Theory

- Fundamental Ideas of Statistical Learning
- Multivariate variability
- Principal Component Analysis
- Gaussian Model
- Estimator of the mean and variance of a Gaussian
- Inference for the mean
- Inference for linear combination of $\underline{\mu}$
- Large scale hypothesis testing and FDR
- Comparing means of multivariate Gaussian Distr.
- Extension to two-way ANOVA.
- Classification
- Evaluating a classifier
- Regression
- Ensemble methods

Latex new feature: $\perp\!\!\!\perp$

Recall orthogonal projection

Remember:

$$||\underline{x}|| = \sqrt{\sum_i u_i^2} = \underline{u}' \cdot \underline{u} =< u, u > \quad \Rightarrow \quad cos\theta = \frac{\underline{u}' \cdot \underline{v}}{||\underline{u}|| \cdot ||\underline{v}||} = \frac{< \underline{u}, \underline{v} >}{||\underline{u}|| \cdot ||\underline{v}||}$$

From this you can derive the orthogonal projection of $\underline{u}$ on $\underline{v}$ as:

$$\pi_{u|v} = \frac{\overbrace{\underline{v} \cdot \underline{v}'}^{\in \mathbb{R}^{n \times n}}}{\underbrace{\underline{v}' \cdot \underline{v}}_{\in \mathbb{R}}} \cdot \underline{u}$$

Operator $T = \frac{\underline{v} \cdot \underline{v}'}{\underline{v}' \cdot \underline{v}}$ has this property:

- linearity;
- symmetric (self-adjoint);
- idempotent $\leftrightarrow T = T \cdot T = T^2$

So T thanks to this property it is an #orthogonal_projection_operator.

Sample Mean

Usually we have as columns each feature and as rows each observation.
#Sample_mean of the first column $\underline{c}_1$ is the projection of it on the $\underline{1}$ (which is the subspace of $\mathbb{R}^n$ without variability).

$$\text{mean of } \underline{c}_1 = \overline{x}_1 \cdot \underline{1} = \frac{1}{n}\sum_i x_{i,1} \cdot 1 = \frac{\underline{1}' \cdot \underline{c}_1}{\underline{1}' \cdot \underline{1}} \cdot \underline{1} = \frac{\underline{1} \cdot \underline{1}'}{\underline{1}' \cdot \underline{1}} \cdot \underline{c}_1 = \pi_{\underline{c}_1|\underline{1}}$$

Remark: $\underline{c}_1$ and $\underline{1}$ are column vectors
Remark: Could be more than a vector with the same mean so we create $\underline{d}_1$

Sample variance and standard deviation

#Sample_standard_deviation = $\sqrt{s_{1,1}} = [\frac{1}{n-1}\sum_i (x_{i,1} - \overline{x}_1)^2]^{\frac{1}{2}}$
#Sample_variance = $s_{1,1} = \frac{1}{n-1}\sum_i (x_{i,1} - \overline{x}_1)^2$
From this we can create the #deviation vector = $\underline{d}_1 = \underline{c}_1 - \overline{x}_1 \cdot \underline{1}$
$\Rightarrow \quad ||d_1|| =$ how good is the approximation done by $\pi_{\underline{c}_1|\underline{1}}$ of $\underline{c}_1$ = $\sqrt{n-1} \cdot \sqrt{s_{1,1}}$
If our data follow a Gaussian distribution we know that:

- 68% of our data are in the interval $[\overline{x}_1 \pm \sqrt{s_{1,1}}]$
- 95% of our data are in the interval $[\overline{x}_1 \pm s_{1,1}]$

Otherwise, try to reach Gaussianity with:

- transformation of our data (example: doing $\log(data)$)
- using #Chebyscev_inequality: $F_k\{x_i \in [\overline{x}_1 \pm k\sqrt{s_{1,1}}]\} < 1 - \frac{1}{k^2}$; if $k = 2 \Rightarrow$ we are covering about 75%
  Interpretation: we are paying the unknown distribution with a larger interval

---

A different ways to calculate $\cos(\theta_{1,2})$:

$$\cos(\theta_{1,2}) = \frac{< d_1, d_2 >}{||d_1|| \cdot ||d_2||} = \frac{d_1' d_2}{\sqrt{n-1}\sqrt{s_{1,1}} \cdot \sqrt{n-1}\sqrt{s_{2,2}}} = \frac{\sum_i (x_{i,1} - \overline{x}_1)(x_{i,2} - \overline{x}_2)}{(n-1)\sqrt{s_{1,1} \cdot s_{2,2}}} = \frac{cov(\underline{x}_1, \underline{x}_2)}{\sqrt{s_{1,1} \cdot s_{2,2}}}$$

So we reach that:

$$\cos(\theta_{1,2}) = \frac{s_{1,2}}{\sqrt{s_{1,1} \cdot s_{2,2}}} = \rho_{1,2} = corr(\underline{x}_1, \underline{x}_2) = \text{ correlation between } \underline{x}_1 \text{ and } \underline{x}_2$$

Given two column $\underline{c}_1, \underline{c}_2$ with their respective $\overline{x}_1, \overline{x}_2, d_1, d_2$ we have two limit case:

- $\theta_{1,2} = 0$:
  $d_2 \in span\{d_1\} \Rightarrow \exists \beta : d_2 = d_1 \cdot \beta \Leftrightarrow x_{i,2} - \overline{x}_2 = \beta \cdot (x_{i,1} - \overline{x}_1) \Rightarrow$ perfect correlation
- $\theta_{1,2} = \frac{\pi}{2}$:
  $\underline{c}_1$ give us none information about $\underline{c}_2$ and viceversa $\Rightarrow \rho_{1,2} = 0 =$ zero correlation

To summarize $\mathbb{X} \in \mathbb{R}^{n \times p}$, where n is the number of observation while p is the number of features for each observation, we can create:

- $\underline{\overline{x}}$ which is the #mean_vector of each features ($\in \mathbb{R}^p$);
- #covariance_matrix $= S \in \mathbb{R}^{p \times p}$ s.t. $s_{j,k} = \frac{1}{n-1}\sum_i (x_{i,j} - \overline{x}_j)(x_{i,k} - \overline{x}_k)$
  From S we can deduce the correlation matrix $\rho \in \mathbb{R}^{p \times p}$.

Remark: standardize a variable means how far each element is from the mean in terms of standard deviation and implies that $S = \rho$

Fundamental Ideas of Statistical Learning

Goal: $\boxed{\text{explain variability of } y \text{ as a function of } \underline{x}}$

Formally: find $\hat{f} : \mathbb{R}^p \to \mathbb{R}$ s.t. $\hat{f}(\underline{x}) = \underset{f(\underline{x})}{argmin}\{\mathbb{E}[|y - f(\underline{x})|^2]\}$

Solution: $\pi_{Y|\sigma(\underline{x})} = \hat{f}(\underline{x}) = \mathbb{E}[y|\underline{x}]$ which is called #regression_function

Rmk - 1: it is the Radon-Nykodyn derivative, because it allows to describe the distribution of y using distribution of features.

Proof:

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

But we notice that:

$$\begin{aligned}2\mathbb{E}[(y - \mathbb{E}[y|\underline{x}])(\mathbb{E}[y|\underline{x}] - f(\underline{x}))] &= 2\mathbb{E}[\mathbb{E}[(y - \mathbb{E}[y|\underline{x}])(\mathbb{E}[y|\underline{x}] - f(\underline{x}))|\underline{x}]] = \\ &= 2\mathbb{E}[(\mathbb{E}[y|\underline{x}] - f(\underline{x})) \cdot \underbrace{\mathbb{E}[(y - \mathbb{E}[y|\underline{x}])|\underline{x}]}_{\mathbb{E}[y|\underline{x}] - \mathbb{E}[y|\underline{x}] = 0}] = 0\end{aligned}$$

Since we remain with:
$\mathbb{E}[|y - \mathbb{E}[y|\underline{x}]|^2] + \mathbb{E}[|\mathbb{E}[y|\underline{x}] - f(\underline{x})|^2]$
to minimize it we need that $f(\underline{x}) = \mathbb{E}[y|\underline{x}]$
□

Since y is a random variable we know that our general model is:
$y = \mathbb{E}[y|\underline{x}] + \epsilon \Rightarrow \mathbb{E}[y] = \mathbb{E}[\mathbb{E}[y|\underline{x}]] + \mathbb{E}[\epsilon] = \mathbb{E}[y] + \mathbb{E}[\epsilon] \Rightarrow \mathbb{E}[\epsilon] = 0$
So we found that: $\epsilon \perp \sigma(\underline{x})$

Problem: how to build an operator $F : \mathbb{X} \to \hat{f}(\underline{x})$ s.t. given a dataset $\mathbb{X}$ it give us the best regression function.

Simple idea: check neighbourhood of the input point.

K-Nearest Neighbors - #knn

Given $x \in \mathbb{R}$ we define as $N_x = \{x_i \text{ in the training set } : \text{ close to } x\}$, $dim\{N_x\} = k$
From that we can estimate $\hat{f}(x) = \frac{1}{k} \sum_{x_i \in N_x} f(x_i)$

Curse of dimensionality

This method isn't always good since it suffer from the problem called #curse_of_dimensionality.
Explanatory example: find r to navigate 10% of an area centered in $\underline{0}$ with radius 1.

p = 1:

$\text{We are working on } \mathbb{R} \Rightarrow \frac{2 \cdot r}{2} = 0,1 \to r = 0,1$

p = 2:

We are working on $\mathbb{R}^2 \Rightarrow \frac{\pi \cdot r^2}{\pi} = 0,1 \rightarrow r = 0,31$

p = 3:

We are working on $\mathbb{R}^3 \Rightarrow \frac{\frac{4}{3}\pi \cdot r^3}{\frac{4}{3}\pi} = 0,1 \rightarrow r = 0,46$

p = 100:

We are working on $\mathbb{R}^{100} \Rightarrow r = 0,1^{\frac{1}{100}} = 0,97$

To solve this problem we can:

- Reduce the number of features p using Principal Component Analysis #PCA ;
- Use structured model to reduce the dimension of $\hat{f}$ so it depends from a parameter $\theta \in \mathbb{R}^k$;

Bias-Variance Tradeoff

Explanatory example

Framework: linear model: $f_\theta(\underline{x}) = \beta_0 + \sum_{i \in \{1,\dots,p\}} \beta_i \cdot x_i$ with $\theta = \{\beta_i\}_{i \in \{1,\dots,p\}}$

Knowing that $y = f(\underline{x}) + \epsilon$

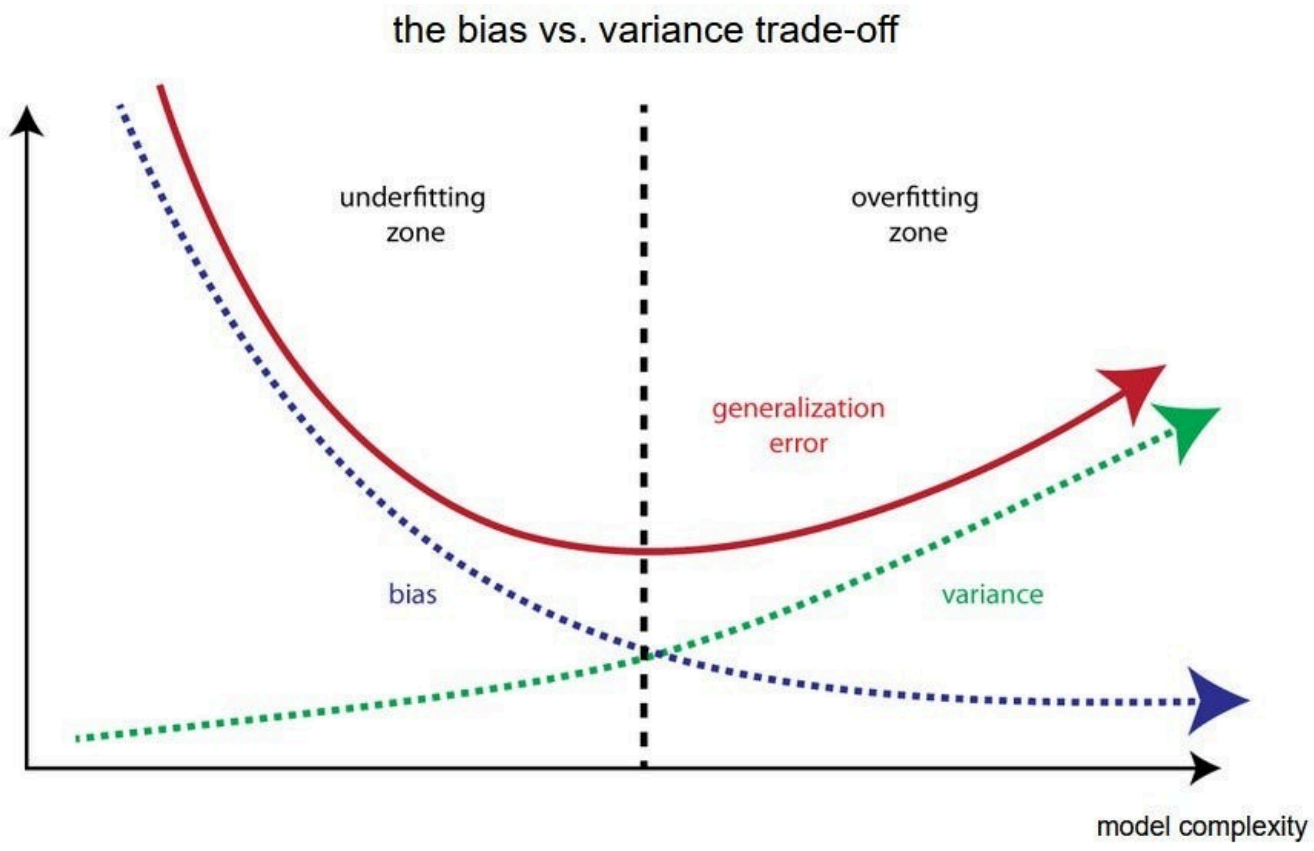
We define #generalization_error $= \mathbb{E}_{\cdot|\mathbb{X}}[(y - \hat{f}(\underline{x}))^2] = \underbrace{(f(\underline{x}) - \hat{f}(\underline{x}))^2}_{\text{reducible error}} + \underbrace{var(\epsilon)}_{\text{irreducible error}}$

There could be 2 causes if the generalization error is 0:

- phenomena is deterministic;
- test data are data from the training set, so we are not learning anything.

So we can create the #expected_generalization_error as:

$$\mathbb{E}[\mathbb{E}_{\cdot|\mathbb{X}}[(y - \hat{f}(\underline{x}))^2]] = \mathbb{E}[(f(\underline{x}) - \hat{f}(\underline{x}))^2] + var(\epsilon) = \underbrace{(f(\underline{x}) - \mathbb{E}[\hat{f}(\underline{x})])^2}_{bias^2} + \underbrace{\mathbb{E}[(\hat{f}(\underline{x}) - \mathbb{E}[\hat{f}(\underline{x})])^2]}_{var(\hat{f}(\underline{x}))} + var(\epsilon)$$



Recall

We work with $\underline{x}_1, \dots, \underline{x}_n \in \mathbb{R}^p$ iid $\sim \underline{x}$ with $\mathbb{E}[\underline{x}] = \underline{\mu}$ and $cov(\underline{x}) = \Sigma$

We have the following sample estimator:

- $\underline{\mu} \to \frac{1}{n}\sum_i^n \underline{x}_i = \overline{\underline{x}}$
- $\Sigma \to \frac{1}{n}\sum_i^n (\underline{x}_i - \overline{\underline{x}})(\underline{x}_i - \overline{\underline{x}})' = S_n$

Prop - properties of sample mean and sample variance

- $\mathbb{E}[\overline{\underline{x}}] = \underline{\mu} \leftarrow$ unbiased
- $cov(\overline{\underline{x}}) = \frac{\Sigma}{n}$
- $\mathbb{E}[S_n] = \frac{n}{n-1}\Sigma \Rightarrow \mathbb{E}[\frac{n-1}{n}S_n] = \Sigma \leftarrow$ we need $n-1$ because S_n must be orthogonal to $span\{1\}$ so the linear subspace of means.

Proof - previous Prop

First point:

$$\mathbb{E}[\overline{\underline{x}}] = \mathbb{E}\begin{bmatrix}\overline{x}_1 \\ \vdots \\ \overline{x}_p\end{bmatrix} = \begin{bmatrix}\mathbb{E}[\overline{x}_1] \\ \vdots \\ \mathbb{E}[\overline{x}_p]\end{bmatrix} \overset{iid}{\Longrightarrow} \forall k \in \{1, \dots, p\} \quad \mathbb{E}[\overline{x}_k] = \frac{1}{n}\sum_i^n \mathbb{E}[x_{i,k}] = \mu_k$$

Second one:

$$\begin{aligned} cov(\overline{\underline{x}}) =& \mathbb{E}[(\overline{\underline{x}} - \underline{\mu})(\overline{\underline{x}} - \underline{\mu})'] = \\ =& \mathbb{E}[\{\frac{1}{n}\sum_i^n (\underline{x}_i - \underline{\mu})\}\{\frac{1}{n}\sum_j^n (\underline{x}_j - \underline{\mu})\}] \\ =& \frac{1}{n^2}\sum_{i,j}\mathbb{E}[(\underline{x}_i - \underline{\mu})(\underline{x}_j - \underline{\mu})] = \begin{cases} 0 & i \neq j \leftarrow \text{ independent} \\ \sigma & i = j \leftarrow \text{identically distribuited} \end{cases} \\ =& \frac{1}{n^2} \cdot n\Sigma = \frac{\Sigma}{n} \end{aligned}$$

□

Take $\underline{c}_i$ as a column of $\mathbb{X}$, we find that #deviation $= \underline{d}_j = \underline{c}_j - \overline{x}_j\underline{1} = \underline{c}_j - \frac{\underline{1}\underline{1}'}{\underline{1}'\underline{1}}\underline{c}_j = [I - \frac{\underline{1}\underline{1}'}{\underline{1}'\underline{1}}]\underline{c}_j$

Rmk: With $[I - \frac{\underline{1}\underline{1}'}{\underline{1}'\underline{1}}]$ be the #orthogonal_projection_operator on $\mathcal{L}^{\perp}(\underline{1})$

We can define the #deviation_matrix $= d = [\underline{d}_1, \dots, \underline{d}_p] \in \mathbb{R}^{n \times p} \quad \to \quad d = [I - \frac{\underline{1}\underline{1}'}{\underline{1}'\underline{1}}]\mathbb{X}$

$$\Rightarrow S = \frac{1}{n-1}\begin{bmatrix}\underline{d}_1'\underline{d}_1 & \underline{d}_1'\underline{d}_2 & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \underline{d}_p'\underline{d}_p\end{bmatrix} = \frac{1}{n-1}d'd = \frac{1}{n-1}\mathbb{X}'[I - \frac{\underline{1}\underline{1}'}{\underline{1}'\underline{1}}]'[I - \frac{\underline{1}\underline{1}'}{\underline{1}'\underline{1}}]\mathbb{X}$$

But since we know that the orthogonal projection operator is symmetric and idempotent we reach:

$$S = \frac{1}{n-1}\mathbb{X}'[I - \frac{\underline{1}\underline{1}'}{\underline{1}'\underline{1}}]\mathbb{X}$$

Multivariate variability

We define:

- #generalized_variance $= det(S)$;
- #total_variance $= trace(S)$

Geometric interpretation - p = 2

$$S = \frac{1}{n-1}\begin{bmatrix} \underline{d}_1'\underline{d}_1 & \underline{d}_1'\underline{d}_2 \\ \underline{d}_2'\underline{d}_1 & \underline{d}_2'\underline{d}_2 \end{bmatrix} \Rightarrow det(S) = \frac{||\underline{d}_1||^2||\underline{d}_2||^2 sin^2\theta_{1,2}}{(n-1)^2}$$

So:

- $det(S)$: catch the squared area of the parallelogram with as sides $\underline{d}_1$ and $\underline{d}_2$.
- $trace(S)$: catch the perimeter of the parallelogram with as sides $\underline{d}_1$ and $\underline{d}_2$, so don't catch changes of $\theta_{1,2}$

Prop

$det(S) = 0 \Leftrightarrow \underline{d}_1, \dots, \underline{d}_p$ are linearly dependent

Dim - previous Prop

$\Leftarrow$:

$$\exists \underline{c} \neq \underline{0} \quad \text{s.t.} \quad c_1\underline{d}_1 + \dots + c_p\underline{d}_p = dc = 0$$

$$\text{Then: } S = \frac{1}{n-1} d' \underbrace{dc}_{=0} = 0 \Rightarrow det(S) = 0$$

$\Rightarrow$:

$$det(S) = 0 \Rightarrow \exists \underline{c} \neq \underline{0} : S\underline{c} = 0$$

$$\Rightarrow \underline{c}' S \underline{c} = \frac{1}{n-1}\underline{c}' d' d \underline{c} = 0$$

$$\Rightarrow ||d\underline{c}||^2 = 0$$

$$\Rightarrow d\underline{c} = 0$$

$$\Rightarrow \underline{d}_1, \dots, \underline{d}_p \text{ are linearly dependent}$$

□

Corollary

If $p \geq n \Rightarrow det(S) = 0$

Interpretation: more features to analyze than data available

Proof - previous Corollary

Take $\underline{d}_1, \dots, \underline{d}_p$ with $p \geq n$, with $\underline{d}_i \in \mathbb{R}^n$ and $\underline{d}_i \in \mathcal{L}^{\perp}(\underline{1})$ so we have that $dim(\mathcal{L}^{\perp}) \leq n-1$

Since $p \geq n$ we have more vectors than degree of freedom $\Rightarrow$ some are linearly dependent

$\Rightarrow det(S) = 0$

□

Prop

$$\begin{cases} \text{S is positive semi-defined} \\ det(S) \neq 0 \end{cases} \Rightarrow S \text{ is positive defined}$$

Proof - previous Prop

Since it is positive semi-defined we know that:

- $\forall \underline{c} \in \mathbb{R}^p \quad \underline{c}' S \underline{c} = \frac{1}{n-1} \underline{c}' d' d \underline{c} = \frac{1}{n-1} ||d\underline{c}'||^2 \geq 0$;
- $\forall i \quad \lambda_i \geq 0$;

By contradiction:
From hypothesis we know that $det(S) \neq 0$ if:
$\exists \underline{c} \neq \underline{0} : \underline{c}' S \underline{c} \Rightarrow ||d\underline{c}||^2 = 0 \Rightarrow d\underline{c} = 0 \Rightarrow \underline{d}_1, \dots, \underline{d}_p$ lin. dep. $\Rightarrow det(S) = 0$ (contraddiction)
Since $\nexists \underline{c} : \underline{c}' S \underline{c} = 0$ then S is positive defined.
□
Since S is positive semi-defined so $\forall i \quad \lambda_i \geq 0 \Rightarrow S = P \Lambda P'$
With:

- $P =$ eigenvectors matrix ($\underline{e}_i$ is the i^{th} eigenvector);
- $\Lambda =$ ordered eigenvalues matrix so $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$

Rmk: if $det(S) \neq 0 \Rightarrow S^{-1} = \sum_i \frac{1}{\lambda_i} \underline{e}_i \underline{e}_i'$

Def - #Mahalanobis_distance

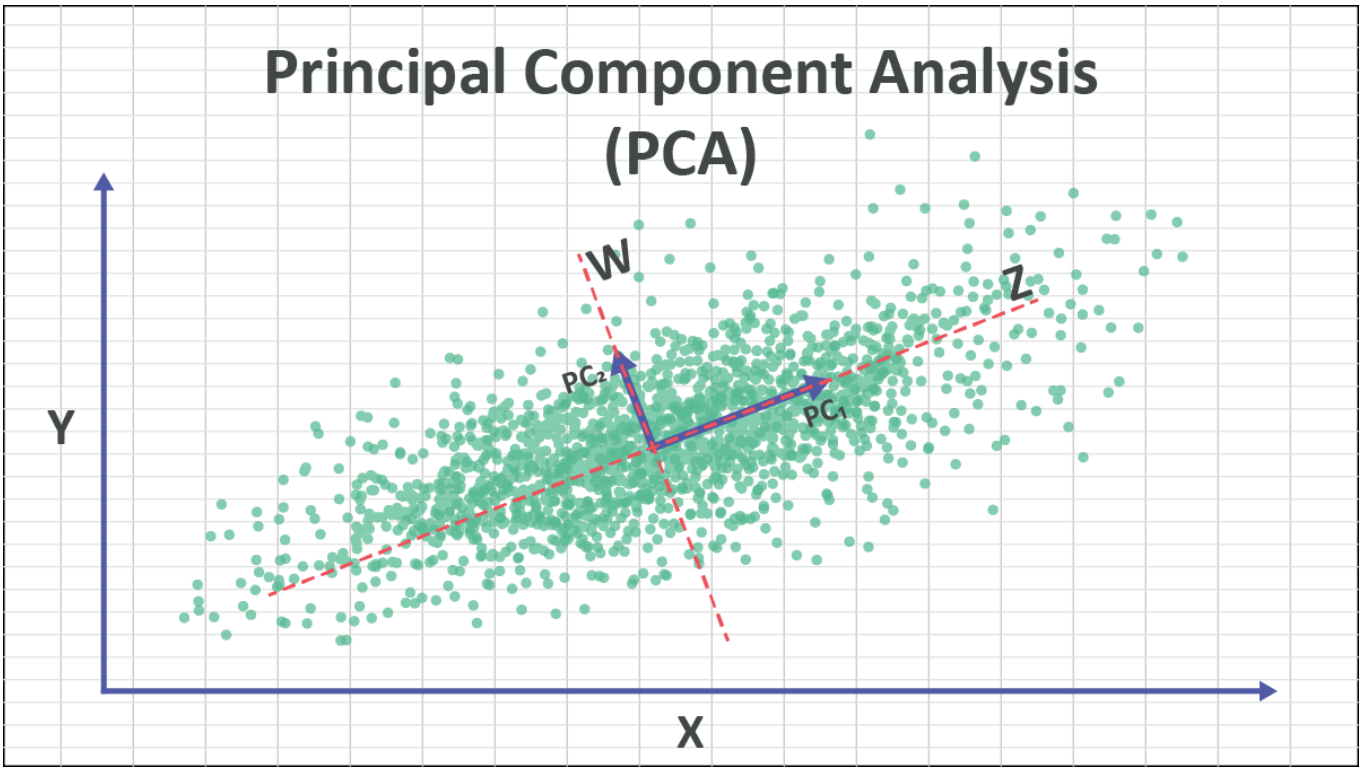
Mahalanobis distance $= \underline{d}^2_{S^{-1}}(\underline{x}, \underline{y}) = (\underline{x} - \underline{y})' S^{-1} (\underline{x} - \underline{y})$
Interpretation: $\underline{d}^2_{S^{-1}}(\underline{x}, \underline{y}) = 3$ means that $\underline{x}$ and $\underline{y}$ are 3 standard deviation apart w.r.t. their joint distribution.
We call $\epsilon_r(\overline{\underline{x}}) =$ sphere centered in $\overline{\underline{x}}$ of radius r w.r.t. Mahalanobis distance
In $\mathbb{R}^n$ this is an ellipse centered in $\overline{\underline{x}}$ with axis in the same direction of $\underline{e}_i$ and semi-axis equal to $\alpha \frac{1}{\sqrt{\lambda_i}} r$ with $(\lambda_i, \underline{e}_i)$ eigenvalues and eigenvectors of S.
$\Rightarrow$ volume of this ellipse $= \alpha \cdot \sqrt{\lambda}_1 \cdot \sqrt{\lambda}_2 \cdot \dots \cdot \sqrt{\lambda}_p \cdot r^p = \sqrt{det(S)} \cdot r^p$

Principal Component Analysis

Intuition behind Principal Component Analisys #PCA

We can always decompose S with the spectral theorem, so it can be written as $S = P \Lambda P'$.
Given $\mathbb{X}$ we can change the coordinates of each vector composing our dataset using P so:
$\forall \underline{x} \in \mathbb{X} \quad \underline{x} \rightarrow \tilde{\underline{x}} = P' \underline{x} \in \tilde{\mathbb{X}}$
After this transformation we need to re-calculate the #covariance_matrix $\tilde{S}$ and we find that:
$\tilde{S} = \frac{1}{n-1} \tilde{\mathbb{X}}' [I - \frac{1 1'}{1' 1}] \tilde{\mathbb{X}} = P' S P = P' P \Lambda P' P = \Lambda$

> **Important:**
> Using as axis the eigenvector of the covariance matrix, covariances between features become 0.
> Idea: covariances depends only by the reference system used.



