HomeWork1

Prerequisites for Machine Learning

Professor: **Dr.B.Nasersharif**

Provider: **Mehran Tamjidi**

Telegram Group: t.me/ML_403_1

Question 1

۱– (لف) یک مجموعه داده با ۵ نمونه که برای هر یک سه ویژگی **X1,X2,X3** در نظر گرفته شده است. به صورت زیر تعریف شده است. بردار میانگین و ماتریس کوواریانس را برای این مجموعه داده به دست آورید.

	X_1	X_2	X_3
X	4.0	2.0	0.6
	4.2	2.1	0.59
	3.9	2.0	0.58
	4.3	2.1	0.62
	4.1	2.2	0.63

(ب) آیا ماتریس کوواریانس این داده ها، غیر منفرد و مثبت معین است؟

(ج) ماتریس تبدیلی بدست اورید که با ضرب در این داده ها، ماتریس کوواریانس داده ها را قطری کند.

First see what we have to do

What is the covariance matrix?

$$\mathrm{Cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})$$

where:

- X_i and Y_i are individual values of features X and Y,
- $\bar{X}$ and $\bar{Y}$ are the mean values of features X and Y,
- n is the number of data points

– Covariance matrix

$$cov[X] = \Sigma = E\left[\left(\underline{X} - \underline{\mu}\right)\left(\underline{X} - \underline{\mu}\right)^T\right] =$$

$$= \begin{bmatrix} E[(x_1-\mu_1)^2] & \cdots & E[(x_1-\mu_1)(x_N-\mu_N)] \\ \vdots & \ddots & \vdots \\ E[(x_1-\mu_1)(x_N-\mu_N)] & \cdots & E[(x_N-\mu_N)^2] \end{bmatrix} =$$

$$= \begin{bmatrix} \sigma_1^2 & \cdots & c_{1N} \\ \vdots & \ddots & \vdots \\ c_{1N} & \cdots & \sigma_N^2 \end{bmatrix}$$

Covariance Matrix

Representing Covariance between dimensions as a matrix e.g. for 3 dimensions:

