Université de Genève

Data Science

TP 1: Linear Algebra

Author: Joao Filipe Costa da Quinta

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

September 28, 2021

UNIVERSITÉ DE GENÈVE

FACULTÉ DES SCIENCES
Département d'informatique

1 - Matrix

.1

When the number of equations, here 3, is strictly larger than the number of variables, here 2, the equations system has no solution.

.2

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

We can now insert the 1st equation into the 2nd 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,b, then we reinsert y into the first equation to get x.

2 - The importance of the mathematical concept behind a code

.1

$def\ project\_on\_first(u, v)$ receives two column vectors as an argument, and it projects v onto u, the projected vector is usually called v'. Visually, it means that v' and u are collinear. This also means that: $\exists\ \alpha$ tq. $v' * \alpha = u$.

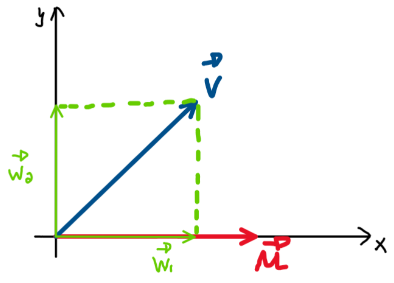

Figure 1: Projection of $\vec{v}$ onto $\vec{u} = \vec{w_1}$

.2

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

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.

3 - Computing Eigenvalues, Eigenvectors, and Determinants

.1

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

$1 * Det\begin{pmatrix}5 & 1\\ 1 & 1\end{pmatrix} - 1 * Det\begin{pmatrix}1 & 1\\ 3 & 1\end{pmatrix} + 3 * Det\begin{pmatrix}1 & 5\\ 3 & 1\end{pmatrix} = 1 * (5-1) - 1 * (1-3) + 3 * (1-15) = 4 + 2 - 42 = -36$

The computer returned the same value.

To compute eigenvalues and eigenvectors we have to compute the characteristic polynomial, P(A):

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

$$(1-\lambda) * Det\begin{pmatrix}(5-\lambda) & 1\\ 1 & (1-\lambda)\end{pmatrix} - 1 * Det\begin{pmatrix}1 & 1\\ 3 & (1-\lambda)\end{pmatrix} + 3 * Det\begin{pmatrix}1 & (5-\lambda)\\ 3 & 1\end{pmatrix}$$

$= (1-\lambda) * (4 - 6\lambda + \lambda^2) - 1 * (2 + \lambda) + 3 * (1 - 15 + 3\lambda) = -36 + 7\lambda^2 - \lambda^3$

.2

.3

4 - Computing Projection Onto a Line

.1

.2