We call as **Principal Component** the linear subspace which maximize the variance of our data.
Given a random variable $\underline{x} \in \mathbb{R}^p$ s.t. $\underline{x} \sim \underline{\mu}, \Sigma$ we want p vectors $\underline{a}_1, \dots, \underline{a}_p$ such that:

1. Find $\underline{a}_1 \in \mathbb{R}^p$ with $||\underline{a}_1|| = 1$ s.t. $Var(\underline{a}_1' \underline{x})$ is maximized;
2. Find $\underline{a}_2 \in \mathbb{R}^p$ with $||\underline{a}_2|| = 1$ s.t. $Var(\underline{a}_2' \underline{x})$ is maximized and $cov(\underline{a}_1 \underline{x}, \underline{a}_2 \underline{x}) = 0$;
3. Find $\underline{a}_3 \in \mathbb{R}^p$ with $||\underline{a}_3|| = 1$ s.t. $Var(\underline{a}_3' \underline{x})$ is maximized and: $cov(\underline{a}_1 \underline{x}, \underline{a}_3 \underline{x}) = 0$, $cov(\underline{a}_2 \underline{x}, \underline{a}_3 \underline{x}) = 0$;
4. ...;

   Recall: $var(\mathbb{X}) = S$ so $var(\underline{a}' \mathbb{X}) = \underline{a}' S \underline{a}$ and $\underline{x}' \underline{x} = ||x||$

Lemma

Take the matrix $B \in p \times p$ symmetric and positive defined s.t. $B = \sum_i^p \lambda_i \underline{e}_i \underline{e}_i'$ we know that:

- $\max_{\underline{x} \in \mathbb{R}^p} \frac{\underline{x}' B \underline{x}}{\underline{x}' \underline{x}} = \lambda_1$ and $\arg\max_{\underline{x} \in \mathbb{R}^p} \frac{\underline{x}' B \underline{x}}{\underline{x}' \underline{x}} = \underline{e}_1$;
- $\max_{\underline{x} \in \mathbb{R}^p, \underline{x} \perp \underline{e}_1} \frac{\underline{x}' B \underline{x}}{\underline{x}' \underline{x}} = \lambda_2$ and $\arg\max_{\underline{x} \in \mathbb{R}^p, \underline{x} \perp \underline{e}_1} \frac{\underline{x}' B \underline{x}}{\underline{x}' \underline{x}} = \underline{e}_2$
- ... so on ...

Proof - previous Lemma

First principal component:
Taking $\underline{x} \neq 0$:

$$\frac{\underline{x}' B \underline{x}}{\underline{x}' \underline{x}} = \frac{\underline{x}' P \Lambda P' \underline{x}}{\underline{x}' \underline{x}} = \frac{\underline{x}' P \Lambda P' \underline{x}}{\underline{x}' P P' \underline{x}} \overbrace{=}^{y = P' \underline{x}} \frac{\underline{y}' \Lambda \underline{y}}{\underline{y}' \underline{y}} \leq \lambda_1 \frac{\sum_i^p y_i^2}{\sum_i^p y_i^2} = \lambda_1$$

If $\underline{x} = \underline{e}_1 \rightarrow \frac{\underline{x}' B \underline{x}}{\underline{x}' \underline{x}} = \lambda_1 \Rightarrow \max_{\underline{x} \in \mathbb{R}^p} \frac{\underline{x}' B \underline{x}}{\underline{x}' \underline{x}} = \lambda_1$ and $\arg\max_{\underline{x} \in \mathbb{R}^p} \frac{\underline{x}' B \underline{x}}{\underline{x}' \underline{x}} = \underline{e}_1$

Second principal component:
Taking $\underline{x} \neq 0$:

$$\frac{\underline{x}'B\underline{x}}{\underline{x}'\underline{x}} = \frac{\underline{x}'P\Lambda P'\underline{x}}{\underline{x}'\underline{x}} = \frac{\underline{x}'P\Lambda P'\underline{x}}{\underline{x}'PP'\underline{x}} \overbrace{=}^{\underline{y}=P'\underline{x}, \perp \underline{e}_1} \frac{\underline{y}'\Lambda\underline{y}}{\underline{y}'\underline{y}} \leq \lambda_2$$

So we reach that: $\max_{\underline{x}\in\mathbb{R}^p, \underline{x}\perp\underline{e}_1} \frac{\underline{x}'B\underline{x}}{\underline{x}'\underline{x}} = \lambda_2$ and $\arg\max_{\underline{x}\in\mathbb{R}^p, \underline{x}\perp\underline{e}_1} \frac{\underline{x}'B\underline{x}}{\underline{x}'\underline{x}} = \underline{e}_2$

It's curious that to satisfy $\underline{y} \perp \underline{e}_1$ we reach that $\underline{y} = P'\underline{x} = \begin{bmatrix} 0 \\ \underline{e}_2\underline{x} \\ \vdots \\ \underline{e}_p\underline{x} \end{bmatrix}$

p principal component:
Taking $\underline{x} \neq 0$:

$$\frac{\underline{x}'B\underline{x}}{\underline{x}'\underline{x}} = \frac{\underline{y}'\Lambda\underline{y}}{\underline{y}'\underline{y}} = \frac{\sum_i^p \lambda_i y_i^2}{\sum_i^p y_i^2} \geq \lambda_p \frac{\sum_i^p y_i^2}{\sum_i^p y_i^2} = \lambda_p$$

If $\underline{x} = \underline{e}_p \rightarrow \frac{\underline{x}'B\underline{x}}{\underline{x}'\underline{x}} = \lambda_p \Rightarrow \min_{\underline{x}\in\mathbb{R}^p} \frac{\underline{x}'B\underline{x}}{\underline{x}'\underline{x}} = \lambda_p$ and $\arg\min_{\underline{x}\in\mathbb{R}^p} \frac{\underline{x}'B\underline{x}}{\underline{x}'\underline{x}} = \underline{e}_p$
□

Rmk: $\underline{a} \perp \underline{b} \Rightarrow cov(\underline{a}, \underline{b}) = \underline{a}'\Sigma\underline{b} = \underline{b}'\Sigma\underline{a} = 0$
We define:

- #scores = y_i = $\underline{e}_i'\underline{x}$ = score of i^{th} principal component, it's the coordinate of the projection of $\underline{x}$ on the $\underline{e}_i$ axis;
- #loadings = $\underline{e}_i$ = vector describing the i^{th} principal component;

$(y_i, \underline{e}_i)$ is the i^{th} principal component.
It's useful to centered our data doing this:
$y_1 = \underline{e}_1'(\underline{x} - \underline{\mu})$

Corollary

$$cov(y_i, y_j) = \lambda_{i,j} \cdot \delta_{i,j} = \begin{cases} \lambda_i & i = j \\ 0 & else \end{cases}$$

Proof - previous Corollary

$$cov(y_i, y_j) = cov(\underline{e}_i'\underline{x}, \underline{e}_j'\underline{x}) = \underline{e}_i'\Sigma\underline{e}_j' = \lambda_i\underline{e}_i\underline{e}_j' = \begin{cases} \lambda_i & i = j \\ 0 & \text{else since } \underline{x}_i \perp \underline{x}_j \end{cases}$$

□

Framework:

- take $\underline{x} \in \mathbb{R}^p$ s.t. $\mathbb{E}[\underline{x}] = \mu$ and $cov(\underline{x}) = \Sigma = P\Lambda P'$ with $det(\Sigma) \neq 0$
- Projection into eigen-space = Principal component
  $= \underline{y} = P'(\underline{x} - \underline{\mu}) = \begin{bmatrix} \underline{e}_1'(\underline{x} - \underline{\mu}) & \dots & \underline{e}_p'(\underline{x} - \underline{\mu}) \end{bmatrix}'$
  - $\Rightarrow cov(\underline{y}) = \Lambda$ and $\mathbb{E}[\underline{y}] = \underline{0}$

Sometimes is useful to give the proportion of variance explained by the i^{th} principal component $y_i = \underline{e}_i(\underline{x} - \underline{\mu})$. To do so we calculate $\frac{\lambda_i}{\sum_j \lambda_j}$.

#PCA 's problem: interpretation of loading

$\forall i \quad y_i = \underline{e}_i'(\underline{x} - \underline{\mu}) = \sum_{j=1}^{p} e_{i,j}(x_j - \mu_j)$ we call j^{th} the component of the i^{th} #loadings $\underline{e}_i$ as $e_{i,j}$.

Prop

$\forall i, k = 1, \dots, p \quad corr(y_i, x_k) = e_{k,i}\frac{\sqrt{\lambda_i}}{\sqrt{\sigma_{kk}}}$

Proof - previous Prop

Recall: $corr(y_i, x_k) = \frac{cov(y_i, x_k)}{\sqrt{\lambda_i}\sqrt{\sigma_{kk}}}$

$cov(y_i, x_k) = cov(\underline{e}_i'(\underline{x} - \underline{\mu}), \underline{u}_k'\underline{x}) \overbrace{=}^{cov(\underline{e}_i'\underline{\mu}, \underline{u}_k'\underline{x}) = 0} cov(\underline{e}_i'\underline{x}, \underline{u}_k'\underline{x}) = \underline{e}_i'\Sigma u_k = \underline{u}_k'\Sigma e_i = \lambda_i \underline{u}_k'\underline{e}_i = \lambda_i e_{i,k}$

$\Rightarrow corr(\underline{y}_i, \underline{x}_k) = \frac{\lambda_i e_{i,k}}{\sqrt{\lambda_i}\sqrt{\sigma_{kk}}} = e_{i,k}\frac{\sqrt{\lambda_i}}{\sqrt{\sigma_{kk}}}$

□

So we find that $\boxed{\text{if } \sigma_{kk} \text{ are of the same order of magnitude}}$ we can interpret component of loadings as related to the correlation.

Otherwise, $\boxed{\text{if } \sigma_{kk} \text{ have different order of magnitude}}$ the first component of the PCA take the component with the biggest variance (biggest as number not as interpretation).

We define V as a diagonal matrix with as elements of the main diagonal each variance, then we call standardize vector $\underline{Z} = V^{-\frac{1}{2}}(\underline{x} - \underline{\mu})$. It has $\mathbb{E}[\underline{Z}] = \underline{0}$ and $cov(\underline{Z}) = \underbrace{V^{-\frac{1}{2}}\Sigma(V^{-\frac{1}{2}})'}_{\underline{\rho}}$

So we apply PCA to the covariance/correlation matrix $\underline{\rho} \Rightarrow \underline{y} = \underbrace{P'}_{\text{eigenvectors of } \underline{\rho}} \underline{Z} = P'V^{-\frac{1}{2}}(\underline{x} - \underline{\mu})$

$tr(\Lambda) = tr(\underline{\rho}) = \sum_i \lambda_i = p \Rightarrow$ *rule of thumb:* select components with $\lambda_i \geq 1$

Remark: usually we estimate Σ with S so we apply PCA to eigenvectors and eigenvalues of S

A different perspective of PCA

Framework: our data are $\underline{x}_1, \dots, \underline{x}_n \in \mathbb{R}^p$

Question: find the linear subspace of dimension h in $\mathbb{R}^p$ closest to our data

$\boxed{\text{We want to find } \underline{\varphi}_1, \dots, \underline{\varphi}_k \text{ orthonormal s.t. } \mathcal{L} = span\{\underline{\varphi}_1, \dots, \underline{\varphi}_k\} \text{ is the closest to } \underline{x}_1, \dots, \underline{x}_n}$.

Algorithm's step:

1. Center our data: $\forall i \in \{1, \dots, n\} \quad \underline{x}_1 - \overline{\underline{x}}, \dots, \underline{x}_n - \overline{\underline{x}}$
2. Project $\underline{x}_1, \dots, \underline{x}_n$ on $\mathcal{L}$: $\pi_{\underline{x}_i|\mathcal{L}} = \sum_{j=1}^{k} \underline{\varphi}_j\underline{\varphi}_j'(\underline{x}_i - \overline{\underline{x}}) \quad \forall i$
3. Find $\mathcal{L}$ s.t. $min\{\sum_{i=1}^{n} ||(\underline{x}_i - \overline{\underline{x}}) - \sum_{j=1}^{k} \underline{\varphi}_j\underline{\varphi}_j'(\underline{x}_i - \overline{\underline{x}})||^2\}$

How to find $\mathcal{L}$?

$$\sum_{i=1}^{n} ||(\underline{x}_i - \overline{\underline{x}}) - \sum_{j=1}^{k} \underline{\varphi}_j \underline{\varphi}_j'(\underline{x}_i - \overline{\underline{x}})||^2 \overbrace{=}^{\underline{v}_i = \underline{x}_i - \overline{\underline{x}}} \sum_{i=1}^{n} ||\underline{v}_i - \sum_{j=1}^{k} \underline{\varphi}_j \underline{\varphi}_j' \underline{v}_i||^2$$

$$= \sum_{i=1}^{n} \underbrace{(\underline{v}_i - \sum_{j}^{k} \underline{\varphi}_j \underline{\varphi}_j' \underline{v}_i)'(\underline{v}_i - \sum_{j}^{k} \underline{\varphi}_j \underline{\varphi}_j' \underline{v}_i)}_{\text{Ⓟ}} = \sum_{i=1}^{n} \underline{v}_i \underline{v}_i' - \sum_{i=1}^{n} \sum_{j=1}^{k} \underline{v}_i' \underline{\varphi}_j \underline{\varphi}_j' \underline{v}_i$$

$$\text{Ⓟ} = \underline{v}_i' \underline{v}_i - 2 \sum_{j}^{k} \underline{v}_i' \underline{\varphi}_j \underline{\varphi}_j' \underline{v}_i + \underbrace{(\sum_{j} \underline{\varphi}_j \underline{\varphi}_j' \underline{v}_i)'(\sum_{t} \underline{\varphi}_t \underline{\varphi}_t' \underline{v}_i)}_{=0 \text{ if } j \neq t \text{ else } = \sum_j^k \underline{v}_i' \underline{\varphi}_j \underline{\varphi}_j' \underline{v}_i} = \underline{v}_i \underline{v}_i' - \sum_{j=1}^{k} \underline{v}_i' \underline{\varphi}_j \underline{\varphi}_j' \underline{v}_i$$

So we reach that:

- $\sum_i^n \underline{v}_i \underline{v}_i'$ doesn't depend from $\underline{\varphi}_j$
- $\sum_i^n \sum_{j=1}^k \underline{v}_i' \underline{\varphi}_j \underline{\varphi}_j' \underline{v}_i$ has to be maximized (since there is a minus before it)

Maximum of $\sum_i^n \sum_{j=1}^k \underline{v}_i' \underline{\varphi}_j \underline{\varphi}_j' \underline{v}_i$:

$$\sum_{i}^{n} \sum_{j=1}^{k} \underline{v}_i' \underline{\varphi}_j \underline{\varphi}_j' \underline{v}_i = \sum_{j=1}^{k} \sum_{i}^{n} \underline{\varphi}_j' \underline{v}_i \underline{v}_i' \underline{\varphi}_j = \sum_{j=1}^{k} \underline{\varphi}_j' (\sum_{i}^{n} \underline{v}_i \underline{v}_i') \underline{\varphi}_j =$$

$$= \sum_{j=1}^{k} \underline{\varphi}_j' (n-1) S \underline{\varphi}_j = (n-1) \sum_{j=1}^{k} \underline{\varphi}_j' S \underline{\varphi}_j$$

Using a previous lemma we know that to maximize: $k = 1 \quad \underline{\varphi}_j = \underline{e}_1; \quad k = 2 \quad \underline{\varphi}_j = \underline{e}_2; \dots$
So we find that $\mathcal{L} = span\{\underline{e}_1, \dots, \underline{e}_k\}$ and the approximation error is: $(n+1) \sum_{i=k+1}^{p} \lambda_i$

Gaussian Model

Framework: $\underline{\mu} \in \mathbb{R}^p; \Sigma = [\sigma_{i,j}] \in p \times p$ positive defined

Recall

$p = 1$:
$\mu \in \mathbb{R}, \sigma_{1,1} > 0 \Rightarrow x_1 \sim N_1(\mu, \sigma_{1,1})$ if the density of x_1 is $\phi(t) = \frac{1}{\sqrt{2\pi\sigma_{1,1}}} exp\{-\frac{(t-\mu)^2}{\sigma_{1,1}}\}$ with $t \in \mathbb{R}$
$p \neq 1$:
$\underline{x} \sim N_p(\mu, \Sigma) \Leftrightarrow \phi(\underline{t}) = \frac{1}{\sqrt{(2\pi)^p det(\Sigma)}} exp\{-\frac{1}{2}(\underline{t} - \underline{\mu})' \Sigma^{-1} (\underline{t} - \underline{\mu})\}$

Prop

$\underline{x} \sim N_p(\underline{\mu}, \Sigma) \Rightarrow \forall \underline{a} \in \mathbb{R}^p \quad \underline{a}' \underline{x} \sim N_1(\underline{a}' \underline{\mu}, \underline{a}' \Sigma \underline{a})$

Corollary

$\underline{x} \sim N_p(\underline{\mu}, \Sigma) \Rightarrow \forall i \in \{1, \dots, p\} \quad x_i \sim N_1(\mu_i, \sigma_{i,i})$

Proof - previous Corollary

Take $\underline{u}_i = [0,\dots,0,\overbrace{1}^{i^{th}\text{ position}},0,\dots,0]' \xrightarrow{\text{prev. prop}} \underline{u}_i\underline{x} \sim_1 (\underline{u}_i'\underline{\mu}, \underline{u}_i'\Sigma\underline{u}_i) = N_1(\mu_i, \sigma_{i,i})$

□

Corollary

$$\begin{cases}\underline{x} \sim N_p(\underline{\mu}, \Sigma) \\ A \in q \times p\end{cases} \Rightarrow A\underline{x} \sim N_q(A\underline{\mu}, A\Sigma A')$$

Proof

Thesis: $\forall \underline{a} \in \mathbb{R}^q \quad \underline{a}'(A\underline{x}) \sim N_1(\underline{a}'A\underline{\mu}, \underline{a}'A\Sigma A'\underline{a})$

$\underline{a}'(A\underline{x}) = (\underline{a}'A)\underline{x} = \overbrace{(A'\underline{a})'}^{\in\mathbb{R}^p}\underline{x} \sim N_1((A'\underline{a})'\underline{\mu}, (A'\underline{a})'\Sigma(A'\underline{a})) = N_1(\underline{a}'A\underline{\mu}, \underline{a}'A\Sigma A'\underline{a})$

From that we conclude that:

$\Rightarrow A\underline{x} \sim N_q(A\underline{\mu}, A\Sigma A')$

□

Corollary

$\underline{d} \in \mathbb{R}^p, \underline{x} \sim N(\underline{\mu}, \Sigma) \quad \Rightarrow \quad \underline{x} + \underline{d} \sim N_p(\underline{\mu} + \underline{d}, \Sigma)$

Observation

$Z_1,\dots,Z_p \text{ iid } \sim N_1(0,1) \Rightarrow \underline{Z} \sim N_p(\underline{0}, I)$

So using the last observation and the previous corollaries we reach:

$$\underline{x} \sim N_p(\underline{\mu}, \Sigma) \Rightarrow \underline{Z} = \Sigma^{-\frac{1}{2}}(\underline{x} - \underline{\mu}) \sim N_p(\underline{0}, I)$$
$$\underline{Z} \sim N_p(\underline{0}, I) \Rightarrow \underline{x} = \Sigma^{\frac{1}{2}}\underline{Z} + \underline{\mu} \sim N_p(\underline{\mu}, \Sigma)$$

Observation

Given $\underline{x} \sim N_p(\underline{\mu}, \Sigma)$ we can find the #Mahalanobis_distance W:

$$W = d^2_{\Sigma^{-1}}(\underline{x}, \underline{\mu}) = (\underline{x} - \underline{\mu})'\Sigma^{-1}(\underline{x} - \underline{\mu}) = \overbrace{(\underline{x} - \underline{\mu})'\Sigma^{-\frac{1}{2}}}^{\underline{Z}'}\overbrace{\Sigma^{-\frac{1}{2}}(\underline{x} - \underline{\mu})}^{\underline{Z}} = \underline{Z}'\underline{Z} = \sum_i^p Z_i^2 \sim \chi^2(p)$$

$\mathbb{P}[d^2_{\Sigma^{-1}}(\underline{x}, \underline{\mu}) \leq \chi^2_{1-\alpha}(p)] = 1 - \alpha$

Corollary

Take $\underline{x} \in \mathbb{R}^p$ compose as $\underline{x} = \begin{bmatrix}\underline{x}_1 \in \mathbb{R}^q \\ \underline{x}_2 \in \mathbb{R}^{p-q}\end{bmatrix} \sim N_p(\begin{bmatrix}\underline{\mu}_1 \\ \underline{\mu}_2\end{bmatrix}, \begin{bmatrix}\Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{2,1} & \Sigma_{2,2}\end{bmatrix}) \Rightarrow \underline{x}_1 \sim N_q(\underline{\mu}, \Sigma_{1,1})$

Proof -previous Corollary

Take $A = [I_{q\times q} \quad 0_{q\times(p-q)}] \in q \times p \Rightarrow A\underline{x} = \underline{x}_1 \sim N_q(\underline{\mu}_1, A\Sigma A' = \Sigma_{1,1})$

□

Prop

$$\underline{x} = \begin{bmatrix} \underline{x}_1 \in \mathbb{R}^q \\ \underline{x}_2 \in \mathbb{R}^{p-q} \end{bmatrix} \sim N_p(\begin{bmatrix} \underline{\mu}_1 \\ \underline{\mu}_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{1,1} & 0 \\ 0 & \Sigma_{2,2} \end{bmatrix}) \Rightarrow \underline{x}_1 \perp\!\!\!\perp \underline{x}_2$$

Theo

$$\underline{x} = \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \end{bmatrix} \begin{matrix} \in \mathbb{R}^q \\ \in \mathbb{R}^{p-q} \end{matrix} \sim N_p(\begin{bmatrix} \underline{\mu}_1 \\ \underline{\mu}_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{2,1} & \Sigma_{2,2} \end{bmatrix}) \Rightarrow \underline{x}_1 | \underline{x}_2 \sim N_q(\underline{\mu}_1 + \Sigma_{1,2}\Sigma_{2,2}^{-1}(\underline{x}_2 - \underline{\mu}_2), \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1})$$

Proof - previous Theo

Let be: $A = \begin{bmatrix} I_{q\times q} & -\Sigma_{1,2}\Sigma_{2,2}^{-1} \\ 0_{q\times(p-q)} & I_{(p-q)\times(p-q)} \end{bmatrix}$ then:

$$A \begin{bmatrix} \underline{x}_1 - \underline{\mu}_1 \\ \underline{x}_2 - \underline{\mu}_2 \end{bmatrix} = \begin{bmatrix} \underline{x}_1 - \underline{\mu}_1 - \Sigma_{1,2}\Sigma_{2,2}^{-1}(\underline{x}_2 - \underline{\mu}_2) \\ \underline{x}_2 - \underline{\mu}_2 \end{bmatrix} \sim N_p(\begin{bmatrix} \underline{0} \\ \underline{0} \end{bmatrix}, \begin{bmatrix} \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1} & 0 \\ 0 & \Sigma_{2,2} \end{bmatrix})$$

$$\Rightarrow \underline{x}_1 - \underline{\mu}_1 - \Sigma_{1,2}\Sigma_{2,2}^{-1}(\underline{x}_2 - \underline{\mu}_2) \perp\!\!\!\perp \underline{x}_2 - \underline{\mu}_2$$

$$\Rightarrow \underline{x}_1 - \underline{\mu}_1 - \Sigma_{1,2}\Sigma_{2,2}^{-1}(\underline{x}_2 - \underline{\mu}_2) | \underline{x}_2 - \underline{\mu}_2 \sim N_q(\underline{0}, \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1})$$

$$\Rightarrow \underline{x}_1 | \underline{x}_2 \sim N_q(\underline{\mu}_1 + \Sigma_{1,2}\Sigma_{2,2}^{-1}(\underline{x}_2 - \underline{\mu}_2), \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1})$$

□

Remark: $\Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1}$ is called partial covariance

Estimator of the mean and variance of a Gaussian

Framework: we have $\underline{x}_1, \dots, \underline{x}_n$ iid $\sim N_p(\underline{\mu}, \Sigma)$ with $\underline{\mu}$ and Σ unknown.

Recall - 1

Estimator:

- $\overline{\underline{x}} = \frac{1}{n}\sum_{i=1}^n \underline{x}_i =$ sample mean
- $S = \frac{1}{n-1}\sum_{i=1}^n (\underline{x}_i - \overline{\underline{x}})(\underline{x}_i - \overline{\underline{x}})' =$ sample Σ

Recall - 2

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

Log-Likehood $= l(\underline{\mu}, \Sigma | \underline{x}_1, \dots, \underline{x}_n) = n \cdot ln(\frac{1}{\sqrt{(2\pi)^p det(\Sigma)}}) - \frac{1}{2}\sum_i^n (\underline{x}_i - \underline{\mu})'\Sigma^{-1}(\underline{x}_i - \underline{\mu})$

Interpretation: likelihood function $\mathcal{L}(\theta | \underline{x})$ is the probability of observing parameter θ assuming $\underline{x}$ as data.