$$C = \begin{bmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{bmatrix}$$

Variances

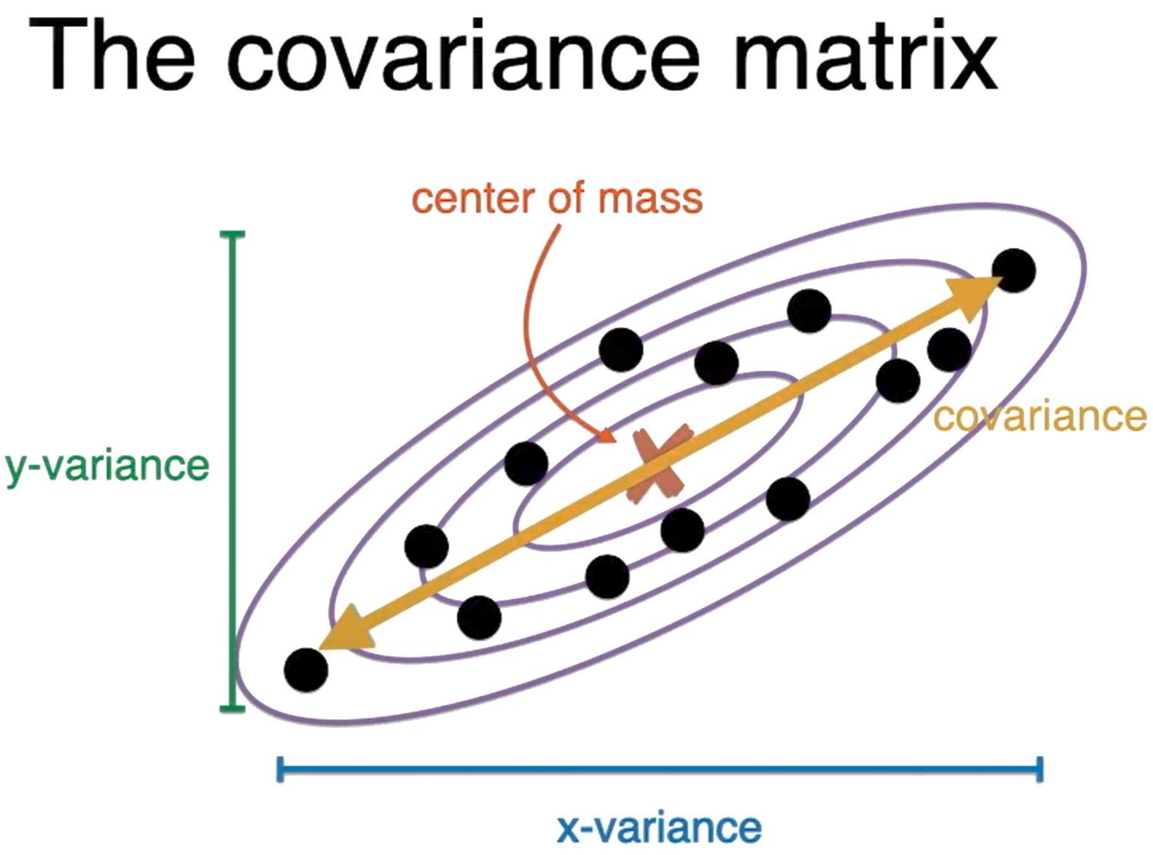
Diagonal is the variances of x, y and z

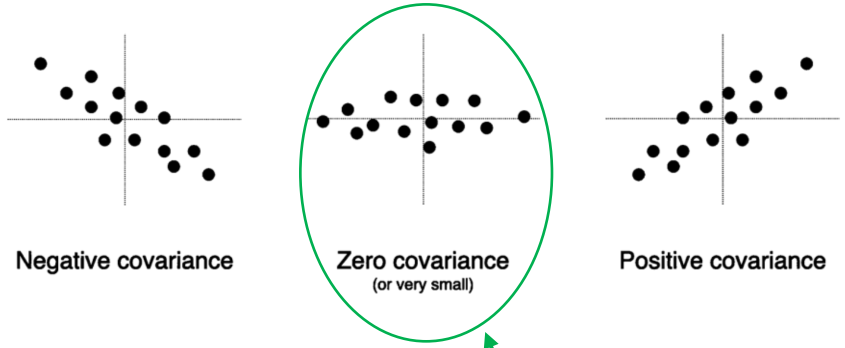
- cov(x,y) = cov(y,x) hence matrix is symmetrical about the diagonal
- N-dimensional data will result in NxN covariance matrix

You also can refer to my video!

Covariance matrix

We could find the relation between features in dataset





We want independent features

$$\mu = Average$$

$$\Sigma = \begin{pmatrix} Var(x) & Cov(x,y) \\ Cov(x,y) & Var(y) \end{pmatrix}$$

Now Lets compute the covariance matrix for the first question!

We have 3 feature vector:

Vector X_1: [4.0, 4.2, 3.9, 4.3, 4.1]

Vector X_2: [2.0, 2.1, 2.0, 2.1, 2.2]

Vector X_3: [0.6, 0.59, 0.58, 0.62, 0.63]

step 0 : we compute the mean vector:

$$\mu_1 = \frac{4.0 + 4.2 + 3.9 + 4.3 + 4.1}{5} = \frac{20.5}{5} = 4.1$$

$$\mu_2 = \frac{2.0 + 2.1 + 2.0 + 2.1 + 2.2}{5} = \frac{10.4}{5} = 2.08$$

$$\mu_3 = \frac{0.6 + 0.59 + 0.58 + 0.62 + 0.63}{5} = \frac{3.02}{5} = 0.604$$

$$\vec{\mu} = [\mu_1, \mu_2, \mu_3]$$

$$= [4{,}1 \,,\, 2{,}08 \,,\, 0{,}604]$$

step 1 : computation of the diagonal elements:

