UNIVERSITÉ DE GENÈVE

ANALYSE ET TRAITEMENT DE L'INFORMATION
14X026

TP 1: Linear Algebra

Author: Joao Costa

E-mail: Joao.Costa@etu.unige.ch

October 2022

UNIVERSITÉ
DE GENÈVE
FACULTÉ DES SCIENCES
Département d'informatique

Exercise 1. Matrix

1. Find a quadratic polynomial, say $f(x) = ax^2 + bx + c$, such that

$$f(1) = 1, \quad f(2) = 9, \quad f(3) = 27$$

$$\begin{cases} a+b+c=1 \\ 4a+2b+c=9 \\ 9a+3b+c=27 \end{cases} \Leftrightarrow \begin{cases} a=5 \\ b=-7 \\ c=3 \end{cases}$$

(1) We can start by writing the first equation: $\longrightarrow c = 1-a-b$

(2) insert step (1) in equation 2 and write it: $\longrightarrow 4a+2b+1-a-b=9 \Leftrightarrow 3a+b=8 \Leftrightarrow b=8-3a$

(3) Rewrite $c = 1-a-b$: $\longrightarrow c = 1-a-(8-3a) \Leftrightarrow c = -7+2a$

(4) Insert equations from (3) and (2) into equation 3: $\longrightarrow 9a+3*(8-3a)+(-7+2a)=27$

(5) Solve step (4): $\longrightarrow 9a+24-9a-7+2a=27 \Leftrightarrow a=5$

(6) Insert step (5) into step (2): $\longrightarrow b = 8-3*5 \Leftrightarrow b=-7$

(7) Insert steps (5) and (6) into step (1): $\longrightarrow c = 1-5+7 \Leftrightarrow c=3$

2. Let a, b be some fixed parameters. Solve the system of linear equations

$$\begin{cases} x+ay=2 \\ bx+2y=3 \end{cases}$$

$$\begin{cases} x+ay=2 \\ bx+2y=3 \end{cases} \Leftrightarrow \begin{cases} x=2-ay \\ bx+2y=3 \end{cases}$$

We can now insert the 1^{st} equation into the 2^{nd} equation as follows:
$b(2-ay)+2y=3 \Leftrightarrow 2b-aby+2y=3 \Leftrightarrow 2b-y(ab-2)=3 \Leftrightarrow y=(-1)*\left(\frac{3-2b}{(ab-2)}\right)$
With given a and b values we can easily compute y, then we reinsert y into the first equation to get x.

Exercise 2. The importance of the mathematical concept behind a code

1. Download, open and run several times the PYTHON file "TP1/some script.py".

   Explain the function `def project_on_first(u, v)` in geometrical terms and then explain the results of this code with clear mathematical concepts (you have to speak about projections and scalar product). *It is very important to understand the mathematical concepts behind a program!!*

$def\ project\_on\_first(u,v)$ receives two column vectors as an argument, and it projects v onto u, the projected vector is usually called v', in the image bellow it is represented by $\vec{w}_1$. Visually, it means that v' and u are collinear. This also means that: $\exists\ \alpha \in \mathbb{R}$ tq. $v' * \alpha = u$.

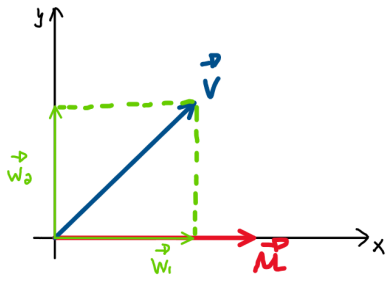

Figure 1: Projection of $\vec{v}$ onto $\vec{u} = \vec{w}_1$

2. Let's consider $u = (u_i)_{1 \leq i \leq 5}$ and $v = (v_i)_{1 \leq i \leq 5}$, two vectors of length 5. How would you clearly write the following part of script using some mathematical notations.

```
# import numpy library as np in order to work on tensors
import numpy as np
# create 2 random vectors of length 5
u = np.random.randn(5)
v = np.random.randn(5)
# perform some computations
r = 0
for ui, vi in zip(u, v) :
    r += ui * vi
```

Name the mathematics behind and rewrite the last 3 lines of code in one.

---

zip() function takes as argument two python lists of same size. It then merges one value from the first list, with another value from the second list (same index), creating a list of tuples.
Let's see an example:

x = zip([1,2], [3,4]) -> x = [(1,3),(2,4)]

This means that the three last lines of code perform a simple dot operation between the two vectors given as argument to zip().