Theo

If $\underline{x}_1, \dots, \underline{x}_n$ iid $\sim N_p(\underline{\mu}, \Sigma)$

Then the Maximum Likehood Estimator (#MLE) estimator for:

- $\underline{\mu}$ is $\hat{\underline{\mu}} = \frac{1}{n}\sum_{i=1}^n \underline{x}_i$ which is unbiased;
- Σ is $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^n (\underline{x}_i - \overline{\underline{x}})(\underline{x}_i - \overline{\underline{x}})'$ which is biased

Prop

If $\underline{x}_1, \dots, \underline{x}_n$ iid $\sim N_p(\underline{\mu}, \Sigma)$
Then: $\overline{\underline{x}} \sim N_p(\underline{\mu}, \frac{1}{n}\Sigma)$

Proof - previous Prop

$$\underline{\tilde{x}} = \begin{bmatrix} \underline{x}_1 \\ \vdots \\ \underline{x}_n \end{bmatrix}; \quad \underline{\tilde{x}} \sim N_{np}(\begin{bmatrix} \underline{\mu} \\ \vdots \\ \underline{\mu}, \end{bmatrix}, \begin{bmatrix} \Sigma & 0 & \dots & \dots & 0 \\ 0 & \Sigma & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix});$$

$$\text{Take } A = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & \dots & 1 & 0 & 0 & 0 & \dots & \dots \\ 0 & 1 & 0 & 0 & \dots & \dots & 0 & 1 & 0 & 0 & \dots & \dots \\ 0 & 0 & 1 & 0 & \dots & \dots & 0 & 0 & 1 & 0 & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \in p \times np$$

Which is n squared identity matrix of dimension p one against others

$$\overline{\underline{x}} = \frac{1}{n} A \underline{\tilde{x}} = \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n x_{i,1} \\ \vdots \\ \frac{1}{n}\sum_{i=1}^n x_{i,n} \end{bmatrix} \sim N_p(\frac{1}{n} A \begin{bmatrix} \underline{\mu} \\ \vdots \\ \underline{\mu}, \end{bmatrix}, \frac{1}{n^2} A \begin{bmatrix} \Sigma & 0 & \dots & \dots & 0 \\ 0 & \Sigma & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} A') = N_p(\underline{\mu}, \frac{1}{n}\Sigma)$$

□

#Wishart_distribution - introduced in 1928

Let $\underline{Z}_1, \dots, \underline{Z}_n$ iid $\sim N_p(\underline{0}, \Sigma) \Rightarrow A = \sum_{i=1}^n \underline{Z}_i \underline{Z}_i' \sim Wish(\Sigma, m)$ where m is the number of independent addends, in this case $m = n$.
Spoiler: it's a multidimensional χ^2.

Prop - 1

Suppose $A_1 \sim Wish(\Sigma, m_1), A_2 \sim Wish(\Sigma, m_2), A_1 \perp\!\!\!\perp A_2 \Rightarrow A_1 + A_2 \sim Wish(\Sigma, m_1 + m_2)$

Proof - previous Prop

$$A_1 = \sum_{i=1}^{m_1} \underline{Z}_i \underline{Z}_i'; \quad A_2 = \sum_{i=1}^{m_2} \underline{\tilde{Z}}_i \underline{\tilde{Z}}_i' \quad \text{with } \underline{Z}_i, \underline{\tilde{Z}}_j \sim N_p(\underline{0}, \Sigma)$$

$$\Rightarrow \text{we call } \underline{Z}_i \text{ as } \underline{w}_{j \in \{1, \dots, m_1\}} \text{ and } \underline{\tilde{Z}}_i \text{ as } \underline{w}_{j \in \{m_1, \dots, m_1 + m_2\}}$$

$$\Rightarrow A_1 + A_2 = \sum_{i=1}^{m_1+m_2} \underline{w}_i \underline{w}_i' \sim Wish(\Sigma, m_1 + m_2)$$

□

Prop - 2

$A \sim Wish(\Sigma, m); C \in k \times p \Rightarrow CAC' \sim Wish(C\Sigma C', m)$

Proof - previous Prop

$A = \sum_{i=1}^{m} \underline{Z}_i \underline{Z}_i' \quad \text{with } \underline{Z}_i \sim N_p(\underline{0}, \Sigma) \text{ iid}$
$CAC' = \sum_{i=1}^{m} \underbrace{C\underline{Z}_i}_{\underline{w}_i} \underbrace{\underline{Z}_i' C}_{\underline{w}_i'} \text{ with } \underline{w}_i \sim N_k(\underline{0}, C\Sigma C') \Rightarrow CAC' \sim Wish(C\Sigma C', m)$

□

Prop - 3

$A \sim Wish(\Sigma, m); \sigma^2 \in \mathbb{R}^+ \backslash \{0\} \Rightarrow \sigma^2 A \sim Wish(\sigma^2 \Sigma, m)$

Proof - previous Prop

$\sigma^2 A = \sigma^2 \sum_{i=1}^{m} \underline{Z}_i \underline{Z}_i' = \sum_{i=1}^{m} \underbrace{\sigma \underline{Z}_i}_{\underline{w}_i} \underbrace{\sigma \underline{Z}_i'}_{\underline{w}_i'} \text{ with } \underline{w}_i \sim N_p(\underline{0}, \sigma^2 \Sigma) \Rightarrow \sigma^2 A \sim Wish(\sigma^2 \Sigma, m)$

□

Prop - 4

Let $p = 1 \Rightarrow Wish(\overbrace{\Sigma}^{\sigma^2}, m) = \sigma^2 \chi^2(m)$

Proof - previous Prop

If $p = 1 \rightarrow \Sigma = \sigma^2 \Rightarrow \frac{1}{\sigma^2} A = \sum_i^m (\frac{Z_i}{\sigma})^2$ with $\frac{Z_i}{\sigma} \sim N_1(0, 1)$
$\Rightarrow \frac{1}{\sigma^2} A \sim \chi^2(m)$
□

Theo

Take $\underline{x}_1, \ldots, \underline{x}_n \text{ iid } \sim N_p(\underline{\mu}, \Sigma)$
Then: $\sum_i^n (\underline{x}_i - \overline{\underline{x}})(\underline{x}_i - \overline{\underline{x}})' \sim Wish(\Sigma, n-1)$
Note: The second parameter is $n-1$ and not n since we are removing the subspace of the mean $span\{\underline{1}\}$.

Corollary

Take $\underline{x}_1, \ldots, \underline{x}_n \text{ iid } \sim N_p(\underline{\mu}, \Sigma)$
Then: $\hat{\Sigma} = \frac{1}{n} \sum_i^n (\underline{x}_i - \overline{\underline{x}})(\underline{x}_i - \overline{\underline{x}})' \sim Wish(\frac{1}{n}\Sigma, n-1) \Rightarrow S \sim Wish(\frac{1}{n-1}\Sigma, n-1)$

Theo

Take $\underline{x}_1, \ldots, \underline{x}_n \text{ iid } \sim N_p(\underline{\mu}, \Sigma)$
Then:

1. $\overline{\underline{x}} \sim N_p(\underline{\mu}, \frac{1}{n}\Sigma)$
2. $(n-1)S \sim Wish(\Sigma, n-1)$
3. $\overline{\underline{x}} \perp\!\!\!\perp S$

Recall - Central Limit Theorem #CLN

Take $\underline{x}_1, \ldots, \underline{x}_n \text{ iid } \sim F(\underline{\mu}, \Sigma)$ with F as generic distribution
Then:

- $\sqrt{n}(\underline{\overline{x}} - \underline{\mu}) \sim AN_p(\underline{0}, \Sigma)$
- if $n \to +\infty \quad \underline{\overline{x}} \sim N_p(\underline{\mu}, \frac{1}{n}\Sigma)$

> **IMPORTANT:**
> It's importante to notice that in this theorem is involed the sample mean!!

Recall - Law Large Number #LNN

Take $\underline{x}_1, \dots, \underline{x}_n$ iid $\sim F(\underline{\mu}, \Sigma)$

$\underline{\overline{x}} \xrightarrow{\mathbb{P}} \underline{\mu}, \quad S \xrightarrow{\mathbb{P}} \Sigma \quad$ as $n \to +\infty$

Inference for the mean

n >> p

Rule of Thumb: is valid if $n > 30p^2$

Framework: Take $\underline{x}_1, \dots, \underline{x}_n$ iid $\sim F(\underline{\mu}, \Sigma)$ with F as generic distribution

By the #CLN we know that:

$\sqrt{n}(\underline{\overline{x}} - \underline{\mu}) \sim AN_p(\underline{0}, \Sigma) \to \underline{\overline{x}} \sim AN_p(\underline{\mu}, \frac{1}{n}\Sigma)$

From this we can say, *approximating*, that:

$(\underline{\overline{x}} - \underline{\mu})'(\frac{1}{n}\Sigma)^{-1}(\underline{\overline{x}} - \underline{\mu}) = \boxed{n(\underline{\overline{x}} - \underline{\mu})'\Sigma^{-1}(\underline{\overline{x}} - \underline{\mu}) \sim \chi^2(p)}$

Rmk: if Σ is known it is a pivotal quantity for $\underline{\mu}$

If we put $n \to +\infty$ we can say that:

$$\boxed{S \xrightarrow[LLN]{\mathbb{P}} \Sigma \Rightarrow n(\underline{\overline{x}} - \underline{\mu})'S^{-1}(\underline{\overline{x}} - \underline{\mu}) = d^2_{(\frac{1}{n}S)^{-1}}(\underline{\overline{x}}, \underline{\mu}) \sim \chi^2(p)}$$

which is pivotal quantity of $\underline{\mu}$ so:

$\mathbb{P}[d^2_{(\frac{1}{n}S)^{-1}}(\underline{\overline{x}}, \underline{\mu}) \leq \chi^2_{1-\alpha}(p)] = 1 - \alpha$

We can create 2 ellipsoid:

- $\epsilon^{\alpha}_{\chi^2_{1-\alpha}(p)}(\underline{\mu}) = \{\underline{x} \in \mathbb{R}^p : d^2_{(\frac{1}{n}S)^{-1}}(\underline{x}, \underline{\mu}) \leq \chi^2_{1-\alpha}(p)\}$
- $\epsilon^{\alpha}_{\chi^2_{1-\alpha}(p)}(\underline{\overline{x}}) = \{\underline{\eta} \in \mathbb{R}^p : d^2_{(\frac{1}{n}S)^{-1}}(\underline{\eta}, \underline{\overline{x}}) \leq \chi^2_{1-\alpha}(p)\}$

So we can see, geometrically, as: $\mathbb{P}[\underline{\overline{x}} \in \epsilon^{\alpha}_{\chi^2_{1-\alpha}(p)}(\underline{\mu})] = 1 - \alpha = \mathbb{P}[\underline{\mu} \in \epsilon^{\alpha}_{\chi^2_{1-\alpha}(p)}(\underline{\overline{x}})]$

From this we can create the confidence region of $\underline{\mu}$ at level $1 - \alpha$:

$$\boxed{\text{Hotelling statistic} = T_0^2 = n(\underline{\overline{x}} - \underline{\mu}_0)'S^{-1}(\underline{\overline{x}} - \underline{\mu}_0) \quad \Rightarrow \quad CR_{1-\alpha}(\underline{\mu}) = \{\underline{\eta} \in \mathbb{R}^p : T_0^2 \leq \chi^2_{1-\alpha}(p)\}}$$

where $\underline{\mu}_0$ is the hypothesis H_0.

We reject H_0 if $T_0^2 > \chi^2_{1-\alpha}(p)$ if we have a confidence level α.

Otherwise, instead of fixing α we can evaluate p-value so at level α

we reject our H_0 if p-value $\leq \alpha$

Note: our $CR_{1-\alpha}(\underline{\mu})$ identifies values of $\underline{\mu}_0$ for which we cannot reject H_0

n ≈ p

Framework: Take $\underline{x}_1, \dots, \underline{x}_n \text{ iid } \sim N_p(\underline{\mu}, \Sigma)$

Rmk: we need Gaussianity hypothesis since it is the cost of not having so much data

#Fisher_distribution - introduced in 1928

Let $Y \sim \chi^2(n), \quad W \sim \chi^2(m), \quad Y \perp\!\!\!\perp W$

Then: $\frac{\frac{Y}{n}}{\frac{W}{m}} \sim F(n,m)$

Rmk: $t = \frac{Z}{\sqrt{\frac{W}{m}}}$ with $Z \sim N(0,1), W \sim \chi^2(m), Z \perp\!\!\!\perp W \Rightarrow t^2 \sim F(1,m)$

Rmk: $F(n,m) \xrightarrow{m \to +\infty} \frac{1}{n}\chi^2(n)$ since $\frac{W}{m} = \frac{1}{m}\sum_i^m Z_i^2 \xrightarrow{LLN} 1$ with $Z_i \sim N(0,1)$

Theo - Hotelling (1931)

Assume: $\underline{x} \sim N_p(\underline{\mu}, \Sigma)$ with $det(\Sigma) > 0$; $W \sim Wish(\Sigma, m)$ with $W \perp\!\!\!\perp \underline{x}$

Then: $\frac{m-p+1}{mp}(\underline{x} - \underline{\mu})' W^{-1} (\underline{x} - \underline{\mu}) \sim F(p, m-p+1)$

Corollary

$$\underline{x}_1, \dots, \underline{x}_n \text{ iid } \sim N_p(\underline{\mu}, \Sigma) \Rightarrow \boxed{n(\overline{\underline{x}} - \underline{\mu})' S^{-1} (\overline{\underline{x}} - \underline{\mu}) = d^2_{(\frac{1}{n}S)^{-1}}(\overline{\underline{x}}, \underline{\mu}) \sim \frac{n-1}{n-p} p \cdot F(p, n-p)}$$

Rmk: is a pivotal quantity

Proof - previous Corollary

$$\sqrt{n}(\overline{\underline{x}} - \underline{\mu}) \sim N_p(\underline{0}, \Sigma); \quad (n-1)S \sim Wish(\Sigma, n-1); \quad \sqrt{n}(\overline{\underline{x}} - \underline{\mu}) \perp\!\!\!\perp S$$

$$\Rightarrow n(\overline{\underline{x}} - \underline{\mu})' S^{-1} (\overline{\underline{x}} - \underline{\mu}) = d^2_{\frac{1}{n}S^{-1}}(\overline{\underline{x}}, \underline{\mu}) \sim \frac{n-1}{n-p} p \cdot F(p, n-p)$$

□

Fixed $\alpha \in [0,1] \quad CR_{1-\alpha}(\underline{\mu}) = \{\underline{\eta} \in \mathbb{R}^p : n(\overline{\underline{x}} - \underline{\mu})' S^{-1} (\overline{\underline{x}} - \underline{\mu}) \leq \frac{n-1}{n-p} p \cdot F_{1-\alpha}(p, n-p)\}$

So in a test of hypothesis $\mathbf{we\ reject\ } \boldsymbol{H_0} \mathbf{\ if\ } \boldsymbol{T_0^2 > \frac{n-1}{n-p} p \cdot F_{1-\alpha}(p, n-p)}$

Rmk:

Squared radius of the ellipse:

- n large: $\chi^2_{1-\alpha}(p)$;
- n small: $\frac{n-1}{n-p} p \cdot F_{1-\alpha}(p, n-p)$

Since: $\frac{n-1}{n-p} p \cdot F_{1-\alpha}(p, n-p) \xrightarrow{n \to +\infty} \chi^2_{1-\alpha}(p)$

Inference for linear combination of $\underline{\mu}$

Framework: we have $\underline{x}_1, \dots, \underline{x}_n \text{ iid } \sim N_p(\underline{\mu}, \Sigma)$

Goal: we want to do inference on linear combination of $\underline{\mu}$

Take a generic $\underline{a} \in \mathbb{R}^p \Rightarrow$ the estimator of $\underline{a}'\underline{\mu}$ is $\underline{a}'\overline{\underline{x}}$

Because we know that:

1. $\underline{a}'\underline{\overline{x}} \sim N_1(\underline{a}'\underline{\mu}, \frac{1}{n}\underline{a}'\Sigma\underline{a}) \Leftrightarrow \underbrace{\frac{\sqrt{n}(\underline{a}'\underline{\overline{x}}-\underline{a}'\underline{\mu})}{\sqrt{\underline{a}'\Sigma\underline{a}}}}_{\text{®}} \sim N_1(0,1)$
2. Starting from $(n-1)S \sim Wish(\Sigma, n-1)$ we can find:
   $\Rightarrow (n-1)\underline{a}'S\underline{a} \sim Wish(\underbrace{\underline{a}'\Sigma\underline{a}}_{\dim=1}, n-1) = (\underline{a}'\Sigma\underline{a})\chi^2(n-1)$
   $\Rightarrow \underbrace{\frac{(n-1)\underline{a}'S\underline{a}}{(\underline{a}'\Sigma\underline{a})}}_{\text{Ⓢ}} \sim \chi^2(n-1)$

Since ® ⫫ Ⓢ $\Rightarrow \frac{\text{®}}{\sqrt{\frac{\text{Ⓢ}}{n-1}}} \sim t(n-1)$

$\Rightarrow \boxed{\frac{\sqrt{n}(\underline{a}'\underline{\overline{x}}-\underline{a}'\underline{\mu})}{\sqrt{\underline{a}'S\underline{a}}} \sim t(n-1) \text{ which is a pivotal quantity of } \underline{a}'\underline{\mu}}.$

Fixed $\alpha \in [0,1]$:

$\mathbb{P}[\frac{\sqrt{n}|\underline{a}'\underline{\overline{x}}-\underline{a}'\underline{\mu}|}{\sqrt{\underline{a}'S\underline{a}}} < t_{1-\frac{\alpha}{2}}(n-1)] = 1-\alpha \Leftrightarrow \mathbb{P}[|\underline{a}'\underline{\overline{x}}-\underline{a}'\underline{\mu}| < t_{1-\frac{\alpha}{2}}(n-1)\sqrt{\frac{\underline{a}'S\underline{a}}{n}}] = 1-\alpha$

So we can create:

- Confidence Interval $= CI_{1-\alpha}(\underline{a}'\underline{\mu}) = [\underline{a}'\underline{\overline{x}} \pm t_{1-\frac{\alpha}{2}}(n-1)\sqrt{\frac{\underline{a}'S\underline{a}}{n}}]$;
- Hypothesis tests with as testing statistics $\boxed{t_0 = \frac{\sqrt{n}(\underline{a}'\underline{\overline{x}}-\delta_0)}{\sqrt{\underline{a}'S\underline{a}}}}$;

For hypothesis tests we can deal with:

1. $\boldsymbol{H_0 : \underline{a}'\underline{\mu} \leq \delta_0}$ vs $\boldsymbol{H_1 : \underline{a}'\underline{\mu} \geq \delta_0}$ $\Rightarrow$ **Reject H_0 if $t_0 > t_{1-\alpha}(n-1)$**
2. $\boldsymbol{H_0 : \underline{a}'\underline{\mu} = \delta_0}$ vs $\boldsymbol{H_1 : \underline{a}'\underline{\mu} \neq \delta_0}$ $\Rightarrow$ **Reject H_0 if $|t_0| > t_{1-\frac{\alpha}{2}}(n-1)$**

Prop

Take $\alpha \in [0,1]$:

$\forall \underline{a} \in \mathbb{R}^p \quad CI_{1-\alpha}(\underline{a}'\underline{\mu}) = [\underline{a}'\underline{\overline{x}} + t_{1-\frac{\alpha}{2}}(n-1)\sqrt{\frac{\underline{a}'S\underline{a}}{n}}] \quad \Leftrightarrow \quad \forall \underline{a} \in \mathbb{R}^p \quad \mathbb{P}[\underline{a}'\underline{\mu} \in CI_{1-\alpha}(\underline{a}'\underline{\mu})] = 1-\alpha$

Rmk: which is **not equal to** $\mathbb{P}[\underline{a}'\underline{\mu} \in CI_{1-\alpha}(\underline{a}'\underline{\mu}), \quad \forall \underline{a} \in \mathbb{R}^p] = 1-\alpha$

Recall

Take $\underline{b}, \underline{d} \in \mathbb{R}^p$ we know that $\cos(\theta) = \frac{\underline{b}'\underline{d}}{||\underline{b}||||\underline{d}||}$ so we deduce that $(\underline{b}'\underline{d})^2 \leq ||\underline{b}||^2||\underline{d}||^2$

Theo - Extended Cauchy-Schwartz

Take $B \in p \times p$ positive defined and symmetric.
Then: $\forall \underline{b}, \underline{d} \in \mathbb{R}^p$ holds $(\underline{b}'\underline{d})^2 \leq (\underline{b}'B\underline{b})(\underline{d}'B^{-1}\underline{d})$
Rmk: the equality holds if $\underline{b} \in \mathcal{L}(B^{-1}\underline{d})$

Proof - previous Theo

$$(\underline{b}'\underline{d})^2 = (\underline{b}'B^{\frac{1}{2}}B^{-\frac{1}{2}}\underline{d})^2 \leq ||B^{\frac{1}{2}}\underline{b}||^2||B^{-\frac{1}{2}}\underline{d}||^2 = (\underline{b}'B^{\frac{1}{2}}B^{\frac{1}{2}}\underline{b})(\underline{d}'B^{-\frac{1}{2}}B^{-\frac{1}{2}}\underline{d}) = (\underline{b}'B\underline{b})(\underline{d}'B^{-1}\underline{d})$$

□

Lemma

Take $B \in p \times p$ positive defined and symmetric and $\underline{d} \in \mathbb{R}^p$

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

Proof - previous Lemma

$$(\underline{x}'\underline{d})^2 \leq (\underline{x}' B \underline{x})(\underline{d}' B^{-1} \underline{d}) \text{ by Cauchy-Schwartz } \Rightarrow \frac{(\underline{x}'\underline{d})^2}{\underline{x}' B \underline{x}} \leq \underline{d}' B^{-1} \underline{d} \quad \forall \underline{x} \neq \underline{0}$$

$$\Rightarrow \max_{\underline{x} \in \mathbb{R}^p, ||\underline{x}|| \neq 0} \frac{(\underline{x}'\underline{d})^2}{\underline{x}' B \underline{x}} = \underline{d}' B^{-1} \underline{d}$$

□

So we find that:

$$\max_{\underline{a} \in \mathbb{R}^p, ||\underline{a}|| \neq 0} n \frac{[\underline{a}'(\overline{\underline{x}} - \underline{\mu})]^2}{\underline{a}' S \underline{a}} = n(\overline{\underline{x}} - \underline{\mu})' S^{-1} (\overline{\underline{x}} - \underline{\mu}) \sim p \frac{n-1}{n-p} F(p, n-p)$$

This is helpful in addressing the problem of resolving $\mathbb{P}[\underline{a}'\underline{\mu} \in CI_{1-\alpha}(\underline{a}'\underline{\mu}), \quad \forall \underline{a} \in \mathbb{R}^p] = 1 - \alpha$ because:

$$\mathbb{P}[\underline{a}'\underline{\mu} \in CI_{1-\alpha}(\underline{a}'\underline{\mu}), \quad \forall \underline{a} \in \mathbb{R}^p] = 1 - \alpha \overset{\text{as before}}{\iff} \mathbb{P}[n \frac{[\underline{a}'\overline{\underline{x}} - \underline{a}'\underline{\mu}]^2}{\underline{a}' S \underline{a}} \leq \overbrace{c^2}^{unknown} = c, \quad \forall \underline{a} \in \mathbb{R}^p]$$

Finding c is the same of maximize $n \frac{[\underline{a}'(\overline{\underline{x}} - \underline{\mu})]^2}{\underline{a}' S \underline{a}}$ so we reach that

$CI_{1-\alpha}(\underline{a}'\underline{\mu}) = [\underline{a}'\overline{\underline{x}} \pm \sqrt{p \frac{n-1}{n-p} F_{1-\alpha}(p, n-p)} \cdot \sqrt{\frac{\underline{a}' S \underline{a}}{n}}]$ and we have finally achieve:

$$\mathbb{P}[\underline{a}'\underline{\mu} \in CI_{1-\alpha}(\underline{a}'\underline{\mu}), \quad \forall \underline{a} \in \mathbb{R}^p] = 1 - \alpha$$

To summarize:

– **t-test (one at time):** $\forall \underline{a} \in \mathbb{R}^p \quad CI_{1-\alpha}(\underline{a}'\underline{\mu}) = [\underline{a}'\overline{\underline{x}} + t_{1-\frac{\alpha}{2}}(n-1)\sqrt{\frac{\underline{a}' S \underline{a}}{n}}]$

– **F-test (multiple at time):** $CI_{1-\alpha}(\underline{a}'\underline{\mu}) = [\underline{a}'\overline{\underline{x}} \pm \sqrt{p \frac{n-1}{n-p} F_{1-\alpha}(p, n-p)} \cdot \sqrt{\frac{\underline{a}' S \underline{a}}{n}} \quad \forall \underline{a} \in \mathbb{R}^p]$

Interpretation: in the first case we want to have confidence values of an interval in $\mathbb{R}$, in the second case we want to have confidence values of each possible direction (like an area in $\mathbb{R}^p$) so caused by #curse_of_dimensionality we need to enlarge the interval to keep the same confidence level.

Geometrically: in the first case you are projecting the elliptic $CI_{1-\alpha}(\underline{a}'\underline{\mu})$ on one direction (which is $\underline{a}$) and this projection is called shadow.

Q: How to not enlarge our CI as in F-test?

R: Bonferroni solution

Goal: we want to take k tests simultaneously and also keep "small" our CI

Take $\underline{a}_1, \dots, \underline{a}_k \in \mathbb{R}^p$ to test; which is a correct $\beta \in [0,1]$ s.t. $\mathbb{P}\{\cap_{i=1}^k [\underline{a}_i' \underline{\mu} \in CI_{1-\beta}(\underline{a}_i' \underline{\mu})]\} = 1 - \alpha$ with $CI_{1-\beta}(\underline{a}_i' \underline{\mu})$ t-tests?