- the diagonal elements of a covariance matrix represent the variance within each feature

$$C_{ii} = \frac{1}{n} \sum_{i=1}^{n} (\mu_i - x_i)^2 , \quad i = 1, \dots, n$$

$$C_{11} = \frac{(4.0-4.1)^2 + (4.2-4.1)^2 + (3.9-4.1)^2 + (4.3-4.1)^2 + (4.1-4.1)^2}{5} = 0.02$$

$$C_{22} = \frac{(2.0-2.08)^2 + (2.1-2.08)^2 + (2.0-2.08)^2 + (2.1-2.08)^2 + (2.2-2.08)^2}{5} = 0.0056$$

$$C_{33} = \frac{(0.6-0.604)^2 + (0.59-0.604)^2 + (0.58-0.604)^2 + (0.62-0.604)^2 + (0.63-0.604)^2}{5} = 0.000344$$

step 2 : Now we compute off-diagonal elements:

- the diagonal elements of a covariance matrix represent the variance within each feature

$$C_{ij} = \frac{1}{n} \sum_{k=1}^{n} (\mu_i - x_i^k)(\mu_j - x_j^k) \quad , \quad i,j = 1, \dots, n$$

$$C_{12} = \frac{(4.0-4.1)(2.0-2.08)+(4.2-4.1)(2.1-2.08)+(3.9-4.1)(2.0-2.08)+(4.3-4.1)(2.1-2.08)+(4.1-4.1)(2.2-2.08)}{5} = 0.006$$

$$C_{13} = \frac{(4.0-4.1)(0.6-0.604)+(4.2-4.1)(0.59-0.604)+(3.9-4.1)(0.58-0.604)+(4.3-4.1)(0.62-0.604)+(4.1-4.1)(0.63-0.604)}{5} = 0.0014$$

$$C_{23} = \frac{(2.0-2.08)(0.6-0.604)+(2.1-2.08)(0.59-0.604)+(2.0-2.08)(0.58-0.604)+(2.1-2.08)(0.62-0.604)+(2.2-2.08)(0.63-0.604)}{5} = 0.00108$$

- **We also know that covariance matrix are symmetric!**

M.Tamjidi

step 3 : Part B !

- Here is our covariance matrix

$$\Sigma = \begin{bmatrix} 0.02 & 0.006 & 0.0014 \\ 0.006 & 0.0056 & 0.00108 \\ 0.0014 & 0.00108 & 0.000344 \end{bmatrix}$$

We also can verify our answer in NumPy !

```
[13] X1 = [4, 4.2, 3.9, 4.3, 4.1]
     X2 = [2, 2.1, 2, 2.1, 2.2]
     X3 = [0.6, 0.59 ,0.58, 0.62, 0.63]
     import numpy as np
     A = np.vstack([X1, X2, X3])
     cov=np.cov(A)
     covariance_matrix = cov *(4/5) #using n instead of n-1 for averaging in the numpy.cov package
     covariance_matrix

     array([[0.02    , 0.006    , 0.0014   ],
            [0.006   , 0.0056   , 0.00108  ],
            [0.0014  , 0.00108  , 0.000344]])

[15] np.mean(A,axis=1)

     array([4.1  , 2.08 , 0.604])
```

Part B : to check the singularity and positive definite property

We can easily check the determinant of matrix!

$$\det(\Sigma) = \begin{vmatrix} 0.02 & 0.006 & 0.0014 \\ 0.006 & 0.0056 & 0.00108 \\ 0.0014 & 0.00108 & 0.000344 \end{vmatrix} = 9.984 \times 10^{-9}$$

The matrix is non-singular matrix (but near singular) because it's determinant is non-zero.

```
[45] det=np.linalg.det(covariance_matrix)
     det

     9.984000000000006e-09
```

Which means that their columns are nearly dependent but altogether we call that independent but correlated feature.