It can be rewritten as: r = np.dot(u,v)

3. Complete the code to construct a vector v orthogonal to the vector u and of the same norm. Comment each line of your code.

Step 1: find the vector $\vec{w}_2$ orthogonal to $\vec{u}$
If we look at Figure 1, we can see that $\vec{w}_1$ is collinear to $\vec{u}$, and that $\vec{w}_2$ is orthogonal to $\vec{u}$. Moreover, $\vec{v} = \vec{w}_1 + \vec{w}_2$, which means we can easily compute $\vec{w}_2$ if we have already computed $\vec{w}_1$.

$$\vec{w}_2 = \vec{v} - \vec{w}_1$$

Step 2: Make it so the orthogonal vector $\vec{w}_2$ has the same norm as vector $\vec{u}$
We must first compute $||\vec{u}||$ as well as $||\vec{w}_2||$.
By multiplying $\vec{w}_2$ by a given real value α we can find a new vector $\vec{w}'_2$ that is collinear to $\vec{w}_2$, but of different norm.

$$\alpha = ||\vec{u}|| \ / \ ||\vec{w}_2||$$

$def\ orthogonal\_norm\_on\_first(u, v)$ is the function inside $some\_script.py$ that does this computation.

Exercise 3. Computing Eigenvalues, Eigenvectors, and Determinants

1. Find the determinant, eigenvectors, and eigenvalues of the matrix

$$\begin{bmatrix} 1 & 1 & 3 \\ 1 & 5 & 1 \\ 3 & 1 & 1 \end{bmatrix}$$

Compare your results with computer's one.

$$Det(A) = Det\begin{pmatrix} 1 & 1 & 3 \\ 1 & 5 & 1 \\ 3 & 1 & 1 \end{pmatrix}$$

$1 * Det\begin{pmatrix} 5 & 1 \\ 1 & 1 \end{pmatrix} - 1 * Det\begin{pmatrix} 1 & 1 \\ 3 & 1 \end{pmatrix} + 3 * Det\begin{pmatrix} 1 & 5 \\ 3 & 1 \end{pmatrix} = 1 * (5-1) - 1 * (1-3) + 3 * (1-15) = 4 + 2 - 42 = -36$
The computer returned the same value.
To compute eigenvalues and eigenvectors we have to compute the characteristic polynomial, P(A):

$$P(A) = Det(A - I\lambda) = 0 <=> Det\begin{pmatrix} (1-\lambda) & 1 & 3 \\ 1 & (5-\lambda) & 1 \\ 3 & 1 & (1-\lambda) \end{pmatrix} = 0$$

$(1-\lambda) * Det\begin{pmatrix} (5-\lambda) & 1 \\ 1 & (1-\lambda) \end{pmatrix} - 1 * Det\begin{pmatrix} 1 & 1 \\ 3 & (1-\lambda) \end{pmatrix} + 3 * Det\begin{pmatrix} 1 & (5-\lambda) \\ 3 & 1 \end{pmatrix}$
$= (1-\lambda) * (4 - 6\lambda + \lambda^2) - 1 * (2 + \lambda) + 3 * (1 - 15 + 3\lambda) = -36 + 7\lambda^2 - \lambda^3 = -(\lambda + 2)(\lambda - 3)(\lambda - 6)$
This means our eigenvalues are: $\{-2, +3, +6\}$, now we just have to compute the respective eigenvectors. This is done in the following way:

$$(A - I\lambda) = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \lambda = \{-2, +3, +6\}$$

Once we solve it we get the respective eigenvectors:

$$\lambda = 6 \longrightarrow \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}, \lambda = 3 \longrightarrow \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}, \lambda = -2 \longrightarrow \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

2. The covariance matrix for n samples $x_1, \ldots, x_n$, each represented by a $d \times 1$ column vector, is given by

$$C = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T,$$

where C is a $d \times d$ matrix and $\mu = \sum_{i=1}^{n} x_i / n$ is the *sample mean*. Prove that C is always positive semidefinite. (Note: A symmetric matrix C of size $d \times d$ is *positive semidefinite* if $v^T C v \geq 0$ for every $d \times 1$ vector v.)

