Ning

24 September 2019

Data Science TP1

Problem 1

1. The function $deviate\_vector$ relies on the other function, $project\_on\_first$. A scalar product between two vectors is calculated by taking a projection of one vector onto another in order to evaluate how much the second vector is scaled by the first vector (the vector we end up projecting). This projection of a vector is calculated by the distance (norm) of the vector multiplied by the cosine of the angle of the vector (between the vector we are projecting to). $deviate\_vector$ itself takes the projection (from $project\_on\_first$) and subtracts it from the original vector, which is how you calculate the rejection ($a_r = a - a_p$). Thus, the function calculates the rejection of the first vector given two vectors using the notion of projection, which is key to calculating the scalar product.

2. Each for loop can be replaced with a summation notation, and the action stored by *r* is the value that we want to sum. Since we have two for loops, we will have two summation notations, going from $i = 1$ to 4, and $j = 1$ to i. The mathematical equivalent will be $\sum_{i=1}^{4}\sum_{j=1}^{i} i(a_{ij})^j$

Problem 2

1. Please see the matlab code for the calculations to the following values. A picture (problem2.PNG) is also provided for reference.

Determinant: $Det(A) = 12$

Eigenvalues: $\lambda = -2, -2,\ 3$

Potential Eigenvectors: $\begin{matrix} -0.577350269189626 \\ 0.577350269189626 \\ 0.577350269189626 \end{matrix}, \begin{matrix} 0.423274388402861 \\ 0.393035522308615 \\ 0.816309910711476 \end{matrix}, \begin{matrix} 0.218280451524610 \\ -0.790510340576316 \\ -0.572229889051707 \end{matrix}$

For the work relating to showing that the vectors are indeed eigenvectors, refer to image provided (problem_2.pdf). The eigenvalues associated with the $v1, v2, v3$ are $3, -2, -2$ respectively. The vectors $v1, v2, v3$ differ from the eigenvectors calculated by matlab.

The calculations for Pdiag([3 -2 -2]) – AP is provided in the matlab code. The expression itself is the step before the equation usually seen in the eigendecomposition, where P is a matrix of the eigenvectors of a given matrix A, and diag is a diagonal matrix with the corresponding eigenvalues to the P matrix along the diagonals. The formula is derived as follows:

$$Av = \lambda v$$

$$AP = P\Lambda$$

$$0 = P\Lambda - AP$$

Where Λ is the diagonal matrix. Our matlab code confirms that $P\Lambda - AP$ yields 0.

2. For the proof relating to the *covariance matrix*, please refer to the image provided (problem_2.pdf).

3. Please refer to the matlab code for all values calculated (problem_2.m). Eigenspectrums are attached as images (eigenspectrum0-4.PNG).

Problem 3

1. Distance is approximately 3.6055.

Given the line α, we can derive the formula for the line as follows:

$$3x - 2y = -6$$

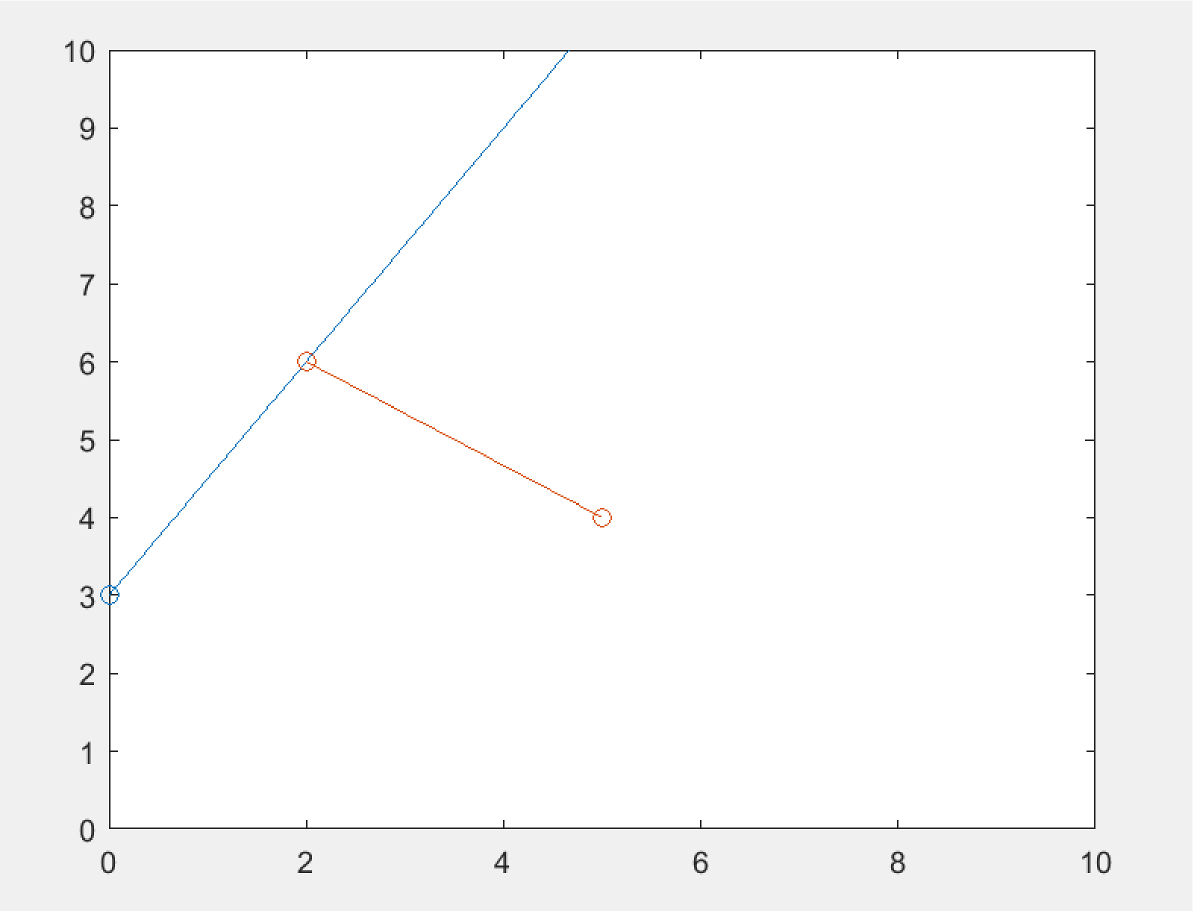
$$y = \frac{3}{2}x + 3$$

We can calculate a line that passes the point $(5,4)$ that is perpendicular to the line y:

$$\delta = -\frac{2}{3}x + \frac{22}{3}$$

We calculate the point of intersection, which is $(2,6)$. The distance between the point of intersection and the point $(5,4)$ is given by $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ (the euclidean distance). In the matlab code, we also provided the calculation for the norm of the vector that is the result of the projection of the point onto the line, which shows that the norm (in this context, magnitude) is the same as the distance.

2.


10
9
8
7
6
5
4
3
2
1
0
0
2
4
6
8
10