}$
Note that $100 >> 1$: this could be because of many different reasons, for instance if x_1 is measured in centimetres, and x_2 in millimetres, even if the range of variability is the same, the variance of x_2 is so much higher simply because of the unit of measure.

In this situation there is a clear problem: in many problems maybe this is not so easy to spot and we cant even change unit of measure!

PCA of Σ vs PCA of ρ:

- PCA of Σ
  The eigenvalues and eigen-vectors are: $\lambda_1 = 100.16, \underline{e}_1 = (0.004, 0.999)^T$ and $\lambda_2 = 0.84, \underline{e}_2 = (0.999, -0.04)^T$
  Then:
  $Y_1 = 0.04X_1 + 0.99X_2 \approx X_2$ and $Y_2 = 0.999X_1 - 0.04X_2 \approx X_1$ We can see that the variability of X_2 is so big that the direction of maximum variability is the one identified by itself! This is non-sense: we could already tell by looking at the co-variance matrix!
- PCA of ρ:
  The eigenvalues and eigen-vectors are: $\lambda_1 = 1.4, \underline{e}_1 = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)^T$ and $\lambda_2 = 0.6, \underline{e}_2 = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)^T$ Then:
  $Y_1 = \frac{1}{\sqrt{2}}(Z_1 + Z_2) = \frac{1}{\sqrt{2}}\left(\frac{X_1 - \mu_1}{\sqrt{\sigma_{11}}} + \frac{X_2 - \mu_2}{\sqrt{\sigma_{22}}}\right)$ and $Y_2 = \frac{1}{\sqrt{2}}(Z_1 - Z_2) = \frac{1}{\sqrt{2}}\left(\frac{X_1 - \mu_1}{\sqrt{\sigma_{11}}} - \frac{X_2 - \mu_2}{\sqrt{\sigma_{22}}}\right)$

PCA on Σ is different from the PCA on ρ!

Thus if some variable has high variability compared to the others, a good idea is to standardise the variables! But it's a lot more difficult to explain a linear combination of standardised variables!

Usually μ and Σ are unknown, but we have data so we can estimate them: $\mathbb{X} = [x_1^T, \dots, x_n^T]^T$ where x_i is the realisation of X_i and $X_i \overset{iid}{\sim} X$ Thus: Σ is estimated by S, μ by $\bar{X}$ so that we perform PCA on S!

$S = \sum_{i=1}^{p} \lambda_i \underline{e}_i \underline{e}_i^T$ then the principal components, y_i, are the projection of the data set on e_i Thus: $y_i = (\underline{e}_1^T x_i, \dots, \underline{e}_p^T x_i)^T$
these new coordinates are called *scores:* how much does the first component score on the first principal component?

Thus: we start with data $\mathbb{X} = (x_1^T, \dots, x_n^T)^T$, then through PCA we represent the same data with:
$(y_1^T, \dots, y_n^T)^T = \begin{bmatrix} y_{11} & \dots & y_{1p} \\ \vdots & \vdots & \vdots \\ y_{n1} & \dots & y_{np} \end{bmatrix}$ where the $i-$th column identifies the scores on the $i-$th principal component.
Suppose now that $\left(\sum_{i=1}^{p} \lambda_i\right)^{-1} \sum_{i=1}^{k} \lambda_i \geq 0.8$ the first k components capture 80% of the total variability: by getting rid of the last $p - k$ components, we just miss 20% of the variability of the data.

Note that there is no theorem for selection the threshold: it depends on the person and on the problem! We may want to select only 2 or 3 components so that we can plot the data that would otherwise be high dimensional!

Note: we can't perform PCA as it is on categorical variable, but there is an equivalent of PCA celled *correspondence analysis*, performed on the table of joint frequencies (contingency tables).

The problem using standard PCA on categorical variables is that if we replace the labels with numbers though, it's not always meaningful! It can be done with ordinal variables with caution, but it it may strongly depend on how we do the

transformation!

For example the difference between elementary schools and middle schools it ain't the same difference between a Master and a phD!

We can perform PCA if we have quantitative discrete variables!

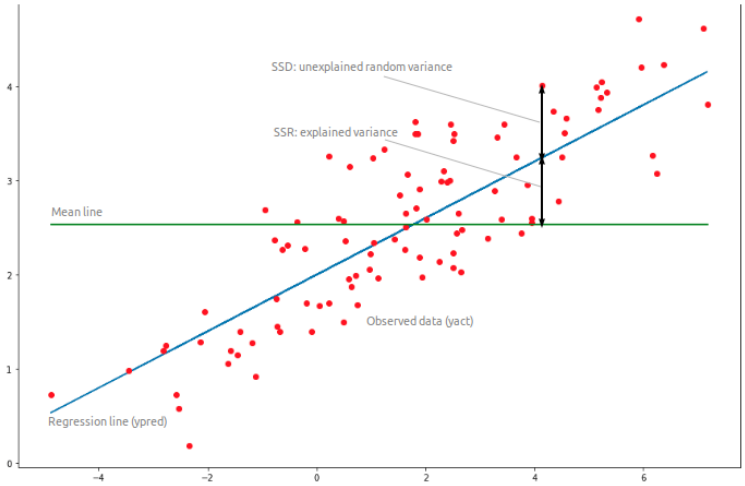
Note: if the smallest eigenvalue is very close to zero it means that there is a linear sub-space which contains all the variables: the determinant is almost equal to zero. The smallest eigenvalue tells us: either the sample size is too small, or that there is some strong correlation between variables so that some can be explained in terms of the others!

Optimal ortho-normal basis

This is a less statistical perspective on PCA: we want to approximate the space where we have the data points, with a subspace that has less dimension, but that it is still linear!

Suppose we have data $\underline{x}_1, \dots, \underline{x}_n \in \mathbb{R}^p$ What is the linear space $\mathcal{L}$ of dimension $k \leq p$ such that $\mathcal{L}$ is closest to the data?

Important: if we have a cloud of points in $\mathbb{R}^2$ then a linear regression could NOT be that linear space: the distance between the a line and a point in $\mathbb{R}^2$ is the orthogonal projection. When we find the regression line we minimise distance, from the point to the line, on the y-axis:



Indeed we can't do regression without axis but we can solve the problem above without axis: in the problem above we are independent of the reference systems!

$\mathcal{L} = \text{Span}(\underline{\eta}_1, \dots, \underline{\eta}_k)$ where $\{\underline{\eta}_i\}$ is the ortho-normal basis spanning $\mathcal{L}$ We want to find the ortho-normal basis $\underline{\eta}_1, \dots, \underline{\eta}_k$ such that $\sum_{i=1}^{n} \left\| (\underline{x}_i - \bar{\underline{x}}) - \sum_{j=1}^{k} \underline{\eta}_j \underline{\eta}_j^T (\underline{x}_i - \bar{\underline{x}}) \right\|^2$ is minimised.

Note that we are just asking that the orthogonal distance between the single datum, after centring, and the liner space, is minimised!

Now set $\underline{v}_i = \underline{x}_i - \bar{\underline{x}}$ then:

$$\left\| (\underline{x}_i - \bar{\underline{x}}) - \sum_{j=1}^{k} \underline{\eta}_j \underline{\eta}_j^T (\underline{x}_i - \bar{\underline{x}}) \right\|^2 = \left\| \underline{v}_i - \sum_{j=1}^{k} \underline{\eta}_j \underline{\eta}_j^T \underline{v}_i \right\|^2 = \left(\underline{v}_i - \sum_{j=1}^{k} \underline{\eta}_j \underline{\eta}_j^T \underline{v}_i \right)^T \left(\underline{v}_i - \sum_{j=1}^{k} \underline{\eta}_j \underline{\eta}_j^T \underline{v}_i \right) =$$

$$= \underline{v}_i^T \underline{v}_i - \sum_{j=1}^{k} \underline{v}_i^T \underline{\eta}_j \underline{\eta}_j^T \underline{v}_i - \sum_{j=1}^{k} \underline{v}_i^T \underline{\eta}_j \underline{\eta}_j^T \underline{v}_i + \left(\sum_{j=1}^{k} \underline{\eta}_j \underline{\eta}_j^T \underline{v}_i \right)^T \left(\sum_{t=1}^{k} \underline{\eta}_t \underline{\eta}_t^T \underline{v}_i \right) =$$

$$= \underline{v}_i^T \underline{v}_i - \sum_{j=1}^{k} (\underline{\eta}_j^T \underline{v}_i)^2 - \sum_{j=1}^{k} (\underline{\eta}_j^T \underline{v}_i)^2 + \sum_{j=1}^{k} (\underline{\eta}_j^T \underline{v}_i)^2 = \underline{v}_i^T \underline{v}_i - \sum_{j=1}^{k} (\underline{\eta}_j^T \underline{v}_i)^2$$

Therefore the above problem can be ri-formulated as follows:

$$\text{Find } \{\underline{\eta}_j\} \text{ ortho-normal basis such that } \sum_{i=1}^{n} \left(\underline{v}_i^T \underline{v}_i - \sum_{j=1}^{k} (\underline{\eta}_j^T \underline{v}_i)^2 \right) \text{ is minimum}$$

Now note that minimising $\sum_{i=1}^{n} \left(\underline{v}_i^T \underline{v}_i - \sum_{j=1}^{k} (\underline{\eta}_j^T \underline{v}_i)^2 \right)$ is equivalent to maximising: $\sum_{i=1}^{n} \sum_{j=1}^{k} (\underline{\eta}_j^T \underline{v}_i)^2$

Now since we have finite sums we can switch the order of summation:

$$\sum_{i=1}^{n} \sum_{j=1}^{k} (\underline{\eta}_j^T \underline{v}_i)^2 = \sum_{j=1}^{k} \sum_{i=1}^{n} \underline{\eta}_j^T \underline{v}_i \underline{v}_i^T \underline{\eta}_j = \sum_{j=1}^{k} \underline{\eta}_j^T \left(\sum_{i=1}^{n} (\underline{x}_i - \bar{\underline{x}})(\underline{x}_i - \bar{\underline{x}})^T \right) \underline{\eta}_j = \sum_{j=1}^{k} \underline{\eta}_j^T (n-1) S \underline{\eta}_j = (n-1) \sum_{j=1}^{k} \underline{\eta}_j^T S \underline{\eta}_j$$

Now:

- If $k = 1$ Then: $\max_{\underline{\eta}: \|\underline{\eta}\|=1} \underline{\eta}^T S \underline{\eta} = \lambda_1$ and $\arg \max_{\underline{\eta}: \|\underline{\eta}\|=1} \underline{\eta}^T S \underline{\eta} = \underline{e}_1$
- if $k = 2$ proceed like PCA, so by induction on k we get: $\underline{\eta}_1 = \underline{e}_1, \ldots, \underline{\eta}_k = \underline{e}_k$ where $\underline{e}_i$ are the eigen-vectors of S

Thus the linear space, of dimension k, closest to the data in terms of minimising all the distance, is the one spanned by the first k eigen-vectors of S: we reach same conclusion of PCA without asking for maximum variability!

How good is this linear space in approximating the data? Consider the sum of squared residuals:

$$\sum_{i=1}^{n} d_e^2 \left((\underline{x}_i - \bar{\underline{x}}), \sum_{j=1}^{k} \underline{e}_j \underline{e}_j^T (\underline{x}_i - \bar{\underline{x}}) \right) = \sum_{i=1}^{n} (\underline{x}_i - \bar{\underline{x}})^T (\underline{x}_i - \bar{\underline{x}}) - (n-1) \sum_{j=1}^{k} \underline{e}_j^T S \underline{e}_j$$

Note that in the second term of the euclidean distance, we used $\underline{e}_j$ and not $\underline{\eta}_j$ because they are the same, as we have seen before!

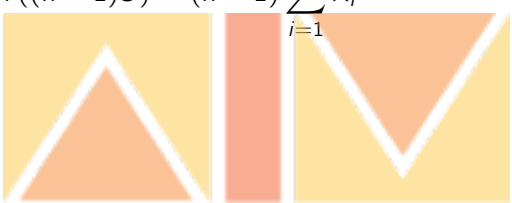
Now:

- The first term is a real number, so if we take it's trace, which is a linear operator, we still get the same number:

$$\sum_{i=1}^{n} (\underline{x}_i - \bar{\underline{x}})^T (\underline{x}_i - \bar{\underline{x}}) = Tr \left(\sum_{i=1}^{n} (\underline{x}_i - \bar{\underline{x}})^T (\underline{x}_i - \bar{\underline{x}}) \right) = \sum_{i=1}^{n} Tr \left((\underline{x}_i - \bar{\underline{x}})^T (\underline{x}_i - \bar{\underline{x}}) \right) =$$

$$= \sum_{i=1}^{n} Tr \left((\underline{x}_i - \bar{\underline{x}})(\underline{x}_i - \bar{\underline{x}})^T \right) = Tr \left(\sum_{i=1}^{n} (\underline{x}_i - \bar{\underline{x}})(\underline{x}_i - \bar{\underline{x}})^T \right) = Tr((n-1)S) = (n-1) \sum_{i=1}^{p} \lambda_i$$

because $Tr(ABC) = Tr(CAB) = Tr(BCA)$



- For the second term we have: $(n-1)\sum_{j=1}^{k} \underline{e}_j^T S \underline{e}_j = (n-1)\sum_{j=1}^{k} \lambda_j \underline{e}_j^T \underline{e}_j = (n-1)\sum_{j=1}^{k} \lambda_j$

Therefore the sum of squared residuals is:

$$SS_{res} = (n-1)\sum_{i=1}^{p} \lambda_i - (n-1)\sum_{j=1}^{k} \lambda_j = (n-1)\sum_{j=k+1}^{p} \lambda_j$$

Therefore we can see that the sum of the eigenvalues we leave out is equal to our approximation error!

The above working out we did is exactly how Pearson introduced PCA in 1900!

Independent Principal Component Analysis (ICA) is an extension of the above approach: we find the directions that maximise variability, and that are stochastically independent (so that the correlation is zero). However there is no analytical solution!

Note: If two variables are stochastically independent then they are uncorrelated. The vice-versa doesn't holds, unless we are working in the Gaussian world.

Note: another possible extension is to forget about linearity: we find the best non-linear space that best approximate the data: we use kernel transformations.

6 Lecture 10: 24th Of March 2020

Multivariate Gaussian Distribution:

Suppose we have a random vector $\underline{X} \in \mathbb{R}^p$ Then: $\nu_{\underline{X}} : \mathcal{B} \to [0,1]$ such that: $\nu_{\underline{X}}(B) = \mathbb{P}(\underline{X} \in B)\ \forall B \subseteq \mathcal{B}$ where: $\mathcal{B}$ is the Borel $\sigma-$algebra over $\mathbb{R}^p$ and $B \in \mathcal{B}$ are the Borel sets over $\mathbb{R}^p$

Then: $\nu_{\underline{X}}(B) = \int_B f(\underline{x}) d\underline{x}$ where f is the density function. It has the following properties:

- $f \geq 0$
- $\int_{\mathbb{R}^p} f(\underline{x}) d\underline{x} = 1$

Definition: Let $\underline{X} \in \mathbb{R}^p$ be a random vector with density f and let $\underline{\mu} \in \mathbb{R}^p$ and let Σ be a $p \times p$ positive definite matrix. Then:
The random vector is Gaussian: $\underline{X} \sim \mathcal{N}_p(\underline{\mu}, \Sigma)$ (where the sub-script p indicates that the distribution is p dimensional)
If: $f(\underline{x}) = \frac{1}{\sqrt{(2\pi)^p det(\Sigma)}} \exp\left(-\frac{1}{2}(\underline{x}-\underline{\mu})^T \Sigma^{-1} (\underline{x}-\underline{\mu})\right) \propto \exp\left(-\frac{1}{2} d^2_{\Sigma^{-1}}(\underline{x}, \underline{\mu})\right)$

Note: the first fraction is just a normalisation constant!

We can see that the the density diminishes exponentially fast with the square of the *Mahalanobis Distance:* induced by Σ^{-1}

Contour Plots for f are given by:

$$\{\underline{x} \in \mathbb{R}^p : f(\underline{x}) = const\} = \{\underline{x} \in \mathbb{R}^p : (\underline{x}-\underline{\mu})^T \Sigma^{-1} (\underline{x}-\underline{\mu}) = const^2\} = \{\underline{x} \in \mathbb{R}^p : d^2_{\Sigma^{-1}}(\underline{x}, \underline{\mu}) = const^2\}$$

For example, if $p = 2$ then the contour plots are ellipses centred in μ, whose axis are $\underline{e}_1, \underline{e}_2$ given that: $\Sigma = \sum_{i=1}^{p} \lambda_i \underline{e}_i \underline{e}_i^T$ is the spectral decomposition of Σ! The length of each axis is proportional to $\sqrt{\lambda_i}$, indeed the length of the axis is the square root of the reciprocal of the eigenvalue, but we need to consider the distance induced by Σ^{-1} and Σ^{-1} has eigenvalues given by λ_i^{-1}

Proposition: if $\underline{X} \sim \mathcal{N}_p(\underline{\mu}, \Sigma)$ then: $\mathbb{E}[\underline{X}] = \underline{\mu}$ and $Cov(\underline{X}) = \Sigma$

Theorem: $\underline{X} \sim \mathcal{N}_p(\underline{\mu}, \Sigma) \iff \underline{a}^T \underline{X} \sim \mathcal{N}_1(\underline{a}^T \underline{\mu}, \underline{a}^T \Sigma \underline{a})\ \forall \underline{a} \in \mathbb{R}^p$ This is a characterisation of the Gaussian distribution! To prove this its enough to use characteristics functions!

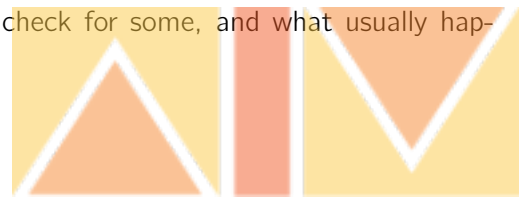
The above **Theorem** however doesn't hold in general!

Note: if we want to define a Gaussian distribution in a Hilbert Space: then if we take the inner product between then density, and an object in that space, and if what we get from the inner product is Gaussian, Then: the previous density we used in the inner product is itself Gaussian.

The impact of the above **Theorem** is practical and theoretical:

- We get data, and then we take a linear combination of the data, for example by projecting in a linear space. Then with *Shapiro Test* and with the *qqplot* we check the Gaussianity assumption of the one dimensional random variable we obtained by projecting the original variable in a linear space: if it holds, for all infinite possible linear combinations, then the multi-variate distribution is Gaussian.

  Since we can't check all the infinite possible linear combinations we just check for some, and what usually happens is that we find one for which it doesn't hold.



- From the theoretical point of view the Theorem is useful to prove many other results.

Corollary: If $\underline{X} = (X_1, \ldots, X_p)^T \sim \mathcal{N}_p(\underline{\mu}, \Sigma)$ Then: $X_i \sim \mathcal{N}_1(\mu_i, \sigma_{ii})$ where $\Sigma = [\sigma_{ij}]$

Thus if a vector is Gaussian then each of its components are Gaussian: indeed each component is a linear combination of the vector since: $X_i = \underline{u}_i^T \underline{X}$ where $\underline{u}_i$ is all zero expect it has a 1 in position i

The vice-versa doesn't hold!

Note: If $\underline{X} \sim \mathcal{N}_2(\underline{0}, I)$ then in this case the contours plot are circles centred in zero, since the Gaussian bell is isotropic (goes down as smoothly in all directions).

Proposition: Let $\underline{X} \sim \mathcal{N}_p(\underline{\mu}, \Sigma)$ and let A be a $q \times p$ matrix of constants. Then: $A\underline{X} \in \mathbb{R}^q$ and $A\underline{X} \sim \mathcal{N}_q(A\underline{\mu}, A\Sigma A^T)$

Proof of the above Proposition: Let $\underline{a} \in \mathbb{R}^q$ be a vector of constants: we need to prove that $\underline{a}^T(A\underline{X})$ has Gaussian distribution.

Then: ${}^T(A\underline{X}) = (\underline{a}^T A)\underline{X} = (A^T \underline{a})^T \underline{X}$ Now since $A^T \underline{a} \in \mathbb{R}^p$ from the above Theorem we have that:
$(A^T \underline{a})\underline{X} \sim \mathcal{N}_1((A^T \underline{a})^T \underline{\mu}, (A^T \underline{a})^T \Sigma (A^T \underline{a})) = \mathcal{N}_1(\underline{a}^T (A\underline{\mu}), \underline{a}^T A\Sigma A^T \underline{a})$

Since this holds for any $\underline{a} \in \mathbb{R}^q$ we have that: $A\underline{X} \sim \mathcal{N}_q(A\underline{\mu}, A\Sigma A^T)$ which is the thesis!

Note: in this proof we used both the necessary and the sufficient condition of the above theorem!

Proposition: If $\underline{X} \sim \mathcal{N}_p(\underline{\mu}, \Sigma)$ and $\underline{d} \in \mathbb{R}^p$ then: $\underline{X} + \underline{d} \sim \mathcal{N}_p(\underline{\mu} + \underline{d}, \Sigma)$ This means that translating a random vector, only translates its mean!

Notation: Suppose now that $\underline{X} = (\underline{X}_1, \underline{X}_2)^T$ with $\underline{X}_1 \in \mathbb{R}^q, \underline{X}_2 \in \mathbb{R}^{p-q}$ with $q < p$ Then:

- $\underline{\mu} = (\underline{\mu}_1, \underline{\mu}_2)^T$ with $\underline{\mu}_1 \in \mathbb{R}^q, \underline{\mu}_2 \in \mathbb{R}^{p-q}$
- $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$ with:
  - Σ_{11} is a $q \times q$ matrix
  - Σ_{12} is a $q \times (p-q)$ matrix
  - Σ_{21} is a $(p-q) \times q$ matrix
  - Σ_{22} is a $(p-q) \times (p-q)$ matrix.

Proposition: Using the above **notation**. If $\underline{X} = (\underline{X}_1, \underline{X}_2)^T \sim \mathcal{N}_p(\underline{\mu}, \Sigma)$ Then: $\underline{X}_1 \sim \mathcal{N}_q(\underline{\mu}_1, \Sigma_{11})$ that is: subset of any number of components of X is still Gaussian.

Proof of the above Proposition: Let $A = [I|O]$ where I is a $q \times q$ matrix and O is a $q \times (p-q)$ matrix.

Now: $A\underline{X} \sim \mathcal{N}_q(A\underline{\mu}, A\Sigma A^T)$ because of previous proposition. But: $A\underline{X} = \underline{X}_1$ and $A\underline{\mu} = \underline{\mu}_1$ and $A\Sigma A^T = \Sigma_{11}$ so we get the thesis!

Theorem: Using the above **notation**. If $\underline{X} = (\underline{X}_1, \underline{X}_2)^T \sim \mathcal{N}_p(\underline{\mu}, \Sigma)$ Then: $\underline{X}_1 \perp\!\!\!\perp \underline{X}_2 \iff \Sigma_{12} = 0$ which in turn implies that $\Sigma_{21} = 0$

Note: this theorem is saying that saying that the co-variance is zero is the same thing as saying that the random vectors are stochastically independent, in the Gaussian world. This doesn't hold for any distribution!

Proof of the Theorem above: We need to write the density and see that it can be split in the product of two densities: $f_{\underline{X}} = f_{\underline{X}_1} \times f_{\underline{X}_2}$ From this it follows that $\underline{X}_1 \perp\!\!\!\perp \underline{X}_2$

Theorem: Using the above **notation**.
If $\underline{X} = (\underline{X}_1, \underline{X}_2)^T \sim \mathcal{N}_p(\underline{\mu}, \Sigma)$ Then: $\underline{X}_1 | \underline{X}_2 = \underline{x}_2 \sim \mathcal{N}_q(\underline{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\underline{x}_2 - \underline{\mu}_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})$

Note that since we write Σ_{22}^{-1}, we assume its invertible: $det(\Sigma_{22}) \neq 0$

Note: we want independent random variables but not independent features: from the dependence of features we make inference!

Theorem: $\underline{Z} \sim \mathcal{N}_p(\underline{0}, I)$ and $\underline{Z} = (Z_1, \ldots, Z_p)$ then $Cov(\underline{Z}) = I \iff Z_1, \ldots, Z_p \overset{iid}{\sim} \mathcal{N}(0, 1)$

Suppose now that A is a $p \times p$ matrix then: $A\underline{Z} \sim \mathcal{N}_p(\underline{0}, AIA^T) = \mathcal{N}_p(\underline{0}, AA^T)$ Then: $A\underline{Z} + \underline{\mu} \sim \mathcal{N}_p(\underline{\mu}, AA^T)$ Now if: $A = \Sigma^{1/2} = \sum_{i=1}^{p} \lambda_i^{1/2} \underline{e}_i \underline{e}_i^T \implies \Sigma^{1/2}\underline{Z} + \underline{\mu} \sim \mathcal{N}_p(\underline{\mu}, \Sigma)$
Thus: if we have vector of standard normal distribution with independent components, then we can generate any multivariate Gaussian distribution by linear transformation!

Note: Looking at the above in reverse we have: $\underline{X} \sim \mathcal{N}_p(\underline{\mu}, \Sigma)$ then: $\Sigma^{-1/2}(\underline{X} - \underline{\mu}) \sim \mathcal{N}_p(\underline{0}, I)$

Proof of above Theorem:
$A = \begin{bmatrix} I & -\Sigma_{12}\Sigma_{22}^{-1} \\ 0 & I \end{bmatrix}$ where I is a $q \times q$ matrix, and $-\Sigma_{12}\Sigma_{22}^{-1}$ is a $q \times (p-q)$ matrix and O is a $(p-q) \times q$ matrix and I is a $(p-q) \times (p-q)$

$$A(\underline{X} - \underline{\mu}) = A(\underline{X}_1 - \underline{\mu}_1, \underline{X}_2 - \underline{\mu}_2)^T = \begin{bmatrix} \underline{X}_1 - \underline{\mu}_1 - \Sigma_{12}\Sigma_{22}^{-1}(\underline{X}_2 - \underline{\mu}_2) \\ \underline{X}_2 - \underline{\mu}_2 \end{bmatrix} \sim \mathcal{N}_p\left((\underline{0}, \underline{0})^T, A\Sigma A^T\right) =$$

$$= \mathcal{N}_p\left((\underline{0}, \underline{0})^T, \begin{bmatrix} \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} & 0 \\ 0 & \Sigma_{22} \end{bmatrix}\right)$$

This means that: $\underline{X}_1 - \underline{\mu}_1 - \Sigma_{12}\Sigma_{22}^{-1}(\underline{X}_2 - \underline{\mu}_2)$ and $\underline{X}_2 - \underline{\mu}_2$ are stochastically independent, since the co-variance block in the last passage are zero!

Therefore:

- We proved independence between two function, of the components of the original vectors.
- We also proved that $\underline{W} = \underline{X}_1 - \underline{\mu}_1 - \Sigma_{12}\Sigma_{22}^{-1}(\underline{X}_2 - \underline{\mu}_2) \sim \mathcal{N}_q(\underline{0}, \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})$

Since $\underline{W} \perp\!\!\!\perp \underline{X}_2$ Then we have that: $\underline{W} | \underline{X}_2 = \underline{x}_2 \sim \mathcal{N}_q(\underline{0}, \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})$ But this is the same distribution, with the same parameters, of $\underline{W}$ thus: $(\underline{W} | \underline{X}_2 = \underline{x}_2) = \underline{X}_1 - \underline{\mu}_1 - \Sigma_{12}\Sigma_{22}^{-1}(\underline{x}_2 - \underline{\mu}_2)$

Thus, given that $\underline{X}_2 = \underline{x}_2$ we have that: $\underline{X}_1 - \underline{\mu}_1 - \Sigma_{12}\Sigma_{22}^{-1}(\underline{x}_2 - \underline{\mu}_2) \sim \mathcal{N}_q(\underline{0}, \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})$ Then by translating we have: $\underline{X}_1 | \underline{X}_2 = \underline{x}_2 \sim \mathcal{N}_q(\underline{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\underline{x}_2 - \underline{\mu}_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})$ which is the thesis.

Note: $Cov(\underline{X}_1 | \underline{X}_2 = \underline{x}_2) = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$ doesn't depend on $\underline{x}_2$ and it's called *partial co-variance!*

Example: Suppose that $p = 2$ and let $\underline{X} = (X, Y)^{-1} \sim \mathcal{N}_2\left((\mu_X, \mu_Y)^T, \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix}\right)$

Now: $Y \sim \mathcal{N}_1(\mu_y, \sigma_{yy})$ so: $\mathbb{P}[Y \in [\mu_y \pm 2\sqrt{\sigma_y}]] = 0.95$ Moreover: $Y|X = x \sim \mathcal{N}_1(\mu_y + \sigma_{xy}\sigma_{xx}^{-1}(x - \mu_X), \sigma_{yy} - \sigma_{xy}\sigma_{xx}^{-1}\sigma_{xy})$

$\rho_{xy} = \dfrac{\sigma_{xy}}{\sqrt{\sigma_{xx}\sigma_{yy}}} \in [-1, 1]$ Thus:

$$Y|X = x \sim \mathcal{N}_1\left(\mu_y + \frac{\sigma_{xy}}{\sigma_{xx}}(x - \mu_x), \sigma_{yy} - \frac{\sigma_{xy}^2}{\sigma_{xx}\sigma_{yy}}\right) = \mathcal{N}_1\left(\mu_Y + \sigma_{xy}(x - \mu_x)\frac{1}{\sigma_{xx}}, \sigma_{yy}(1 - \rho^2)\right)$$

Since $Y \sim \mathcal{N}_1(\mu_y, \sigma_{yy})$ and $1 - \rho^2 \leq 1$ Thus: if there is some correlation between X, Y the uncertainty of $Y|X = x$ is smaller than the one of Y

Namely knowing some information reduces the variability of a factor of $1 - \rho^2$

$\mathbb{E}[Y|X = x] = \mu_y + \dfrac{\sigma_{xy}}{\sigma_{xx}}(x - \mu_x) \implies y = \mu_y + \dfrac{\sigma_{xy}}{\sigma_{xx}}(x - \mu_x)$ which is a linear regression function!

In the Gaussian world, the regression function (i.e: the best we can do if we want to make prediction minimising MSE) is a linear function!

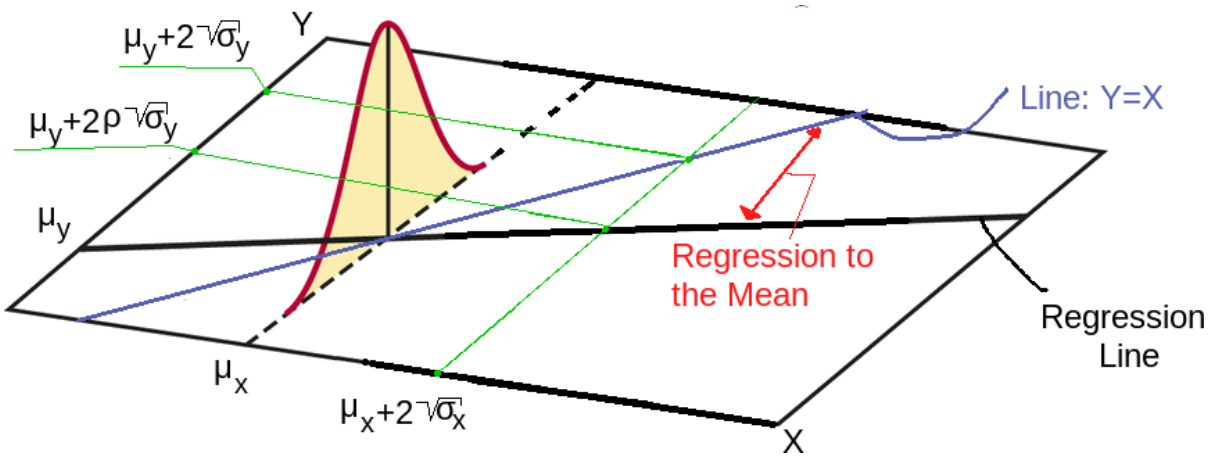
Now dividing the above linear regression function by the standard deviation of y we have:

$$\frac{y - \mu_y}{\sqrt{\sigma_{yy}}} = \frac{\sigma_{xy}}{\sqrt{\sigma_{xx}}\sqrt{\sigma_{yy}}}\frac{(x - \mu_x)}{\sqrt{\sigma_{xx}}} \implies \frac{y - \mu_y}{\sqrt{\sigma_{yy}}} = \rho_{xy}\frac{x - \mu_x}{\sqrt{\sigma_{xx}}}$$

where ρ_{xy} is the correlation coefficient.

So if we take standardised data the above regression function is a line going through the origin with slope equal to σ_{xy}

Consider the following figure:



We can see that if we are 2 standard deviations above the mean in x, then due to the regression line we are only $2\rho\sqrt{\sigma_{yy}}$ standard deviations above the mean in y

Therefore exceptionalities in x are diminished in y and there is a scaling depending on the correlation between x, y: we predict on the regression line, so the prediction are closer to the mean, with respect to the data we use as input for making that prediction! This is regression to the mean!

Note that in the above figure the Bell Distribution is the Gaussian distribution $\mathcal{N}(\mu_y, \sigma_{yy})$

For example a very tall dad has a tall kid but not as tall. This is clear regression to the mean effect: since $|\rho| < 1$ exceptionalities have to be reduced!

Very often the regression effect causes the regression fallacy: we try to interpret the regression effect as if there is some sort of causation explanation for it.

For example suppose Alex got 30 in Calculus 1, and Y in Calculus 2:

- The regression to the mean effect says that probably Alex will get 28 in Calculus 2!
- The regression fallacy is saying: Alex relaxed and didn't study so hard because he got 30 and that's why he got less in Calculus 2! No! There is no causation!

7 Lecture 11: 26th Of March 2020

Proposition: Consider a random vector $\underline{X} \sim \mathcal{N}_p(\underline{\mu}, \Sigma)$ If $det(\Sigma) > 0$ then: $(\underline{X} - \underline{\mu})^T \Sigma^{-1} (\underline{X} - \underline{\mu}) \sim \chi^2(p)$ where p are the degrees of freedom!

Proof of the Proposition above: Recall that if $Z_1, \dots, Z_p \overset{iid}{\sim} \mathcal{N}_1(0,1)$ then: $\sum_{i=1}^{p} Z_i^2 \sim \chi^2(p)$

Consider the spectral decomposition of Σ then: $\Sigma = \sum_{i=1}^{p} \lambda_i \underline{e}_i \underline{e}_i^T$ If $P = [\underline{e}_1 \dots \underline{e}_p]$ and $\Lambda = diag(\lambda_1, \dots, \lambda_p)$ then:

$\Sigma = P \Lambda P^T$ Thus: $\Lambda^{-1/2} P^T (\underline{X} - \underline{\mu}) \sim \mathcal{N}_p(\underline{0}, \Lambda^{-1/2} P \Sigma P^T \Lambda^{-1/2})$ since it's a linear transformation of a Gaussian variable.

Then: $\Lambda^{-1/2} P \Sigma P^T \Lambda^{-1/2} = \Lambda^{-1/2} P^T P \Lambda P^T P \Lambda^{-1/2} = \Lambda^{-1/2} \Lambda \Lambda^{-1/2} = I$ since $P'P = I$

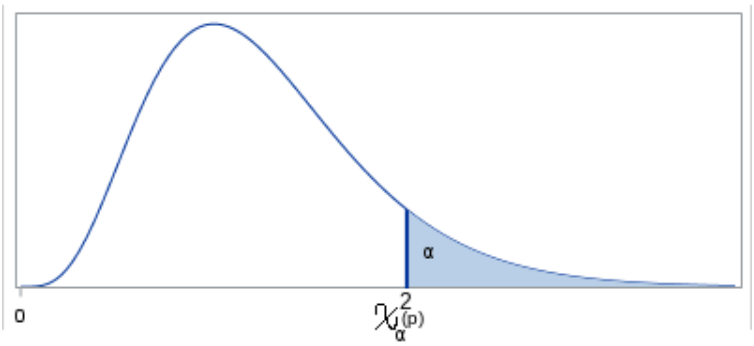
Therefore: $\underline{Z} = \Lambda^{-1/2} P^T (\underline{X} - \underline{\mu}) \sim \mathcal{N}_p(\underline{0}, I)$ and if $\underline{Z} = (Z_1, \dots, Z_p)^T$ then: $Z_1, \dots, Z_p \overset{iid}{\sim} \mathcal{N}_1(0,1)$

So: $(\underline{X} - \underline{\mu})^T \Sigma^{-1/2} (\underline{X} - \underline{\mu}) = (\underline{X} - \underline{\mu})^T P \Lambda^{-1/2} \Lambda^{-1/2} P^T (\underline{X} - \underline{\mu}) = \underline{Z}^T \underline{Z} = \sum_{i=1}^{p} Z_i^2 \sim \chi^2(p)$ which is the thesis!

Observation: What happens if $det(\Sigma) = 0$?
$\Sigma = \sum_{i=1}^{p} \lambda_i \underline{e}_i \underline{e}_i^T$ with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k > 0 = \lambda_{k+1} = \dots = \lambda_p$ That is: $k = rank(\Sigma)$ Then we can define the

Moore-Penrose Generalised Inverse: $\Sigma^\dagger = \Sigma^- = \sum_{i=1}^{k} \frac{1}{\lambda_i} \underline{e}_i \underline{e}_i^T$ Then in this case: $(\underline{X} - \underline{\mu})^T \Sigma^\dagger (\underline{X} - \underline{\mu}) \sim \chi^2(k)$

Corollary: Let $\alpha \in (0,1)$ then if $det(\Sigma) > 0$ we have: $\mathbb{P}\left[(\underline{X} - \underline{\mu})^T \Sigma^{-1} (\underline{X} - \underline{\mu}) \leq \chi^2_\alpha(p)\right] = 1 - \alpha$ where $\chi^2_\alpha(p)$ is the quantile of order α



The above corollary is saying that if we take the *Mahalanobis Distance*, then we have an ellipses centred in $\underline{\mu}$ with squared radius equal to $\chi^2_\alpha(p)$ then that neighbourhood contains $1 - \alpha$ of the mass!

Estimating parameters of the Gaussian model: estimators of μ, Σ

We have our data $\mathbb{X} = [\underline{x}_1^T, \dots, \underline{x}_n^T]^T$ and we assume $\underline{x}_i^T$ is the observation from a random vector $\underline{X}_i$ We assume that $\underline{X}_1, \dots, \underline{X}_n \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}, \Sigma)$ where $\underline{\mu}$ and Σ are unknown.

The obvious choice is to take the sample mean and sample variance: $\bar{\underline{X}} = \frac{1}{n} \sum_{i=1}^{p} \underline{X}_i$ and $S = \frac{1}{n-1} \sum_{i=1}^{n} (\underline{X}_i - \bar{\underline{X}})(\underline{X}_i - \bar{\underline{X}})^T$

which are unbiased estimator for $\underline{\mu}$ and Σ

However in the Gaussian world we have a model for the density of the random vector so there is another way to find

the best estimator: MLE

Maximum Likelihood Estimation (MLE)

If we have a continuous density then the probability of a single value is zero: we want to make a slight abuse of notation now!

$$\mathbb{P}[\underline{X}_1 = d\underline{x}_1, \dots, \underline{X}_n = d\underline{x}_n] \underset{iid}{=} \prod_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^p det(\Sigma)}} \exp\left(-\frac{1}{2}(\underline{x}_i - \underline{\mu})\Sigma^{-1}(\underline{x}_i - \underline{\mu})\right) d\underline{x}_1 \dots d\underline{x}_n$$

Definition: $L : (\underline{\mu}, \Sigma) \to \mathbb{P}[\underline{X}_1 = d\underline{x}_1, \dots, \underline{X}_n = d\underline{x}_n]$ given $\underline{X}_i = \underline{x}_i$, this function is called **likelihood**. It is the likelihood for having those parameters given that we have observed that data!

Finding the MLE means finding the values of parameters so that the likelihood is maximised.

Theorem:

$$\arg \max_{(\underline{\mu},\Sigma):\ \underline{\mu}\in\mathbb{R}^p, \Sigma \text{ is a } p\times p \text{ positive definite matrix}} L(\underline{\mu}, \Sigma | \underline{X}_1 = \underline{x}_1, \dots, \underline{X}_n = \underline{x}_n) = (\hat{\underline{\mu}}, \hat{\Sigma})$$

where $L(\underline{\mu}, \Sigma | \underline{X}_1 = \underline{x}_1, \dots, \underline{X}_n = \underline{x}_n) = \prod_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^p det(\Sigma)}} \exp\left(-\frac{1}{2}(\underline{x}_i - \underline{\mu})\Sigma^{-1}(\underline{x}_i - \underline{\mu})\right)$

Then: $\hat{\underline{\mu}} = \bar{\underline{X}}$ sample mean and $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(\underline{X}_i - \bar{\underline{X}})(\underline{X}_i - \bar{\underline{X}}) = \frac{n-1}{n}S$ which is a biased estimator for Σ

MLE: We don't know if the estimator is right on average (i.e: unbiased) but it's the best we can do today: we maximise today the likelihood of having observed what we have observed! Moreover MLE have nice properties which may be really useful when thing aren't nice (i.e: we don't have Gaussianity)

Invariance property of MLE: Suppose $\underline{\theta} \in \mathbb{R}^k$ is a parameter, for instance: $\underline{\theta} = (\underline{\mu}, \Sigma)$ and suppose that $\hat{\underline{\theta}} = \hat{\underline{\theta}}(\text{data})$ is the MLE estimator of $\underline{\theta}$ If $h : \mathbb{R}^k \to \mathbb{R}^j$ is a mapping, then: $h(\hat{\underline{\theta}}) = \widehat{h(\underline{\theta})}$

Example: We know that $\hat{\Sigma} = \frac{n-1}{n}S$ is the MLE for Σ

Now if $\underline{X}_i \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}, \Sigma)\ \forall i = 1, \dots, n$ how do we find an estimator of λ_1, the first eigenvalue of Σ?

Well $\hat{\lambda}_1$ is such that $\hat{\Sigma} = \sum_{i=1}^{p} \hat{\lambda}_i \hat{\underline{e}}_i \hat{\underline{e}}_i^T$ thanks to the **Invariance Property!**

Uncertainty related with the estimates: whats the distribution of the estimators?

We want to find distribution of $\bar{\underline{X}}, \hat{\Sigma}$ assuming that the samples are generated by: $\underline{X}_1, \dots, \underline{X}_n \tilde{iid} \mathcal{N}_p(\underline{\mu}, \Sigma)$

Proposition: $\bar{\underline{X}} \sim \mathcal{N}_p\left(\underline{\mu}, \frac{1}{n}\Sigma\right)$ This means that the sample mean is Gaussian: indeed its a linear combination of Gaussian variables!

Proof: Take a long column vector: $\tilde{\underline{X}} = (\underline{X}_1, \dots, \underline{X}_n)^T \in \mathbb{R}^{np}$ this is an $np \times 1$ matrix!

Then: $\tilde{\underline{X}} \sim \mathcal{N}_{np}((\underline{\mu}, \dots, \underline{\mu})^T, diag(\Sigma, \dots, \Sigma))$ since each block has Gaussian distribution and are all independent!

Let $A = [I, I, \dots, I]$ be an $p \times np$ matrix: we have n matrices I which are $p \times p$ matrices!

Now: $\frac{1}{n}A\tilde{\underline{X}} = \bar{\underline{X}}$ indeed $A\tilde{\underline{X}} = \left(\sum_{i=1}^{n} X_{i1}, \dots, \sum_{i=1}^{n} X_{ip}\right)^T$ Thus:

$$\bar{\underline{X}} = \frac{1}{n}A\tilde{\underline{X}} \sim \mathcal{N}_p\left(\frac{1}{n}A(\underline{\mu}, \dots, \underline{\mu})^T, \frac{1}{n^2}A\ \text{diag}(\Sigma, \dots, \Sigma)\ A^T\right) = \mathcal{N}_p\left(\underline{\mu}, \frac{1}{n^2}n\Sigma\right)$$

which is the thesis!

Definition: Let $\underline{Z}_1, \dots, \underline{Z}_m \overset{iid}{\sim} \mathcal{N}_p(\underline{0}, \Sigma)$ with Σ a $p \times p$ non-singular matrix. Then:

$$\sum_{i=1}^{m} \underline{Z}_i \underline{Z}_i^T \sim Wishart(\Sigma, m)$$

Note that the $\underline{Z}_i$ can be interpreted as co-variance matrices! Note that the *Wishart Distribution* is defined on a Riemannian Manifold, in which each point is a matrix! So now we have the notion of distribution of a matrix!

Note that in the above we have three parameters: p, m, Σ

Properties of the Wishart Distribution:

1) Given $A_1 \sim Wishart(\Sigma, m_1), A_2 \sim Wishart(\Sigma, m_2)$ If they are stochastically independent: $A_1 \perp\!\!\!\perp A_2$ Then: $A_1 + A_2 \sim Wishart(\Sigma, m_1 + m_2)$

   Proof: $A_1 = \sum_{i=1}^{m_1} \underline{Z}_i \underline{Z}_i^T$ with $\underline{Z}_1, \dots, \underline{Z}_{m_1} \overset{iid}{\sim} \mathcal{N}_p(\underline{0}, \Sigma)$ and $A_2 = \sum_{i=1}^{m_2} \tilde{\underline{Z}}_i \tilde{\underline{Z}}_i^T$ with $\tilde{\underline{Z}}_1, \dots, \tilde{\underline{Z}}_{m_2} \overset{iid}{\sim} \mathcal{N}_p(\underline{0}, \Sigma)$

   Now we take them in order: $\underline{Z}_1, \dots, \underline{Z}_{m_1}, \tilde{\underline{Z}}_1, \dots, \tilde{\underline{Z}}_{m_2}$ Then: re-naming them, maintaining the order we have: $\underline{W}_1, \dots, \underline{W}_{m_1}, \underline{W}_{m_1+1}, \dots, \underline{W}_{m_1+m_2}$ Since everything is independent we have: $\underline{W}_1, \dots, \underline{W}_{m_1+m_2} \overset{iid}{\sim} \mathcal{N}_p(\underline{0}, \Sigma)$

   Thus: $A_1 + A_2 = \sum_{i=1}^{m_1} \underline{Z}_i \underline{Z}_i^T + \sum_{i=1}^{m_2} \tilde{\underline{Z}}_i \tilde{\underline{Z}}_i^T = \sum_{i=1}^{m_1+m_2} \underline{W}_i \underline{W}_i^T \sim Wishart(\Sigma, m_1 + m_2)$

2) If C is $k \times p$ constant matrix and $A \sim Wishart(\Sigma, m)$ then: $CAC^T \sim Wish(C\Sigma C^T, m)$

   Proof: $A = \sum_{i=1}^{m} \underline{Z}_i \underline{Z}_i^T$ with $\underline{Z}_i \overset{iid}{\sim} \mathcal{N}_p(\underline{0}, \Sigma)$ then: $CAC^T = \sum_{i=1}^{m} C\underline{Z}_i \underline{Z}_i^T C^T = \sum_{i=1}^{m} \underline{W}_i \underline{W}_i^T$ where: $\underline{W}_i = C\underline{Z}_i$

   Since $\underline{W}_i \overset{iid}{\sim} \mathcal{N}_k(\underline{0}, C\Sigma C^T)$ we have that: $CAC^T \sim Wishahrt(C\Sigma C^T, m)$ by the previous definition!

3) Take $\sigma^2 \in \mathbb{R}$ with $\sigma^2 > 0$ and $A \sim Wishart(\Sigma, m)$ Then: $\sigma^2 A \sim Wishart(\sigma^2 \Sigma, m)$

   Proof: $\sigma^2 A = \sum_{i=1}^{m} \sigma \underline{Z}_i \underline{Z}_i^T \sigma$ since $\underline{Z}_i \overset{iid}{\sim} \mathcal{N}_p(\underline{0}, \Sigma)$

   Call $\underline{W}_i = \sigma \underline{Z}_i$ Then: $\underline{W}_i \overset{iid}{\sim} \mathcal{N}_p(\underline{0}, \sigma^2 \Sigma)$ Then by the previous definition: $\sigma^2 A \sim Wish(\sigma^2 \Sigma, m)$

4) The Wishart is a multivariate extension of the χ^2 indeed:

   If $A \sim Wish(\Sigma, m)$ with $\Sigma = \sigma^2$ that is a 1×1 matrix (i.e: a number) Then: $p = 1$ Moreover: $A = \sum_{i=1}^{m} Z_i Z_i$ Note that now Z_i is not a vector!

   Since $Z_1, \dots, Z_m \overset{idd}{\sim} \mathcal{N}_1(0, \sigma^2)$ we have: $\frac{1}{\sigma^2} A = \sum_{i=1}^{m} \frac{Z_i}{\sigma} \frac{Z_i}{\sigma}$ Calling: $W_i = \frac{Z_i}{\sigma}$ then: $W_i \overset{iid}{\sim} \mathcal{N}_1(0, 1)$ Thus: $\frac{1}{\sigma^2} A \sim \chi^2(m) \implies A \sim \sigma^2 \chi^2(m)$

   Conclusion: If $A \sim Wish(\Sigma, m)$ with Σ a 1×1 matrix. Then: $A \sim \Sigma \chi^2(m)$

Note: $a\chi^2(m)$ is just shorthand for writing: $\frac{1}{a} A \sim \chi^2(m)$ For instance: $X \sim 5\mathcal{N}_1(\mu, \sigma^2)$ means: $\frac{X}{5} \sim \mathcal{N}_1(\mu, \sigma^2)$ So **attention:** its the random variable you multiply and not the distribution!

Property: If $\underline{0} \neq \underline{c} \in \mathbb{R}^p$ and $A \sim Wishart(\Sigma, m)$ where Σ is a $p \times p$ matrix then: $\underline{c}^T A \underline{c} \sim Wishart(\underline{c}^T \Sigma \underline{c}, m)$ by **Property 2)** above.

But now $\underline{c}^T A \underline{c} > 0$ is a number! Thus: $Wishart(\underline{c}^T \Sigma \underline{c}, m) \sim \underline{c}^T \Sigma \underline{c} \chi^2(m) \implies \underline{c}^T A \underline{c} \sim (\underline{c}^T \Sigma \underline{c}) \chi^2(m)$ Which means that: $\underline{c}^T A \underline{c} \frac{1}{\underline{c}^T \Sigma \underline{c}} \sim \chi^2(m)$

Theorem: If $\underline{X}_1, \dots, \underline{X}_n \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}, \Sigma)$ Then: $\sum_{i=1}^{n} (\underline{X}_i - \bar{\underline{X}})(\underline{X}_i - \bar{\underline{X}})^T \sim Wishart(\Sigma, n-1)$

Note: By estimating the mean we removed one degree of freedom: indeed we went from a n to $n-1$ degrees of freedom!

Corollary: By using the above **Theorem** and the **Property 3)** we have the following:

$$S = \frac{1}{n-1} \sum_{i=1}^{n} (\underline{X}_i - \bar{\underline{X}})(\underline{X}_i - \bar{\underline{X}}) \sim Wishart\left(\frac{1}{n-1}\Sigma, n-1\right) \text{ and } \hat{\Sigma} = \frac{n-1}{n} S \sim Wishart\left(\frac{1}{n}\Sigma, n-1\right)$$

Summary of everything in one Theorem: Let $\underline{X}_1, \dots, \underline{X}_n \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}, \Sigma)$ then:

1) $\bar{\underline{X}} \sim \mathcal{N}_p\left(\underline{\mu}, \frac{1}{n}\Sigma\right)$
2) $(n-1)S \sim Wishart(\Sigma, n-1)$
3) $\bar{\underline{X}} \perp\!\!\!\perp S$ It means that they are stochastically independent!

Theorem: $\bar{\underline{X}}$ and S are sufficient statistics: if the data is generated by Gaussian distribution, then all we need to know, to do statistics, is $\bar{\underline{X}}$ and S

Note: we can transform data and make it more Gaussian! Some useful transformations for making data more Gaussian are:

- Suppose the data has an empirical density with long tail similar like a χ^2, then we can take a logarithm transformation: it just compress the first part and extend the second part. Note that the logarithm needs to be centred at suitable point!

  We can also try Box-Cox transformations!

- If x are proportions, so they belong in $[0, 1]$, we can take a sigmoidal transformation.

  Examples are: $\log\left(\frac{x}{1-x}\right) = logit(x)$ and $\arctan x$

Asymptotic results:

- **Law of Large numbers (LLN):**
  Suppose we have an infinite sequence of random vectors $\underline{X}_1, \dots, \underline{X}_n, \dots$ which are independent and identically distributed, with: $\mathbb{E}[\underline{X}_1] = \underline{\mu}$ and $Cov(\underline{X}_1) = \Sigma$ Suppose that these exist finite (e.g: $\underline{X}_i$ are in L^2)

  Then: $\bar{\underline{X}} = \frac{1}{n} \sum_{i=1}^{n} \underline{X}_i$ converges in probability (i.e: in measure) to μ as $n \to \infty$

  Moreover: $S = \frac{1}{n-1} \sum_{i=1}^{n} (\underline{X}_i - \bar{\underline{X}})(\underline{X}_i - \bar{\underline{X}})^T$ converges in probability to Σ as $n \to \infty$

  Note: convergence in probability implies convergence in law!

- **Central Limit Theorem (CLT):**
  Suppose we have an infinite sequence of random vectors $\underline{X}_1, \ldots, \underline{X}_n, \ldots$ which are independent and identically distributed, with: $\mathbb{E}[\underline{X}_1] = \underline{\mu}$ and $Cov(\underline{X}_1) = \Sigma$ Suppose that these exist finite (e.g: $\underline{X}_i$ are in L^2)

  Then: $\sqrt{n}(\bar{\underline{X}} - \underline{\mu}) \sim AN_p(\underline{0}, \Sigma)$ where AN means asymptotically normal, that is:
  for large n one can approximate the distribution of $\sqrt{n}(\bar{\underline{X}} - \underline{\mu})$ with $\mathcal{N}_p(\underline{0}, \Sigma)$ Note that we have convergence in Law!

  In practice: for large n we have $\bar{\underline{X}} \sim \mathcal{N}_p\left(\underline{\mu}, \frac{1}{n}\Sigma\right)$ approximately!

  Note:
  If the sample is large everything is Gaussian? NO! If the sample is large, then the sample mean has a distribution which can be approximated by a Gaussian.

  Therefore 1 billion coin tosses remain coin tosses: the sample mean of a billion Bernoulli had a distribution that is a Gaussian.

  So if the sample is large we need to check for Gaussianity!

8 Lecture 13: 30th Of March 2020

Inference for the mean μ

We want to use data, which is a partial information about the true population, to infer something about the true population! Assume our information from the data comes from: $\mathbb{R}^p \ni \underline{X}_1, \dots, \underline{X}_n$ random vectors independent and identically distributed, with $\mathbb{E}[\underline{X}_1] = \underline{\mu}$ and $Cov(\underline{X}_1) = \Sigma$ with $det(\Sigma) > 0$

Curse of dimensionality: as p gets larger we need larger and larger amounts of data, indeed n must grow exponentially fast with respect to p

$\bar{\underline{X}} = \frac{1}{n}\sum_{i=1}^{n} \underline{X}_i$ is the sample mean and is the point estimator for $\underline{\mu}$ What is the uncertainty related with this estimate given by the estimator?

Case 1: n is large: $n >> p$

We know that the **CLT** holds: $\sqrt{n}(\bar{\underline{X}} - \underline{\mu}) \sim \mathcal{N}_p(\underline{0}, \Sigma)$ approximately.

Then: $(\sqrt{n}(\bar{\underline{X}} - \underline{\mu}) - \underline{0})^T \Sigma^{-1} (\sqrt{n}(\bar{\underline{X}} - \underline{\mu}) - \underline{0}) \sim \chi^2(p)$ Thus: $n(\bar{\underline{X}} - \underline{\mu})^T \Sigma^{-1} (\bar{\underline{X}} - \underline{\mu}) \sim \chi^2(p)$

We don't know Σ so its not enough for inference on $\underline{\mu}$!

But $S \to \Sigma$ as $n \to \infty$ by **LLN**, so since n is large we assume it holds: Thus, for large n, we have:

$$n(\bar{\underline{X}} - \underline{\mu})^T S^{-1} (\bar{\underline{X}} - \underline{\mu}) \sim \chi^2(p)$$

Now: the only unknown part is the mean which is want we want to estimate! Therefore the expression above is a **pivotal statistics**: it acts as a pivot around which we build up the inferential procedure. Note that for a random variable to be a **pivotal statistics** we need know its distribution without knowing the value of μ (which is unknown since we want to estimate it!).

