Ning

24 September 2019

Data Science TP1

#### Problem 1

1. The function “deviate\_vector” relies on the other function, “project\_on\_first,” which takes the second vector given and does a projection of the vector onto the first vector given and then multiplies them together, which is how you calculate the scalar product. “deviate\_vector” itself takes the projection and subtracts it from the original vector, which is how you calculate the rejection (ar = a - a*p*). So the function calculates the rejection of the first vector given two vectors.

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. The mathematical equivalent will be

#### 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:

Eigenvalues:

Eigenvectors: ,

For the work relating to showing that the vectors are indeed eigenvectors, refer to image provided (problem2.PNG). These vectors are different than the eigenvectors found above.

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:

Where is the diagonal matrix.

2. For the proof relating to the *covariance matrix*, please refer to the image provided (problem\_2.PNG).

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. For the work relating to the distance, please refer to the attached image (calculation\_point\_projection). Distance is approximately .

2. For the image related to the point projection, please see image (point\_projection.PNG).

#### [EXPLANATION]