Part B - First approach : to see the positive definity we can check it's eigen value

- If all eigenvalues are positive, then the quadratic form is positive definite.
- If some eigenvalues are zero and some are positive or negative, then the quadratic form is semi-definite.

And the matrix is positive definite because it's eigenvalues are bigger than zero. And also it's symmetric and all its eigenvalues are positive. This covariance matrix has positive and nonzero eigen value and also it's symmetric so we can call it positive definite.

```
lambdas,M=np.linalg.eig(covariance_matrix)
lambdas,M
```

λ_1 λ_2 λ_3

```
(array([0.02230155, 0.00351509, 0.00012736]),
 array([[ 0.93676841,  0.34958469, -0.0159843 ],
        [ 0.34148069, -0.92313136, -0.1766902 ],
        [ 0.0765238 , -0.16005947,  0.98413672]]))
```

eigenvalues > 0

P.D

Part B - Second approach : to see the positive definity we can check it's eigen value

The other way to prove that it's positive definite is to solving the equation (1) for covariance matrix.

Lets do it!

– For a square matrix A

- if $x^T Ax > 0 \quad \forall x \neq 0$, then A is said to be positive-definite (i.e., the covariance matrix)
- $x^T Ax \geq 0 \quad \forall x \neq 0$, then A is said to be positive-semi-definite

$$\underset{1\times 3}{x^T} \underset{3\times 3}{A} \underset{3\times 1}{X} > 0 \quad (1)$$

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} > 0$$

$$\begin{bmatrix} x_1A_1 + x_2D + x_3G & x_1B + x_2E + x_3H & x_1C + x_2F + x_3I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} =$$

$$= \left[x_1^2A + x_1x_2D + x_1x_3G + x_1x_2B + x_2^2E + x_2x_3H + x_3x_1C + x_2x_3F + x_3^2I \right]$$

covariance is symmetric

$$\boxed{Ax_1^2 + Ex_2^2 + Ix_3^2 + \overbrace{(B+D)}^{2D}x_1x_2 + \overbrace{(G+C)}^{2C}x_1x_3 + \overbrace{(H+F)}^{2H}x_2x_3} \quad (*)$$

Where A=0.02 ,E=0.0056,I=000344, B=0.006,C=0.0014,H=0.00108 considering all nonzero.

Sub

$$\Rightarrow 0.02x_1^2 + 0.0056x_2^2 + 0.00344x_3^2 + 0.012x_1x_2 + 0.0028x_1x_3 + 0.00216x_2x_3 \Rightarrow$$

$$\Rightarrow (0.02x_1 + 0.03x_2)^2 + (0.07x_1 + 0.02x_2)^2 + (0.03x_2 + 0.036x_3)^2 + 0.147x_1^2 +$$

$$+ 0.0038x_2^2 + 0.001744x_3^2$$

$\Rightarrow$ All coefs are positive & non-negative

M.Tamjidi

$$\begin{bmatrix} x_1A_1 + x_2D + x_3G & x_1B + x_2E + x_3H & x_1C + x_2F + x_3I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} =$$

$$= \begin{bmatrix} x_1^2A + x_1x_2D + x_1x_3G + x_1x_2B + x_2^2E + x_2x_3H + x_3x_1C + x_2x_3F + x_3^2I \end{bmatrix}$$

covariance is symmetric

$$\boxed{Ax_1^2 + Ex_2^2 + Ix_3^2 + \overbrace{(B+D)}^{2D}x_1x_2 + \overbrace{(G+C)}^{2C}x_1x_3 + \overbrace{(H+F)}^{2H}x_2x_3} \quad (*)$$

Where A=0.02 ,E=0.0056,I=000344, B=0.006,C=0.0014,H=0.00108 considering all nonzero.