From the above we have that:

$$\mathbb{P}[n(\bar{\underline{X}} - \underline{\mu})^T S^{-1} (\bar{\underline{X}} - \underline{\mu}) \leq \chi^2_\alpha(p)] = 1 - \alpha \text{ if } \alpha \in (0,1) \implies \mathbb{P}[d^2_{S^{-1}}(\bar{\underline{X}}, \underline{\mu}) \leq \chi^2_\alpha(p)] = 1 - \alpha$$

The equation in brackets identifies an ellipse, whose radius depends on α, and it is centred in $\underline{\mu}$:

$$\mathcal{E}^\alpha_{S^{-1}}(\underline{\mu}) = \{\underline{x} \in \mathbb{R}^p : d^2_{S^{-1}}(\bar{\underline{X}} - \underline{\mu}) < \chi^2_\alpha(p)\}$$

We can also define the ellipse centred in $\bar{\underline{X}}$:

$$\mathcal{E}^\alpha_{S^{-1}}(\bar{\underline{X}}) = \{\underline{\eta} \in \mathbb{R}^p : d^2_{S^{-1}}(\underline{\eta}, \bar{\underline{X}}) \leq \chi^2_\alpha(p)\}$$

The two ellipses have the same axes, the same lengths of the semi-axis but a different centre:

$$\bar{\underline{X}} \in \mathcal{E}^\alpha_{S^{-1}}(\underline{\mu}) \iff \underline{\mu} \in \mathcal{E}^\alpha_{S^{-1}}(\bar{\underline{X}}) \text{ indeed: } d(\bar{\underline{X}}, \underline{\mu}) = d(\underline{\mu}, \bar{\underline{X}})$$

Thus: $\mathbb{P}[d^2_{S^{-1}}(\bar{\underline{X}}, \underline{\mu}) \leq \chi^2_\alpha(p)] = 1 - \alpha$ is equivalent to $\mathbb{P}[\bar{\underline{X}} \in \mathcal{E}^\alpha_{S^{-1}}(\underline{\mu})] = 1 - \alpha = \mathbb{P}[\underline{\mu} \in \mathcal{E}^\alpha_{S^{-1}}(\bar{\underline{X}})]$ where:

- In the first case $\bar{\underline{X}}$ is random and the ellipse is given, but we don't know it since we don't know $\underline{\mu}$
- In the second case $\underline{\mu}$ is not known but its given while the ellipse is random (randomly generated once we have observed the data). Moreover we have that $1 - \alpha$ times the random ellipse will cover $\underline{\mu}$

  Since the random ellipse will cover $\underline{\mu}$ with a probability $1 - \alpha$ what is the confidence region of level $1 - \alpha$ for $\underline{\mu}$? Well:

$$CR_{1-\alpha}(\underline{\mu}) = \{\underline{\eta} \in \mathbb{R}^p : \underline{\eta} \in \mathcal{E}^\alpha_{S^{-1}}(\bar{\underline{X}})\} = \{\underline{\eta} \in \mathbb{R}^p : n(\underline{\eta} - \bar{\underline{X}})^T S^{-1} (\underline{\eta} - \bar{\underline{X}}) \leq \chi^2_\alpha(p)\}$$

Thus from the point above we have expression for the confidence region around the point estimate for mean: this also quantifies how uncertain we are about this value.

Indeed the more ellipse is dense and shrinked around its centre the less is uncertain: the uncertainty is given by $det(S)$ as it represent the volume of the ellipse!

Since we have a pivotal quantity we can also do **Testing:**

Suppose that: $H_0 : \underline{\mu} = \underline{\mu}_0 vs H_1 : \underline{\mu} \neq \underline{\mu}_0$ where $\underline{\mu}_0$ is given and known, and where H_1 is the alternative hypothesis.

Fix now a level $\alpha \in (0, 1)$ (usually small): we want to use the data to decide among the two hypothesis.

- We can reject H_0 when its true: this is called **Type I Error**. With α we can control the **Type I Error**
- We can reject H_1 when its true: this is called **Type II Error**

Most of the time H_0 is the assumption we want to reject!

Note: this is not the case for the Shapiro test!

Consider now the pivotal statistic: $T_0^2 = n(\underline{\bar{X}} - \underline{\mu}_0)^T S^{-1}(\underline{\bar{X}} - \underline{\mu}_0)$

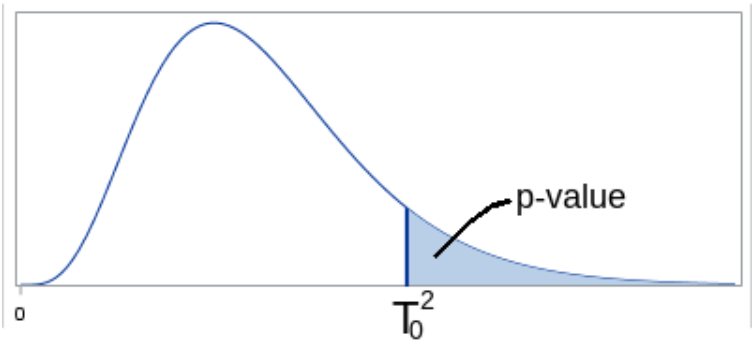
If H_0 is true: $\underline{\mu} = \underline{\mu}_0$ so that $T_0^2 \sim \chi^2(p)$

When do we have proof against H_0? When $\underline{\bar{X}}$ is very far from $\underline{\mu}_0$! Thus we reject H_0 if $T_0^2 > \chi^2_\alpha(p)$ so we only make an error α percent of times!

The idea is the following: if we see today something that happens 1 time out of 100000 times then: either the assumption is wrong or we are observing a miracle. But statistics doesn't believe in mircales and so we believe that the assumption H_0 is wrong!

Therefore the **Rejection Region** of level α is given by $\{T_0^2 > \chi^2_\alpha(p)\}$

Consider the following figure:



We can see that the $p-$value is on the right of T_0^2 therefore: $p - \text{value} \leq \alpha \iff T_0^2 > \chi^2_\alpha(p)$ Then:

- If the $p-$value is very small, it means we can reject H_0
- If the $p-$ value is very large, it means we cannot reject H_0

Thus, after we perform the test, we communicate the $p-$value we obtained and then we decide if its small enough to reject H_0

Case 2: n is small

All the above is based on knowing the distribution of a pivotal statistics! How do we know it if n is small?

We could transform the data into more Gaussian data but sometimes we cant do it!

Suppose that $\underline{X}_i \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}, \Sigma)$ with $det(\Sigma) > 0$ Then what is the distribution of $n(\bar{\underline{X}} - \underline{\mu})^T S^{-1}(\bar{\underline{X}} - \underline{\mu})$?

Definition: Suppose $Y \sim \chi^2(n)$ and $W \sim \chi^2(m)$ If $Y \perp\!\!\!\perp W$, that is they are stochastically independent, then: $\frac{Y/n}{W/m} \sim F(n, m)$ which is the **Fisher(-Snedecor) Distribution.**

Note: If $t \sim t(n)$ T-student distribution then: $t = \frac{Z}{\sqrt{\frac{W}{n}}}$ where:

- $Z \sim N(0, 1)$
- $W \sim \chi^2(n)$
- Z, W are stochastically independent: $Z \perp\!\!\!\perp W$

Thus: $t^2 = \frac{Z^2/1}{W/n} \sim F(1, n)$ since $Z^2 \sim \chi^2(1)$ and $Z \perp\!\!\!\perp W \implies Z^2 \perp\!\!\!\perp W$

Convergence in distribution: $F(n, m) \to \frac{1}{n}\chi^2(n)$ when $m \to \infty$ indeed:

$F(n, m) = \frac{Y/n}{W/m}$ but $W \sim \chi^2(m)$ so $W = \sum_{i=1}^{m} Z_i^2$ with $Z_1, \ldots, Z_m \overset{iid}{\sim} N(0, 1)$ thus:

$$\mathbb{E}[Z_1^2] = Var[Z_1] = 1 \implies \frac{1}{n}\sum_{i=1}^{m} Z_i^2 \to 1 \text{ as } m \to \infty \text{ by LLN}$$

Therefore: $\frac{W}{m} \to 1$ as $m \to \infty$ and thus: $\frac{Y/n}{W/n} \to \frac{Y}{n}$ as $m \to \infty \implies \frac{Y}{n} \sim \frac{1}{n}\chi^2(n)$ as $m \to \infty$

Hotelling's Theorem: If $\underline{X} \sim \mathcal{N}_p(\underline{\mu}, \Sigma)$ with $det(\Sigma) > 0$ and If $W \sim Wishart\left(\frac{1}{m}\Sigma, m\right)$ where: Σ is a $p \times p$ matrix.

Suppose that $\underline{X} \perp\!\!\!\perp W$ then: $\frac{m - p + 1}{mp}(\underline{X} - \underline{\mu})^T W^{-1}(\underline{X} - \underline{\mu}) \sim F(p, m - p + 1)$

Corollary: Suppose $\underline{X}_1, \ldots, \underline{X}_n \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}, \Sigma)$ with $det(\Sigma) > 0$ Then:

$$n(\bar{\underline{X}} - \underline{\mu})^T S^{-1}(\bar{\underline{X}} - \underline{\mu}) \sim \frac{(n-1)p}{n-p} F(p, n - p)$$

Proof of the above Corollary: $\sqrt{n}(\bar{\underline{X}} - \underline{\mu}) \sim \mathcal{N}_p(\underline{0}, \Sigma)$ Note this is not CLT but its an exact result.

We know that $S \sim Wishart\left(\frac{1}{n-1}\Sigma, n - 1\right)$ and $\bar{\underline{X}} \perp\!\!\!\perp S$ Thus, we can use Hotelling's Theorem with $m = n - 1$ so:

$$\frac{n - 1 - p + 1}{(n-1)p} n(\bar{\underline{X}} - \underline{\mu})^T S^{-1}(\bar{\underline{X}} - \underline{\mu}) \sim F(p, n - 1 - p + 1) \implies \frac{n-p}{(n-1)p} n(\bar{\underline{X}} - \underline{\mu})^T S^{-1}(\bar{\underline{X}} - \underline{\mu}) \sim F(p, n - p)$$

which is the thesis. More synthetically we write it as: $n(\bar{\underline{X}} - \underline{\mu})^T S^{-1}(\bar{\underline{X}} - \underline{\mu}) \sim \frac{(n-1)p}{n-p} F(p, n - p)$

Definition: $T^2 = n(\bar{\underline{X}} - \underline{\mu})^T S^{-1}(\bar{\underline{X}} - \underline{\mu})$ is **Hotelling's T^2 Statistic**

Confidence regions: Now our pivotal quantity is **Hotelling's T^2 Statistic** Then:

$$\mathbb{P}\left[n(\bar{\underline{X}}-\underline{\mu})^T S^{-1}(\bar{\underline{X}}-\underline{\mu}) \leq \frac{(n-1)p}{n-p}F_\alpha(p,n-p)\right] = 1-\alpha = \mathbb{P}\left[d^2_{S^{-1}}(\bar{\underline{X}}-\underline{\mu}) \leq \frac{(n-1)p}{n-p}F_\alpha(p,n-p)\right]$$

Thus: $\begin{cases} \mathbb{P}[\underline{\mu} \in \mathcal{E}^\alpha_{S^{-1}}(\bar{\underline{X}})] = 1-\alpha \text{ with } \alpha \in (0,1) \\ \mathcal{E}^\alpha_{S^{-1}}(\bar{\underline{X}}) = \left\{\underline{\eta} \in \mathbb{R}^p : d^2_{S^{-1}}(\bar{\underline{X}}-\underline{\mu}) \leq \frac{(n-1)p}{n-p}F_\alpha(p,n-p)\right\} \end{cases} \implies CR_{1-\alpha}(\underline{\mu}) = \mathcal{E}^\alpha_{S^{-1}}(\bar{\underline{X}})$

Note that the radius of the ellipse here is different from the one found in the previous case in which we had **Large** n

Observation: What if n is large and we have Gaussian sample? Well:

$$(n-1)p\frac{1}{n-p}F_\alpha(p,n-p) \to p\frac{1}{p}\chi^2(p) = \chi^2(p) \text{ as } n \to \infty$$

because $(n-1)\frac{1}{n-p} \to 1$ and $F_\alpha(p,n-p) \to \frac{1}{p}\chi^2(p)$ Therefore we get same result as when n is large and we have no Gaussian assumption!

Hypothesis Testing: $H_0 : \underline{\mu} = \underline{\mu}_0$ vs $H_1 : \underline{\mu} \neq \underline{\mu}_0$

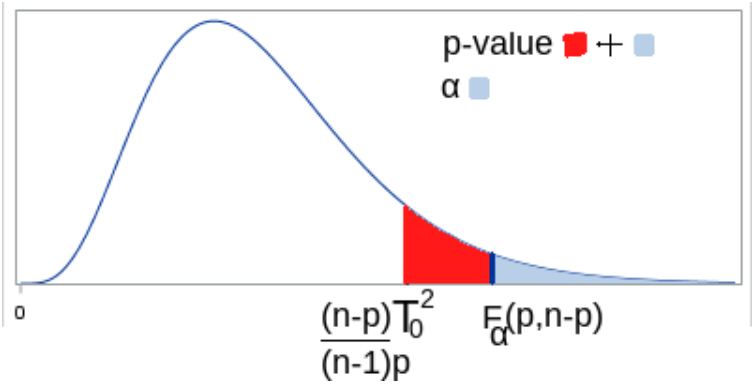
At level $\alpha \in (0,1)$ we have: $T_0^2 = n(\bar{\underline{X}}-\underline{\mu}_0)^T S^{-1}(\bar{\underline{X}}-\underline{\mu}_0)$ which is Hotelling's T^2 statistics when H_0 is true.

If H_0 is true then $T_0^2 \sim \frac{(n-1)p}{n-p}F(p,n-p)$ So If H_0 is true we reject when values of T_0^2 are large.

Indeed as always if it is very rare to see this value for T_0^2 when H_0 is true we reject H_0

Conclusion: We reject H_0 if $T_0^2 > (n-1)p\frac{1}{n-p}F_\alpha(p,n-p)$

Attention to the $p-$value here: we are multiplying the quantile by $\frac{(n-1)p}{n-p}$



1) How do the Confidence Interval and the Confidence Region change following correlations among variables?
2) There is a temptation to resort to to tools of Stat101 of uni-variate statistics in order to make inference on a multivariate mean.

   As any temptation should be avoided, but we can work on it to take out some good things!

For instance consider the following **Example:** Suppose that $\underline{X}_1, \ldots, \underline{X}_{10} \overset{iid}{\sim} \mathcal{N}_2(\underline{\mu}, \Sigma)$

Suppose $\bar{\underline{X}} = \underline{0}$ and $S = I = diag(1,1)$ Here $n = 10, p = 2$ Then:

$CR_{1-\alpha}(\mu) = \{\underline{\eta} \in \mathbb{R}^2 : 10(\underline{\eta} - \bar{\underline{X}})^T S^{-1}(\underline{\eta} - \bar{\underline{X}}) \leq 9 \times 2\frac{1}{8} F_\alpha(2,8)\} = \{\underline{\eta} \in \mathbb{R}^2 : 10\underline{\eta}^T I \underline{\eta} \leq 9 \times 2\frac{1}{8} F_\alpha(2,8)\}$

If we e choose $\alpha = 0.1 \implies 1 - \alpha = 0.9 \implies F_{0.1}(2,8) = 3.11 \implies CR_{0.9}(\underline{\mu}) = \{\underline{\eta} \in \mathbb{R}^2 : 10(\eta_1^2 + \eta_2^2) \leq 6.997\}$ where $\underline{\eta} = (\eta_1, \eta_2)^T$

Thus: $CR_{0.9}(\underline{\mu}) = \{\underline{\eta} \in \mathbb{R}^2 : \eta_1^2 + \eta_2^2 \leq 0.6997\}$

So the $CR_{0.9}(\underline{\mu})$ is a circle centred in 0 and with a squared radius equal to 0.6997

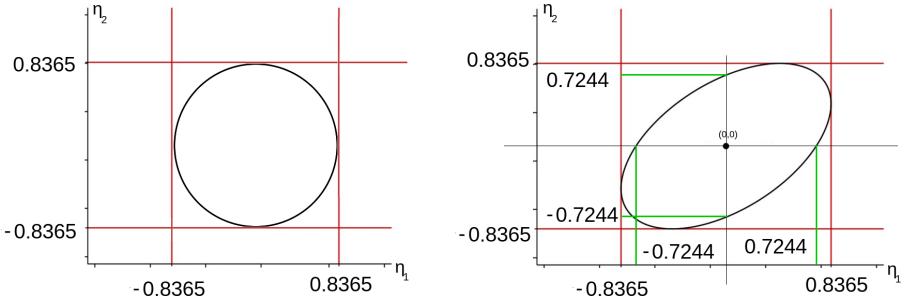
This means that the random ellipse produced this circle, but we can't say if $\underline{\mu}$ is within this circle with probability 0.9! What we can say is that with probability 0.9 the above algorithm produces a random ellipse which includes $\underline{\mu}$

Now we change S: we assume that $\bar{\underline{X}} = \underline{0}$ and that $S = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$ So now X_1 and X_2 are correlated!

Thus: $CR_{0.9}(\underline{\mu}) = \{\underline{\eta} \in \mathbb{R}^2 : 10\underline{\eta}^T S^{-1} \underline{\eta} \leq 6.997\}$ And now: $S^{-1} = \begin{bmatrix} \frac{4}{3} & -\frac{2}{3} \\ -\frac{2}{3} & \frac{4}{3} \end{bmatrix}$ Thus: $10\underline{\eta}^T S^{-1} \underline{\eta} = 10\frac{4}{3}(\eta_1^2 - \eta_1\eta_2 + \eta_2^2)$

Therefore: $CR_{0.9}(\underline{\mu}) = \{\underline{\eta} \in \mathbb{R}^2 : \eta_1^2 - \eta_1\eta_2 + \eta_2^2 \leq \frac{3}{4} 0.6997\}$ which is an ellipse centred in zero.

Thus we can see that the confidence region changed just by changing the co-variance: the more the two components are correlated the more this ellipse will be squished!



Where on the left is the first case in which S is diagonal and the figure on the right is the case with the new S

Therefore the uncertainty on $\underline{\mu}$ is changing but not because we have different point estimate of $\bar{\underline{X}}$ but only because the co-variance is changing!

Note: The above observation is something we read on the estimator of the co-variance not on the one of the mean!

Now from Stat101 we remember that $\bar{\underline{X}} = (\bar{X}_1, \bar{X}_2)^T$ and $S = [s_{ij}]$ with $s_{11} = s_{22} = 1$

Back then we would compute the confidence interval for the single components:

- $CI_{0.9}(\mu_1) = \left[\bar{X}_1 \pm t_{1-\alpha/2}(n-1)\sqrt{\frac{s_{11}}{n}}\right] = \left[\bar{X}_1 \pm t_{0.95}(9)\sqrt{1/10}\right] = [\pm 0.5796]$ since $\bar{\underline{X}} = 0$
- The Confidence interval for μ_2 is the same since everything is symmetric: $IC_{0.9}(\mu_2) = [\pm 0.5796]$

But now we didn't use the co-variance to compute these intervals so the two above don't change if we change variance! They are the same for the two different cases of S that we presented above!

Thus the tools of Stat101 can't work in a multi-variate setting! Note there is a temptation to say: $IC(\underline{\mu}) = IC(\mu_1) \times IC(\mu_2)$ **NO! This is rubbish!**

Indeed: $P[\mu_1 \in CI_{0.9}(\mu_1), \mu_2 \in CI_{0.9}(\mu_2)] = (0.9)^2 < 0.9$ if $\bar{X}_1 \perp\!\!\!\perp \bar{X}_2$ and therefore the Cartesian product of the two intervals cannot generate a region that has the same confidence!!

Note: If we say we can use CLT if $n = 100$ and $p = 1$ then we need $n = (100)^p$ for generic p!

9 Lecture 14: 31st Of March 2020

Note: $\sqrt{n}(\bar{\underline{X}} - \underline{\mu}) \sim \mathcal{N}_p(\underline{0}, \Sigma) \implies \bar{\underline{X}} \sim \mathcal{N}_p\left(\underline{\mu}, \frac{1}{n}\Sigma\right)$ then the *Mahalanobis Distance* induced by $\frac{1}{n}\Sigma$ is:

$$d^2_{(\Sigma/n)^{-1}}(\bar{\underline{X}}, \underline{\mu}) = n(\bar{\underline{X}} - \underline{\mu})^T \Sigma^{-1} (\bar{\underline{X}} - \underline{\mu})$$

Confidence Intervals for Linear Combinations of the Mean

Suppose again we are in the case in which the **number of samples** n **is small**! Suppose that $\underline{X}_1, \dots, \underline{X}_n \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}, \Sigma)$

Hence: $\bar{\underline{X}} = \frac{1}{n}\sum_{i=1}^{n} \underline{X}_i$ is an estimator for $\underline{\mu}$

One at-the-time confidence interval for μ

What is the uncertainty when we make inference on one linear combination, of the mean, at the time? Let $\underline{a} \in \mathbb{R}^p$ Then: an estimator for $\underline{a}^T\underline{\mu}$ is given by $\underline{a}^T\bar{\underline{X}}$ which is unbiased and its also the MLE since we are working with Gaussian Variables!

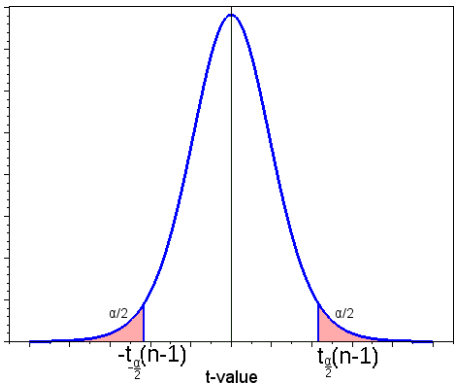
Since all components are Gaussian: $\mathbb{R} \ni \underline{a}^T\bar{\underline{X}} \sim \mathcal{N}_1\left(\underline{a}^T\underline{\mu}, \frac{1}{n}\underline{a}^T\Sigma\underline{a}\right) \implies \frac{\underline{a}^T\bar{\underline{X}} - \underline{a}^T\underline{\mu}}{\sqrt{\underline{a}^T\Sigma\underline{a}}}\sqrt{n} \sim \mathcal{N}_1(0,1)$

We know that $(n-1)S \sim Wishart(\Sigma, n-1) \implies (n-1)\underline{a}^T S\underline{a} \sim (\underline{a}^T\Sigma\underline{a})\chi^2(n-1)$ Moreover: $\bar{\underline{X}} \perp\!\!\!\perp S$ Hence:
$\frac{\underline{a}^T\bar{\underline{X}} - \underline{a}^T\underline{\mu}}{\sqrt{\underline{a}^T\Sigma\underline{a}}}\sqrt{n} \sim \mathcal{N}(0,1)$ and $\sqrt{\frac{(n-1)\underline{a}^T S\underline{a}}{(n-1)\underline{a}^T\Sigma\underline{a}}} \sim \sqrt{\frac{1}{n-1}\chi^2(n-1)}$

The division of the two above give us: $\frac{\underline{a}^T\bar{\underline{X}} - \underline{a}^T\underline{\mu}}{\sqrt{\underline{a}^T S\underline{a}}}\sqrt{n} \sim t(n-1)$ which is a pivotal statistics!

Thus: $\mathbb{P}\left[\frac{|\underline{a}^T\underline{X} - \underline{a}^T\underline{\mu}|}{\sqrt{\underline{a}^T S\underline{a}}} < t_{\alpha/2}(n-1)\right] = 1-\alpha$ for $\alpha \in (0,1)$ Therefore:

$$\mathbb{P}\left[\underline{a}^T\underline{\mu} \in \left[\underline{a}^T\bar{\underline{X}} \pm t_{\alpha/2}(n-1)\sqrt{\underline{a}^T S\underline{a}}\frac{1}{\sqrt{n}}\right]\right] = 1-\alpha \implies CI_{1-\alpha}(\underline{a}^T\underline{\mu}) = \left[\underline{a}^T\bar{\underline{X}} \pm t_{\alpha/2}(n-1)\sqrt{\underline{a}^T S\underline{a}}\frac{1}{\sqrt{n}}\right]$$



Example: If $\underline{a} = (0, 0, \dots, 0, \underbrace{1}_{i-th \text{ position}}, 0, \dots, 0)^T$ Then:

$$\underline{a}^T\underline{\mu} = \mu_i \implies CI_{1-\alpha}(\mu_i) = \left[\bar{X}_i \pm t_{\alpha/2}(n-1)\sqrt{\frac{S_{ii}}{n}}\right]$$

Example:
$a = (0, 0, \dots, 0, \underbrace{1}_{i-th \text{ position}}, 0, \dots, 0, \underbrace{-1}_{j-th \text{ position}}, 0, \dots, 0)'$ Then:

$$\underline{a}^T\underline{\mu} = \mu_i - \mu_j \implies CI_{1-\alpha}(\mu_i - \mu_j) = \left[\bar{X}_i - \bar{X}_j \pm t_{\alpha/2}(n-1)\sqrt{(s_{ii} - s_{ij} + s_{jj})\frac{1}{n}}\right]$$

So we can now build confidence intervals for any linear combination of the components of μ We proved that:

$$\mathbb{P}\left[\underline{a}^T\underline{\mu} \in \left[\underline{a}^T\bar{\underline{X}} \pm t_{\alpha/2}(n-1)\sqrt{\underline{a}^T S\underline{a}\frac{1}{n}}\right]\right] = 1-\alpha \ \forall \underline{a} \in \mathbb{R}^p$$

Which are **one-at-the-time** $CI_{1-\alpha}(\underline{a}^T\underline{\mu})$

Remark: Having built confidence intervals we can also do testing now, for example:

- $H_0 : \underline{a}^T\underline{\mu} = \delta_0$ vs $H_1 : \underline{a}^T\underline{\mu} > \delta_0$ Then:

  If H_0 is true: $\frac{\underline{a}^T\bar{\underline{X}} - \underline{a}^T\delta_0}{\sqrt{\underline{a}^T S\underline{a}}}\sqrt{n} \sim t(n-1)$ Thus we reject at level $\alpha \in (0,1)$ if: $\frac{\underline{a}^T\bar{\underline{X}} - \delta_0}{\sqrt{\underline{a}^T S\underline{a}}}\sqrt{n} > t_\alpha(n-1)$

  Note that in this case we don't have symmetric quantiles like in the last figure of the previous page, but we have quantile $t_\alpha(n-1)$ which has shaded area equal to α on the right! So it's like the figure above but only on the right and with α and not $\alpha/2$

- If $H_0 : \underline{a}^T\underline{\mu} = \delta_0$ vs $H_1 : \underline{a}^T\underline{\mu} \neq \delta_0$ Then we reject at level $\alpha \in (0,1)$ if: $\frac{|\underline{a}^T\bar{\underline{X}} - \delta_0|}{\sqrt{\underline{a}^T S\underline{a}}}\sqrt{n} > t_{\alpha/2}(n-1)$

  Note that in this case we have symmetric quantiles like in the last figure of the previous page.

simulatenous confidence interval for linear combinations of the mean

In the above **we did NOT** prove that:

$$\mathbb{P}\left[\underline{a}^T\underline{\mu} \in \left[\underline{a}^T\bar{\underline{X}} \pm t_{\alpha/2}(n-1)\sqrt{\underline{a}^T S\underline{a}\frac{1}{n}}, \forall \underline{a} \in \mathbb{R}^p\right]\right] = 1-\alpha$$

That's totally wrong! We didn't prove that all of these CI together cover any combination of μ! What's wrong is $t_{\alpha/2}(n-1)$: we need to substitute this in order to find the simultaneous confidence interval!

Example:

- One-at-the-time Confidence Interval: $\mathbb{P}(\text{coin = head}) = \frac{1}{2}\ \forall k$ coins tosses
- Simultaneous Confidence Interval: $\mathbb{P}(\text{coin = head } \forall k \text{ coin tosses}) = \left(\frac{1}{2}\right)^k = (0.5)^k$

We have the following situation to solve:

$$\mathbb{P}\left[\underline{a}^T\underline{\mu} \in \left[\underline{a}^T\bar{\underline{X}} \pm \star\star\sqrt{\underline{a}^T S\underline{a}\frac{1}{n}},\ \forall \underline{a} \in \mathbb{R}^p\right]\right] = 1-\alpha$$

we want to find $\star\star$ such that the statement is true! We want, for any possible linear combination, the overall confidence interval of level $1-\alpha$

Excursus on Linear Algebra

Suppose $\underline{b}, \underline{d} \in \mathbb{R}^p$ then: $\frac{<\underline{b},\underline{d}>}{\|\underline{b}\|\|\underline{d}\|} = cos(\theta)$ where θ is the angle between the two vectors. Now take the square:

$$\left(\frac{<\underline{b},\underline{d}>}{\|\underline{b}\|\|\underline{d}\|}\right)^2 = cos^2(\theta) \implies \left(\frac{<\underline{b},\underline{d}>}{\|\underline{b}\|\|\underline{d}\|}\right)^2 \leq 1$$

Equality holds if: $\underline{b} \in Span(\underline{d})$ that is if: $\underline{b} \propto \underline{d}$ Hence: $(\underline{b}^T\underline{d})^2 \leq \left(\sum_i b_i^2\right)\left(\sum_i d_i^2\right)$ Equality holds if $\underline{b} \in Span(\underline{d})$

Recall now the Cauchy-Schwarz Inequality: $\int fg \leq \left(\int f^2\right)\left(\int g^2\right)$

Note: here we do all statistics in $\mathbb{R}^p$ but it can be transferred to any Hilbert Space! Thus functional data analysis comes out naturally!

Now take B a $p \times p$ positive definite matrix. We want to prove that: $(\underline{b}^T\underline{d})^2 \leq (\underline{b}^T B\underline{b})(\underline{d}^T B^{-1}\underline{d})$ which is the **Extended Cauchy-Schwarz Inequality**.

Proof of the Extended Cauchy-Schwarz Inequality: Since B is positive definite we can take it's inverse. Then:

$\underline{b}^T\underline{d} = \underline{b}^T B^{1/2}B^{-1/2}\underline{d} \implies (\underline{b}^T\underline{d})^2 = (\underline{b}^T B^{1/2}B^{-1/2}\underline{d})^2 \leq (\underline{b}^T B^{1/2}B^{1/2}\underline{b})(\underline{d}^T B^{-1/2}B^{-1/2}\underline{d})$ which is the thesis.

Note that in the last passage we applied to each of the two term the regular Cauchy-Schwarz Inequality.

In the above equality holds if $\underline{b}^T B^{1/2} \in Span(B^{-1/2}\underline{d}) \iff \underline{b} \in Span(B^{-1}\underline{d})$

Proposition: Maximum Lemma Let B be a $p \times p$ positive definite matrix. If $\underline{d} \in \mathbb{R}^p$ Then:

$$\max_{\underline{x}\in\mathbb{R}^p:\underline{x}\neq\underline{0}} \frac{(\underline{x}^T\underline{d})^2}{\underline{x}^T B\underline{x}} = \underline{d}^T B^{-1}\underline{d}$$

Proof of Maximum Lemma: $(\underline{x}^T\underline{d})^2 \leq (\underline{x}^T B\underline{x})(\underline{d}^T B^{-1}\underline{d})$

If $\underline{x} \neq \underline{0}$ then: $\underline{x}^T B\underline{x} > 0$ since B is positive definite. Therefore: $\frac{(\underline{x}^T\underline{d})^2}{\underline{x}^T B\underline{x}} \leq \underline{d}^T B^{-1}\underline{d}\ \forall \underline{x} \neq \underline{0}$

Equality holds if $\underline{x} \in Span(B^{-1}\underline{d})$ and thus we have the thesis.

Let $\underline{X}_1, \ldots, \underline{X}_n \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}, \Sigma)$ and $\bar{\underline{X}} = \frac{1}{n}\sum_{i=1}^n \underline{X}_i$ Let $\underline{a} \in \mathbb{R}^p$ then: $\frac{(\underline{a}^T(\bar{\underline{X}} - \underline{\mu}))^2}{\underline{a}^T S\underline{a}} n$ by the Maximum Lemma we know:

$$\max_{\underline{a}\in\mathbb{R}^p} \frac{(\underline{a}^T(\bar{\underline{X}} - \underline{\mu}))^2}{\underline{a}^T S\underline{a}} n = n(\bar{\underline{X}} - \underline{\mu})^T S^{-1}(\bar{\underline{X}} - \underline{\mu}) = T^2 \sim (n-1)p\frac{1}{n-p}F(p, n-p)$$

Where we used the Corollary to Hotelling's Theorem! Thus:

$$\mathbb{P}\left[|\underline{a}^T(\bar{\underline{X}} - \underline{\mu})|\frac{1}{\sqrt{\underline{a}^T S\underline{a}}}\sqrt{n} \leq c, \forall \underline{a} \in \mathbb{R}^p\ \underline{a} \neq \underline{0}\right] = 1 - \alpha \text{ where } c \text{ is unknown}$$

The above probability is the same as computing: $\mathbb{P}\left[\max_{\underline{a}\in\mathbb{R}^p\ \underline{a}\neq\underline{0}} \frac{|\underline{a}^T(\bar{\underline{X}} - \underline{\mu})|}{\sqrt{\underline{a}^T S\underline{a}}}\sqrt{n} \leq c\right] = 1 - \alpha$

Now we take the square and get:

$$\mathbb{P}\left[\max_{\underline{a}\in\mathbb{R}^p\ \underline{a}\neq\underline{0}} \left(\frac{\underline{a}^T(\bar{\underline{X}} - \underline{\mu})}{\sqrt{\underline{a}^T S\underline{a}}}\sqrt{n}\right)^2 \leq c^2\right] = 1 - \alpha \implies \mathbb{P}\left[(n-1)\frac{p}{n-p}F(p, n-p) \leq c^2\right] = 1 - \alpha \implies$$

$$c^2 = (n-1)\frac{p}{n-p}F_\alpha(p, n-p) \implies \mathbb{P}\left[|\underline{a}^T(\bar{\underline{X}} - \underline{\mu})|\frac{\sqrt{n}}{\sqrt{\underline{a}^T S\underline{a}}} \leq \sqrt{(n-1)\frac{p}{n-p}F_\alpha(p, n-p)}\ \forall \underline{a} \in \mathbb{R}^p\right] = 1 - \alpha$$

Thus for Simultaneous Confidence Interval the right quantile is the square root of an F distribution and not of a $t-$student distribution!

So the Simultaneous Confidence Interval is given by:

$$SimCI_{1-\alpha}(\underline{a}^T\underline{\mu}) = \left[\underline{a}^T\bar{\underline{X}} \pm \sqrt{(n-1)\frac{p}{n-p}F_\alpha(p,n-p)}\sqrt{\underline{a}^T S\underline{a}\frac{1}{n}}\right]$$

Thus now can take all the possible linear combination we want and we get a confidence interval that is globally correct $(1-\alpha)\%$ of times! Indeed all of the interval are correct with probability $1-\alpha$: all of the interval cover the linear combination $(1-\alpha)\%$ of the time they are used!

So if we have an ellipse of the $CR_{1-\alpha}(\underline{\mu})$ then its projection along any direction $\underline{a}$ gives us the simultaneous confidence interval $SimCI_{1-\alpha}(\underline{a}^T\underline{\mu})$

Note: The Simultaneous Confidence Intervals are also called Scheffe Confidence Intervals, and also T^2 Confidence Intervals!

Note: Simultaneous Confidence Intervals are the linear envelope of the confidence region!

Bonferroni's Method for Simultaneous Confidence Intervals for a finite number of linear combinations of μ
Now we would like an an in between: we want confidence intervals for a fixed number of linear combinations we have specified in advance. This is because the Simultaneous Confidence Intervals we have seen above cover all possible linear combination, but due to this they are very large. So if we fix a finite number of linear combinations we get something smaller!

Given $\underline{a}_1,\ldots,\underline{a}_k \in \mathbb{R}^p$ we want to find $CI(\underline{a}_1^T\underline{\mu}),\ldots,CI(\underline{a}_k^T\underline{\mu})$ with a simultaneous confidence of $1-\alpha$ where: $\alpha \in (0,1)$

Note: If k is too big the Bonferroni's Confidence Interval is not so better (smaller) than the general Simultaneous Confidence Interval!

Let $CI_{1-\alpha}(\underline{a}^T\underline{\mu})$ be a One-at-the-time Confidence Interval, for $\underline{a}^T\underline{\mu}$, of level $1-\alpha$ For example:

$$CI_{1-\alpha}(\underline{a}^T\underline{\mu}) = \left[\underline{a}^T\bar{\underline{X}} \pm t_{\alpha/2}(n-1)\sqrt{\frac{\underline{a}^T S\underline{a}}{n}}\right]$$

Then:

$$\mathbb{P}\left[\bigcap_{i=1}^k\{\underline{a}_i^T\underline{\mu} \in CI_{1-\alpha}(\underline{a}_i^T\underline{\mu})\}\right] = 1-\mathbb{P}\left[\bigcup_{i=1}^k\{\underline{a}_i^T\underline{\mu} \notin CI_{1-\alpha}(\underline{a}_i^T\underline{\mu})\}\right] \geq$$

$$\geq 1-\sum_{i=1}^k \mathbb{P}\left[\underline{a}_i^T\underline{\mu} \notin CI_{1-\alpha}(\underline{a}_i^T\underline{\mu})\right] = 1-\sum_{i=1}^k \alpha = 1-k\alpha$$

The above follows from the fact that: $P(A\cup B) = P(A)+P(B)-P(A\cap B) \leq P(A)+P(B)$ The inequality is called Bonferroni's Inequality!

$$\text{Hence: } \mathbb{P}\left[\bigcap_{i=1}^k\{\underline{a}_i^T\underline{\mu} \in CI_{1-\alpha/k}(\underline{a}_i^T\underline{\mu})\}\right] \geq 1-\alpha$$

So for each linear combination we take a confidence interval of level $1-\alpha/k$ so that the overall confidence interval is of level $1-\alpha$

Conclusion: The **Bonferroni's Simultaneous Confidence Interval** $\underline{a}_1^T\underline{\mu},\ldots,\underline{a}_k^T\underline{\mu}$ is given by:

$$BonfCI_{1-\alpha}(\underline{a}_i^T\underline{\mu}) = \left[\underline{a}_i^T\bar{\underline{X}} \pm t_{\alpha/2k}(n-1)\sqrt{\frac{\underline{a}^T S\underline{a}}{n}}\right]$$

We can see that if $k = \infty$ the confidence interval is $[-\infty, \infty]$ so Bonferroni's Confidence Intervals only work with small finite number of linear combinations!

We have that the Bonferroni's Simultaneous Confidence Interval is larger than the One-at-the-time Confidence Interval but smaller than the Simultaneous Confidence Interval!

Simultaneous Testing with the Bonferroni's Simultaneous Confidence Interval can be done but since it's very conservative (very strict) it is not used in practice, because with big data we would never come to reject the null hypothesis!

Anyway Simultaneous Testing Bonferroni's way can be done as follows:
If $H_0 : \underline{a}_1^T \underline{\mu} = \delta_1 \wedge \cdots \wedge \underline{a}_k^T \underline{\mu} = \delta_k$ vs $H_1 : \exists$ at least an i such that: $\underline{a}_i^T \underline{\mu} \neq \delta_i$ then: We reject at level $\alpha \in (0, 1)$ if:

For at least one i we have: $\dfrac{|\underline{a}_i^T \bar{\underline{X}} - \delta_i|}{\sqrt{\underline{a}_i^T S \underline{a}_i}} \sqrt{n} > t_{\alpha/2k}(n-1)$ Indeed: If H_0 is true the probability of rejecting at least one hypothesis $\underline{a}_i^T \underline{\mu} = \delta_i$ when in fact all of them are true is given by:

$$\mathbb{P}\left[\bigcup_{i=1}^{k} \left\{ \frac{|\underline{a}_i^T \bar{\underline{X}} - \delta_i|}{\sqrt{\underline{a}_i^T S \underline{a}_i}} \sqrt{n} > t_{\alpha/2k}(n-1) \right\} | H_0 \text{ is true} \right] \leq \sum_{i=1}^{k} \mathbb{P}\left[\frac{|\underline{a}_i^T \bar{\underline{X}} - \delta_i|}{\sqrt{\underline{a}_i^T S \underline{a}_i}} \sqrt{n} > t_{\alpha/2k}(n-1) | H_0 \text{ is true} \right] = \sum_{i=1}^{k} \alpha/k = \alpha$$

So the overall probability of at least rejecting one null hypothesis when in fact all of them are true is equal to α!

Note: in the above the first inequality is called Bonferroni's Inequality!

Something better, not as conservative is the **False Discovery Rate!**

10 Lecture 15: 2nd Of April 2020

Note:

- If we have One-at-the-time Confidence Intervals, then we reject if we are above $t_{\alpha/2}(n-1)$ or symmetrical if we are below $-t_{\alpha/2}(n-1)$
- If we have Bonferroni's Simultaneous Confidence Intervals with k hypothesis, then we reject if above $t_{\alpha/2k}(n-1)$ or below $-t_{\alpha/2k}(n-1)$ So the more higher k the more conservative the procedure gets, the more the quantiles are closer to infinity!

  Indeed since we test k hypothesis simultaneously, and we want overall level α, then for each single test we need to consider a test much smaller: $\alpha/2k$!

  Conclusion: It's nice mathematically but not applicable in the real world!

False Discovery Rate (FDR)

We want something less conservative but still good: The **False Discovery Rate (FDR)** invented in 1995 by Benjamini and Hochberg is what we look for.

Suppose we have k tests to be performed simultaneously: H_{0i} vs H_{1i} with $i = 1, \dots, k$ Let $\mathcal{D}$ be a strategy, for example let it be: Bonferroni's SimCi, Random Guess, One-at-time CI, or any other strategy we can come up with.

With each strategy we are making a certain amount of errors: each time we do a test we can either reject H_0 or not, and we need to do it k times:

Truth \ Decision	Not Reject H_0	Reject H_0
H_0	U	V
H_1	T	S

Call $k_0 = U + V$ and $k_1 = k - k_0 = T + S$ where: T are the false negatives, V are the false positive!

T, S are the errors, while U, S are correct! Call $V + S = R$ and $U + T = k - R$

We don't know k_0 but we know $k, R, k - R$ these are the only observable quantities!

Suppose now that $\mathcal{D}$ is Bonferroni's strategy, and call I_0 the set of true H_{0i} So from the previous definition we have: $|I_0| = k_0$

Given $\alpha \in (0, 1)$ Bonferroni's SimCI rejects H_{0i} at level α/k Bonferroni is controlling the fact that $V \geq 1$, so it controls the number of false positive, indeed:

$$\mathbb{P}[V \geq 1] = \mathbb{P}\left[\bigcup_{j \in I_0} \{H_{0j} \text{ is rejected}\}\right] \leq \sum_{j \in I_0} \mathbb{P}[H_{0j} \text{ is rejected}] \leq \sum_{j \in I_0} \alpha/k = k_0 \frac{\alpha}{k} \leq k \frac{\alpha}{k} = \alpha \text{ since } k_0 < k$$

Thus if we use Bonferroni's SimCi we are sure that the probability we reject one or more of the true hypothesis is less than α

In this setting, this is called *Family Wise Error Rate (FWER)*. Namely: $FWER = \mathbb{P}[V \geq 1]$

Suppose now that we use a strategy that aims at controlling the false positive rate: we can control the false positive rate, which is $\frac{V}{R}$!

Note: V are called false discoveries, while S are called true discoveries. Then: $\frac{V}{R}$ is a random variable: we can only try to control its expected value which is $\mathbb{E}\left[\frac{V}{R}\right]$ and this is called **False Discovery Rate (FDR)**

Note: V random and we will never observe it, whereas R is random but we will observe it, after the experiment!

Note: We assume that in the case in which $R = 0 \implies \frac{V}{R} = 0$ Define now: $Q = \begin{cases} 0 \text{ if } R = 0 \\ \frac{V}{R} \text{ if } R > 0 \end{cases}$ With this new notation we have: $FDR = \mathbb{E}[Q]$

Observation: if there is nothing to be discovered, then all the assumptions H_{0i} are true thus $k_0 = k$

Therefore $k - k_0 = 0$ so we don't have neither false negative nor true discoveries, therefore: $S = T = 0 \implies V = R \implies Q = \{0, 1\} \implies FDR = \mathbb{E}[Q] = 0 \cdot \mathbb{P}[R = 0] + 1 \cdot \mathbb{P}[R > 0] = \mathbb{P}[R > 0] = \mathbb{P}[V > 0] = \mathbb{P}[V \geq 1] = FWER$ so if there are no discoveries, $FDR = FWER$

Suppose there exists at least something to be discovered, namely: $k_0 < k$ in this case:

- If $V = 0$ then: $Q = 0 \implies FDR = 0$
- If $V > 0$ then: $R > 0 \implies \frac{V}{R} \neq 0$ Indeed: $\frac{V}{R} < 1$ since $R = V + S$

Therefore: $FDR = \mathbb{E}[Q] \leq \mathbb{E}[\mathbb{1}[V > 0]]$ where $\mathbb{1}$ is the characteristic function!

Then since: $\mathbb{1}[V > 0] = \begin{cases} 0 \text{ if } V = 0 \implies Q = 0 \\ 1 \text{ if } V > 0 \implies Q \leq 1 \end{cases} \implies FDR \leq \mathbb{E}[\mathbb{1}[V > 0]] = \mathbb{P}[V \geq 1] = FWER$

Therefore we proved that no matter the number of null hypothesis are true, then $FDR \leq FWER$ and this is exactly why FDR is so appealing: FDR is weaker than $FWER$, so maybe we can control FDR without being so conservative as Bonferroni is!

Consider a procedure such that: $FDR \leq \alpha$ Then we know that the probability of error (FWER) could be much higher but we don't care.

Therefore if we need to do a lot of multiple testing we control FDR: we are not controlling the probability of the type I error (which will be higher), otherwise we wouldn't have any applicability!

Conclusion: Bonferroni is good but not usable, FDR is less good but usable!

A procedure fo controlling FDR is the so called **Benjamini-Hochberg Strategy**

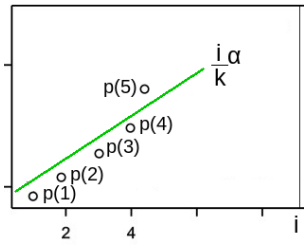
Let p_i be the $p-$value of H_{0i} vs H_{1i} For example if $t_i = \sqrt{n}\frac{|\underline{a}_i^T \underline{\bar{X}} - \mu_{0i}|}{\sqrt{\underline{a}_i^T S \underline{a}_i}}$ Then we have at the right of t_i an area of $p_i/2$ and on the left of $-t_i$ an area of $p_i/2$

Now order the above $p-$values and suppose that: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(k)}$ Let $\alpha \in (0, 1)$ and compute the following:

$$m = \max\left\{ i \in \{1, \ldots, k\} : p_{(i)} \leq \frac{i}{k}\alpha \right\}$$

Note that α, k are fixed so we have a line: $\alpha\frac{i}{k}$ which is a function of i.

we reject all the hypothesis for which the index of the hypothesis is below that threshold: We reject $H_{0(1)}, H_{0(2)}, \ldots, H_{0(m)}$



Note: in the above figure we would have $m = 4$

Note: $H_{01} \neq H_{0(1)}$ where: $H_{0(1)}$ is the hypothesis associated to the $p-$value $p_{(1)}$ which is the smallest one! Thus we reject the hypothesis whose $p-$value is below the line!

Theorem: Benjamini-Hochberg If $p_1, \ldots, p_k$ are independent, that is: we perform k independent test, Then: $FDR \leq \alpha$

Theorem: Benjamini-Yekutieli

- If $p_1, \ldots, p_k$ are positively correlated, then the same procedure works fine so that:
  We reject $H_{0(i)}$ if $i \leq m = \max\left\{j \in \{1, \ldots, k\} : p_{(j)} \leq \frac{j}{k}\alpha\right\}$
- If $p_1, \ldots, p_k$ are negatively correlated, we need a different procedure:
  We reject $H_{0(i)}$ if $i \leq m^* = \max\left\{j \in \{1, \ldots, k\} : p_{(j)} \leq \frac{j}{C_{(k)}k}\alpha\right\}$ where $C_{(k)} = \sum_{j=1}^{k} \frac{1}{j}$

Note: mixed cases are not covered: we need both ALL positively correlated, or ALL negatively correlated $p-$values!

Comparing means of Gaussian distributions: Case 1) Paired data

Suppose we have n statistical units and each unit is observed twice so for each unit we have a vector: $\underline{X}_{1i} = (x_{1i1}, \ldots, x_{1ip})$ and $\underline{X}_{2i} = (x_{2i1}, \ldots, x_{2ip})$ with $i = 1, \ldots n$

We assume $\underline{X}_{1i}$ independent and identically distributed with mean $\underline{\mu}_1$ and we assume $\underline{X}_{2i}$ independent and identically distributed with mean $\underline{\mu}_2$

Our goal is to make inference on the difference of the mean: $\mu_1 - \mu_2$

Note: we need to have paired data: the two vectors are paired in the sense that both are observation for the same statistical unit! For example:

- The statistical unit could be the same person and we could measure his heart beat, his pressure both before and after a treatment!
- The statistical unit could be a family and we observe the degree of education of father and mother and then the income of father and mother!
- What is not good is making the paring after having observed the sample. Consider the following bullshit example:

  We take a bunch of Italians and we measure their height, weight and diet, and we take bunch of French people and we measure the same things!

  Then we pair the tallest french guy with the tallest Italian guy and so on and so forth in decreasing height!

  NO: this pairing is not real its not telling us anything about the statistical unit!

If we are not observing the same statistical unit twice it's bullshit! Indeed we want to take into account the dependence that comes out of the fact that we are observing the same statistical unit twice!

Conclusion: we want to see if treatment has an effect: is there a change? What is the statistical unit? Why is it observed twice?

Set: $\underline{D}_i = \underline{X}_{1i} - \underline{X}_{2i}$ for $i = 1, \dots, n$ then: $\mathbb{E}[\underline{D}_i] = \underline{\mu}_1 - \underline{\mu}_2$ Our goal is to do inference on $\mathbb{E}[\underline{D}_i]$

Assume now that: $\underline{D}_1, \dots, \underline{D}_n \overset{iid}{\sim} \mathcal{N}_p(\underline{\delta}, \Sigma_D)$ where $\underline{\delta} = \underline{\mu}_1 - \underline{\mu}_2$

Let $\bar{\underline{D}} = \frac{1}{n}\sum_{i=1}^{n} \underline{D}_i$ be the sample mean, and let $S_D = \frac{1}{n-1}\sum_{i=1}^{n} (\underline{D}_i - \bar{\underline{D}})(\underline{D}_i - \bar{\underline{D}})^T$ be the sample co-variance.

Then:

$$n(\bar{\underline{D}} - \underline{\delta})^T S_D^{-1} (\bar{\underline{D}} - \underline{\delta}) \sim \frac{(n-1)p}{n-p} F(p, n-p)$$

This our pivotal quantity and we can do inference:

- The confidence region of level $1-\alpha$ with $\alpha \in (0,1)$ is:

$$CR_{1-\alpha}(\underline{\delta}) = CR_{1-\alpha}(\underline{\mu}_1 - \underline{\mu}_2) = \left\{ \underline{\delta} \in \mathbb{R}^p : n(\bar{\underline{D}} - \underline{\delta})^T S_D^{-1} (\bar{\underline{D}} - \underline{\delta}) \leq \frac{(n-1)p}{n-p} F_\alpha(p, n-p) \right\}$$

- Test: $H_0 : \underline{\mu}_1 - \underline{\mu}_2 = \underline{\delta}_0$ vs $H_1 : \underline{\mu}_1 - \underline{\mu}_2 \neq \underline{\delta}_0$ Then: we reject at level $\alpha \in (0,1)$ if:

$$n(\bar{\underline{D}} - \underline{\delta}_0)^T S_D^{-1} (\bar{\underline{D}} - \underline{\delta}_0) > \frac{(n-1)p}{n-p} F_\alpha(p, n-p)$$

- Simultaneous confidence interval of level $1-\alpha$ for the components of the mean with $i = 1, \dots, p$ is:

$$simCI_{1-\alpha}(\mu_{1i} - \mu_{2i}) = \left[\bar{D}_i \pm \sqrt{\frac{(n-1)p}{n-p} F_\alpha(p, n-p)} \sqrt{\frac{S_{D_{ii}}}{n}} \right]$$

- Bonferroni Simultaneous Confidence Interval of level $1-\alpha$ is:

$$BonfCI_{1-\alpha}(\mu_{1i} - \mu_{2i}) = \left[\bar{D}_i \pm t_{\alpha/(2p)}(n-1) \sqrt{\frac{S_{D_{ii}}}{n}} \right]$$

- Since we found the pivotal quantity everything comes out!

Comparing means of Gaussian distributions: Case 2) Repeated Uni-Variate Measures

Each statistical unit is observed q times: we follow the unit along time (instances), and we take a measure, which is a number and not a vector, at each moment!

For example we may want to measure colesterol of the same person each day of the month.

Each statistical unit is $\underline{X}_i = (X_{i1}, \dots, X_{iq})^T$ and it represents q measurements of the same thing! We have: $i = 1, \dots, n$

X_{ij} is the measure of the same quantity in instance $j = 1, \dots, q$ for unit $i = 1, \dots, n$ Then: let $\underline{\mu} = \mathbb{E}[\underline{X}_i] = (\mu_1, \dots, \mu_q)^T$

Our goal is to check whether there differences in the components of $\underline{X}_i$

For example we may want to setup a test: $H_0 : \mu_1 = \dots = \mu_q$ vs $H_1 : \exists i, j$ such that: $\mu_i \neq \mu_j$

The idea is that we give a drug to patient and we want to check if the average level of colesterol is going down!

The sample mean $\underline{\bar{X}}$ is the estimator for $\underline{\mu}$: how do we check to prove that something changes along the q components?

Definition: the Contrast Matrix C is a $(q-1)\times q$ matrix such that:

1) The rows of C are linearly independent
2) $C\cdot \underline{1} = \underline{0}$

This means that if: $C = \begin{bmatrix}\underline{c}_1^T & \cdots & \underline{c}_{q-1}^T\end{bmatrix}^T$ with $\underline{c}_i \in \mathbb{R}^q$ Then C is a contrast if:

- $\underline{c}_1, \dots, \underline{c}_{q-1}$ are linearly independent
- $\underline{c}_i^T \underline{1} = 0\ \forall i$ that is: $\underline{c}_i^T \perp \underline{1}\ \forall i$ This means that the space they span is the linear space orthogonal to the linear space generated by $\underline{1}$ That is: $Span(\underline{c}_1, \dots, \underline{c}_{q-1}) = \mathcal{L}^{\perp}(\underline{1})$

Examples:

- $C = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & 1 & -1 \end{bmatrix}$ here we are computing increments among components (discrete derivative)
- $C = \begin{bmatrix} 1 & 0 & \dots & 0 & -1 \\ 0 & 1 & 0 & \dots & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 1 & -1 \end{bmatrix}$ we have a reference (base line): is μ_i against the base line is any different.

There are infinite number of contrast matrix: any basis of $\mathcal{L}^{\perp}(\underline{1})$ is good and there are infinite basis!

Suppose we want to test: $H_0 : \mu_1 = \dots = \mu_q$ vs $H_1 : \exists i,j : \mu_i \neq \mu_j$ we can rephrase this test as follows:

$$H_0 : \underline{\mu} \in \mathcal{L}(\underline{1}) \text{ vs } H_1 : \underline{\mu} \notin \mathcal{L}(\underline{1}) \iff H_0 : C\underline{\mu} = 0 \text{ vs } H_1 : C\underline{\mu} \neq 0$$

Indeed $C\underline{\mu}$ is projection of $\underline{\mu}$ in the orthogonal space to $\mathcal{L}(\underline{1})$: if $\underline{\mu} \in \mathcal{L}(\underline{1}) \implies C\underline{\mu} = 0$

Now we can consider $C\underline{\bar{X}}$ as estimator for $C\underline{\mu}$ and we can just check if this estimator is close or not to zero! Indeed: $C\underline{\bar{X}} \sim \mathcal{N}_p\left(C\underline{\mu}, \frac{1}{n}C\Sigma C^T\right)$ But we need to estimate: $C\Sigma C^T$

We know that: $(n-1)CSC^T \sim Wishart(C\Sigma C^T, n-1)$ and we know that: $\bar{X} \perp\!\!\!\perp S \implies C\underline{\bar{X}} \perp\!\!\!\perp (n-1)CSC^T$ Then by using Hotelling's Theorem we have that:

$$n(C\underline{\bar{X}} - C\underline{\mu})^T(CSC^T)^{-1}(C\underline{\bar{X}} - C\underline{\mu}) \sim (n-1)(q-1)\frac{1}{n-q+1}F(q-1, n-q+1)$$

Which is a pivotal quantity: we can do all the inference we like on $C\underline{\mu}$

Example: Suppose we want to test: $H_0 : \mu_1 = \dots = \mu_q$ vs $H_1 : \exists i,j : \mu_i \neq \mu_j$ Then: we reject H_0 at level $\alpha \in (0,1)$ if: $n(C\underline{\bar{X}} - C\underline{\mu})^T(CSC^T)^{-1}(C\underline{\bar{X}} - C\underline{\mu}) \geq (n-1)(q-1)\frac{1}{n-q+1}F_{\alpha}(q-1, n-q+1)$

Note: Let C_1 be a different contrast matrix different from C Do we reach a different conclusion? No: The contrast matrix is not given by the problem, is just a way to define a linear basis on the space orthogonal to $\underline{1}$. Indeed: all the basis are equivalent indeed!

Indeed it can be proved that the T^2 statistics we get in the two cases is the same.

11 Lecture 17: 6th Of April 2020

Multi-variate Analysis of Variance (MANOVA)

We have g independent samples coming from g different populations:
$\underline{X}_{11}, \dots, \underline{X}_{1n_1} \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}_1, \Sigma)$ and $\underline{X}_{21}, \dots, \underline{X}_{2n_2} \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}_2, \Sigma)$ and so on and so forth, until: $\underline{X}_{g1}, \dots, \underline{X}_{gn_g} \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}_g, \Sigma)$

Our goal is to make inference on the means $\underline{\mu}_1, \dots, \underline{\mu}_g$ of this population, even though this procedure is called analysis of variance! We want to see if there is enough variability among the estimator of these means to guarantee that they are different!

We want to compare the variability *between* the mean with the variability *within* the population!

Suppose we need to decide a certain dose of fertiliser so to optimise crop: whats the best setting for the parameter (dose of fertiliser)? We treat different statistical unit for each group: are there any differences in the means of the output the experiment is generating in the treatment?

It's important to notice that each group has the same co-variance matrix Σ, so that to make the MANOVA less troublesome. But how do we know if they are the same since they are unknown? Well we estimate them with sample co-variance and then we do some testing on equality of co-variance and then decide if we are satisfied enough to say they are the same!

Note: if the test says that the co-variance matrices are not the same then we try to transform data until the assumption is satisfied!

Note: in its generality, in which we can have any g and any p the MANOVA problem is still an open problem, which is being tackled in a non-parametric way!

Case 1: $g = 2$ and $p \geq 1$

In this setting we have two samples: $\underline{X}_{11}, \dots, \underline{X}_{1n_1} \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}_1, \Sigma)$ and $\underline{X}_{21}, \dots, \underline{X}_{2n_2} \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}_2, \Sigma)$ which we suppose are independent! Moreover note that since we work with Gaussian distribution then n_1 and n_2 can be of any size and we don't have to use the CLT!

