Projet AD/CS

TP 1

Rapport

Samran Fatima Zohra

Sajid Badr

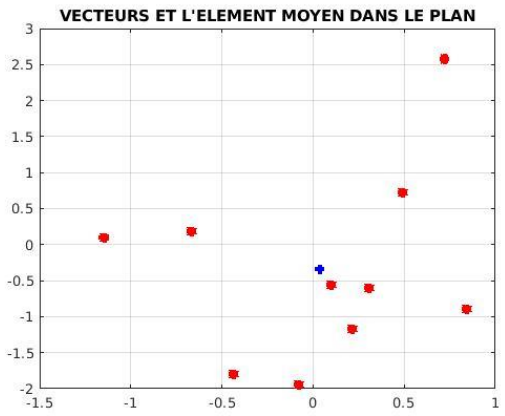
Tyoubi Anass

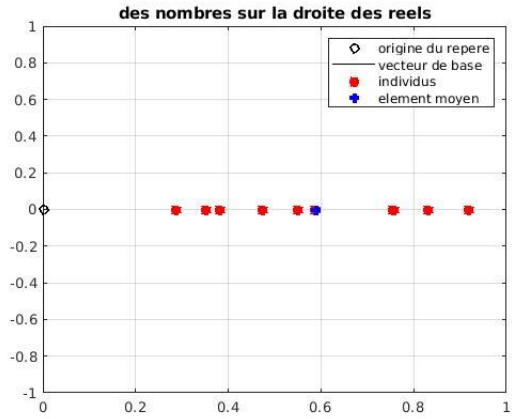
Table des matières :

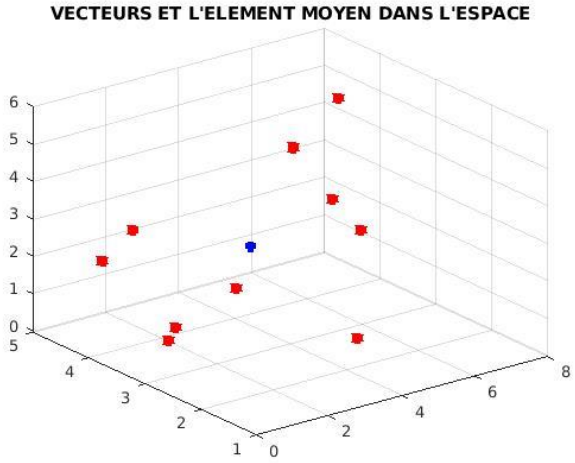
Partie 1 : Visualiser les données.

Question 1 : Les données sur lesquelles nous avions appliqué l'ACP pendant le TP n°1 d'analyse de données étaient les 3 sous matrices bidimensionnelles où on stocke une image codé en RVB (Rouge, Vert, Bleu). Le tableau de données de X dans ce TP correspondait à une matrice de taille (hauteur x largeur) x3 où on représente dans les lignes chaque couple de coordonnées (i,j) de la matrice de l'image au niveau de gris, et dans chaque colonne respectivement l'intensité lumineuse du rouge, bleu et vert de la coordonnée (i,j).

Question 2 : On complète le script *visualisation.m* et on visualise un ensemble de données soit selon un axe, un plan ou un espace.







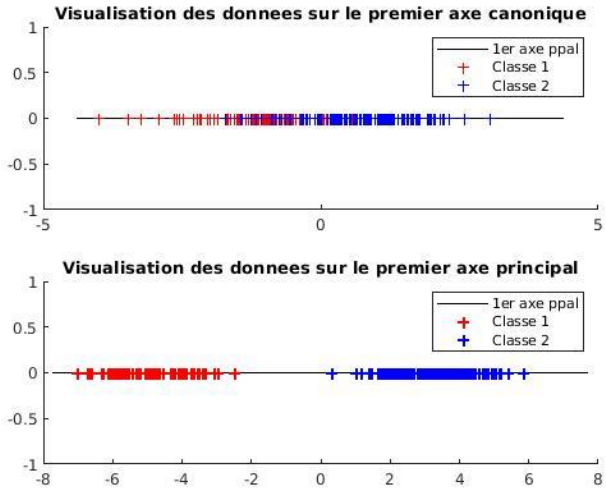
Partie 2 : L'Analyse en composantes principales.

Question 3 : On constate que la projection des individus sur les 2 premiers axes principaux est plus étalée que celle sur les deux premiers axes de la base canonique. Cela est dû au fait qu'on retrouve plus d'informations sur les individus suivant la projection sur les axes principaux. Lors de l'exécution d*'ACP.m*, on voit bien que le pourcentage d'information apportée par le premier et second axe principal excède 95%, c'est pour cela qu'une projection sur ces deux axes suffirait pour une bonne représentation des individus.

Question 4 : ∑ est symétrique réelle, d'où diagonalisable. On peut quantifier l'information contenue dans les q premières composantes principales à partir de cette matrice en la diagonalisant puis en triant dans l'ordre décroissant ses valeurs propres et leurs vecteurs propres associés. Ainsi les q nouveaux vecteurs propres sont alors les q premières composantes principales.

Partie 3 : L'ACP et la classification de données.