Bonferroni Idea: $\beta = \frac{\alpha}{k}$

We can arrive to this with simple calculations:

$$\begin{aligned}
\mathbb{P}\{\cap_{i=1}^k [\underline{a}_i' \underline{\mu} \in CI_{1-\beta}(\underline{a}_i' \underline{\mu})]\} &= 1 - \alpha = \\
&= 1 - \mathbb{P}\{\cup_{i=1}^k [\underline{a}_i' \underline{\mu} \notin CI_{1-\beta}(\underline{a}_i' \underline{\mu})]\} = \\
&= 1 - \sum_{i=1}^k \underbrace{\mathbb{P}\{\underline{a}_i' \underline{\mu} \notin CI_{1-\beta}(\underline{a}_i' \underline{\mu})\}}_{=\beta} = \\
&= 1 - k\beta \\
&\Rightarrow \alpha = k\beta
\end{aligned}$$

From this we can create Bonferroni Confidence Interval (one at time for each $\underline{a}_i$):

$BCI_{1-\frac{\alpha}{k}}(\underline{a}_i' \underline{\mu}) = [\underline{a}_i' \overline{\underline{x}} + t_{1-\frac{\alpha}{2k}}(n-1)\sqrt{\frac{\underline{a}_i' S \underline{a}_i}{n}}]$

So we can also expand this theory to testing:

$$H_0 = \begin{cases} \underline{a}_1' \underline{\mu} = \delta_1 \\ \underline{a}_2' \underline{\mu} = \delta_2 \\ \vdots \\ \underline{a}_k' \underline{\mu} = \delta_k \end{cases} \quad \text{vs} \quad H_1 = \text{at least 1 fails}$$

So we reject H_0 at level α if for at least one $i \in \{1, \dots, k\}$ $\frac{\sqrt{n}|\underline{a}_i' \overline{\underline{x}} - \delta_i|}{\sqrt{\underline{a}_i' S \underline{a}_i}} > t_{1-\frac{\alpha}{2k}}(n-1)$

> Taking as p_i the p-value of the test $H_{0,i}$ vs $H_{1,i}$ we know that we reject $H_{0,i}$ if $p_i < \frac{\alpha}{k}$.

So we find:

$$\begin{aligned}
\mathbb{P}[\text{Reject } H_0 | H_0 \text{ holds}] &= \mathbb{P}[\cup_{i=1}^k \{\text{Reject } H_{0,i}\} | \cap_{i=1}^k H_{0,i}] \\
&\leq \sum_{i=1}^k \mathbb{P}[\text{Reject } H_{0,i} | H_{0,i} \text{ holds}] = \sum_{i=1}^k \frac{\alpha}{k} = \alpha
\end{aligned}$$

Large scale hypothesis testing and FDR

The solution reported above has a huge limit, if $k \uparrow$ then $\beta \to 0$. This is a problem in cases like large scale hypothesis testing, for example in genomics.

Solution: False Discovery Rate (#FDR) by Benjamini&Hochberg (1995)

We want to test k hypothesis with a strategy D (for example Bonferroni test). We can construct this table:

$truth \downarrow$	Not reject H_0	Reject H_0	
H_0	v	V	k_0
H_1	T	S	$k - k_0$
	$k - R$	R	k

Where:

- $V =$ false discoveries;
- $T =$ missed discoveries;
- $S =$ true discoveries;
- $R =$ rejections (which are the only things observable)

We create $I_0 = \{i \in \{1, \dots, k\} : H_{0,i} \text{ holds}\}$.
In the following calculation we use Bonferroni as strategy D:

$$\begin{aligned}\mathbb{P}[V \geq 1] = \mathbb{P}[\text{ at least 1 false discovery}] &= \mathbb{P}[\cup_{j \in I_0}\{\text{Reject } H_{0,j}\} | \cap_{j \in I_0} H_{0,j}] \\ &\leq \sum_{j \in I_0} \mathbb{P}[\text{Reject } H_{0,j} | H_{0,j} \text{ holds}] \\ &= \sum_{j \in I_0} \frac{\alpha}{k} = \frac{k_0}{k}\alpha \\ &\leq \alpha\end{aligned}$$

The previous probability is called *Family-Wise Error Rate* (#FWER) and for Bonferroni hold $FWER \leq \alpha$.

So we can define the proportion of false discoveries among discoveries $= Q = \begin{cases} 0 & R = 0 \\ \frac{V}{R} & R > 0 \end{cases}$

From above we create the *False Discovery Rate* $= FDR = \mathbb{E}[Q]$

Observation - 1 ($k_0 = k$)

We don't have any discovery so $Q = \begin{cases} 0 & V = 0 \\ 1 & V > 0 \end{cases} \Rightarrow FDR = FWER$

Observation - 2 ($k_0 < k$)

$Q = \begin{cases} 0 & V = 0 \\ \frac{V}{R} \leq 1 & V > 0 \end{cases} \Rightarrow Q \leq \mathbb{1}_{V>0} \Rightarrow FDR = \mathbb{E}[Q] \leq \mathbb{E}[\mathbb{1}_{V>0}] = \mathbb{P}[V > 0] = FWER$
So we reach that controlling FDR is less restricting that controlling $FWER$.

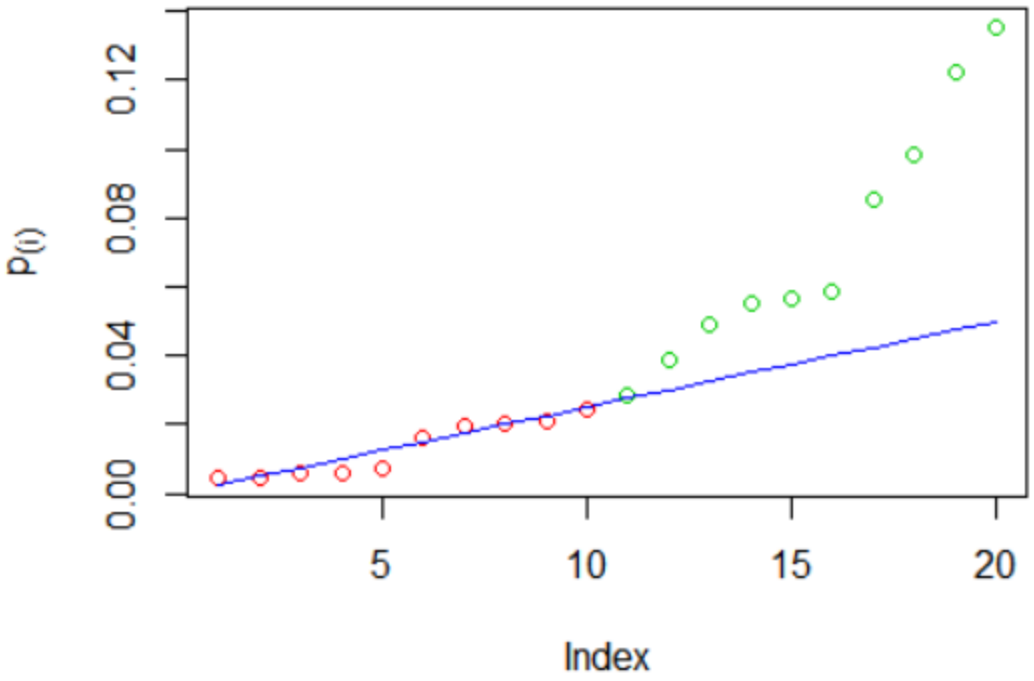
How to control FDR?

Framework: p_i is the p-value of $H_{0,i} \quad \text{vs} \quad H_{1,i}$

1. Ordering the p-value such that the biggest one is $p_{(1)}$ while the smallest one is $p_{(k)}$.
2. Taking $\alpha \in (0, 1)$, find m which is the greatest index s.t. $p_{(m)} < m\frac{\alpha}{k}$, same as:
   $m = \arg\max_{i \in \{1, \dots, k\}}\{p_{(i)} \leq i\frac{\alpha}{k}\}$
3. Reject $H_{0,(1)}, \dots, H_{0,(m)}$

In the following picture we can see that the blue line is $i\frac{\alpha}{k}$ while the point are p_i. The Bonferroni threshold will be a horizontal line at level $\frac{\alpha}{k}$ so if k increase we are subject to accepting more

and more $H_{0,(i)}$.



Theo

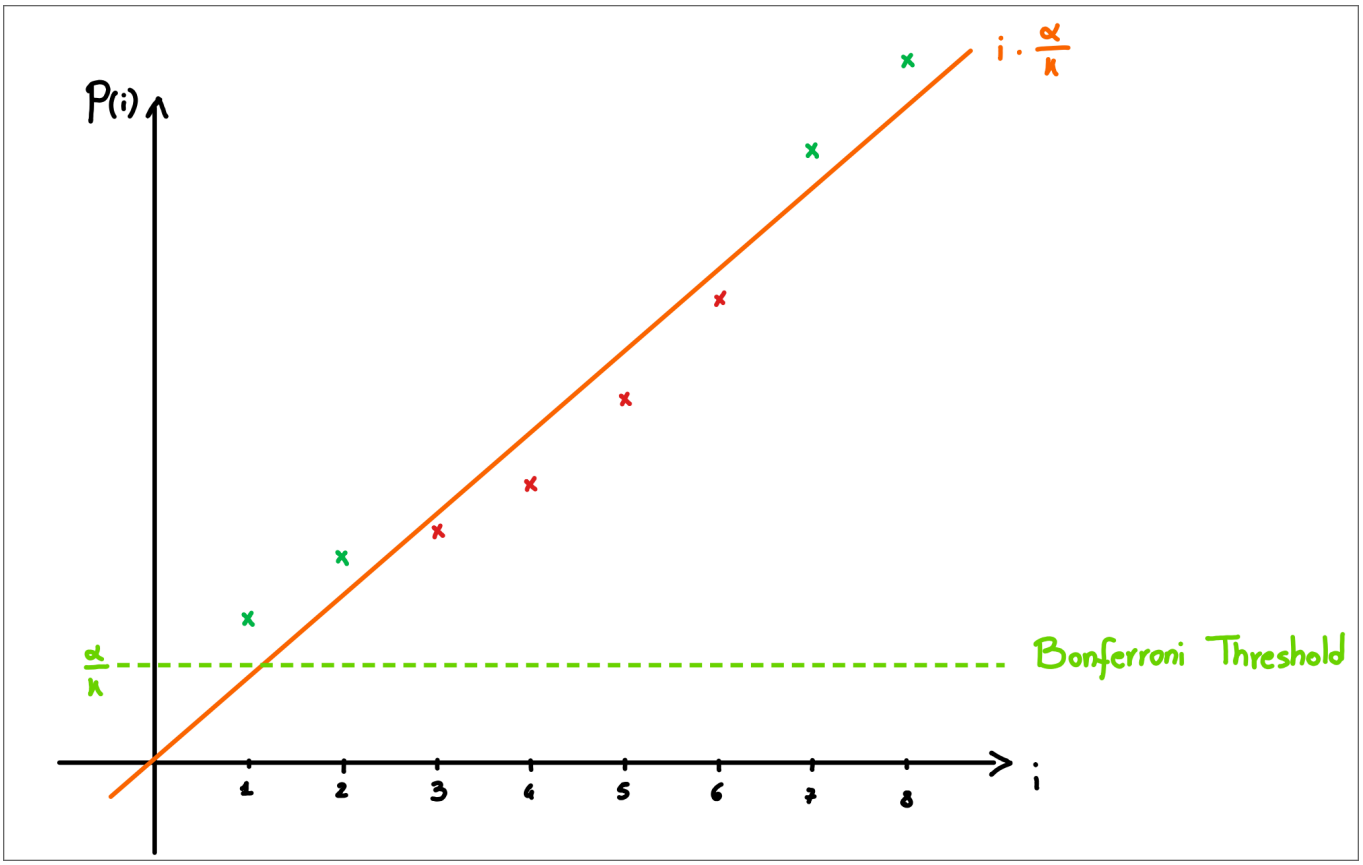
If $p_i, \dots, p_k$ are independent, this strategy control FDR at level α.
Rmk: each p-value p_i is a function of $\underline{x}_1, \dots, \underline{x}_n$

Q: What if p_i aren't independent?

Theo - Benjamini ad Yekuteli (2001)

- $p_1, \dots, p_k$ are positively correlated $\Rightarrow$ holds B&H (1995) strategy
- $p_1, \dots, p_k$ are negatively correlated $\Rightarrow m^* = \arg\max_{i \in \{1,\dots,k\}} \{p_i \leq i \frac{\alpha}{k \cdot C(k)}\}$ and $C(k) = \sum_{j=1}^{k} \frac{i}{j}$

Observation



In a situation like this, how different strategies works:

- **Benjamini&Hochberg:** Since $m = 6$ reject $H_{0,(1)}, \dots, H_{0,(6)}$
- **Efron (2010):** which is a strong version of B&H, reject $H_{0,(i)}$ if $p_{(i)} \leq i\frac{\alpha}{k}$ so reject $H_{0,(3)}, \dots, H_{0,(6)}$
- **Bonferroni:** doesn't reject

Comparing means of multivariate Gaussian Distr.

Framework:

We have paired data of n statistical units observed twice: $\underline{x}_{1,i}, \underline{x}_{2,i} \in \mathbb{R}^p$ with $\mathrm{i} \in \{1, \dots, n\}$

So we have:

$$\begin{pmatrix} \underline{x}_{1,1} \\ \underline{x}_{2,1} \end{pmatrix}, \begin{pmatrix} \underline{x}_{1,2} \\ \underline{x}_{2,2} \end{pmatrix}, \dots, \begin{pmatrix} \underline{x}_{1,n} \\ \underline{x}_{2,n} \end{pmatrix} \text{ iid } \sim N_{2p}\left(\begin{pmatrix} \underline{\mu}_1 \\ \underline{\mu}_2 \end{pmatrix}, \overbrace{\Sigma}^{\in 2p \times 2p} \right)$$

Rmk: independence is between vectors not between components.

Problem to solve: $H_0 : \underline{\mu}_1 - \underline{\mu}_2 = \underline{\delta} \quad \text{vs} \quad H_1 : \underline{\mu}_1 - \underline{\mu}_2 \neq \underline{\delta}$

We create:

$\underline{D}_i \in \mathbb{R}^p$ s.t. $\underline{D}_i = \underline{x}_{1,i} - \underline{x}_{2,i}$ iid $\sim N_p(\underline{\delta}, \Sigma_D)$

$$\Rightarrow \overline{\underline{D}} = \frac{1}{n}\sum_{i=1}^{n} \underline{D}_i \text{ and } S_D = \frac{1}{n-1}\sum_{i=1}^{n} (\underline{D}_i - \overline{\underline{D}})(\underline{D}_i - \overline{\underline{D}})'$$

So we reach that: $\boxed{n(\overline{\underline{D}} - \underline{\delta})' S^{-1} (\overline{\underline{D}} - \underline{\delta}) \sim p\frac{n-1}{n-p} F(p, n-p)}$ which is a pivotal quantity and useful to conclude, using theory developed in previous chapters, that taking $\alpha \in (0, 1)$:

$$CR_{1-\alpha}(\underline{\mu}_1 - \underline{\mu}_2) = CR_{1-\alpha}(\underline{\delta}) = \{\underline{\delta} \in \mathbb{R}^p : n(\overline{\underline{D}} - \underline{\delta})' S^{-1} (\overline{\underline{D}} - \underline{\delta}) \leq p\frac{n-1}{n-p} F_{1-\alpha}(p, n-p)\}$$

$$CR_{1-\alpha}(\underline{a}'\underline{\delta}) = \{\underline{a}'\overline{\underline{D}} \pm \sqrt{p\frac{n-1}{n-p}F_{1-\alpha}(p,n-p)}\sqrt{\frac{\underline{a}' S_D \underline{a}}{n}}\}$$

Univariate case

Framework: we repeat q times each measure.

Take $\underline{x}_1, \dots, \underline{x}_n$ iid $\sim N_q(\underline{\mu}, \Sigma)$ so each

$\underline{x}_i = [x_{i,1}, \dots, x_{i,q}]' \in \mathbb{R}^q \leftarrow q$ measurements of the same quantity

Goal: test $H_0 : \mu_1 = \dots = \mu_q \quad$ vs $\quad H_1 : \exists i, j \text{ s.t. } \mu_i \neq \mu_j$

We can define the #Contrast_matrix C as:

$C \in (q-1) \times q$ is a contrast matrix if $C = \begin{bmatrix} \underline{c}_1' \\ \vdots \\ \underline{c}_{q-1}' \end{bmatrix}$ with $\underline{c}_1, \dots, \underline{c}_{q-1}$ are linearly independent and

$\forall i \quad \underline{c}_i' \underline{1} = 0.$

We can rewrite this definition as:

$$\boxed{C \text{ is a contrast matrix if } \mathcal{L}^{\perp}(\underline{1}) = span(\underline{c}_1, \dots, \underline{c}_{q-1})}$$

So we can rewrite our test as: $H_0 : C\underline{\mu} = \underline{0} \quad$ vs $\quad H_1 : C\underline{\mu} \neq \underline{0}$

Interpretation: a contrast matrix is used to specify comparisons among group means. It allows to test hypotheses about specific combinations of group means beyond simple pairwise comparisons.

Recall

We already know that:

- $C\overline{\underline{x}}$ is unbiased for $C\underline{\mu}$;
- $\overline{\underline{x}} \sim N_q(\underline{\mu}, \frac{1}{n}\Sigma) \Rightarrow C\overline{\underline{x}} \sim N_{q-1}(C\underline{\mu}, \frac{1}{n}C\Sigma C')$
- S is an estimator for $\Sigma \Rightarrow (n-1)CSC' \sim Wish(C\Sigma C', n-1) \perp\!\!\!\perp C\overline{\underline{x}}$
  - Hotelling Theorem $n(C\overline{\underline{x}} - C\underline{\mu})'(CSC')^{-1}(C\overline{\underline{x}} - C\underline{\mu}) \sim \frac{(n-1)(q-1)}{n-q+1} F(q-1, n-q+1)$ which is pivotal

We define the test statistics as $\boxed{T_0^2 = n(C\overline{\underline{x}})'(CSC')^{-1}C\overline{\underline{x}}}$ so for a fixed α *we reject* H_0 *if* $T_0^2 > \frac{(n-1)(q-1)}{n-q+1} F_{1-\alpha}(q-1, n-q+1)$

Obs

Let $C, \tilde{C}$ two contrast matrices, take $B \in (q-1) \times (q-1)$ with $det(B) \neq 0$ such that $C = B\tilde{C}$ which is the operator to change basis. So:

$T_0^2 = n(C\overline{\underline{x}})'(CSC')^{-1}C\overline{\underline{x}} = n(B\tilde{C}\overline{\underline{x}})'(B\tilde{C}S\tilde{C}'B')^{-1}B\tilde{C}\overline{\underline{x}} = n(\tilde{C}\overline{\underline{x}})'(\tilde{C}S\tilde{C}')^{-1}\tilde{C}\overline{\underline{x}} = \tilde{T}_0^2$

Obs

Let $\underline{x}_1, \dots, \underline{x}_n$ iid $\sim N_q(\underline{\mu}, \Sigma)$, have $\mathcal{L}^\perp = span(\underline{c}_1, \dots, \underline{c}_{p-k})$ so our $C = \begin{bmatrix} \underline{c}_1' \\ \vdots \\ \underline{c}_{p-k}' \end{bmatrix}$ and we can also test: $H_0 : C\underline{\mu} = \underline{0} \quad \text{vs} \quad H_1 : C\underline{\mu} \neq \underline{0}$

So our test statistics $\boxed{T_0^2 = n(C\overline{\underline{x}})'(CSC')^{-1}C\overline{\underline{x}} \sim \frac{(n-1)(p-k)}{n-p+k}F(p-k, n-p+k)}$

Multivariate case

Each unit isn't a number but become a vector of dimension l that is observed q times.
So to construct the test we made:

$$H_0 : \begin{cases} \mu_{1,1} = \dots = \mu_{q,1} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ \mu_{1,l} = \dots = \mu_{q,l} \end{cases} \quad \text{vs} \quad H_1 : \exists i, j, k \text{ s.t. } \mu_{i,k} \neq \mu_{j,k}$$

tips: change rappresentation to achieve
$\underline{x}_i = (x_{i,1,1}, \dots, x_{i,1,l}, x_{i,2,1}, \dots, x_{i,2,l}, \dots, \dots, x_{i,q,1}, \dots, x_{i,q,l}) \in \mathbb{R}^{lq}$ in this way C is a block matrix where each block is a contrast matrix.

Multivariate ANalysis Of VAriance #Manova

Framework: To complicate things we can take sample from different Gaussian distribution. So our sample become: $\underline{x}_{1,1}, \dots, \underline{x}_{1,n_1}$ iid $\sim N_p(\underline{\mu}_1, \Sigma); \dots; \underline{x}_{g,1}, \dots, \underline{x}_{g,n_g}$ iid $\sim N_p(\underline{\mu}_g, \Sigma)$

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

$\underline{x}_{1,1}, \dots, \underline{x}_{1,n_1}$ iid $\sim N_p(\underline{\mu}_1, \Sigma) \perp\!\!\!\perp \underline{x}_{2,1}, \dots, \underline{x}_{2,n_2}$ iid $\sim N_p(\underline{\mu}_2, \Sigma)$

Goal: inference of $\underline{\mu}_1 - \underline{\mu}_2$

We have:

- $\overline{\underline{x}}_1 - \overline{\underline{x}}_2 \sim N_p(\underline{\mu}_1 - \underline{\mu}_2, (\frac{1}{n_1} + \frac{1}{n_2})\Sigma) \Rightarrow (\frac{1}{n_1} + \frac{1}{n_2})^{-\frac{1}{2}}[(\overline{\underline{x}}_1 - \overline{\underline{x}}_2) - (\underline{\mu}_1 - \underline{\mu}_2)] \sim N_p(\underline{0}, \Sigma)$
- S_1, S_2 both estimate Σ so we create
  $S_{pooled} = \frac{(n_1-1)S_1 + (n_2-1)S_2}{n_1+n_2-2} \Rightarrow (n_1+n_2-2)S_{pooled} \sim Wish(\Sigma, n_1+n_2-2)$

So we can develop Hotelling theorem obtaining:

$$\boxed{\begin{aligned} &(\frac{1}{n_1} + \frac{1}{n_2})^{-1}[(\overline{\underline{x}}_1 - \overline{\underline{x}}_2) - (\underline{\mu}_1 - \underline{\mu}_2)]' S_{pooled}^{-1}[(\overline{\underline{x}}_1 - \overline{\underline{x}}_2) - (\underline{\mu}_1 - \underline{\mu}_2)] = \\ &= d_{[(\frac{1}{n_1}+\frac{1}{n_2})S]^{-1}}(\overline{\underline{x}}_1 - \overline{\underline{x}}_2, \underline{\mu}_1 - \underline{\mu}_2) \sim \frac{(n_1+n_2-2)p}{n_1+n_2-1-p}F(p, n_1+n_2-1-p) \end{aligned}}$$

Which is a pivotal quantity.
Take $\alpha \in (0,1)$:
$CR_{1-\alpha}(\underline{\mu}_1 - \underline{\mu}_2) = \{\underline{\nu} = \underline{\mu}_1 - \underline{\mu}_2 : d_{[(\frac{1}{n_1}+\frac{1}{n_2})S]^{-1}}(\overline{\underline{x}}_1 - \overline{\underline{x}}_2, \underline{\nu}) \leq \frac{(n_1+n_2-2)p}{n_1+n_2-1-p}F_{1-\alpha}(p, n_1+n_2-1-p)\}$

Anova

Framework: we have unit of $p = 1$ features and we collect g different unit each one $n_{i \in \{1,\dots,g\}}$ times.

Our sample is made by:

$x_{1,1},\dots,x_{1,n_1}$ iid $\sim N_1(\mu_1,\sigma^2)$; $x_{2,1},\dots,x_{2,n_2}$ iid $\sim N_1(\mu_2,\sigma^2)$; $\dots$; $x_{g,1},\dots,x_{g,n_g}$ iid $\sim N_1(\mu_g,\sigma^2)$

Where $x_{i,k} \perp\!\!\!\perp x_{j,l} \quad \forall i,k,j,l$

Firstly, we reparameter the problem by breaking down $\boxed{\mu_i = \mu + \tau_i}$ so we can express each $\boxed{x_{i,j} = \mu + \tau_i + \epsilon_{i,j}}$ with $\epsilon_{i,j} \in N(0,\sigma^2)$.

Problem: we pass from $g-$parameters (μ_i) to $(g+1)-$parameters (μ, τ_i).

Goal - 1: Find estimator for μ

$\overline{x} = \frac{1}{n}\sum_{i=1}^{g}\sum_{j=1}^{n_i} x_{i,j}$ with $n = n_1 + n_2 + \dots + n_g$

Is it unbiased?