Moreover if we didn't have Gaussian Variables, the Curse of Dimensionality exists, and since $p = 3$ then we would need a huge samples with n equal to one million! Since sometimes such large samples aren't available we assume to have Gaussian Variables!

Our goal is to make inference on $\underline{\mu}_1 - \underline{\mu}_2$ So to say: Are the patients taking and not-taking the drug behaving differently?

The sample mean is an estimator for the mean so: $\overline{\underline{X_1}} = \frac{1}{n_1} \sum_{j=1}^{n_1} \underline{X}_{1j}$ is an estimator for μ_1 and $\overline{\underline{X_2}} = \frac{1}{n_2} \sum_{j=1}^{n_2} \underline{X}_{2j}$ is an estimator for μ_2

Since the two samples are Gaussian we know what their sample mean look like:

$$\overline{\underline{X_1}} \sim \mathcal{N}_p\left(\underline{\mu}_1, \frac{1}{n_1}\Sigma\right) \text{ while } \overline{\underline{X_2}} \sim \mathcal{N}_p\left(\underline{\mu}_2, \frac{1}{n_2}\Sigma\right)$$

These two quantities are independent, since the samples are independent, so that:

$$\overline{\underline{X_1}} - \overline{\underline{X_2}} \sim \mathcal{N}_p\left(\underline{\mu}_1 - \underline{\mu}_2, \frac{1}{n_1}\Sigma + \frac{1}{n_2}\Sigma\right) \implies \left(\frac{1}{n_1} + \frac{1}{n_2}\right)^{-1/2} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - (\underline{\mu}_1 - \underline{\mu}_2)\right] \sim \mathcal{N}_p(\underline{0}, \Sigma)$$

Moreover: $S_1 = \frac{1}{n_1 - 1} \sum_{j=1}^{n_1} (\underline{X}_{1j} - \overline{\underline{X}}_1)(\underline{X}_{1j} - \overline{\underline{X}}_1)^T$ is an estimator for Σ and the equivalent holds for S_2

Since $(n_1 - 1)S_1 \sim Wishart(\Sigma, n_1 - 1)$ and $(n_2 - 1)S_2 \sim Wishart(\Sigma, n_2 - 1)$ Then since these two are independent we conclude that: $(n_1 - 1)S_1 + (n_2 - 1)S_2 \sim Wishart(\Sigma, n_1 + n_2 - 2)$

The weighted average of the two estimator for Σ is:

$$S_{pooled} = \frac{(n_1 - 1)S_1 + (n_2 - 1)S_2}{n_1 + n_2 - 2} \text{ such that } (n_1 + n_2 - 2)S_{pooled} \sim Wishart(\Sigma, n_1 + n_2 - 2)$$

Moreover we know that: $S_{pooled} \perp\!\!\!\perp \overline{\underline{X}}_1 \wedge S_{pooled} \perp\!\!\!\perp \overline{\underline{X}}_2$ since $S_1 \perp\!\!\!\perp \overline{\underline{X}}_1 \wedge S_2 \perp\!\!\!\perp \overline{\underline{X}}_2$

Thus:

$$(n_1 + n_2 - 2)S_{pooled} \perp\!\!\!\perp \left(\frac{1}{n_1} + \frac{1}{n_2}\right)^{-1/2} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - (\underline{\mu}_1 - \underline{\mu}_2)\right]$$

Thus we can use Hotelling's Theorem so that:

$$\left(\frac{1}{n_1} + \frac{1}{n_2}\right)^{-1} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - (\underline{\mu}_1 - \underline{\mu}_2)\right]^T S_{pooled}^{-1} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - (\underline{\mu}_1 - \underline{\mu}_2)\right] \sim \frac{(n_1 + n_2 - 2)p}{n_1 + n_2 - 1 - p} F(p, n_1 + n_2 - 1 - p)$$

Note that we just got our pivotal statistics: we now can make all inference we like! For example:

Testing $H_0 : \underline{\mu}_1 - \underline{\mu}_2 = \underline{\delta}_0$ vs $H_1 : \underline{\mu}_1 - \underline{\mu}_2 \neq \underline{\delta}_0$ We reject H_0 at level $\alpha \in (0, 1)$ if:

$$\left(\frac{1}{n_1} + \frac{1}{n_2}\right)^{-1} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - \underline{\delta}_0\right]^T S_{pooled}^{-1} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - \underline{\delta}_0\right] > \frac{(n_1 + n_2 - 2)p}{n_1 + n_2 - 1 - p} F_\alpha(p, n_1 + n_2 - 1 - p)$$

Moreover:

$$CR_{1-\alpha}(\underline{\mu}_1 - \underline{\mu}_2) =$$

$$= \left\{ \underline{\delta} \in \mathbb{R}^p : \left(\frac{1}{n_1} + \frac{1}{n_2}\right)^{-1} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - \underline{\delta}\right]^T S_{pooled}^{-1} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - \underline{\delta}\right] \leq \frac{(n_1 + n_2 - 2)p}{n_1 + n_2 - 1 - p} F_\alpha(p, n_1 + n_2 - 1 - p) \right\}$$

Special weird case: Behrens-Fisher Problem

First of all note that this is still open in its total generality! Suppose now that $g = 2$ and $p \geq 1$

If Σ is the same then S_1, S_2 are surely different but we can believe that they are realisation coming from the same random variable.

Is their difference enough to believe Σ is different?

Suppose that $\underline{X}_{11}, \ldots, \underline{X}_{1n_1} \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}_1, \Sigma_1)$ and $\underline{X}_{21}, \ldots, \underline{X}_{2n_2} \overset{iid}{\sim} \mathcal{N}_p(\underline{\mu}_2, \Sigma_2)$ with $\Sigma_1 \neq \Sigma_2$

We can perform a test, namely: $H_0 : \Sigma_1 = \Sigma_2$ vs $H_1 : \Sigma_1 \neq \Sigma_2$ But:

- Maybe there are tests out there that try to perform such test but most of them are based on the Gaussianity assumption (e.g: Extended Levene Test)
- Another possibility is to use non parametric testing: we work out a distance capturing the distance between S_1, S_2 and then we check if this distance is big enough!

  With this there a few problems: we need to define the distance between positive definite matrices, which are points of Riemannian Manifold: we don't have a vector space structure.

Suppose now that the samples size n_1, n_2 are large: then we don't need the Gaussianity assumption since we use asymptotic theory! Therefore we just suppose that: $\underline{X}_{11}, \dots, \underline{X}_{1n_1} \overset{iid}{\sim} (\underline{\mu}_1, \Sigma_1)$ and $\underline{X}_{21}, \dots, \underline{X}_{2n_2} \overset{iid}{\sim} (\underline{\mu}_2, \Sigma_2)$ so we are just specifying the mean and the variance!

Note: we still suppose that the two samples are independent though!

By the **CLT** we know that: $\overline{\underline{X}}_1 \sim \mathcal{N}_p\left(\underline{\mu}_1, \frac{1}{n_1}\Sigma_1\right)$ and $\overline{\underline{X}}_2 \sim \mathcal{N}_p\left(\underline{\mu}_2, \frac{1}{n_2}\Sigma_2\right)$ and they are still independent. Thus:

$$\overline{\underline{X}}_1 - \overline{\underline{X}}_2 \sim \mathcal{N}_p\left(\underline{\mu}_1 - \underline{\mu}_2, \frac{1}{n_1}\Sigma_1 + \frac{1}{n_2}\Sigma_2\right)$$

Therefore:

$$\left(\frac{1}{n_1} + \frac{1}{n_2}\right)^{-1} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - (\underline{\mu}_1 - \underline{\mu}_2)\right]^T \left[\frac{1}{n_1}\Sigma_1 + \frac{1}{n_2}\Sigma_2\right]^{-1} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - (\underline{\mu}_1 - \underline{\mu}_2)\right] \sim \chi^2(p)$$

But:

- $S_1 \to \Sigma_1$ as $n_1 \to \infty$ in probability (and thus in law)
- $S_2 \to \Sigma_2$ as $n_2 \to \infty$ in probability (and thus in law)

Thus if n_1 and n_2 are large enough we can apply **LLW** and **CLT** so that we get our pivotal quantity:

$$\left(\frac{1}{n_1} + \frac{1}{n_2}\right)^{-1} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - (\underline{\mu}_1 - \underline{\mu}_2)\right]^T \left[\frac{1}{n_1}S_1 + \frac{1}{n_2}S_2\right]^{-1} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - (\underline{\mu}_1 - \underline{\mu}_2)\right] \sim \chi^2(p)$$

With the pivotal quantity we can do all of our inference, for example:

$$CR_{1-\alpha}(\underline{\mu}_1 - \underline{\mu}_2) = \left\{\underline{\delta} \in \mathbb{R}^p : \left(\frac{1}{n_1} + \frac{1}{n_2}\right)^{-1} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - \underline{\delta}\right]^T \left[\frac{1}{n_1}S_1 + \frac{1}{n_2}S_2\right]^{-1} \left[(\overline{\underline{X}}_1 - \overline{\underline{X}}_2) - \underline{\delta}\right] \leq \chi^2_\alpha(p)\right\}$$

Note: $n_1.n_2$ need to be large enough with respect to the number of parameters p, indeed the sample size need to increase exponentially!

Case 2: $g \geq 1$ and $p = 1$

This is the Classical ANOVA case: $p = 1$ so its not multivariate. We have g samples: $X_{11}, \dots, X_{1n_1} \overset{iid}{\sim} \mathcal{N}_1(\mu_1, \sigma^2)$ and $X_{21}, \dots, X_{2n_2} \overset{iid}{\sim} \mathcal{N}_1(\mu_2, \sigma^2)$, and so on and so forth, until: $X_{g1}, \dots, X_{gn_g} \overset{iid}{\sim} \mathcal{N}_1(\mu_g, \sigma^2)$

All the g samples are supposed to be independent: so we have independence in each samples and also across each row since each component of each sample is independent and identically distributed.

Note: Since $p = 1$ we have one feature for each statistical unit. Moreover we suppose that σ^2 is the same for all g

Our goals are:

1) $H_0 : \mu_1 = \dots = \mu_g$ (e.g: fertiliser has no effect) vs $H_1 : \exists i \neq j : \mu_i \neq \mu_j$ (e.g: fertiliser has some effect)
2) If we reject H_0 we need to estimate the μ_i so that we find the best (best according to our problem).

Now we make a new parametrisation of the problem: $\mu_i = \mu + \tau_i$ where:

- μ is independent of group in which we are: its the overall mean!
- τ_i with $i = 1, \dots, g$ This tells us how different we are from overall mean because of the treatment effect.

Thus we moved from a problem with g parameters: μ_i, to a problem with $g+1$ parameters: μ, τ_i

Thus we need to add a constraint so that we don't over-parametrise the problem!

The model for our random variables X_{ij} becomes:

$$X_{ij} = \mu + \tau_i + \epsilon_{ij} \text{ where } \mu \in \mathbb{R}, \tau_i \in \mathbb{R} \ \ \forall i = 1, \dots, g \text{ and } \epsilon_{ij} \overset{iid}{\sim} \mathcal{N}_1(0, \sigma^2) \text{ for } i = 1, \dots, g; ; j = 1, \dots, n_i$$

This is the ANOVA model, which can be viewed as a very specific linear model.

The above it's not complete: we only need an additional constraint to avoid over-parametrising the problem.

We obtain it as follows: we impose that the estimator for the overall mean μ is unbiased! So:
Suppose $n_1 + \cdots + n_g = n$ then we take the overall sample mean $\overline{X} = \frac{1}{n} \sum_{i=1}^{g} \sum_{j=1}^{n_j} X_{ij}$ With:

$$\mathbb{E}[\overline{X}] = \frac{1}{n} \sum_{i=1}^{g} \sum_{j=1}^{n_j} \mathbb{E}[X_{ij}] = \frac{1}{n} \sum_{i=1}^{g} \sum_{j=1}^{n_j} (\mu + \tau_i) = \frac{1}{n} \sum_{i=1}^{g} (n_i \mu + \tau_i n_i) = \mu \frac{1}{n} \sum_{i=1}^{g} n_i + \frac{1}{n} \sum_{i=1}^{g} n_i \tau_i = \mu + \frac{1}{n} \sum_{i=1}^{g} n_i \tau_i$$

Thus: $\overline{X}$ is unbiased $\iff \mathbb{E}[\overline{X}] = 0 \iff \sum_{i=1}^{g} n_i \tau_i = 0$

We impose that $\underline{X}$ is unbiased so we impose: $\sum_{i=1}^{g} n_i \tau_i = 0$ This is the constraint we need to add to the above ANOVA model to get it complete. By adding we reduce the number of free parameters, so that we don't over prametrise the problem.

Note: if $n_1 = \cdots = n_g$ we have balanced design and the constraint becomes: $\sum_{i=1}^{g} t_i = 0$ this is not really realistic though!

Now we want to find the estimators for the treatment effect τ_i, to do so: we take the sample mean in group i and we discount what we already estimated in the base line: $\overline{X}_i - \overline{X}$ where $\overline{X}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} X_{ij}$ is the sample mean in group i

Since $\mathbb{E}[\overline{X_i} - \overline{X}] = \mathbb{E}[\overline{X_i}] - \mu = \mu + \tau_i - \mu = \tau_i$ we have that it's an unbiased estimator for the treatment effect! Note that this is due to the fact that we imposed $\overline{X}$ to be unbiased!

Geometry of variance decomposition

Consider a basis of vectors in $\mathbb{R}^g$ given by:

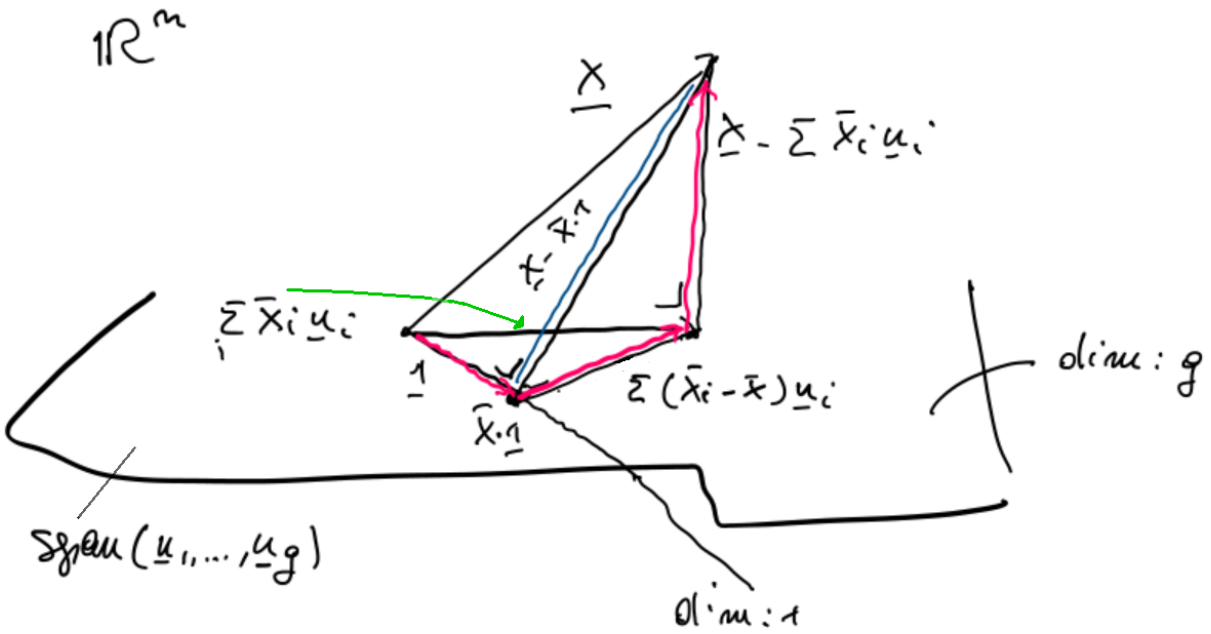
$$\underline{u}_1, \dots, \underline{u}_g \text{ where: } \underline{u}_1 = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Bigg\} n_1 \quad \text{and so on and so forth: } \underline{u}_g = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \Bigg\} n_g$$

Note: these are called dummy variables in linear models!

We have that:

1) $\underline{u}_1, \ldots, \underline{u}_g$ are linearly independent
2) $\underline{u}_1, \ldots, \underline{u}_g$ are orthogonal by construction
3) $\underline{1} \in Span(\underline{u}_1, \ldots, \underline{u}_g)$, indeed: $\underline{1} = \sum_{i=1}^{g} \underline{u}_i$

Now: let $\underline{X} = (X_{11}, \ldots, X_{1n_1}, X_{21}, \ldots, X_{g1}, \ldots, X_{gn_g})^T \in \mathbb{R}^n$ be the vector of observations. Then:



Call $\underline{X}^{\perp}$ the orthogonal projection of $\underline{X}$ on $Span(\underline{u}_1, \ldots, \underline{u}_g)$ So that: $\underline{X}^{\perp} = \sum_{i=1}^{g} \frac{\underline{u}_i \underline{u}_i^T}{\underline{u}_i^T \underline{u}_i} \underline{X} = \sum_{i=1}^{g} \frac{1}{n_i} \sum_{j=1}^{n_i} X_{ij} \underline{u}_i = \sum_{i=1}^{g} \bar{X}_i \underline{u}_i$

Now call: $\star = \underline{X} - \underline{X}^{\perp} = \underline{X} - \sum_{i=1}^{g} \bar{X}_i \underline{u}_i$

The orthogonal projection of $\underline{X}$ on $Span(\underline{1})$ is given by: $\frac{\underline{1}\underline{1}^T}{\underline{1}^T \underline{1}} \underline{X} = \frac{1}{n} \sum_{i=1}^{g} \sum_{j=1}^{n_i} X_{ij} \underline{1} = \bar{X} \cdot \underline{1}$

The orthogonal projection of $\underline{X}^{\perp} = \sum_{i=1}^{g} \bar{X}_i \underline{u}_i$ on $Span(\underline{1})$ is given by:

$$\frac{\underline{1}\underline{1}^T}{\underline{1}^T \underline{1}} \left(\sum_{i=1}^{g} \bar{X}_i \underline{u}_i \right) = \sum_{i=1}^{g} \left(\bar{X}_i \underline{1}\underline{1}^T \frac{1}{\underline{1}^T \underline{1}} \underline{u}_i \right) = \frac{1}{n} \left(\sum_{i=1}^{g} n_i \bar{X}_i \right) \underline{1} = \frac{1}{n} \left(\sum_{i=1}^{g} \sum_{j=1}^{n_i} X_{ij} \right) \underline{1} = \bar{X} \cdot \underline{1}$$

Now since: $\underline{X}^{\perp} - \bar{X} \cdot \underline{1} = \sum_{i=1}^{g} (\bar{X}_i - \bar{X}) \underline{u}_i$ Thus we conclude that:

$$\underline{X} = \bar{X} \cdot \underline{1} + (\underline{X}^{\perp} - \bar{X} \cdot \underline{1}) + (\underline{X} - \underline{X}^{\perp})$$

Where each of the three component is orthogonal to each other, hence:

$$\underline{X} = (\bar{X} \cdot \underline{1}) + \left(\sum_{i=1}^{g} (\bar{X}_i - \bar{X}) \underline{u}_i \right) + \left(\underline{X} - \sum_{i=1}^{g} \bar{X}_i \underline{u}_i \right)$$

Since the three vectors are orthogonal:

$$\|\underline{X}\|^2 = \left\|\bar{X} \cdot \underline{1}\right\|^2 + \left\| \sum_{i=1}^{g} (\bar{X}_i - \bar{X}) \underline{u}_i \right\|^2 + \left\| \underline{X} - \sum_{i=1}^{g} \bar{X}_i \underline{u}_i \right\|^2$$

Thus we have that:

$$\sum_{i=1}^{g} \sum_{j=1}^{n_i} X_{ij}^2 = n\bar{X}^2 + \sum_{i=1}^{g} (\bar{X}_i - \bar{X})^2 n_i + \sum_{i=1}^{g} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2$$

This is the **decomposition of variance**: $SS_{obs} = SS_{mean} + SS_{treatment} + SS_{residuals}$ where: $SS_{residuals}$ is what is not captured by a linear combination of $\underline{u}_1, \ldots, \underline{u}_g$ with $\underline{X}$

There is a second possibility for the above formula, namely:

$$\left\|\underline{X} - \bar{X} \cdot \underline{1}\right\|^2 = \left\| \sum_{i=1}^{g} (\bar{X}_i - \bar{X}) \underline{u}_i \right\|^2 + \left\| \underline{X} - \sum_{i=1}^{g} \bar{X}_i \underline{u}_i \right\|^2$$

So that:

$$\sum_{i=1}^{g} \sum_{j=1}^{n_i} (X_{ij} - \bar{X})^2 = \sum_{i=1}^{g} (\bar{X}_i - \bar{X})^2 n_i + \sum_{i=1}^{g} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2 \implies SS_{centered} = SS_{treatment} + SS_{residuals}$$

Both formulas are called **Variance Decomposition Formula**.

In the formula we just found we see that the the total variability of the data around the mean in split in two components: the variability between groups plus the variability withing groups! Indeed:

- Consider the variability between groups: if each unit was equal to its mean what would be the variability of the data set?
- Consider the variability within groups: we sum over all group variability within the group!

Now we have that: $H_0 : \mu_1 = \cdots = \mu_g$ vs $H_1 : \exists i \neq j : \mu_i \neq \mu_j$ is equivalent to: $H_0 : \tau_1 = \cdots = \tau_g = 0$ vs $H_1 : \exists \tau_i \neq 0$

If H_0 is true all the means are the same: so we expect that the two projection to be the same so that the vector $\left(\sum_{i=1}^{g} (\bar{X}_i - \bar{X}) \underline{u}_i \right)$ is small with respect to $\left(\underline{X} - \sum_{i=1}^{g} \bar{X}_i \underline{u}_i \right)$

Therefore we reject H_0 if: $\dfrac{\left\| \left(\sum_{i=1}^{g} (\bar{X}_i - \bar{X}) \underline{u}_i \right) \right\|^2}{\left\| \left(\underline{X} - \sum_{i=1}^{g} \bar{X}_i \underline{u}_i \right) \right\|^2}$ is large, that is if: $\dfrac{\sum_{i=1}^{g} (\bar{X}_i - \bar{X})^2 n_i}{\sum_{i=1}^{g} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2}$ is large

How large? we need to know what is the distribution of the above fraction if H_0 is true:
$\underline{X}$ is Gaussian so each of the three pieces of the Variance Decomposition are Gaussian, since they are linear transformations of $\underline{X}$ Moreover each of the three vectors are orthogonal, so since they are Gaussian this implies that they are also independent!

Therefore we have three independent Gaussian so in the above fraction we have the quotient of the square of two Gaussian vector and so we have a chi-squared distribution!

Indeed we can prove that:

$$F_0 = \frac{\frac{1}{g-1}\sum_{i=1}^{g}(\bar{X}_i - \bar{X})^2 n_i}{\frac{1}{n-g}\sum_{i=1}^{g}\sum_{j=1}^{n_i}(X_{ij} - \bar{X}_i)^2} \sim F(g-1, n-g)$$

Indeed the dimension of the space we work in is g, while dimension of $\underline{1}$ is 1 so first vector of the variance decomposition lives in a space of dimension 1

The second vectors lives in a space of dimension $g-1$ since it's orthogonal to the linear space of dimension 1 and we stated in a space of dimension g That is: $\sum_{i=1}^{g}(\bar{X}_i - \bar{X})\underline{u}_i \in Span(\underline{1})^{\perp} \cap Span(\underline{u}_1, \ldots, \underline{u}_g)$

The third vector lives in a space of dimension $n-g$ since we projected on a linear space of dimension g That is: $\underline{X} - \sum_{i=1}^{g}\bar{X}_i\underline{u}_i \in Span(\underline{u}_1, \ldots, \underline{u}_g)^{\perp}$

So the three vectors aren't free to move their component: the first vector has 1 degree of freedom, the second vector has $g-1$ degrees of freedom, and the third vector has $n-g$ degrees of freedom!

Note that anyway all the three vectors have n component, so that's why they are called degrees of freedom, and that is why they are also the normalisation constant in the F_0 above.

Note: If $\underline{X} \sim \mathcal{N}_p(\underline{\mu}, \sigma^2 I)$ and if $P : \mathbb{R}^p \to \mathbb{R}^p$ is an orthogonal projection. Then:
$P\underline{X}$ and $(I-P)\underline{X}$ are independent.

Note: $(I-P)$ projects on the orthogonal sub-space to which P projects onto.

Remember: In general: zero correlation doesn't imply independence where as independence always implies zero correlation!

Note that: $SS_{residuals} = \sum_{i=1}^{g}(\bar{X}_i - \bar{X})^2 n_i = \sum_{i=1}^{g}(n_i - 1)S_i^2$ where $S_i^2 = \frac{1}{n-1}\sum_{j=1}^{n_i}(X_{ij} - \bar{X}_i)^2$ is an estimator of σ^2 in group i

We know that $S_i \sim \sigma^2\chi^2(n_i - 1)$ moreover: S_i are independent as they refer to samples which are independent.

Thus: $SS_{residuals} \sim \sigma^2\chi^2(n-g)$ since we sum g independent chi-squared distribution.

Therefore: If $H_0 : \tau_i = \tau_2 = \cdots = \tau_g = 0$ is true: $\sum_{i=1}^{g}\sum_{j=1}^{n_i}(X_{ij} - \bar{X})^2 = (n-1)S^2 \sim \sigma^2\chi^2(n-1)$ Hence if H_0 is true: $SS_{centered} \sim \sigma^2\chi^2(n-1)$ but: $SS_{centered} = SS_{treatment} + SS_{residuals}$ and $SS_{residuals} \sim \sigma^2\chi^2(n-g)$ and they are independent thus: $SS_{treatment} = \sigma^2\chi^2(g-1)$

If H_0 is true: $F_0 = \frac{\frac{1}{g-1}SS_{treatment}}{\frac{1}{n-g}SS_{residuals}} \sim F(g-1, n-g)$ since its quotient of two independent chi squared!

Therefore we reject H_0 at level $\alpha \in (0,1)$ if $F_0 > F_\alpha(g-1, n-g)$ from which we get $p-$values, IC, and so on and so forth.

12 Lecture 19: 9th Of April 2020

General Case: MANOVA $p \geq 1$ and $g \geq 2$

Here we are in a mutlivariate setting: we have g groups of statistical units, but each group is treated with a different level of the treatment.

What we might observe in the end are either due to the variability (ϵ) or due to the difference that has been generated by the different levels of treatments.

Everything is the same expect for the treatment: unlike ANOVA, here for each statistical unit we have a vector of measurements (so we have more features observed for each statistical unit!)

$p \geq 1, g \geq 2$ thus: $\mathbb{R}^p \ni \underline{X}_{ij} = \underline{\mu} + \underline{\tau}_i + \underline{\epsilon_{ij}}$ with $i = 1, \dots, g$ and $j = 1, \dots, n_i$

Since we are still over parameterising the problem we need to impose that: $\sum_{i=1}^{g} n_i \underline{\tau}_i = \underline{0}$ and we assume $\underline{\epsilon}_{ij} \overset{iid}{\sim} \mathcal{N}(\underline{0}, \Sigma)$

Note that: For a fixed component k of $\underline{X}_{ij}$ we have ANOVA: $X_{ijk} = \mu_k + \tau_{ik} + \epsilon_{ijk}$ for $i = 1, \dots, g$ and $j = 1, \dots, n_i$ and $k = 1, \dots, p$ Where: we impose $\sum n_i \tau_{ik} = 0$ and $\epsilon_{ijk} \overset{iid}{\sim} \mathcal{N}(0, \sigma_{kk})$

The statistical units and groups are independent, but there could be dependence between the components, for instance between component k and l and this dependence is expressed by σ_{kl}

Note: p independent ANOVA on components is not equivalent to MANOVA: we aren't considering the correlation!

We have that $\bar{\underline{X}}$ is an estimator for $\underline{\mu}$, $\bar{\underline{X}}_i$ is an estimator for $\underline{\mu}_i$ and $\bar{\underline{X}}_i - \bar{\underline{X}}$ is an estimator for $\underline{\tau}_i$

All of these three estimators are unbiased due to the constraint: $\sum_{i=1}^{g} n_i \underline{\tau}_i = \underline{0}$ which makes us avoid over-parametrisation of the problem!

Our goal is to perform a test of the kind: $H_0 : \underline{\tau}_1 = \dots = \underline{\tau}_g = 0$ vs $H_1 : \exists\, \underline{\tau}_i \neq \underline{0}$

Co-variance Decomposition Formula

We have that:

$$\sum_{i=1}^{g}\sum_{j=1}^{n_i}(\underline{X}_{ij} - \bar{\underline{X}})(\underline{X}_{ij} - \bar{\underline{X}})^T = \sum_{i=1}^{g}(\bar{\underline{X}}_i - \bar{\underline{X}})(\bar{\underline{X}}_i - \bar{\underline{X}})n_i + \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\underline{X}_{ij} - \bar{\underline{X}}_i)(\underline{X}_{ij} - \bar{\underline{X}}_i)^T$$

This is easy to prove indeed we just need to use the identity: $\underline{X}_{ij} - \bar{\underline{X}} = (\bar{\underline{X}}_i - \bar{\underline{X}}) + (\underline{X}_{ij} - \bar{\underline{X}}_i)$

Now we define two quantities:

- $B = \sum_{i=1}^{g} n_i(\bar{\underline{X}}_i - \bar{\underline{X}})(\bar{\underline{X}}_i - \bar{\underline{X}})^T$ This is the Between Co-variability: what would be co-variability in the data set if every statistical unit was equal to its mean?
- $W = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\underline{X}_{ij} - \bar{\underline{X}}_i)(\underline{X}_{ij} - \bar{\underline{X}}_i)^T$ This is the Within co-variability: if we are in group i, what is the variability with respect to the mean of this group?

We do the same for each group and sum up these co-variability, indeed $W = \sum_{i=1}^{g} (n_i - 1)S_i$ where S_i is the sample co-variance computed in group i Thus W is a pooled co-variability since we are averaging all the estimations for Σ

This is the reason why we need the same Σ, otherwise W wouldn't have the above expression!

Note that by analogy with ANOVA, we would like to consider $\frac{B}{W}$ but we can't take the ratio of two matrices! There are different proposals:

- **Wilks-Lambda Proposal:** $\Lambda_W = \dfrac{det(W)}{det(B+W)}$

  If Λ_W is is large it means that the generalised variance within groups its no so different from the overall generalised variance!

  This means that the treatment didn't produce much effect: the treatment didn't increase the variability among groups!

  Therefore we reject $H_0 : \underline{\tau}_1 = \cdots = \underline{\tau}_g = \underline{0}$ if Λ_w is small.

- **Pillai-Lambda Proposal:** $\Lambda_p = tr(B \cdot (B+W)^{-1})$ We reject $H_0 : \underline{\tau}_1 = \cdots = \underline{\tau}_g = \underline{0}$ if Λ_P is large.

- **Lawley-Hotelling Lambda Proposal:** $\Lambda_{LH} = tr(BW^{-1})$ We reject $H_0 : \underline{\tau}_1 = \cdots = \underline{\tau}_g = \underline{0}$ if Λ_{LH} is large.

Note: all the three above are function of the eigenvalues $\lambda_1, \ldots, \lambda_s$ of BW^{-1}

Note: B is a $p \times p$ matrix, and we obtain it from vectors $\bar{\underline{X}}_i - \bar{\underline{X}}$ which live in a $g-1$ dimensional space: if $g - 1 < p$ then $det(B) = 0 \implies$ There are only s eigenvalues where $s = min(g-1, p) = rank(B)$

Therefore we don't take $\dfrac{det(B)}{det(W)}$ as test statistics because most times $det(B) = 0$!

In the above what does it mean large or small? we would need to know the distribution of the Λ under H_0 above but we don't know them: they are generally unknown so we capture them through simulations (e.g: Monte-Carlo).

About Λ_W:

- If $p \geq 1$ and $g = 2, 3$ then the distribution under H_0 is known: we have a perfect analytical solution.
- If $p = 2$ and $g \geq 1$ the distribution under H_0 is known!
- The asymptotic distribution of Λ_W is known for $n \to \infty$, indeed under H_0 its given by:

$$-\left(n - 1 - \frac{(p+g)}{2}\right) \log(\Lambda_W) \sim \chi^2(p(g-1))$$

  This is known as **Bartlett's Approximation!**

  Note that we need all n_i growing!

  This in this case we reject at level $\alpha \in (0,1)$ if $-\left(n - 1 - \frac{(p+g)}{2}\right) \log(\Lambda_W) > \chi^2_\alpha(p(g-1))$

  Note that since we have a minus in front of the pivotal statistics, we are taking $-\log$: so we reject for small value of Λ_W, so we reject for values of $-\log \Lambda_W$ which are big.

If H_0 has been rejected, then was there any effect? We want to estimate the effect of the treatment: we need to compare $\underline{\tau}_i, \underline{\tau}_k$ component wise, using Bonferroni's Simultaneous Confidence Intervals for $\tau_{il} - \tau_{kl}$ with $i, k = 1, \dots, g$ and $l = 1, \dots, p$

An estimator is given by: $\bar{X}_{il} - \bar{X}_{kl} \sim \mathcal{N}\left(\tau_{il} - \tau_{kl}, \frac{1}{n_i}\sigma_{ll} + \frac{1}{n_k}\sigma_{ll}\right)$

Moreover $\frac{W}{(n-g)}$ is an estimator for Σ so $\dfrac{W_{ll}}{(n-g)}$ is an estimator for σ_{ll}

We need to do p confidence interval, one for each component, and for each component we compare two groups so we need to compute $pg(g-1)\frac{1}{2}$ Bonferroni's Simultaneous Confidence Intervals of overall level $1-\alpha$ Thus:

$$BonfSimCI_{1-\alpha}(\tau_{il} - \tau_{kl}) = \left[\bar{X}_{il} - \bar{X_{kl}} \pm t_{\frac{\alpha}{2}.\frac{2}{(pg(g-1))}}(n-g)\sqrt{\frac{W_{ll}}{n-g}\left(\frac{1}{n_i} + \frac{1}{n_k}\right)}\right]$$

Note: We could also do normal Simultaneous Confidence Intervals but they are large!

Note: If in the above we set $p = 1$ we get the Bonferroni's Simultaneous Confidence Intervals in the case of ANOVA.

Two Ways MANOVA

Before we saw a one way MANOVA, now we want to perform two ways MANOVA: we introduce simultaneously two different treatments.

We have a factor 1 for treatment 1 with different levels: $1, \dots, g$ and a factor 2 for treatment 2 with different levels: $1, \dots, b$ Thus we have an $g \times b$ matrix of the treatments!

We observe: $X_{ijk} \in \mathbb{R}$ which is the result of an experiment when factor 1 is at level i and factor 2 is at level j Moreover we have k statistical units in this group.

Note that this group is characterised by two levels of two factors!

Moreover we have that $k = 1, \dots, n$ We assume the same sample size for each group so we have a balanced experiment!

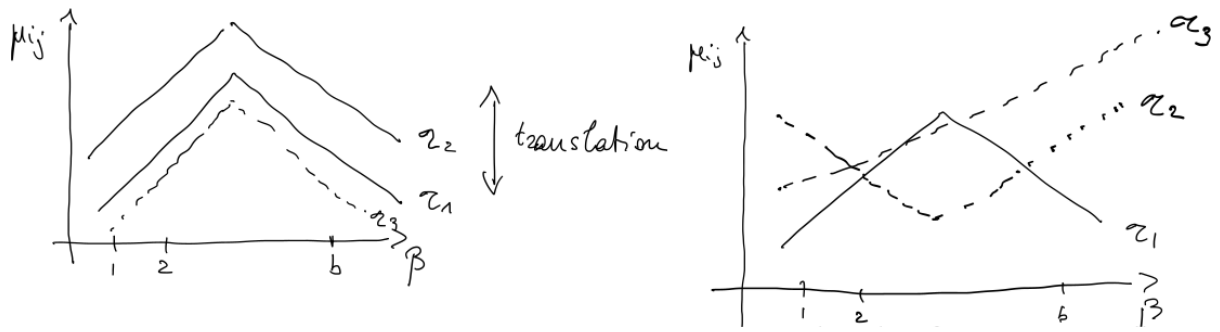
Otherwise it's more complicated and it's a branch of statistics called design of experiments!

For simplicity suppose $p = 1$ that is: $X_{ijk} \in \mathbb{R}$ we have the following two ways ANOVA model:

- $X_{ijk} = \mu + \tau_i + \beta_j + \gamma_{ij} + \epsilon_{ijk}$
- $\mu \in \mathbb{R}$ is the overall mean
- $\tau_i, \beta_j \in \mathbb{R}$ are the effects of treatment 1 and the effects of treatment 2 with $i = 1, \dots, g$ and $j = 1, \dots, b$
- We consider also interactions between the treatments: $\gamma_{ij} \in \mathbb{R}$
- $\epsilon_{ijk} \in \mathbb{R}$ are the residual with $k = 1, \dots, n$ Moreover $\epsilon_{ijk} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$ and $\sigma^2 \in \mathbb{R} > 0$
- To avoid over-parametrisation of the problem we impose:

$$\sum_{i=1}^{g} \tau_i = 0 \text{ and } \sum_{j=1}^{b} \beta_j = 0 \text{ and } \sum_{i=1}^{g} \gamma_{ij} = 0 \text{ and } \sum_{j=1}^{b} \gamma_{ij} = 0$$

Do we want interaction or a purely additive model, in which the effects of the treatments purely add up? Let's see the graphical difference:



In the left figure we see the model without interaction: it is the additive model: $\mu_{ij} = \mu + \tau_i + \beta_j$ where μ_{ij} is the mean for group i, j and $\gamma_{ij} = 0$ Translation of the profile of μ_{ij} is the only possibility!

In the right figure we see the model with interaction: it's the complete model $\mu_{ij} = \mu + \tau_i + \beta_j + \gamma_{ij}$ We can see that the two factor interact so the effect of β on overall mean depends on level of factor 1

The second model seems richer: should we always use it? No, it has many more parameter for the mean, so we have less degrees of freedom for estimating σ^2!

Indeed if we use the complete model we have less bias for the mean but we have greater uncertainty about what we are doing!

As a matter of fact if we don't have much data its a bad choice to use complete model: good point estimate but, we are over-fitting data: no estimate of variability!

This is always due to the **bias-variance trade-off!**

In general **the Decomposition of Variance** is given by:

$$\sum_i \sum_j \sum_k (X_{ijk} - \bar{X})^2 = \sum_{i=1}^{g} (\bar{X}_{i\bullet} - \bar{X})^2 bn + \sum_{j=1}^{b} (\bar{X}_{\bullet j} - \bar{X})^2 gn + \sum_i \sum_j (\bar{X}_{ij} - \bar{X}_{i\bullet} - \bar{X}_{\bullet j} + \bar{X})^2 n + \sum_i \sum_j \sum_k (X_{ijk} - \bar{X}_{ij})^2$$

which can be written as:

$$SS_{centered} = SS_{treatment1} + SS_{treatment2} + SS_{interaction} + SS_{residuals}$$

where:

- $SS_{centered} = \sum_i \sum_j \sum_k (X_{ijk} - \bar{X})^2$ is the total variability
- $\bar{X}_{i\bullet}$ is the mean along the $i-$th row
- $\bar{X}_{\bullet j}$ is the mean along the $j-$the column: here we fix the level for factor 2 and we take the average of all observation with that fixed factor!
- $\bar{X}_{ij}$ is the overall mean in group i, j
- $SS_{treatment1} = \sum_{i=1}^{g} (\bar{X}_{i\bullet} - \bar{X})^2 bn$ and $SS_{treatment2} = \sum_{j=1}^{b} (\bar{X}_{\bullet j} - \bar{X})^2 gn$

- $SS_{interaction} = \sum_i \sum_j (\bar{X}_{ij} - \bar{X}_{i\bullet} - \bar{X}_{\bullet j} + \bar{X})^2 n$ and $SS_{residuals} = \sum_i \sum_j \sum_k (X_{ijk} - \bar{X}_{ij})^2$ is the Sum of Squared Residuals

Note that:

- $SS_{centered}$ has 1 degree of freedom!
- $SS_{treatment1}$ has $g-1$ degrees of freedom and $SS_{treatment2}$ has $b-1$ degrees of freedom!
- $SS_{intereaction}$ has $(g-1)(b-1)$ degrees of freedom!
- $SS_{residuals}$ has $gb(n-1)$ degrees of freedom!

Note that the degrees of freedom coincide with the dimension of the space where we project these vectors!

Note that if $n < 2$ we don't have degrees of freedom available for residual: n is the number of repetition for each cell!

So if $n < 2$ the model would fit perfectly all the observations with no error: no good we are over-fitting! Moreover we have no estimate of σ^2!

Thus if we have $n = 1$ we use an additive model, for which we have:

- $SS_{centered}$ has 1 degree of freedom!
- $SS_{treatment1}$ has $g-1$ degrees of freedom!
- $SS_{residuals}$ has $gb(n-1) + (g-1)(b-1)$ degrees of freedom!

So if we want to Test: $H_0 : \gamma_{ij} = 0 \; \forall i,j$ vs $H_1 : \exists \gamma_{ij} \neq 0$ Then: we need to compare $SS_{treatment}$ and $SS_{residuals}$

Are the interactions introducing enough variability so to conclude that they are there?

We reject H_0 at level $\alpha \in (0,1)$ if: $\frac{\frac{1}{(g-1)(b-1)} SS_{interaction}}{\frac{1}{gb(n-1)} SS_{residuals}} > F_\alpha((g-1)(b-1), gb(n-1))$ indeed we are taking the ratio of two chi-squared independent random variables!

If we reject it means we need to keep complete model, otherwise If we don't reject H_0 we can use the additive model: in this case we Test: $H_0 : \tau_1 = \cdots = \tau_g = 0$ vs $H_1 : \exists \tau_i \neq 0$ Then:

We reject H_0 at level $\alpha \in (0,1)$ if: $\frac{\frac{1}{g-1} SS_{treatmenet1}}{\frac{1}{gb(n-1)+(g-1)(b-1)} SS_{residuals}} > F_\alpha(g-1, gb(n-1) + (g-1)(b-1))$

Note: in the above we have seen two ways ANOVA, but everything can be generalised to $n-$ways MANOVA!

13 Lecture 23: 21st Of April 2020

Classification

It's a wide and basic topic: we observe some features of a statistical unit and we use this info to attribute this statistical unit into some class.

Each statistical unit is represented by a vector $(\underline{X}^T, L)$ where $\underline{X} = (x_1, \dots, x_p)^T \in \mathcal{X}$ is a vector of features, and for example $\mathcal{X} = \mathbb{R}^p$ Note that the features can be qualitative, quantitative or both: length, volume, colour of eyes and so on and so forth.

L is a label: it declares membership to a group, so $L \in \{1, 2, \dots, g\}$ these are labels not necessarily numbers.

Our goal is to find a classifier, a function $\delta : \mathcal{X} \to \{1, \dots, g\}$ so that given the features of a unit the classifier tells us the label of that statistical unit.

Thus we need to assign labels through a reasonable guess about membership in a group! There are two possible situations for learning:

- **Supervised situation:**
  We have available a training set, in this specific form:

$$\mathbb{X} = \begin{array}{c} \begin{matrix} x_1 & \dots & x_p & L \end{matrix} \\ \begin{pmatrix} x_{11} & \dots & x_{1p} & \ell_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & \dots & x_{np} & \ell_n \end{pmatrix} \end{array} = \begin{bmatrix} \underline{x}_1^T & \ell_1 \\ \vdots & \vdots \\ \underline{x}_n^T & \ell_n \end{bmatrix}$$

  We sample n statistical units and we can observe both the features and the labels: $\underline{x}_i \in \mathcal{X}$ and $\ell_i \in \{1, \dots, g\}$

  The goal is to learn the *optimal* (in what sense?), δ based on:

  - Modelling assumptions: the model
  - Data $\mathbb{X}$

  This activity is called **discriminant analysis** (i.e: **supervised classification**)

- **Unsupervised situation:**
  We have data available in the following form:$\mathbb{X} = \begin{bmatrix} \underline{x}_1^T & \ell_1 \\ \vdots & \vdots \\ \underline{x}_n^T & \ell_n \end{bmatrix}$, so we have observed feature for n statistical units, with $\underline{x}_i \in \mathcal{X}$ and $\ell_i \in \{1, \dots, g\}$ but the labels are hidden: we can only observe the feature $\underline{x}_i^T$

  We don't have labels either because they are not there or we can't observe them: we believe they are there but we cant observe them!

  Our goal is to:
  a)] Based on the features we want to estimate the labels $\hat{\ell_1}, \dots, \hat{\ell_g}$ and $\hat{g}$ indeed we don't know how many labels are there. This activity is called **cluster analysis:** we want to find clusters in the data.

  After having found the clusters we want to learn how to assign a new unit to a cluster based on the feature so: **b)** we find the optimal $\delta : \mathcal{X} \to \{1, \dots, \hat{g}\}$ based on:

  - Modelling assumptions: model

– Data: $\hat{\mathbb{X}} = \begin{bmatrix} \underline{x}_1^T & \hat{\ell}_1 \\ \vdots & \vdots \\ \underline{x}_n^T & \hat{\ell}_n \end{bmatrix}$

So discriminant analysis is performed after we have performed cluster analysis! Note that sometimes we stop at cluster analysis! Sometimes we use step **b)** to see how robust is our classification: can it be generalised out of the training set?

Note: In real life we have a data set for which for some units we have observed labels and for some we haven't: *semi-supervised (partially supervised) learning*

Note: In any serious clusterisation exercise we don't stop at clusters but we try to see whats makes up the clusterisation! There is a loop of supervised and un-supervised, although the labels are estimated from data.

Ingredients for supervised models for classifications

1) $\underline{X}|L = i \sim f_i(\underline{x})$ where for simplicity we assume $\underline{x} \in \mathbb{R}^p$ so $f_i : \mathbb{R}^p \to [0, \infty)$ is a density.

We assume that the distribution of the features in the various group is different: if they were not different there is no hope of associating a label to some observed features!

Example: For example the distribution of the number of eyes between males and females is the same so its a useless feature!!

We need to check that $f_i \neq f_j$: otherwise we can't use this information for saying something about the labels! Note that to do so we can use MANOVA.

2) Prior probabilities: $\mathbb{P}[L = i] = p_i \ \forall i = 1, \dots, g$ the only requirements is: $p_i \geq 0$ and $\sum_{i=1}^{g} p_i = 1$

These are dependent on the boundary conditions (context dependent): they may be different for different problems, researchers and areas.

Note: the prior distribution is chosen according to the type of problem, time of the day, by the person doing research: there is a subjectivity ingredient that people don't like but its totally natural!

3) Cost of mis-classification: what happens if we attribute a unit to group 1 when in fact the unit belongs to group 2? Whats the cost of this error?

True label / Attributed label by δ	1	2	...	g
1	$c(1\|1)$	$c(1\|2)$	...	c(1|g)
2	$c(2\|1)$	$c(2\|2)$	...	c(2|g)
...	...	...	...	...
g	$c(g\|1)$	$c(g\|2)$	...	c(g|g)

So $c(i|j)$ is the cost for attributing to group i a unit belonging to group j

Note that we assume $c(i|j) \geq 0 \ \forall i, j$ and $c(i|i) = 0 \ \forall i$

Moreover we are **not** requiring symmetry: $c(i|j) = c(j|i)$

Note: in some cases we have data to estimate the prior probabilities!

Optimality criterion for choosing δ

If $\delta : \mathcal{X} \to \{1, \cdots, g\}$ then we can define: $R_i = \{\underline{x} \in \mathcal{X} : \delta(\underline{x}) = i\} = \delta^{-1}(i)\}$ so its the set of all the features mapped by δ into i

We would like R_i to be measurable so we would like δ to be measurable so suppose they both are Borel measurable.

Note: $\{R_1, \ldots, R_g\}$ is a (finite and measurable) partition of $\mathcal{X}$ thus: $R_i \cap R_j = \emptyset \; \forall i \neq j$ and $\bigcup_{i=1}^{g} R_i = \mathcal{X}$

Conclusion: specifying δ is equivalent to specifying $\{R_i\}$

For simplicity assume $\mathcal{X} = \mathbb{R}^p$ so we have an euclidean space and $g = 2$ so we have a dichotomous problem.

In this case $\delta : \mathbb{R}^p \to \{1, 2\}$ so $R_1 = \{\underline{x} \in \mathbb{R}^p : \delta(\underline{x}) = 1\}$ while $R_2 = \{\underline{x} \in \mathbb{R}^p : \delta(\underline{x}) = 2\} = R_1^c$

For a given delta we want to specify the **Expected Cost for Mis-classification (ECM)**, which we want then to minimise and find the delta that minimises it!

We have that: $ECM(\delta) = \int_{R_2} c(2|1) f_1(\underline{x}) p_1 d\underline{x} + \int_{R_1} c(1|2) f_2(\underline{x}) p_2 d\underline{x}$

Our goal is to solve the following optimisation problem:

$$\text{Find } \delta \text{ that is, find: } R_1, R_2 \text{ which minimises } ECM(\delta)$$

Here the unknowns are the Borel sets on which we want to compute the integral!

We now manipulate a bit the expression of the **ECM** so that:

$$ECM(\delta) = \int_{\mathbb{R}^p} c(2|1) f_1(\underline{x}) p_1 d\underline{x} - \int_{R_1} c(2|1) f_1(\underline{x}) p_1 d\underline{x} + \int_{R_1} c(1|2) f_2(\underline{x}) p_2 d\underline{x} =$$

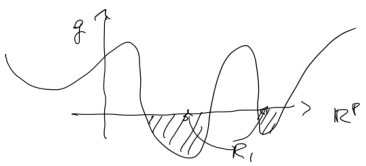
$$= c(2|1) p_1 - \int_{R_1} c(2|1) f_1(\underline{x}) p_1 d\underline{x} + \int_{R_1} c(1|2) f_2(\underline{x}) p_2 d\underline{x} =$$

$$= c(2|1) p_1 + \int_{R_1} \left[c(1|2) f_2(\underline{x}) p_2 - c(2|1) f_1(\underline{x}) p_1 \right] d\underline{x}$$

How to we identify R_1 such that $ECM(\delta)$ is minimised?

Suppose $g(\underline{x}) = [c(1|2) f_2(\underline{x}) p_2 - c(2|1) f_1(\underline{x}) p_1]$ so: $ECM(\delta) = c(2|1) p_1 + \int_{R_1} g(\underline{x}) d\underline{x}$

Note that we can only choose R_1 How should we choose it? We want to integrate g on where g is negative so to reduce the fixed uncontrollable cost of $ECM(\delta)$



Then the optimal R_1 is given by:

$$R_1 = \{\underline{x} \in \mathbb{R}^p : g(\underline{x}) \leq 0\} = \{\underline{x} \in \mathbb{R}^p : c(1|2)f_2(\underline{x})p_2 - c(2|1)f_1(\underline{x})p_1 \leq 0\} = \{\underline{x} \in \mathbb{R}^p : c(1|2)f_2(\underline{x})p_2 \leq c(2|1)f_1(\underline{x})p_1\}$$

So that:

$$R_2 = R_1^c = \{\underline{x} \in \mathbb{R}^p : c(2|1)f_1(\underline{x})p_1 < c(1|2)f_2(\underline{x})p_2\}$$

Note: Note that in R_2 we have $<$ and in R_1 $\leq$ it doesn't make a difference since the data we assumed to be in $\mathbb{R}^p$ so its continuous! It would make a difference on discrete data!!

Conclusion: we have characterised R_1, R_2 and so the optimal classifier is: $\delta(\underline{x}) = \begin{cases} 1 \text{ if } \underline{x} \in R_1 \\ 2 \text{ if } \underline{x} \in R_2 \end{cases}$ where R_1, R_2 are specified above.

We see that we attribute to class 1 those statistical unit for which we will have to pay the least if we make a mistake:
a mistake in the sense of attributing a unit to group 1 when in fact it belong to group 2

General case: we have $g \geq 2$ groups

δ is equivalent to $\{R_1, \ldots, R_g\}$ partition of $\mathbb{R}^p$ We want to compute the expected cost of mis-classification:

$$ECM(\delta) = \sum_{k=2}^{g} \int_{R_k} c(k|1)f_1(\underline{x})p_1 d\underline{x} + \sum_{k\neq 2}^{g} \int_{R_k} c(k|2)f_2(\underline{x})p_2 d\underline{x} + \cdots + \sum_{k=1}^{g-1} \int_{R_k} c(k|g)f_g(\underline{x})p_g d\underline{x} =$$

$$= \int_{R_1} \sum_{k\neq 1} c(1|k)f_k(\underline{x})p_k d\underline{x} + \int_{R_2} \sum_{k\neq 2} c(2|k)f_k(\underline{x})p_k d\underline{x} + \cdots + \int_{R_g} \sum_{k\neq g} c(g|k)f_k(\underline{x})p_k d\underline{x}$$

Now we want to find $R_1, \ldots, R_g$ which minimises $ECM(\delta)$ and thus:

$$R_1 = \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k\neq 1} c(1|k)f_k(\underline{x})p_k \leq \sum_{k\neq j} c(j|k)f_k(\underline{x})p_k, j = 2, \ldots, g \right\}$$

$$R_2 = \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k\neq 2} c(2|k)f_k(\underline{x})p_k \leq \sum_{k\neq j} c(j|k)f_k(\underline{x})p_k, j = 1, \ldots, g \text{ and } j \neq 2 \right\}$$

In general:

$$R_i = \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k\neq i} c(i|k)f_k(\underline{x})p_k \leq \sum_{k\neq j} c(j|k)f_k(\underline{x})p_k, j = 1, \ldots, g \text{ and } j \neq i \right\}$$

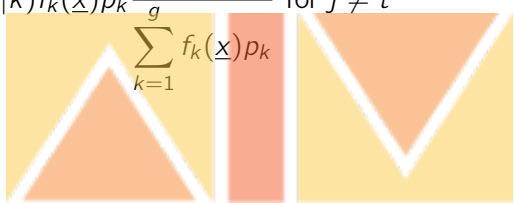
So the optimal classifier is: $\delta(\underline{x}) = i \iff \underline{x} \in R_i$ with those R_i defined above!

Note that the inequality defining membership to a class is given by:

$$\delta(\underline{x}) = t \text{ for } t \in \{1, \ldots, g\} \iff \sum_{k\neq t} c(t|k)f_k(\underline{x})p_k \leq \sum_{k\neq j} c(j|k)f_k(\underline{x})p_k \text{ for } j \neq t$$

Now we divide both sides for $\sum_{k=1}^{g} f_k(\underline{x})p_k > 0$, so we assume its not zero obviously! We get:

$$\delta(\underline{x}) = t \text{ for } t \in \{1, \ldots, g\} \iff \sum_{k\neq t} c(t|k)f_k(\underline{x})p_k \frac{1}{\sum_{k=1}^{g} f_k(\underline{x})p_k} \leq \sum_{k\neq j} c(j|k)f_k(\underline{x})p_k \frac{1}{\sum_{k=1}^{g} f_k(\underline{x})p_k} \text{ for } j \neq t$$



Now note that: $\sum_{k=1}^{g} f_k(\underline{x})p_k = \sum_{k=1}^{g} \mathbb{P}[\underline{X} = \underline{x}|L = k]\mathbb{P}(L = k) = \mathbb{P}[\underline{X} = \underline{x}]$ from the law of total probability.

Note: the above holds morally because we have continuous distribution!

Moreover we have that:

$$\frac{f_k(\underline{x})p_k}{\sum_{k=1}^{g} f_k(\underline{x})p_k} = \mathbb{P}[\underline{X} = \underline{x}|L = k]\mathbb{P}[L = k]\frac{1}{\mathbb{P}[\underline{X} = \underline{x}]} = \mathbb{P}[\underline{X} = \underline{x}, L = k]\frac{1}{\mathbb{P}[\underline{X} = \underline{x}]} = \mathbb{P}[L = k|\underline{X} = \underline{x}]$$

From Bayes Theorem!

Note: the above holds morally because we have continuous distribution!

Thus we have that:

$$\delta(\underline{x}) = t \text{ for } t \in \{1, \dots, g\} \iff \sum_{k \neq t} c(t|k)\mathbb{P}(L = k|\underline{X} = \underline{x}) \leq \sum_{k \neq j} c(j|k)\mathbb{P}(L = k|\underline{X} = \underline{x}) \text{ for } j \neq k$$

This one above is another description of the optimal classifier: the first term of the inequality is the expected posterior cost for group t

---

Now we make a big assumption: suppose all the costs are the same, namely: $c(i|j) = const > 0 \; \forall i \neq j$ then:

$$\delta(\underline{x}) = t \text{ for } t \in \{1, \dots, g\} \iff \sum_{k \neq t} \mathbb{P}(L = k|\underline{X} = \underline{x}) \leq \sum_{k \neq j} \mathbb{P}(L = k|\underline{X} = \underline{x}) \text{ for } j \neq k$$

Note: here we have summations only because the labels are discrete! But the features are continuous so saying: $\mathbb{P}(L = k|\underline{X} = \underline{x})$ is abuse of language so we should write $P(L = k|\underline{X} = \underline{x})d\underline{x}$, so again as we said above it only holds morally!

Applying basic laws of probability the above expression is:

$$\delta(\underline{x}) = t \text{ for } t \in \{1, \dots, g\} \iff 1 - \mathbb{P}(L = t|\underline{X} = \underline{x}) \leq 1 - \mathbb{P}(L = j|\underline{X} = \underline{x}) \text{ for } j \neq k$$

This this becomes:

$$\delta(\underline{x}) = t \text{ for } t \in \{1, \dots, g\} \iff \mathbb{P}(L = j|\underline{X} = \underline{x}) \leq \mathbb{P}(L = t|\underline{X} = \underline{x}) \text{ for } j \neq k$$

We can see that we attribute the labels to the group which maximises the posterior probability!