Write your solution here TODO

3. In this portion of the exercise, we will calculate the eigenvalues of the covariance matrices of six data sets listed as follows:

filename	n	d	description
`tp1_artificialdata[1-3]`	1024	100	Artificial data generated from various auto-regression (AR-1) models
`tp1_artificialdata4`	1024	100	Random Gaussian data
`tp1_freyfaces`	1965	560	Facial images of a man named Brendan
`tp1_digit2`	5958	784	Hand-written images of "2"

To access each data set, go to "TP1/data" and download `tp1_*`. Each file contains a $n \times d$ data matrix with rows representing n different samples in $\mathbb{R}^d$ For example, `tp1_artificialdata1` contains a data matrix of size 1024×100 (1024 samples, 100 features). Once each dataset has been imported into `Python`, complete the following tasks:

(a) Compute the covariance matrix of each dataset;

(b) Compute the eigenvalues for each covariance matrix;

(c) Compute the determinant of each covariance matrix;

(d) Compute the product of the eigenvalues for each covariance matrix;

(e) Display the eigenspectrum for each covariance matrix in a 2D plot, where the x-axis shows the rank of the eigenvalues, ranging from 1 (the largest eigenvalue) to 100 (the 100-th largest eigenvalue), and the y-axis shows the corresponding eigenvalue.

Describe the relationship between the product of the eigenvalues and the determinant of each covariance matrix. In addition, describe the observations you make regarding the plot of the spectrum of eigenvalues of each covariance matrix.

$tp\_1\_exercise\_3\_3.ipynb$ is the file with all the code for this exercise.

Exercise 4. Computing Projection Onto a Line

There are given a line $\alpha : 3x - 2y = -6$ and a point A with the coordinates $(5, 4)$.

1. Find a distance from the point A to the line α
2. Explain your code using a sketch and some mathematics (there should be a scalar product somewhere).

.1

The distance is 3.61.
$def\ compute\_distance\_line\_a()$ is the function inside $some\_script.py$ that does this computation.

.2

We have a function $\alpha : 3x - 2y = -6$ that describes a line, and we have a point $A(5,4)$. Let's see the steps required to compute the distance between A and f, figure 2 is a sketch of what is explained in the item bellow:

(1) We can rewrite it as $y = f(x) = \frac{-6-3x}{-2}$

(2) Choose two values for x, and compute the respective images: $f(x_1) = y_1$ and $f(x_2) = y_2$

(3) Compute a vector $\vec{u}$ that is collinear to our function f, $\vec{u} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix}$

(4) Compute $\vec{v}$, vector between a point of the function f and our point A, $\vec{v} = \begin{bmatrix} A_x - x_1 \\ A_y - y_1 \end{bmatrix}$

(5) Compute $\vec{w}_1$ the projection of $\vec{v}$ onto $\vec{u}$, $\vec{w}_1$ and $\vec{u}$ are collinear

(6) Compute $\vec{w}_2 = \vec{v}$ - $\vec{w}_1$, $\vec{w}_2$ is orthogonal to $\vec{u}$

(7) Compute $||\vec{w}_2||$, which is the distance between the point A and the line represented by the function f

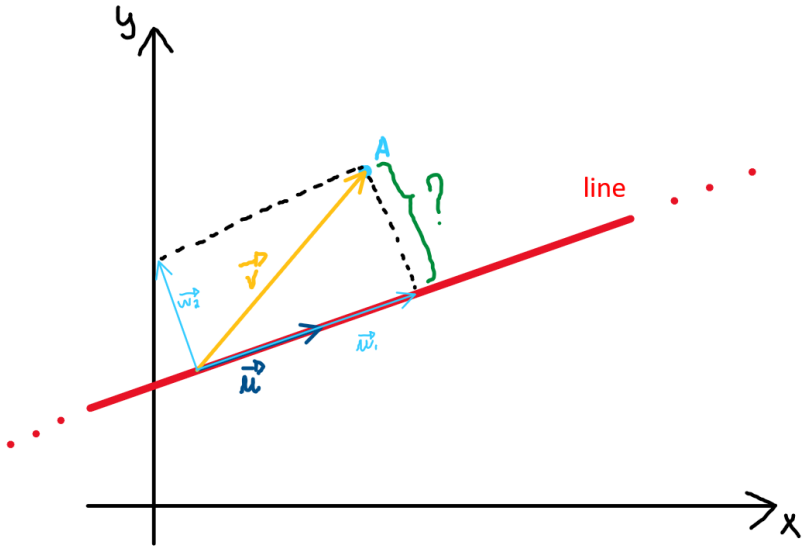

Figure 2: Distance between line and a point A

Supplements

1. Define and explain the mean and the variance on some examples.
2. Define and explain what is a vector space, a projection and a scalar product.
3. Define and explain what is a vector space, a basis and a change of basis transformation matrix.
4. Present the eigen-value decomposition and the singular-value decomposition.
5. Be ready to answer questions from slide 37 *DS.02.highDimension*.