$\mathbb{E}[\overline{x}] = \frac{1}{n}\sum_{i=1}^{g}\sum_{j=1}^{n_i}\mathbb{E}[x_{i,j}] = \frac{1}{n}\sum_{i=1}^{g}\sum_{j=1}^{n_i}(\mu+\tau_i) = \mu + \frac{\sum_i n_i\tau_i}{n}$

Idea: To unbiased the estimator of μ and to resolve the problem of too much parameters we add a constrain: $\sum_i n_i\tau_i = 0$

Goal - 2: Find estimator for τ_i

$\overline{x}_i = \frac{1}{n}\sum_{j=1}^{n_i} x_{i,j}$ with $n = n_1 + n_2 + \dots + n_g$

$\Rightarrow \mathbb{E}[\overline{x}_i - \overline{x}] = \mu + \tau_i - \mu = \tau_i$

Recall - Decomposition of variance

framework: take $\underline{x} = (x_{1,1},\dots,x_{1,n_1},x_{2,1},\dots,x_{2,n_2},\dots,\dots,\dots,x_{g,1},\dots,x_{g,n_g})' \in \mathbb{R}^n$ and define

$\underline{u}_1' = \left[\underbrace{1\dots1}_{n_1}0\ \dots\ 0\right]$, $\underline{u}_2' = \left[\underbrace{0\dots0}_{n_1}\underbrace{1\dots1}_{n_2}0\ \dots\ 0\right]$, $\underline{u}_3' = \left[\underbrace{0\dots0}_{n_1+n_2}\underbrace{1\dots1}_{n_3}0\ \dots\ 0\right]$, $\dots$

By construction we know that the following properties holds for $\underline{u}_i$:

- $\underline{u}_1,\dots,\underline{u}_g$ are linearly independent;
- $<\underline{u}_i,\underline{u}_j>=0 \quad \forall i \neq j$ so they are orthogonal each others;
- $\underline{1} \in span\{\underline{u}_1,\dots,\underline{u}_g\}$

So we reach:

$\underline{x} = \underbrace{\overline{x}\cdot\underline{1}}_{\mu} + \underbrace{\sum_{i=1}^{g}(\overline{x}_i-\overline{x})\underline{u}_i}_{\sum_i(\mu_i-\mu)=\sum_i\tau_i=\tau} + (\underline{x} - \sum_g^{i=1}\overline{x}_i\cdot\underline{u}_i)$

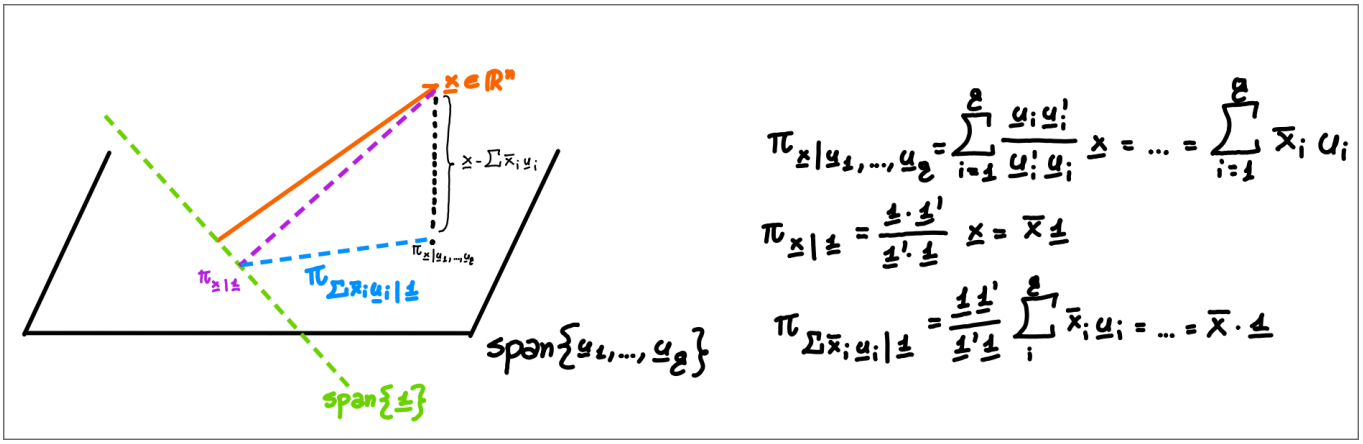
Using Pitagora's theorem can write:

$||\underline{x}||^2 = ||\overline{x}\cdot\underline{1}||^2 + ||\sum_{i=1}^{g}(\overline{x}_i-\overline{x})\underline{u}_i||^2 + ||\underline{x} - \sum_g^{i=1}\overline{x}_i\cdot\underline{u}_i||^2$

$\Rightarrow \underbrace{\sum x_{i,j}^2}_{SS_{TOT}} = \underbrace{n\overline{x}^2}_{SS_{mean}} + \underbrace{\sum_{i=1}^{g} n_i(\overline{x}_i-\overline{x})^2}_{SS_{treatment}} + \underbrace{\sum_{i=1}^{g}\sum_{j=1}^{n_i}(x_{i,j}-\overline{x}_i)^2}_{SS_{residual}} \leftrightarrow$ #1_decomposition_of_variance

$\Rightarrow \overbrace{||\underline{x}-\overline{x}\cdot\underline{1}||^2}^{SS_{centered}} = \overbrace{||\sum_{i=1}^{g}(\overline{x}_i-\overline{x})\underline{u}_i||^2}^{SS_{treatment}} + \overbrace{||\underline{x} - \sum_g^{i=1}\overline{x}_i\cdot\underline{u}_i||^2}^{SS_{residual}} \leftrightarrow$ #2_decomposition_of_variance

Rmk: SS stands for Sum of Squares



Goal - 3: Test $H_0 : \mu_1 = \cdots = \mu_g \quad$ vs $\quad H_1 : H_0^c$ equivalent to $H_0 : \tau_1 = \cdots = \tau_g \quad$ vs $\quad H_1 : H_0^c$

Idea: reject H_0 if $SS_{treatment}$ is large w.r.t. $SS_{residual}$ so we take the ratio $\frac{SS_{treatment}}{SS_{residuals}}$

$SS_{residuals} = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{i,j} - \overline{x}_i)^2 = \sum_{i=1}^{g} (n_i - 1) S_i^2 \overset{\text{since groups are } \perp\!\!\!\perp}{\Longrightarrow} SS_{residuals} \sim \sigma^2 \chi^2 (n - g)$

If H_0 is true:

- $\sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{i,j} - \overline{x})^2 = (n-1) S \sim \sigma^2 \chi^2 (n-1)$
- $SS_{treatment} \sim \sigma^2 \chi^2 (g-1)$

Remark: $\overline{x}_i$ is the mean of n_i measurements of a unit while $\overline{x}$ is the overall mean

So $\frac{SS_{treatment}}{g-1} \frac{n-g}{SS_{residuals}} \sim F(g-1, n-g)$

$\Rightarrow$ **reject H_0 at level α if $\frac{SS_{treatment}}{g-1} \frac{n-g}{SS_{residuals}} > F_{1-\alpha}(g-1, n-g)$**

Manova

Framework: we have unit of $p \geq 1$ features and we collect g different unit each one $n_{i \in \{1,\dots,g\}}$ times.

Take $x_{i,j}$ iid $\sim N_p(\underline{\mu}, \Sigma)$ with $i \in \{1, \dots, g\}$ and $j \in \{1, \dots, n_i\}$

We reparametrize so reach: $\boxed{\underline{x}_{i,j} = \underline{\mu} + \underline{\tau}_i + \underline{\epsilon}_{i,j}}$ with $\underline{\epsilon}_{i,j} \sim N_p(\underline{0}, \Sigma)$ and as constrain $\sum_i n_i \underline{\tau}_i = 0$

Decomposition of covariance

Taking $\underline{\overline{x}} = \frac{1}{n} \sum_{i=1}^{g} \sum_{j}^{n_i} \underline{x}_{i,j}$ we deduce:

$$\sum_{i=1}^{g} \sum_{j=1}^{n_i} (\underline{x}_{i,j} - \underline{\overline{x}})(\underline{x}_{i,j} - \underline{\overline{x}})' = \sum_{i=1}^{g} n_i (\underline{\overline{x}}_i - \underline{\overline{x}})(\underline{\overline{x}}_i - \underline{\overline{x}})' + \sum_{i=1}^{g} \sum_{j}^{n_i} (\underline{x}_{i,j} - \underline{\overline{x}}_i)(\underline{x}_{i,j} - \underline{\overline{x}}_i)' = B + W$$

Goal: Test $H_0 : \underline{\mu}_1 = \cdots = \underline{\mu}_g \quad$ vs $\quad H_1 : H_0^c$ equivalent to $H_0 : \underline{\tau}_1 = \cdots = \underline{\tau}_g \quad$ vs $\quad H_1 : H_0^c$

Test statistics proposals:

- #Wilks $\Lambda_W = \frac{det(W)}{det(W+B)} \Rightarrow$ reject H_0 if it is too small
- #Pillai $\Lambda_P = trace(B(B+W)^{-1}) \Rightarrow$ reject H_0 if it is too big
- #Hotelling-Lawley $\Lambda_{HL} = trace(BW^{-1}) \Rightarrow$reject H_0 if it is too big

Obs: each test statistics described before could be expressed in terms of the eigenvalues of BW^{-1}

Question: Which is the distribution of #Wilks statistics Λ_W if H_0 holds?

Theo - Bartlett asymptotic approximation

If H_0 holds then: $-(n-1-\frac{p+g}{2})\ln(\Lambda_W) \sim \chi^2(p(g-1))$
So reject H_0 at level α if $-(n-1-\frac{p+g}{2})\ln(\Lambda_W) > \chi^2_{1-\alpha}(p(g-1))$
If we reject H_0 we want to create a $CI(\tau_{i,l}-\tau_{k,l})$ with $i,k \in \{1,\dots,g\}$ and $l \in \{1,\dots,p\}$.
To do so firstly we need a point estimator for $\tau_{i,l}-\tau_{k,l}$.
Since $\underline{\tau}_i$ is estimated by $\underline{\overline{x}}_i - \underline{\overline{x}}$ we conclude that $\tau_{i,l}$ is estimated by $\overline{x}_{i,l}-\overline{x}_l$.
$\Rightarrow \tau_{i,l}-\tau_{k,l}$ is estimated by $\overline{x}_{i,l}-\overline{x}_{k,l} \sim N(\tau_{i,l}-\tau_{k,l}, \sigma_{l,l}(\frac{1}{n_i}+\frac{1}{n_k}))$ but we don't know $\sigma_{l,l}$ so we need to estimate it.
$\Rightarrow$ since Σ is estimated by $S_{pooled} = \frac{1}{n-g}\sum_{i=1}^g (n_i-1)S_i = \frac{W}{n-g}$ with
$S_i = \frac{1}{n_i-1}\sum_{i=1}^{n_i}(\underline{x}_{i,j}-\overline{x}_i)(\underline{x}_{i,j}-\overline{x}_i)'$ we use as estimator of $\sigma_{l,l}$ the $S_{pooled_{l,l}} = \frac{w_{l,l}}{n-g}$
Now we can create Bonferroni $CI_{1-\alpha}(\tau_{i,l}-\tau_{k,l}) = [\overline{x}_{i,l}-\overline{x}_l \pm t_{1-\frac{\alpha}{2Ⓢ}}(n-g)\sqrt{\frac{w_{ll}}{n-g}(\frac{1}{n_i}+\frac{1}{n_g})}]$ with as $Ⓢ = p\frac{g(g-1)}{2}$

Extension to two-way ANOVA.

Framework: we have 2 treatment (factors), the first treatment is made by g levels while the second one has b levels.
We have two possible models for our $\mu_{i,j}$:

1. Complete model: $\mu_{i,j} = \mu + \tau_i + \beta_j + \gamma_{i,j} \leftarrow \gamma_{i,j}$ model the relation between first and second treatment level.
2. Additive model: $\mu_{i,j} = \mu + \tau_i + \beta_j \leftarrow$ effect of first treatment are independent from the second treatment level

Remark: link useful to understand
Remark: $\overline{x}_{i,j} = \frac{1}{n}\sum_{k=1}^n x_{i,j,k}$
Remark: $\overline{x} = \frac{1}{n\cdot g\cdot b}\sum_{i=1}^g\sum_{j=1}^b\sum_{k=1}^n x_{i,j,k}$
Our estimators are:

- $\mu \to \overline{x}$
- $\tau_i \to \overline{x}_{i,\cdot} - \overline{x}$
- $\beta_j \to \overline{x}_{\cdot,j} - \overline{x}$
- $\gamma_{i,j} \to \overline{x}_{i,j} - (\overline{x}_{\cdot,j}-\overline{x}) - (\overline{x}_{i,\cdot}-\overline{x}) - \overline{x} = \overline{x}_{i,j} - \overline{x}_{\cdot,j} - \overline{x}_{i,\cdot} + \overline{x}$

Our constrains are:

- $\sum_i \tau_i = 0$
- $\sum_j \beta_j = 0$
- $\sum_i \gamma_{i,j} = 0 \quad \forall j$
- $\sum_j \gamma_{i,j} = 0 \quad \forall i$

Decomposition of variance

$$\sum_{i}^{g}\sum_{j}^{b}\sum_{k}^{n}(x_{i,j,k}-\overline{x})^2 =$$

$$\underbrace{\sum_{i}^{g} nb(\overline{x}_{i,\cdot}-\overline{x})^2}_{SS_{\text{treatment 1}}} + \underbrace{\sum_{j}^{b} ng(\overline{x}_{\cdot,j}-\overline{x})^2}_{SS_{\text{treatment 2}}} + \underbrace{\sum_{i}^{g}\sum_{j}^{b} n(\overline{x}_{i,j}-\overline{x}_{i,\cdot}-\overline{x}_{\cdot,j}+\overline{x})^2}_{SS_{\text{interactions}}} + \underbrace{\sum_{i}^{g}\sum_{j}^{b}\sum_{k}^{n}(x_{i,j,k}-\overline{x}_{i,j})^2}_{SS_{residuals}}$$

With:

- $SS_{\text{treatment 1}} =$ variability explain by treatment 1 has $g-1$ degree of freedom
- $SS_{\text{treatment 2}} =$ variability explain by treatment 2 has $b-1$ degree of freedom
- $SS_{\text{interactions}} =$ variability explain by treatment 1 and treatment 2 interactions has $(g-1)(b-1)$ degree of freedom
- $SS_{\text{residuals}} =$ variability remaining has $gb(n-1)$ degree of freedom

How to test if we need to model interactions: $H_0 : \gamma_{i,j} = 0 \quad \forall i,j \quad \text{vs} \quad H_1 : \exists \gamma_{i,j} \neq 0$

We reject H_0 if $\boxed{\frac{SS_{interact}}{(g-1)(b-1)}\frac{gb(n-1)}{SS_{res}} > F_{1-\alpha}((g-1)(b-1), gb(n-1))}$

If H_0 holds we can use additive model.

Remark: testing treatment one, similar to one-way ANOVA, is equal to check $\frac{SS_{\text{treatment 1}}}{g-1}\frac{gb(n-1)}{SS_{res}} > F_{1-\alpha}(g-1, gb(n-1))$

Classification

Framework: each unit is represented by $(\underline{x}', L)$ where L is the label while $\underline{x}'$ are features

Goal: learning $\delta : X \to \text{Label-space}$

There are 2 possible ways to approaches to this problem:

1. **Discriminant Analysis - Supervised Learning:** we train our model starting from a training set with features $\underline{x}'$ and labels L;
2. **Cluster Analysis - Unsupervised Learning:** we don't know a-priori labels so we cluster together units with similar features.

#Discriminant_Analysis - Supervised Learning

To use this approach we need 3 ingredients:

- *Distribution of features*: $\underline{x}|L = i \sim f_i(\underline{x}) \leftarrow$ within-class density of features given the class label, used into calculation of likelihood;
- *Prior distribution*: $\mathbb{P}[L = i] = p_i \quad \text{with } i \in \{1, \dots, g\} \leftarrow$ represents the initial belief about the probability of each class occurring before observing any data, used into calculation of posterior probabilities;
- *Cost of mis-classification*: $c(i|j) =$ cost of attributing unit to group i while it belongs to group j

Remark: we usually have a costs matrix C and it would be desirable if $diag(C) = c(i,i) = 0$

Remark: learning a classification function δ is equivalent to a learn a partition $\{R_1, \dots, R_g\}$ of X

s.t. $R_i \cap R_j = \emptyset$ if $i \neq j$ and $\cup_i^g R_i = X$
Remark: usually we want to minimize the #ECM (Expected Cost of Misclassification)

Example - dichotomous classifier

$\delta \to \{R_1, R_2\}$ so if $\delta(\underline{x}) = 1$ implies that $\underline{x} \in R_1$

$$\begin{aligned} ECM(\delta) &= \int_{R_2} c(2|1) f_1(\underline{x}) p_1 d\underline{x} + \int_{R_1} c(1|2) f_2(\underline{x}) p_2 d\underline{x} \\ &= \underbrace{\int_X c(2|1) f_1(\underline{x}) p_1 d\underline{x}}_{c(2|1)p_1} - \int_{R_1} c(2|1) f_1(\underline{x}) p_1 d\underline{x} + \int_{R_1} c(1|2) f_2(\underline{x}) p_2 d\underline{x} \\ &= c(2|1)p_1 + \int_{R_1} (-c(2|1) f_1(\underline{x}) p_1 + c(1|2) f_2(\underline{x}) p_2) d\underline{x} \end{aligned}$$

δ optimal is the one which minimize $ECM(\delta)$ so:
$R_1 = \{\underline{x} \in X : c(2|1) f_1(\underline{x}) p_1 \geq c(1|2) f_2(\underline{x}) p_2\}$
$R_2 = \{\underline{x} \in X : c(2|1) f_1(\underline{x}) p_1 \leq c(1|2) f_2(\underline{x}) p_2\}$

> **Idea:**
> $\mathbb{P}(X = \underline{x}, cls = 1) = f_1(\underline{x}) p_1$
> => $\int_{R_2} c(2|1) f_1(\underline{x}) p_1 d\underline{x} = \int_{R_2} c(2|1) \mathbb{P}(X = \underline{x}, cls = 1) d\underline{x}$ = probs of misclassify all the obs of the class 1 as class 2

Obs:

$$\delta(\underline{x}) = t \in \{1, \dots, g\} \Leftrightarrow \frac{1}{\sum_{l=1}^g f_l(\underline{x}) p_l} \sum_{k \neq t} c(t|k) f_k(\underline{x}) p_k \leq \frac{1}{\sum_{l=1}^g f_l(\underline{x}) p_l} \sum_{k \neq j} c(j|k) f_k(\underline{x}) p_k \quad \forall j \neq t$$

> **Idea:**
> Recalling that: $\frac{f_k(\underline{x}) p_k}{\sum_{j=1}^g f_j(\underline{x}) p_j} = \frac{\mathbb{P}[\underline{x}|L=k]\mathbb{P}(L=k)}{\sum_{j=1}^g \mathbb{P}[\underline{x}|L=j]\mathbb{P}(L=j)} = \mathbb{P}[L = k|\underline{x}] =$ posterior probability
> So, we reach:
> $\delta(\underline{x}) = t \Leftrightarrow \sum_{k \neq t} c(t|k) \mathbb{P}[L = k|\underline{x}] \leq \sum_{k \neq j} c(j|k) \mathbb{P}[L = k|\underline{x}] \quad \forall j \neq t$
> We are minimize ECM (Excpected Cost of Misclassification).

Example - Bayes classifier

$c(i, j) = d = constant \geq 0 \quad \forall i \neq j$ and $c(i, i) = 0 \quad \forall i$
Optimal δ:
$\delta(\underline{x}) = t \Leftrightarrow \sum_{k \neq t} \mathbb{P}[L = k|\underline{x}] \leq \sum_{k \neq j} \mathbb{P}[L = k|\underline{x}] \quad \forall j \neq t$

> **Idea:**
> We are minimizing the sum of the posterior probabilities of assign a datum to a wrong cluster.

Example - Maximum Likelihood classifier

$c(i,j) = d = constant \geq 0 \quad \forall i \neq j; c(i,i) = 0 \quad \forall i; p_1 = p_2 = \cdots = \frac{1}{g}$

Optimal δ:

$\delta(\underline{x}) = t \Leftrightarrow \frac{f_t(\underline{x})p_t}{\sum_{j=1}^{g} f_j(\underline{x})p_j} \geq \frac{f_k(\underline{x})p_k}{\sum_{j=1}^{g} f_j(\underline{x})p_j} \Leftrightarrow f_t(\underline{x}) \geq f_k(\underline{x}) \quad \forall k \neq t$

If the features are Gaussian $[\text{ so } \underline{x}|L \sim N_p(\mu_i, \Sigma_i)]$ implies that the Bayes classifier become:

$$\delta(\underline{x}) = t \Leftrightarrow \sum_{k \neq t} \mathbb{P}[L = k|\underline{x}] \leq \sum_{k \neq j} \mathbb{P}[L = k|\underline{x}] \quad \forall j \neq t$$

$$\rightarrow 1 - \sum_{k \neq t} \mathbb{P}[L = k|\underline{x}] \geq 1 - \sum_{k \neq j} \mathbb{P}[L = k|\underline{x}] \quad \forall j \neq t$$

$$\rightarrow \mathbb{P}[L = t|\underline{x}] \geq \mathbb{P}[L = j|\underline{x}] \quad \forall j \neq t$$

$$\rightarrow f_t(\underline{x})p_t \geq f_j(\underline{x})p_j \quad \forall j \in \{1, \ldots, g\}$$

$$\rightarrow p_t \frac{\exp\{-\frac{1}{2}(\underline{x} - \underline{\mu}_t)'\Sigma_t^{-1}(\underline{x} - \underline{\mu}_t)\}}{[(2\pi)^p \det(\Sigma_t)]^{\frac{1}{2}}} \geq p_j \frac{\exp\{-\frac{1}{2}(\underline{x} - \underline{\mu}_j)'\Sigma_j^{-1}(\underline{x} - \underline{\mu}_j)\}}{[(2\pi)^p \det(\Sigma_j)]^{\frac{1}{2}}}$$

$$\rightarrow \ln(p_t) - \frac{1}{2}\ln(\det(\Sigma_t)) - \frac{1}{2}d_{\Sigma_t^{-1}}(\underline{x}, \underline{\mu}_t) \geq \ln(p_j) - \frac{1}{2}\ln(\det(\Sigma_j)) - \frac{1}{2}d_{\Sigma_j^{-1}}(\underline{x}, \underline{\mu}_j)$$

Obs:

If $p_1 = p_2 = \cdots = p_g = \frac{1}{g}$ and $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_g$ then our classifier become:

$\delta(\underline{x}) = t \Leftrightarrow \underline{x} \in R_t \Leftrightarrow d_{\Sigma^{-1}}(\underline{x}, \underline{\mu}_t) \leq d_{\Sigma^{-1}}(\underline{x}, \underline{\mu}_j)$

Interpretation: we are assigning a point to the group with the closest mean in a distribution sense, since the distance cited above is the #Mahalanobis_distance.

QDA

Hypothesis: Gaussianity

We call **Quadratic Discriminant Function** $= d_t^Q(\underline{x}) = \ln(p_t) - \frac{1}{2}\ln(\det(\Sigma_t)) - \frac{1}{2}d_{\Sigma_t^{-1}}(\underline{x}, \underline{\mu}_t)$

So we can rewrite our Bayes classifier with Gaussian features using the previous definition:

$\delta(\underline{x}) = t \Leftrightarrow \underline{x} \in R_t \Leftrightarrow d_t^Q(\underline{x}) \geq d_j^Q(\underline{x}) \leftarrow$ which is *Quadratic Discriminant Analysis* (#QDA)

Interpretation: we are maximizing log-likelihood

LDA

Hypothesis: Gaussianity and Homoscedasticity

$\underline{x}\Sigma^{-1}\underline{\mu}_t + \ln(p_t) - \frac{1}{2}\underline{\mu}_t\Sigma^{-1}\underline{\mu}_t \geq \underline{x}\Sigma^{-1}\underline{\mu}_j + \ln(p_j) - \frac{1}{2}\underline{\mu}_j\Sigma^{-1}\underline{\mu}_j \quad \forall j \neq t$

We call **Linear Discriminant Function** $= d_t(\underline{x}) = \underline{x}\Sigma^{-1}\underline{\mu}_t + \ln(p_t) - \frac{1}{2}\underline{\mu}_t\Sigma^{-1}\underline{\mu}_t$

So we can rewrite our Bayes classifier with Gaussian features using the previous definition:

$\delta(\underline{x}) = t \Leftrightarrow \underline{x} \in R_t \Leftrightarrow d_t(\underline{x}) \geq d_j(\underline{x}) \leftarrow$ which is *Linear Discriminant Analysis* (#LDA)

Remark: We use the training set to estimate $f_i(\underline{x}) \quad \forall i$

#LDA: $\mu_1, \ldots, \mu_g, \Sigma$ should be estimate from data

#QDA : $\mu_1, \dots, \mu_g, \Sigma_1, \dots, \Sigma_g$ should be estimate from data
So we estimate:

- μ_i with $\underline{\overline{x}}_i = \frac{1}{n_i} \sum_{\{j:e_j=i\}} \underline{x}_j$
- Σ_i with $S_i = \frac{1}{n_i - 1} \sum_{\{j:e_j=i\}} (\underline{x}_j - \underline{\overline{x}}_i)(\underline{x}_j - \underline{\overline{x}}_i)' \rightarrow$ for QDA
- Σ with $S_{\text{pooled}} = \frac{1}{n-g} \sum_i (n_i - 1) S_i \rightarrow$ for LDA with $n = n_1 + n_2 + \dots + n_g$

Since we have a lot of things to estimate we need that $n_1, n_2, \dots, n_g$ is large with respect to p.

Naive Bayes Classifier

A classifier which not require a large sample size training set is **Naive Bayes Classifier** which parameterize Σ_i as a diagonal matrix so $d_t^Q(\underline{x})$ become:
$d_t^Q(\underline{x}) = \ln(p_t) - \frac{1}{2} \sum_{i=1}^p \ln(\sigma_{ii}^{(t)}) - \frac{1}{2} \sum_{i=1}^p \frac{(x_i - \overline{x}_{t,i})^2}{\sqrt{\sigma_{i,i}^{(t)}}}$ with $\sigma_{i,i}^{(t)} = \frac{1}{n_t - 1} \sum_{\{j:e_j=i\}} (\underline{x}_{j,i} - \overline{x}_{t,i})^2$

Example #knn - classifier

Fix $k \geq 1$ then $N_k(\underline{x}) = \{ \text{k units } \underline{x}_i \text{ which are closest to } \underline{x} \}$
We assign the label t to $\underline{x}$ if it is the most frequent label in $N_k(\underline{x})$

Fisher argument for #LDA

Goal: We develop the #LDA classifier framework without assuming Gaussinity.

> **Idea:** It is a method designed to find a linear combination of features that separates two or more classes of objects or events. The core idea behind Fisher's argument for LDA is to project high-dimensional data onto a lower-dimensional space in such a way that maximizes the separability among known categories.

Framework: LDA framework
Define $B = \text{Variance matrix between group means} = \frac{1}{g-1} \sum_{i=1}^g (\underline{\mu}_i - \underline{\overline{\mu}})(\underline{\mu}_i - \underline{\overline{\mu}})'$ with
$\underline{\overline{\mu}} = \frac{1}{g} \sum_{i=1}^g \underline{\mu}_i$ and as $\underline{\mu}_i = \mathbb{E}[\underline{x} | L = i]$
So with $\underline{a} \in \mathbb{R}^p$:

- $\mathbb{E}[\underline{a}' \underline{x} | L = i] = \underline{a}' \underline{\mu}_i$
- $var[\underline{a}' \underline{x} | L = i] = \underline{a}' \Sigma \underline{a}$

Problem: find $\underset{\underline{a} \in \mathbb{R}^p}{arg\,max} \frac{\underline{a}' B \underline{a}}{\underline{a}' \Sigma \underline{a}} = \underset{\underline{a} \in \mathbb{R}^p}{arg\,max} \frac{\frac{1}{g-1} \sum_{i=1}^g (\underline{a}' \underline{\mu}_i - \underline{a}' \underline{\overline{\mu}})^2}{\underline{a}' \Sigma \underline{a}}$

Solution: same reasoning as #PCA we find that $\underset{\underline{a} \in \mathbb{R}^p}{arg\,max} \frac{\underline{a}' B \underline{a}}{\underline{a}' \Sigma \underline{a}} = \Sigma^{-\frac{1}{2}} \underline{e}_1$ with $\underline{e}_i$ are the eigenvectors of $\Sigma^{-\frac{1}{2}} B \Sigma^{-\frac{1}{2}}$.
We can rewrite $\underline{x}$ using $\underline{a}_i$ as reference system so $\underline{x} \rightarrow \tilde{\underline{x}} \Rightarrow \underline{a}_i' \underline{x}$ are called *Fisher Discriminant Scores* and have a useful feature: $cov(\underline{a}_i \underline{x}, \underline{a}_j \underline{x}) = \begin{cases} 0 & i \neq j \\ 1 & \text{else} \end{cases} \Rightarrow cov(A\underline{x}) = I$

Interpretation: how well-separated the classes are along the FLD axis.

So, how to build a classifier?
We estimate:

- μ_i with $\overline{\underline{x}}_i = \frac{1}{n_i} \sum_{\{j:e_j=i\}} \underline{x}_j$
- Σ_i with $S_i = \frac{1}{n_i - 1} \sum_{\{j:e_j=i\}} (\underline{x}_j - \overline{\underline{x}}_i)(\underline{x}_j - \overline{\underline{x}}_i)'$
- Σ with $S_{\text{pooled}} = \frac{1}{n-g} \sum_i (n_i - 1) S_i \rightarrow$ for LDA with $n = n_1 + n_2 + \cdots + n_g$

So our $\hat{B} = \frac{1}{g-1} \sum_{i=1}^{g} (\underline{x}_i - \overline{\underline{x}})(\underline{x}_i - \overline{\underline{x}})'$
So: $\delta(\underline{x}) = t \Leftrightarrow \sum_{i=1}^{k} (\tilde{x}_i - \tilde{x}_{t,i})^2 \leq \sum_{i=1}^{k} (\tilde{x}_i - \tilde{x}_{j,i})^2$
Interpretation: we assign $\underline{x}$ to the closest projection on the Fisher Discriminant Scores.