This classifier we just obtained assuming constant (and equal) costs of mis-classification is called **Bayes Classifier** which is a special case of the optimum classifier!

Note: the assumption behind **Bayes Classifier** might not be right for our problem! Each problem has a need for a different type of classifier!

Making a different assumption we obtain a different classifier: assume $c(i|j) = const > 0 \forall i \neq j$ and assume the priors are the same: $p_1 = \dots = p_g = \frac{1}{g}$ then:

Since $\mathbb{P}(L = j|\underline{X} = \underline{x}) = \frac{f_j(\underline{x})p_j}{\mathbb{P}[\underline{X} = \underline{x}]}$ thus we have:

$$\delta(\underline{x}) = t \text{ for } t \in \{1, \dots, g\} \iff \frac{f_j(\underline{x})p_j}{\mathbb{P}[\underline{X} = \underline{x}]} \leq \frac{f_t(\underline{x})p_t}{\mathbb{P}[\underline{X} = \underline{x}]} \text{ for } j \neq k$$

Thus we have that:

$$\delta(\underline{x}) = t \text{ for } t \in \{1, \dots, g\} \iff f_j(\underline{x}) \leq f_t(\underline{x}) \text{ for } j \neq k$$

We can see that in this we attribute to the group for which the likelihood of what we have observed is maximised.

This classifier we just obtained is called **MLE Classifier**!

Note that we don't specify neither the costs nor the priors but only the density of the features in the different groups!

Note that its not that we don't specify them: it's just that since we assume they are equal they simplify!

Note: not every classification problem can be solved under such assumptions!

We need to be careful: does the patient have flu? We might assume that the probability of having flu is the same of not having flu. What happens? Do we set both prior equal to 50%? NO!

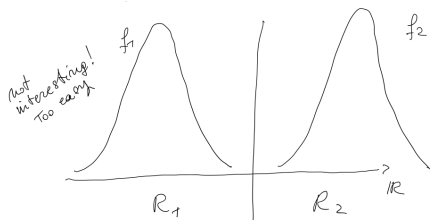
Be careful using a classifier: looks like we don't need to classify anything but its not true!

Where do we get f_j? From the training set!

Where do we get the prior from? Not necessarily from the training set!

Where do we get the cost from? From the problem-domain knowledge and not from the training set!

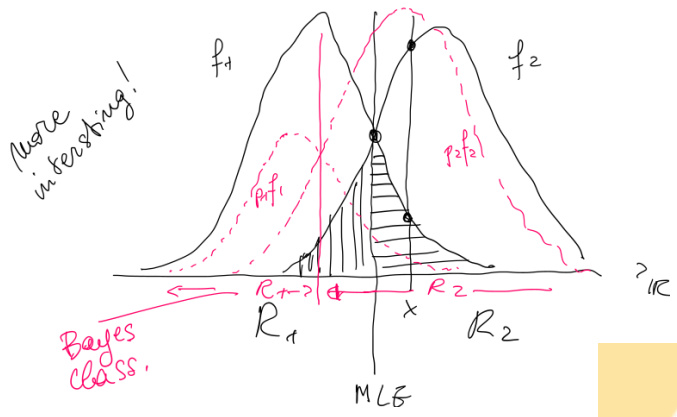
Example: Suppose that $g = 2$ and $p = 1$



This is easy situation and we aren't interested: the features in group 1 are not showing in group 2 and vice-versa!

Indeed saying that all males have beard and all woman don't is not interesting!

Consider the following situation:



This is interesting.

In the figure we see the partition we would get from the **MLE Classifier**

It's like saying that if we see a very tall person we suppose its male! Whats the error? The integral of f_1 over R_2 plus the integral of f_2 over R_1, represented as the shaded area in the figure.

Suppose we have now a prior: we pick individuals at random in the mechanical engineering department: 80% are males.

So the prior is different because we are considering a specific population with the proportion between male and female different from the proportion in the true whole population!

So we multiply the two previous densities by p_1, p_2 the density are now re-scaled: see again the figure above and in the dashed pink line we can see the **Bayes Classifier**!

Note that the **Bayes Classifier** has has moved the threshold of the **MLE Classifier** to the left! The prior has modified the density and we are more conservatives and we make less errors!

So now if we take different cost of mis-classification it will have some effect: we weight again differently the two densities.

That's why prior and costs are important: the same problem in a different context has different prior and costs but same densities f_1, f_2

Example: the prior distribution of how people are dangerous at night is higher than in the morning!

We don't move the data only the prior: just by doing this we change the way we classify the same individual!

14 Lecture 24: 23rd Of April 2020

We have seen that the **optimal classifier** which minimises **ECM** is given by:

$$\delta(\underline{x}) = i \text{ if } \underline{x} \in R_i \text{ with } i = 1, \dots, g \text{ with: } R_i = \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k \neq i} c(i|k) f_k(\underline{x}) p_k \leq \sum_{k \neq j} c(j|k) f_k(\underline{x}) p_k \text{ for } j = 1, \dots, g \right\}$$

The R_i we have seen can be re-written as:

$$R_i = \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k \neq i} c(i|k) \mathbb{P}[L = k | \underline{X} = \underline{x}] \leq \sum_{k \neq j} c(j|k) \mathbb{P}[L = k | \underline{X} = \underline{x}] \text{ for } j = 1, \dots, g \right\}$$

Observation:

1) We need to specify the cost up to a multiplicative constant, which is greater than zero. Indeed the unit of measure doesn't matter!

   So if we multiply everything by a constant, which is greater than zero, the inequality (and so R_i) don't change!

2) The general optimal classier above its a **Bayesian Classifier** and its sort of a reference classifier!

   Indeed we compute the posterior probabilities and then we compute the expected cost with respect to this posterior probability.

Suppose now that $c(i|j) = const > 0 \forall i \neq j$ then we have the **Bayes Classifier** characterised by:

$$R_i = \{ \underline{x} \in \mathbb{R}^p : \mathbb{P}[L = i | \underline{X} = \underline{x}] \geq \mathbb{P}[L = j | \underline{X} = \underline{x}] \text{ for } j = 1, \dots, g \}$$

so we attribute $\underline{x}$ to the label which has the maximum posterior probability!

The **Bayes Classifier** is more flexible than what it looks like, and it does not forget about the costs!

For instance consider the optimal classifier with $c(i|k) = c_k \geq 0$ with $i, k = 1, \dots, g$ so have same cost for each group! Then:

$$R_i = \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k \neq i} c_k f_k(\underline{x}) p_k \leq \sum_{k \neq j} c_k f_k(\underline{x}) p_k \text{ for } j = 1, \dots, g \right\} =$$

$$= \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k \neq i} c_k f_k(\underline{x}) p_k \frac{1}{\sum_j c_j p_j} \leq \sum_{k \neq j} c_k f_k(\underline{x}) p_k \frac{1}{\sum_j c_j p_j} \text{ for } j = 1, \dots, g \right\}$$

Note: $\sum_j c_j p_j > 0$ as indeed at least one cost is positive, so we don't divide by zero and we don't change the sign of the inequality!

Now set $\pi_k = c_k p_k \frac{1}{\sum_j c_j p_j}$ with $k = 1, \dots, g$ these act as prior distributions, indeed:

- $\pi_k \geq 0$
- $\sum_k \pi_k = 1$

So we can re-write the above as follows:

$$R_i = \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k \neq i} f_k(\underline{x})\pi_k \leq \sum_{k \neq j} f_k(\underline{x})\pi_k \text{ for } j = 1, \ldots, g \right\}$$

So its like we still have a Bayes Classifier but with π_k as priors!

So if we have cost, we can modify the priors to take into account the costs and then we still get a Bayes classifier! Indeed cost and prior play a very similar role!

Note: we also workout the optimal classifier when:

- $c(i|k) = c_i > 0$, for $i, k = 1, \ldots, g$ Here we change the threshold for the posterior probability for deciding when a unit belongs to a group.
- $c(i|k) = c_i h_k$ for $i, k = 1, \ldots, g$ Here we change both the prior and the threshold!

In each case we get a slight modification of the Bayes classifier!

We could even use thing as complicated as $c(i|k) = c^{\alpha_i} b^{\beta_k}$ we just have to change the cost structure!

Conclusion: Even if we use the Bayes classifier as a reference, we can introduce easily a cost structure as long as the cost structure follows some factorisation principle!

Special cases of a Bayes Classifier: Assume that the priors are Gaussian: $\underline{X}|L = i \sim \mathcal{N}_p(\underline{\mu}_i, \Sigma_i)$ for $i = 1, \ldots, g$ Then:

$$\mathbb{P}[L = i|\underline{X} = \underline{x}] \geq \mathbb{P}[L = j|\underline{X} = \underline{x}] \implies f_i(\underline{x})p_i \frac{1}{\mathbb{P}[\underline{X} = \underline{x}]} \geq f_j(\underline{x})p_j \frac{1}{\mathbb{P}[\underline{X} = \underline{x}]} \implies f_i(\underline{x})p_i \geq f_j(\underline{x})p_j \implies$$

$$\frac{p_i}{\sqrt{(2\pi)^p det(\Sigma_i)}} \exp\left(-\frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1}(\underline{x} - \underline{\mu}_i)\right) \geq \frac{p_j}{\sqrt{(2\pi)^p det(\Sigma_j)}} \exp\left(-\frac{1}{2}(\underline{x} - \underline{\mu}_j)^T \Sigma_j^{-1}(\underline{x} - \underline{\mu}_j)\right)$$

Taking the log:

$$\log(p_i) - \frac{1}{2}\log(det(\Sigma_i)) - \frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1}(\underline{x} - \underline{\mu}_i) \geq \log(p_j) - \frac{1}{2}\log\big(det(\Sigma_j)\big) - \frac{1}{2}(\underline{x} - \underline{\mu}_j)^T \Sigma_j^{-1}(\underline{x} - \underline{\mu}_j)$$

Definition: $d_i^Q : \mathbb{R}^p \to \mathbb{R}$ with: $d_i^Q(\underline{x}) = \log(p_i) - \frac{1}{2}\log(det(\Sigma_i)) - \frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1}(\underline{x} - \underline{\mu}_i)$ for $i = 1, \ldots, g$ are called **Quadratic Discriminant Scores Functions**

In this case the optimal Bayes Classifier (QDA) is defined by:

$$\delta(\underline{x}) = i \text{ if } \underline{x} \in R_i \text{ where } i = 1, \ldots, g \text{ and } R_i = \{\underline{x} \in \mathbb{R}^p : d_i^Q(\underline{x}) \geq d_j^Q(\underline{x}) \text{ for } j = 1, \ldots, g\}$$

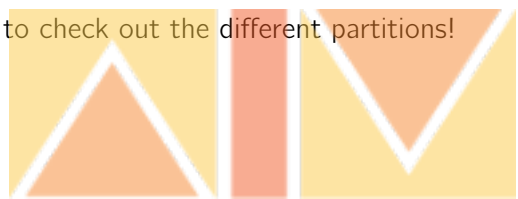
This is called **Quadratic Discriminant Analysis**

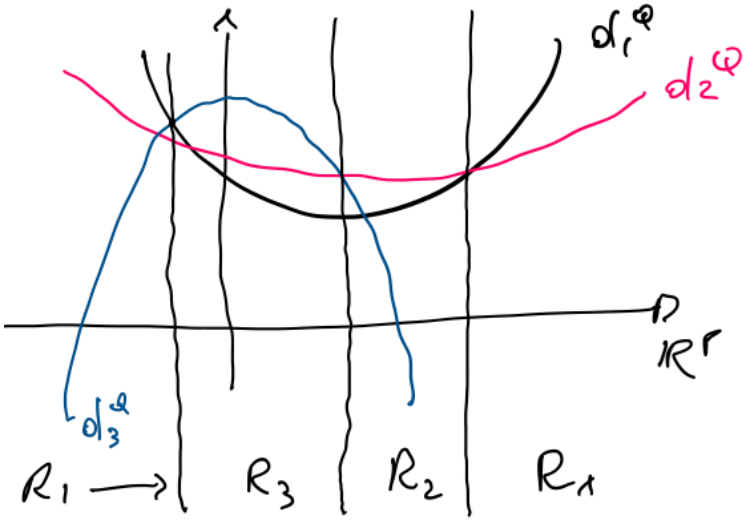
In this case the distribution of the features is Gaussian in each group! Since it's a Bayes Classifier all cost are equal, but as seen above we can have special cost structure!

Note: This is very used: we only need to estimate the mean and Σ_i for each group!

Example: Consider the figure below: we take the maximum of the curves in order to define the three regions R_1, R_2, R_3

Note: if we are in a space of dimensions more than 3 we consider the boundaries to check out the different partitions!





Note: the fact that **QDA** is quadratic comes from the term $\frac{1}{2}(\underline{x}-\underline{\mu}_i)^T\Sigma_i^{-1}(\underline{x}-\underline{\mu}_i) = d_{\Sigma^{-1}}(\underline{x},\underline{\mu}_i)^2$ Therefore:
QDA says: the closest (with respect to the Mahalanobis Distance) you are to mean of $\underline{\mu}_i$ the more probable your label is from that class!

Moreover since the above term has a minus before it in the expression of d_i^Q we have that the greater the distance the smaller d_i^Q!

Consider the **QDA** classifier but suppose also that: $\Sigma_1 = \cdots = \Sigma_g = \Sigma$ then $det(\Sigma)$ are all the same and so they simplify:

$$d_i^Q(\underline{x}) \geq d_j^Q(\underline{x}) \implies \log(p_i) - \frac{1}{2}(\underline{x}-\underline{\mu}_i)^T\Sigma^{-1}(\underline{x}-\underline{\mu}_i) \geq \log(p_j) - \frac{1}{2}(\underline{x}-\underline{\mu}_i)^T\Sigma_i^{-1}(\underline{x}-\underline{\mu}_i)$$

Now note that: $\frac{1}{2}(\underline{x}-\underline{\mu}_i)^T\Sigma^{-1}(\underline{x}-\underline{\mu}_i) = -\frac{1}{2}\underline{x}^T\Sigma^{-1}\underline{x} + \underline{\mu}_i^T\Sigma^{-1}\underline{x} - \frac{1}{2}\underline{\mu}_i^T\Sigma^{-1}\underline{\mu}_i$

Since the first term (the quadratic one) is the same in the above inequality, in both sides, it cancels out so we have:

$$\log(p_i) + \underline{\mu}_i^T\Sigma^{-1}\underline{x} - \frac{1}{2}\underline{\mu}_i^T\Sigma^{-1}\underline{\mu}_i \geq \log(p_j) + \underline{\mu}_j^T\Sigma^{-1}\underline{x} - \frac{1}{2}\underline{\mu}_j^T\Sigma^{-1}\underline{\mu}_j$$

Definition: $d_i : \mathbb{R}^p \to \mathbb{R}$ defined as: $d_i(\underline{x}) = \log(p_i) + \mu_i^T\Sigma^{-1}\underline{x} - \frac{1}{2}\underline{\mu}_i^T\Sigma^{-1}\underline{\mu}_i$ are linear in $\underline{x}$ and are called **Linear Discriminant Score Functions**

In this case the above reduces to: $d_i(x) \geq d_j(x)$ and we have that the optimal classifier, which is called **Linear Discriminant Analysis (LDA)**, is given by:

$$\delta(\underline{x}) = i \text{ if } \underline{x} \in R_i \text{ with } i, = 1, \ldots, g \text{ and } R_i = \{\underline{x} \in \mathbb{R}^p : d_i(\underline{x}) \geq d_j(\underline{x}) \text{ for } j = 1, \ldots, g\}$$

This is very robust, also if data are not really Gaussian! Moreover next lecture we will see that we can get **LDA** without assuming that the distributions in the groups are Gaussian!

Conclusion:

- **QDA** works less well: we need Gaussian data otherwise its weak! We need to assume that the only difference is on the variance matrix!
- **LDA**: we assume that the only difference is upon the mean so it works well!

Now use training data to estimate all the parameters that enter into the model!

For example we use the training set to estimate the mean and the co-variances for **LDA-QDA**

Otherwise we use training set for estimating distribution, or if we have a preferred distribution we fit it to the distribution!

Note: We can't use training set to estimate prior unless the proportion of units of group i in the training set is an estimate to the proportion of units of group i in the true population!

Usually training set are not set up like above: we should sample at random to do the above, but instead we build training set so that all groups are well represented!

So many times the proportion of units belonging to group i in the training set doesn't correspond to the same thing in the population!

For example if we want to diagnose a person with yellow fever, if we sample from 10000 people in Milan we probably get 1 person with yellow fever! But we can't discriminate with one unit out of 10000 in the training set!

So many times if we make the training set representative of the population it's useless!

Thus what is done is build up training set with 5000 people ill and 5000 people not ill, so that we get some difference! This is not representative of the population, thus if we use this training set to estimate prior it's wrong!

Indeed the above would be like saying that half of Milan population has yellow fever! It's bullshit!

Thus estimating priors with respect to the proportion in the training set is dangerous and most time wrong! Note that R by default does this, so watch out!

We need to decide good prior without looking at proportion in the training set!

The training set can't be a random sample from population: we want to maximise the chances of distinguishing between two classes so we need a $50-50$ balance in the training set, so that it's not representative of the population!

Note that priors are either subjective or we can use the incidence of a illness in the population! We need to estimate the proportion of classes, and prior probabilities: these are two different things we need to do!

Now we want to talk about estimating parameters with the training set: $\mathbb{X} = [\underline{x}_1^T \ell_1, \dots, \underline{x}_n^T, \ell_n]^T$ with $\underline{x}_i \in \mathbb{R}^p$ and $\ell_i \in \{1, \dots, g\}$ Then:

- For **QDA**:
  we use the sample mean: $\hat{\underline{\mu}}_k = \dfrac{1}{n_k} \displaystyle\sum_{i:\ell_i=k} \underline{X}_i = \bar{\underline{X}}_k$ where $n_k = \#\{i = 1, \dots, n : \ell_i = k\}$
  we use the sample co-variance: $\hat{\Sigma}_k = \dfrac{1}{n_k - 1} \displaystyle\sum_{i:\ell_i=k} (\underline{X}_i - \bar{\underline{X}}_k)(\underline{X}_i - \bar{\underline{X}}_k)^T = S_k$
- For **LDA**:
  we use the sample mean: $\hat{\underline{\mu}}_k = \bar{\underline{X}}_k$ where $n_k = \#\{i = 1, \dots, n : \ell_i = k\}$
  we need just one sample co-variance: $\hat{\Sigma} = \dfrac{1}{n-g} \displaystyle\sum_{k=1}^{g} (n_k - 1) S_k$ which is the pooled estimator for the co-varinace!

Problem: to compute the **QDA-LDA** we need to invert matrices! If the sample co-variance are hard to invert, then we have a problem!

If p is large with respect to n then the sample co-variance can be singular! Moreover if we also have sparse features, due to missing data, then it's almost impossible to compute S_i^{-1}

Note: Indeed the deviation vectors would be linearly dependent if $p > n$!

Thus the only thing we can do here is assume a (parametric) model for the co-variances to reduce dimensionality of the problem!

Thus we need to suppose that Σ_i depend only on few parameters! For example we can assume that in each group the components of $\underline{x}$ are independent!

This is a strong assumption, which results in having: $\Sigma_k = diag(\sigma_{11}^{(k)}, \dots, \sigma_{pp}^{(k)})$ so that we get the **Naive Bayes Classifier:**

$$\text{for } k = 1, \dots, g \text{ we have: } d_k^Q(\underline{x}) = \log(p_k) - \frac{1}{2}\log(det(\Sigma_k)) - \frac{1}{2}\sum_{j=1}^{p}(\underline{x}_j - \underline{\mu}_{kj})^2 \frac{1}{\sigma_{jj}^{(k)}}$$

where μ_{kj} is the $j-$th component of μ_k

So here we need to estimate μ_k with the sample mean! Moreover: $\sigma_{jj}^{(k)} = \frac{1}{n_k - 1}\sum_{i:\ell_i = k}(x_{ij} - \bar{X}_{kj})^2$ with $j = 1, \dots, p$ is estimated with the sample co-variance.

Note: the above is doable even if $p >> n$!

Note that we need at least two observation in each group for each component otherwise we can't estimate the variance!

The above as we said is the **Naive Bayes (Gaussian) Classifier** where **Naive** comes from the fact that we assume independence: indeed we are naive assuming that the components are independent!

Note that we can use a different distribution instead of a Gaussian, for example we can construct a **Naive Bayes (Poisson) Classifier**!

Indeed we can construct any from any uni-variate distribution!

Note: even if we know the assumption are wrong this classier is not that bad!

Note: Moreover we can also assume to have the same $\sigma_{ii}^{(k)}$

Note: doing PCA first and then classification: this is a bad recipe.

Given some classifier how do we evaluate how good it is? We estimate the error rate!

suppose $\delta : \mathbb{R}^p \to \{1, \dots, g\}$ is a classifier then the actual error rate of δ **AER** is given by:

$$AER(\delta) = \sum_{k \neq 1}\int_{R_k} f_1(\underline{x})p_1 d\underline{x} + \sum_{k \neq 2}\int_{R_k} f_2(\underline{x})p_2 d\underline{x} + \dots + \sum_{k \neq g}\int_{R_k} f_g(\underline{x})p_g d\underline{x}$$

So the **AER** is the overall proportion of mistakes the classifier makes when applied to any unit in the population!

The problem is that we can't compute it since we don't know the right prior and densities, so we need to estimate it using the estimate of prior and density estimate from the training set.

People don't like doing the above since they prefer to estimate the **AER** using an objective error rate, and they want to do it non-parametrically! So we want to calculate the **APER**

Suppose we have two groups to simplify calculations, then:

$$AER(\delta) = \int_{R_2} f_1(\underline{x}) p_1 d\underline{x} + \int_{R_1} f_2(\underline{x}) p_2 d\underline{x}$$

Now we compute the **confusion matrix** by applying δ to the training set:

Actual label \ Attributed Labbel	L_1	L_2
L_1	n_{11}	n_{12}
L_2	n_{21}	n_{22}

where: $n_{11} + n_{12} + n_{21} + n_{22} = n$ with $n_{11} + n_{12} = n_1$ and $n_{21} + n_{22} = n_2$

So the **Apparent Error Rate (APER)** is given by:

$$APER(\delta) = \frac{n_{12} + n_{21}}{n}$$

So it's the number of mistakes over the total trials we make.

This is bad since it too optimistic: we learn a classifier from the training set and we applied it to the training set to calculate **APER**! So its good on the training set since we minimised over it to get our classifier!

Therefore we need to see the error in the population, over a new unit!

Now note that:

$$APER(\delta) = (n_{12} + n_{21})\frac{1}{n} = \frac{n_1}{n} \cdot \frac{n_{12}}{n_1} + \frac{n_2}{n} \cdot \frac{n_{21}}{n_2} = \hat{p}_1 \int_{R_2} \widehat{f_1(\underline{x})} d\underline{x} + \hat{p}_2 \int_{R_1} \widehat{f_2(\underline{x})} d\underline{x}$$

where for example $\hat{p}_1$ is the proportion of units in group 1 in the training set, which is an estimate of p_1

But we said we never assumed to have random sample from population so this is no good, indeed the frequencies in the training set are no estimate for the prior since the training set is not representative of the population!

Thus we need a better estimate of the **AER:** we use leave-one-out cross validation: we use some data for training and some which has not been used for training we use it for testing, so that we remove the above optimistic bias!

How to use training set both for learning and for testing?

- we take one statistical unit and we remove it from the training set
- we train the classifier without that statistical unit
- we use this classifier to check the error and we check it's error on the statistical unit we left out
- we repeat this for all statistical unit!

This can be see as the following algorithm:
for all the units from $i = 1, \dots, n$ repeat:

1) take unit i out of the training set: $\mathbb{X}_{-i} = [x_1^T \ell_1, \dots, x_n^T \ell_n]^T$ so it's $\mathbb{X}$ without the $i-$th row!
2) we train the classifier δ on $\mathbb{X}_{-i}$ from which we get: $\delta_{-i} : \mathbb{R}^p \to \{1, \dots, g\}$

   δ_{-i} is very similar to δ, which is the one trained on the whole data!

   It's the same classifier trained on two different set, so it differs only for one unit!

   Thus if the result of δ_{-i} would be very different from the one of δ then the unit i is an outlier since we wouldn't have a robust classifier!
3) we apply δ_{-i} to $\underline{x}_i$ So we compute the estimated label: $\delta_{-i}(\underline{x}_i) = \hat{\ell}_i$
4) we check the error: $\epsilon_i = \begin{cases} 1 \text{ if } \hat{\ell}_i \neq \ell_i \\ 0 \text{ otherwise} \end{cases}$

Finally we estimate **AER** as:

$$\frac{1}{n} \sum_i \epsilon_i = AE\hat{R}(\delta)$$

which is the **leave one out estimate (L10 or LOO)** for the **AER**

This way we use all the data we have and we check error on a unit that has not been used in training!

Note: $A\hat{E}R$ is an estimate of **AER** for δ trained on the whole data!

There is a problem Leave-one-out is computationally expensive! Moreover there is too much variability: changing the training set yield different estimate!

The solution is to use $k-$fold cross validation: we leave out k observation at the time!

Note: **LOO** is the same as $n-$fold cross validation!

15 Lecture 25: 24th Of April 2020

Leave-one-out cross validation procedure for estimating the Actual Error Rate has small bias but high variance, thus in order to reduce the variance without scarificing too much of the bias, because of the bias-variance trade-off, we want to use cross validation in a different setting!

We try $k-$fold cross validation: note that this is a computer intensive technique and prior to using it we need to choose k!

0) we set $k < n$ usually $k = 5, 10$ for n sample size large enough. Then we randomly split the units of the training set in k parts.

   We take $\mathbb{X}$ and randomise it: we permute the rows of it (note there are $n!$ permutations so we choose one at random) and then we split in k parts:

   Note: if we didn't permute the rows it might happen that the very first 10 rows are all related to units belonging to the same group and so on and so forth: rubbish procedure!

- Then for $j = 1, \dots, k$

  1) we first hold out part j from the training set and so we build $\mathbb{X}_{-\text{part } j}$ which is $\mathbb{X}$ except that part j is missing.
  2) we train the classifier δ on this new training set $\mathbb{X}_{-\text{part } j}$ and so we get: $\delta_{-\text{part}_j} : \mathbb{R}^p \to \{1, \dots, g\}$
  3) we apply this classifier $\delta_{-\text{part}_j}$ to the unit that has been hold out, namely part_j so that $\delta(-\text{part}_j)(\text{part}_j)$

  Now we count the errors: $\frac{1}{n_j} \sum_{i \in \text{part}_j} \epsilon_i = Err_j$ This is the error rate computed when we apply the classifier on part j where $n_j = \#\text{part}_j = \#\{i \in \{1, \dots, n\} : i \in \text{part}_j\}$ where $\epsilon_i = \begin{cases} 1 \text{ if } \delta_{-\text{part}_j}(x_i) \neq \ell_i \text{ i.e: we make an error} \\ 0 \text{ otherwise} \end{cases}$

  End for

4) we estimate $AER(\delta)$ by taking the mean of all the error rates: $\hat{AER}(\delta) = \frac{1}{n} \sum_{j=1}^{k} n_j Err_j$

   Note that this may be a weighted mean because maybe we couldn't split the data set in k parts exactly equal!

Note: if $k = n$ then we have LOO (L1O) cross-validation, moreover note that we can use L1O if we have small data set!

$k-$fold cross-validation is more flexible than L1O cross-validation:
we get an estimate of the actual error rate and we can also compute its variability so we know the uncertainty about this estimate! This can't be done with L10!

Indeed we can initialise $k-$fold cross-validation in different ways B times, where each time we select, at random, a different permutation of the rows of the training set, before splitting!

This way we get B estimate of the **AER!** Each time we get an estimate so we end up with: $\hat{AER}_1(\delta), \dots, \hat{AER}_B(\delta)$

Then we get an estimate of the variability, indeed computing the mean of the above we have:

$$\hat{AER}_m(\delta) = \frac{1}{B} \sum_{b=1}^{B} \hat{AER}_i(\delta)$$

which is an estimate of $\mathbb{E}[A\hat{E}R(\delta)]$

Now we can compute its variance:

$$Var(A\hat{E}R(\delta)) = \frac{1}{B-1}\sum_{b=1}^{B}(A\hat{E}R_i(\delta) - A\hat{E}R_m(\delta))^2$$

Now we can compute confidence intervals indeed, using CLT we have:

$$CI_{1-\alpha}(E[A\hat{E}R(\delta)]) = \left[A\hat{E}R_m(\delta) \pm z_{\alpha/2}\sqrt{Var[A\hat{E}R(\delta)]\frac{1}{B}}\right]$$

Note that in order to use the CLT we need to choose B large enough!

Note that in the above $z_{\alpha/2}$ corresponds to having an area of $\alpha/2$ to the right of $z_{\alpha/2}$

Note that we can choose B as big as we want as long as it's less than $n!$, since we only have $n!$ different permutations of the rows of the data set!

Note that the above doesn't work in L1O cross-validation because even permuting rows we get the same result!

So why does $k-$fold cross validation reduces the variability?

- LOO is based on the average: Consider δ_{-i} with $i = 1, \dots, n$ Then these n classifiers are much correlated because are all based on the same training set expect for one observation!

  So by taking the average we don't reduce the variability: it would only work if we have independent observations but these are instead strongly correlated!

  Indeed: $Var\left(\frac{1}{n}\sum_i \epsilon_i\right) \approx Var(\epsilon_j)$

- With $k-$fold cross-validation we create k classifiers, which are less correlated since they are based on training sets that are more different, since each time we hold out k observations, so that there is more chance for each classifier $\delta_{-\text{part}_j}$ to be less correlated!

  Thus: $Var\left[\frac{1}{n}\sum_{j=1}^{k} n_j Err_j\right] < Var[Err_j]$ so there is a high chance that: $Var\left[\frac{1}{n}\sum_{j=1}^{k} n_j Err_j\right] < Var\left[\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\right]$

Note that by using $k-$fold cross-validation we are increasing the bias, so if bias increases too much its better to use LOO cross-validation!

Note: taking means and averages reduce the bias only if we do it on independent things!

Conclusion: With cross-validation we can check what is the best model among many different ones.

More on LDA: Fisher's Argument for LDA

LDA is very robust to Gaussian assumption: now we introduce it without the gaussian assumption so that we show its robust!

Moreover LDA is a way to reduce the dimensionality of the problem so that we can show with pictures what our classifier

is doing!

Consider $\mathbb{R}^p \ni \underline{X}|L = i \sim \underline{\mu}_i, \Sigma$ for $i = 1, \ldots, g$ So the co-variance structure is the same for all groups! Note that we are not assuming Gaussianity!

Pick a direction $\underline{a} \in \mathbb{R}^p$ then: $\mathbb{E}[\underline{a}^T \underline{X}|L = i] = \underline{a}^T \underline{\mu}_i$ and $VAR[\underline{a}^T \underline{X}|L = i] = \underline{a}^T \Sigma \underline{a}$ for $i = 1, \ldots, g$

Now we want to find direction along which, the co-variability between groups with respect to the variability within groups, is maximised:

- The co-variability between groups is givne by

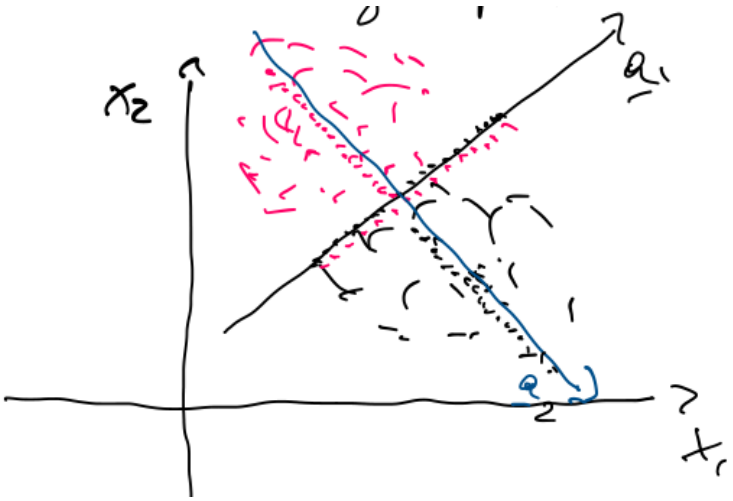
$$B = \frac{1}{g-1} \sum_{i=1}^{g} (\underline{\mu}_i - \bar{\underline{\mu}})(\underline{\mu}_i) - \bar{\underline{\mu}})^T$$

  where $\bar{\underline{\mu}} = \frac{1}{g} \sum_{i=1}^{g} \underline{\mu}_i$

- The co-variability withing groups is Σ

Idea: find $\underline{a} \in \mathbb{R}^p$ which maximise the separation (variability) between groups!

Consider the following figure:



Then a_1 is a direction which doesn't separate between the two groups!

On a_2 the projection of the two blobs is different and separable: we can separate the two groups! Indeed we can say that those above a certain threshold are in class 1 and so on and so forth!

Therefore we have identified a linear separation between the two groups!

Thus our optimisation problem is:

$$\arg\max_{\underline{a} \in \mathbb{R}^p} \frac{\underline{a}^T B \underline{a}}{\underline{a}^T \Sigma \underline{a}} = \arg\max_{\underline{a} \in \mathbb{R}^p} \frac{1}{g-1} \sum_{i=1}^{g} (\underline{a}^T \underline{\mu}_i - \underline{a}^T \bar{\underline{\mu}})^2 \frac{1}{\underline{a}^T \Sigma \underline{a}}$$

Now we want to re-parameterise the problem: $\underline{u} = \Sigma^{1/2} \underline{a}$ where we suppose Σ is invertible so that: $\underline{a} = \Sigma - 1/2 \underline{u}$ thus:

$$\frac{\underline{a}^T B \underline{a}}{\underline{a}^T \Sigma \underline{a}} = \frac{\underline{u}^T \Sigma^{-1/2} B \Sigma^{-1/2} \underline{u}}{\underline{u}^T \Sigma^{-1/2} \Sigma \Sigma^{-1/2} \underline{u}} = \frac{\underline{u}^T \Sigma^{-1/2} B \Sigma^{-1/2} \underline{u}}{\underline{u}^T \underline{u}}$$

Then from the Maximisation Lemma (used in PCA) we have that if: $\Sigma^{-1/2}B\Sigma^{-1/2} = \sum_{i=1}^{s} \lambda_i \underline{e}_i \underline{e}_i^T$ where $s = \min\{p, g-1\}$,

where:

- p is the order of Σ
- $g-1$ are the degrees of freedom of B as it's the co-variance computed with g data, but then we estimate the overall mean so we only have $g-1$ degrees of freedom!

  Note that B is a $p \times p$ matrix!

Thus:

$$\arg\max_{\underline{u}} \frac{\underline{u}^T \Sigma^{-1/2} B \Sigma^{-1/2} \underline{u}}{\underline{u}^T \underline{u}} = \underline{e}_1$$

where as usual λ_1 is the largest eigenvalue!

Thus:

$$\arg\max_{\underline{a}} \frac{\underline{a}^T B \underline{a}}{\underline{a}^T \Sigma \underline{a}} = \Sigma^{-1/2} \underline{e}_1$$

Moreover we can take, as in PCA: $\underline{a}_2 = \Sigma^{-1/2}\underline{e}_2, \ldots, \underline{a}_s = \Sigma^{-1/2}\underline{e}_s$ which are the best discriminating directions after having already considered $\underline{a}_1$

IT can be checked that if: $A = [\underline{a}_1^T, \ldots, \underline{a}_s^T]^T$ Then:

$$Cov(\underline{a}_i \underline{X}, \underline{a}_j \underline{X}) = \underline{a}_i^T \Sigma \underline{a}_j = \underline{e}_i^T \Sigma^{-1/2} \Sigma \Sigma^{-1/2} \underline{e}_j = \underline{e}_i^T \underline{e}_j = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ otherwise} \end{cases} \implies Cov(A\underline{X}) = I$$

Indeed the eigen-vectors are orthogonal to each other!

Thus we have found the directions of maximum separation and the scores on all directions are uncorrelated!

$\underline{a}_1^T \underline{X}, \underline{a}_2^T \underline{X}, \ldots,$ are called First, Second, $\ldots,$ **Fisher's Discriminant Scores.**

We apply this to the estimated mean and co-variance, since $\underline{\mu}_i$ and Σ are not known!

We estimate them using the training data: $\hat{\underline{\mu}}_i = \bar{\underline{X}}_i$ and $\hat{\Sigma} = S_{pooled} = \frac{1}{n-g} \sum_{i=1}^{g} (n_i - 1) S_i$

We can use **Fisher's Discriminant Scores** also for dimensional reduction!

Now we see how to build a classifier by means of **Fisher's Discriminant Scores**.

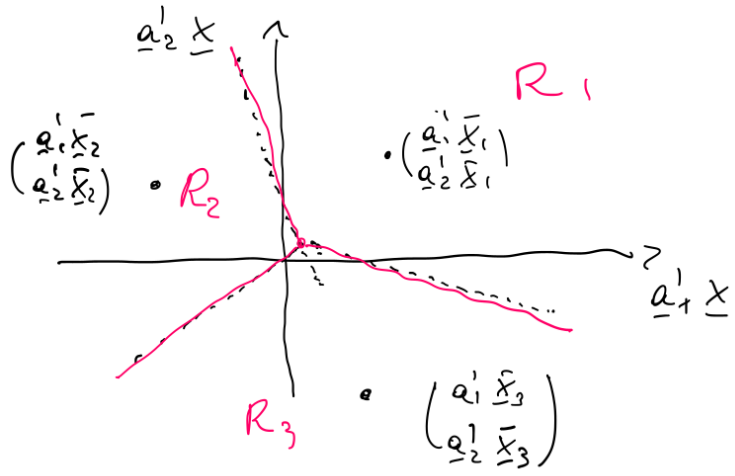
we project $\bar{\underline{X}}_i$ to get: $[\underline{a}_1^T \bar{\underline{X}}_i, \ldots, \underline{a}_s^T \bar{\underline{X}}_i]^T$ but we cut,like in PCA, at level k so we consider the matrix only until: $\underline{a}_k^T \bar{\underline{X}}_i$ with $i = 1, \ldots, g$

now we want to do classification: to classify the unit for which we have observed $\underline{X}$ we project it on the Fisher's scores space:
$\underline{X} \to [\underline{a}_1^T \underline{X}, \ldots, \underline{a}_k^T \underline{X}]^T$ and we attribute $\underline{X}$ to the closest mean: $[\underline{a}_1^T \bar{\underline{X}}_i, \ldots, \underline{a}_k^T \bar{\underline{X}}_i]^T$ for $i = 1, \ldots, g$

Thus we have our classifier:

$$\delta(\underline{x}) = i \text{ if } \underline{x} \in R_i \text{ with } i = 1, \ldots, g \text{ wher } R_i = \left\{ \underline{x} \in \mathbb{R}^p : \sum_{j=1}^{k} (\underline{a}_j^T \underline{X} - \underline{a}_j^T \bar{\underline{X}}_i)^2 \leq \sum_{j=1}^{k} (\underline{a}_j^T \underline{X} - \underline{a}_j^T \bar{\underline{X}}_h)^2 \text{ for } h = 1, \ldots, g \right\}$$

For example consider the following figure:



we have three group means $(\underline{a}_1^T \bar{\underline{X}}_i, \underline{a}_2^T \bar{\underline{X}}_i)^T$ where $i = 1, 2, 3$ and we have a point $(\underline{a}_1^T \underline{x}, \underline{a}_2^T \underline{x})^T$

We compute the distance to this point between all the three group means above: we attribute this point to the group whose mean is closest to our new point!

Note that in the above figure all the points on one side of the boundary are closets to the group mean which identifies that boundary!

In the above we are building up the **Voronoi Tessellation** of this plane in Fisher's Coordinates: first we project everything on fisher coordinates and then we build up the **Voronoi tessellation** and then that will give us the classifier!

Note that the boundaries between the different regions are all linear: indeed we are minimising the square distance.

Theorem: this classifier is exactly LDA when we take equal priors: indeed we didn't use the prior to build up this classifier: $p_1 = \cdots = p_g = \frac{1}{g}$

Note: the above is used with $k = 2, 3$ for graphical reasons: we can plot nice figures! For classification we use LDA, which is overall pretty good!

ISLR: MOOC chapter 5

We build the classier from the prior, and the posterior trough Bayes Theorem!

Logistic regression doesn't model and goes straight to classification: very basic and powerful, quick regression for classification.

Consider the slides of the chapter 5 then:
$P[L = no] = \frac{9667}{10000}$ we estimate it from frequency since we suppose its random sample from population.

Moreover: $P[L = yes] = \frac{333}{10000} = 0.03$

Therefore the error rate of flipping a not fair coin following these prior distribution is 3% so the model error of 2.75% is not so much smaller than a random, not fair, coin!

The above is not a promising result: the benchmark is important! The benchmark is what we can do without using

the information given by the features, which is given by the prior distributions!

The mis-classification error of the NO is $\frac{23}{9667} = 0.2\%$ but the mis-classification error of YES is: $\frac{252}{333} = 75.7\%$!

This means that the model mis-classifies 75% of the defaulters: bad classifier!

LDA is a Bayes Classifier: we assign a person to NO if the posterior of NO larger than the posterior probability of YES

So we assign to NO if the posterior probability is greater than 0.5 since the posterior must sum up to 1, that is: $P[L = No|X = x] > 0.5$

To keep track of the cost we change the threshold: we are more conservative about saying no, than saying yes! So: if $P[L = NO|X = x] > t$ then we say YES, otherwise NO.

This rule is different from Bayes Classier and from LDA since we change the threshold because we take into account the cost!

Consider slide 34: we see that to reduce the false negative rate we set $t < 0.1$ indeed its better to lower false negative since false positive aren't as bad! The total error rate anyway doesn't change much.

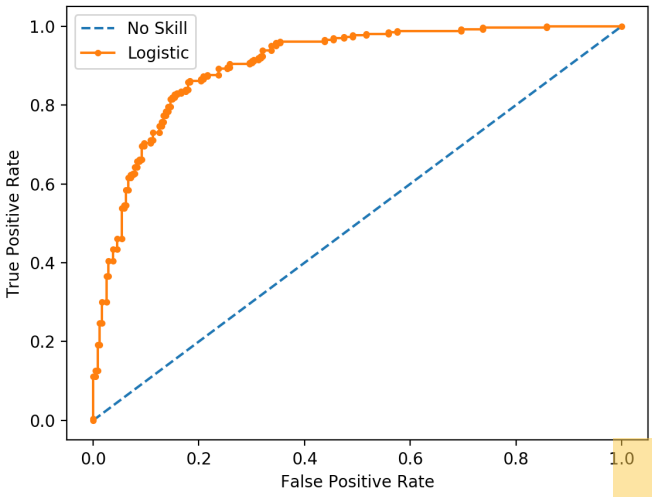
Note that the above is the same thing as changing the cost of mis-classification!

Building the ROC curve:
We have the false positive rate on the $x-$axis and the true positive rate on the $y-$axis. By changing the threshold we get the ROC curve.

The more we have a tight turn on the top left, the higher the separating power we have, indeed we would like one curve that follows half of a square.

The worse we can have is a ROC curve equal to the line $y = x$, so we define AUC, the area under the curve, as the are between the ROC curve and the line $y = x$

The higher AUC the better the classier! In the following figure we see an example of a ROC curve:



The above is hard to apply if we have more than 2 groups! Indeed the ROC curve is widely used in medical trials, where we test whether a drug works or not!

Note: there is a wrong formula on slide 37, as the $\log(det(\Sigma))$ is missing!

Note: logistic regression has many problem when there is a neat (clear) separation: non-identifiable model! Instead LDA is good when there is a neat separation!

KNN: $k-$nearest Neighbour classier: brute force machine learning classifier. We have data and labels, and if we have a new observation we take a k neighbourhood of that point and we take the the k closest points and we pick the majority label: if $k = 3$ and if 2 out of 3 closest points are in class 1 then we put this new datum in class 1

The problem of KNN is over-fitting! Moreover note that we choose k by cross validation.

Note: if we have a qualitative random variable is not necessarily discrete.

For example if we have the colour of eyes: brown, yellow we can set brown equal to class 1 and yellow equal to class 2 LDA wouldn't work because we can't order objectively two colours!

For example we could label brown as 10000 and yellow as 10 but LDA would behave totally differently from before!

We can use LDA with respect to real discrete random variables but where we actually measure something: such as the number of children!

Instead in logistic regression we can keep both qualitative and quantitative variables by introducing dummy variables (e.g: one hot encoding).

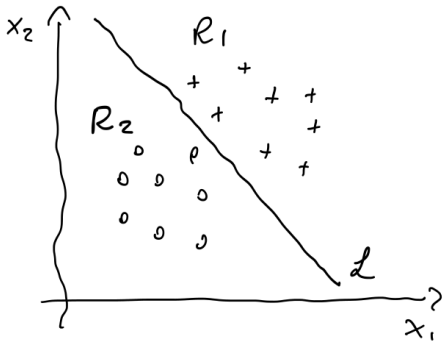
16 Lecture 27: 28th Of April 2020

Support Vector Machine

Consider a supervised classification problem with only two groups: dichotomous supervised classification problem.

The training set is: $\mathbb{X} = \begin{bmatrix} \underline{x}_1 & \ell_1 \\ \vdots & \vdots \\ \underline{x}_n & \ell_n \end{bmatrix}$ we have n statistical units and n labels which we have observed, where $\underline{x}_i \in \mathbb{R}^p$ and $\ell_i \in \{1, 2\}$

We want to find the separating hyper-plane, which separate the two groups:



the hyper-plane defines a partitions of $\mathbb{R}^p$, which is equivalent to a classifier: $\{R_1, R_2\} \leftrightarrow \delta$

We know already that if were using **LDA** the output of the algorithm would be an hyper-plane dividing the two groups: what if we can't find an hyper-plane splitting the two groups? when is it the case that such an hyper-plane exist?

We want to answer to this same question but in a more general setting:
Given two sets $A, B \subseteq \mathbb{R}^p$ when can they be separated by a hyper-plane?

- If it exist we can try to find the best one with respect to some optimality criterion
- If it doesn't exist it's useless trying to find it!

Let $CH(A), CH(B)$ be the convex-hulls containing A, B respectively, so they are the smallest convex set containing A, B. Now if:

1) $CH(A), CH(B) \neq \emptyset$ Which is always the case when $A, B \neq \emptyset$
2) $CH(A) \cap CH(B) = \emptyset$
3) either $CH(A)$, or $CH(B)$, is an open set

Then: $\exists$ a separating hyper-plane!

Note: the above result is a consequence of the Geometric Form of Hanh-Banach Theorem!

Note: Condition **3)** can be replaced with the following:

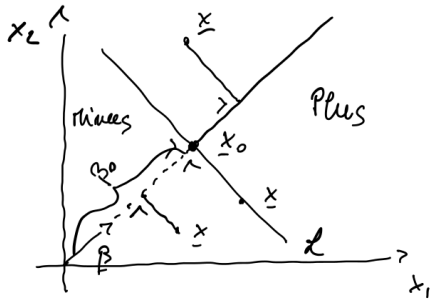
$$3')CH(A), CH(B) \text{ are both closed sets and at least one of them is compact}$$

This is always true if A, B are sets with finite cardinality, so we just need to check **2)**!

Let $\mathcal{L}$ be an hyper-plane in $\mathbb{R}^p$, so it's an affine sub-space of $\mathbb{R}^p$ of dimension $p-1$ Then: to identify $\mathcal{L}$ it it's enough to specify the direction orthogonal to the hyper-plane $\mathcal{L}$

Let's call this direction $\underline{\beta} \in \mathbb{R}^p$ so: $\underline{\beta} \perp \mathcal{L}$ and since there is an infinite number of them we choose the one such that: $\left\|\underline{\beta}\right\| = 1$

We have the following situation:



where $\underline{x}_0 \in \mathcal{L} \cap Span(\underline{\beta})$ and $\beta_0 = \|\underline{x}_0\|$ Then:
for any $\underline{x} \in \mathcal{L} \implies \pi_{\underline{x}|\underline{\beta}} = \underline{x}_0$ so the projection of $\underline{x}$ on the linear space generated by $\underline{\beta}$ is $\underline{x}_0$

Also the vice-versa holds, therefore:

$$\underline{x} \in \mathcal{L} \iff \pi_{\underline{x}|\underline{\beta}} = \underline{x}_0 \iff \underline{\beta}^T \underline{x} = \beta_0 \iff \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p = \beta_0 \text{ for } \underline{x} = (x_1, \ldots, x_p)$$

Therefore if $\underline{x} \notin \mathcal{L}$ then: $\pi_{\underline{x}|\underline{\beta}}$ is either larger or smaller then β_0 so we can identify a partition of $\mathbb{R}^p$ Indeed:

- $\underline{x} \in$ Plus $\iff \pi_{\underline{x}|\underline{\beta}} > \beta_0 \iff \underline{\beta}^T \underline{x} > \beta_0$
- $\underline{x} \in$ Minus $\iff \pi_{\underline{x}|\underline{\beta}} < \beta_0 \iff \underline{\beta}^T \underline{x} < \beta_0$

Indeed $\underline{\beta}^T \underline{x} - \beta_0$ measures, with sign, the distance between $\underline{x}$ and $\mathcal{L}$

Let's go back to $\mathbb{X}$: we want to re-parametrise the labels. Assume there exists a separating hyper-plane $\mathcal{L}$, then: $y_i = \begin{cases} 1 \text{ if } \ell_i = 1 \\ -1 \text{ if } \ell_i = 2 \end{cases}$ where we suppose to have two groups: Group 1 is: $R_1 =$ Plus and Group 2 is: $R_2 =$ Minus
with the above re-parametrisation we have that: $y_i(\underline{\beta}^T \underline{x}_i - \beta_0) \geq 0 \; \forall i = 1, \ldots, n$

Moreover $y_i(\underline{\beta}^T \underline{x}_i - \beta_0)$ it's the distance, without sign, between point $\underline{x}_i$ and the separating hyper-plane $\mathcal{L}$

Now comes the optimisation part: we find the separating hyper-plane such that the smallest of these distances is as large as possible.

Let $M_1 = \min_i y_i(\underline{\beta}^T \underline{x}_i - \beta_0) \geq 0 \; \forall i = 1, \ldots, n$ Note thaat M_1 is called Margin.

Optimal separating hyper-plane: find $\mathcal{L}$, that is: find $\underline{\beta}, \beta_0$, such that M_1 is maximum.

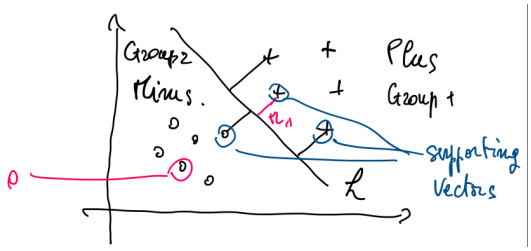
The above is a well posed optimisation problem, which can be written equivalently as follows:

$$\max_{\underline{\beta}, \beta_0} M \text{ such that } \left\|\underline{\beta}\right\| = 1 \text{ and } y_i(\underline{\beta}^T \underline{x}_i - \beta_0) \geq M \; \forall i = 1, \ldots, n$$

To define the separating hyper-plane we only need the points closest to the boundary and are the only points which can influence it, and these are called supporting vectors!

Note: if we move points that aren't supporting vectors, then the boundary identified doesn't' change! Thus the optimal separating hyper-plane is very robust to modification of the data set with respect to points that aren't supporting vectors.

We have the following situation:

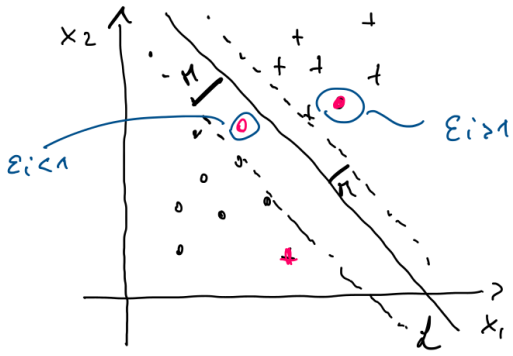


Note that this very different from **LDA**, in which we use the whole data set for estimating the variance and the mean: indeed mean is not robust with respect to outliers! So for **LDA** moving one observation will move the mean, and so the boundary!

Moreover note how **LDA** is based on the probability of belonging to a group, whereas **SVM** is a geometric direct method with no probabilities and no priors!

What happens if the units in the training set $\mathbb{X}$ aren't linearly separable? There are two options:

- **Approach 1:** We have the following situation:



we allow some overlapping: so we have some mis-classified points.

To have points mis-classified we need to pay something since there is some constraint not respected, so we have a soft constraint and a soft version of the previous optimisation problem:

$$\max_{\underline{\beta},\beta_0} M \text{ such that}$$

$$\|\underline{\beta}\| = 1 \text{ and } y_i(\underline{\beta}^T \underline{x}_i - \beta_0) \geq M(1-\epsilon_i)\ \forall i = 1,\dots,n \text{ where:}$$

$$\epsilon_i \geq 0 \text{ and must satisfy the Budjet Constraint: } \sum_i \epsilon_i \leq C$$

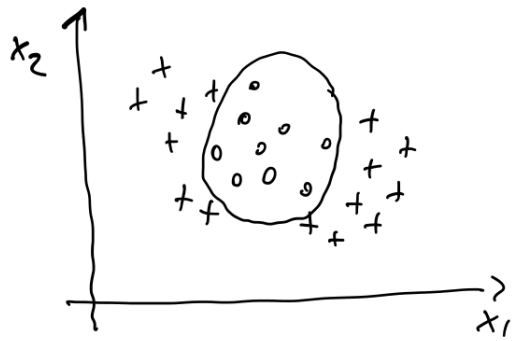
Note that we need to tune C, which is a penalisation (or control) parameter, until we find good enough one!

Notice how, if $\epsilon_i > 1$, then we are allowing for $M < 0$ so we can have mis-classified points: points that are on the wrong side of the partition outside the margin!

These are the points which are mis-classified outside the dotted lines in the figure above!

- **Approach 2:** we use kernel methods.

  This is old approach that can be applied to any linear method and it's useful when we don't see clear linear boundaries between the two groups, such as:



  Remember that always a linear method is linear with respect to the feature but not necessarily with respect to the original ones, so we can still use a linear model which is non-linear in terms of the original features and this without using Kernels!

  For example if $\mathbb{X} = \begin{bmatrix} z_{11} & z_{12} & \ell_1 \\ \vdots & \vdots & \vdots \\ z_{n1} & z_{n2} & \ell_2 \end{bmatrix}$ so we have two features z_1, z_2

  Now we are free to transform our variables: taking powers (e.g: box-cox), taking the log, and so. For instance we can take:
  $x_1 = z_1, x_2 = z_2, x_3 = z_1^2, x_4 = z_2^2, x_5 = z_1 z_2, x_6 = sin(z_1), \ldots, x_p = log(z_2)$ and we build up a new training set $\mathbb{X}$ with all the new features $x_i, i = 1, \ldots, p$ and label L

  So: $\mathcal{L} : \beta_1 x_1 + \cdots + \beta_p x_p = \beta_0$ is not linear in the original features, indeed it would be: $\beta_1 z_1 + \beta_2 z_2 + \beta_3 z_1^2 + \cdots + = \beta_0$

  Note: the larger p, the more separated are the points, so eventually by increasing p we will be able to linearly separate the points! But: due to the curse of dimensionality there is a risk of over-fitting, if p is too large!

Note that **LDA** and **SVM** are almost equal: we can find a good **SVM** and then we can modify **LDA** accordingly.

They both work really well then two groups are separated or almost separated, otherwise logistic regression works well!

What if we have more than 2 groups? We have two options:

- **OVA: one-versus-all**
  We develop a classifier that separates between group 1 and all the others, then another classifier between group 2 and all the others and so... So we fit K different binary (2 class) classifiers.

  Then we assign x^* to the class with the largest posterior $(\hat{f}_k(x^*))$

- **OVO: one-versus-one**
  We build $\binom{(}{k}, 2)$ pairwise classifiers, and we assign x^* to the class that wins the most pairwise competitions!

Note that if K too large we must use OVO!

17 Lecture 28: 30th Of April 2020

Unsupervised classification: cluster analysis

We want to perform classification but we don't have labels! So the training set has the form: $\mathbb{X} = [\underline{x}_1^T, \dots, \underline{x}_n^T]^T$

We can't see the labels: we believe there are labels but they are hidden!

Note that $\underline{x}_i \in \mathbb{R}^p$ can be either categorical or quantitative variables!

The goal is to *estimate* the labels $\hat{\ell}_i$ with $i = 1, \dots, \hat{g}$ This is a very qualitative (non-parametric) approach!

We also want to estimate the number of groups $\hat{g}$!

There are two main approaches:

- The first approach is parametric and it requires modelling of the features. It boils down to solve a complicated optimisation task, and it is often based on maximum likelihood! An example of this is the so called Expectation-Maximisation Clustering Algorithm: the idea is that units belonging to the same group belong to the same Gaussian distribution with a similar variance!
- The second Approach is non-parametric and it's based on similarity (or dissimilarities): we will focus on this!

In any case there is a basic idea: units belonging to the same group (i.e: cluster) are more similar, or less dissimilar, than units belonging to two different groups (i.e: clusters)!

We need a tool to quantify dissimilarities: how do we capture them? For us the dissimilarity function it's a non-negative function $d : \mathbb{R}^p \times \mathbb{R}^p \to [0, \infty)$ which is not necessarily a distance. We require the following properties:

1) $\forall \underline{x} \in \mathbb{R}^p : d(\underline{x}, \underline{x}) = 0$ Two units with the same feature vector have zero dissimilarity! So the dissimilarity matrix has all zeroes on the diagonal!

   For example: two human beings have zero dissimilarity if we have only one feature which is: number of eyes!

1′) $\forall \underline{x}, \underline{y} \in \mathbb{R}^p : d(\underline{x}, \underline{y}) = 0 \iff \underline{x} = \underline{y}$ This is a bit more strong than the above one! Indeed if $d(\underline{x}, \underline{y}) = 0$ it is not necessarily true that $\underline{x} = \underline{y}$!