$$\underbrace{Dx_1^2 + Dx_2^2 + 2Dx_1x_2}$$

$\Longleftarrow$ Just look the way we simplified it.

$$2Dx_1x_2 = \underline{D(x_1+x_2)^2} - Dx_1^2 - Dx_2^2$$

M.Tamjidi

Part C - First approach : to see the positive definity we can check it's eigen value

To obtain a diagonal covariance matrix, we need to use the covariance matrix to find the modal matrix (assuming our data follows a Gaussian distribution). By doing so, we can transform our data to have uncorrelated features.
To achieve this, we calculate the eigenvalues manually, as shown below.

step 1 : First we will find eigen values

$$\Sigma = \begin{bmatrix} 0.02 & 0.006 & 0.0014 \\ 0.006 & 0.0056 & 0.00108 \\ 0.0014 & 0.00108 & 0.000344 \end{bmatrix}$$

We know that : $AV = \lambda V$ which $V \neq 0$

$\Rightarrow AV = \lambda V \longrightarrow (A - \lambda I)V = 0 \longrightarrow |A - \lambda I| = 0$

$$|AV - \lambda I| = \begin{vmatrix} 0,02-\lambda & 0,006 & 0,0014 \\ 0,006 & 0,0056-\lambda & 0,00108 \\ 0,0014 & 0,00108 & 0,000344-\lambda \end{vmatrix} = -\lambda^3 + 0,025944\lambda^2 - 0,00008\lambda + 9,984 \times 10^{-9}$$

$\boxed{\lambda_2 = 0,00012736}$ $\boxed{\lambda_2 = 0,003515}$ $\boxed{\lambda_3 = 0,022305}$

M.Tamjidi

step 2 : Then we will find corresponding eigen values

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

$$u_1 = \frac{v_1}{\|v_1\|} = [0{,}93676,\ 0{,}34958,\ -0{,}0159]$$

$$u_2 = \frac{v_2}{\|v_2\|} = [0{,}34148,\ -0{,}92313,\ -0{,}17669]$$

$$u_3 = \frac{v_3}{\|v_3\|} = [0{,}07652,\ -0{,}16005,\ 0{,}98413]$$

We use the normalized eigen vectors for making diagonal matrix of covariance.

$$M = [v_1, v_2, v_3] \longrightarrow M^T = \begin{bmatrix} 0{,}9367 & 0{,}3495 & -0{,}0159 \\ 0{,}34148 & -0{,}92313 & -0{,}1766 \\ 0{,}076 & -0{,}16005 & 0{,}1984 \end{bmatrix}$$

step 3 : Now lest remove the correlations

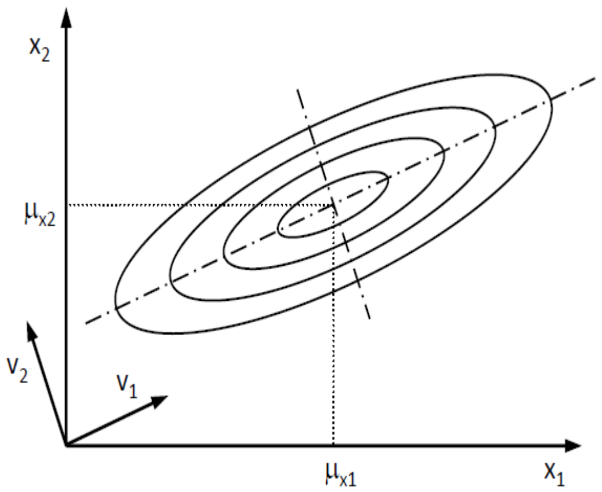
Note that for the verification we can see that M matrix has orthogonal rows and columns.

Meaning that M.MT=0 And also we can verify our finding in NumPy.