Evaluating a classifier

framework: $\mathbb{X} = \text{training set} = \begin{bmatrix} \vdots & \vdots \\ \underline{x}_k & l_k \\ \vdots & \vdots \end{bmatrix}$ with $\underline{x}_k \in \mathbb{R}^p$ and $l_k \in \{1, 2, \ldots, g\}$

Recall:
$AER(\delta) = \text{Actual Error Rate} = \sum_{k \neq 1} \int_{\mathbb{R}_k} f_1(\underline{x}) p_1 d\underline{x} + \sum_{k \neq 2} \int_{\mathbb{R}_k} f_2(\underline{x}) p_2 d\underline{x} + \cdots + \sum_{k \neq g} \int_{\mathbb{R}_k} f_g(\underline{x}) p_g d\underline{x}$
Q: How to evaluate a classifier? How to estimate $AER(\delta)$ from data?
For sake of simplicity our reasoning are in the framework of $g = 2$. So, the correct
$AER(\delta) = \sum_{k \neq 1} \int_{\mathbb{R}_k} f_1(\underline{x}) p_1 d_{\underline{x}} + \sum_{k \neq 2} \int_{\mathbb{R}_k} f_2(\underline{x}) p_2 d_{\underline{x}}$
How to estimate it from data?

- Apply δ to $\mathbb{X}$;
- Construct the confusion matrix;
- $\hat{AER}(\delta) = APER(\delta) = \text{APpearent Error Rate} = \frac{n_{1,2} + n_{2,1}}{n} \leftarrow$ spoiler: is too optimistic

Remember that the confusion matrix is construct as follow:

$\downarrow$ truth \ estimated $\rightarrow$	1	2	
1	$n_{1,1}$	$n_{1,2}$	$n_{1,\cdot}$
2	$n_{2,1}$	$n_{2,2}$	$n_{2,\cdot}$
	$n_{\cdot,1}$	$n_{\cdot,2}$	

With a bit of computation we reach that:
$APER(\delta) = \hat{p_1} \int_{\mathbb{R}_2} \widehat{f_1(\underline{x})} d\underline{x} + \hat{p_2} \int_{\mathbb{R}_1} \widehat{f_2(\underline{x})} d\underline{x} = \frac{n_{1,\cdot}}{n} \frac{n_{1,2}}{n_{1,\cdot}} + \frac{n_{2,\cdot}}{n} \frac{n_{2,1}}{n_{2,\cdot}}$

Obs

In case of dichotomous $\{0, 1\}$-classifier we know this metrics coming from the confusion matrix:

- **precision = PPV = positive predicted value:** $\frac{n_{1,1}}{n_{\cdot,1}} \Rightarrow \text{precision} \uparrow \text{false positive} \downarrow$
- **APER=** $\frac{n_{1,0} + n_{0,1}}{n}$
- **recall = sensitivity =** $\frac{n_{1,1}}{n_{1,\cdot}} \Rightarrow \text{recall} \uparrow \text{false negative} \downarrow$
- **specificity =** $\frac{n_{0,0}}{n_{\cdot,0}}$

To improve quality of $A\hat{E}R(\delta)$ we can:

- Construct confusion matrix on a test set;
- Use cross-validation on $\mathbb{X}$ to extract from it the test set (Leave-One-Out) and repeat m-times;
- k-fold cross validation: split $\mathbb{X}$ into k subset and use cross validation inside each subset, repeat it $B-$times.

Using $k-$fold cross validation we are decreasing the correlation between models trained so the variance of estimators given by this technique is lower than the one given by the cross-validation Leave-One-Out.

Support Vector Machine #SVM

Main contributor to this theory was Vapnick in '90.

Framework: We have a dychotomous problem and $\mathbb{X}$ is our dataset.

Idea: Find an hyperplane s.t. separate points which belongs into group 1 to ones belonging group 2.

What is an hyperplane?

$\mathcal{L}$:= hyperplane in $\mathbb{R}^p$:= affine subspace of dimension $p-1$

So $\exists \underline{\beta} \in \mathbb{R}^p$ with $||\underline{\beta}|| = 1$ s.t. $\underline{\beta} \perp \mathcal{L}$

Let be $\underline{x}_0 = span\{\underline{\beta}\} \cap \mathcal{L}$ and $\beta_0 = ||\underline{x}_0|| \leftarrow$ so where $\underline{\beta}$ and $\mathcal{L}$ meet

Given $\underline{x} \in \mathbb{R}^p$ we know that:

$$\underline{x} \in \mathcal{L} \quad \Leftrightarrow \quad \pi_{\underline{x}|\underline{\beta}} = \underline{x}_0 = \frac{\underline{\beta} \cdot \underline{\beta}'}{\underline{\beta}' \cdot \underline{\beta}} \cdot \underline{x} \quad \leftrightarrow \quad \underline{\beta}' \cdot \underline{x} = \beta_0$$

Question 1: Given $A, B \in \mathbb{R}^p$ when can I separate them with an hyperplane?

Let $CH(A), CH(B)$ be the convex-hull generated by A and B.

Assumptions (Geometric Hahn-Banach theorem):

- $CH(A) \neq \emptyset$, $CH(B) \neq \emptyset$;
- $CH(A) \cap CH(B) = \emptyset$
- Either $CH(A)$ or $CH(B)$ is open
  $\Rightarrow$ $\exists$separating hyperplane

Obs:

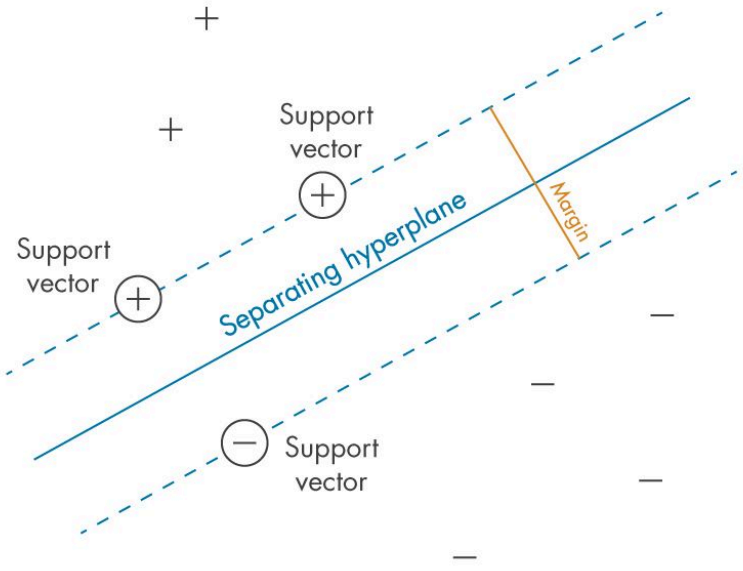
3^{rd} assumption could be rewrite as:

$CH(A)$ and $CH(B)$ are closed and at least one of them is compact

Question 2: Find the best hyperplane $\mathcal{L}$

Assume $\mathbb{X}$ is s.t. $\exists \mathcal{L}$ separating classes 1 from classes 2.

We define $\mathbb{R}^+$ and $\mathbb{R}^-$ the half planes divided by $\mathcal{L}$.



Let $\underline{y}_i = \pm 1$ and if it positive $\underline{x}_i$ belongs to class 1 otherwise belongs to class 2.
Define $\text{Margin} = M_1 := min_{\forall i=1,\dots,n}\{\underline{y}_i \cdot (\underline{\beta}' \cdot \underline{x}_i - \underline{\beta}_0)\}$

The best $\mathcal{L}$ is that one which maximize M_1.

The problem to solve is:

$$\begin{cases} max_{\underline{\beta},\underline{\beta}_0} M \\ ||\underline{\beta}|| = 1 \\ \underline{y}_i \cdot (\underline{\beta}' \cdot \underline{x}_i - \underline{\beta}_0) \geq M \qquad \forall i \end{cases}$$

This problem is the most strict version of our problem so is called *hard problem*.
Solution:

$$\underline{\hat{\beta}} = \sum_{i=1}^{n} \hat{\lambda}_i y_i \underline{x}_i, \quad \hat{\beta}_0 \in \mathbb{R} \quad \Rightarrow \quad \hat{f}(\underline{x}) = \underline{\hat{\beta}}' \underline{x} - \hat{\beta}_0 = \sum_{i=1}^{n} \hat{\lambda}_i y_i \underline{x}_i' \cdot \underline{x}_i - \hat{\beta}_0$$

So our classifier became $c(\underline{x}) = sign\{\hat{f}(\underline{x})\}$
For that we define $\text{support} := \{\text{points which find the hyperplane}\} \subset \mathbb{X}$
If our data are so mixed we have 3 approach to try to find a classifier:

- Using *soft problem*, so the last request in the hard problem becomes

  $$\underline{y}_i \cdot (\underline{\beta}' \cdot \underline{x}_i - \underline{\beta}_0) \geq (1 - \epsilon_i) M \qquad \forall i$$

  with $\epsilon_i \geq 0$ and $\sum_{i=0}^{n} \leq c$ where c is called *budget constrain*.
  Interpretation: we are admitting that some point of one group could be in the half space of the other and viceversa.
- Transforming our data, or our axis, in such a way that the hyperplane become clear.
- Instead of using $\underline{x}_i' \cdot \underline{x}_i$ we can define a new kernel $k(\underline{x}, \underline{w}) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ so we can write $\hat{f}(\underline{x}) = \sum_{i=1}^{n} \hat{\lambda}_i y_i k(\underline{x}_i, \underline{x}_i) - \hat{\beta}_0$
  Interpretation: kernel function are the tool with we transform our data such that is more clear the dividing hyperplane.

Example - kernels

- d-degree polynomials = $k(\underline{x}, \underline{w}) = (1 + \underline{x}'\underline{w})^d$
- radial-basis function = $k(\underline{x}, \underline{w}) = \exp\{-\gamma||\underline{x} - \underline{w}||^2\}$
- $k(\underline{x}, \underline{w}) = tanh\{k_1 \cdot \underline{x}'\underline{w} + k_2\}$ ← common in neural networks

Unsupervised Learning - #Cluster-analysis

It is useful when given $\mathbb{X}$ neither labels associate or the number of groups in the dataset.
Idea: unit in the same cluster are more similar than units in different clusters.
Firstly, we need a measure of similarity/dissimilarity. It should have the following properties:

- $d(\underline{x}, \underline{x}) = 0 \qquad \forall \underline{x} \in \mathbb{R}^p$
  - Stronger version: $d(\underline{x}, \underline{y}) = 0 \Leftrightarrow \underline{x} = \underline{y}$
- $d(\underline{x}, \underline{y}) = d(\underline{y}, \underline{x}) \qquad \forall \underline{x}, \underline{y} \in \mathbb{R}^p$
- $d(\underline{x}, \underline{y}) \leq d(\underline{x}, \underline{z}) + d(\underline{z}, \underline{y}) \qquad \forall \underline{x}, \underline{y}, \underline{z} \in \mathbb{R}^p$
  - Stronger version: $d(\underline{x}, \underline{y}) \leq max\{d(\underline{x}, \underline{z}), d(\underline{z}, \underline{y})\} \qquad \forall \underline{x}, \underline{y}, \underline{z} \in \mathbb{R}^p$

If our similarity function has all the stronger version of each condition it is an **ultra metrics**.
If our similarity function has all conditions and the first one in the stronger version it is a **metrics**.
If our similarity function has all conditions in their base form it is a **pseudo metrics**.

Example - Ultrametric

Let $\underline{x} = \{\alpha, \beta, \gamma, \sigma, \epsilon, \theta\}, \quad \underline{y} = \{\alpha, \beta, \gamma, \omega, \nu, \mu\}$
Define $d(\underline{x}, \underline{y}) = \frac{1}{2^n}$ where n is the index of the first element which differs in the objects. In this case is 4.
Some trivial distances with $\underline{x}, \underline{y} \in \mathbb{R}^p$:

- #Euclidean_distance : $d(\underline{x}, \underline{y}) = \sqrt{(x-y)'(x-y)}$
- #Mahalanobis_distance : $d(\underline{x}, \underline{y}) = \sqrt{(x-y)'\Sigma^{-1}(x-y)}$
- #Minkowsky_distance : $d^r(\underline{x}, \underline{y}) = \sum_{i=1}^{p} |x_i - y_i|^r$
- #Camberra_distance : $d(\underline{x}, \underline{y}) = \sum_{i=1}^{p} \frac{|x_i - y_i|}{x_i + y_i}$

Some trivial distances with $\underline{x}, \underline{y} \in \{0, 1\}^p$:

- #Euclidean_distance : $d(\underline{x}, \underline{y}) = \sqrt{(x-y)'(x-y)}$ (which indicates the number of discordances)

From contingency matrix:

	0	1
0	a	b
1	c	d

- $d(\underline{x}, \underline{y}) = 1 - \frac{a}{p}$ where p is the total number of observations

- $d(\underline{x}, \underline{y}) = \sqrt{2 \cdot (1 - corr(x, y))}$

It's very useful to create the D matrix with distances between each x_i and all other x_j.
$\Rightarrow D$ has the following properties:

- Each element on the diagonal is 0;
- It's symmetric if d is symmetric and reflexive.

Distances between finite subset of $\mathbb{R}^p$

Let U, V be two finite subset (cluster) of $\mathbb{R}^p$.
How to calculate $d(U, V)$?
Possible solutions are :

- #Single_linkage : $d(U, V) = \min\{d(\underline{x}, \underline{y}) : \underline{x} \in U, \underline{y} \in V\}$
- #Complete_linkage : $d(U, V) = \max\{d(\underline{x}, \underline{y}) : \underline{x} \in U, \underline{y} \in V\}$
- #Average_linkage : $d(U, V) = \frac{1}{\#U \cdot \#V} \sum_{\underline{x} \in U, \underline{y} \in V} d(\underline{x}, \underline{y})$
- #Centroid_distance : $d(U, V) = d(\underline{c}_U, \underline{c}_V)$ with $\underline{c}_U$ and $\underline{c}_V$ the centroid of U and V

Hierarchical Agglomerative Cluster Algorithms

Let D be the distances matrix and chosen a linkage #Hierarchical_Agglomerative_Cluster works in this way:

```
Initialization: each unit is a Cluster
while(!convergence condition reached){
        cluster together 2 closest clusters;
        update D;
}
```

Let $\mathbb{X}$ with $\underline{x}_i \in \mathbb{R}^p$ be the training set and choose $d : \mathbb{R}^p \times \mathbb{R}^p \to [0, +\infty)$ as pseudo-metric, metric and ultra-metric.

Hierarchical Clustering - #Ward_method

Hypothesis: d must be the #Euclidean_distance, so we use: $d^2(\underline{x}, \underline{y}) = (\underline{x} - \underline{y})'(\underline{x} - \underline{y})$
We call C a cluster of finite point in $\mathbb{R}^p$, it is the area where we classify the point of $\mathbb{X}$ with a certain label.
Since we want to find: $\forall i \quad \overline{\underline{x}}_i = argmin_{\underline{x} \in \mathbb{R}^p} \sum_{\underline{x}_i \in C_i} d^2(\underline{x}, \underline{x}_i) = \frac{1}{|C_i|} \sum_{\underline{x}_i \in C_i} \underline{x}_i =$ barycentre of C_i
From that we can define the #error_sum_squares $ESS_i = \sum_{\underline{x} \in C_i} d^2(\underline{x}, \overline{\underline{x}}_i)$ which describe the variability of all the data $\underline{x}$ in our cluster C_i around the barycentre $\overline{\underline{x}}_i$.
So we can create a cumulative index called $ESS = \sum_i ESS_i$

```
Initialization: each unit is a cluster so ESS = 0
while(!convergence condition reached){
```

```
        merge together 2 clusters which imply the minimum ESS increase;
}
```

Not Hierarchical Clustering

Goal: cluster $\mathbb{X}$ in k clusters s.t. $\mathbb{X} = \cup_i^k C_i$ and $\forall i \neq j \quad C_i \cap C_j = \emptyset$

#K-means

It' s a generalization of #Ward_method with d different from the euclidean.
Calculate $\overline{\underline{x}}_i = argmin_{\underline{x} \in \mathbb{R}^p} \sum_{\underline{x}_i \in C_i} d^2(\underline{x}, \underline{x}_i) =$ centroid of C_i
Goal: find C_i s.t. $min\{\sum_i^k \sum_{\underline{x}_i \in C_i} d^2(\underline{x}, \overline{\underline{x}}_i)\} = min\{\sum_i^k ESS_i\}$

```
Initialization: create k random centroid
or
Initialization: create k random subset

While(centroid change){
        Compute centroid for each cluster;
        Assign every unit to cluster with the nearest centroid;
}
```

Since the computation of centroid is a difficult problem we can simplify this step calculating medoids so become $\overline{\underline{x}}_i = argmin_{\underline{x} \in \mathbb{X}} \sum_{\underline{x}_i \in C_i} d^2(\underline{x}, \underline{x}_i)$ and the algorithm change also names: $\text{k-means} \rightarrow \text{k-medoids}$

How to choose k?

Secchi's tips:

- If you have a dendrogram choose a cut-point s.t. clusters are robust to some oscillation of it, don't choose it near a split;
- Express the cost function $ESS = \sum_i^k \sum_{\underline{x}_i \in C_i} d^2(\underline{x}, \overline{\underline{x}}_i)$ in function of k and choose the k with the elbow rule (if the increasing of k is longer than the decrease of the cost function it's time to stop)

Problems

Using Hierarchical or Not-Hierarchical cluster we have some problems:

- #Complete_linkage, #Average_linkage, #K-means, #Ward_method tend to create ellipsoidal clusters in the metric choosen;
- #Single_linkage has the problem of the chain effect;

Non-parametric density-based clustering

What is neighbourhood of $\underline{x}$ in $\mathbb{R}^p$?

We call:
ϵ-neighbourhood of $\underline{x} = N_\epsilon(\underline{x}) = \{\underline{y} \in \mathbb{R}^p : d(\underline{x}, \underline{y}) < \epsilon\}$
Which is a sphere in the sense of d chosen.
We also define $|N_\epsilon(\underline{x})| = \#\{\text{points } \in \mathbb{X}, \in N_\epsilon(\underline{x})\}$

Density-Based Spatial Clustering of Applications with Noise #DBSCAN

Idea: cluster are region with high density of points, low density areas are boundaries between clusters, noises or outliers.
We have to fix in advance:

- Radius $\epsilon > 0$;
- minPts $\in \mathbb{N} \setminus \{0\}$

After that we have to define some names:

- x is called *core-point* if $|N_\epsilon(\underline{x})| \geq minPts$;
- x is called *border-point* if $|N_\epsilon(\underline{x})| < minPts$ but $\exists \underline{x}_j$ core-point s.t. $\underline{x}_i \in N_\epsilon(\underline{x}_j)$;
- x is called *outlier* in every other cases.

As last definitions we need:

- $\underline{x}_j$ is *directly density reachable* from $\underline{x}_i$ if: $\underline{x}_j \in N_\epsilon(\underline{x}_i)$ with $\underline{x}_i$ a core-point;
- $\underline{x}_j$ is *density reachable* from $\underline{x}_i$ if there are $\underline{y}_{i \in \{1,\dots,k\}}$ with $k \geq 0$ s.t.:
  - $\underline{y}_{i \in \{1,\dots,k-1\}}$ are core points;
  - $\underline{y}_1 = \underline{x}_i$ and $\underline{y}_k = \underline{x}_j$
  - $\underline{y}_j \in N_\epsilon(\underline{y}_{j-1}) \quad \forall j \in \{1, \dots, k\}$
- $\underline{x}_j$ is *density connected* from $\underline{x}_i$ if there is an $\underline{x}$ s.t. both $\underline{x}_i$ and $\underline{x}_j$ are density reachable from $\underline{x}$

DBSCAN identifies a cluster $C \subset \mathbb{X}$ s.t.:

1. if $\underline{x}_j$ is density reachable from $\underline{x}_i \in C \Rightarrow \underline{x}_j \in C$;
2. $\forall \underline{x}_i, \underline{x}_j \in C \quad \underline{x}_i, \underline{x}_j$ must be density connected.

MultiDimensional Scaling #MDS

Create the distance matrix D
Goal: create a new dataset $\tilde{\mathbb{X}}$ with $\underline{y}_i \in \mathbb{R}^q$ and $q << p$ where we can use #Euclidean_distance and $d_{euclidean}(\underline{y}_i, \underline{y}_j) = \delta_{i,j} \simeq d_{ij} \in D$
There are two approches to solve this problem:

- *Classical MDS:* find $\underline{y}_{i \in \{1,\dots,k\}} = argmin \sum_{i \neq j} (d_{i,j} - \delta_{i,j})^2$
- *Kruskal MDS:* find $\underline{y}_{i \in \{1,\dots,k\}} = argmin \underbrace{\frac{\sum_{i \neq j} (\theta(d_{i,j}) - \delta_{i,j})^2}{\sum_{i \neq j} \delta_{i,j}}}_{\text{stress function}}$ with $\theta : \mathbb{R} \to \mathbb{R}$ monotone

Remark: in the classical MDS, if D is done with euclidian distance we can use #PCA to capture the first q variables which better describe the variability of our data. So become the same thing.

Regression

Recall

We call *target variable* $y \in \mathbb{R}$ while vector of *features* $\underline{x} \in \mathbb{R}^p$.

Goal: explain y variability as $f(\underline{x})$.

Remember that: #regression_function = $\mathbb{E}[y|\underline{x}] : \mathbb{R}^p \to \mathbb{R}$

Rmk: if $(y, \underline{x}) \sim N_{p+1} \Rightarrow \mathbb{E}[y|\underline{x}] = \beta_0 + \beta_1 x_1 + \ldots \leftarrow$ is linear to the regressor;

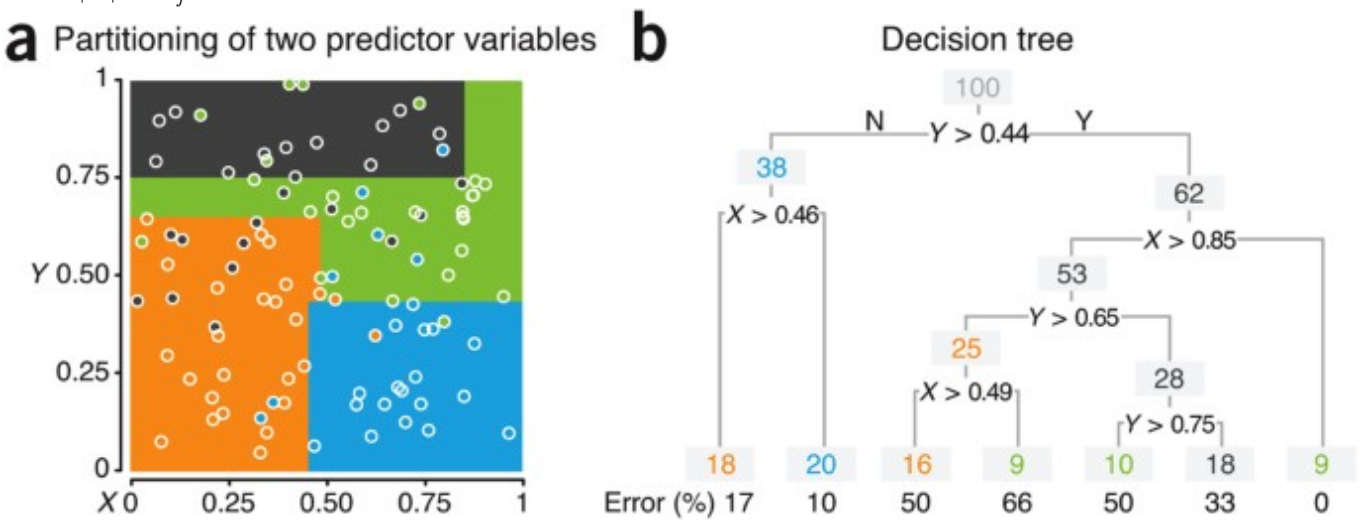
The general model is $y = \mathbb{E}[y|\underline{x}] + \epsilon$ and we want to find $\hat{f}(\underline{x})$ to estimate the regression function.

There main categories of method to do it:

- *totally data driven:* we haven't a model of f, is very good for prediction but really bad to interpret and is often used in data-mining and machine learning approches. (CART)
- *structured model:* we know how looks like f and we use data to find unknown parameters of the model chosen.

Classification And Regression Trees #CART

Idea: split features space $\mathbb{R}^p$ into j partitions $R_1, R_2, \ldots, R_j$ and predict y in R_i as $\overline{y}_i = \frac{1}{|R_i|} \sum_{\underline{x}_j \in R_i} y_i$



The main difficult is to find partitions $R_1, R_2, \ldots, R_j$ with following properties:

- $\cup_{k=1}^{j} R_k = \mathbb{R}^p$;
- $R_i \cap R_j = \emptyset \quad \forall i \neq j$;
- $R_i \cap \mathbb{X} \neq \emptyset \quad \forall i \rightarrow$ to ensure the existence of the mean $\overline{y}_i$

Optimal Criterion: find j and $R_1, R_2, \ldots, R_j = argmin\{\sum_{k=1}^{j} \sum_{\underline{x}_i \in R_k} (y_i - \overline{y}_i)^2\}$

This optimal criteria is NP-complete so CART suppose that $R_1, R_2, \ldots, R_j$ are rectangles.

We have to introduce the following function depending of the split (s_k^*):

$$u(s_k^*) = \underbrace{\sum_{i=1}^{n}(y_i - \overline{y}_k)^2}_{\text{variance pre-split}} - \underbrace{[\sum_{x_{k,i} > s_k^*}(y_i - \overline{y}_k^+)^2 + \sum_{x_{k,i} < s_k^*}(y_i - \overline{y}_k^-)^2]}_{\text{variance post-split}}$$

```
foreach feature i{
        find split associate to feature i which maximize u;
}
choose best split, the one that maximizes u;
split feature space in 2 partition;

repeat for each partition if stopping criteria isn't reached.
```

Typical stopping criteria: $|R_i| < threshold$ where R_i is the partition where the algorithm restart in.
This method is very good because can be read by anyone since create a binomial tree.
A problem of this algorithm is that it's easy to overfit our training set. To avoid this problem it's useful to modify $u(s_k^*)$ adding a penalization to the number of partitions j:

$$u(s_k^*) = \sum_{k=1}^{j}\sum_{i=1}^{n}(y_i - \overline{y}_k)^2 + \alpha \cdot j$$

where α is choose with cross validation.
Rmk: without any constrain to j the model will overfit training data putting $j = |\mathbb{X}|$ and creating a partition with only 1 point;
Rmk: work also for categorical value simply our split from a threshold become a subset of categorical variables analyzed.