   Consider for example $f, g \in L^2 : d(f, g) = \|f - g\|_{L^2}^2 \implies d(f, g) \not\iff f = g$ in fact f, g are equal almost everywhere but not everywhere.

2 $d(\underline{x}, \underline{y}) = d(\underline{y}, \underline{x})$ $\forall \underline{x}, \underline{y} \in \mathbb{R}^p$ so it's symmetric!

   This is not the case in the case for Kullback–Leibler (KL) Divergence, which is a dissimilarity measure!

   For us if we have a dissimilarity that is not symmetric we make it symmetric, for instance by summing the two dissimilarities and dividing by two!

3) $\forall \underline{x}, \underline{y}, \underline{z} \in \mathbb{R}^p : d(\underline{x}, \underline{y}) \leq d(\underline{x}, \underline{z}) \leq d(\underline{z}, \underline{y})$ This is the triangle inequality.

Definition: If d satisfies $1', 2, 3$ then d is a metric (distance).

Definition: If d satisfies $1, 2, 3$ then it's called *pseudo-metric*.

sometimes we may want stronger properties, such as:

$$4) \quad \forall \underline{x}, \underline{y}, \underline{z} \in \mathbb{R}^p : d(\underline{x}, \underline{y}) \leq \max(d(\underline{x}, \underline{z}), d(\underline{z}, \underline{y}))$$

Note that this is stronger than 3) indeed it implies it!

Definition: if d satisfies $1', 2, 4$ then d is called *ultra-metric*: some algorithms we will meet can be seen as algorithm that start with a dissimilarity (something even weaker than a pseudo-metric) and the algorithm just finds an ultra-metric!

Note: Clustering just finds the closest ultra-metric with respect to some dissimilarity specified in the beginning!

Examples:

- $d(\underline{x}, \underline{y}) = \sqrt{(\underline{x} - \underline{y})^T(\underline{x} - \underline{y})} = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2}$ is the euclidean metric.

  Often we apply it to standardise variables in clustering: first we standardise vectors with respect to the standard deviation and then apply euclidean distance, so that along the components we have the right meter (i.e: the standard deviation!)

  Otherwise the euclidean metric will consider a difference in $1kg$ of bread we eat, and $1euro$ of total expense, as the same, but the variability between these two is much different!

- $d_{\Sigma^{-1}}(\underline{x}, \underline{y}) = \sqrt{(\underline{x} - \underline{y})^T\Sigma^{-1}(\underline{x} - \underline{y})}$ is the **Mahalanobis distance**.

  The problem is (as it happens with standardisation!) that we assume the same co-variance structure in every group but this is not necessarily true and we can't even check whether this is true or not!

  Note: standardisation is like doing the Mahalanobis distance but only saying that the diagonal of Σ is the same across each group and not the entire co-variance structure!

- We have a whole set of Minkowski Distances: ℓ_m such that $d(\underline{x}, \underline{y}) = \left(\sum_{i=1}^{p}|x_i - y_i|^m\right)^{1/m}$ Note that:

  - For $m = 2$ we have the euclidean distance.

    Note: minimising the euclidean distance of a cloud of points we find the mean (baricentre)
  - For $m = 1$ we have the Manhattan distance. It's called like this because it's how you move in Manhattan: there is a grid structure in the city and you cant move across buildings but along avenues and street!

    Note: minimising the Manhattan distance of a cloud of points we find the meadian!
  - For $m = \infty$ we go towards ℓ_∞ which is the maximal difference between components

  Note that if $m < 1$ then every component is important!

- if $\underline{x}, \underline{y} \in (\mathbb{R}^p)^+$, that is: we have points with non negative components, then we have the Canberra Distance, defined as:

$$d(\underline{x}, \underline{y}) = \sum_{i=1}^{p}\frac{|x_i - y_i|}{x_i + y_i}$$

  This is useful if we have economic quantities!

Dissimilarities for categorical variables

Suppose $\underline{x} \in \{0, 1\}^p$ for instance $\underline{x} = (0, 1, 1, 0, 0, 1)^T$ Then we can always move from a categorical variable to a vector of $\{0, 1\}$ (e.g: one hot encoding) For example:
If: $x \in blue, red, brown$ then $x \to (x_1, x_2)$ and:

- $x_1 = 1$ if blue and $x_1 = 0$ if not blue
- $x_2 = 1$ if red and $x_2 = 0$ if not red

So that: $(1,0)$ is blue, $(0,1)$ is red, $(0,0)$ is brown and $(1,1)$ is impossible!

Then if $\underline{x}, \underline{y} \in \{0,1\}^p$ then using the euclidean distance: $d(\underline{x}, \underline{y}) = \sqrt{\sum (x_i - y_i)^2} \implies d^2(\underline{x}, \underline{y})$ is equal to the number of discordances between the components of $\underline{x}, \underline{y}$

Idea for extension: with: $a + b + c + D = p$ Then: $d_e^2(x, y) = c + b$ is the Euclidean Distance. There are other

	$y = 1$	$y = 0$
$x = 1$	a	b
$x = 0$	c	d

possibilities: we can consider the percentage of dissimilarities: $d(\underline{x}, \underline{y})^2 = \frac{(c+b)}{p}$ or we can consider $d(\underline{x}, \underline{y})^2 = 1 - \frac{a}{p}$

Note: There are also mixed cases in which we have both quantitative and categorical variables. Then the final distance could be a linear (or convex or affine) combination of the two:

$$d(\underline{x}, \underline{y})^2 = \lambda d_Q^2(x^q, y^q) + (1 - \lambda) d_C^2(y^c, x^c)$$

where $\lambda \in (0,1)$ and where the sub-script q stands for quantitative part and the sub-script c for categorical part.

Sometimes we don't cluster the units but the variables of the training set: suppose $\mathbb{X} = [\underline{y}_1, \ldots, \underline{y}_p]$ and call $\underline{x}_i$ the columns of $\mathbb{X}$

The goal is to cluster variables: are there variables more or less related? Are there variables talking about same thing?

One obvious way to measure similarity between variables is correlation:

$$d^2(Var_i, Var_j) = d(\underline{X}_i, \underline{X}_j) = 2(1 - Cor(\underline{X}_i, \underline{X}_j))$$

from which we get cosine distance indeed correlation is the angle between variables

---

Each clustering problem has its own right dissimilarity function and choosing the right one is the problem!

Anyway, once we choose our dissimilarity function the training set is transformed into the dissimilarity matrix: this will be the input of the clustering algorithm!

If $\mathbb{X} = [\underline{x}_1^T, \ldots, \underline{x}_n^T]^T$ with $\underline{x}_i \in \mathbb{R}^p$ then the dissimilarity matrix is given by: $D = [d_{ij}]$ where $d_{ij} = d(x_i, x_j)$ for $i, j = 1, \ldots, n$ Note that D is an $n \times n$ matrix!

If d is a metric then $d_{ij} = d_{ji}$ then D is symmetric, moreover we have zeroes on the diagonal: $d_{ii} = 0$ if 1 is satisfied.

Thus if $1, 2$ are satisfied we have that D is a triangular matrix!

Dissimilarity between clusters: suppose U, V are two clusters, that is: finite sets of points in $\mathbb{R}^p$ then: $d(U, V) =$?

There are many possibilities:

- Single Linkage (sl): $d_{sl}(U, V) = \min\{d(x, y) : \underline{x} \in U, \underline{y} \in V\}$

- Complete Linkage (cl): $d_{cl}(U,V) = \max\{d(x,y) : \underline{x} \in U, \underline{y} \in V\}$
- Average Linkage (av): $d_{av}(U,V) = \frac{1}{\#U\#V} \sum_{\underline{x}\in U, \underline{y}\in V} d(\underline{x},\underline{y})$

Hierarchical Agglomerative Clustering

This is an iterative algorithm and at each iteration we move to an hierarchy of clusters in the training set.

It's agglomerative because we move from step k to step $k+1$ by merging clusters. We stop when we have one single cluster!

Note: the opposite of Hierarchical Agglomerative Clustering is called Hierarchical Divisive Clustering, in which we start top-down!

First we suppose to have a dissimilarity measure between pints and the linkage (dissimilarity between cluster) Then:

- initialisation: each unit is a cluster
- Until convergence repeat:
  - step 1: merge the two clusters which are less dissimilar
  - step 2: compute the new dissimilarity matrix D

  We stop when we only have one big cluster!

The dendrogram is the graphical representation of this algorithm!

Note that as input we only need D and not the training set!

Now we see an example in which we use single linkage and in which $n = 5$:

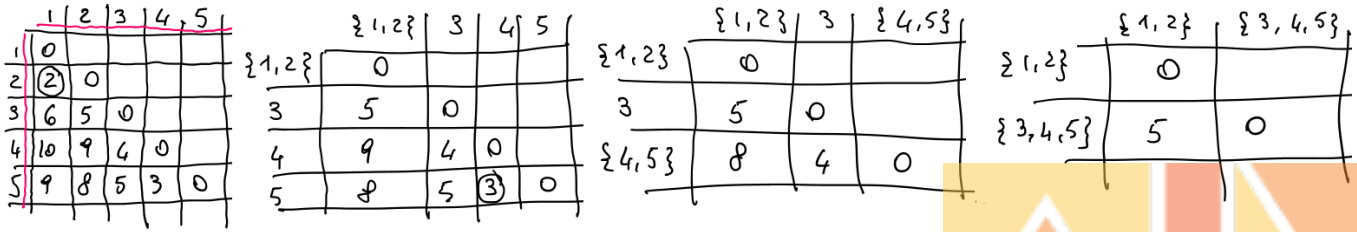
- At iteration 1:
  we merge the two closest units: $d(1,2) = 2$ so we merge units $1, 2$ into the cluster $\{1,2\}$

  Now we update D with: $d(\{1,2\},3) = \min(d(1,3), d(2,3) = 5$
- At iteration 2:
  we merge the two closest clusters: $d(5,4) = 3$ so we merge them into the cluster $\{4,5\}$

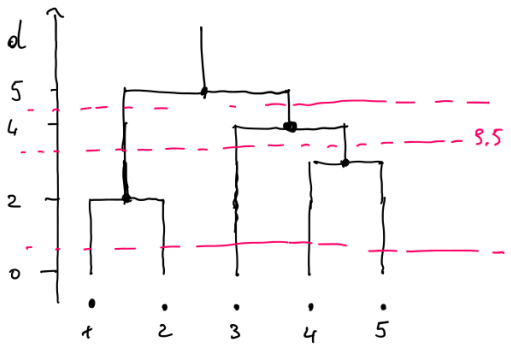
  Now we update D with: $d(\{4,5\},\{1,2\}) = \min(d(4,1), d(4,2), d(5,1), d(5,2)) = 8$
- At iteration 3:
  we merge 3 and $\{4,5\}$ at distance 4 so we get: $\{3,4,5\}$

  Then we update D
- At iteration 4:
  we merge the two reaming cluster at distance 5: so we merge $\{1,2\}$ and $\{3,4,5\}$ into $\{1,2,3,4,5\}$

In the figure below we see the different distance matrices, along the iterations:



In the figure below we see the dendrogram which is the graphical representation of the algorithm:



Note that to represent it we used the distance at which we merged the clusters, so we get a summary of the algorithm. Moreover this is what we look at to understand the clustering structure. Note that the graphical representation is computationally expensive!

Now we need to decide where to cut the dendrogram: at which distance do we look at the data set?

For instance:

- At a distance of 3.5 we have three clusters: $\{1, 2\}, 3, \{4, 5\}$
- At a distance of 4.5 we see two clusters
- At distance of 0.5 we see five clusters

The farther away the clustering structure appears!

Note that if we don't have, on the $x-$axis of the dendrogram, units in nice order we get a bad picture: that's why the method is computationally expensive!

Indeed we need to explore all the permutation of the units to find the right permutation: otherwise the branches of the dendrogram would cross each other and we wouldn't be able to understand the figure!

Now we can compute a new distance matrix based on the dendrogram:

$C =$

	1	2	3	4	5
1	0				
2	2	0			
3	5	5	0		
4	5	5	4	0	
5	5	5	4	3	0

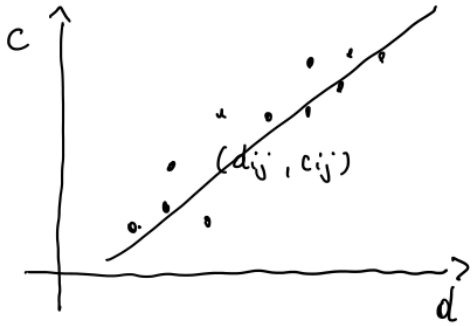
In this distance matrix the distance between units is the distance at which the units were merged in the same cluster!

This is called **Cophenetic distance** and it's an ultra metric!

Given a couple of units we associate two distance: the original one and the **Cophentic one** so:

$$(x_i, x_j) \to (d_{ij}, c_{ij}) \quad \forall i, j = 1, \dots, n$$

Then we can produce a plot:



in which on the $x-$axis we have the distance D, and on the $y-$axis we have the distance C, and each points of this space is (d_{ij}, c_{ij})

Now we can comptue the correlation of the points (d_{ij}, c_{ij}) thus:

$$|Corr\{(d_{ij}, c_{ij}) : i, j = 1, \dots, n\}| = |Corr((D, C))| = |cpcc|$$

Which is the **Cophenetic Correlation Coefficient** and the closer it is to 1 and -1, the better the clustering structure in C is representing a true structure that is in the D matrix!

Note that since it's a correlation coefficient it's in between $[-1, 1]$!

Note: The correlation between two matrices is the correlation between all the points in the matrix!

Note: if $cppcc = 0$ then we couldn't capture any clustering structure by the dendrogram!

With the **Cophenetic Correlation Coefficient** we can measure different types of clustering structure captured by the dendrogram!

For example we try hierarchical clustering changing linkage, we try them all and see the best **Cophenetic Correlation Coefficient**!

Deciding where to cut the dendrogram is hard: one good idea is to jitter the data and see what happens!

Note: single linkage often generates a chain effect: we create a cluster that is a chain that goes across different clusters! So if clusters are ellipsoid blobs of data we use complete linkage or average linkage!

Note: complete linkage and average linkage have the opposite problem: they generate ellipsoidal clusters!

18 Lecture 29: 4th Of May 2020

Ward's Method for Hierarchical Clustering

Instead of deciding a linkage for joining up the clusters we follow a different criterion. Let: $\tau = \{\underline{x}_i : i = 1, \dots, n\} \subseteq \mathbb{R}^p$ be the training set: this is the same as $\mathbb{X}$ And let $C_1, \dots, C_k$ be clusters, Then:

$$ESS_j = \sum_{\underline{x}_i \in C_j} (\underline{x}_i - \bar{\underline{x}}_j)^T (\underline{x}_i - \bar{\underline{x}}_j) = \sum_{\underline{x}_i \in C_j} \left\| \underline{x}_i - \bar{\underline{x}}_j \right\|^2$$

where $\bar{\underline{x}}_k = \dfrac{1}{\#C_j} \displaystyle\sum_{\underline{x}_i \in C_j} \underline{x}_i$ with $j = 1, \dots, k$ this is the within cluster variability

Then: $ESS = ESS_1 + \cdots + ESS_k$

Ward's method proceeds iteratively bottom up (so it's an aggolmerative method): at each iteration we merge the two clusters which generate the minimum increase in ESS

We don't have merge distance of cluster but ESS! Also on the dendrogram we plot on $y-$axis the ESS and not the distance.

This algorithm tends to create ellipsoidal cluster: no chaining effect problem, as opposed to complete linkage!

Wards method is realted to $k-$means: it's not hierarchical clustering. We will also see variations of it, such as $k-$medoids

---

k-means

We introduce $k-$means in a general setting: let $\tau = \{\underline{x}_i : i = 1, \dots, n\} \subseteq \mathbb{R}^p$ be the training set with $\tau \subseteq \mathbb{R}^p$ and let $d : \mathbb{R}^p \times \mathbb{R}^p \to [0, \infty)$ be a dissimilarity measure not necessarily a metric!

Given $k \geq 1$: we need to choose it a-priori, finding k clusters means: identifying k subsets of the training set: $C_1, \dots, C_k \subseteq \tau$ and they need to be such that:

- $C_i \cap C_j = \emptyset \; \forall i \neq j$
- $\displaystyle\bigcup_{j=1,\dots,k} C_j = \tau$

So we need to find a partition of the training set in k subsets!

Note: there are method where we allow overlapping of clusters and unit that aren't clustered! e.g: fuzzy clustering.

Now we define centroids: given a cluster $C_j \subseteq \tau$ we define:

$$\bar{\underline{x}}_j = \arg\min_{\underline{x} \in \mathbb{R}^p} \sum_{\underline{x}_i \in C_j} d^2(\underline{x}_i, \underline{x})$$

So the centroid is a point which minimises the sum of all the squared distances from that point to the cluster!

Note: If d is the euclidean distance then $\bar{\underline{x}}_j$ is the baricentre of the cluster (mean), so:

$$\bar{\underline{x}}_j = \frac{1}{\#C_j} \sum_{\underline{x}_i \in C_j} \underline{x}_i$$

So the optimal clustering is the solution of this problem:

$$\text{Find } C_1, \dots, C_k \text{ such that: } \sum_{j=1}^{k} \sum_{\underline{x}_i \in C_j} d^2(\underline{x}_i, \bar{\underline{x}}_j) \text{ is minimised}$$

So we minimise the total variability across the cluster summed for all clusters!

We find the solution through iterative procedure which usually finds a local optimum, as it is not guarantee to find global optimal!

k-means algorithm for solving the above problem:

- **Initialisation step:**
  we either assign at random the units of the training set among the k subsets $C_1, \dots, C_k$ and then we go to **Step 1**.

  or we assign at random k centroids $\underline{\bar{x}}_1, \dots, \underline{\bar{x}}_k$ in $\mathbb{R}^p$ and then go to **Step 2**.

  Note: the final output will strongly depends on this random assignment: so each time we run $k-$means we get different result!

  So we need to repeat the algorithm a few times to be sure the result is robust.

- Iterate until convergence these two steps:
  - **Step 1:**
    for $j = 1, \dots, k$ compute the centroid of cluster C_j so that at the end of this step we will have: $\underline{\bar{x}}_1, \dots, \underline{\bar{x}}_k$
  - **Step 2:**
    for all $\underline{x}_i \in \tau$ we assign each unit to the cluster C_j if: $d^2(\underline{x}_i, \underline{\bar{x}}_j) \leq \min\{d^2(\underline{x}_i, \underline{\bar{x}}_j) : j = 1, \dots, k\}$

  Convergence: we stop when the centroids at **Step 1** are the same as those computed in the previous iteration!

The algorithm may iterate forever: if we have Euclidean distance it was proved that it converges, otherwise we need to prove it (very hard) for a generic distance.

Otherwise we try it and fingers crossed!

Note: if we use a general d that is not euclidean, then we may iterate forever.

The difficult step is **step 1:** an easy way out is to minimise with respect to only the training set and not $\mathbb{R}^p$ so that:

$$\underline{\bar{x}}_j = \arg\min_{\underline{x} \in \tau} \sum_{\underline{x}_i \in C_j} d^2(\underline{x}_i, \underline{x}) \text{ for } j = 1, \dots, k$$

In this case $\underline{\bar{x}}_j$ is called **medoid** and not **centroid** and the algorithm is called $k-$**medoid** and not $k-$**means.**

In this case we need to compute only a finite number of steps and the clusters will be specified from units that we have observed and not on the centroid!

Moreover the mean will be the average mean that nobody knows since we minimise in τ and not in $\mathbb{R}^p$! Indeed there is no such thing as the average man!

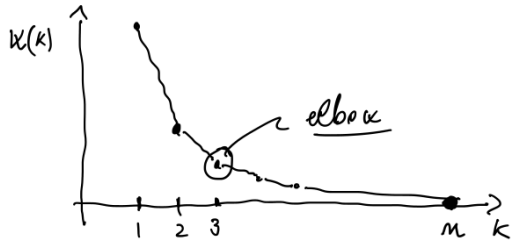
We can run the algorithm for different k and find the right one:

$$W(k) = \sum_{j=1}^{k} \sum_{\underline{x}_i \in C_j} d^2(\underline{x}_i, \underline{\bar{x}}_j)$$

We plot this with respect to k

Note that if $k = n$ then each point is a cluster and then: $W(k) = 0$

We select k when in the plot of $W(k)$ we see en elbow, for instance:



The less is k then the easier it is to explain what is going on!

When we work with function, tensors, images, and not points: clustering can mostly only be done with $k-$means.

Note: we can work with different features: we take the data transform it (e.g: kernel methods) and then apply $k-$means.

We can use many clustering method, but the problem is deciding the similarity measure!

Note that there is no obvious link between principal components and clusters: it may be harmful to do first PCA and then clustering!

The same holds for PCA and classification!

Graphical representations

We would like to reduce the dimensionality so that we can spot clusters more easily. There are many possibilities:

- We can use Fisher's Scores after LDA: after clustering we have labels so we can use LDA and then take Fisher's Scores.
- We can use **multi-dimensional scaling (MDS):**

  This is a technique for clustering and graphical representation.

  We have our training set $\tau = \{\underline{x}_i : i = 1, \dots, n\} \subset \mathbb{R}^p$ in $\mathbb{R}^p$ and we have our distance d

  We want to represent the problem in $\mathbb{R}^q$ with $q < p$ (for graphical reason it's nice if $q = 1, 2, 3$) then: we want to use the euclidean distance d_e

  Moreover we are willing to change the data set into a new one: $\tau' = \{\underline{y}_i : i = 1, \dots, n\}$ with $\underline{y}_1, \dots, \underline{y}_n$ such that: $d_e(\underline{y}_i, \underline{y}_j) \approx d(\underline{x}_i, \underline{x}_j)$

  Note: What do we mean by close in the above? it needs to be specified!

  Then if $D = [d_{ij}]$ is the original distance matrix, then we want to find $\Delta = [\delta_{ij}]$ which is the Euclidean distance matrix, such that: $d_{ij} \approx \delta_{ij}$

  Example: if we have different cities in the world, as distance between two cities we can take fly time, indeed cities are on a sphere so euclidean distance wouldn't be good. Moreover fly time could be asymmetric because of streams!

Suppose we render flight time symmetric: we then want a map in $\mathbb{R}^2$ with all the cities and the distance between the points is as close as possible to the fly time.

The problem with this is that the solution is not unique, because in an euclidean space any other solution we get by rigid transformation will preserve the euclidean distance, and it will still be a valid transformation!

Thus the first solution is not very evocative: so we need to try it many times!

So we can reflect, rotate and translate, so that maybe we get a good interpret-able result!

If $d = d_e$ that is: we started from euclidean distance, the linear sub-space of dimension q closest to the original training set is given by the PCA, so it's given by the space spanned by stopping at the first q principal components!

Thus we don't use multidimensional scaling but we just apply PCA, indeed:

- To perform PCA we need to decompose a $p \times p$ matrix
- To perform MDS we need to decompose a $n \times n$ matrix

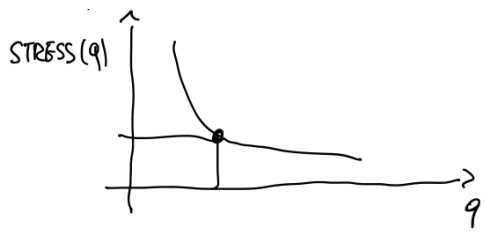
Objective function for MDS: what do we mean by close? What do we mean by approximating D with Δ? there are a few options:

- Classical MDS:
  we minimise the sum of squares deviations
  $$\sum_{i \neq j} (d_{ij} - \delta_{ij})^2$$
- More recent MDS (Kruskal et.al):
  we minimise the stress
  $$STRESS = \frac{\sum_{i \neq j} (\theta(d_{ij}) - \delta_{ij})^2}{\sum_{i \neq j} \delta_{ij}^2}$$
  where $\theta : \mathbb{R} \to \mathbb{R}$ is a monotone (increasing) function!

  we want to minimise $STRESS$ both with respect to τ' and with respect to θ

  Obviously we evaluate only for θ in a certain class otherwise there are an infinite amount of different θ!

  Then we choose q by looking at STRESS scree-plot:



  We stop at an elbow!

Note that there is no Theorem that says we get the right thing: sometimes it works!

Note: There are people that in the STRESS take the fourth power instead of the square and so on.

Moreover note that MDS can be use in principle for clustering: we use MDS and then we have a training set that is in $\mathbb{R}^q$ with d_e so we can then apply any clustering technique!

MDS offers us the transformation of the features!

19 Lecture 31: 7th Of May 2020

Regression: CART and linear models for regression in multivariate setting

We have training set (data set) given by n statistical units:

$$\mathbb{X} = \begin{pmatrix} x_1 & \dots & x_p & y \\ x_{11} & \dots & x_{1p} & y_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & \dots & x_{np} & y_n \end{pmatrix} = \begin{bmatrix} \underline{x}_1^T & \ell_1 \\ \vdots & \vdots \\ \underline{x}_n^T & \ell_n \end{bmatrix}$$

We want to use information from $\underline{x}_i$ to predict y: what is the link?We want an empirical model connecting $\underline{x}_i$ with y not necessarily for prediction!

Note: The $\underline{x}_i$ are covariates: independent variables! Whereas y is the output or dependent variable!

So our general goal is to *explain* y in terms of $\underline{x}_i$ So we want to explain the distribution (or variability) of y in terms of $\underline{x}_i$

Note: if y was a deterministic constant there would be no statistics to be performed!

Note: Regression is concerned with estimating: $\mathbb{E}[Y|\underline{X} = \underline{x}] = f(\underline{x})$ this function f is unknown to us, and it's the so called *regression function*!

We want to use $\mathbb{X}$ to estimate f by means of a function $\hat{f}$

Note: If y is a label this becomes a classification problem!

Suppose that $y \in \mathbb{R}$ and $\underline{x}_i \in \mathbb{R}^p$ Here we assume y is a quantitative variable!

There are many approaches but we focus on two basic approaches:

- Totally Non-Parametric (data driven) approach: this is fine when we don't know much about f

  This is good for prediction if we take good care of the bias-variance trade-off as we have a problem of over-fitting the data!

  This is less fine when we know something about f as we do in engineering!

- Parametric (model based) approach: we use some prior knowledge about the problem!

  This is good for prediction and for interpretation: we can quantify the uncertainty about the prediction and of the parameters we are fitting.

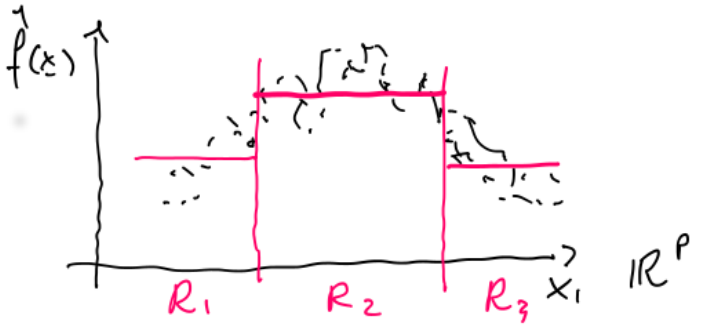
We focus mainly on the parametric approach!

We talk now a bit about a few tools about the non-parametric approach:
CART: Classification and Regression Trees are the basic brick through which we build up new ways to do computer intensive regression totally data driven!

The basic idea is that $\hat{f}$ is piece-wise constant: we need to build a partition $\{R_1, \dots, R_j\}$ of the features (covariates) space $\mathbb{R}^p$ and in each element of the partition we estimate y through a constant!

Note: j is finite but unknown!

So we assume $\hat{f}$ to be constant over the elements R_i of the partition! We have the following situation:



CART takes the mean of the y observed in that specific element of the partition as the constant to be associated to $\hat{f}$ over that element of the partition.

So: $\bar{y}_i = \frac{1}{\#R_i} \sum_{\underline{x}_j \in R_i} y_i \; \forall i = 1, \dots, j$ so that: $\hat{f}(\underline{x}) = \sum_{i=1}^{j} \bar{y}_i \mathbb{1}[\underline{x} \in R_i]$

Note: the partition $\{R_1, \dots, R_j\}$ is a finite partition of $\mathbb{R}^p$ such that: $R_i \cap \{\underline{x}_1, \dots, \underline{x}_n\} \neq \emptyset \; \forall i = 1, \dots, j$ So we need observations in each element of the partition!

How to find the best $\{R_1, \dots, R_j\}$ and j? We introduce a cost function which imposes that the mean is a good representative for all the y corresponding to the units in that partition.

So our goal is to minimise with respect to $\{R_1, \dots, R_j\} \wedge j$ the following:

$$\sum_{i=1}^{j} \sum_{\underline{x}_i \in R_j} (y_i - \bar{y}_i)^2$$

Note that we find groups among $\underline{x}_i$, clusters, minimising the variability of the y_i which is what we want to predict!

Note: the above looks very similar to $k-$means!

The optimal situation is the one in which we have one point in each partition but this completely over-fits data: zero error on training set but huge prediction error!

So we minimise the above without over-fitting the data: we find a greedy algorithm to solve (in the statistical sense: there are many solutions) this optimisation problem! Indeed **CART** does this!

So **CART** is an iterative process with the following steps:

- **Step 1:** consider the cut-off $s_1 \in \mathbb{R}$ such that: $\sum_{i=1}^{n} (y_i - \bar{y})^2 - \left[\sum_{x_{j1} < s_1} (y_j - \bar{y}_1)^2 + \sum_{x_{j1} \geq s_1} (y_j - \bar{y}_2)^2 \right]$ is maximised!

  Where: $\bar{y}_1 = \frac{1}{\#\{j : x_{j1} < s_1\}} \sum_{x_{j1} < s_1} y_j$ and $\bar{y}_2 = \frac{1}{\#\{j : x_{j1} \geq s_1\}} \sum_{x_{j1} \geq s_1} y_j$

  We are maximising the total variability minus the sum of the single total variability we have after the set is split in two!

  We find the split with respect to the first variables which makes the larger difference in terms of the total variability: the two groups after the split have less variability that the total variability we have before the split!

Repeat for $x_2, \ldots, x_p$ This generates the split $s_1, \ldots, s_p$

Now we choose the cut-off s_j^* which maximises:

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 - \left[\sum_{x_{j1}<s_1}(y_j - \bar{y}_1)^2 + \sum_{x_{j1}\geq s_1}(y_j - \bar{y}_2)^2\right]$$

Hence $\mathbb{R}^p$ is split into: $R_1 = \{\underline{x} \in \mathbb{R}^p : x_j \leq s_j^*\}$ and $R_2 = \{\underline{x} \in \mathbb{R}^p : x_j > s_j^*\}$

We now iterate **Step 1** on the elements of partition!

We stop when the number of units within an element of a partition is less than a certain threshold which we need to decide!

We don't want to stop at 1 because as said above we would be over-fitting!

So we stop partitioning R_j if $\#R_j < C$ where C needs to be decided!

Anyway the problem is over-fitting: minimise bias and maximise variability: maximising prediction error!

We take care of over-fitting to grow a large tree following the above algorithm and then we prune the tree: we penalise large tees.

Note: in linear models this is analog of penalising models with too many variables!

Note: in the above we get a partition in hyper-rectangles: but we can generalised to different shapes!

Thus, we choose $\alpha > 0$, by cross-validation, we grow a large tree following the above algorithm and then we prune the tree bottom-up: we merge some rectangles to minimise the following cost function:

$$W(m, \alpha) = \sum_{i=1}^{m}(y_i - \bar{y})^2 + \alpha m$$

This function depends on m and α: wee fix α, then we plot with respect to m and choose m with the elbow plot!

CART don't beat linear models, but they are building blocks of more efficient methods which beat linear models!

For instance random forest: we build up many trees, from many different training set! So we have an ensemble of trees not so much correlated: the final prediction is the average!

Note that we build many training set by bootstrap, and we can also bootstrap the variables!

The only problem is that we loose interpretation of the prediction!

Linear models for regression, model base (parametric) approach

We have training set (data set) given by n statistical units:

$$\mathbb{X} = \begin{pmatrix} x_1 & \ldots & x_p & y \\ x_{11} & \ldots & x_{1p} & y_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & \ldots & x_{np} & y_n \end{pmatrix} = \begin{bmatrix} \underline{x}_1^T & \ell_1 \\ \vdots & \vdots \\ \underline{x}_n^T & \ell_n \end{bmatrix}$$

Now we build the design matrix: $\mathbb{Z} = \begin{bmatrix} 1 & z_{11} & \dots & z_{1r} \\ 1 & z_{21} & \dots & z_{2r} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & z_{n1} & \dots & z_{nr} \end{bmatrix}$ is a $n \times (r+1)$ matrix, where the columns $z_1, \dots, z_r$ are known functions (transformations) of $x_1, \dots, x_p$ which we need to cook up!

Linear model: $\mathbb{E}[Y|z_1, \dots, z_r] = \beta_0 + \beta_1 z_1 + \dots + \beta_r z_r$ where β_i are the unknowns $r+1$ parameters!

So the model for Y is:

- Given $\underline{z} = (1, z_1, \dots, z_r)^T$ we have: $Y = f(\underline{z}) + \epsilon$
- f is capturing all the information the data is giving us in terms of estimating the mean and ϵ is the part left out and it's called residual.
- $f(\underline{z}) = \beta_0 + \beta_1 z_1 + \dots + \beta_r z_r$
- $\mathbb{E}[\epsilon] = 0$ and $\epsilon \perp\!\!\!\perp \underline{z}$

The model is linear in the $\underline{z}$, but z_i could be a non linear function of an x_i! So we can model non linear behaviour in the original variables!

Consider the data: $\underline{y} = (y_1, \dots, y_n)^T$ (which even this could be transformed of the original target variable) then consider the design matrix: $\mathbb{Z} = \begin{bmatrix} 1 & z_{11} & \dots & z_{1r} \\ 1 & z_{21} & \dots & z_{2r} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & z_{n1} & \dots & z_{nr} \end{bmatrix}$

The model is: $\underline{y} = \mathbb{Z}\underline{\beta} + \underline{\epsilon}$ where: $\underline{\beta} = (\beta_0, \dots, \beta_r)^T$ and $\underline{\epsilon} \in \mathbb{R}^n$ is such that: $\mathbb{E}[\underline{\epsilon}] = \underline{0}$

Assumption: suppose that $Cov(\underline{\epsilon}) = \sigma^2 I$ so they are not independent but just uncorrelated! Note that σ^2 needs to be estimated!

Then: $y_i = \beta_0 + \beta_1 z_{i1} + \dots + \beta_r z_{ir} + \epsilon_i$ with $\epsilon_1, \dots, \epsilon_n$ uncorrelated with same variance and mean zero, which are independent of the $\underline{z}_i$

Linear models are very flexible: for instance one way ANOVA is a particular case of a linear model! Indeed: $X_{11}, \dots, X_{1n_1} \overset{iid}{\sim} \mathcal{N}(\underline{\mu}_1, \sigma^2)$ and so on and so forth until $X_{g1}, \dots, X_{gn_g} \overset{iid}{\sim} \mathcal{N}(\underline{\mu}_g, \sigma^2)$ which are all independent!

now let $\underline{y} = (x_{11}, \dots, x_{1n_1}, x_{21}, \dots, x_{gn_g})^T \in \mathbb{R}^n$ where: $n = n_1 + \dots + n_g$ Then the design matrix is:

$$\mathbb{Z} = [\underline{z}_0\ \underline{z}_1\ \dots\ \underline{z}_g] \text{ where: } \underline{z}_0 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \text{ and } \underline{z}_1 = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Big\} n_1 \text{ and so on and so forth: } \underline{z}_g = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \Big\} n_g$$

Moreover: $\underline{\beta} = (\mu, \tau_1, \dots, \tau_g)$ So that: $\underline{y} = \mathbb{Z}\underline{\beta} + \underline{\epsilon}$

To have ANOVA we need to assume that $\underline{\epsilon} \sim \mathcal{N}(\underline{0}, \sigma^2 I)$

Thus: $x_{ij} = \mu + \tau_i + \epsilon_{ij}$ for $i = 1, \dots, g$ and $j = 1, \dots, n_i$ and $\epsilon_{ij} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

In the above $\mathbb{Z}$ is not full rank: indeed summing the columns from the second to last we get the first column!

Indeed we have an over-parametrisation of the problem as we have seen in ANOVA, so we need to impose: $\sum_{i=1}^{g} n_i \tau_i = 0$ so that we don't over-parametrise the problem!

To satisfy that constraint is enough to take $\mathbb{Z}$ as follows:

$$\mathbb{Z} = [\underline{z}_0 \ \underline{z}_1 \ \dots \ \underline{z}_g] \text{ where: } \underline{z}_0 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \text{ and } \underline{z}_1 = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \\ -\frac{n_1}{n_g} \\ \vdots \\ -\frac{n_1}{n_g} \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} n_1 \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} n_g \end{matrix} \quad \text{and so on and so forth: } \underline{z}_g = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \\ -\frac{n_{g-1}}{n_g} \\ \vdots \\ -\frac{n_{g-1}}{n_g} \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} n_{g-1} \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} n_g \end{matrix}$$

Indeed from the above linear constraint we have: $\tau_g = -\frac{1}{n_g} \sum_{i=1}^{g} n_i \tau_i$

In this case $\mathbb{Z}$ is an $n \times g$ matrix and it is full rank, whereas above it was an $n \times (g+1)$ and it wasn't full rank!

Here the $\mathbb{Z}$ specifies that each statistical unit come from a certain group of the treatment!

Before instead $\mathbb{Z}$ was made up of $z_1, \dots, z_{g+1}$ dummy variables: we didn't get it from data set it is just there to remind us that information we are analysing comes from the different groups!

Moreover we can add more quantitative variables so we can enrich the ANOVA to get the so called **ANCOVA**!

Fitting the linear model: how do we estimate $\beta_0, \dots, \beta_r$ and σ^2?

We see a classical way, called **Ordinary Least Squares (OLS)** Here we have that the following can be solved analytically:

$$\underline{\hat{\beta}} = \arg \min_{\underline{\beta} \in \mathbb{R}^{r+1}} \left\| \underline{y} - \mathbb{Z}\underline{\beta} \right\|^2$$

If $\mathbb{Z}$ is full rank then: $\underline{\hat{\beta}} = (\mathbb{Z}^T \mathbb{Z})^{-1} \mathbb{Z}^T \underline{Y}$

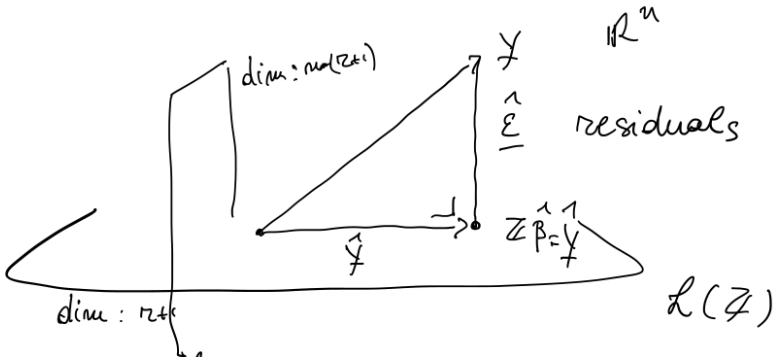
This is a linear model: indeed we see that we are just doing a linear transformation of $\underline{y}$ through a function of the design matrix!

Proof of the above: $\underline{y} = \mathbb{Z}\underline{\beta} + \underline{\epsilon}$ is our model for the data.

We have that $\mathbb{Z}\underline{\beta} \in \mathcal{L}(\mathbb{Z})$ since its a linear combination of the columns of $\mathbb{Z}$, so it belong to the linear space spanned by the columns of $\mathbb{Z}$

We are trying to find the vector closest to $\underline{y}$ in terms of euclidean distance in this space: the orthogonal projection!

So the optimisation problem is solved through projection and we get $\mathbb{Z}\underline{\hat{\beta}}$ We have the following situation:



$\underline{\hat{\beta}}$ is such that: $\mathbb{Z}\underline{\hat{\beta}} = \pi_{\underline{y}|\mathcal{L}(\mathbb{Z})}$ To compute this projection we need an ortho-normal basis for the space spanned by the columns of $\mathbb{Z}$

Note: since $\mathbb{Z}$ is full rank we have that $\mathbb{Z}^T\mathbb{Z}$ is an $(r+1)(r+1)$ square matrix which is full rank so that it's invertible!

Since $\mathbb{Z}^T\mathbb{Z}$ is symmetric and invertibile we take its spectral decomposition:

$$\mathbb{Z}^T\mathbb{Z} = \sum_{i=1}^{r+1} \lambda_i \underline{e}_i \underline{e}_i^T \text{ with } \lambda_1 \geq \cdots \geq \lambda_{r+1} > 0 \implies (\mathbb{Z}^T\mathbb{Z})^{-1} = \sum_{i=1}^{r+1} \lambda_i^{-1} \underline{e}_i \underline{e}_i^T$$

Note: if $\mathbb{Z}$ is not full rank we take the pseudo-inverse!

Now set: $\underline{q}_i = \sqrt{\frac{1}{\lambda_i}}\mathbb{Z}\underline{e}_i$ with $i = 1, \ldots, r+1$ but:

1) $\underline{q}_i \in \mathcal{L}(\mathbb{Z})$ by definition

2) $\underline{q}_i^T \underline{q}_j = \sqrt{\frac{1}{\lambda_i\lambda_j}} \left(\underline{e}_i^T \mathbb{Z}^T \mathbb{Z} \underline{e}_j\right) = \lambda_j \frac{1}{\sqrt{\lambda_i\lambda_j}} \underline{e}_i^T \underline{e}_j = \begin{cases} 0 \text{ if } i \neq j \\ 1 \text{ if } i = j \end{cases}$

So we have $r+1$ vectors that are ortho-normal so we have a basis for $\mathcal{L}(\mathbb{Z})$ which is given by: $\{\underline{q}_1, \ldots, \underline{q}_{r+1}\}$

Now we can project $\underline{y}$ on the elements of this basis:

$$\pi_{\underline{y}|\mathcal{L}(\mathbb{Z})} = \sum_{i=1}^{r+1} \pi_{\underline{y}|\underline{q}_i} = \sum_{i=1}^{r+1} \underline{q}_i \underline{q}_i^T \frac{1}{\underline{q}_i^T \underline{q}_i} \underline{y} = \sum_{i=1}^{r+1} \lambda_i^{-1} \mathbb{Z}\underline{e}_i \underline{e}_i^T \mathbb{Z}^T \underline{y} = \mathbb{Z}\left(\sum_{i=1}^{r+1} \lambda_i^{-1} \underline{e}_i \underline{e}_i^T\right) \mathbb{Z}^T \underline{y} = \mathbb{Z}(\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\underline{y}$$

Thus we can conclude that:

$$\pi_{\underline{y}|\mathcal{L}(\mathbb{Z})} = \mathbb{Z}(\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\underline{y} = \underline{\hat{y}}$$

which are the fitted values.

Note that $\pi_{\underline{y}|\mathcal{L}(\mathbb{Z})} = \underline{\hat{y}} = H\underline{y}$ where H is the **Hat Matrix**: it's the operator that puts a hat on $\underline{y}$!

Thus we have that: $\mathbb{Z}\underline{\hat{\beta}} = \underline{\hat{y}} \implies \underline{\hat{\beta}} = (\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\underline{y} = \mathbb{Z}^{-1}H$ which is the thesis

Definition: $\underline{\hat{y}} = H\underline{y}$ are the fitted values and $\underline{\hat{\epsilon}} = \underline{y} - H\underline{y} = (I - H)\underline{y}$ is the vector of the residuals!

By definition it follows that: $\underline{\hat{y}} \perp \underline{\hat{\epsilon}}$ indeed: $\underline{y} = \underline{\hat{y}} + \underline{\hat{\epsilon}}$

The fitted values is what we can say about $\underline{y}$ from the feature $\underline{x}$, and the vector of residual is what can't be capture from the information we have in the features $\underline{x}$

Note that it's just Pythagoras Theorem: we are just saying that $\underline{y}$ is the sum of two orthogonal vectors!

Note: $dim(\mathcal{L}(\mathbb{Z})) = r + 1$ and $dim(\mathcal{L}^{\perp}(\mathbb{Z})) = n - (r + 1)$ Note that $\underline{\hat{\epsilon}}$ lives in $\mathcal{L}^{\perp}(\mathbb{Z})$

Thus if we take r very large we fill up $\mathbb{R}^n$ with the linear space generated by the columns of $\mathbb{Z}$! Moreover if $r = n - 1$ then we get no residual: perfect fitting! This is bad: over-fitting! No prediction power!

20 Lecture 32: 8th Of May 2020

We have seen that in Linear Models $\underline{y}$ vector of dependent variables that we want to model as: $\underline{y} = \mathbb{Z}\underline{\beta} + \underline{\epsilon}$ where $\underline{y} \in \mathbb{R}^n$ and $\mathbb{Z}$ is the $n \times (r+1)$ design matrix and $\underline{\beta} \in \mathbb{R}^{r+1}$

We know that $\underline{\epsilon} \in \mathbb{R}^n$ with $\mathbb{E}[\underline{\epsilon}] = \underline{0}$ and $Cov(\underline{\epsilon}) = \sigma^2 I$ where σ^2 is a parameter of the model needed to be estimated!

We have seen that **OLS** is equivalent to finding $\hat{\underline{y}} \in \mathcal{L}(\mathbb{Z})$ closet to $\underline{y}$ in the sense of euclidean distance in $\mathbb{R}^+$

Note: the right distance is the Euclidean one because $Cov(\underline{\epsilon}) = \sigma^2 I$ so we have an identity matrix!

If we had a different co-variance structure we would have had to consider the Mahalanobis Distance and we would have the Generalised Least Squares!

We have seen $\hat{\underline{y}}$ is the orthogonal projection of $\underline{y}$ on $\mathcal{L}(\mathbb{Z})$ where:

- $\hat{\underline{y}} = \pi_{\underline{y}|\mathcal{L}(\mathbb{Z})}$ is the vector of fitted values
- $\hat{\underline{\epsilon}} = \pi_{\underline{y}|\mathcal{L}^{\perp}(\mathbb{Z})}$ is the vector of residuals, which is different from $\underline{\epsilon}$

  We would like it to be a realisation of $\underline{\epsilon}$ but we will see it just a proxy as it can't be a realisation of it!
- $\hat{\underline{y}} \perp \hat{\underline{\epsilon}}$ and $\underline{y} = \hat{\underline{y}} + \hat{\underline{\epsilon}}$

Observations:

1) if $Rank(\mathbb{Z}) = r+1 \leq n$ so that $\mathbb{Z}$ is full rank, then we know that $\hat{\underline{y}} = \mathbb{Z}(\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\underline{y} = H\underline{y}$ so this is a linear transformation function of the design matrix. Note that H is the orthogonal projection (it's an operator) on $\mathcal{L}(\mathbb{Z})$

   Moreover $\hat{\underline{\beta}} = (\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\underline{y}$ and $\hat{\underline{\epsilon}} = (I - H)\underline{y}$ so the residuals and fitted values are linear combination of $\underline{y}$

2) if $Rank(\mathbb{Z}) = k < r+1 \leq n$ so that $\mathbb{Z}$ is not full rank, then the geometry is the same:
   we need to project $\underline{y}$ on $\mathcal{L}(\mathbb{Z})$ but now this space has less dimension than before, indeed its dimension is k!

   But: $(\mathbb{Z}^T\mathbb{Z}) = \sum_{i=1}^{r+1} \lambda_i \underline{e}_i \underline{e}_i^T$ with $\lambda_1 \geq \cdots \geq \lambda_k > 0 = \lambda_{k+1} = \cdots = \lambda_{r+1}$ so since we cannot invert $\mathbb{Z}$ we take the generalised Moore-Penrose Inverse, so that: $(\mathbb{Z}^T\mathbb{Z})^{\dagger} = (\mathbb{Z}^T\mathbb{Z})^{-} = \sum_{i=1}^{r+1} \lambda_i^{-1} \underline{e}_i \underline{e}_i^T$ then we proceed like yesterday!

   The projection is unique and is given by: $\hat{\underline{y}} = \mathbb{Z}(\mathbb{Z}^T\mathbb{Z})^{\dagger}\mathbb{Z}^T\underline{y}$

   Note that we have $r+1$ variables and k basis vectors so there is an infinite number of solutions for the β: the linear system is under-determined!

   One possibility is: $\hat{\underline{\beta}} = (\mathbb{Z}^T\mathbb{Z})^{\dagger}\mathbb{Z}^T\underline{y}$ but this is not unique anymore!

   Either we reduce the number of parameters (like we did in the ANOVA with a linear constraint) or we are happy to declare the infinite number of possible solutions!

3) If $Rank(\mathbb{Z}) = r+1 = n$ so that it's a squared full rank matrix, then the columns are now basis for the entire $\mathbb{R}^n$

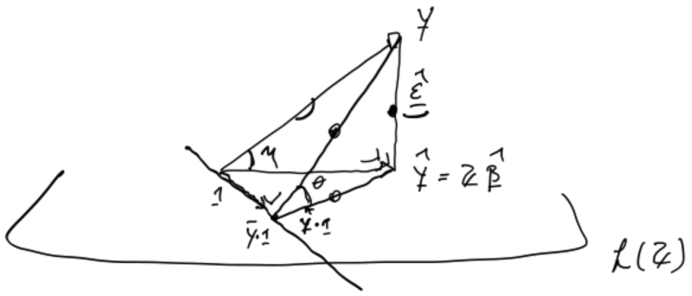
We have perfect fitting: $\hat{\underline{\epsilon}} = \underline{0} \implies \underline{y} = \hat{\underline{y}}$

This is no good: we are over-fitting, since we have perfectly interpolated $\underline{y}$!

We have no prediction power, and we have no way to estimate the variance since we have no residuals!

Note: if we had a deterministic phenomenon, then it would have been ok, otherwise we are just over-fitting!

How good is the fit of linear model? We need to define the coefficient of determination:



Assume $rank(\mathbb{Z}) = r+1 \leq n$ although it doesn't matter much! Then since the three vectors are orthogonal we have that: $\left\|\underline{y}\right\|^2 = \left\|\hat{\underline{y}}\right\|^2 + \|\hat{\underline{\epsilon}}\|^2 \implies \sum_{i=1}^{n} y_i^2 = \sum_{i=1}^{n} \hat{y}_i^2 + \sum_{i=1}^{n} \hat{\epsilon}_i^2$ which is: $SS_{tot} = SS_{reg} + SS_{res}$

Now $\underline{1} \in \mathcal{L}(\mathbb{Z})$ since it's its first column so we can project on it $\underline{y}$ to have: $\bar{y} \cdot \underline{1}$ and we get the same thing if we project $\hat{\underline{y}}$

That is: $\pi_{\underline{y}|\mathcal{L}(\underline{1})} = \bar{y} \cdot \underline{1} = \pi_{\hat{\underline{y}}|\mathcal{L}(\underline{1})}$ in fact:

- $\pi_{\hat{\underline{y}}|\mathcal{L}(\underline{1})} = \dfrac{\underline{1} \cdot \underline{1}^T}{\underline{1}^T \underline{1}} \hat{\underline{y}} = \dfrac{\underline{1} \cdot \underline{1}^T}{\underline{1}^T \underline{1}} H\underline{y}$
- But $\underline{1}^T H = (H^T \underline{1})^T$
- H is an orthogonal projection so it's symmetric, thus: $\underline{1}^T H = (H\underline{1})^T = \underline{1}^T$ since $\underline{1} \in \mathcal{L}(\mathbb{Z})$

So: $\pi_{\hat{\underline{y}}|\mathcal{L}(\underline{1})} = \dfrac{\underline{1} \cdot \underline{1}^T}{\underline{1}^T \underline{1}} H\underline{y} = \dfrac{\underline{1} \cdot \underline{1}^T}{\underline{1}^T \underline{1}} \underline{y} = \bar{y} \cdot \underline{1}$ Hence: $\left\|\underline{y} - \bar{y} \cdot \underline{1}\right\|^2 = \left\|\hat{\underline{y}} - \bar{\underline{y}} \cdot \underline{1}\right\|^2 + \|\hat{\underline{\epsilon}}\|^2$ which is the **decomposition of variance!** This can be re-written as:

$$\sum_{i=1}^{n} (y_i - \bar{y})^2 = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n} \hat{\epsilon}_i^2$$

which can be synthesised as $CSS_{tot} = CSS_{reg} + SS_{res}$ where CSS_{tot} is the Centred Sum Of Squares Total.

Thus from the above it follows that:

$$1 = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} + \frac{\sum_{i=1}^{n} \hat{\epsilon}_i^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \implies R^2 = 1 - \frac{\sum_{i=1}^{n} \hat{\epsilon}_i^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} = 1 - \frac{SS_{res}}{CSS_{tot}} = 1 - sin^2(\theta)$$

where θ is the angle between $\hat{\underline{\epsilon}}$ and $\hat{y} - \bar{y} \cdot \underline{1}$

So R^2 it's the proportion of total variability explained by the regression model! Note that:

- If $R^2 = 1$ then: $\theta = 0$ so $\underline{\hat{\epsilon}} = \underline{0}$ perfect fit! No good: we are over-fitting!
- If $R^2 = 0$ then: $\theta = \frac{\pi}{2}$ so $\underline{\hat{y}} - \bar{y} \cdot \underline{1} = 0$ Thus there is no improvement then using the mean for predicting!

Note: all of this works because the first column of the design matrix is made up of ones, so that $\underline{1} \in \mathcal{L}(\mathbb{Z})$

If $\underline{1} \notin \mathcal{L}(\mathbb{Z})$ we can't use R^2 Indeed all of the above doesn't hold as R^2 otherwise could even be negative, but it's a cosine squared, so it can't!

If $\underline{1} \notin \mathcal{L}(\mathbb{Z})$ then $\beta_0 = 0$ so we have: *Regression Through the Origin*

Note: sometimes we may want to do Regression Through The Origin in which case we wouldn't use R^2 though!

Note: we can make $\underline{1} \in \mathcal{L}(\mathbb{Z})$ even if we have regression through the origin!

Even if $\underline{1} \notin \mathcal{L}(\mathbb{Z})$ we still have that: $\left\|\underline{y}\right\|^2 = \|\hat{y}\|^2 + \|\hat{\epsilon}\|^2$ So in this case we can compute:

$$\tilde{R}^2 = 1 - \frac{\|\underline{\hat{\epsilon}}\|^2}{\left\|\underline{\hat{y}}\right\|^2}$$

This is a measure of fit but not a proportion of fit!

To take into account the Model Complexity we can compute the **Adjusted Coefficient of Determination**

Since $\underline{\hat{\epsilon}} \in \mathcal{L}^{\perp}(\mathbb{Z})$ but $dim(\mathcal{L}^{\perp}(\mathbb{Z})) = n - (r+1)$ Then it only has $n - (r+1)$ free components, all the other $r+1$ are fixed!

But $\underline{y} - \bar{y} \cdot \underline{1} \in \mathcal{L}^{\perp}(\underline{1})$ and this space has $n-1$ dimension: even though vector has n components we only have $n-1$ degrees of freedom!

So:

$$R^2_{adj} = 1 - \frac{\frac{1}{n-(r+1)} \sum_{i=1}^{n} \hat{\epsilon}_i^2}{\frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2}$$

Note that we divide by the degrees of freedom of the space in which each vector belongs to: we are trying to balance the bias-variance trade-off!

R^2_{adj} takes into account the complexity of the model! Note that if $r+1 = n$ then R^2_{Adj} isn't even compute-able because we have $1 - \frac{0}{0}$!

Properties of $\hat{\beta}, \hat{\epsilon}$

Assume that $rank(\mathbb{Z}) = r + 1 \leq n$ so that $\mathbb{Z}$ is full rank. Then:
Theorem:

1) $\mathbb{E}[\underline{\hat{\beta}}] = \underline{\beta}$ so $\underline{\hat{\beta}}$ is an unbiased estimator
2) $Cov(\underline{\hat{\beta}}) = \sigma^2(\mathbb{Z}^T\mathbb{Z})^{-1}$
3) $\mathbb{E}[\underline{\hat{\epsilon}}] = \underline{0}$
4) $Cov(\underline{\hat{\epsilon}}) = \sigma^2(I - H)$

5) $\mathbb{E}[\underline{\hat{\epsilon}}^T\underline{\hat{\epsilon}}] = \mathbb{E}\left[\sum_{i=1}^{n}\hat{\epsilon}_i^2\right] = \sigma^2(n-(r+1))$

Proof of the Theorem: it's just computation:

- $\mathbb{E}[\underline{\hat{\beta}}] = \mathbb{E}[(\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\underline{y}]$ but $\mathbb{E}[\underline{y}] = \mathbb{E}[\mathbb{Z}\underline{\beta}+\underline{\epsilon}] = \mathbb{Z}\underline{\beta}+\mathbb{E}[\underline{\epsilon}] = \mathbb{Z}\underline{\beta}+0$ thus: $\mathbb{E}[\underline{\hat{\beta}}] = (\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\mathbb{Z}\underline{\beta} = \underline{\beta}$
- $Cov(\underline{\hat{\beta}}) = Cov((\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\underline{y}) = (\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\left(I\sigma^2\right)\mathbb{Z}(\mathbb{Z}^T\mathbb{Z})^{-1} = \sigma^2(\mathbb{Z}^T\mathbb{Z})^{-1}$
- $\mathbb{E}[\underline{\hat{\epsilon}}] = \mathbb{E}[(I-H)\underline{y}] = \mathbb{E}[\underline{y}] - \mathbb{E}[H\underline{y}] = \mathbb{Z}\underline{\beta} - \mathbb{E}[\mathbb{Z}\underline{\hat{\beta}}] = \mathbb{Z}\underline{\beta} - \mathbb{Z}\underline{\beta} = 0$
- $Cov(\underline{\hat{\epsilon}}) = Cov((I-H)\underline{y}) = (I-H)\sigma^2\underline{y}(I-H)^T = \sigma^2(I-H)(I-H)^T = \sigma^2(I-H)^2 = \sigma^2(I-H)$ since $I-H$ is an orthogonal projection which is self-adjoint and idem-potent!
- $\mathbb{E}[\underline{\hat{\epsilon}}^T\underline{\hat{\epsilon}}] = tr\mathbb{E}[\underline{\hat{\epsilon}}^T\underline{\hat{\epsilon}}] = \mathbb{E}[tr(\underline{\hat{\epsilon}}^T\underline{\hat{\epsilon}})] = \mathbb{E}[tr(\underline{\hat{\epsilon}}\underline{\hat{\epsilon}})] = \mathbb{E}[tr((I-H)\underline{yy}^T(I-H)^T]$

  Since the trace is a linear operator, moreover $(\underline{\hat{\epsilon}}^T\underline{\hat{\epsilon}})$ is a number and $(\underline{\hat{\epsilon}}\underline{\hat{\epsilon}})$ is an $n\times n$ matrix and along its diagonal we have the residuals squared!