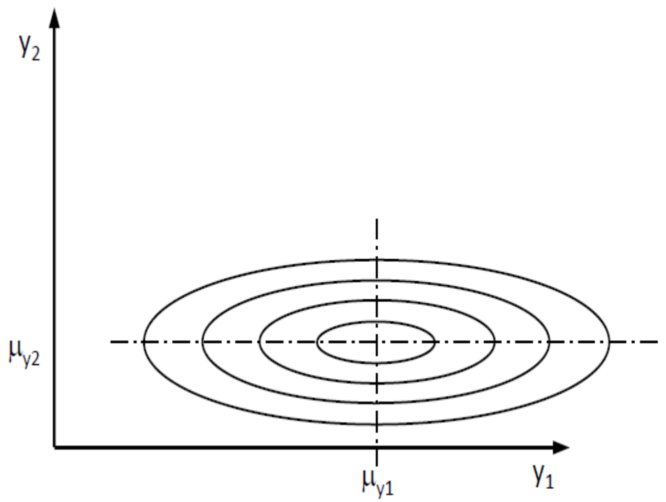
$$\Sigma M = M\Lambda \text{ with } M = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ v_1 & v_2 & & v_N \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} \quad \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix}$$

$$f_x(x) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right]$$

$$f_y(y) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left[-\frac{(y_i-\mu_{y_i})^2}{2\lambda_i}\right]$$



$$y = M^T x$$



M.Tamjidi

```
[150] uncrolated=M.T@A
      uncrolated
```

```
array([[ 4.47594928,  4.69668579,  4.38074197,  4.79265835,  4.64021798],
       [-0.54395965, -0.56475525, -0.57571693, -0.53459857, -0.69842923],
       [ 0.17316444,  0.14245719,  0.15508013,  0.17038286,  0.16575207]])
```

```
[154] new_cov_matrix=np.cov(uncrolated)
      new_cov_matrix
```

```
array([[ 2.78769350e-02,  3.50204060e-18, -4.85455790e-18],
       [ 3.50204060e-18,  4.39386507e-03, -3.64092268e-19],
       [-4.85455790e-18, -3.64092268e-19,  1.59199930e-04]])
```

```
np.round(new_cov_matrix,5)
```

```
array([[ 0.02788,  0.     , -0.     ],
       [ 0.     ,  0.00439, -0.     ],
       [-0.     , -0.     ,  0.00016]])
```

And we can see that the multiplication of MT to our data can make our new covariance diagonal which means that our data are now uncorrelated.

Question 2

۲– یک ویروس نادر در نظر بگیرید که یک نفر از هر ۱۰۰۰ نفر را آلوده می کند. یک آزمایش خوب اما نه کاملا مطمئن برای تشخیص بیماری ناشی از این ویروس وجود دارد که برای آن **specificity , sensitivity** به ترتیب برابر ۹۸٪ و ۹۹٪ است. نتیجه آزمایش شخصی مثبت بوده است. احتمال اینکه شخص با این ویروس آلوده شده باشد، چقدر است؟

- **Prevalence of Condition**: $P(\text{Condition}) = \frac{1}{1000} = 0.001$ ← Prior
- **Sensitivity** (True Positive Rate): $P(\text{Positive Test}|\text{Condition}) = 0.98$
- **Specificity** (True Negative Rate): $P(\text{Negative Test}|\text{No Condition}) = 0.99$

We want posterior probability : $P(\text{Condition}|\text{Positive Test})$

Question 2

$P(\text{cond}) = \frac{1}{1000}$

$\text{Specificity} = P(\text{Neg} \mid -\text{cond}) = 0{,}98$ ①

$\text{Sensitivity} = P(\text{Pos} \mid \text{cond}) = 0{,}99$ ②

$P(\text{cond} \mid +) = ?$

② $\rightarrow P(\text{neg} \mid \text{cond}) = 0{,}01$

① $\rightarrow P(\text{Pos} \mid -\text{cond}) = 0{,}2$

$$P(\text{cond} \mid +) = \frac{P(+ \mid \text{cond})\, P(\text{cond})}{P(+)} \quad (*)$$