Linear Model for Regression

We call each feature as x_i where i indicate the column number representing that feature.
Recall that:

$$\mathbb{X} = \begin{bmatrix} \underline{x}_1' & y_1 \\ \dots & \dots \\ \underline{x}_n' & y_n \end{bmatrix} \text{ with } \underline{x}_i' = (x_{i,1}, x_{i,2}, \dots, x_{i,p})$$

From that we can create a matrix to store each transformation of the raw data contained in $\mathbb{X}$, it is called #design_matrix $\in n \times (r+1)$:

$$\mathbb{Z} = \begin{bmatrix} 1 & z_{1,1} & \dots & z_{1,r} \\ \dots & \dots & \dots & \dots \\ 1 & z_{n,1} & \dots & z_{n,r} \end{bmatrix}$$

We call each feature of our design matrix as $z_i = h_i(x_1, x_2, \dots, x_p)$ where $i \in \{1, \cdots, r\}$ indicate the column number representing that feature and $h_i(x_1, x_2, \dots, x_p)$ is the transformation of the raw-features.
Without supposing Gaussianity, this type of models is linear in z_i so we find that the #regression_function is:

$$\mathbb{E}[y|x_1, \dots, x_p] = \beta_0 + \beta_1 h_1(x_1, x_2, \dots, x_p) + \beta_2 h_2(x_1, x_2, \dots, x_p) + \cdots + \beta_r h_r(x_1, x_2, \dots, x_p)$$

From that we assume the general model of y is:

$$y = \beta_0 + \beta_1 z_1 + \cdots + \beta_r z_r + \epsilon \quad \Rightarrow \quad \boxed{\underline{y} = (y_1, \dots, y_n)' = \mathbb{Z}\underline{\beta} + \underline{\epsilon}}$$

We know that $\mathbb{E}[\epsilon] = \underline{0}$ and $cov(\underline{\epsilon}) = \sigma^2 I$ with I the identity matrix so each ϵ_i is uncorrelated to others.

Example - #ANOVA

Our starting point is the ANOVA system of equations:

$$\begin{cases} x_{1,1}, \dots, x_{1,n_1} & iid \sim N(\mu_1, \sigma^2) \\ x_{2,1}, \dots, x_{2,n_2} & iid \sim N(\mu_2, \sigma^2) \\ \dots \\ x_{g,1}, \dots, x_{g,n_g} & iid \sim N(\mu_g, \sigma^2) \end{cases}$$

Where each row is $\perp\!\!\!\perp$ to others and $n = n_1 + n_2 + \cdots + n_g$
We know that $\underline{y} = (x_{1,1}, \dots, x_{1,n_1}, \dots, \dots, x_{g,1}, \dots, x_{g,n_g}) \in \mathbb{R}^n$
So our design matrix become:

$$\mathbb{Z} = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots \\ 1 & \dots & 0 & \dots & \dots \\ 1 & 1 & 0 & 0 & \dots \\ 1 & 0 & 1 & 0 & \dots \\ 1 & \dots & \dots & \dots & \dots \\ 1 & 0 & 1 & 0 & \dots \\ 1 & 0 & 0 & 1 & \dots \\ 1 & \dots & \dots & \dots & \dots \end{bmatrix} \in n \times (g+1)$$

Where the ones in the first column are exactly n_1, in the second n_2 and so on.
We can also define $\underline{\beta} = (\mu, \tau_1, \dots, \tau_g)$ and $\underline{\epsilon} \sim N(\underline{0}, \sigma^2 I)$ with I the identity matrix.
So we reach that:

$$\underline{y} = \mathbb{Z}\underline{\beta} + \underline{\epsilon} \Leftrightarrow \begin{cases} x_{i,j} = \mu + \tau_i + \epsilon_{i,j} \\ \epsilon_{i,j} \quad iid \sim N(0, \sigma^2) \end{cases}$$

We know that this problem is over-parametrized, due to the fact that $\mathbb{Z}$ has not full rank, so we have to add the following constrain: $\sum_{i=1}^{g} n_i \tau_i = 0$
From the constrained we deduce that $\tau_g = -\sum_{i=1}^{g-1} \frac{n_i}{n_g} \tau_i$ so we can rewrite the design matrix as:

$$\mathbb{Z} = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots \\ 1 & \dots & 0 & \dots & \dots \\ 1 & 1 & 0 & 0 & \dots \\ 1 & 0 & 1 & 0 & \dots \\ 1 & \dots & \dots & \dots & \dots \\ 1 & 0 & 1 & 0 & \dots \\ 1 & 0 & 0 & 1 & \dots \\ 1 & \dots & \dots & \dots & \dots \\ 1 & \dots & \dots & \dots & \dots \\ 1 & \frac{n_2}{n_g} & \frac{n_3}{n_g} & \frac{n_4}{n_g} & \dots \\ 1 & \dots & \dots & \dots & \dots \\ 1 & \frac{n_2}{n_g} & \frac{n_3}{n_g} & \frac{n_4}{n_g} & \dots \end{bmatrix} \in n \times g$$

Estimate $\underline{\beta}$ and σ^2:

Remember: #ordinary_least_squared $:= \hat{\underline{\beta}} = \underset{\underline{\beta} \in \mathbb{R}^{r+1}}{\arg\, min} ||\underline{y} - \mathbb{Z}\underline{\beta}||^2$

Prop

If $\mathbb{Z}$ is full rank ($\leftrightarrow \text{rank}(\mathbb{Z}) = r + 1 \leq n$)

Then:

$$\boxed{\begin{aligned} &\hat{\underline{y}} = \pi_{\underline{y}|\mathcal{L}(\mathbb{Z})} = \mathbb{Z}(\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'\underline{y} \\ &\hat{\underline{\beta}} = (\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'\underline{y} \end{aligned}}$$

Proof - previous Prop

Starting points:

- $\mathbb{Z}'\mathbb{Z} \in (r+1) \times (r+1)$
- $\mathbb{Z}'\mathbb{Z} = \sum_{i=1}^{r+1} \lambda_i \underline{e}_i \underline{e}_i'$ with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{r+1} > 0$
- If $\mathbb{Z}$ isn't full rank else $\lambda_{r+1} = 0$

$\Rightarrow (\mathbb{Z}'\mathbb{Z})^{-1} = \sum_{i=1}^{r+1} \frac{1}{\lambda_i} \underline{e}_i \underline{e}_i'$

Define $\forall i \in \{1, \dots, r+1\} \quad \underline{q}_i = \frac{1}{\sqrt{\lambda_i}} \mathbb{Z}\underline{e}_i$

So:

$$\begin{cases} \forall i \quad \underline{q}_i \in \mathcal{L}(\mathbb{Z}) \\ \underline{q}_i'\underline{q}_j = \frac{1}{\sqrt{\lambda_i \lambda_j}} \underline{e}_i' \mathbb{Z}'\mathbb{Z}\underline{e}_j \overbrace{=}^{\mathbb{Z}'\mathbb{Z}\underline{e}_j = \lambda_j \underline{e}_j} \frac{\lambda_j}{\sqrt{\lambda_i \lambda_j}} \underline{e}_i'\underline{e}_j = \begin{cases} 0 & else \\ 1 & i = j \end{cases} \end{cases} \Rightarrow \underline{q}_1, \dots, \underline{q}_{r+1} \text{ is an orthonormal basis of } \mathcal{L}(\mathbb{Z})$$

$$\Rightarrow \pi_{\underline{y}|\mathbb{Z}} = \sum_{i=1}^{r+1} \underbrace{\frac{\underline{q}_i \underline{q}_i'}{\underline{q}_i'\underline{q}_i}}_{=1} \underline{y} = \sum_{i=1}^{r+1} \underline{q}_i \underline{q}_i' \underline{y} = \sum_{i=1}^{r+1} \frac{1}{\lambda_i} \mathbb{Z}\underline{e}_i \underline{e}_i' \mathbb{Z}'\underline{y} = \mathbb{Z}\left(\sum_{i=1}^{r+1} \frac{1}{\lambda_i} \underline{e}_i \underline{e}_i'\right) \mathbb{Z}'\underline{y} = \mathbb{Z}\hat{\underline{\beta}} = \overbrace{\mathbb{Z}(\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'}^{=H}\underline{y}$$

$\Rightarrow \hat{\underline{y}} = H\underline{y} \Rightarrow \hat{\underline{\beta}} = (\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'\underline{y}$

□

Remark:

- $rank(\mathbb{Z}) = k < r+1 \leq n : \lambda_k > 0$ while $\lambda_{k+1} = \cdots = \lambda_{r+1} = 0$ so $\nexists (\mathbb{Z}'\mathbb{Z})^{-1}$ but we can use **Moore-Pensore Inverse** $(\mathbb{Z}'\mathbb{Z})^{-}$ called also **Generalized Inverse**
- $rank(\mathbb{Z}) = n = r+1 : \mathcal{L}(\mathbb{Z}) = \mathbb{R}^n$ so $\underline{y} = \underline{\hat{y}} \Rightarrow H = I \Rightarrow$ interpolating data

---

framework: $\mathbb{Z}$ is full rank

We know that $\boxed{H = \mathbb{Z}(\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'}$ is the #orthogonal_projection_operator on $\mathcal{L}(\mathbb{Z})$. So $I - H$ is the orthogonal projection operator on $\mathcal{L}^{\perp}(\mathbb{Z})$ where $dim(\mathcal{L}^{\perp}(\mathbb{Z})) = n - (r+1)$

We find that: $\underline{\hat{\epsilon}} = (I - H)\underline{y} \Rightarrow \underline{\hat{\epsilon}} \perp \underline{\hat{y}}$

So we reach the #1_decomposition_of_variance:

$$||\underline{y}||^2 = SS_{tot} = ||\underline{\hat{y}}||^2 + ||\underline{\hat{\epsilon}}||^2 = SS_{regression} + SS_{residuals}$$

recall: $\pi_{\underline{y}|\underline{1}} = \overline{y} \cdot \underline{1}$

Starting from: $\pi_{\underline{\hat{y}}|\underline{1}} = \frac{\underline{1}\underline{1}'}{\underline{1}'\underline{1}}\underline{\hat{y}} = \frac{\underline{1}\underline{1}'}{\underline{1}'\underline{1}}H\underline{y}$

Remember that orth. proj. oper. is symmetric so: $\underline{1}'H = (H'\underline{1})' = (H\underline{1})' = \underline{1}'$

We reach that: $\pi_{\underline{\hat{y}}|\underline{1}} = \frac{\underline{1}\underline{1}'}{\underline{1}'\underline{1}}\underline{y} = \overline{y} \cdot \underline{1} = \pi_{\underline{y}|\underline{1}}$ so $\underline{y}$ and $\underline{\hat{y}}$ have the same mean

From the previous result we arrive at #2_decomposition_of_variance using the Corrected Sum of Squares (CSS):

$$||\underline{y} - \overline{y} \cdot \underline{1}||^2 = CSS = ||\underline{\hat{y}} - \overline{y} \cdot \underline{1}||^2 + ||\underline{\hat{\epsilon}}||^2 = CSS_{regression} + CSS_{residuals}$$

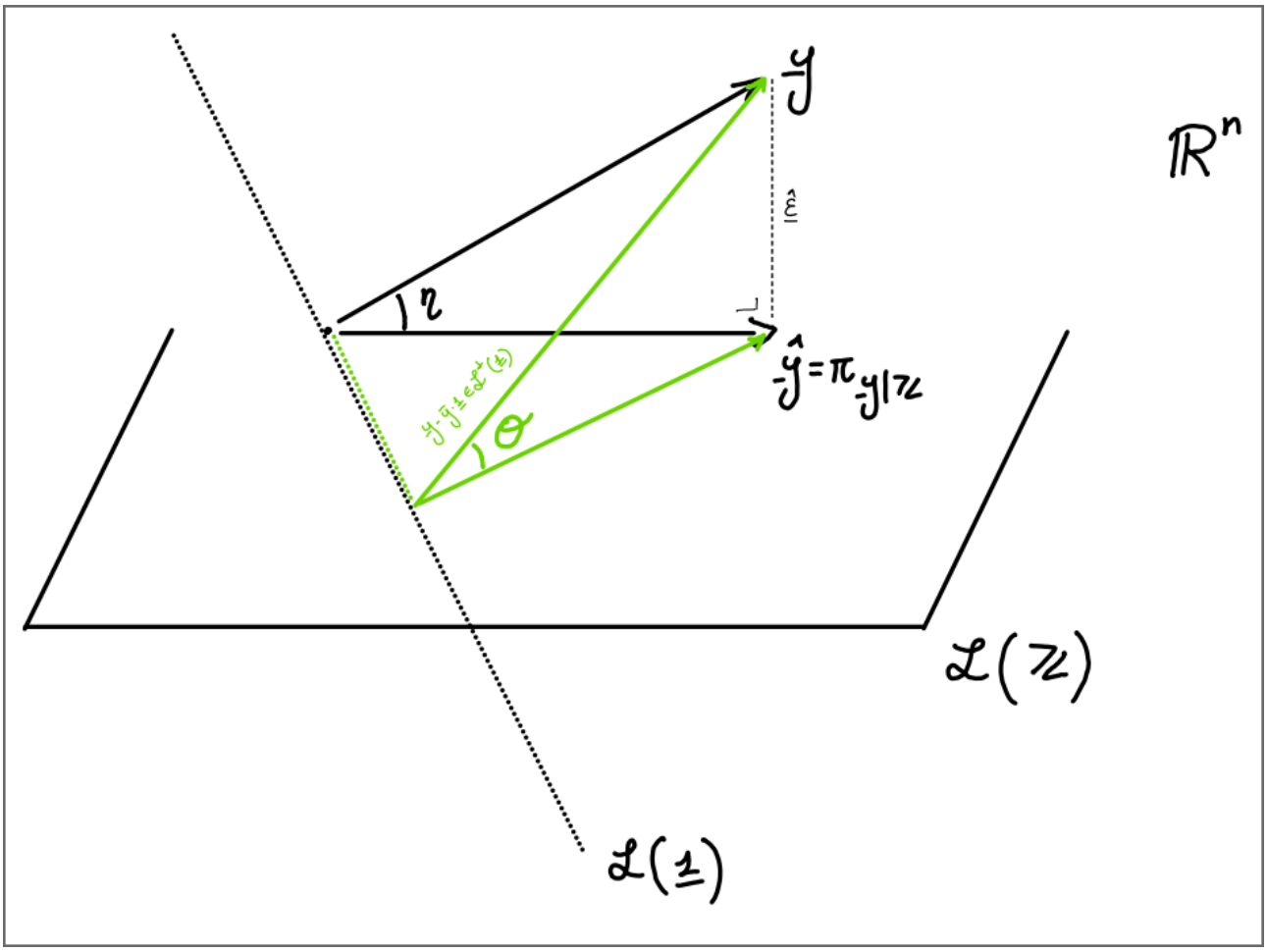
So we can define #coefficient_of_determination as:

$$R^2 = 1 - \frac{\sum_i \underline{\hat{\epsilon}}_i^2}{\sum_i (\underline{y}_i - \underline{\hat{y}}_i)^2} = 1 - \frac{||\underline{\hat{\epsilon}}||^2}{||\underline{\hat{y}}||^2} = 1 - \frac{SS_{residuals}}{SS_{regression}} = 1 - \sin^2\theta = \cos^2\theta$$

Interpretation: portion of variability explained by $\underline{\hat{y}}$

Remarks:

- All this deduction are valid since the first column of $\mathbb{Z}$ is $\underline{1} \Rightarrow \mathbb{E}[\underline{\hat{y}}] = \mathbb{E}[\underline{y}]$
- From 1^{st} decomposition of variance we can deduce $\tilde{R}^2 = 1 - \frac{||\underline{\hat{\epsilon}}||^2}{||\underline{y}||^2} = \cos^2\eta$
- $R^2_{adjusted} = 1 - \frac{\sum_i \underline{\hat{\epsilon}}_i^2}{n-(r+1)} : \frac{\sum_i (\underline{y}_i - \underline{\hat{y}}_i)^2}{n-1}$ is R^2 adjusted by the degree of freedom, it's a measure of how good is a model fitted



Prop

- $R^2 = 1 \Rightarrow \theta = 0 \Rightarrow \underline{y} = \underline{\hat{y}} \Rightarrow$ Interpolation;
- $R^2 = 0 \Rightarrow \theta = \frac{\pi}{2} \Rightarrow \underline{y} \perp \mathcal{L}(\mathbb{Z}) \Rightarrow \underline{\hat{y}} = \overline{y} \cdot \underline{y}$

Properties $\underline{\beta}$ and $\underline{\hat{\epsilon}}$:

Framework:

- $\mathbb{Z}$ full rank $\Leftrightarrow rank(\mathbb{Z}) = r + 1$
- $\underline{\hat{\beta}} = (\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'\underline{y}$
- $\underline{\hat{\epsilon}} = (I - H)\underline{y} \Rightarrow \underline{\hat{\epsilon}} \perp \underline{\hat{y}}$

Prop

1. $\mathbb{E}[\underline{\hat{\beta}}] = \underline{\beta} \leftarrow$ unbiased
2. $cov(\underline{\hat{\beta}}) = \sigma^2(\mathbb{Z}'\mathbb{Z})^{-1}$
3. $\mathbb{E}[\underline{\hat{\epsilon}}] = \underline{0}$
4. $cov(\underline{\hat{\epsilon}}) = \sigma^2(I - H) \neq \sigma^2 I = cov(\underline{\epsilon})$
5. $\mathbb{E}[\underline{\hat{\epsilon}}'\underline{\hat{\epsilon}}] = \mathbb{E}[||\underline{\hat{\epsilon}}||^2] = [n - (r+1)]\sigma^2 \Rightarrow \mathbb{E}[\frac{\underline{\hat{\epsilon}}'\underline{\hat{\epsilon}}}{n-(r+1)}] = \sigma^2 \leftarrow$ unbiased

Proof - previous Prop

1. $\begin{cases} \mathbb{E}[\hat{\underline{\beta}}] = \mathbb{E}[(\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'\underline{y}] = (\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'\mathbb{E}[\underline{y}] \\ \mathbb{E}[\underline{y}] = \mathbb{E}[\mathbb{Z}\underline{\beta} + \underline{\epsilon}] = \mathbb{E}[\mathbb{Z}\underline{\beta}] \end{cases} \Rightarrow \mathbb{E}[\hat{\underline{\beta}}] = (\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'\mathbb{Z}\underline{\beta} = \underline{\beta}$
2. $cov(\hat{\underline{\beta}}) = (\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'\underbrace{cov(\underline{y})}_{=\sigma^2 I}\mathbb{Z}(\mathbb{Z}'\mathbb{Z})^{-1} = \sigma^2(\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'\mathbb{Z}(\mathbb{Z}'\mathbb{Z})^{-1} = \sigma^2(\mathbb{Z}'\mathbb{Z})^{-1}$
3. $\mathbb{E}[\hat{\underline{\epsilon}}] = \mathbb{E}[(I-H)\underline{y}] = \overbrace{(I-H)}^{\in\mathcal{L}^{\perp}(\mathbb{Z})}\overbrace{\mathbb{Z}}^{\in\mathcal{L}(\mathbb{Z})}\underline{\beta} = 0$
4. $cov(\hat{\underline{\epsilon}}) = (I-H)cov(\underline{y})(I-H)' = \sigma^2(I-H)(I-H)' \xrightarrow{\text{symmetric + idempotent}} = \sigma^2(I-H)$ with $det(I-H) = 0$ since is an $n \times n$ matrix which project on a space of dimension $n-(r+1)$
5. Recall:

   1) $\hat{\underline{\epsilon}} = (I-H)\underline{y} = (I-H)(\mathbb{Z}\underline{\beta} + \underline{\epsilon}) = (I-H)\underline{\epsilon}$

   2) $tr(H) = tr(\mathbb{Z}'(\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}) = tr(\mathbb{Z}'\mathbb{Z}(\mathbb{Z}'\mathbb{Z})^{-1}) = tr(I_{(r+1)\times(r+1)}) = r+1$

   We can reach:

   $$\mathbb{E}[\hat{\underline{\epsilon}}'\hat{\underline{\epsilon}}] = tr(\mathbb{E}[\hat{\underline{\epsilon}}'\hat{\underline{\epsilon}}]) = \mathbb{E}[tr(\hat{\underline{\epsilon}}'\hat{\underline{\epsilon}})] = \mathbb{E}[tr(\hat{\underline{\epsilon}}\hat{\underline{\epsilon}}')]$$

   $$\overset{1^{st}\text{ recall}}{\Rightarrow} \mathbb{E}[tr((I-H)\underline{\epsilon}\underline{\epsilon}'(I-H)')] = tr((I-H)\mathbb{E}[\underline{\epsilon}\underline{\epsilon}'](I-H)) = \overbrace{tr(\sigma^2(I-H))}^{=\dim(\text{space into proj})} = \sigma^2(n - tr(H)$$

   $$\overset{2^{nd}\text{ recall}}{\Rightarrow} = \sigma^2[n-(r+1)]$$

   □

Obs

- $cov(\hat{\underline{\beta}}) = \sigma^2(\mathbb{Z}'\mathbb{Z})^{-1} \rightarrow$ usually $\hat{\underline{\beta}}_i$ aren't uncorrelated unless $\mathbb{Z}'\mathbb{Z} = \alpha I$ with $\alpha \in \mathbb{R}$
- When designing an experiment we can control variability of $\hat{\underline{\beta}}$ controlling $\mathbb{Z}$

Change of paradigm $\underline{\epsilon} \sim N(\underline{0}, \sigma^2 I)$

Prop

Adding gaussianity assumption to $\underline{\epsilon}$ bring us some nice properties:

1. $\hat{\underline{\beta}}$ and $\hat{\sigma}^2 = \frac{\hat{\underline{\epsilon}}'\hat{\underline{\epsilon}}}{n}$ are **MLE** estimator of $\underline{\epsilon}$ and σ^2;
2. $\hat{\underline{\beta}} \sim N_{r+1}(\underline{\beta}, \sigma^2(\mathbb{Z}'\mathbb{Z})^{-1})$
3. $\hat{\underline{\epsilon}} \sim N_n(\underline{0}, \sigma^2(I-H))$
4. $\hat{\underline{\epsilon}} \perp\!\!\!\perp \hat{\underline{\beta}}$
5. $\hat{\underline{\epsilon}}'\hat{\underline{\epsilon}} \sim \sigma^2\chi(n-(r+1))$

Proof - previous Prop

Since as assumption we know that $\underline{\epsilon} \sim N(\underline{0}, \sigma^2 I) \Rightarrow \underline{y} \sim N(\mathbb{Z}\underline{\beta}, \sigma^2 I)$

1. Write likehood and do differentiation;

   2, 3, 4)

$$\begin{bmatrix}\hat{\underline{\beta}}\\ \hat{\underline{\epsilon}}\end{bmatrix} = \underbrace{\begin{bmatrix}(\mathbb{Z}'\mathbb{Z})\mathbb{Z}\\ I-H\end{bmatrix}}_{A}\underline{y} \sim N(\begin{bmatrix}\underline{\beta}\\ \underline{0}\end{bmatrix}, \begin{bmatrix}(\mathbb{Z}'\mathbb{Z})^{-1} & \underline{0}\\ \underline{0} & I-H\end{bmatrix}) = N(A\mathbb{Z}\underline{\beta}, \sigma^2 AIA')$$

2. $\hat{\underline{\epsilon}} \sim N_n(\underline{0}, \sigma^2(I-H)) \Rightarrow \hat{\underline{\epsilon}}'\hat{\underline{\epsilon}} = d^2(\hat{\underline{\epsilon}}, \underline{0}) = \frac{1}{\sigma^2}(\hat{\underline{\epsilon}}-\underline{0})' \underbrace{(I-H)^{-}}_{\text{Penrose Inverse}} (\hat{\underline{\epsilon}}-\underline{0}) \sim \chi^2(n-(r+1))$

□

Framework: $\mathbb{Z}$ is full rank and $\underline{\epsilon} \sim N(\underline{0}, \sigma^2 I)$

Recall: $S^2 = \frac{\hat{\underline{\epsilon}}'\hat{\underline{\epsilon}}}{n-(r+1)}$

Goal: find $CI(\underline{\beta})$ one-at-time

Obs

- $\text{®} = \frac{1}{\sigma^2}(\hat{\underline{\beta}}-\underline{\beta})'(\mathbb{Z}'\mathbb{Z})(\hat{\underline{\beta}}-\underline{\beta}) \sim \chi^2(r+1)$
- $\text{Ⓢ} = \frac{1}{\sigma^2}\hat{\underline{\epsilon}}'\hat{\underline{\epsilon}} \sim \chi^2(n-(r+1))$
- $\text{®} \perp\!\!\!\perp \text{Ⓢ}$

From the previous observation we find:

$$\frac{(\hat{\underline{\beta}}-\underline{\beta})'(\mathbb{Z}'\mathbb{Z})(\hat{\underline{\beta}}-\underline{\beta})}{r+1}\frac{n-(r+1)}{\hat{\underline{\epsilon}}'\hat{\underline{\epsilon}}} \sim F(r+1, n-(r+1))$$
$$\Leftrightarrow \frac{1}{S^2}(\hat{\underline{\beta}}-\underline{\beta})'(\mathbb{Z}'\mathbb{Z})(\hat{\underline{\beta}}-\underline{\beta}) \sim (r+1)F(r+1, n-(r+1))$$

From this we can construct the confidence interval at level α as:

$$CI_{1-\alpha}(\underline{\beta}) = \{\underline{\mu} \in \mathbb{R}^{r+1} : (\hat{\underline{\beta}}-\underline{\mu})'(\mathbb{Z}'\mathbb{Z})(\hat{\underline{\beta}}-\underline{\mu}) \leq (r+1)S^2 F_{1-\alpha}(r+1, n-(r+1))\}$$

So we can say that taking $\underline{a} \in \mathbb{R}^{r+1}$ we reach: $\underline{a}'\hat{\underline{\beta}} \sim N_1(\underline{a}'\underline{\beta}, \sigma^2\underline{a}'(\mathbb{Z}'\mathbb{Z})^{-1}\underline{a})$
This implies that:

$$\boxed{\frac{\underline{a}'(\hat{\underline{\beta}}-\underline{\beta})}{\sigma\sqrt{\underline{a}'\mathbb{Z}'\mathbb{Z}\underline{a}}} \cdot \frac{\sqrt{\sigma^2}}{\sqrt{S^2}} = \frac{\underline{a}'(\hat{\underline{\beta}}-\underline{\beta})}{S\sqrt{\underline{a}'\mathbb{Z}'\mathbb{Z}\underline{a}}} \sim t(n-(r+1)) \leftarrow \text{Pivotal quantity}}$$

$\Rightarrow CI_{1-\alpha}(\underline{a}'\underline{\beta}) = \{\underline{a}'\hat{\underline{\beta}} \pm S\sqrt{\underline{a}'\mathbb{Z}'\mathbb{Z}\underline{a}} \cdot t_{1-\frac{\alpha}{2}}(n-(r+1))\}$ which is one-at-time
$\Rightarrow CI_{1-\alpha}(\beta_i) = \{\hat{\beta}_i \pm S\sqrt{diag_i(\mathbb{Z}'\mathbb{Z})} \cdot t_{1-\frac{\alpha}{2}}(n-(r+1))\}$ with $\underline{a} = [0, \dots, 0, 1, 0, \dots, 0]$ which is one-at-time.