  $(I-H)\underline{y} = (I-H)(\mathbb{Z}\underline{\beta}+\underline{\epsilon}) = (I-H)\underline{\epsilon}$ Thus:

  $$\mathbb{E}[\underline{\hat{\epsilon}}^T\underline{\hat{\epsilon}}] = \mathbb{E}[tr((I-H)\underline{\epsilon\epsilon}^T(I-H)^T)] = tr((I-H)\mathbb{E}[\underline{\epsilon\epsilon}^T](I-H)^T) = \sigma^2 tr((I-H)(I-H)^T) = \sigma^2 tr(I-H)$$

  since $I-H$ is idem potent! So: $\mathbb{E}[\hat{\epsilon}^T\hat{\epsilon}] = \sigma^2(tr(I) - Tr(H)) = \sigma^2(n-tr(H))$

  But: $tr(H) = tr(\mathbb{Z}(\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T) = tr(\mathbb{Z}^T\mathbb{Z}(\mathbb{Z}^T\mathbb{Z})^{-1}) = tr(I_{r+1}) = r+1$ from which we get thesis, indeed:

  $$\mathbb{E}[\underline{\hat{\epsilon}}^T\underline{\hat{\epsilon}}] = \sigma^2(tr(I) - Tr(H)) = \sigma^2(n-(r+1))$$

Note:

1) $Cov(\underline{\hat{\beta}}) = \sigma^2(\mathbb{Z}^T\mathbb{Z})^{-1}$ since we have total control of the design matrix, we can choose it such that $(\mathbb{Z}^T\mathbb{Z})^{-1}$ will be small (i.e: with small trace or determinant) so that we are less uncertain about $\underline{\hat{\beta}}$

   Out of this there is the entire field called **Design of Experiments!**

- $\mathbb{E}[\underline{\hat{\epsilon}}^T\underline{\hat{\epsilon}}] = \sigma^2(n-(r+1)) \implies \mathbb{E}\left[\underline{\hat{\epsilon}}^T\underline{\hat{\epsilon}}\frac{1}{n-(r+1)}\right] = \sigma^2$ so we have an unbiased estimator!

  Definition: $S^2 = \frac{1}{n-(r+1)}\underline{\hat{\epsilon}}^T\underline{\hat{\epsilon}}$

  Corollary: S^2 is unbiased

Now we add an assumption: we assume from now on that $\underline{\epsilon} \sim \mathcal{N}_n(\underline{0}, \sigma^2 I)$ Thus, since $\underline{y} = \mathbb{Z}\underline{\beta}+\underline{\epsilon} \implies \underline{y} \sim \mathcal{N}_n(\mathbb{Z}\underline{\beta}, \sigma^2 I)$

Theorem: If $\underline{\epsilon} \sim \mathcal{N}_n(\underline{0}, \sigma^2 I)$ then:

1) $\underline{\hat{\beta}}$ and $\hat{\sigma}^2 = \underline{\hat{\epsilon}}^T\underline{\hat{\epsilon}}\frac{1}{n}$ are maximum likelihood estimators
2) $\underline{\hat{\beta}} \sim \mathcal{N}_{r+1}(\underline{\beta}, \sigma^2(\mathbb{Z}^T\mathbb{Z})^{-1})$
3) $\underline{\hat{\epsilon}} \sim \mathcal{N}_n(\underline{0}, \sigma^2(I-H))$
4) $\underline{\hat{\epsilon}} \perp\!\!\!\perp \underline{\hat{\beta}}$

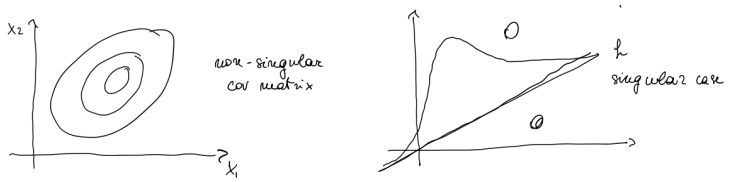
5) $\hat{\underline{\epsilon}}^T \hat{\underline{\epsilon}} = \sum_{i=1}^{n} \hat{\underline{\epsilon}}_i^2 \sim \sigma^2 \chi^2(n-(r+1))$

Note that we assume $\mathbb{Z}$ to be full rank!

Proof:

- For 1): Just compute derivative of log likelihood
- For 2),3) and 4) Note that:
$$\begin{bmatrix} \hat{\underline{\beta}} \\ \hat{\underline{\epsilon}} \end{bmatrix} = \begin{bmatrix} (\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T \\ I - \mathbb{Z}(\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T \end{bmatrix} \underline{y} \text{ and } \underline{y} \sim \mathcal{N}_n(\underline{0}, \sigma^2 I)$$
Then we just need to verify: $Cov \begin{bmatrix} \hat{\underline{\beta}} \\ \hat{\underline{\epsilon}} \end{bmatrix} = \begin{bmatrix} (\mathbb{Z}^T\mathbb{Z})^{-1} & 0 \\ 0 & I-H \end{bmatrix}$ and so since we have two zero blocks off diagonal than the two vectors are independent!
- For 5): $\hat{\underline{\epsilon}} \sim \mathcal{N}_n(\underline{0}, \sigma^2(I-H))$ but:
$(I-H)$ is a singular matrix, indeed $det(I-H) = 0$ as its rank is $n-(r+1)$ and not n!

  So we have a singular Gaussian distribution: it's a Gaussian distribution defined on a sub-space of Lebesgue measure equal to zero:



  So the error vector $\underline{\epsilon}$ is a non-singular Gaussian, whereas the residual vector $\hat{\underline{\epsilon}}$ is a singular Gaussian constrained on a linear sub-space!

  To prove this we need to compute: $\hat{\underline{\epsilon}}^T (I-H)^{-1} \hat{\underline{\epsilon}}$ but $I-H$ is singular!

  So we can compute the Moore-Penrose Generalised inverse so: $\hat{\underline{\epsilon}}^T (I-H)^{\dagger} \hat{\underline{\epsilon}} \sim \sigma^2 \chi^2(n-(r+1))$ and this is the Mahalanobis Distance of a Gaussian from its mean. Then: $\hat{\underline{\epsilon}}^T \hat{\underline{\epsilon}} \sim \sigma^2 \chi^2(n-(r+1))$

Since $\hat{\underline{\beta}} \sim \mathcal{N}_{r+1}(\underline{\beta}, \sigma^2(\mathbb{Z}^T\mathbb{Z})^{-1})$ and it's non singular since we take $\mathbb{Z}$ full rank, so, the Mahalanobis Distance of $\hat{\underline{\beta}}$ from its mean is:
$$(\hat{\underline{\beta}} - \beta)^T (\sigma^2(\mathbb{Z}^T\mathbb{Z})^{-1})^{-1} (\hat{\underline{\beta}} - \underline{\beta}) \sim \chi^2(r+1)$$
Moreover: $\hat{\underline{\epsilon}}^T \hat{\underline{\epsilon}} \sim \sigma^2 \chi^2(n-(r+1))$ and this is independent from the above one, so their ratio is a Fisher distribution, that is:
$$\frac{1}{r+1} \frac{1}{\sigma^2} (\hat{\underline{\beta}} - \underline{\beta})^T (\mathbb{Z}^T\mathbb{Z}) (\hat{\underline{\beta}} - \underline{\beta}) \frac{1}{\frac{\hat{\underline{\epsilon}}^T \hat{\underline{\epsilon}}}{\sigma^2(n-(r+1))}} \sim F(r+1, n-(r+1))$$
So the above becomes, since the denominator is $\sigma^2 S^2$:
$$(\hat{\underline{\beta}} - \underline{\beta})^T (\mathbb{Z}^T\mathbb{Z}) (\hat{\underline{\beta}} - \underline{\beta}) \frac{1}{S^2} \sim (r+1) F(r+1, n-(r+1))$$
and this is our pivotal quantity!

Now we can perform all of our statistical inference: we can build test, confidence region, confidence interval.

For example the confidence region for $\underline{\beta}$ is:

$$CR_{1-\alpha}(\underline{\beta}) = \left\{\underline{\beta} \in \mathbb{R}^{r+1} : \frac{1}{S^2}(\hat{\underline{\beta}} - \underline{\beta})^T(\sigma^2\mathbb{Z}^T\mathbb{Z})(\hat{\underline{\beta}} - \beta) \leq (r+1)F_\alpha(r+1, n-(r+1))\right\}$$

Thus $1-\alpha$ percent of the time we use this, which is an ellipse centred in $\hat{\underline{\beta}}$, we have that the true $\underline{\beta}$ is in it!

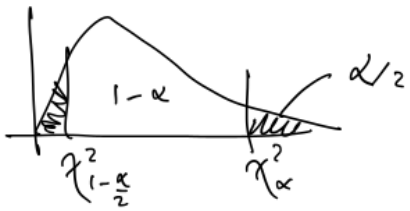
For example the confidence interval for σ^2 is:

$$CI_{1-\alpha}(\sigma^2) = \left\{\sigma^2 \in \mathbb{R} : \chi^2_{1-\alpha/2}(n-(r+1)) \leq (n-(r+1))S^2\frac{1}{\sigma^2} \leq \chi^2_\alpha(n-(r+1))\right\} =$$

$$= \left\{\sigma^2 \in \mathbb{R} : (n-(r+1))S^2\frac{1}{\chi^2_{\alpha/2}(n-(r+1))} \leq \sigma^2 \leq (n-(r+1))S^2\frac{1}{\chi^2_{1-\alpha/2}(n-(r+1))}\right\}$$

Indeed the sum of residuals squared is such that: $\hat{\underline{\epsilon}}^T\hat{\underline{\epsilon}} \sim \sigma^2\chi^2(n-(r+1)) \implies (n-(r+1))S^2\frac{1}{\sigma^2} \sim \chi^2(n-(r+1))$

Moreover, for the quantiles, we have that:



For example the Simultaneous Confidence Interval for $\underline{a}^T\underline{\beta}$ with $\underline{a} \in \mathbb{R}^{r+1}$ is:

From the Maximum Lemma we have that:

$$\max_{\underline{a}\in\mathbb{R}^{r+1}} \frac{1}{S^2}\frac{(\underline{a}^T(\hat{\beta}) - \underline{\beta}))^2}{\underline{a}^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{a})} = \frac{1}{S^2}(\hat{\underline{\beta}} - \underline{\beta})^T(\mathbb{Z}^T\mathbb{Z})(\hat{\underline{\beta}} - \underline{\beta}) \sim (r+1)F(r+1, n-(r+1))$$

So that:

$$SimCI_{1-\alpha}(\underline{a}^T\underline{\beta}) = \left\{\underline{a}^T\hat{\underline{\beta}} \pm \sqrt{\underline{a}^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{a}}\sqrt{S^2(r+1)F_{1-\alpha}(r+1, n-(r+1))}\right\}$$

21 Lecture 33: 11th Of May 2020

Recall the expression of the last lecture for the Simultaneous Confidence Interval for $\underline{a}^T\underline{\beta}$, then as a special case, we have:

$$SimCI_{1-\alpha}(\beta_i) \text{ for } i=0,\dots,r \text{ is given by } \left[\hat{\beta}_i \pm \sqrt{diag_i\left[(\mathbb{Z}^T\mathbb{Z})^{-1}\right]}\sqrt{S^2(r+1)F_\alpha(r+1,n-(r+1))}\right]$$

Observation: R will compute one-at-the-time confidence intervals:

$$CI_{1-\alpha}(\beta_i) \text{ for } i=0,\dots,r \text{ given by } \left[\hat{\beta}_i \pm \sqrt{diag_i\left[(\mathbb{Z}^T\mathbb{Z})^{-1}\right]}\sqrt{S^2}t_{\alpha/2}(n-(r+1))\right]$$

and not simultaneous confidence intervals! So we need to fix the $p-$values and the confidence levels!

The obvious compromise is to use Bonferroni's Confidence Intervals: so if we want overall level $1-\alpha$ then we need to ask R for a confidence interval of leve $1-\alpha/k$ where k is the number of β_i for which we want confidence intervals!

Testing for a set of β_i

Each β_i is the factor of increase on the expected value of $\underline{y}$ for an increase of one unit in the feature multiplying β_i

So β_i have strong impact: each one is the derivative of our model with respect to the variable that is multiplying that β_i

So after fitting the model we want to test the parameters β_i we have obtained.

A typical one is: $H_0 : C\underline{\beta} = \underline{0}$ vs $H_1 : C\underline{\beta} \neq \underline{0}$

This is without loss of generality, indeed we could consider: $H_0 : C\underline{\beta} = \underline{k}_0$ vs $H_1 : C\underline{\beta} \neq \underline{k}_0$

Note that C is a $p \times (r+1)$ matrix that is given: so we test p different linear combination of the components of $\underline{\beta}$ to see if they are zero or not!

We have that: $C\underline{\beta} = \begin{bmatrix} c_{11}\beta_0 + c_{12}\beta_1 + \cdots + c_{1(r+1)}\beta_r \\ \dots \\ c_{p1}\beta_0 + c_{p2}\beta_1 + \cdots + c_{p(r+1)}\beta_r \end{bmatrix}$ Then $C\hat{\underline{\beta}}$ is an estimator for $C\underline{\beta}$ and under H_0 we have:

$$C\hat{\underline{\beta}} \sim \mathcal{N}_p(\underline{0}, \sigma^2 C(\mathbb{Z}^T\mathbb{Z})^{-1}C)$$

Note that everything is all good since we have assumed $\mathbb{Z}$ to be full rank!

Now remember that:

$$\hat{\underline{\epsilon}}^T\hat{\underline{\epsilon}} \sim \sigma^2\chi^2(n-(r+1))$$

Now: $\hat{\underline{\epsilon}}^T\hat{\underline{\epsilon}} \perp\!\!\!\perp C\hat{\underline{\beta}} \implies$ Their quotient is an F distribution so that:

$$\frac{\frac{1}{p}(C\hat{\underline{\beta}})^T(\sigma^2 C(\mathbb{Z}^T\mathbb{Z})^{-1}C)^{-1}(C\hat{\underline{\beta}})}{\frac{\hat{\underline{\epsilon}}^T\hat{\underline{\epsilon}}}{\sigma^2(n-(r+1))}} \sim F(p, n-(r+1))$$

So that:

$$\frac{1}{S^2}(C\hat{\underline{\beta}})^T\left[C(\mathbb{Z}^T\mathbb{Z})^{-1}C^T\right]^{-1}(C\hat{\underline{\beta}}) \sim p \cdot F(p, n-(r+1))$$

So we reject at level α if:

$$\frac{1}{S^2}(C\hat{\underline{\beta}})^T\left[C(\mathbb{Z}^T\mathbb{Z})^{-1}C^T\right]^{-1}(C\hat{\underline{\beta}}) > pF_\alpha(p, n-(r+1))$$

A very special case is when we want to check if a sub-set of parameters are zero, that is: $H_0 = \beta_r = \beta_{r-1} = \cdots = \beta_{r-(p-1)} = 0$ vs $H_1 = \exists \beta_i \neq 0$ for $i = r-(p-1), \ldots, r$ thus we want to see if the last regressors $z_{r-(p-1)}, \ldots, z_r$ can be taken out of the model!

Indeed if their beta are not statistical significantly different from zero we remove these regressors and get a simpler model: we avoid over-fitting, bias-variance trade-off and curse of dimensionality!

Indeed we gain degrees of freedom for estimating the variability: we gain degrees of freedom for the space in which $\underline{\hat{\epsilon}}$ lives, so we have less uncertainty about σ^2!

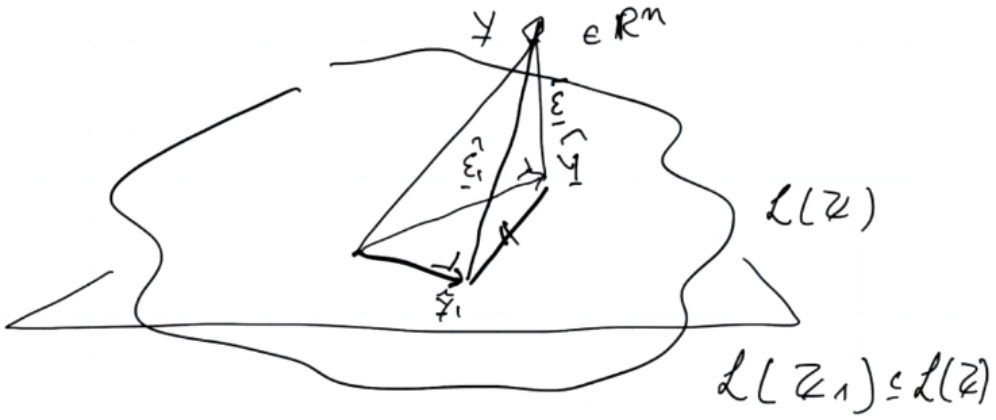
It's enough to take C, an $p \times (r+1)$ matrix, made up of all zeroes and then an identity matrix in the last p-columns, that is:

$$C = \begin{bmatrix} 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & 0 & \ldots & 0 & 1 \end{bmatrix} = [0 I_p]$$

Note that we can shuffle the β_i we want to test so that we don't have to necessarily use the last ones! So we can test for any set of the betas and see if they are zero, indeed the model is linear and additive and we can shuffle the coefficient times regressors terms!

Geometry of the above Test

Consider the following situation, where within $\mathcal{L}(\mathbb{Z})$ there is another linear space $\mathcal{L}(\mathbb{Z}_1)$



Thus: $\mathbb{Z} = [\mathbb{Z}_1, \mathbb{Z}_2]$ where $\mathbb{Z}_2$ is made up of the last p columns

With the above test we compare: $\underline{y} = \mathbb{Z}\underline{\beta} + \underline{\epsilon}$ with: $\underline{y} = \mathbb{Z}_1\underline{\beta}_1 + \underline{\epsilon}_1$ So checking if the last regressors are zero it's like we are comparing two models for explaining the same $\underline{y}$

Note that the two models are nested: the second one is obtained by the first one setting p regressors equal to zero!

Fitting the big model we project on $\mathbb{L}(\mathbb{Z})$ and when we fit the smaller model we project on $\mathcal{L}(\mathbb{Z}_1)$ If the difference between the two respective residual vectors is small then the models are very similar and so we can accept H_0

Thus we reject $H_0 : \beta_r = \beta_{r-1} = \cdots = \beta_{r-(p-1)} = 0$ if:

$$SS_{res}(\mathbb{Z}_1) - SS_{res}(\mathbb{Z}) = \underline{\hat{\epsilon}}_1^T \underline{\hat{\epsilon}}_1 - \underline{\hat{\epsilon}}^T \underline{\hat{\epsilon}} \text{ is big}$$

How big? It depends on the distribution. It can be proved that:

$$\frac{1}{S^2 p} \left(SS_{res}(\mathbb{Z}_1) - SS_{res}(\mathbb{Z}) \right) \sim F(p, n - (r+1))$$

so it's the same test as before!

Note that: $S^2 = \dfrac{\underline{\hat{\epsilon}}^T \underline{\hat{\epsilon}}}{n-(r+1)} = \dfrac{SS_{res}(\mathbb{Z})}{n-(r+1)}$

A more special case is the following: can we take out all the regressors? Is it worth doing the regression at all or can we take the mean of $\underline{y}$ for prediction and for explaining the phenomena?

Note: the intercept β_0 is the only thing we have in this case!

Thus we test: $H_0 : \beta_i = 0 \; \forall i = 1, \ldots, r$ vs: $H_1 : \exists \beta_i \neq 0$ for $i = 1, \ldots, r$

In this case: $\mathbb{Z}_1 = [1, \ldots, 1]^T$ so $\underline{\hat{\epsilon}}_1^T \underline{\hat{\epsilon}}_1 = \sum_{i=1}^{n} (y_i - \bar{y})^2$ so that:

$$\frac{(SS_{res}(\mathbb{Z}_1) - SS_{res}(\mathbb{Z}))}{S^2 p} = \frac{\left(\sum_{i=1}^{n} (y_i - \bar{y})^2 - \sum_{i=1}^{n} (\hat{\epsilon}_i)^2 \right)}{r \dfrac{\sum_{i=1}^{n} \hat{\epsilon}_i^2}{n-(r+1)}} = \frac{\dfrac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}{r}}{\dfrac{\sum_{i=1}^{n} \hat{\epsilon}_i^2}{n-(r+1)}} \sim F(r, n-(r+1))$$

This is the F test in the summary of the regression in the summary of R! Moreover note that here $p = r$

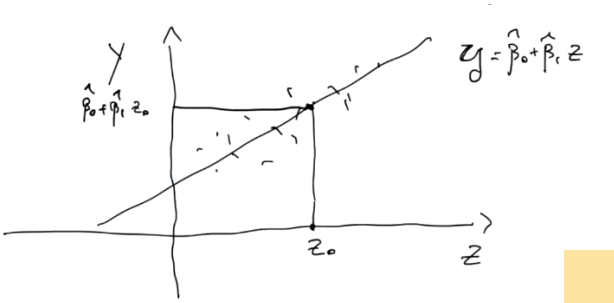
We are just checking if the variability explained by the model with respect to the variability explained by the residual is big enough to guarantee that the model has some meaning or not!

Regression model: prediction vs understand whats going on

Now we see how to see how to use the model to make prediction! Assume we have our model: $\underline{y} = \mathbb{Z}\underline{\beta} + \underline{\epsilon}$ model for the training set. Then:
let $\underline{z}_0 = (1, z_0, \ldots, z_{or})^T$ be a given vector capturing the values of the regression for which we want the prediction of y

Then: $y_0 = \underline{z}_0^T \underline{\beta} + \epsilon_0$ is a new statistical unit, with $\epsilon_0 \perp\!\!\!\perp \underline{\epsilon}$ and $\epsilon_0 \sim \mathcal{N}(0, \sigma^2)$

We want to predict y_0:



What's the uncertainty?

Using the line we are predicting the expected value of y_0 given $\underline{z}_0$ and not y_0 indeed: $\mathbb{E}[y_0|\underline{z}_0] = \underline{z}_0^T\underline{\beta}$ and a natural estimator for $\underline{z}_0^T\underline{\beta}$ is $\underline{z}_0^T\hat{\underline{\beta}}$

So the predictor is a linear combination of the $\hat{\beta}_i$

Gauss-Markov Theorem: $\underline{z}_0^T\hat{\underline{\beta}}$ is the best estimator, that is: the one with minimum variance, among those linear in $\underline{y}$ and unbiased!

So $\underline{z}_0^T\hat{\underline{\beta}}$ is BLUE, Best Linear Unbiased Estimator, for $\underline{z}_0^T\underline{\beta}$

Can we get something more about the estimate? What's the uncertainty?

We know that $\underline{z}_0^T\hat{\underline{\beta}} \sim \mathcal{N}_1(\underline{z}_0^T\underline{\beta}, \sigma^2\underline{z}_0^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0)$ since $\hat{\underline{\beta}} \sim \mathcal{N}_{r+1}(\underline{\beta}, \sigma^2(\mathbb{Z}^T\mathbb{Z})^{-1})$ and we are just taking a linear transformation of it!

Moreover: $\hat{\underline{\epsilon}}^T\hat{\underline{\epsilon}} \sim \sigma^2\chi^2(n-(r+1))$ and this is independent from $\underline{z}_0^T\hat{\underline{\beta}}$ Then:

$$\frac{\underline{z}_0^T\hat{\underline{\beta}} - \underline{z}_0^T\underline{\beta}}{\sqrt{\sigma^2\underline{z}_0(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0}} \sim \mathcal{N}_1(0,1)$$

Thus:

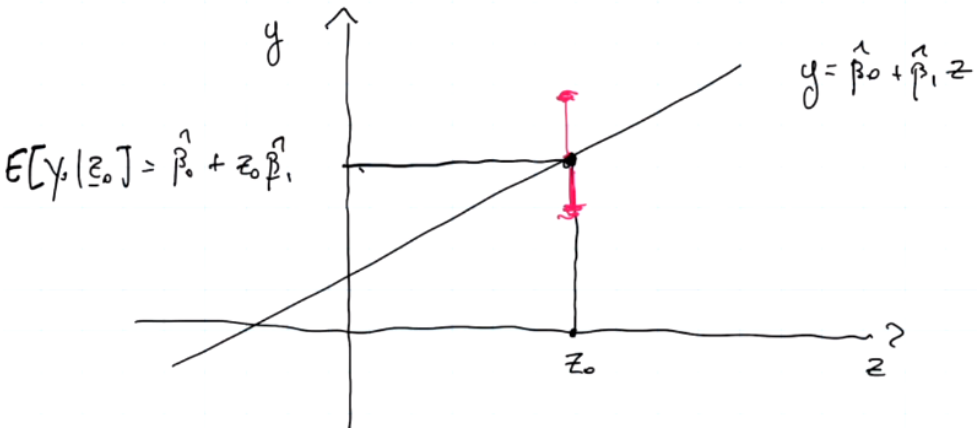
$$\frac{\frac{\underline{z}_0^T\hat{\underline{\beta}} - \underline{z}_0^T\underline{\beta}}{\sqrt{\sigma^2\underline{z}_0(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0}}}{\sqrt{\frac{\hat{\underline{\epsilon}}^T\hat{\underline{\epsilon}}}{\sigma^2(n-(r+1))}}} \sim t(n-(r+1)) \implies \frac{\underline{z}_0^T\hat{\underline{\beta}} - \underline{z}_0^T\underline{\beta}}{S\sqrt{\underline{z}_0^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0}} \sim t(n-(r+1))$$

The above one is our pivotal quantity so we can do all of our inference:
For example:

$$CI_{1-\alpha}(\underline{z}_0^T\underline{\beta}) = CI_{1-\alpha}(\mathbb{E}[y_0|\underline{z}_0]) = \left[\underline{z}_0^T\hat{\underline{\beta}} \pm S\sqrt{\underline{z}_0^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0}\, t_{\alpha/2}(n-(r+1))\right] \text{ for } \alpha \in (0,1)$$

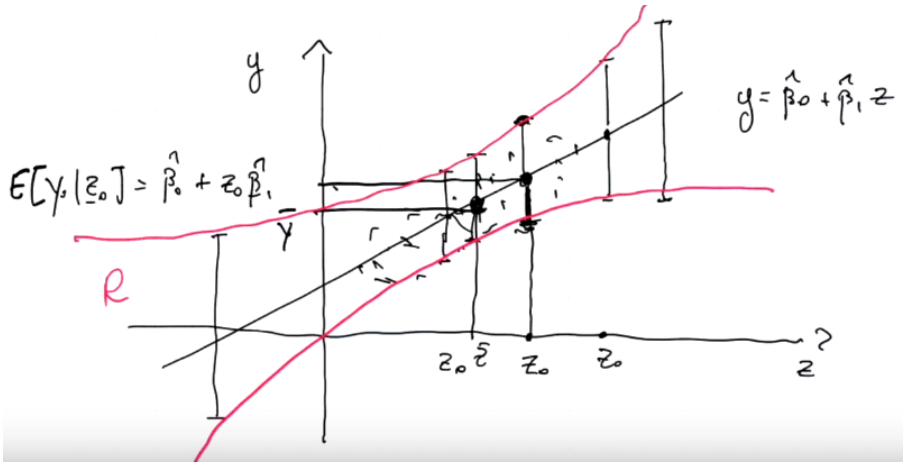
So we have the prediction point estimate and the uncertainty is given by the term to the right of $\pm$

There is one possible problem:



In the above we built a confidence interval for $\mathbb{E}[y_0|\underline{z}_0]$ and $1-\alpha$ times we use this confidence interval it covers the right value of $\mathbb{E}[y_0|\underline{z}_0]$

Now if repeat this for all possible $\underline{z}_0$ we have:



We can see that the closer we are to the baricenter $\bar{z}$ the smaller the intervals are.

This is because of the term $\sqrt{\underline{z}_0^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0}$ which increases as we move from the mean of the data!

Now here is the **problem**:
Consider the red R region, which is the envelope of all the confidence intervals. Does thee true regression line fall in this regione $1-\alpha$ percent of the time? **No!** R is the envelope of one-at-the-time confidence intervals and not of simultaneous confidence intervals!

Many people indeed interpret this region wrongly!

But we can take the simultaneous confidence intervals: we create a region that covers the linear model for all possible values of z

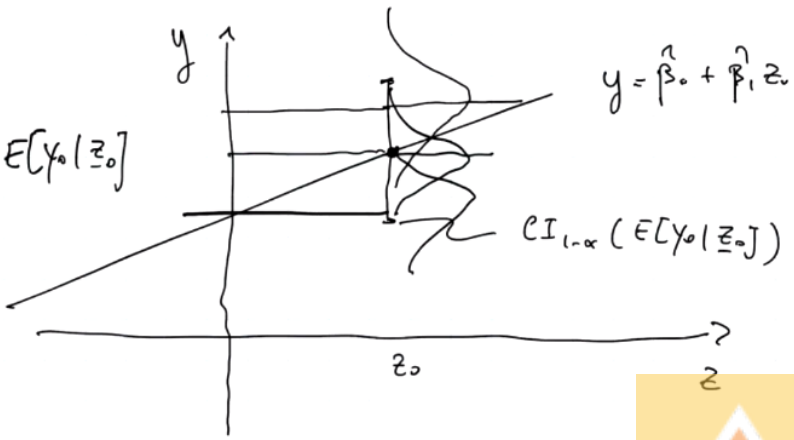
$$SimCI_{1-\alpha}(\underline{z}_0^T\underline{\beta}) = SimCI_{1-\alpha}(\mathbb{E}[y_0|\underline{z}_0]) = \left[\underline{z}_0^T\hat{\underline{\beta}} \pm S\sqrt{\underline{z}_0^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0}\sqrt{(r+1)F_\alpha(r+1, n-(r+1))}\right]$$

Plotting this we get a larger region that includes also the previous envelope R!

We can now guarantee that the overall confidence level is $1-\alpha$ percent even if make 1 billion prediction!

Within this new region moreover there will be the true linear model with confidence $1-\alpha$

In the above we have seen the prediction for $\mathbb{E}[y_0|\underline{z}_0]$ Well, what about y_0?



We now that: $y_0 \sim \mathcal{N}(\underline{z}_0^T\underline{\beta}, \sigma^2)$ through the linear model we predict it's mean!

We have through the confidence interval the uncertainty about the mean of the distribution of y_0!

Can we find an interval I such that: $\mathbb{P}[y_0 \in I|\underline{z}_0] = 1-\alpha$? Yes: this is a **Prediction Interval (PI)** for the actual observation of y_0

Obviously this PI will be larger than the CI for the mean: with the PI we are uncertain about the centre of the distribution and there is an extra variability due to the fact that y is different from its mean!

We know that: $\underline{z}_0^T\hat{\underline{\beta}} \sim \mathcal{N}_1(\underline{z}_0^T\underline{\beta}, \sigma^2\underline{z}_0^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0)$ this is in the training set!

Then y_0 is a new statistical unit: not in the training set! Thus: $y_0 \perp\!\!\!\perp \underline{z}_0^T\hat{\underline{\beta}}$ since $\epsilon_0 \perp\!\!\!\perp \underline{\epsilon}$

Thus: $y_0 - \underline{z}_0^T\hat{\underline{\beta}} \sim \mathcal{N}(0, \sigma^2(1+\underline{z}_0^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0))$ and $\hat{\underline{\epsilon}}^T\hat{\underline{\epsilon}} \sim \sigma^2\chi^2(n-(r+1))$ Moreover: $\hat{\underline{\epsilon}}^T\hat{\underline{\epsilon}} \perp\!\!\!\perp y_0 - \underline{z}_0^T\hat{\underline{\beta}}$ since $\hat{\underline{\epsilon}}^T\hat{\underline{\epsilon}} \perp\!\!\!\perp \hat{\underline{\epsilon}}^T$ and $\hat{\underline{\epsilon}}^T\hat{\underline{\epsilon}} \perp\!\!\!\perp \hat{\underline{\epsilon}}$

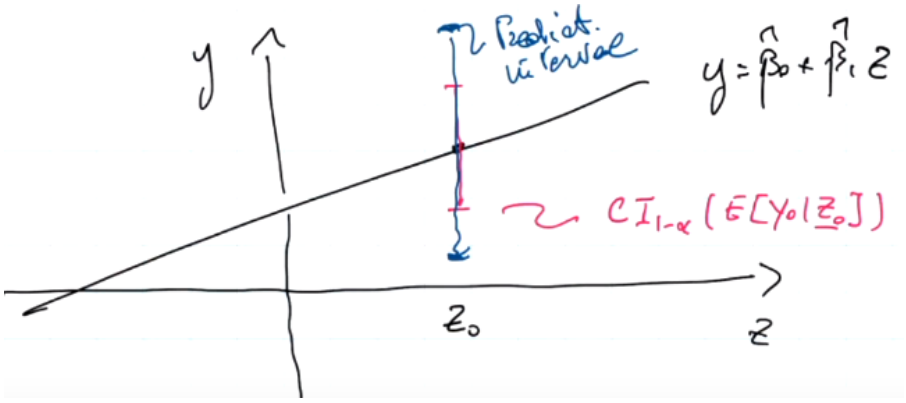
Thus $\dfrac{y_0 - \underline{z}_0^T\hat{\underline{\beta}}}{\sqrt{\sigma^2(1+\underline{z}_0^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0}} \sim \mathcal{N}(0,1)$ and $\sqrt{\dfrac{\hat{\underline{\epsilon}}^T\hat{\underline{\epsilon}}}{\sigma^2(n-(r+1))}}$ is a chi-squared. Thus their quotient is a $t-$student that is:

$$\frac{y_0 - \underline{z}_0^T\hat{\underline{\beta}}}{\sqrt{\dfrac{\hat{\underline{\epsilon}}^T\hat{\underline{\epsilon}}}{\sigma^2(n-(r+1))}}} = \frac{y_0 - \underline{z}_0^T\hat{\underline{\beta}}}{S\sqrt{1+\underline{z}_0^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0}} \sim t(n-(r+1))$$

So the prediction interval of probability $1-\alpha$ is:

$$PI_{1-\alpha}(y_0) = \left[\underline{z}_0^T\hat{\underline{\beta}} \pm S\sqrt{1+\underline{z}_0^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0}\, t_{\alpha/2}(n-(r+1))\right]$$

We see that the prediction interval is larger than the confidence interval:



Conclusion:

- Confidence interval are used for mean of what we want to predict
- Prediction interval are used for what we predict!

  They take into account the variability σ^2 of distribution plus the uncertainty about the mean of y, so that they are larger!

In the above we did one-at-the-time prediction interval so if we plot all of them we won't find a prediction region for y!

If we want such prediction region we use simultaneous prediction intervals:

$$SimPI_{1-\alpha} = \left[\underline{z}_0^T\hat{\underline{\beta}} \pm S\sqrt{1+\underline{z}_0^T(\mathbb{Z}^T\mathbb{Z})^{-1}\underline{z}_0}\sqrt{(r+2)F_\alpha(r+2,n-(r+1))}\right]$$

All we have done is for OLS, what happens if we have correlated errors? We have **GLS: Generalised Least Squares!**

$\underline{y} = \mathbb{Z}\underline{\beta} + \underline{\epsilon}$ with $\mathbb{E}[\underline{\epsilon}] = 0$ but now: $Cov(\underline{\epsilon}) = W\sigma^2 \neq \sigma^2 I$ as in OLS, where W is an $n \times n$ positive definite matrix.

We want to find:

$$\hat{\underline{\beta}} = \arg\min_{\underline{\beta}}(\underline{y} - \mathbb{Z}\underline{\beta})^T W^{-1}(\underline{y} - \mathbb{Z}\underline{\beta})$$

Note that if $W = I$ we have OLS! Note that we use W^{-1} for notation reason!

$$(\underline{y} - \mathbb{Z}\underline{\beta})^T W^{-1}(\underline{y} - \mathbb{Z}\underline{\beta}) = (W^{-1/2}\underline{y} - W^{-1/2}\mathbb{Z}\underline{\beta})^T(W^{-1/2}\underline{y} - W^{-1/2}\mathbb{Z}\underline{\beta}) = \left\|W^{-1/2}\underline{y} - W^{-1/2}\mathbb{Z}\underline{\beta}\right\|^2 \implies$$

$$\hat{\underline{\beta}} = (\mathbb{Z}^T W^{-1}\mathbb{Z})^{-1}\mathbb{Z}^T W^{-1}\underline{y}$$

Note: by changing notation: $\tilde{y} = W^{-1/2}y$ and $\tilde{\mathbb{Z}} = W^{-1/2}\mathbb{Z}$ we are back to OLS!

Indeed the above it's likes transforming data and then applying OLS! In fact:

$$W^{-1/2}y = W^{-1/2}\mathbb{Z}\underline{\beta} + W^{-1/2}\underline{\epsilon} \text{ but } Cov(W^{-1/2}\underline{\epsilon}) = \sigma^2 W^{-1}$$

For example assume $\underline{\epsilon}$ is such that: $Cov(\underline{\epsilon}) = \sigma^2\Sigma$ so the errors are correlated, and let σ^2 unknown and Σ be known!

Then take $W = \Sigma$ so that:

$$\hat{\underline{\beta}} = \arg\min_{\underline{\beta}}(\underline{y} - \mathbb{Z}\underline{\beta})^T\Sigma^{-1}(\underline{y} - \mathbb{Z}\underline{\beta}) = (\mathbb{Z}^T\Sigma^{-1}\mathbb{Z})^{-1}\mathbb{Z}^T\Sigma^{-1}\underline{y}$$

Note that: $Cov(\Sigma^{-1/2}\underline{\epsilon}) = \sigma^2\Sigma^{-1/2}\Sigma\Sigma^{-1/2} = \sigma^2 I$ so we are back to the euclidean case: we transformed the variables such that the error term is uncorrelated, with the same variance along the components!

Note: in first step above where we compute $\hat{\underline{\beta}}$ we use Mahalanobis Distance! Once the co-variance structure is such that we have the error uncorrelated we can use the euclidean distance!

Consider the special case in which $Cov(\underline{\epsilon}) = \sigma^2 diag(w_1, \ldots, w_n)$ so that: $\Sigma = diag(w_1, \ldots, w_n)\sigma^2$ so statistical units have different variance!

Thus we get **Weighted Least Squares** which is a special case of GLS!

An example of **Weighted Least Squares** is the following: suppose that y_i, with $i = 1, \ldots, n$, is a mean of n_i observations with the same variance! Then: $Var(y_i) = \frac{\sigma^2}{n_i}$

Note that this example could be: y_i is the average wage in Milan, y_{i+1} is the average wage in Turin, and so on and so forth. Here n_i is the population of the people in the different cities!

Then: $W = diag(\frac{1}{n_1}, \ldots, \frac{1}{n_n})$

Another example of **Weighted Least Squares** is the following: suppose we observe y_i for $i = 1, \dots, n$ as the sum of n_i observations! Then: $Var(y_i) = n_i\sigma^2$ so $W = diag(n_1, \dots, n_n)$

NOte that this example could be: y_i is the number of corona virus cases in Milan and y_{i+1} is the number of corona virus cases in Turn, and so on and so forth.

Note the importance of the fact that n_i are different so that they play with a different weight in the model!

Indeed we give more weight to observation of less variance, since we are less uncertain about them!

Thus corona virus cases can't just adjust the number of cases by population as the more population the greater the variance in the estimation!

Conclusion: Weighted Least Squares can be seen either as change of metric or as transformation of data!

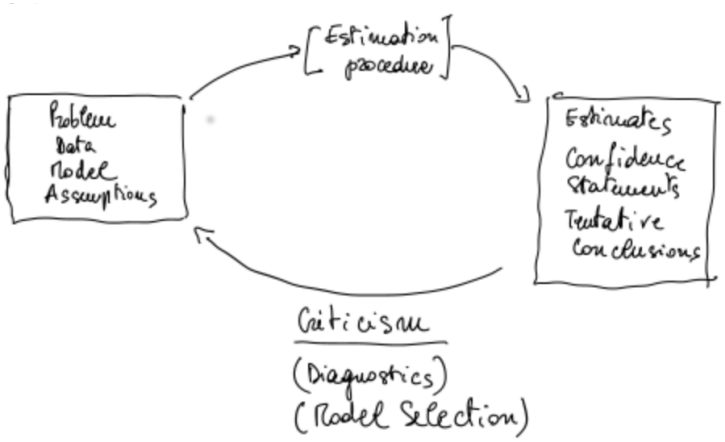
22 Lecture 34: 12th Of May 2020

Diagnostics for linear models

R^2 is not enough! After the analysis we need to check how good is the model, to check the assumption of the model, to check if the results we get from the model makes sense or not.

This is when we criticise the model and usually we end up with a new model better for the analysis!

Box described the process of data analysis as:



Diagnostics is part of the criticism: we don't stop with the first answer: do those answers make sense? What are the wrong assumptions? what are the weak parts of the analysis?

Although diagnostics is an art there is a set of basic tools we all use!

Take home message: criticise the model!

Diagnostics for linear models is based on some basic tools:

- **Residuals Analysis:** we look for outliers, we check for eteroschedasticity (i.e: non constant variance), we check for normality, we check for auto-correlation, and so on and so forth.
- **Influential Cases (Statistical Units)**
- **Collinearity among the regressors**

Residuals Analysis

Remember the difference between the error term and the residuals:

- $\underline{y} = \mathbb{Z}\underline{\beta} + \underline{\epsilon}$ with $\underline{\epsilon} \in \mathbb{R}^n$ with: $\mathbb{E}[\underline{\epsilon}] = \underline{0}$ and $Cov(\underline{\epsilon}) = \sigma^2 I$ and maybe its Gaussian but it's not necessary!
- $\underline{\hat{\epsilon}} = (I - H)\underline{y}$ is the projection of $\underline{y}$ on $\mathcal{L}^{\perp}(\mathbb{Z})$ although it has n components it belong to a sub-space: $\underline{\hat{\epsilon}} \in \mathcal{L}^{\perp}(\mathbb{Z})$

  We know that: $\mathbb{E}[\underline{\hat{\epsilon}}] = \underline{0}$ and $Cov(\underline{\hat{\epsilon}}) = \sigma^2(I - H)$ so the residuals are correlated!

  If $\underline{\epsilon}$ is Gaussian then $\underline{\hat{\epsilon}}$ is a singular Gaussian: $\underline{\hat{\epsilon}} \sim \mathcal{N}_n(\underline{0}, \sigma^2(I - H))$

So $\underline{\hat{\epsilon}}$ are not realisations of $\underline{\epsilon}$

Are there cases where the residual is big, so that the fitting of the model there is very poor? We search for large $\hat{\epsilon}_i$

We need to **Residual Plot:** We plot the fitted values $\hat{y}_i$ on the $x-$axis against the residuals $\hat{\epsilon}_i$ on the $y-$axis!

We would like to see small residuals with no shape: we would like a random cloud with roughly zero mean!

If we see a particular shape then there is some information that is left in the residuals that we can still capture through a different model!

If the residual plot is such that there is a variability that is not constant with respect to the fitted values then the model is no good!

For example if we have a funnel shaped residual plot then we have eteroschedasticity: non-constant variance! This is a problem that needs to be fixed!

Indeed our basic model assumes that the we have same variance! We can fix it:

- By weighted linear regression, if we have one of the last two cases presented last lecture (e.g: y_i are sums or averages of groups of different sizes).
- Otherwise we can transform the data y through a variance stabilising transformation. There are general tricks, such as taking the log if we have a kind of chi-squared distribution.

If in the residual plot we have a few points far off the cloud then maybe those points are outliers: they are outliers with respect to the residuals

But wait, outliers shouldn't be eliminated! Why are they outliers? Are they generated by errors in the data set?

Are these outliers signals of existence of a different population that we didn't sample? Sometimes we just want to spot and identify outliers!

We only throw them away when they aren't interesting and they just screw up the analysis!

We can also plot the residuals $\hat{\epsilon}_i$ on the $y-$axis and on the $x-$axis we put one regressor z_{ik}, where $k = 1, \ldots, r$

Here we want a nice cloud with no pattern!

If we don't get a nice random cloud, then maybe there is some polynomial relationships between the z_{ik} and y: so maybe we should insert this polynomial relationships in the model!

Or maybe there is some periodicity?

Note that: $Cov(\underline{\hat{\epsilon}}) = \sigma^2(I - H) \implies Var(\hat{\epsilon}_i) = \sigma^2(1 - h_{ii})$ for $i = 1, \ldots, n$ where $h_{ii} = diag_i(H)$

So $\hat{\epsilon}_i$ have different variances, so we might want to repeat the residual analysis using the residual standardised by their variance:

$$\frac{\hat{\epsilon}_i}{\sqrt{S^2(1 - h_{ii})}}$$

which are the **Studentised Residuals!**

Note: we use S^2 as it's a estimator of σ^2

Now with this plot an outliers is a case for which the Studentised Residuals is above 3 or above 4

Maybe we spot outliers in terms of the unit of measure of their variability, this way!

Gaussianty: We want to check Gaussianity, by making a QQplot of the residual or of the Studentised Residuals!

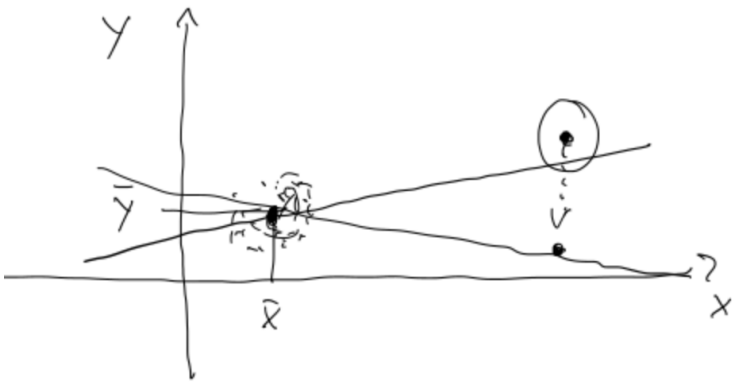
The residuals will never be Gaussian, we just need to check they are not too far from being Gaussian!

Note: in time series analysis we are worried about auto-correlation: how large is the auto-correlation? We can use the **Dublin-Watson Test!**

Influential Cases

We also want to have a control about the influential cases in the analysis. Are there cases that drive the entire analysis no matter how much data we use?

Consider for instance the following example:



The linear regression is strongly influenced by the datum far-off the right of the plot! That datum will have an important effect, the so called **leveraging effect**!

Indeed the regression goes through the mean of the baricenter and the linear regression tries to go near the influential point!

How do we get a grasp on influential points with big leveraging effect?

Note: $Var(\hat{\epsilon}_i) = \sigma^2(1 - h_{ii})$ for $i = 1, \dots, n$ and h_{ii} is called leveraging!

If $h_{ii} \approx 1 \implies Var(\hat{\epsilon}_i) \approx 0 \implies \hat{\epsilon}_i \approx 0$ since $\mathbb{E}[\hat{\epsilon}_i] = 0$ by definition!

But h_{ii} only depends on the design matrix, indeed: $h_{ii} = diag_i(\mathbb{Z}(\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T)$ So even before doing regression we can make sure, by using the design matrix, that an influential point has a small residual!

Watch Out: this can be done for mystification: how to lie with statistics!

Note: $0 \leq h_{ii} \leq 1$ Indeed this can be proved using the fact that: $H = H^T$ and $H^2 = H$

Leverages cases are important and we don't want them!

Another way to identify influential cases is to work out what happens by taking out that specific case out of the data set: does the model change?

If the model doesn't change that case is not very influential!

Holding out case i to check if its influential:

- $\mathbb{X}$ is the original data set from which we get $\mathbb{Z}$ which is an $n \times (r+1)$ matrix
- Then by taking out case i we get: $\mathbb{X}_{-i}$ from which we get $\mathbb{Z}_{-i}$ which is an $(n-1) \times (r+1)$ matrix

Now:

- From $\underline{y} = \mathbb{Z}\underline{\beta} + \underline{\epsilon}$ we get $\hat{\underline{\beta}} \in \mathbb{R}^{r+1}$
- From $\underline{y}_{-i} = \mathbb{Z}_{-i}\underline{\beta} + \underline{\epsilon}_{-i}$ we get: $\hat{\underline{\beta}}^{(i)} \in \mathbb{R}^{r+1}$

If $\hat{\underline{\beta}}^{(i)}$ and $\hat{\underline{\beta}}$ are very different then case i is influential!

To check how different they are we compute the following distance:

$$D_i = \frac{(\hat{\underline{\beta}}^{(i)} - \hat{\underline{\beta}})^T (\mathbb{Z}^T\mathbb{Z})(\hat{\underline{\beta}}^{(i)} - \hat{\underline{\beta}})}{S^2(r+1)}$$

How far are $\hat{\underline{\beta}}^{(i)}$ and $\hat{\underline{\beta}}$ in terms of the Mahalanobis Distance?

Note: D_i is called **Cook's Distance.**

We conclude case i is influential if D_i is large. How do we do it?

- We compare with the quantiles of $F(r+1, n-(r+1))$ to check whether or not $\hat{\underline{\beta}}$ is outside the confidence region centred in $\hat{\underline{\beta}}$

  If $\hat{\underline{\beta}}$ is outside then they are indeed very distant!

- We have a rule of thumb: we check the cases for which $D_i > 1$ as the median of $F(r+1, n-(r+1))$ is always close to 1 for reasonable values of r, n!

  If $D_i > 1$ then distance is bigger than what we would have expected!

Note: $D_i = \left(\frac{\hat{\epsilon}_i}{S\sqrt{1-h_{ii}}}\right)^2 \frac{h_{ii}}{1-h_{ii}} \frac{1}{r+1}$ so:

- D_i is a function of Studentised residuals, which is the first term!
- D_i is a monotone function of the Leverage h_{ii}, which is in the second term!

Collinearity

We have a regressor that is almost a linear combination of other regressors, so it's not capturing many new information!

So we are close to have the columns of the design matrix to be linearly dependent!

Note: we always assume $r+1 \leq n$

So in this case $\mathbb{Z}^T\mathbb{Z}$ is almost singular and so we are close to not be able to compute $\hat{\beta}$

Moreover $Cov(\underline{\hat{\beta}}) = \sigma^2(\mathbb{Z}^T\mathbb{Z})^{-1}$ is almost infinity if $\mathbb{Z}^T\mathbb{Z}$ is almost singular! We have huge uncertainty!

This is **dangerous**: since the variability is so big if a $\hat{\beta}_i$ we though was positive is actually negative!

Thus $\hat{\beta}_i$ has the opposite effect on the phenomenon we are trying to explain!

For example suppose that $\hat{\beta}_i > 0$ is saying that unemployment rate is correlated to inflation: so we make a policy to reduce inflation!

Well actually due to the huge variability $\hat{\beta}_i$ is actually negative and so no surprise if then the policy, by reducing inflation, makes an increase in the unemployment rate!

Working around this problem is hard!

Recall that $\underline{\hat{y}} = \pi_{\underline{y}|\mathcal{L}(\mathbb{Z})}$ Then we can use Gram-Schmidt to generate an orthogonal basis for $\mathcal{L}(\mathbb{Z})$:

- $\mathbb{Z} = [\underline{1} \;\; \underline{z}_1 \;\; \dots \;\; \underline{z}_r]$ where for instance $\underline{z}_1 = [z_{11}, \dots, z_{n1}]^T$
- Then we fix one vector: $\underline{q}_1 = \underline{1}$ so that:
  - $\underline{q}_2 = \underline{z}_1 - \pi_{\underline{z}_1|\underline{1}}$
  - $\underline{q}_3 = \underline{z}_2 - \pi_{\underline{z}_2|\underline{q}_1,\underline{q}_2}$
  - And so on and so forth: $\underline{q}_r = \underline{z}_r - \pi_{\underline{z}_r|\underline{z}_1,\dots,\underline{z}_{r-1}}$

Then $\underline{q}_r$ is the residual when we regress $\underline{z}_r$ on $[\underline{1} \;\; \underline{z}_1 \;\; \dots \;\; \underline{z}_{r-1}]$

But: $\hat{\beta}_r = \dfrac{\underline{q}_r^T\underline{y}}{\underline{q}_r^T\underline{q}_r}$ so: $Var(\hat{\beta}_r) = \dfrac{1}{(\underline{q}_r^T\underline{q}_r)^2}\sigma^2\underline{q}_r^T\underline{q}_r = \dfrac{\sigma^2}{\underline{q}_r^T\underline{q}_r}$

Now apply the decomposition of variance formula applied to the regression of z_r on $[\underline{1} \;\; \underline{z}_1 \;\; \dots \;\; \underline{z}_{r-1}]$ Then:

$$\sum_{i=1}^n (z_{ir} - \bar{z}_r)^2 = \sum_{i=1}^n (\hat{z}_{ir} - \bar{z}_r)^2 + \underline{q}_r^T\underline{q}_r \implies \underline{q}_r^T\underline{q}_r = \sum_{i=1}^n (z_{ir} - \bar{z}_r)^2 - \sum_{i=1}^n (\hat{z}_{ir} - \bar{z}_r)^2 = \sum_{i=1}^n (z_{ir} - \bar{z}_r)^2(1 - R_r^2)$$

where R_r^2 is the R^2 of the regression of z_r on $[\underline{1} \;\; \underline{z}_1 \;\; \dots \;\; \underline{z}_{r-1}]$

Thus:

$$Var(\hat{\beta}_r) = \frac{\sigma^2}{\sum_{i=1}^n (z_{ir} - \bar{z}_r)^2(1 - R_r^2)} = \frac{\sigma^2}{(n-1)S_r^2}\frac{1}{1 - R_r^2}$$

This formula is true for any regressor as we can change the order of every regressor, so in general:

$$Var(\hat{\beta}_j) = \frac{\sigma^2}{\sum_{i=1}^n (z_{ij} - \bar{z}_j)^2(1 - R_j^2)} = \frac{\sigma^2}{(n-1)S_j^2}\frac{1}{1 - R_j^2} \text{ for } j = 1, \dots, r$$

Then:

- $Var(\hat{\beta}_j)$ goes down if $\sum_{i=1}^n (z_{ij} - \bar{z}_j)^2$ goes up!

  So we take the regressor in the design matrix very spread out so that the variability of the estimator is decreased!

- Also: $Var(\hat{\beta}_j)$ goes up if R_j^2 goes up and for $R_j^2 = 1$ the variability is infinity! R_j^2 goes up if z_j could be expressed as a linear combination of the other regressors: collinearity!

Definition: $VIF(\hat{\beta}_j) = \dfrac{1}{1 - R_j^2}$ is the **Variance Inflation Factor**, as it inflates the variance of the coefficient, with respect to the situation in which all the regressors z_i are orthogonal!

We can compute VIF and if $VIF(\hat{\beta}_j) > 5$, or > 10, then watch out: we might take out that variable!

Model (or variable) Selection

In the above we have seen we might want to take out many variables from the model, in all the above for different reasons. How can we do it?

We have our training data and the design matrix: $\mathbb{Z} = [\underline{1} \quad \underline{z}_1 \quad \dots \quad \underline{z}_r]$ where $\underline{z}_j \in \mathbb{R}^n$ How many linear models can we build from this design matrix?

We have two possibilities for each variables so we have 2^r models: pretty big!

We have two lines of attack:

- **Greedy approach:** we check all the possibilities. Very computationally expensive!
  For $k = 0, \dots, r$ repeat:
  - We fit the $\binom{r}{k}$ models with k regressors
  - We compute R^2 and we choose among these the model with k regressors with the highest R^2 Let this be: R_k^2

  End for.

  Now we get: $R_0^2, \dots, R_r^2$ with: $R_0^2 \leq R_1^2 \leq \dots \leq R_r^2$

  We plot R_k^2 or $R_{adj,k}^2$ or AIC, or BIC, and then we find an elbow and choose the best k!

  Note: to use AIC, since it's based on maximum-likelihood, then we need to have Gaussian data, since AIC by default uses the likelihood computed from Gaussian!

  This model fails as soon as r is big as it computationally expensive!

- **Step-wise iterative approach:** we look for a good model and not for the optimal one! There are two ways: **forward** or **backward**.

  Suppose we go **forward**, then:
  We start with the best model with 1 regressor, then repeat until convergence:
  - Add one variable which increases the fit of the model

  When is convergence? We use the F-test to decide when to stop! Suppose that $k' > k$ then:

$$\frac{(SS_{res}(\mathbb{Z}_k) - SS_{res}(\mathbb{Z}_{k'}))}{k' - k} \frac{1}{\dfrac{SS_{res}(\mathbb{Z}_{k'})}{n - (k' + 1)}}$$

  We stop as soon as the $p-$value of the test is large: it doesn't pay to move to a larger model!

  Note that with **forward** we don't have the collinearity problem!

Backward selection we go the opposite way: we start with the completed model and at each step we take out variables!

Note: there are plenty of variations: we may add two variables and remove 1, or add 3 variables and so on.

Note: we can do this iterative approach only if the models are nested!

There are more modern procedure that tells you why you should take out some specific variable: PCA regression, Ridge and Lasso regression.

23 Lecture 35: 14th Of May 2020

Note: $\underline{y} = \mathbb{Z}\underline{\beta} + \underline{\epsilon}$ is the model for the observed data from OLS we get: $\hat{\underline{\beta}} = (\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\underline{y}$ assuming $\mathbb{Z}$ has full rank!