$P(+) = P(+, \text{cond}) + P(+, -\text{cond})$

$P(+) = P(+ \mid \text{cond})\, P(\text{cond}) + P(+ \mid -\text{cond})\, P(-\text{cond})$

$$P(+) = 0{,}99 \times \frac{1}{1000} + 0{,}2 \times \frac{999}{1000}$$

M.Tamjidi

$\Rightarrow P(+) = 0.02079$

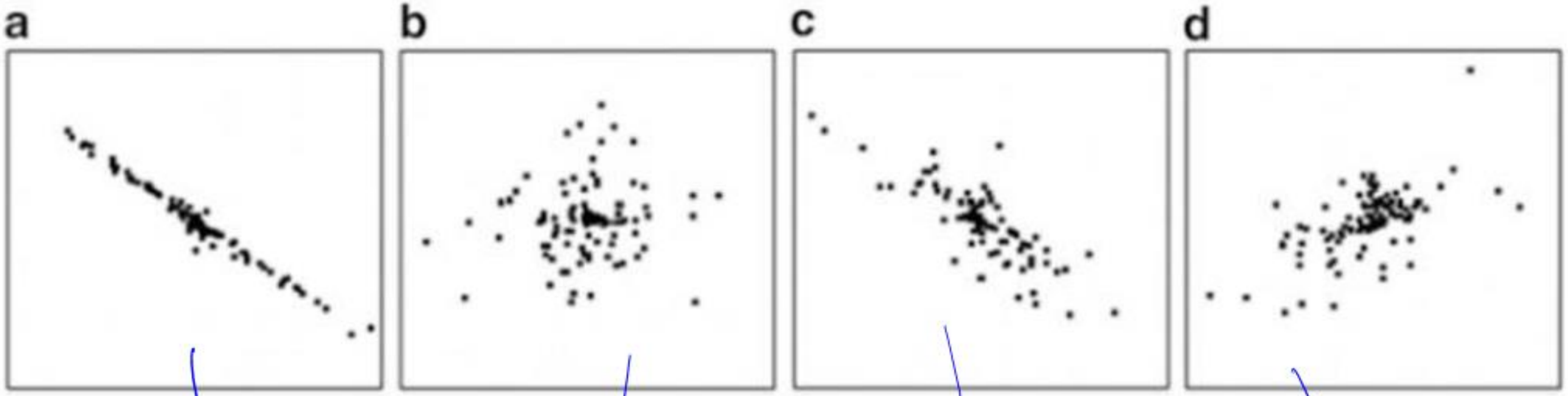
according to $\circledast$ $\Rightarrow$

$$P(\text{cond} \mid +) = \frac{\overbrace{P(+\mid \text{cond})}^{0.99}\ \overbrace{P(\text{cond})}^{\frac{1}{100}}}{\underbrace{P(+)}_{0.02079}} = \boxed{0.04762}$$

final ans

M.Tamjidi

۳– نمودار پراکندگی یک ویژگی بر حسب یک ویژگی دیگر در شکلهای زیر در چهار حالت نشان داده است. چهار مقدار ۰,۱٤ و ۰,۹۹– ، ۰,٤۳ و ۰,۷۷– برای ضریب همبستگی در نظر گرفته شده است. این مقادیر را به شکل مناسب منتسب کنید.



a : -0/99 b : 0/14 c : -0/77 d : 0/43

M.Tamjidi

۴– کدام یک از مجموعه بردارهای زیر مستقل خطی هستند؟

a.

$$x_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$$

b.

$$x_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Part A

The first one is **linearly dependent**. the reason is that they are linear combination of each other.

$2x_1 - x_2 = x_3$ or $\underline{2x_1 - x_2 - x_3 = 0}$

dependency

$$x_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$$

in general case the vectors are nonlinear if $\rightarrow$ $\boxed{\sum a_i x_i = 0 \quad \forall i \in \mathbb{R}}$

The other way is also checking out it's determinant which is equal to zero.