$\boxed{\textbf{WARNING}\text{: not use this to create a } CI(\underline{\beta}) \text{ because aren't simultaneously}}$

Goal: find $CI(\underline{\beta})$ simultaneously

$$\max_{\underline{a}\in\mathbb{R}^{r+1}} \frac{[\underline{a}'(\hat{\underline{\beta}}-\underline{\beta})]^2}{S^2(\underline{a}'\mathbb{Z}'\mathbb{Z}\underline{a})} = \frac{1}{S^2}(\hat{\underline{\beta}}-\underline{\beta})'(\mathbb{Z}'\mathbb{Z})(\hat{\underline{\beta}}-\underline{\beta}) \sim (r+1)F(r+1,n-(r+1))$$

So we can construct the confidence interval at level α as:

$$CI_{1-\alpha}(\underline{a}'\underline{\beta}) = [\underline{a}'\hat{\underline{\beta}} \pm S(\underline{a}'\mathbb{Z}'\mathbb{Z}\underline{a})\sqrt{(r+1)F_{1-\alpha}(r+1,n-(r+1))}]$$

Goal: find $CI(\sigma^2)$

$$\frac{1}{\sigma^2}\hat{\underline{\epsilon}}'\hat{\underline{\epsilon}} = \frac{[n-(r+1)]S^2}{\sigma^2} \sim \chi^2(n-(r+1))$$

So we can construct the confidence interval at level α as:

$$CI_{1-\alpha}(\sigma^2) = \left[\frac{[n-(r+1)]S^2}{\chi^2_{1-\frac{\alpha}{2}}(n-(r+1))}; \frac{[n-(r+1)]S^2}{\chi^2_{\frac{\alpha}{2}}(n-(r+1))}\right]$$

Framework: $C \in p \times (r+1)$, contains linear combinations of $\underline{\beta}$
Goal: test $H_0 : C\underline{\beta} = 0 \quad \text{vs} \quad H_1 : C\underline{\beta} \neq 0$
Since we know that: $C\hat{\underline{\beta}} \sim N_p(C\underline{\beta}, \sigma^2 C(\mathbb{Z}'\mathbb{Z})^{-1}C)$
We reach:

$$\frac{\frac{1}{\sigma^2}(C\hat{\underline{\beta}})'(C(\mathbb{Z}'\mathbb{Z})^{-1}C)^{-1}(C\hat{\underline{\beta}})}{p} \cdot \frac{\sigma^2}{S^2} \sim F(p,n-(r+1))$$

So we can construct our test statistics F such that:

$$\boxed{F = \frac{(C\hat{\underline{\beta}})'(C(\mathbb{Z}'\mathbb{Z})^{-1}C)^{-1}(C\hat{\underline{\beta}})}{S^2} \sim p \cdot F(p,n-(r+1))}$$

Reject H_0 if at level α if $F > p \cdot F_{1-\alpha}(p, n-(r+1))$

Goal: test $H_0 : \beta_r = \beta_{r-1} = \cdots = \beta_{r-(p-1)} = 0 \quad \text{vs} \quad H_1 : else$
This test con be rewritten as the case before using as C:

$$C = \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} = [0 \quad I_{p\times p}] \in p \times (r+1)$$

With this is like to say that $\mathbb{Z} = [\mathbb{Z}_1, \mathbb{Z}_2]$ and we want to test our reduced model $\mathbb{Z}_1$ against the full model $\mathbb{Z}$, since we are testing that betas of $\mathbb{Z}_2$ are zeros.
**We reject $\boldsymbol{H_0}$ if $\boldsymbol{SS_{res}(\mathbb{Z}_1) - SS_{res}(\mathbb{Z})}$ is large knowing that
$\frac{SS_{res}(\mathbb{Z}_1) - SS_{res}(\mathbb{Z})}{S^2 p} \sim F(p, n-(r+1))$
Special case: when we want to test: $H_0 : \beta_1 = \beta_2 = \dots = \beta_r = 0 \quad \text{vs} \quad H_1 : else$
So our

$$\mathbb{Z}_1 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \Rightarrow SS_{res}(\mathbb{Z}_1) = \sum_1^n (y_i - \bar{y})^2 \Rightarrow \frac{SS_{res}(\mathbb{Z}_1) - SS_{res}(\mathbb{Z})}{S^2 r} = \frac{\sum_1^n (y_i - \bar{y})^2 - \sum \hat{\epsilon_i}^2}{S^2 r} \sim F(r, n-(r+1))$$

This test is done by default as F-test in R packages.

Goal: Prediction of y
We have a model for the phenomenon which is $y_0 = \underline{Z}_0' \underline{\beta} + \epsilon_0$ with $\underline{Z}_0$ the vector collecting the "weights" of each regressor.
We know that the prediction $\mathbb{E}[y_0 | \underline{Z}_0] = \underline{Z}_0' \underline{\beta}$ which we know its unbiased predictor is $\underline{Z}_0' \underline{\hat{\beta}}$

Theo - Gauss-Markov

$\underline{Z}_0' \underline{\hat{\beta}}$ is the Best Linear Unbiased Estimator (#BLUE) of $\underline{Z}_0' \underline{\beta}$

Framework: we change the paradigm, now $cov(\underline{\epsilon}) = \sigma^2 \Sigma$ with $\Sigma \in n \times n \rightarrow$ to model heteroscedasticity + correlations between ϵ_i and ϵ_j

Case 1 - Σ known but not σ^2

We use #Generalized_least_squared to find $\underline{\hat{\beta}}$.

$$\underline{\hat{\beta}} = \underset{\underline{\beta}}{argmin}[(\underline{y} - \mathbb{Z}\underline{\beta})' \Sigma^{-1} (\underline{y} - \mathbb{Z}\underline{\beta})] \leftarrow \text{ minimize Mahalanobis distance}$$

We do the following change of variables:

$$(\underline{y} - \mathbb{Z}\underline{\beta})' \Sigma^{-1} (\underline{y} - \mathbb{Z}\underline{\beta}) = (\underline{y} - \mathbb{Z}\underline{\beta})' \Sigma^{-\frac{1}{2}} \Sigma^{-\frac{1}{2}} (\underline{y} - \mathbb{Z}\underline{\beta}) = ||\underline{\tilde{y}} - \tilde{\mathbb{Z}}\underline{\beta}||^2 \quad \text{with } \underline{\tilde{y}} = \Sigma^{-\frac{1}{2}} \underline{y} \text{ and } \tilde{\mathbb{Z}} = \Sigma^{-\frac{1}{2}} \mathbb{Z}$$

$$\Rightarrow \underline{\hat{\beta}} = \underset{\underline{\beta}}{argmin} ||\underline{\tilde{y}} - \tilde{\mathbb{Z}}\underline{\beta}||^2$$

$$\Rightarrow \underline{\hat{\beta}} = (\tilde{\mathbb{Z}}' \tilde{\mathbb{Z}})^{-1} \tilde{\mathbb{Z}}' \underline{\tilde{y}} = (\mathbb{Z}' \Sigma^{-1} \mathbb{Z})^{-1} \mathbb{Z}' \Sigma^{-1} \underline{y}$$

Idea: $\underline{\tilde{y}}$ is a change of the reference system in a way such that $cov(\underline{\tilde{\epsilon}}) = \Sigma^{-\frac{1}{2}} cov(\underline{\epsilon}) \Sigma^{-\frac{1}{2}} = \sigma^2 I$

Case 2 - Σ and σ^2 unknown

Possible solutions:

1. parameterize Σ, we will see this idea in Linear Mixed Model;
2. Estimate Σ iteratively from residual using as starting point $\Sigma = I$;
3. Use appropriate transformation of $\underline{y}$ and/or $\mathbb{Z}$;

Example 1 - Σ known

$y_i =$ mean of n_i independent observations with the same variance σ^2

So we know that $var(y_i) = \frac{\sigma^2}{n_i}$ and we can model $\Sigma = \begin{bmatrix} \frac{1}{n_1} & 0 & \dots & 0 \\ 0 & \frac{1}{n_2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \frac{1}{n_n} \end{bmatrix}$.

Someone called this Weighted Least Squared (#WLS).

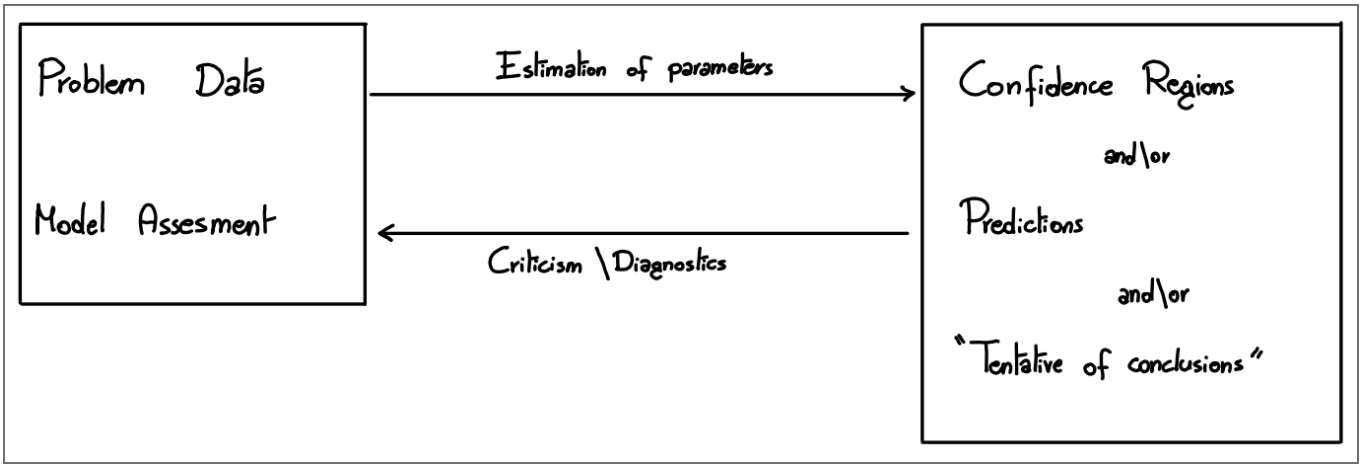
Example 2 - Σ known

$y_i =$ sum of n_i independent observations with the same variance σ^2 (like GDP of each states)

So we know that $var(y_i) = \sigma^2 n_i$ and we can model $\Sigma = \begin{bmatrix} n_1 & 0 & \dots & 0 \\ 0 & n_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & n_n \end{bmatrix}$.

Another example of #WLS

Diagnostics for a linear model



General way to do diagnostic is to control the following stuff:

- residuals analysis, outliers, heteroscedasticity, normality test, auto-correlation;
- Influential cases;
- collinearity;

Residuals analysis

We have:

- Abstract model $\rightarrow \underline{y} = \mathbb{Z}\underline{\beta} + \underline{\epsilon}$ with $\mathbb{E}[\underline{\epsilon}] = 0$ and $cov(\epsilon) = \sigma^2 I$

- Fitted model $\rightarrow \underline{\hat{y}} = \mathbb{Z}\underline{\hat{\beta}} + \underline{\hat{\epsilon}}$ with $\mathbb{E}[\underline{\hat{\epsilon}}] = 0$ and $cov(\underline{\hat{\epsilon}}) = \sigma^2(I - H)$

Problem: If in the abstract model $\underline{\epsilon} \sim N_n(\underline{0}, \sigma^2 I)$, which is a distribution on $\mathbb{R}^n$ then in the fitted model $\underline{\epsilon} \sim N(\underline{0}, \sigma^2(I-H))$ which is a distribution in $\mathcal{L}^{\perp}(\mathbb{Z})$.
Someone prefers to do analysis on the studentized residuals of unit i $= \frac{\hat{\epsilon}_i}{S\sqrt{1-h_{i,i}}}$ with $h_{i,i} = diag(H)_{i,i} =$ leverages $\in (0,1)$ since H is a projector matrix.
So if $h_{i,i} \uparrow \Rightarrow var(\hat{\epsilon}_i) \downarrow 0$ and also if $\mathbb{E}[\hat{\epsilon}_i] = 0$ then $\hat{\epsilon}_i = 0$

#Cook_distance and influential cases

We call $\mathbb{Z}_{-i}$ which is $\mathbb{Z}$ without the i-row $\Rightarrow$ we can write the abstract model as $\underline{y}_{-i} = \mathbb{Z}_{-i}\underline{\beta}_{-i} + \underline{\epsilon}_{-i}$ and also we can estimate with $\underline{\hat{\beta}}_{-i}$.
To see if a unit is influential we can compare $\underline{\hat{\beta}}$ and $\underline{\hat{\beta}}_{-i}$ and to compare it we use the Cook distance:

$$D_i = \frac{(\underline{\hat{\beta}} - \underline{\hat{\beta}}_{-i})(\mathbb{Z}'\mathbb{Z})(\underline{\hat{\beta}} - \underline{\hat{\beta}}_{-i})}{S^2(r+1)} = (\frac{\hat{\epsilon}_i}{S\sqrt{1-h_{i,i}}})^2 \frac{h_{i,i}}{1-h_{i,i}} \frac{1}{r+1} \sim F(r+1, n-(r+1))$$

Rule of thumb: delete unit with $D_i > 1$

Model selection

Since with r regressors we can have 2^r models to find the best one we can:

```
for(k = 1:r){
        fit all possible models with k regressors;
        choose best one (using GOF indicator)
}
```

Why collinearity is a problem?

Since from #ordinary_least_squared we have found that:

- $\underline{\hat{\beta}} = (\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}'\underline{y}$ if $\mathbb{Z}$ is full rank

If there are collinear regressors, tend to be $\mathbb{Z}'\mathbb{Z}$ singular so the inversion of it explode
$\Rightarrow \quad var(\underline{\hat{\beta}}) = \sigma^2(\mathbb{Z}'\mathbb{Z})^{-1}$ significantly increases
Curios is to find that:
$var(\hat{\beta}_j) = \frac{\sigma^2}{\sum_{i=1}^n (z_{i,j} - \bar{z}_j)^2} \cdot \frac{1}{1-R_j^2}$ with R_j^2 as #coefficient_of_determination when z_j is regressed from $z_{i \in \{1,\dots,r\} \setminus \{j\}}$
Remark: $\sum_{i=1}^n (z_{i,j} - \bar{z}_j)^2 \uparrow \quad \Rightarrow \quad var(\hat{\beta}_j) \downarrow$
Remark: If regressors are orthogonal than $R_j^2 = 0 \quad \forall j \quad$ otherwise $\quad R_j^2 \uparrow 1 \Rightarrow var(\hat{\beta}_j) \uparrow$ caused by collinearity.

Tips: red alarm when $VIF = \text{variance inflation factor} = \frac{1}{1-R_j^2} \geq 5$

Solution: To limit this problem we adding the constrain of keep $var(\hat{\underline{\beta}})$ low

New nomenclature

We centered our data, so we reformulate our #ordinary_least_squared as:

- $\underline{y} \to \underline{y}^* = \underline{y} - \overline{y} \cdot \underline{1}$
- $\mathbb{Z} \to \mathbb{Z}^* = \begin{bmatrix} z_{1,1} - \overline{z}_1 & z_{1,2} - \overline{z}_2 & \dots & z_{1,r} - \overline{z}_r \\ z_{2,1} - \overline{z}_1 & z_{2,2} - \overline{z}_2 & \dots & z_{2,r} - \overline{z}_r \\ \vdots & \vdots & \vdots & \vdots \\ z_{n,1} - \overline{z}_1 & z_{n,2} - \overline{z}_2 & \dots & z_{n,r} - \overline{z}_r \end{bmatrix} \in n \times r$
- Ⓢ $\begin{cases} \hat{\underline{\beta}}^* = \arg\, min_{\underline{\beta} \in \mathbb{R}^r} ||\underline{y}^* - \mathbb{Z}^* \underline{\beta}||^2 \\ \hat{\beta}_0 = \overline{y} - \sum_{i=1}^r \hat{\beta}_i^* \overline{z}_i \end{cases}$

CAUTION: So from now on we will call $\underline{y}^*$ as $\underline{y}$ and $\mathbb{Z}^*$ as $\mathbb{Z}$.

#Ridge_regression

Useful reference: link

Add a parameter and change Ⓢ into:

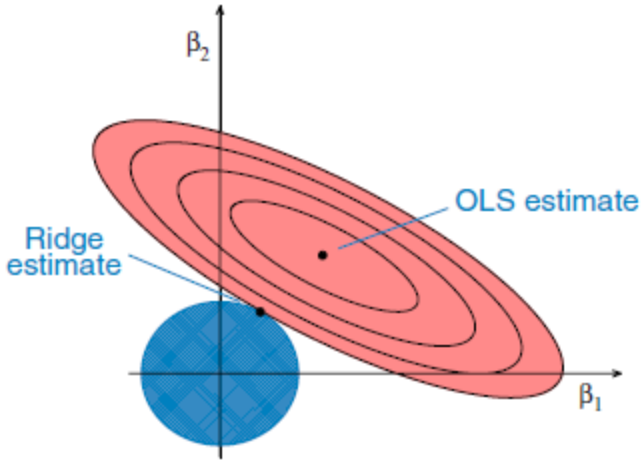
$$\begin{cases} \hat{\underline{\beta}}_{Ridge} = \arg\, min_{\underline{\beta} \in \mathbb{R}^r} ||\underline{y} - \mathbb{Z}\underline{\beta}||^2 \\ ||\underline{\beta}||^2 \leq s \end{cases}$$

Relation Between OLS and Ridge Regression

But knowing that $\hat{\underline{y}} = H\underline{y} = \mathbb{Z}\overbrace{(\mathbb{Z}'\mathbb{Z})^{-1}\mathbb{Z}\underline{y}}^{\hat{\underline{\beta}}_{OLS}} = \mathbb{Z}\hat{\underline{\beta}}_{OLS}$ we can simplify find that:

$$\begin{aligned} ||\underline{y} - \mathbb{Z}\underline{\beta}||^2 &= ||\underline{y} - \hat{\underline{y}} + \hat{\underline{y}} - \mathbb{Z}\underline{\beta}||^2 \\ &= ||\underline{y} - \hat{\underline{y}} - \mathbb{Z}(\underline{\beta} - \hat{\underline{\beta}}_{OLS})||^2 \\ &= ||\overbrace{\hat{\underline{\epsilon}}}^{\in \mathcal{L}^\perp(\mathbb{Z})} - \overbrace{\mathbb{Z}(\underline{\beta} - \hat{\underline{\beta}}_{OLS})}^{\in \mathcal{L}(\mathbb{Z})}||^2 \\ &= ||\hat{\underline{\epsilon}}||^2 - ||\mathbb{Z}(\underline{\beta} - \hat{\underline{\beta}}_{OLS})||^2 \\ &\Rightarrow \begin{cases} \hat{\underline{\beta}}_{Ridge} = \arg\, min_{\underline{\beta} \in \mathbb{R}^r} ||\mathbb{Z}(\underline{\beta} - \hat{\underline{\beta}}_{OLS})||^2 \\ ||\underline{\beta}||^2 \leq s \end{cases} \end{aligned}$$

So we can see it as:



How to find Ridge regressors

To find it we have to use lagrangian: $\arg\ min_{\underline{\beta}\in\mathbb{R}^r}||\underline{y}-\mathbb{Z}\underline{\beta}||^2+\lambda||\underline{\beta}||^2$

So we reach: $\underline{\hat{\beta}}_{Ridge}=(\mathbb{Z}'\mathbb{Z}-\lambda I)^{-1}\mathbb{Z}'\underline{y}$

Problem: $\underline{\hat{\beta}}_{Ridge}$ is biased because of λ in the lagrangian

But Hoerl&Kennard in 1970 find that:

$\forall$ regression problem $\exists\lambda^*:\mathbb{E}[||\underline{\hat{\beta}}_{Ridge}-\underline{\beta}||^2]\leq\mathbb{E}[||\underline{\hat{\beta}}_{OLS}-\underline{\beta}||^2]$ with λ^* found using cross validation

#PCA_regression

Another way to manage collinearity is to use #PCA on $\mathbb{Z}$ to find $k\leq r$ orthogonal regressors

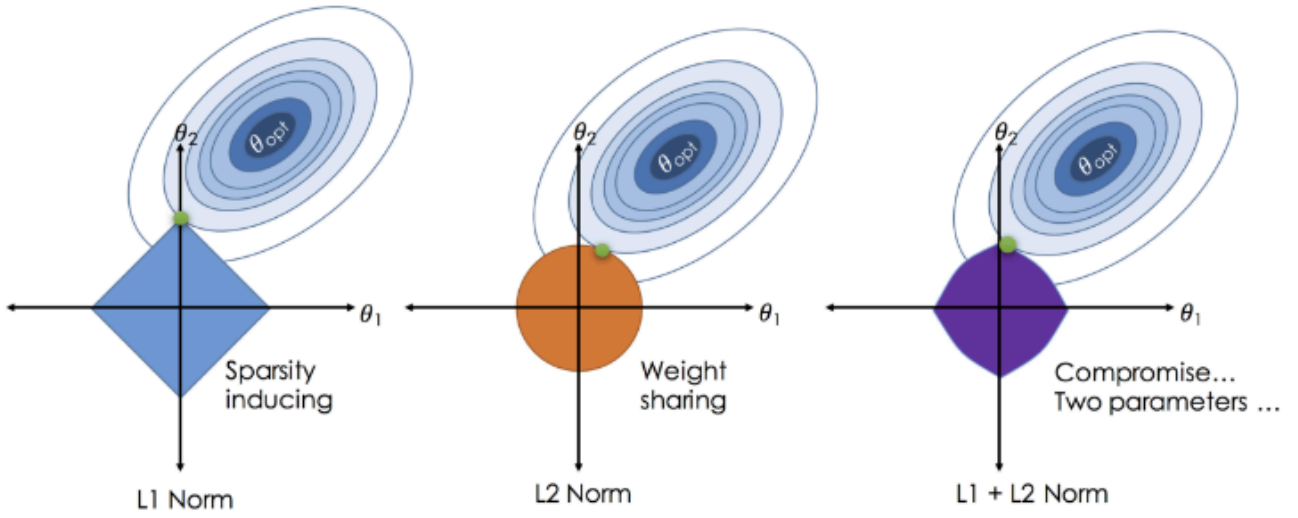
Problem: PCA regression and Ridge regression doesn't support sparse solution in term of $z_1,\dots,z_r$, so they only limit the effect of some regressor but not eliminate it

Intuition Tibshirami (1996)

Use Ridge regression but change norm on which limit the $\underline{\beta}$. In this way, we are also selecting regressors since someones become 0.

So it become:

$$\begin{cases}\underline{\hat{\beta}}=\arg\ min_{\underline{\beta}\in\mathbb{R}^r}||\underline{y}-\mathbb{Z}\underline{\beta}||^2\\ ||\underline{\beta}||_1\leq s\end{cases}\quad\rightarrow\quad\text{which is called LASSO}$$



From this we can spike the area of search of our $\hat{\underline{\beta}}$. Usually, much is spike our area and more difficult is the resolution of the lagrangian.

Note: $\arg\, min_{\underline{\beta}\in\mathbb{R}^r}||\underline{y}-\mathbb{Z}\underline{\beta}||^2+\lambda_1||\underline{\beta}||_1+\lambda_2||\underline{\beta}||_2^2$ is called Elastic Net

Since now we have studied that, to do regression\classification we have to:

- Take a training set $\mathbb{X}$;
- Fit a model on that training set $M_{\mathbb{X}}$;
- Give a new features vector $\underline{x}$ predict with $M_{\mathbb{X}}(\underline{x})$

Ensemble methods

Problem: we have to deal with the bias-variance trade off

Q: how to reduce variance without dealing with bias?

If is true that $\mathbb{E}[\epsilon]=0$ we can take B *independent* measurements y and use the mean of these to find the real value μ since we know that:

$$\overline{y}=\frac{1}{B}\sum_{i=1}^{B}y_i \quad \Rightarrow \quad \mathbb{E}[\overline{y}]=\mu \text{ and } var(\overline{y})=\frac{1}{B}\sigma^2 \quad \Rightarrow \quad B\uparrow \text{ then } var(\overline{y})\downarrow$$

So we are reducing variance without adding a bias.

Idea: generate B training set $\mathbb{X}_i$ and fit B models $M_{\mathbb{X}_i}$ and take as final model $M=\frac{1}{B}\sum_{i=1}^{B}M_{\mathbb{X}_i}$

Cons:

- Boring and costly to manually fit B models;
- If I have access to B training set why to not use it fit a unique big model?

So, usually we have a single training set and we want to build multiples of it.

Idea: Use #Bagging which is the composition of #bootstrap and aggregate the data obtained in dataset.

What is bootstrap?

It is a simple algorithm:

```
Initialization: training set of n units
for(i in 1:n){
        sample randomly one unit from the training set;
}
```

What is bagging?

It is very similar to bootstrap but we aggregate samples into a new training set.

```
Initialization: training set of n units
for(i in 1:n){
        sample randomly one unit from the training set;
        add the sampled unit to a new training;
}
```

In this way, the new training set $\mathbb{X}^*$ contains some of the units of the original $\mathbb{X}$ but with some copies, so $\mathbb{X}^* \subset \mathbb{X}$.

$\mathbb{P}[u \notin \mathbb{X}^*]$?

$\mathbb{P}[u \notin \mathbb{X}^*] = (1 - \frac{1}{n})^n \xrightarrow{n \to +\infty} e^{-1} \approx \frac{1}{3}$
So we have approximately $\frac{2}{3}$ of the original units of $\mathbb{X}$ in $\mathbb{X}^*$, remaining observations are called Out-Of-Bag observations #OOB.
This causes that $\mathbb{X}^* \not\perp \mathbb{X}$ so we reach that using the strategy of taking the mean model M we reach that: $var(M) \in (\frac{\sigma^2}{B}, \sigma^2)$

Framework: we use #Regression_tree as model
Idea: To increase independence between models we don't search among each features and cut the best one but for each model we sample randomly $k < p$ features and do the best cut in that k features. This increase independence between model so we reduce variance.
From that come out #random_forest
And to see performance of each model we can use its set of #OOB observations.

Boosting

This technique work fitting sequentially decision trees each one on the residual of the previous tree.