Then the fitted model for the mean of $\underline{y}$ is: $y_0 = \underline{z}_0^T\hat{\underline{\beta}}$

The fitted model always go through the baricentre of the observed data: for instance suppose $\underline{z}_0 = \dfrac{\mathbb{Z}^T\underline{1}}{\underline{1}^T\underline{1}} = [1 \;\; \bar{z}_1 \;\; \dots \;\; \bar{z}_r]^T$ so it's the vectors of the mean of the regressors!

Then: $y_0 = \underline{z}_0^T\hat{\underline{\beta}} = \dfrac{\underline{1}^T\mathbb{Z}}{n}(\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\underline{y} = \frac{1}{n}\underline{1}^T H\underline{y} = \frac{1}{n}(H\underline{1})^T\underline{y} = \frac{1}{n}\underline{1}^T\underline{y} = \bar{y}$ Hence: $\bar{y} = \hat{\beta}_0 + \hat{\beta}_1\bar{z}_1 + \cdots + \hat{\beta}_r\bar{z}_r \implies \hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{z}_1 - \cdots - \hat{\beta}_r\bar{z}_r$ So a different representation for the fitted model is:

$$y_0 - \bar{y} = \hat{\beta}_1(z_1 - \bar{z}_1) + \cdots + \hat{\beta}_r(z_r - \bar{z}_r)$$

So instead of working with the original variables, we can first centre the data on the baricentre: $z - \bar{z}$ and $y - \bar{y}$ and then fit the model!

So **centering:** from $\underline{y}$ we consider $\mathbb{R}^n \ni \underline{y}^* = [y_1 - \bar{y}, \dots, y_n - \bar{y}]^T$ and from $\mathbb{Z}$ we consider: $\begin{bmatrix} z_{11} - \bar{z}_1 & \dots & z_{1r} - \bar{z}_r \\ \vdots & & \\ z_{n1} - \bar{z}_1 & \dots & z_{nr} - \bar{z}_r \end{bmatrix} = \mathbb{Z}^*$

which is an $n \times r$ matrix!

So the OLS problem, with this new notation, becomes:

$$\arg\min_{\underline{\beta} \in \mathbb{R}^r} \left\| \underline{y}^* - \mathbb{Z}^*\underline{\beta} \right\|^2 = \hat{\underline{\beta}}^*$$

with $\hat{\beta}_0 == \bar{y} - \hat{\beta}_1\bar{z}_1 - \cdots - \hat{\beta}_r\bar{z}_r$ and with: $\hat{\beta}_i = \hat{\beta}_i^*$

So now since we already have $\hat{\beta}_0$ then we have one less dimension in this problem!

From now on we assume the variable have been centred and we drop the $*$ notation as a superscript!

Thus then to go back to the original system, after getting estimates of $\hat{\underline{\beta}}$, we fix $\hat{\beta}_0$ so that we can go back to the original variables!

Note that we fix $\hat{\beta}_0$ so that the model goes through the baricentre!

We use this assumptions in the following!

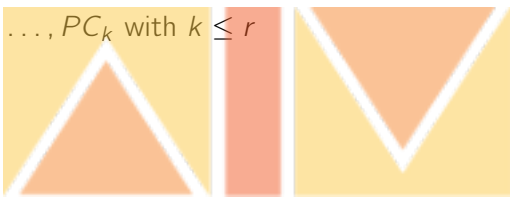
Collinearity and variable selection in linear models: 1) PCA Regression, 2) Ridge Regression & 3) Lasso

1) PCA Regression:

The problem of collinearity is due to the fact that the regressors aren't orthogonal, thus we can transform them so that we get new orthogonal regressors: we just do PCA of the dependent variables!

$\mathbb{Z} = [\underline{z}_1, \dots, \underline{z}_r]$ with $\underline{z}_i \in \mathbb{R}^n$ (Remember: these are already centred variables!) Then we perform PCA of $\mathbb{Z}$ so that we get: $PC_1, \dots, PC_r$

We can then check for an elbow and reduce the dimensionality: we consider $PC_1, \dots, PC_k$ with $k \leq r$



Now we get: $\mathbb{Z}^* = [\underline{PC}_1, \ldots, \underline{PC}_k]$ and we then fit: $\underline{y} = \mathbb{Z}^*\underline{\beta} + \underline{\epsilon}$ where $\underline{\beta} \in \mathbb{R}^k$ since we reduced the dimensionality!

The fitted model is given by: $y_0 = \underline{z}_0^T\hat{\underline{\beta}} = \hat{\beta}_1 PC_1 + \cdots + \hat{\beta}_k PC_k$ where: $\underline{z}_0^T = (PC_1, \ldots, PC_k)^T$

Now:

- $PC_1 = e_{11}z_1 + \cdots + e_{r1}z_r$ since it's a linear projection of the original variables
- And so on and so forth, until: $PC_k = e_{1k}z_1 + \cdots + e_{rk}z_r$

Substituting this into the fitted model:

$$y_0 = z_1(e_{11}\hat{\beta}_1 + \cdots + e_{1k}\hat{\beta}_k) + z_2(e_{21}\hat{\beta}_1 + \cdots + e_{2k}\hat{\beta}_k) + \cdots + z_r(e_{r1}\hat{\beta}_1 + \cdots + e_{rk}\hat{\beta}_k) = z_1\hat{\gamma}_1 + \cdots + z_r\hat{\gamma}_r$$

We solved the problem of collinearity since the principal components are orthogonal, but there is no guarantee that the principal component are good directions for prediction: we might have thrown away the information of $\mathbb{Z}$ correlated with y

Note: there are algorithms that do dimension reduction without loosing too much correlation (e.g: slice-inverse regression).

We still have r variables: the solution is not sparse in terms of $z_1, \ldots, z_r$ But how can we do variable selection? we see in the following with ridge and lasso regression!

Regularisation: 2) Ridge Regression (Hoerl & Kernard)

Collinearity might make the variance of the $\hat{\beta}_i$ explode: let's try to keep the variability under-control.

The optimisation problem for **Ridge Regression** is the following:

$$\arg\min_{\underline{\beta}} \left\|\underline{y} - \mathbb{Z}\underline{\beta}\right\|^2 \text{ and } \left\|\underline{\beta}\right\|^2 \leq s$$

we are constraining $\underline{\beta}$ to be small in norm, so that it cannot have high variability!

Note that we can re-state the problem in an equivalent way:

$$\left\|\underline{y} - \mathbb{Z}\underline{\beta}\right\|^2 = \left\|\underline{y} - \bar{\underline{y}} - \mathbb{Z}(\underline{\beta} - \hat{\underline{\beta}})\right\|^2 \text{ with } \hat{\underline{y}} = H\underline{y} \text{ and } \hat{\underline{\beta}} = (\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\underline{y}$$

This can then be re-written as:

$$\left\|\underline{y} - \mathbb{Z}\underline{\beta}\right\|^2 = \left\|\hat{\underline{\epsilon}} - \mathbb{Z}(\underline{\beta} - \hat{\underline{\beta}})\right\|^2 = \left\|\hat{\underline{\epsilon}} - \mathbb{Z}(\underline{\beta} - \hat{\underline{\beta}})\right\|^2 = \|\hat{\underline{\epsilon}}\|^2 + \left\|\mathbb{Z}(\underline{\beta} - \hat{\underline{\beta}})\right\|^2$$

since $\mathbb{Z}(\underline{\beta} - \hat{\underline{\beta}}) \in \mathcal{L}(\mathbb{Z})$ and $\hat{\underline{\epsilon}} \in \mathcal{L}^{\perp}(\mathbb{Z})$

Hence:

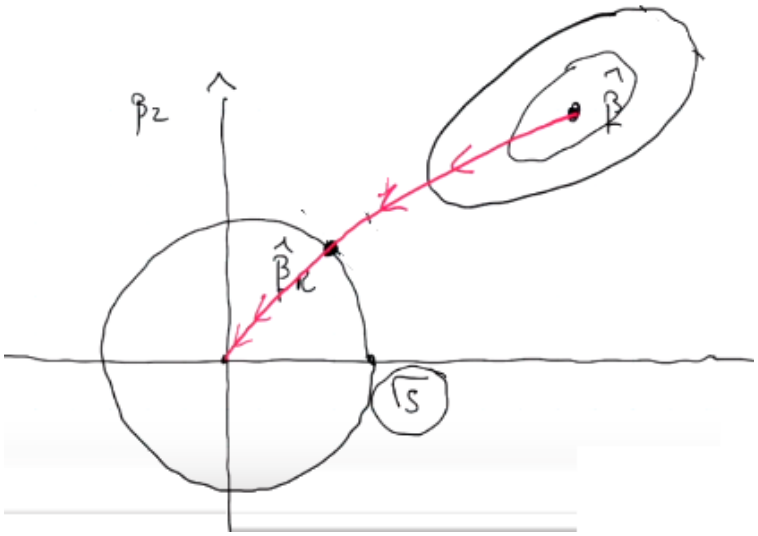
$$\arg\min_{\underline{\beta}} \|\hat{\underline{\epsilon}}\|^2 + \left\|\mathbb{Z}(\underline{\beta} - \hat{\underline{\beta}})\right\|^2 = \|\hat{\underline{\epsilon}}\|^2 + \arg\min_{\underline{\beta}} \left\|\mathbb{Z}(\underline{\beta} - \hat{\underline{\beta}})\right\|^2 \text{ with } \left\|\underline{\beta}\right\|^2 \leq s$$

since $\hat{\underline{\epsilon}}$ doesn't depend on $\underline{\beta}$

Remember that: $\hat{\underline{\beta}} = (\mathbb{Z}^T\mathbb{Z})^{-1}\mathbb{Z}^T\underline{y}$ so that the above becomes:

$$\arg\min_{\underline{\beta}} (\underline{\beta} - \hat{\underline{\beta}})^T\mathbb{Z}^T\mathbb{Z}(\underline{\beta} - \hat{\underline{\beta}}) \text{ with } \underline{\beta}^T\underline{\beta} \leq s$$

So in the $\underline{\beta}$ space we have the following geometry:



We want a solution inside the spherical neighbourhood of radius $\sqrt{s}$, that is a solution of the OLS so that $\underline{\hat{\beta}}$ belong to the contour of the objective function which are ellipsoidal!

The solution is where the contour are tangent to the spherical neighbourhood (i.e: constraints) which is $\underline{\hat{\beta}}_R$

Note: The larger s the larger the constraint, so the original OLS solution may satisfy the new problem.

As s decreases the ridge regression shrinks the solution towards zero: at the end $s = 0$ so that: $\underline{\hat{\beta}}_R = 0$ no regression!

Indeed that's why $\underline{\hat{\beta}}_R$ is called shrinkage estimator!

Note that when we write $\underline{\hat{\beta}}$ we mean: $\underline{\hat{\beta}}_{OLS}$ the solution obtained from OLS!

The solution of the above optimisation problem is obtained by using the Lagrange methods which implies to solve:

$$\arg\min_{\underline{\beta}} \left\|\underline{y} - \mathbb{Z}\underline{\beta}\right\|^2 + \lambda\left\|\underline{\beta}\right\|^2$$

So we want a solution for the OLS problem but we penalise large solutions (large in terms of square norm)!

Where λ is a function of s and is the Lagrange multiplier of the Lagrangian function!

Note: since s is free also λ is free!

Note moreover that:

- For λ very large we penalise a lot solutions too far from zero, so as $s \to 0$ then $\lambda \to \infty$ and vice-versa.
- If we take $\lambda = 0$ we have the OLS solution!

The above problem is solvable by differentiating with respect to $\underline{\beta}$ so that:

$$\underline{\hat{\beta}}_R = (\mathbb{Z}^T\mathbb{Z} + \lambda I)^{-1}\mathbb{Z}^T\underline{y}$$

We see how the solution solves the collinearity problem: we sum a diagonal matrix with different value so that we have the matrix is always invertible!

Note that:

- $\underline{\hat{\beta}}_R$ is a biased estimator of $\underline{\beta}$
- It has a lower mean squared error: There always exists a proper choice of λ such that we have:

$$\mathbb{E}\left[\left\|\underline{\hat{\beta}}_R - \underline{\beta}\right\|^2\right] < \mathbb{E}\left[\left\|\underline{\hat{\beta}}_{OLS} - \underline{\beta}\right\|^2\right]$$

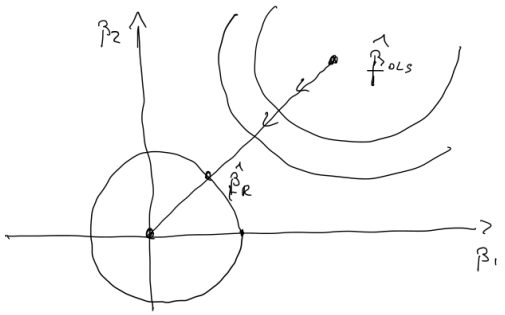
  We have higher bias but lower variance!

If λ too small we are not doing any regression as the solution goes to the origin, and if λ is too big we are going to the OLS solution!

So how to choose λ? Cross-validation: we find the best λ which minimises the prediction error for y

Unlike the OLS the estimator $\underline{\hat{\beta}}_R$ is not scale invariant so before using it we need to standardise both the regressor and the dependent variables before fitting the Ridge Regression!

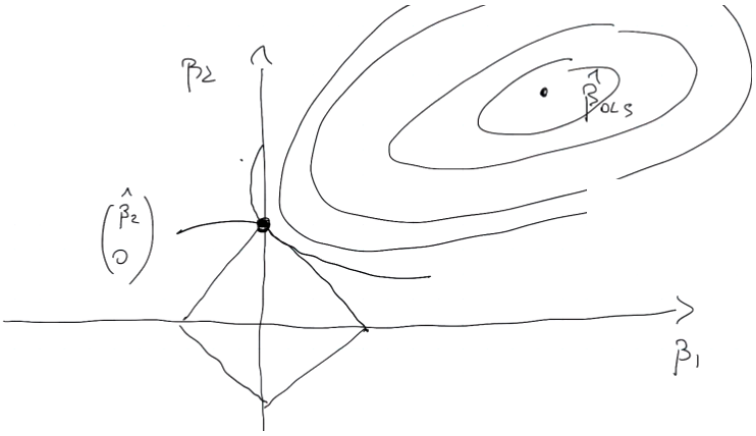
Note that if we standardise the regressors then they might be orthogonal so that the above figure describing the geometry of the ridge becomes:



So that $\hat{\beta}_R$ is just $\hat{\beta}_{OLS}$ re-scaled.

Regularisation: 3) Lasso Regression (Tibshirani)

It's similar to ridge but here we change the constraint so that we have the situation represented below:



So instead of spherical constraints we take a diamond constraint: so that we can hope that the tangent point with the contour point will be on one of the vertices so that one of the $\hat{\beta}_i$ will be exactly zero!

Not only we shrink the $\underline{\hat{\beta}}$ but we also select the variables: feature selection!

This is not possible with the ridge regression!

This corresponds to solving the following optimisation problem:

$$\arg\min_{\underline{\beta}}(\underline{\beta}-\hat{\underline{\beta}})^T\mathbb{Z}^T\mathbb{Z}(\underline{\beta}-\hat{\underline{\beta}}) \text{ with } \left\|\underline{\beta}\right\|_1=\sum_{i=1}^{r}|\beta_i|\leq s$$

So the constraint instead of using the L^2 norm of $\underline{\beta}$ uses the L^1 norm!

The price that we pay is that we have no analytical solution for this problem!

This is a biased estimator so we have higher bias but less variance: again it's bias-variance trade-off!

The above problem can be written with the Lagrangian as:

$$\arg\min_{\underline{\beta}}\left\|\underline{y}-\mathbb{Z}\underline{\beta}\right\|^2+\lambda\left\|\underline{\beta}\right\|_1$$

We find the optimal λ through cross-validation on the prediction error for y!

What if we take an even more spiky constraint? We can take a $p<1$ norm! They are all variations of Lasso:

$$\arg\min_{\underline{\beta}}\left\|\underline{y}-\mathbb{Z}\underline{\beta}\right\|^2 \text{ with } \left\|\underline{\beta}\right\|_q\leq s \text{ where } \left\|\underline{\beta}\right\|_q=\left(\sum_i|\beta_i|^q\right)^{1/q}$$

If $q=2$ we have Ridge, If $q=1$ we have Lasso. The problem is that for $q<1$ the problem is no more convex!

Note that has $q\to 0$ the above it's a pure model selection problem: which variables should we consider?

Anyway with Lasso and other $q<1$ methods, anyway take care of collinearity and of the model selection problem!

We can take both Lasso and Ridge at the same time: we move to machine learning! No properties of the estimator, no uncertainty measure, just prediction:

$$\arg\min_{\underline{\beta}}\left\|\underline{y}-\mathbb{Z}\underline{\beta}\right\|+\lambda_1\left\|\underline{\beta}\right\|_1+\lambda_2\left\|\underline{\beta}\right\|_2^2$$

This are called **Elastic Nets:** the overall idea is that we fit a regression plus a penalisation!

This works well if we have background knowledge on the type of solution we might want to get in an ideal setting! So we penalise solutions too far from what ideally we expect to get!

Connections with GLM (Generalised Linear Models)

The basic model is: $y=f(\underline{z})+\epsilon$ Then:

- In Linear regression we assume: $f(\underline{z})=\mathbb{E}[y|\underline{z}]$ and we set: $f(\underline{z})=\beta_0+\beta_1z_1+\cdots+\beta_rz_r$

  In linear regression we assume: $\mathbb{E}[\epsilon]=0$ and $Var[\epsilon]=\sigma^2 \implies \mathbb{E}[y|\underline{z}]=f(\underline{z})$ and $Var[y|\underline{z}]=\sigma^2$

  Sometimes we even assume $\epsilon\sim\mathcal{N}(0,\sigma^2) \implies y|\underline{z}\sim\mathcal{N}(f(\underline{z}),\sigma^2)$

- In generalised linear models we set: $g(\mathbb{E}[y|\underline{z}])=\beta_0+\beta_1z_1+\cdots+\beta_rz_r$ where g is a particular transformation.

In GLM we assume that: $y|\underline{z} \sim F$ where F can be a Poisson, Bernoulli, Gaussian, and so on!

What do we pay? Least Squares doesn't work any longer so we need to use maximum likelihood to estimate the parameters $\underline{\beta}$!

We need to optimise the likelihood which can be very complex, so the optimisation is usually done numerically!

The function g is called *link-function:*

- If Y is Gaussian then: $g(\mu) = \mu$ so g is the identity. This is *Linear Regression!*
- If Y is $Be(p)$ then: $g(p) = logit(p) = \log\left(\frac{p}{1-p}\right)$. This is *Logit (Logistic) Regression!*
- If Y is $Poisson(\lambda)$ then: $g(\lambda) = \log(\lambda)$. This is *Poisson Regression!*
- If Y is $Be(p)$ then: $g(p) = \Phi^{-1}(p)$ where Φ is the cumulative distribution of $N(0, 1)$. This is *Probit Regression!*

Everything we said about Ridge and Lasso can be applied here!

Since we use maximum likelihood and not least squares we don't have any variance nor properties for the estimators!

24 Lecture 37: 18th Of May 2020

Permutation Test for inference: non-parametric approach! We tackle null-hypothesis tests in model with no Gaussian Assumption!

Permutation Test: remove Gaussianity assumption and sample size (avoid using CLT)!

- If $1 = p < n = \infty$ thanks to the CLT Gaussianity is not a key point.
- If $1 = p < n \leq \infty$ with the $t-$distribution is meant to model situations in which the sample size is not very large.

  The Gaussianity of data is required for the $t-$test: uni-variate Gaussianity is not difficult to asses anyhow!
- If $1 \leq p < n \leq \infty$ with Hotelling's T^2 relies on multi-variate Gaussian data. If p increases due to the curse of dimensionality the multi-variate Gaussianity of data is hard to assess.
- If $1 \leq n < p \leq \infty$ high dimensional test rely on the multi-variate Gaussianity of data and they are not robust with respect to the violation of Gaussianity!

  Powerful Gaussianity test are not available in the high dimensional setting!
- If $1 \leq n < p = \infty$ in the functional case, normality is basically an unverifiable assumption.

Indeed all parametric tests are only exact either asymptotically or under Gaussian assumption, so we need something different:
Permutation test: do test when we have high dimensional functional data and remove Gaussian assumption!

Fishers' Argument for permutation test: two population $t-$test

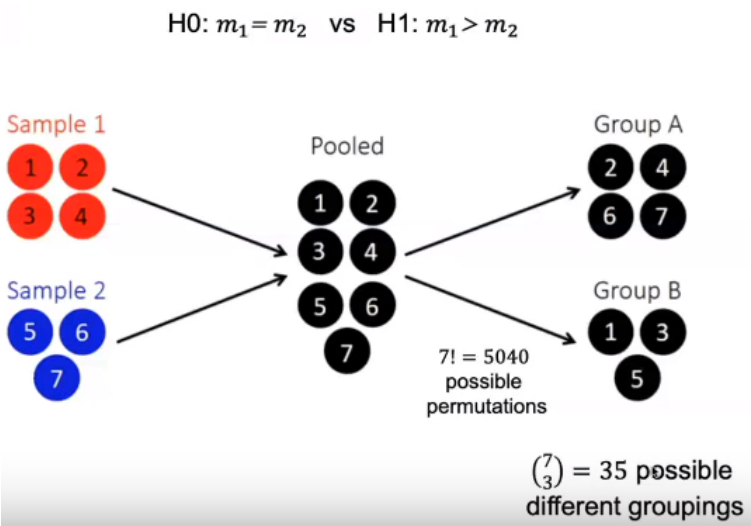
> *Let us suppose, for example, that we have measurements of the stature of* ***a hundred Englishmen and a hundred Frenchmen****. It may be that* ***the first group are, on the average, an inch taller than the second****, although the two sets of heights will overlap widely. [...] The simplest way of understanding quite rigorously, yet without mathematics, what the calculations of the test of significance amount to, is to consider what would happen* ***if our two hundred actual measurements were written on cards, shuffled without regard to nationality, and divided at random into two new groups of a hundred each****. This division could be done in an enormous number of ways, but though the number is enormous it is a finite and a calculable number. We may suppose that* ***for each of these ways the difference between the two average statures is calculated****. Sometimes it will be less than an inch, sometimes greater.* ***If it is very seldom greater than an inch, in only one hundredth, for example****, of the ways in which the sub-division can possibly be made, the statistician will have been right in saying that* ***the samples differed significantly****. For* ***if,*** *in fact****, the two populations were homogeneous, there would be nothing to distinguish the particular subdivision*** *in which the Frenchmen are separated from the Englishmen* ***from among the aggregate of the other possible separations*** *which might have been made. Actually, the statistician does not carry out this very simple and very tedious process, but his conclusions have no justification beyond the fact that they agree with those which could have been arrived at by this elementary method.*
>
> Fisher, R. A. (1936). The coefficient of racial likeness and the future of craniometry, *Journal of the Anthropological Institute of Great Britain and Ireland*, pp. 57-63.

There are $\binom{200}{100}$ possible groups of 100 people starting from 200 people

$\frac{1}{100}$ is the $p-$value in the permutation test frame work: indeed $p-$value is a measure of centrality of the sample!

Instead of comparing the two samples with respect to all the infinite possible sample drawn from the two population, we compare the two samples with all those samples, that under the same null hypothesis, will provide the same likelihood!

Consider the following setting:



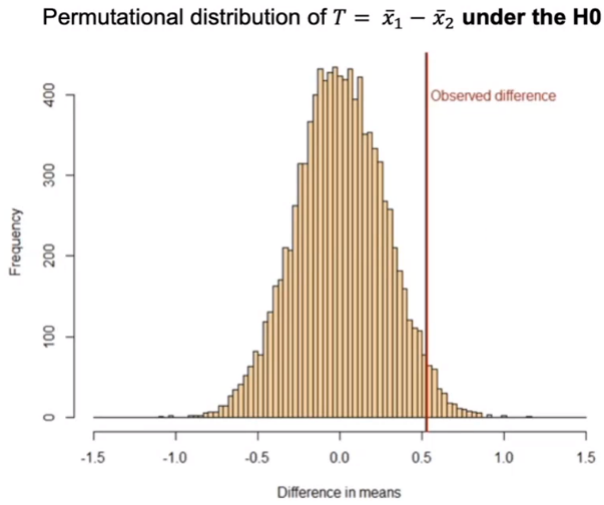
We have two samples and we build the pooled set: we don't care about the order! Then we divide it into two groups having same sample sizes as the original one!

There are $\binom{7}{3} = 35$ possible different groupings and $7! = 5040$ possible permutations. The idea is to compare the two samples we have at the start with the 35 we can get from permuting!

In permutation test we sample from the pooled set instead of from a Gaussian distribution since we have no idea of the distribution of the data and thus this permutation sampling is the only thing which makes sense for any possible distribution!

As test statistics we use T which is the difference between the two sample mean given the pooled sample!

We build an histogram on the 35 different possible value of the possible grouping of the test statistics T:



Note that this is a proper distribution: it's the distribution of the test statistics, under H_0, conditional on the pooled set! It's also called permutational distribution!

This is a discrete distribution with 35 possible values all with the same probability: $\frac{1}{35}$, indeed the seven random variables are independent and identically distributed under the null hypothesis!

So in permutation test the distribution of the permutation statistics is a uniform discrete distribution!

Under H_1 we still have a conditional discrete distribution with a support of 35, but we don't have a uniform distribution but different probabilities for each one of the 35 possible values.

Indeed under H_1 we have that larger values of $\bar{x}_1 - \bar{x}_2$ are more probable.

We define a $p-$value as: $p = \frac{1}{35}\sum_{k=1}^{35} \mathbb{1}(T_k^* > T_0)$ which is the fraction of the number of samples that lead to larger values of $\bar{x}_1 - \bar{x}_2$

Can we extend the inference on the overall population or is limited to the pooled set? Yes the inference is unconditional even if the probabilities are conditional!

If the above rejects the null hypothesis with probability α for every pooled set then it reject the null hypothesis, with probability α, whatever the pooled set is!

Corner-stones of permutational inference:

- We can make fewer assumptions as possible on the data distribution.
- How they work:
  - We need to identify the proper conditional distribution for our problem: we make likelihood invariant transformation of the sample under H_0 but not under H_1, so not only permutation!

    This guarantees that the conditional distribution is uniform discrete under H_0 but not H_1
  - We need to identify a good test-statistics: no idea of optimality since we have no underlying distribution hypothesis of the data!

    This test-statistics has to be sensitive to the violation of the null hypothesis, thus we can test same hypothesis with different permutation statistics, but then we should use $p-$value adjustments (e.g: Bonferroni's)

    This allows us to work in purely metric spaces: we can build test on complex functional data
- Inferential properties:
  - Permutation test are exact also for finite sample (differently from bootstrap test)
  - Consistency: power (error type I) goes to 1 if sample size goes to ∞ This is true if the test statistics is properly chosen!
  - Asymptotic equivalence to parametric test: doing permutation test with statistics taken from parametric world and all the assumption of the parametric world hold then the results are the same only asymptotically!

    This is used in two ways:
    * If a parametric test statistics is optimal we can borrow it for permutation test
    * Parametric tests can be seen as computational shortcut, of non-parametric, when the sample size is big and the assumptions hold!
- Permutational test cost explodes as the sample increase!

  So in practice we don't explore all possible permutation but use Conditional Monte-Carlo to randomly sample

permutation and then fix the p-value.

We then build confidence interval for $p-$value

Likelihood invariant transformations:

- If we have two-population test or one-way ANOVA we permute the value, which is equivalent to group labels permutations!
- If we have one-population test and paired two-population test: in general case there is only the identity transformation!

  If we assume symmetry of the distribution, which is a weaker assumption than the Gaussianity one, we can re-center in H_0 and swap signs.

  Here the number of transformations is 2^n where n is sample size;
- If we have independence test, we can use pair re-coupling: we dismount the pairs and obtain two new samples
- If we have linear model: Linear regression and multi-way ANOVA, we can make F-like tests.

  We use response permutations: we randomly permute the y (responses) over the sample units.

  We can make T-like tests: assess one factor or one single regressor.

  We use permutations of residual of restricted model: here we have no exact result but only asymptotic!

  Indeed we can't permute the errors, since we can't observe them, but only the residuals which are only estimates!

  It can be proven that this test, if the data aren't Gaussian, converges to the asymptotic distribution much faster than the parametric test.

25 Lecture 39: 21st Of May 2020

Spatial Statistics

Every datum is always geo-referenced: time and location stamp! Spatial statistics: domain where we look at datum can be $n-$dimensional space very abstract!

Basic idea: Two datum that are close in this space are more correlated! Close with respect to what? It also depends on the notion of distance we consider!

In general in spatial statistics we assume we have some observations referred to some spatial locations. We have three possible types of data, namely:

- **Geo-statistics data:** we will focus on this.
  We have a domain $D \subseteq \mathbb{R}^d$ in which we can identify points $s_i \in D$ and in each point we have some observations z_{s_i} (e.g: pollution).

  So we have geographical locations $s_1, \dots, s_n \in D \subseteq \mathbb{R}^d$ which we assume to be fixed and not random!

  At these locations we have observations $z_{s_1}, \dots, z_{s_n}$ random objects (e.g: variables, vectors).

  Our goal is:

  - To make models: how does the pollution vary in Milan?
  - Study spatial dependence and define models for this
  - Make prediction: we have some new location s_0 Can we make predictions in s_0?

    We will introduce the Kriging method for this!

- **Lattice (areal) data:** we have a domain D divided in small sub-domains (e.g: grid data or regional data).

  So we have grid locations $s_1, \dots, s_n$ which cover the entire domain (e.g: partition) and in s_i, a sub-domain, we have an observation z_{s_i}

  Now the difference is that s_i is not a point but a sub-region: typically we don't want to make prediction since we have observations all over D, but rather we want to model, or cluster!

  Note that s_i are still fixed!

  Typical models are Markovian Random Fields!

- **Point processes (or patterns):** our locations s_i are now random! So we have a random process realised on some locations $\{s_i, \dots, s_n\}$

  Examples are murders by a serial killer or earthquakes locations!

  We want to understand if there is a clustering or a bigger presence of locations in some areas with respect to other areas!

  The above is the standard point process! But we can also have the so called Marked Point Process in which we also have an observations for each random location!

  So we have the couple (s_i, z_{s_i}) for $i = 1, \dots, n$ where s_i is a random site and z_{s_i} is an observation of some feature in s_i

Examples are: s_i epicentre of earthquake and z_{s_i} is the magnitude!

Geo-statistics

Why should we care about spatial dependence when we have spatial data? Suppose we have our sampling sites $s_1, \dots, s_n \in D$ and real random variables z_{s_i} for $i = 1, \dots, n$

Suppose z_{s_i} comes from a linear model, so that: $z_{s_i} = \sum_{\ell=0}^{L} a_\ell f_\ell(s_i) + \delta_{s_i}$ with $s_i \in D$ where:

- a_ℓ are the regression coefficients
- $f_\ell(s_i)$ are the regressors
- δ_{s_i} are the residuals, which corresponds to the the ϵ_i errors of linear model!

If we want to estimate the regression coefficients how can we do it?
$\mathbb{E}[\delta_{s_i}] = 0$ but since they are spatially distributed we have that $Cov(\delta_{s_i}, \delta_{s_j}) \neq 0$ since we suppose the proximity among different locations induce a spatial dependence in the observations!

If $\underline{\delta} = (\delta_{s_1}, \dots, \delta_{s_n})$ is the vector of residuals then $Cov(\underline{\delta}) = \Sigma$ not proportional to an identity matrix!

If we use OLS and forget about spatial dependence:

$$\hat{a}_{OLS} = (\mathbb{F}^T\mathbb{F})^{-1}\mathbb{F}^T\underline{z}$$

where: $\mathbb{F}$ is the design matrix and $\underline{z}$ is the vector of observations! Then:

$$Cov(\hat{a}_{OLS}) = (\mathbb{F}^T\mathbb{F})^{-1}(\mathbb{F}^T\Sigma\mathbb{F})(\mathbb{F}^T\mathbb{F})^{-1}$$

Note that in the previous classes we studied linear models in the Independent and Identically Distributed case in which we had $\Sigma = \sigma^2 I$

If we use GLS then:

$$\hat{a}_{GLS} = (\mathbb{F}^T\Sigma^{-1}\mathbb{F})^{-1}\mathbb{F}^T\Sigma^{-1}\underline{z} \implies Cov(\hat{a}_{GLS}) = (\mathbb{F}^T\Sigma^{-1}\mathbb{F})^{-1}$$

Note: $Cov(\hat{a}_{OLS}) - Cov(\hat{a}_{GLS}) \geq 0$ that is: $Cov(\hat{a}_{OLS}) - Cov(\hat{a}_{GLS})$ is a positive semi-definite matrix thus $\hat{a}_{OLS}$ has a higher variance!

So we need to care about spatial dependence: all the optimality conditions of the independent case are no longer valid!

Spatial Dependence

In order to take advantage of spatial dependence we need to measure spatial dependence!

Suppose that we have $s_1, \dots, s_n$ sites and $z_{s_1}, \dots, z_{s_n}$ observations, then we assume that we have a random process $\{z_s, s \in D\}$, from which these observations come from!

Note: s varies with continuity in D!

Trying to model the spatial dependence among locations means trying to build a model for the spatial dependence of the random field $\{z_s, s \in D\}$

We assume that:

- $\mathbb{E}[z_s] < \infty \; \forall s \in D$
- $Var[z_s] < \infty \; \forall s \in D$

This is not enough to estimate the spatial dependence and make predictions!

Definition:

- The *Spatial Mean* of the process $\{z_s, s \in D\}$ is given by $m_s = \mathbb{E}[z_s]$ for any $s \in D$
- The *co-variance function* of the process $\{z_s, s \in D\}$ is given by $c(s_1, s_2) = Cov(z_{s_1}, z_{s_2})$

We want to estimate the spatial mean and the co-variance function from the data as in almost all cases we don't know them!

To do so we need some stationarity assumption:
Definition: $\{z_s, s \in D\}$ is *second-order stationary* if:

- $\mathbb{E}[z_s] = m \; \forall s \in D$ is constant: it doesn't depend on s
- $Cov(z_{s_1}, z_{s_2}) = c(s_1 - s_2) \; \forall s_1, s_2 \in D$ where c is called **co-variogram**, which is a real valued function!

  Without this we only have repetition in space and not independent repetition and so we wouldn't be able to estimate the co-variance from the data.

  The co-variance in different directions is different!

Note: If D has more than one dimension we don't write $\underline{s}$ but just s for simplicity!

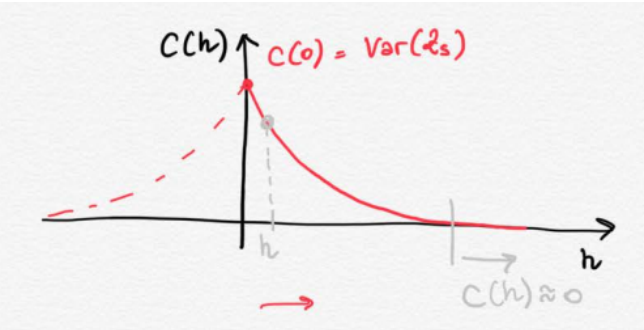
Properties of the co-variogram

- Since the co-variance is symmetric then c is symmetric and so: $c(-h) = c(h) \; \forall h \in \mathbb{R}^d$
- It's bounded: $|c(h)| \leq c(\underline{0}) = c(s - s) = Cov(z_s, z_s) = Var[z_s] \; \forall h \in \mathbb{R}^d$

  Note that this is just *Cauchy-Schwarz Inequality*
- It's positive definite, namely: $\sum_{i,j} \lambda_i \lambda_j c(s_i - s_j) \geq 0 \; \forall \lambda_i, \lambda_j \in \mathbb{R}, \; \forall s_i, s_j \in D$

  This means that if we use c for building co-variance matrices then the resulting matrices will be positive definite!

Example: Suppose $D \subset \mathbb{R}$ so it's an interval. A typical *co-variogram*, according to the properties above, is the following:



So at certain point $C(h)$ goes to zero as $h \to \infty$!

Definition: Suppose we have second order stationarity. Then the **variogram** is defined as:

$$2\gamma(s_1 - s_2) = Var(z_{s_1} - z_{s_2})$$

So the *variogram* defines the variance of an increment of the process as a function of the increment among locations!

Under stationarity the mean is constant and so:

$$2\gamma(s_1 - s_2) = \mathbb{E}[(z_{s_1} - z_{s_2})^2] - (m_{s_1} - m_{s_2})^2 = \mathbb{E}[(z_{s_1} - z_{s_2})^2]$$

Since $(m_{s_1} - m_{s_2})^2 = 0$!

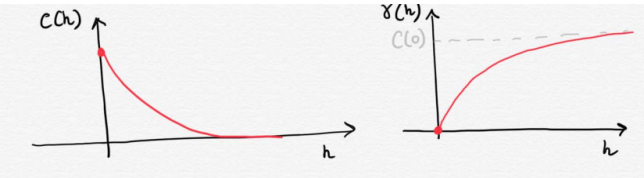
Note:

$$2\gamma(s_1 - s_2) = Var[z_{s_1}] + Var[z_{s_2}] - 2Cov(z_{s_1}, z_{s_2}) = c(0) + c(0) - 2c(s_1 - s_2)$$

Thus: $\gamma(s_1 - s_2) = c(0) - c(s_1 - s_2)$ where γ is the *semi-variogram!*

So if we know the *variogram* we know the *co-variogram* and vice-versa!

In one dimension we have the following graphs for the *co-variogram* and for the *semi-variogram:*



Note that if $h \to \infty$ then: $\gamma \to c(0)$

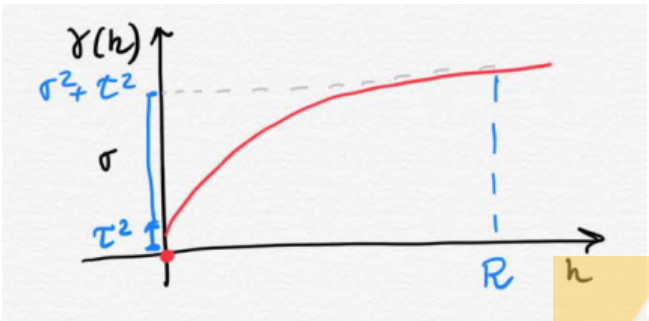
Algebraic Properties of the vario-gram:

- It's symmetric: $\gamma(-h) = \gamma(h)$
- It's null in the origin: $\gamma(0) = 0$
- We have the **Conditional Negative Definiteness:**

  $$\sum_{i,j} \lambda_i \lambda_j \gamma(s_i - s_j) \leq 0 \; \forall \lambda_i, \lambda_j \in \mathbb{R} : \; \sum_i \lambda_i = 0 \;\; \forall s_i, s_j \in D$$

  Note that since $\sum_i \lambda_i = 0$ then not all the co-variance matrices of the process can be generated by the vario-gram but only those that come from increments of the process!

- **We have some Structural Properties:**
  Note that the variogram can have a discontinuity in zero, τ^2, called **nugget effect** and thus the realisations of our process can be discontinuous:



So if we expect our process to be continuous and we see a nugget effect in it's vario-gram it means that we have a measurement error that is acting on top of the process!

Note that:

- $\sigma^2 + \tau^2$ is the horizontal asymptote of the semi-variogram and it's called **sill**: it's the variance of the process.
- σ^2 is called **partial sill**
- R is the **range**: it's the value for which the variogram reaches the *sill*!

  So the range R is quantifying the amount of dependence we have among our data (i.e: of the field)!

  The larger R the larger the dependence between the elements of the process!

- For convenience we assume **Isotropy:**
  A second order stationary field is **isotropic** if $Cov(z_{s_i}, z_{s_j}) = c(\|s_i - s_j\|)$ for $s_i, s_j \in D$

  If we have two couple of points with the same distance then they will have same co-variance, even if in different directions!

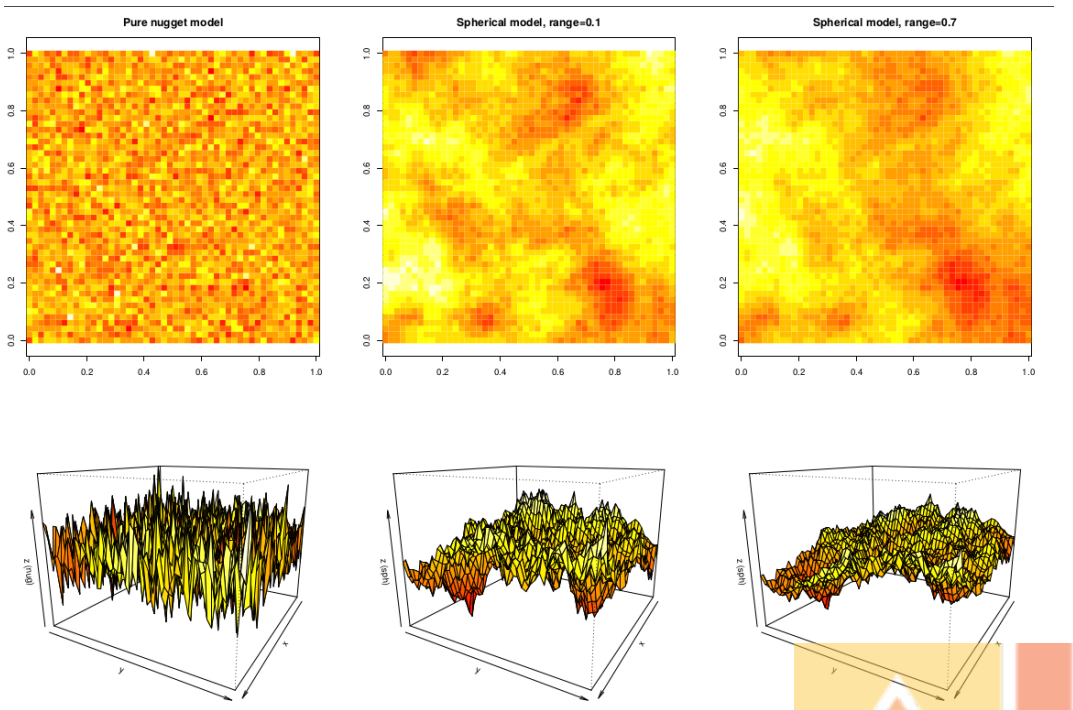
- Note that we have more properties which we don't see!

Now we see a few examples of realisations of a field with different co-variance models to see how much the structural properties matter!

Moro-ever we see how important is how the vario-gram behaves close to the origin: is the vario-gram linear or more than linear in zero?

This influences the smoothness of the realisations we expect from the vario-gram, indeed the closer to zero the derivative in zero the smoother the model!

Consider the following figure in which we have three examples of realisations:



Then:

- The first realisation comes from a field that is completely un-correlated as it's a random noise: we don't see any patterns, nor spatial dependence!

  No smoothness in the realisation!

  In the figure below we see the 3−dimensional representation of the realisation and above the contour plot of the realisation!

- In the central panel we have realisation from a spherical model that has a vario-gram that is linear in the origin!

  The second and third panel differ only form the range of the vario-gram: the second panel has a range of 0.1 while the third panel has a range of 0.7

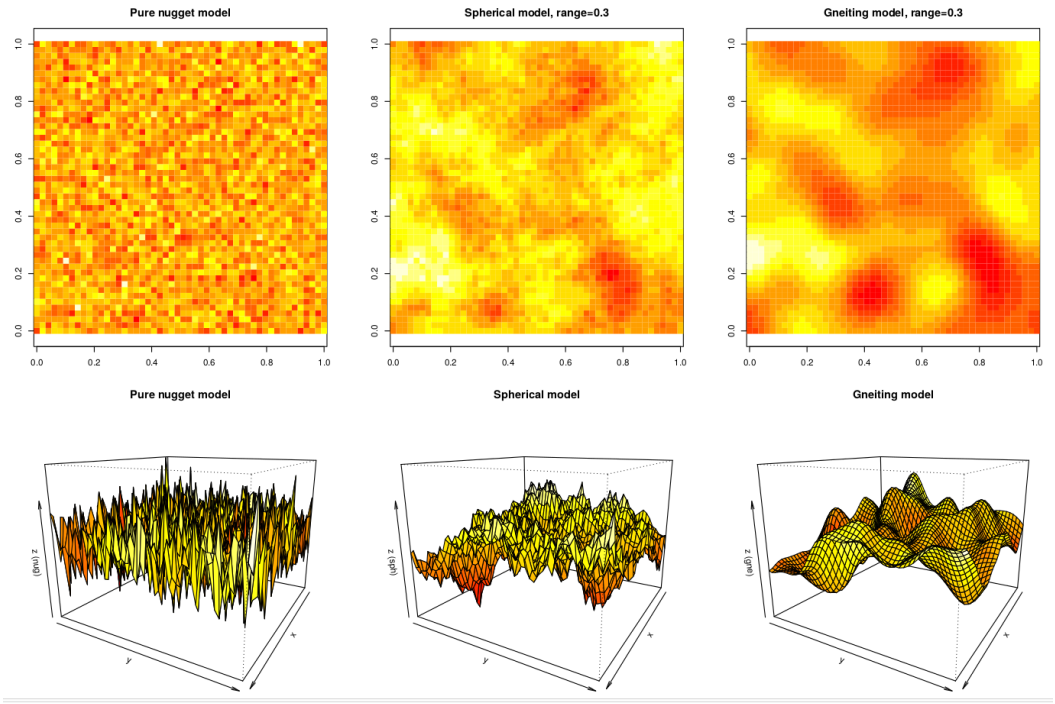
  The third panel is smoother compared to the second one: indeed looking at the contour plots we see that the hot-spots tend to be bigger in the third panel compared to the second panel!

  Indeed this is spatial dependence coming into play:

  – In the third panel a point 0.7 distant from the origin, at max, is still correlated.
  – Whereas in the second panel a point 0.2 distance from the origin, at max, is still correlated!

    Thus in this case the correlation disappears sooner!

Consider the following figure in which we have three examples of realisations:



Then:

- The left panel is a white noise
- The central panel is the **spherical model**, with a range of 0.3: this is a model whose vario-gram is linear!

- The third panel is the **Gneiting model** with a range of 0.3: this is a model whose vario-gram is quadratic in zero!

We see that the second and third panel, having the same range, have spots of similar dimension in the contour plot, but we have a clear difference in the regularity (smoothness) of the realisations!

Indeed the closer the derivative (of the vario-gram), in the origin, is to zero, the smoother the realisation of the process!

How can we estimate the spatial dependence? We need to estimate the vario-gram!

To do so we assume that we have a second order stationarity and isotropic random field, thus:

$$2\gamma(s_1 - s_2) = \mathbb{E}[(z_{s_1} - z_{s_2})^2] = Var(z_{s_1} - z_{s_2})$$

Since we are assuming that $(m_{s_1} - m_{s_2})^2 = 0$ due to second order stationarity!

Note: If $(m_{s_1} - m_{s_2})^2 \neq 0$ the following procedure provides a biased estimate!

We need to estimate an expected value: so we use the empirical estimate of the expected value from data, and so we need to compute the sample mean.

We build the following estimator:

$$2\hat{\gamma}(h) = \frac{1}{|N(h)|} \sum_{(i,j) \in N(h)} (z_{s_i} - z_{s_j})^2$$

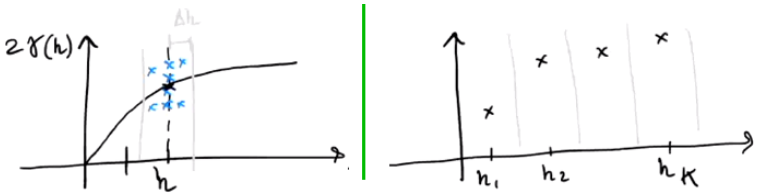
where $N(h) = \{(i,j) : \|s_i - s_j\| = h\}$

So we could do the above $\forall h$ and reconstruct the vario-gram, but in reality we don't have observations for every h!

Indeed for some h we don't have so many observations and so we don't consider h but slices of thickness Δh

So we now define $N(h)$ as $N(h) = \{(i,j) : \|s_i - s_j\| \in (h - \Delta h, h + \Delta h)\}$

We have the following situation:



So for each slice we compute the sample mean: $h_1, \ldots, h_k$ so we have a discrete approximation of the vario-gram!

The problem is that we don't get a vario-gram which makes sense: if we join the above point the result may not respect the very needed properties defined above (e.g: symmetry, Conditional Negative Definiteness, ...)

So the empirical estimator is used as a first guess of the vario-gram, to evaluate a number of property that our vario-gram should have!

Note: Non-parametric methods here aren't so good since it's hard to check the Conditional Negative Definiteness!

Thus we need a model: we use parametric families of vario-gram $2\gamma(h, \theta)$ with $\theta \in \Theta$ that we know full-fill our desired properties for the vario-gram!

Then we look for the θ which best approximates those points that we find in the empirical estimator!

There are many parametric families which can also be combined, indeed if γ_1, γ_2 are two different models of vario-gram then $\gamma_1 + \gamma_2$ is still valid!

Moreover a number of other properties holds, such as: $\alpha\gamma_1$ is still valid, if $\alpha > 0$

So once we define our parametric family (or families) of models for the vario-gram then we want to find the best parameter vector θ to fit the empirical estimates.

We can use **OLS,GLS** or others to find:

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \sum_{k=1}^{K} (\hat{\gamma}(h_k) - \gamma(h_k, \theta))^2$$

Note that this is typically a non-linear problem as the above function is typically non-linear in the range!

Note that some of the parametric families of model are on the notes, while others we will see them in the R Lab!

Note that the vario-gram is scale dependent and the parametric families are metric dependent!

Note that the nugget effect is controlling the noise to signal ratio!

26 Lecture 40: 22nd Of May 2020

We have our sample sites $s_i \in D \subseteq \mathbb{R}^d$ with $i = 1, \dots, n$ and we have real valued random variables which are observations z_{s_i}

How can we make prediction at a target location s_0 for which we have no observations? We want to build a predictor for the un-observed z_{s_0}

We assume that the observations are partial observations of a random field $\{z_s, s \in D\}$ In general we find the best measurable transformation of the data, that is:

$$z^*_{s_0} = f(z_{s_1}, \dots, z_{s_n})$$

where f is best: it minimises some criterion (e.g: expected value fo the error), so that f is such that:

$$\min \mathbb{E}[(z_{s_0} - f(z_{s_1}, \dots, z_{s_n}))^2]$$

The solution of this general problem is the conditional expectation: $\mathbb{E}[z_{s_0}|z_{s_1}, \dots, z_{s_n}]$ This solution has some properties:

1) The solution not only minimises $\mathbb{E}[(z_{s_0} - f(z_{s_1}, \dots, z_{s_n}))^2]$ but also:

$$\mathbb{E}[(z_{s_0} - f(z_{s_1}, \dots, z_{s_n}))^2|z_{s_1}, \dots, z_{s_n}]$$

2) The solution is unbiased: $\mathbb{E}[\mathbb{E}[z_{s_0}|z_{s_1}, \dots, z_{s_n}]] = \mathbb{E}[z_{s_0}]$

3) The solution is interpolating the data. So If $s_0 = s_i$ Then: $\mathbb{E}[z_{s_i}|z_{s_1}, \dots, z_{s_n}] = z_{s_i}$

4) If the random field $\{z_s, s \in D\}$ is Gaussian, then we know that: $\mathbb{E}[z_{s_0}|z_{s_1}, \dots, z_{s_n}] = \lambda_0 + \sum_i \lambda_i z_{s_i}$

So how can we find the solution if the random field is not Gaussian? We void to look for the conditional expectation but we look among the linear combinations of the data which minimise

$$\mathbb{E}[(z_{s_0} - f(z_{s_1}, \dots, z_{s_n}))^2]$$

We look for a linear, unbiased and interpolating (of the data), predictor!

This is the idea behind **Kriging:** look for best linear unbiased predictor. So we look for:

$$z^*_{s_0} = \lambda^*_0 + \sum_i \lambda^*_i z_{s_i}$$

where: $\lambda^*_0, \dots, \lambda^*_n$ solve a constrained optimisation problem, namely:

$$\min_{\lambda_i \in \mathbb{R}} \mathbb{E}\left[\left(z_{s_0} - \left(\lambda_0 + \sum_i \lambda_i z_{s_i}\right)\right)^2\right] \text{ subject to } \mathbb{E}[\lambda_0 + \sum_i \lambda_i z_{s_i}] = \mathbb{E}[z_{s_0}]$$

This is an approximation of the real solution (i.e: conditional expectation) in the Gaussian Case. This doesn't hold in general!

Note that for now we haven't made any assumptions on the field! To solve the above problem we need to make some assumptions!

Depending on the assumptions we have different types of **Kriging**:

- If the mean is unknown then we have two sub-cases:

  - If we have second order stationarity we have **Ordinary Kriging!**
  - If we have non-stationarity we have **Universal Kriging!**
- If the mean is known we have **Simple Kriging!** This case is too ideal and not applicable in reality so we don't see it!

Let's for now focus on **Ordinary Kriging**: we suppose our data comes from a random field $\{z_s, s \in D\}$ that is second order stationary, so that:

- $\mathbb{E}[z_s] = m \; \forall s \in D$
- $Cov(z_{S_1}, z_{s_2}) = c(s_1 - s_2) \; \forall s_1, s_2 \in D$ which is the co-variogram.

We assume c to be **known** and we assume that the mean m of the random field is **unknown**!

The **Kriging Problem** can be formulated as follows: Let $z_{s_0}^{\lambda} = \lambda_0 + \sum_i \lambda_i z_{s_i}$ Then we want to find $\lambda_0, \dots, \lambda_n$ so that:

$$(1) \quad \min \mathbb{E}[(z_{s_0} - z_{s_0}^{\lambda})^2] \text{ subject to unbiased-ness: } \mathbb{E}[z_{s_0}^{\lambda}] = \mathbb{E}[z_{s_0}]$$

In order to solve this, we first tackle then unbiased-ness: we need to impose that $\mathbb{E}[z_{s_0}^{\lambda}] = \mathbb{E}[z_{s_0}]$ thus:

$$\mathbb{E}[z_{s_0}^{\lambda}] = \lambda_0 + \sum_i \lambda_i \mathbb{E}[z_{s_i}] = \lambda_0 + \sum_i \lambda_i m = \mathbb{E}[z_{s_0}] = m$$

Note that all this follows by stationarity!

Thus the unbiased-ness constraint imposes that: $\lambda_0 = 0$ and $\sum_i \lambda_i = 1$

Note: since we don't know m the above result is stronger than unbiased-ness as it's Uniform unbiased-ness as the above holds for any value of m!

So the objective functional, taking into account the unbiased-ness constraint, becomes: $z_{s_0}^{\lambda} = \sum_i \lambda_i z_{s_i}$

Now we write the Lagrangian Function: $\Phi(\underline{\lambda}, \mu) = \mathbb{E}[(z_{s_0} - z_{s_0}^{\lambda})^2] + 2\mu \left(\sum_i \lambda_i - 1 \right)$ where μ is the Lagrangian Multiplier.

Now:

$$\mathbb{E}[(z_{s_0} - z_{s_0}^{\lambda})^2] = Var\left[z_{s_0} - \sum_i \lambda_i z_{s_i} \right] = Var[z_{s_0}] + Var\left[\sum_i \lambda_i z_{s_i} \right] - 2Cov\left(z_{s_0}, \sum_i \lambda_i z_{s_i} \right) =$$

$$= c(0) + \sum_i \sum_j \lambda_i \lambda_j Cov(z_{s_i}, z_{s_j}) - 2 \sum_i \lambda_i Cov(z_{s_0}, z_{s_i}) =$$

$$= c(0) + \sum_i \sum_j \lambda_i \lambda_j c(s_i - s_j) - 2 \sum_i \lambda_i c(s_0 - s_i)$$

This is due to unbiased-ness!

Thus the Lagrangian becomes: $\Phi(\underline{\lambda}, \mu) = c(0) + \sum_i \sum_j \lambda_i \lambda_j c(s_i - s_j) - 2 \sum_i \lambda_i c(s_0 - s_i) + 2\mu \left(\sum_i \lambda_i - 1 \right)$

Now we set to zero the partial derivatives: $\frac{\partial \Phi}{\partial \lambda_i} = 0 \; \forall i = 1, \dots, n$ and $\frac{\partial \Phi}{\partial \mu} = 0$ Thus:

$$\begin{cases} \frac{\partial \Phi}{\partial \lambda_i} = 2 \sum_j \lambda_j c(s_i - s_j) - 2c(s_0 - s_i) + 2\mu = 0 \;\; \forall i = 1, \dots, n \\ \frac{\partial \Phi}{\partial \mu} = \sum_i \lambda_i - 1 = 0 \end{cases}$$

So we get the **Kriging System**:

$$\begin{cases} \sum_j \lambda_j c(s_i - s_j) + \mu = c(s_0 - s_i) \; \forall i = 1, \dots, n \\ \sum_i \lambda_i = 1 \end{cases}$$

Solving this system we get the weights! The above is a linear system in λ and μ so it can be written in matrix form:

$$\begin{bmatrix} c(s_1 - s_1) & \dots & c(s_1 - s_n) & 1 \\ \dots & \dots & \dots & \dots \\ c(s_n - s_1) & \dots & c(0) & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ \mu \end{bmatrix} = \begin{bmatrix} c(s_0 - s_1) \\ \vdots \\ c(s_0 - s_n) \\ 1 \end{bmatrix}$$

The first matrix is a block matrix: in the top-left we have the co-variance matrix (all matrix but last row and last column)!

So if $\Sigma = Cov(\underline{z})$ so that $\Sigma_{ij} = Cov(z_{s_i}, z_{s_j}) = c(s_i - s_j)$ thus we can re-write the above system in block matrix:

$$\begin{bmatrix} \Sigma & \underline{1} \\ \underline{1}^T & 0 \end{bmatrix} \begin{bmatrix} \underline{\lambda} \\ \mu \end{bmatrix} = \begin{bmatrix} \underline{\sigma}_0 \\ 1 \end{bmatrix}$$

where: $\sigma_{0_i} = Cov(z_{s_0}, z_{s_i}) = c(s_0 - s_i)$

Now we solve the system above: we get $\underline{\lambda}^*$ from which we get: $z^*_{s_0} = \underline{\lambda}^{*T} \underline{z}$

Note: we can get an estimate of the variance of the prediction error. Indeed the **Kriging Variance** is given by:

$$\sigma^2_{OrdKrig}(s_0) = \Phi(\underline{\lambda}^*, \mu^*) = c(0) - \sum_i \lambda^*_i c(s_0 - s_i) - \mu^*$$

The **Kriging Predictor** is the one which makes the above uncertainty as small as possible, under unbiased-ness!