M.Tamjidi

We also can check it with NumPy

```python
import numpy as np

x1 = np.array([2, -1, 3])
x2 = np.array([1, 1, -2])
x3 = np.array([3, -3 ,8])

matrix = np.vstack([x1, x2, x3])

# Perform row reduction (Gaussian elimination)

# Check the rank of the reduced row echelon form (rref) matrix
rank = np.linalg.matrix_rank(matrix)

if rank == 3:
    print("The vectors are linearly independent. rank= ",rank)
else:
    print("The vectors are linearly dependent. rank= ",rank)
```

```
The vectors are linearly dependent. rank=  2
```

We also can check it with NumPy

Part B

The second one is **linearly independent**. Because they cannot find a linear combination of them and they form a basis for their column space for no coefficient in **R.**

we can also can find out with gaussian elimination and then checking it’s rank

```
import numpy as np

x1 = np.array([1, 2, 1, 0, 0])
x2 = np.array([1, 1, 0, 1, 1])
x3 = np.array([1, 0, 0, 1, 1])

matrix = np.vstack([x1, x2, x3])

# Perform row reduction (Gaussian elimination)

# Check the rank of the reduced row echelon form (rref) matrix
rank = np.linalg.matrix_rank(matrix)

if rank == 3:
    print("The vectors are linearly independent. rank= ",rank)
```

```
The vectors are linearly independent. rank=  3
```

So the part B is only linearly independent

۵– تبدیل خطی زیر را در نظر بگیرید

$$\Phi : \mathbb{R}^3 \to \mathbb{R}^4$$

$$\Phi\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} 3x_1 + 2x_2 + x_3 \\ x_1 + x_2 + x_3 \\ x_1 - 3x_2 \\ 2x_1 + 3x_2 + x_3 \end{bmatrix}$$

ماتریس تبدیل A و رتبه آن را تعیین کنید.

Answer

This matrix reduces the dimension by a linear multiplication. Because the function is linear it can also be denoted by a matrix A !

$$\underset{4\times3}{\bar{A}}\;\underset{3\times1}{\bar{X}} = \underset{4\times1}{\bar{B}}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 3X_1 + 2X_2 + X_3 \\ X_1 + X_2 + X_3 \\ X_1 - 3X_2 \\ 2X_1 + 3X_2 + X_3 \end{bmatrix}$$

⇒ by assuming the correspondence we can find the coefficients

So the transformation is :

$$A = \begin{bmatrix} 3 & 2 & 1 \\ 1 & 1 & 1 \\ 1 & -3 & 0 \\ 2 & 3 & 1 \end{bmatrix}$$

This matrix has three independent column and only one dependent row **(row 4 can be write as a linear combination of the others)** So there are three independent row and three independent columns

$\text{rank } A = \boxed{3}$

We can also verify our claim by numpy!

```
import numpy as np

matrix = np.array([[3, 2, 1],
                   [1, 1, 1],
                   [1, -3, 0],
                   [2, 3, 1]])

rank = np.linalg.matrix_rank(matrix)

print("Rank of the matrix:", rank)
```

```
Rank of the matrix: 3
```

```python
import numpy as np

# Define the matrix
matrix = np.array([[3, 2, 1],
                   [1, 1, 1],
                   [1, -3, 0],
                   [2, 3, 1]])

# Perform row reduction (Gaussian elimination)

# Check the rank of the reduced row echelon form (rref) matrix
rank = np.linalg.matrix_rank(matrix)

if rank == 3:
    print("The rows of the matrix are linearly independent.")
else:
    print("The rows of the matrix are linearly dependent.")
```

```
The rows of the matrix are linearly independent.
```

Thank you!

Feel free to contact me

Mehrant.0611@gmail.com
Telegram: @Mttnt