Note that all the above is made assuming c is known! In practice we don't know it so we estimate it from the data!

So we have some steps:

- Take data and estimate $\hat{\gamma}$ from which we get $\hat{c}$ from which we get $\hat{\Sigma}$
- Then we solve the **Kriging System** above with $\hat{\Sigma}$ in place of Σ

So at the end we get a **Plug-in Kriging Predictor**!

Since the **Kriging Variance** uses c then that doesn't consider the uncertainty in estimate the co-variogram itself!

How to go from $\hat{\gamma}$ to $\hat{c}$? We know that $\gamma(h) = c(0) - c(h) \implies c(h) = c(0) - \gamma(h)$

But under stationarity the vario-gram has an horizontal asymptote at $c(0)$ since the process is assumed to be stationary!

So we can compute the co-variogram from the variogram as: $c(h) = c(0) - \gamma(h) = \lim_{\tilde{h}\to\infty} \gamma(\tilde{h}) - \gamma(h)$

Note that the Kriging predictor is interpolating even though we are not imposing it!

Universal Kriging: we no more assume stationarity!

Given the observation $z_{s_1}, \dots, z_{s_n}$ from the non-stationary random field $\{z_s, s \in D\}$ We assume the random field can be written as follows:

$$z_s = m_s + \delta_s \ \forall s \in D$$

where: $m_s = \mathbb{E}[z_s]$ is the **drift** and $\delta_s = z_s - m_s$ is the **residual**!

Now we assume that: $m_s = \sum_{\ell=0}^{L} a_\ell f_\ell(s)$ where:

- a_ℓ are unknown coefficients
- $f_\ell(s)$ are the regressors which we assume are known over the entire domain D!
  Note that this is a strong assumption but is needed!

Moreover we assume that $\{\delta_s, s \in D\}$ is a random field with:

- $\mathbb{E}[\delta_s] = 0$
- $Cov(\delta_{s_1}, \delta_{s_2}) = c(s_1 - s_2) = Cov(z_{s_1}, z_{s_2})$ so that its stationary!

So we allow a non-stationary drifter but a second order stationary random field for the residuals!

The **Kriging Problem** is the same as above: $z^\lambda_{s_0} = \lambda_0 + \sum_i \lambda_i z_{s_i}$ Then we want to find $\lambda_0, \dots, \lambda_n$ so that minimise the variance of prediction error under unbiased-ness, that is:

$$(1) \quad \min \mathbb{E}[(z_{s_0} - z^\lambda_{s_0})^2] \text{ subject to unbiased-ness: } \mathbb{E}[z^\lambda_{s_0}] = \mathbb{E}[z_{s_0}]$$

In order to solve the above problem, first we impose unbiased-ness, namely: $\mathbb{E}[z^\lambda_{s_0}] = \mathbb{E}[z_{s_0}]$ Thus:

$$\mathbb{E}[z^\lambda_{s_0}] = \lambda_0 + \sum_i \lambda_i \mathbb{E}[z_{s_i}] = \lambda_0 + \sum_i \lambda_i m_{s_i} = \lambda_0 + \sum_i \lambda_i \sum_{\ell=0}^{L} a_\ell f_\ell(s_i) = \mathbb{E}[z_{s_0}] = m_{s_0} = \sum_{\ell=0}^{L} a_\ell f_\ell(s_0)$$

Thus: $\lambda_0 + \sum_i \lambda_i \sum_{\ell=0}^{L} a_\ell f_\ell(s_i) = \sum_{\ell=0}^{L} a_\ell f_\ell(s_0) \implies \lambda_0 = 0$ and $\sum_i \lambda_i f_\ell(s_i) = f_\ell(s_0) \ \forall \ell = 0, \dots, L$

Now we write down the Lagrangian Function (i.e: the objective functional):

$$\Phi(\underline{\lambda}, \mu) = \mathbb{E}[(z_{s_0} - z^\lambda_{s_0})^2] + 2\sum_{\ell=0}^{L} \mu_\ell \left(\sum_i \lambda_i f_\ell(s_i) - f_\ell(s_0) \right)$$

where μ is the Lagrangian Multiplier and where $z^\lambda_{s_0}$ has $\lambda_0 = 0$

The first term is the same as before, while the second is different as it accounts for more constraints!

Making the same calculations as before we get:

$$\Phi(\underline{\lambda}, \mu) = c(0) + \sum_i \sum_j \lambda_i \lambda_j c(s_i - s_j) - 2\sum_i \lambda_i c(s_0 - s_i) + 2\sum_{\ell=0}^{L} \mu_\ell \left(\sum_i \lambda_i f_\ell(s_i) - f_\ell(s_0) \right)$$

Now we set the partial derivatives, as before, equal to zero:

$$\begin{cases} \dfrac{\partial \Phi}{\partial \lambda_i} = 2\sum_j \lambda_j c(s_i - s_j) - 2c(s_0 - s_i) + 2\sum_{\ell=0}^{L} \mu_\ell f_\ell(s_i) = 0 \\ \dfrac{\partial \Phi}{\partial \mu_\ell} = \sum_i \lambda_i f_\ell(s_i) - f_\ell(s_0) = 0 \end{cases}$$

Making the computations we get the **Universal Kriging System**:

$$\begin{cases} \sum_j \lambda_j c(s_i - s_j) + \sum_{\ell=0}^{L} \mu_\ell f_\ell(s_i) = c(s_0 - s_i) \\ \sum_i \lambda_i f_\ell(s_i) = f_\ell(s_0) \end{cases}$$

In block-matrix from the above can be written as:

$$\begin{bmatrix} \Sigma & \mathbb{F} \\ \mathbb{F}^T & O \end{bmatrix} \begin{bmatrix} \underline{\lambda} \\ \underline{\mu} \end{bmatrix} = \begin{bmatrix} \underline{\sigma}_0 \\ \underline{f}_0 \end{bmatrix}$$

where:

- $\mathbb{F}$ is the design matrix of the linear model $\underline{z} = \mathbb{F}\underline{a} + \underline{\delta}$ where $\underline{\delta}$ is the vector of residuals (i.e: ϵ of the previous lectures and not $\hat{\epsilon}$)
- In the above O is just a matrix of zeros
- $\underline{f}_0$ is the *design vector*, that is: $f_{0_\ell} = f_\ell(s_0)$
  Note that it is here that we see the importance of knowing the regressors all over D!

Note that this has the same structure as before although before we only had the intercept as regressor!

So from above we solve the system we get λ^* so that we get the solution $z^*_{s_0} = \lambda^* \underline{z}$

Note that in this case the **Kriging Variance** shouldn't be used as it provides a big underestimated estimation of the variability of the prediction error!

Indeed all of the above is based on the fact that Σ is known but we don't know it in practice so we need to estimate $\hat{\gamma}$ from which we get $\hat{c}$ from which we get $\hat{\Sigma}$

But here the random field is non stationary so we don't know how to estimate $\hat{\gamma}$!

If we knew the residuals δ_{s_i} then we could estimate $\hat{\gamma}$ from this residuals under second order stationary!

We don't know the residuals so:

- We estimate the residuals $\hat{\delta}_{s_i}$
- From this we estimate the vario-gram $\hat{\gamma}$ from which we get $\hat{c}$ from which we get $\hat{\Sigma}$

Seems reasonable? Well let's focus on the residual estimations: $\hat{\delta}_{s_i} = z_{s_i} - \hat{m}_{s_i}$ where: $\hat{m}_{s_i} = \sum_{\ell} \hat{a}_\ell f_\ell(s_i)$

So using **GLS** we get:

$$\hat{a}_{GLS} = (\mathbb{F}^T \Sigma^{-1} \mathbb{F})^{-1} \mathbb{F}^T \Sigma^{-1} \underline{z}$$

But this expression has inside Σ^{-1} so to estimate $\hat{a}$ we need Σ! We are stuck!

So we use iterative algorithm as we have non-linearities:

0) We initialise $\hat{a} = \hat{a}_{OLS} = (\mathbb{F}^T \mathbb{F})^{-1} \mathbb{F}^T \underline{z}$
1) Now we compute $\hat{\delta_{s_i}}$
2) From this we compute $\hat{\gamma}$ from which we get $\hat{c}$ and $\hat{\Sigma}$
3) We can now update: $\hat{a} = \hat{a}_{GLS} = (\mathbb{F}^T \Sigma^{-1} \mathbb{F})^{-1} \mathbb{F}^T \hat{\Sigma}^{-1} \underline{z}$

We repeat step **1)** through **3)** until convergence!

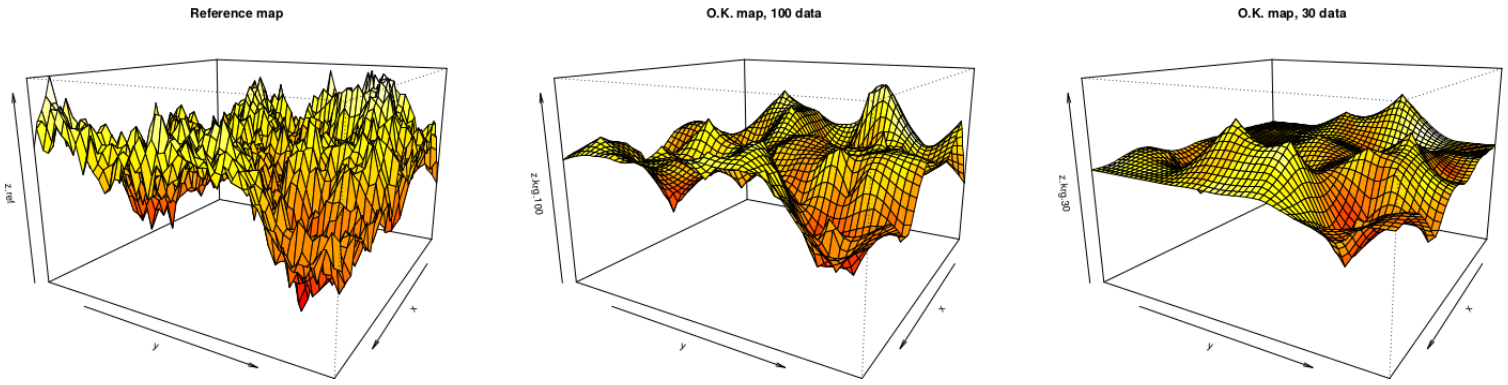
Note that the above iterative algorithm hasn't been proved to converge but it happens and it happens very fast!

Note the **Universal Kriging Variance** assumes Σ is known but the uncertainty in the estimation can be huge as at each step we have uncertainty in the iterative algorithm!

Base Line: don't use the **Universal Kriging Variance**

Anyhow from the above iterative method we get $\hat{\Sigma}$ which we can plug in in the **Kriging System** above and we get the **Plug-in Universal Kriging Predictor**!

So how does Krigging works in practice? Consider the following figure:



Where:

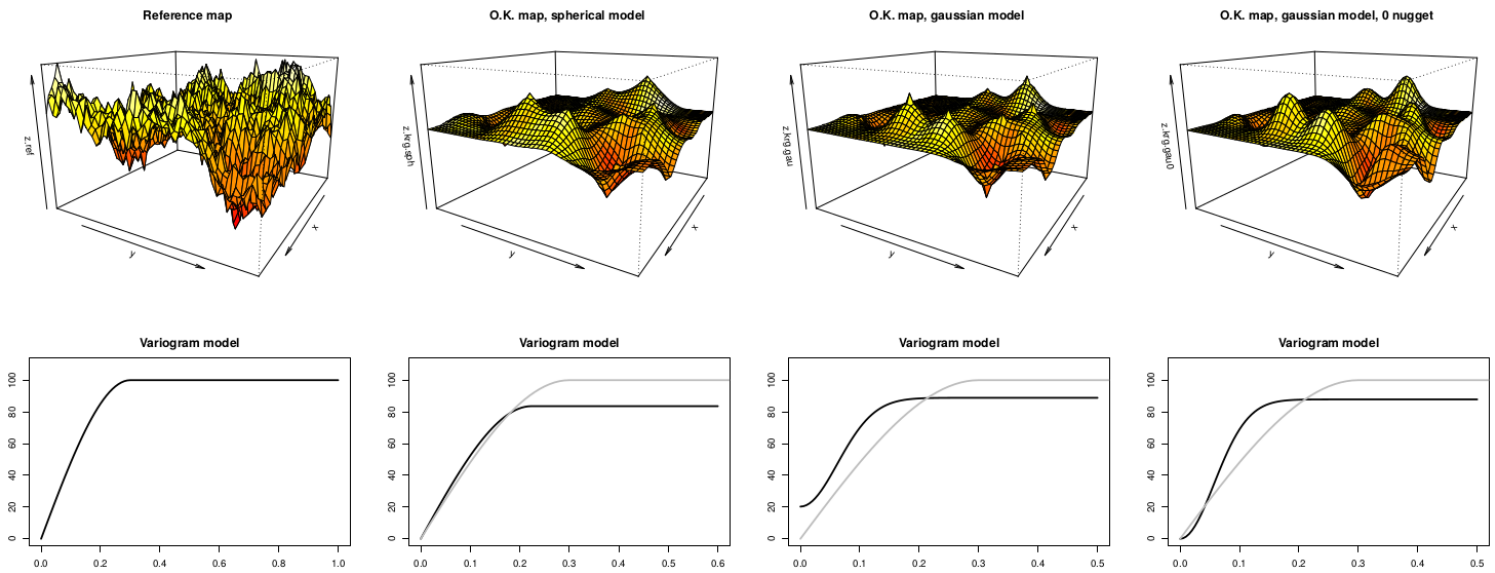
- In the left we have our realisation of the field
- In the centre we have the Ordinary Kriging reconstruction based on 100 data
- On the right we have a Ordinary Kriging reconstruction based on 30 data!

Note how these predictions are smoother than the realisation map, as were based on a spherical model that is linear in zero!

When we are far from the data we loose spatial dependence and the prediction is based on the estimation of the mean of the field!

Note that the above anyway interpolates the data!

Consider now the following figure:



We have three reconstructions of the same process realisation which is in the left-most panel, which are done:

- With a spherical model that is linear in the origin without nuggets:
- With a Gaussian model that is quadratic in the origin with a nugget!
  Note that due to the nugget we see weird behaviour: where we have data we interpolate the data and then we have a jump!

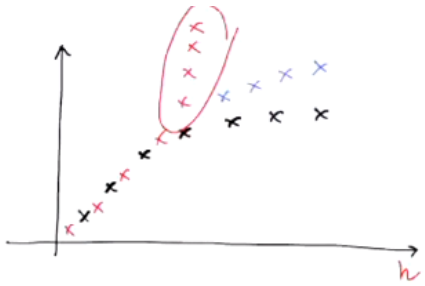
  So we would have a smooth and differentialble surface expect for where we have jumps!

  Note that we see cones but really are discontinuities: this is due to the fact that the prediction is done on a fine grid!
- With a Gaussian model that is quadratic in the origin without nuggets!
  This is not only continuous but also differentiable! This is due to the fact that in the origin the model is quadratic!

So we should choose the vario-gram structure in order to have realisations that are coherent with the kind of realisations that we would expect from the phenomenon!

The default choice is a spherical (or exponential) model with a linear vario-gram in the origin!

Note that if we model a phenomenon as stationary when in fact it's not, we would have the following situation in the vario-gram:



If the phenomenon is non-stationary we have the red crosses!

This is because of a property of the vario-gram: it must be slower than quadratic as $h \to \infty$

Indeed: $2\gamma(s_1 - s_2) = Var(z_{s_1} - z_{s_2}) = \mathbb{E}[(z_{s_1} - z_{s_2})^2] - (m_{s_1} - m_{s_2})^2$ So if the phenomenon is non-stationary then m_s is not constant and so $(m_{s_1} - m_{s_2}) \neq 0$

So we would be estimating with a positive bias as we are estimating: $2\gamma(s_1 - s_2) + (m_{s_1} - m_{s_2})^2$

27 Lecture 43: 28th Of May 2020

Functional Data Analysis

We have complex and high dimensional data having a function nature, re-presentable by curves, surfaces for example!

Examples are three-dimensional curves representing carotid artery of people, or traffic of cellular phones observed on a spatial lattice!

Note: functional data is also called object-oriented data analysis!

Informally functional data are entities that can be described through a function (e.g: curve, surface, image)!

a functional data set consist of sample of functional observations and so we still have a discrete grid!

However the observed values, reflect a smooth variation of the phenomenon, so one might be interested, not only in point-wise values, but also in differential properties of the data!

So why do we want to move to a functional representation? How can we do it (e.g: functional smoothing)?

If the data comes from a continuous phenomenon, then it makes sense to consider such datum coming from a continuous function!

If we have for instance the height of 10 girls for 31 years, the $p = 31 >> n = 10$ in which not much would be done!

Instead, we embedded the $n = 10$ functions, in a space in which we see a cloud of 10 points representing our data!

Then we can take derivatives and all the stuff we can do with usual functions, in mathematics!

For instance we could then compute the acceleration which may be very interesting, with that we could then observe very interesting phenomena!

If we have different functions, we can align them, and then take the average, so that it makes sense!

The space in which we embed the functional data should consider the properties of these functions: if we have a density function then it must integrate to 1 and the space in which we embed it should consider this!

Typical goals of **FDA**:

- Represent data in ways that aid further analysis, display the data to highlight their salient features!
- Study the main sources of pattern and variation among the data!
- Explain an outcome (response) using input-independent variable information!
- We might want to classify the data, or compare groups of data, with respect to certain type of variations!

In this course we look at:

- Representing data: given raw-discrete observations, represent the data through a functional from! How do to functional smoothing?
- We want to reduce the high dimensionality of the representation space and highlights the main sources of variability (functional PCA)!

- Alignment and clustering of data!

Note that alignment is vital in order to then conduct sound data analysis!

For example people have different biological clock and so functional growth datum should be aligned before confronting them with others!

Outline of the course:

- Hilbert Space model for functional data
- Smoothing and interpolation of functional data
- FDA and dimensionality reduction in Hilbert Spaces
- Data alignment and clustering

We need some packages in R, namely: *fda, Refund, mgcv, fdakma, fdaPDE*. We may also need the MatLab code *PACE*.

The main book of reference in *Functional Data Analysis by Ramsay* and *Inference for Functional Data With Application by Horvath*!

---

1) Hilbert space model for functional data:
Hilbert spaces are the simplest spaces in which to embed the functional data, so this doesn't always work (e.g: co-variance matrices are embedded in Riemannian Manifolds)!

In Hilbert Spaces the points are just functions and so many methods of multi-variate statistics can be easily extended, since we can endow this space with the same type of geometry: vector space structure and inner product!

Once we define an inner product (i.e: a bi-linear, symmetric, positive definite form) in a Hilbert Space we can then work with Pythagoras' Theorem and with orthogonal projections!

A real Hilbert space H is an inner product space that is complete, in the norm induced by the inner product!

Moreover a Hilbert space is separable if it contains a dense countable sub-set!

If H is a separable Hilbert space then we have an ortho-normal basis and we can express any point of this space with such basis!

The space of real-value squared-integrable functions L^2 is an example of a Hilbert space, in which the inner product is defined as:

$$(f_1, f_2) = \int (f_1(t) \cdot f_2(t))dt$$

Recall that more precisely L^2 is the quotient space with respect to the equivalence relation:

$$x = y \text{ if } \int |x(t) - y(t)|^2 dt = 0$$

B^2 is the Bayes Hilbert space, and it's the space of density functions on a close interval I with logarithm in L^2

Note that the geometry of L^2 wouldn't make sense for density functions: if you point-wise sum two densities you don't get a density!

We have the following inner product:

$$\langle f, g\rangle_B = \frac{1}{2\eta}\int_I\int_I \ln\left(\frac{f(t)}{f(s)}\right)\ln\left(\frac{g(t)}{g(s)}\right)dtds$$

Here the sum is called *perturbation* and it's given by: $(f \oplus g)(t) = \dfrac{f(t)g(t)}{\int_I f(s)g(s)ds}$

The product by a constant is called *powering* and it's given by: $(\alpha \odot f)(t) = \dfrac{f(t)^\alpha}{\int_I f(s)^\alpha ds}\ t \in I$

Since B^2 is isomorphic to L^2 then we can always transform the data in L^2 through isometric isomorphism! One is the *centred log-ration transformation* which is given by:

$$clr(f)(t) = f_c(t) = \ln(f(t)) - \frac{1}{\eta}\int_I \ln(f(s))ds$$

Note that exponential families of distributions are simply linear spaces in B^2 Moreover the *perturbation* is the Bayes update rule!

We then need to choose an embedding for the data: it should be suitable for the characteristics of the object we are analysis and for the goal of the analysis!

We may need Sobolev Geometry, so that we take into account of derivatives in the norms! We may want to compute torsion of the curves, and so on!

Let H be a Hilbert space whose points are functions defined on a closed interval $T = [t_{min}, t_{max}]$ Then:

- **Definition:** a functional random variable is a random element (i.e: measurable function with values in a Hilbert space) on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ in the space $H : X : \Omega \to H$
- **Definition:** A functional datum x is a realisation of a functional random variable. So for $\omega \in \Omega$ we have: $x = X(\omega) : T$
- **Definition:** A functional data set is a collection of realisation of functional datum x_i!

Let $X : \Omega \to H$ be a functional random variable in H and suppose that $\mathbb{E}[\|X\|_H^4] < \infty$ Then:
Definition: We call *Fréchet* mean of X, the unique element μ of H that solves:

$$\arg\inf_{x\in H}\mathbb{E}[\|X - x\|_H^2]$$

Note:

- If $H = B^2$ then: $\mu = clr^{-1}(\mathbb{E}[clr(X)])$
- If $H = L^2$ then: $E[X(t)] = \mu(t)$

  Note that: $E[X](t) = E[X(t)]$ so we can forget about the Fréchet definition of mean of X!

In any H we can estimate the mean with the sample estimator $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$

Obviously here the meaning of sum is different based on H, for example: the above summation is a sequence of perturbations if $H = B^2$

Let $X : \Omega \to H$ be a zero-mean functional random variable in H such that $\mathbb{E}[\|X\|_H^4] < \infty$ Then:
Definition: We call *co-variance operator* of X, the operator from H to H, defined as:

$$C_x = \mathbb{E}[\langle X, x\rangle X], x \in H$$

It needs to be symmetric and positive semi-definite. Moreover we need to require that the sum of the eigenvalues is finite!

This implies that co-variance operators are *Hilbert-Schmidt Operators*!

If $H = \mathbb{R}^p$ then the co-variance operator coincides with the linear operator defined by the co-variance matrix: $\mathbb{E}[\langle x, X\rangle x] = \mathbb{E}[XX^T x] = \sum x$

If $H = L^2$ then the co-variance operator can be defined through a kernel operator:

$$[Cx](t) = \int_I c(s,t)x(s)d(s) \quad \text{for } x \in L^2$$

where the co-variance kernel is precisely the point-wise co-variance: $c(s,t) = \mathbb{E}[X(s)X(t)]$

In any H the co-variance operator can be estimated through the sample co-variance operator, which is given by:

$$Sx = \frac{1}{N}\sum_{i=1}^{N} \langle X_i, x\rangle X_i \quad \text{for } x \in H$$

If $H = L^2$ then we can use the following alternative definition:

$$[Sx](t) = \int_T \hat{c}(s,t)x(s)d(s) \quad \text{for } x \in L^2 \text{ with } \hat{c}(s,t) = \frac{1}{N}\sum_{i=1}^{N} X(s)X(t)$$

Note that in the heat-map plot of the sample co-variance kernel, the main diagonal, goes from the bottom left, to the top right!

---

2) Smoothing and interpolation of functional data:
This is considered a pre-processing: we smooth the functions separately but not all together, albeit we already decided the embedding space and the smoothing criterion!

To perform smoothing we can then use a specific basis, so that we can easily do the computation!

Typical observations of functional data are discrete and noisy: the record of each function x_i contains n_i pairs (t_{ij}, y_{ij}) with $j = 1, \ldots, n_i$

We models these pairs as a linear model: $y_j = x(t_j) + \epsilon_j$ where we don't know the function x So that's the goal of the regression problem!

For each i we aim to reconstruct the underlying functional observation function x_i from the records (t_{ij}, y_{ij}) with $j = 1, \ldots, n_i$

Note that the assumptions on the properties of x_i, such as it's smoothness, will reflect on the way we proceed to reconstruct the data!

Depending on our prior knowledge on the measurement error (i.e: on the properties of the noise ϵ_j), we can decide to:

- Perform **Interpolation:** the functional form reconstructed actually interpolates its discrete observations (noiseless measurements)
- Perform **Smoothing:** the functional form is smoother than the actual observations (noisy measurement)

If there is measurement error, then by interpolating we carry this error in the functional representation: this can be bad in computing derivatives as the effect of noise is amplified!

Thus smoothing is usually preferred!

To interpolate or to smooth we need to choose a target functional form for x_i, that may depend on some parameters, and then we estimate the functional form based on the pair (t_{ij}, y_{ij})

The choice of the functional form depends on various factors:

- Which features do we wanna extract? For instance we may want regularity of the functional form if the target is a derivative (i.e: it's the differential information of the function).
- Why functional space embedding do we choose? When we choose a Hilbert space embedding we automatically identify possible ortho-normal basis!

In most cases Hilbert space embedding is employed and functions are represented by basis functions!

A set of basis functions is a set of known functions that are linearly independent and allows us to approximate arbitrarily well any function as a linear combination of a, sufficiently large number of, K of these functions!

How large should be the basis? that's a problem!

More precisely, if the basis functions are ϕ_k then we express a function x by the following linear expansion:

$$x = \underline{c}^T \underline{\phi} = \underline{\phi}^T \underline{c} \implies x(t) = \sum_{k=1}^{K} c_k \phi_k(t)$$

where $\underline{\phi} = [\phi_1, \dots, \phi_k]$ and $\underline{c} = [c_1, \dots, c_k]^T$

Now we know that in a Hilbert space we can always find an ortho-normal basis that allows approximating, with any desired precision, any element of the space through the following expansion:

$$x = \sum_{n=1}^{K} \langle x, u_n \rangle u_n$$

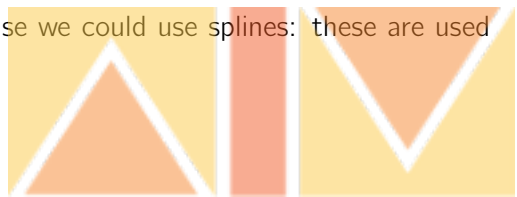
From the basis we get a design matrix and then we project orthogonally to the linear space generated by these function!

We can use the Fourier Basis functions: good for periodic phenomenon. These are very robust: small variability but could be biased!

Moreover they are also fast to be computed: Fast Fourier Transform! We have that:

$$\hat{x}(t) = c_0 + c_1 \sin(\omega t) + \cdots \implies \phi_0(t) = 1, \phi_{2r-1}(t) = \sin(r\omega t) \text{ and } \phi_{2r}(t) = \cos(r\omega t)$$

The Fourier Basis functions can't be used for discontinuous function. In such case we could use splines: these are used for non-periodic functional data!



In each sub-interval the spline is a polynomial of a certain order, which is equal to the degree plus one! To construct an $m-$order spline:

- We divide the interval of definition T into L sub-intervals: to do so we fix a set of *knots*
- Over each interval the spline is defined as a polynomial of order m: this is the number of constants needed to define the polynomial!

Note that each polynomial connect with continuity up to $m-2$ order of their derivatives!

We use these splines since then the derivative of the smoothed function is smooth!

We can use splines to fit discontinuous functions: we have two knots one on top of each other!

To gain flexibility in a spline we can increase the number of its knots: for example we can put more knots where the function exhibits more variability!

The total number of parameters needed to define a spline function with non-overlapping knots is the order plust the number of interiors knots: $m + L - 1$

B-splines are the most used in statistics: each of this spline are localised!

To actually performing smoothing we use least squares: we can use OLS to minimise the sum of squared errors between fitted values and observations!

We have the basis functions ϕ_i with $i = 1, \dots, K$ and we have the pairs of points (t_j, y_j) with $j = 1, \dots, n$ then we want to estimate the coefficients c_k of the linear model:

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t) + \epsilon_j$$

Our design matrix is:

$$\Phi = \begin{bmatrix} \phi_1(t_1) & \dots & \phi_k(t_1) \\ \vdots & \vdots & \vdots \\ \phi_1(t_n) & \dots & \phi_k(t_n) \end{bmatrix}$$

which is an $n \times k$ matrix!

From the theory of linear regression we know the solution is obtained by minimising the sum of squared errors between fitted values and observations, namely:

$$SMSSE(\underline{y}|\underline{c}) = \sum_{j=1}^{n} \left[y_j - \sum_{k}^{K} c_k \phi_k(t_j) \right]^2$$

Therefore, we get:

$$\underline{\hat{c}} = (\Phi^T \Phi)^{-1} \Phi^T \underline{y}$$

$$\underline{\hat{y}} = \Phi \underline{\hat{c}}$$

Note that $\underline{\hat{y}}$ is the projection of $\underline{y}$ over the space entreated by the columns of Φ, that are the evaluations of the basis functions in the measurements points!

OLS smoothing are appropriate if the measurement error are independent and identically distributed!

The degree of smoothness of the estimated curve depends on the number of basis functions employed!

We can use variable selection, and add more basis functions, and use a step-wise method to decide the degree we need!

We might risk to over-fit the data if we use too high degrees!

The error explodes when computing derivatives if we choose too smooth functions!

If the measurement error are not independent and identically distributed we need to use GLS weighted least squares! In this case we minimise the weighted sum of squared errors between fitted values and observations, that is:

$$SMSSE(\underline{y}|\underline{c}) = (\underline{y} - \Phi\underline{c})^T W(\underline{y} - \Phi\underline{c})$$

Where the matrix W is assumed to be positive definite and can be set for instance equal to the co-variance matrix of the errors!

The solution is found as: $\underline{\hat{c}} = (\Phi^T W\Phi)^{-1}\Phi^T W\underline{y}$

Note that here $\underline{c}$ is our β of regression!

We can then compute the variance of the estimator for the coefficients and we can compute confidence limits: we can compute one-at-the time confidence bands and so on!

For instance the variance of the estimator for the coefficients is:

$$Var[\underline{c}] = (\Phi^T W\Phi)^{-1}\Phi^T W\Sigma_e\Phi(\Phi^T W\Phi)^{-1}$$

This in the case of unweighted least squares and independent and identically distributed errors reduces to:

$$Var[\underline{c}] = \sigma^2(\Phi^T\Phi)^{-1}$$

The variance of the point-wise estimate of the curve is then obtained as the diagonal of the matrix:

$$Var[\underline{\hat{y}}] = \Phi Var[\underline{c}]\Phi^T$$

In the case of unweighted least squares and independent and identically distributed errors reduces to:

$$Var[\underline{\hat{y}}] = \sigma^2\Phi(\Phi^T\Phi)^{-1}\Phi^T = \sigma^2 S$$

The variance of the errors can be estimated from the residual sum of squares, namely:

$$s^2 = \frac{1}{n-K}\sum_j^n (y_j - \hat{y}_j)^2$$

As usual the problem is the bias-variance trade-off!

To choose K we need to evaluate when a drop in sampling variance occurs for a range of candidates K!

The larger the K the better fit, but there is higher risk to fit the noise! If K is too small we may miss important features of the underlying function we wish to estimate!

Thus for large K we have small bias and high variance, whereas for small K we have a big bias but small variance!

We can deiced how smooth we want the function and introduce a penalisation, so that we get penalised regression (e.g: lasso), so that we penalise when our smoothed function is far from the desired result!

We can thus quantify the roughness through the second derivative and then minimise the penalises SSE: for instance we can quantify the roughness through the second derivative (which measures the curvature of the function), that is:

$$Pen_2(x) = \int [D^2x(s)]^2 ds$$

Note that this is zero if we have a straight line. Then given λ we find x that minimise the penalised SSE, given by:

$$PenSSE_\lambda(x|\underline{y}) = [\underline{y} - x(\underline{t})]^T W[\underline{y} - x(\underline{t})]^2 + \lambda Pen_2(x)$$

We can then do (generalised) cross-validation for choosing the smoothing parameter!

λ i the smoothing parameter:

- If $\lambda \to \infty$ we give more emphasis to the penalisation and the fitted curve will be a straight line!
- If $\lambda \to 0$ then the curve approaches the smoothest twice-differentiable curve that interpolates our data!

Note: The functional form of x is a consequence of the objective function: for instance the curve x that minimises $PenSSE$ is a cubic spline with knots at the data points t_j

We can re-express the penalisation in the general case, as:

$$Pen_m(x) = \int [D^m x(s)]^2 ds = \underline{c}^T R \underline{c}$$

where: $R = \int D^m \phi(s) D^m \phi^T(s) ds$

Note: The final result for $\hat{c}$ is the one of ridge in the case in which $R = I$!

Plugging the above expression in the objective functions yields:

$$PenSSE_m(\underline{y}, \underline{c}) = [\underline{y} - \Phi \underline{c}]^T W[\underline{y} - \Phi \underline{c}]^2 + \lambda Pen_m(x)$$

This is minimised for: $\underline{\hat{c}} = (\Phi^T W \Phi + \lambda R)^{-1} \Phi^T W \underline{y}$

The criterion with which we fit the data implies a decision on the embedded space!

Remark: Embedded space and smoothing are much related!

The generalised cross-validation is defined as:

$$GCV(\lambda) = n^{-1} SSE \frac{1}{[n^{-1}\text{trace}(I - S_{\phi,\lambda})]^2}$$

where: $\underline{\hat{y}} = \Phi(\Phi^T W \Phi + \lambda R)^{-1} \Phi^T W \underline{y} = S_{\phi,\lambda} \underline{y}$

28 Lecture 44: 29th Of May 2020

3) FDA and Dimensionality reduction in Hilbert spaces:
Recall PCA, assume $N > p$ and that there is no collinearity (i.e: design matrix full rank). Then given a data set of N zero-mean multivariate observations $\underline{X}_1, \dots, \underline{X}_N$ in $\mathbb{R}^p$ we want to find the ortho-normal directions $\underline{a}_1, \dots, \underline{a}_p$ of maximum variability (for the data set)!

Equivalently for $k = 1, \dots, p$ we want to find:

$$\underline{a}_k = \arg\max_{\underline{a} \in \mathbb{R}^p} Var(\underline{a}^T \underline{X}) \text{ subject to } \underline{a}^T \underline{a} = 1 \text{ and } \underline{a}_j^T \underline{a}_k = 0 \ \forall j \neq k$$

We can re-write the above problem as follows:

$$\underline{a}_k = \arg\max_{\underline{a} \in \mathbb{R}^p} \frac{1}{N} \sum_{i=1}^{N} \langle \underline{a}, \underline{X}_i \rangle^2 \text{ subject to } \|\underline{a}\| = 1 \text{ and } \langle \underline{a}_j, \underline{a}_k \rangle \ \forall j \neq k$$

Note that if $\underline{X}_1, \dots, \underline{X}_N$ are not zero-mean they can be centred by subtracting the sample mean: then fro unbiased-ness we divide by $N - 1$ instead of N

If we call S the sample co-variance matrix of $\underline{X}_1, \dots, \underline{X}_N$ then the principal components are found as the eigen-vectors of the matrix S

Indeed for $k = 1, \dots, p$ they solve the eigen-equation: $S\underline{e}_k = \lambda_k \underline{e}_k$ where the eigenvalue λ_k associated with the eigen-vector $\underline{e}_k$ represents the variability along the direction $\underline{e}_k$

Recall that we call *score* x_{ik} the projection of the observation $\underline{X}_i$ along the direction $\underline{e}_k$, namely: $x_{ik} = \underline{X}_i^T \underline{e}_k = \langle \underline{X}_i, \underline{e}_k \rangle$

Moreover the *loadings* are the components of the eigen-vector: the weights we use for the linear combination of the original variables!

Can we do the same in any Hilbert space using its inner product? Yes!

Given a data set of N zero-mean functional observation $X_1, \dots, X_N$ in H, we want to find the directions of maximum variability, in H, of the data set.

So for $k = 1, \dots, N$ we want to find ξ_k maximising:

$$\frac{1}{N} \sum_{i=1}^{N} \langle \xi, X_i \rangle_H^2 \text{ subject to} \|\xi\| = 1 \text{ and } \langle \xi_i, \xi_j \rangle_H = 0 \ \forall j \neq k$$

We look for an ortho-normal system in H that maximises the variability of the corresponding projections: indeed $\langle \xi, X_i \rangle_H$ is the projection of X_i, along the direction ξ, in H!

Note that since $\langle \xi, X_i \rangle_H$ is a scalar and hence we are maximising the sample variance in the usual sense!

Note that with N data we have only $N - 1$ principal components, since one degree of freedom is used to compute the sample mean (for projecting everything on the mean)!

Functional Principal components are related with the eigen-decomposition of the functional counterpart of the (sample) co-variance function.

The sample co-variance operator is defined as:

$$S_x = \frac{1}{N}\sum_{i=1}^{N} \langle X_i, x\rangle X_i \quad \text{for } x \in H$$

In $H = L^2$ this is equivalently defined as:

$$[S_x](t) = \int_T \hat{c}(s,t)x(s)d(s) \text{ for } x \in L^2 \text{ with } \hat{c}(s,t) = \frac{1}{N}\sum_{i=1}^{N} X(s)X(t)$$

Note that if data are centred on the sample mean we can divide by $N-1$ for unbiased-ness!

So if S is the sample co-variance operator of $X_1, \ldots, X_N$ then the function principal components $\xi_1, \ldots, \xi_N$ are found as the eigen-functions of the operator S.

Thus they solve the eigen-equations: $S\xi_k = \lambda_k \xi_k$ where:

- The eigenvalue λ_k associated with the eigen-vector ξ_k represents the variability along the direction ξ_k
- The functional *scores* x_{ik} are the projection of the observation X_i along the direction ξ_k, that is: $x_{ik} =< X_i, \xi_k >$

We look as usual at an elbow in the cumulative percentage of total variance, in the scree plot!

So in this case the loadings ξ_i are functions: this is hard!

We can make box-plots of the scores along the first p directions to investigate the possible presence (and influence) of the outliers!

So we can plot the mean, plus or minus, the eigen-functions multiplied by a proper constant, such as the standard deviation along the component: $\bar{X} \pm \sqrt{\lambda_k}\xi_k$

This way we see how does this function look when we move one standard deviation above the mean in that direction, indicated by ξ_k

This way we infer the properties of the eigen-functions!

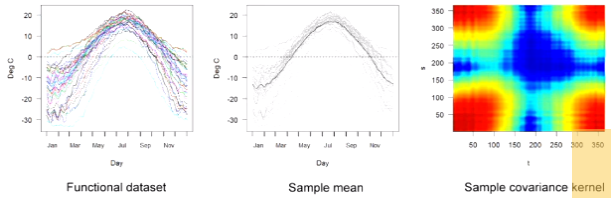
We can also plot the projection of the data set along each component: $\bar{X} + x_{ik}\xi_k$ Or we can do it along the first p components!

So: $\bar{X} + \sum_{k=1}^{p} x_{ik}\xi_k$ so that we see for which p we can successfully reconstruct our data so that we can then reduce the dimensionality of the data set!

Recall that the problem of **FPCA** can be see as the one of finding the space of dimension k that best approximates the data in the mean square sense.

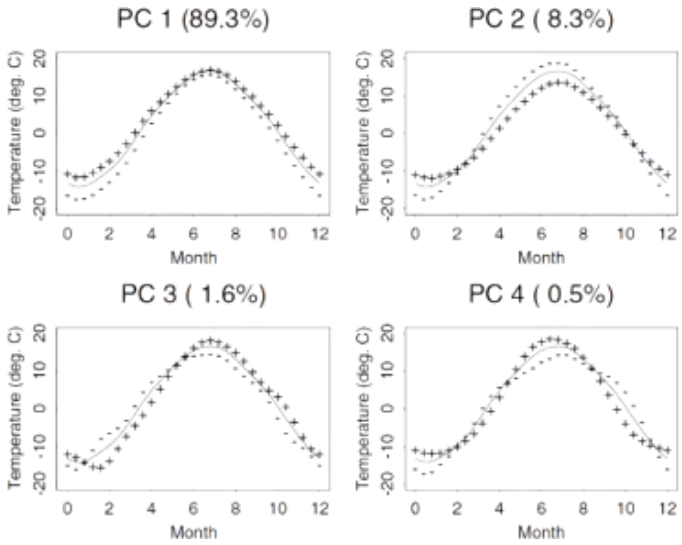
If $k = 0$ then this space is the zero-dimensional space identified by the sample mean!

Consider the following figure:



The kernel of the co-variance operator is on the right. Now suppose we compute the eigen-functions of the operator generated by that kernel.

In the following figure we see an interpretation of the principal components:



Where:

- In the top left we try to capture the meaning of the first principal component that is explaining 89% of the variability of the data set.

  What is this principal component doing?

  - If we take the mean and we move one standard deviation in the direction of the first principal component, what we get is the line identified by the pluses!

    We get a statistical unit that has the same shape that of the mean, expect that it is higher: we are increasing the average temperature and we are warmer moreover across the entire year, especially in the winter months!
  - If we take the mean and we move one standard deviation in the direction, opposite of the first principal component, what we get is the line identified by the minuses!

    We get a statistical unit that has the same shape that of the mean, expect that it is lower: we are decreasing the average temperature and we are warmer moreover across the entire year, especially in the winter months!

  Therefore we can say that the first principal component is discriminating between cold places and warm places!

- Consider now the top right where we try to capture the meaning of the second principal component that is explaining 8% of the variability of the data set.

  What is this principal component doing?

  - If we take the mean and we move one standard deviation in the direction of the second principal component, what we get is the line identified by the pluses!

    We see that it's a contrast between places where we have warmer than the mean winters, but cooler summers than the average!

So a positive score on the second principal component indicates a place that is more continental: more stable temperature along the year!

– If we take the mean and we move one standard deviation in the opposite direction of the second principal component, what we get is the line identified by the minuses!

  We see that it's a contrast between places where we have warmer than the mean summers, but cooler winters than the average!

  So a negative score on the second principal component indicates a place that is more continental: more stable temperature along the year!

29 Lecture 45: 4th Of June 2020

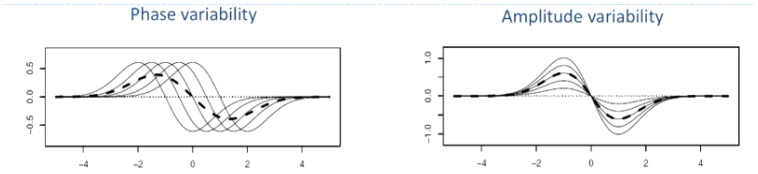
Data Alignment and Clustering in Functional Data:
Note that alignment is a specific issue with functional data: we have common patterns, but the shape, or some features, of the functions are not happening at the same time!

Functions follows a similar course but they are not aligned! This phenomenon is called phase variability and amplitude variability!

Phase variability: different curves exhibit more or less the same features but these features occur at different times, or space locations, for different statistical unit!

If not taken properly into account the misalignment acts as confounding factor and may blur subsequent analysis!

Amplitude variability: this is the variability of the phenomenon along the $y-$axis!



Given n curves $c_1(t), \dots, c_n(t)$ we want to find suitable warping functions $h_1(t), \dots, h_n(t)$ such that: $c_1(h_1(t)), \dots, c_n(h_n(t))$ are most similar!

The functions h_i should be increasing: they capture the phase variability! Then amplitude variability is the remaining variability among the aligned curves in the vertical direction!

In some cases time or location is merely shifted from curve to curve, for example because the measurements are started at random time points!

For these situations it is natural to use $h_i(t) = t + \delta_i$ while in other situations phase variations is a matter of dilation in which the natural choice of warping function would be $h_i(t) = \alpha_i t$

Yet in other situations the time, or space, deformation is more complex!

Registration (alignment) of a set of function: we want to find suitable warping functions $h_1(t), \dots, h_n(t)$ such that: $c_1(h_1(t)), \dots, c_n(h_n(t))$ are most similar!

We have two approaches, that can be mixed up:

- Landmark Alignment: for each function we find specific land-mark, present in every function of the sample.

  This can be the most prominent peak, a valley, a zero, an inflection point and so on!

  We want then to align these land-marks so that they occur at the same abscissa points!

  Land-marks are in general significant, uni-vocally identifiable, shape-events, in a curve.

  We have n curves $c_i : [0, T] \to \mathbb{R}^d$ We suppose that:

  - We have L land-marks, for the $i-$th curve are located at $t_{i1}, \dots, t_{1L}$
  - We have a template curve c_0 with landmark locations $t_{01}, \dots, t_{0L}$ If we don't have it we can define t_{0j} as the average of the t_{ij}

Warping function for the $i-$th curve is any strictly increasing function h_i such that:

- $h_i(0) = 0$
- $h_i(t_{0j}) = t_{ij}\ \forall j = 1, \ldots, L$
- $h_i(T) = T$

As notation we say that the warping functions will be the inverse of these functions h_i!

Then we are left with points: $(0,0), (t_{01}, t_{i1}), \ldots, (t_{1L}, t_{iL}), (T,T)$ which are interpolated by a piece-wise line, by a polygon or by any higher order monotone, strictly increasing, spline!

This is a difficult approach as it may require significant user input and can be sensitive to the accuracy of the land-mark identification!

In some applications is not possible to identify well-defined features that can be taken as landmarks!

Thus we may need for registration purposes to compute the derivatives of the function but if we don't perform data smoothing the derivatives are just noisy!

- Continuous Approach: we define a measure of (dis-)similarity between curves, that are aligned in order to maximise (or minimise) their (dis-)similarity!

  We choose the optimal warping function in some class of admissible warping functions in order to minimise the final distance among the curve, or equivalently in order to maximise their final similarity!

  The problem of decoupling amplitude and phase variability is not uni-vocally defined since different measures of distance, or similarity, between curves can be considered, as well as different classes of admissible warping functions!

  Indeed we can have translations, dilation, increasing linear transformations or more complex increasing transformations!

  All of this leads to different registration results!

  The choice of the couple formed by (dis-)similarity measure and admissible warping functions defines the distinction between phase variability and amplitude variability in the specific problem under analysis!

  Therefore this choice must thus be problem specific.

Decoupling phase and amplitude variabilities:
(ρ, W) must satisfy properties that ensure that the alignment problem is well-posed and the corresponding procedure is coherent!

W is the group of warping functions and ρ is the (dis-)similarity measure!

ρ must be bounded, reflexive, symmetric and transitive!

W must be a convex vector space with a group structure with respect to function composition!

(ρ, W) must satisfy properties of coherence:

- W-invariance of the index: $\rho(c_1, c_2) = \rho(c_1 \circ h, c_2 \circ h)\ \forall h \in W$

This means that with the same warping function h we cannot change the (dis-)similarity between two curves!

- Aligning one curve to another is the same as aligning the other curve to the first one, namely:

$$\rho(c_1 \circ h_1, c_2 \circ h_2) = \rho(c_1 \circ h_1 \circ h_2^{-1}, c_2) = \rho(c_1, c_2 \circ h_2 \circ h_1^{-1})$$

Note that in order to have the inverse defined W must have a group structure, as indeed requested above!

Thus we have that (ρ, W) defines, on the considered set of functions C, a partition in equivalence classes!

In the following table we see couples of dissimilarity and functions that are coherent:

dissimilarity d	warpings W
$\|\|c_1 - c_2\|\|$	W_{shift}
$\|\|c_1' - c_2'\|\|$	W_{shift}
$\|\|(c_1 - \bar{c_1}) - (c_2 - \bar{c_2})\|\|$	W_{shift}
$\left\|\left\|(c_1' - \bar{c_1'}) - (c_2' - \bar{c_2'})\right\|\right\|$	W_{shift}
$\left\|\left\|\frac{c_1}{\|c_1\|} - \frac{c_2}{\|c_2\|}\right\|\right\|$	$W_{affinity}$
$\left\|\left\|\frac{c_1'}{\|c_1'\|} - \frac{c_2'}{\|c_2'\|}\right\|\right\|$	$W_{affinity}$
$\left\|\left\|\text{sign}(c_1')\sqrt{\|c_1'\|} - \text{sign}(c_2')\sqrt{\|c_2'\|}\right\|\right\|$	$W_{diffeomorphism}$

If a template (prototype) curve ϕ is known, then it's enough to align each curve to this template!

If the template is unknown then it must be estimated from the data, leading to a complex optimisation problem, which is the following:

$$\text{Find } \phi \in C \text{ and } \underline{h} = \{h_1, \ldots, h_N\} \subset W \text{ such that } \frac{1}{N}\sum_{i=1}^{N} \rho(\phi, c_i \circ h_i) \geq \frac{1}{N}\sum_{i=1}^{N} \rho(\psi, c_i \circ g_i)$$

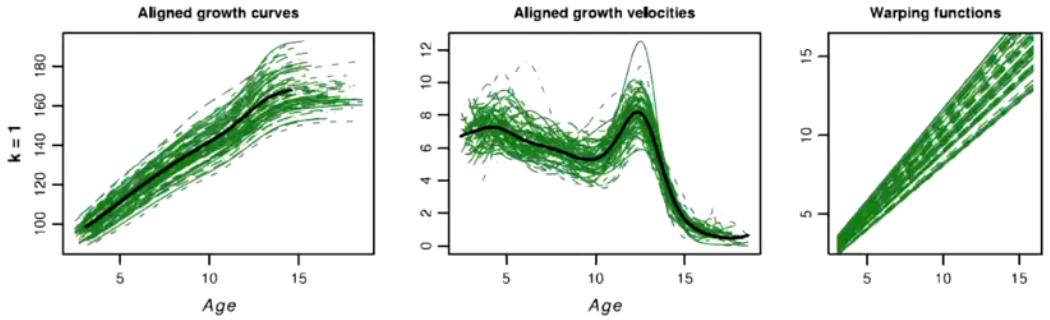
$$\text{for any other } \psi \in C \text{ and } \underline{g} = \{g_1, \ldots, g_N\} \subset W$$

We can solve this problem with the **Iterative Procrustes Procedure** which alternates:

- *Template Estimation Step:* the template centre-line is estimated from the curves obtained in the previous alignment step!
- *Alignment Step:* the centre-lines are aligned to the template centre-line, estimated in the previous template estimation step!

EM Algorithm, K-means and many others follow this Iterative paradigm!

If we try to align the Berkeley Growth Study Data we get the following result:



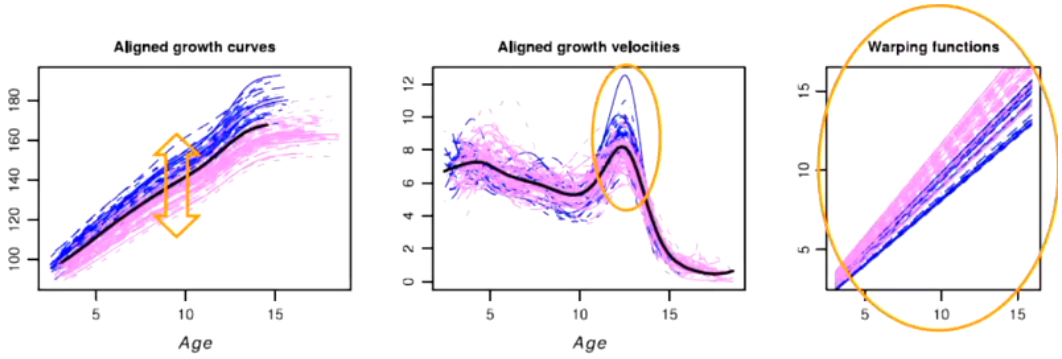
where the index of similarity used is the cosine of the angle between the derivatives of the growth curves, so this is exactly correlation, and it's defined as:

$$\rho(c_i, c_j) = \int_{S_{ij}} c_i'(s)c_j'(s)ds \frac{1}{\sqrt{\int_{S_{ij}} c_i'(s)^2 ds}\sqrt{\int_{S_{ij}} c_j'(s)^2 ds}}$$

Note that $\rho(c_i, c_j) = 1 \iff \exists a \in \mathbb{R}, b \in \mathbb{R}^+$ such that $c_i(t) = a + bc_j(t)$

Moreover $W = \{h : h(t) = mt + q \text{ with } m \in \mathbb{R}^+, q \in \mathbb{R}\}$ This means that we allow linear transformations, with positive slope!

Is this alignment pointing out some differences in the growth between boys and girls? Identifying with pink girls and with blue boys we get:



Where:

- We see a neat separation between the warping functions: girls have higher and steeper warping functions, meaning that the biological clock of girls is running faster and it starts before!
- Once the biological clocks are aligned, looking at the first picture, we see something obvious if we think of men, less so if we think of children.

  Namely we see that the height of boys stochastic-ally dominates the one of girls for any registered biological age!
- Moreover looking at the figure in the centre we see that boys have more pronounced growth during puberty, as they have a more prominent growth velocity peak!

Therefore alignment is finding something that is biologically sustained, and this is comforting!

K-mean alignment: it's the intersection between continuous alignment and functional $k-$mean clustering!

We can use the package *fdakma* in R!

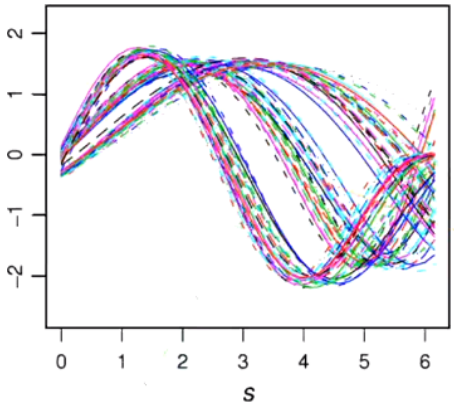
If W is the identity we get $k-$means clustering and if $K = 1$ we get continuous alignment!

Now:

- The goal of alignment is decoupling phase and amplitude variability!
- The goal of $k-$means clustering is decoupling within and between-cluster amplitude variability

Thus: the goal of $k-$means alignment is to identify phase variability, within-cluster amplitude variability and between-cluster amplitude variability!

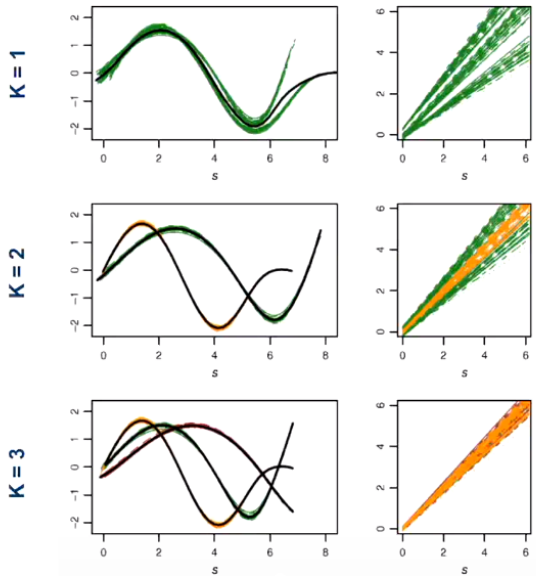
Consider the following synthetic data:



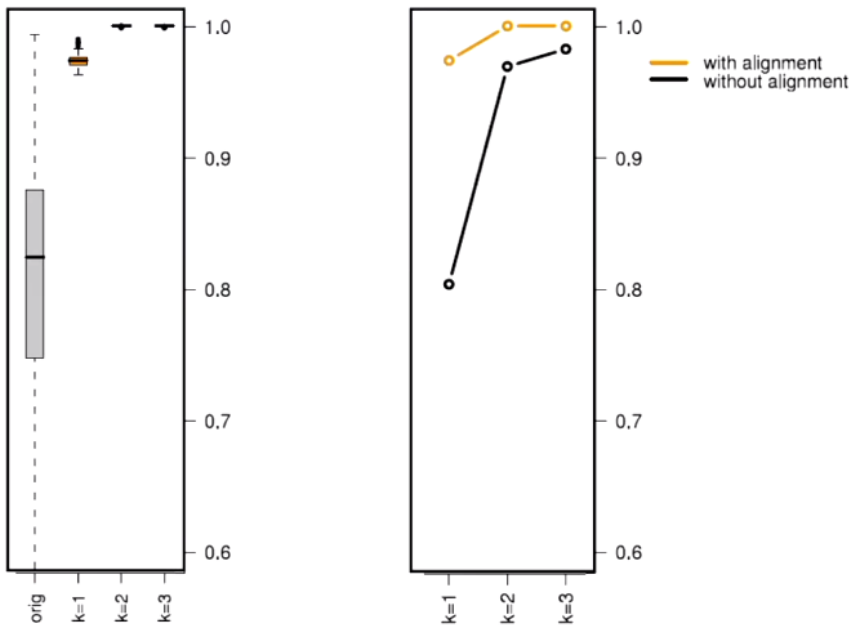
Qualitatively we see three clusters, but there are only two clusters in fact the data was synthetically generated as follows:

- The first batch of curves is generated from one template adding a white noise to the coefficients!
- The second batch of curves is generated from a second template adding a white noise to the coefficients!
- The third batch of curves is obtained by time-warping the first batch of curves!

Applying $k-$means with alignment we get the following result, for different k:



We then plot scree plot and see for an elbow to choose k, both with alignment, and without alignment:



On the left we see the (dis-)similarity box-plot for the curves without doing anything, with $k-$means alignment for $k = 1, 2, 3$

We see that the higher k the more similar the distributions!

On the right we see the medians of the distribution with and without alignment:

- We see for $k = 2$ a clear elbow in the case of alignment!
- We see for $k = 2$ a huge elbow in the case of no-alignment!

However with no-alignment we never reach the optimality we reach with alignment, that is: the black curve is always below the yellow one!

Thus by this we conclude $k = 2$ and that the third group is obtained only by warping, indeed the data was generated like that!