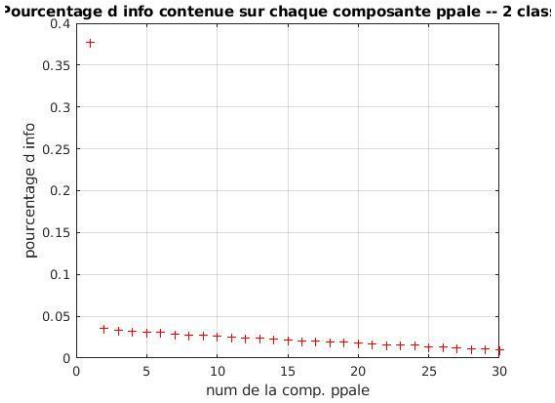
Question 5 : On complète le script *classification.m* dans les endroits précisés dans le code. On obtient :

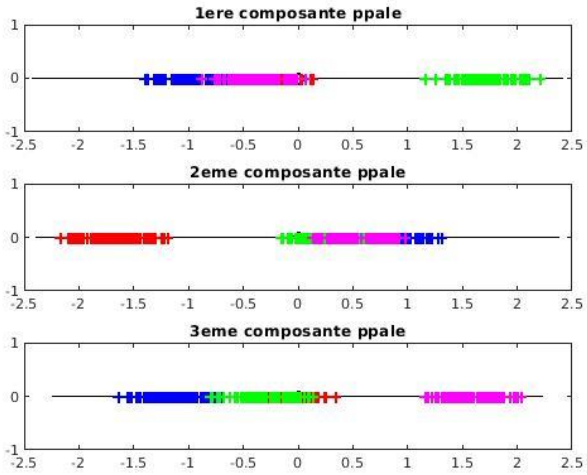


La visualisation sur le premier axe principal nous a permis de bien distinguer entre les deux classes, alors que la projection sur le premier axe canonique ne permet pas une bonne visualisation.

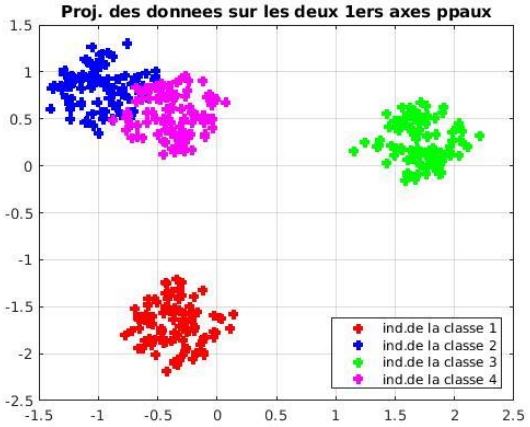
Ainsi, on n'a besoin que d'un seul axe principal pour bien percevoir les deux classes étudiées. Et cela apparaît bien dans la figure suivante :

Quand il s'agit de 4 classes, en faisant le même calcul et en projetant par exemple sur le premier, deuxième et troisième axe principal, on détecte à chaque fois une classe. Si l'on utilise les 3 projections on peut détecter les 3 classes.

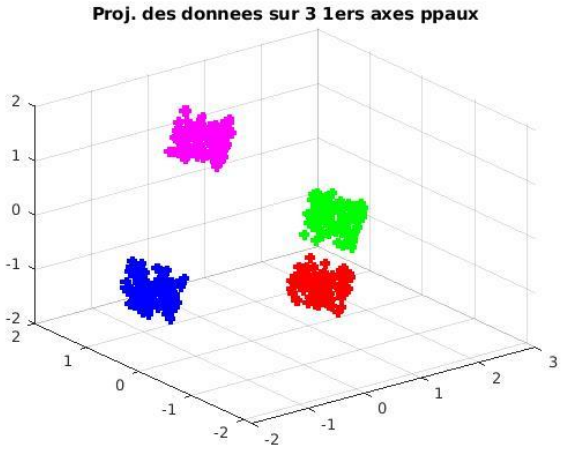




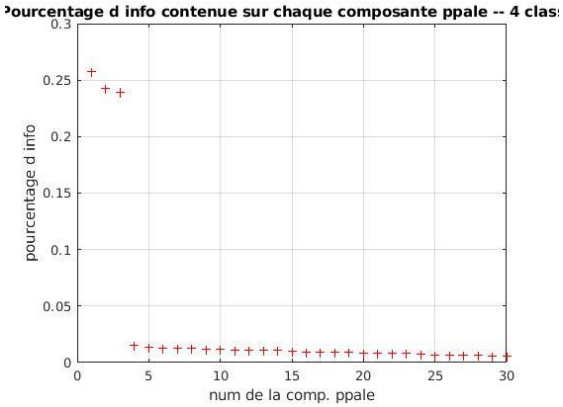
Quand on projette sur les deux premiers axes principaux dans le plan, la distinction entre les deux classes (1 et 3) est possible, mais les deux autres ne sont pas bien distinguables. Ainsi, la détection de 2 classes est claire dans le plan.



La projection dans l'espace sur les 3 axes principaux permet de bien visualiser les 4 classes et de les détecter toutes à la fois.

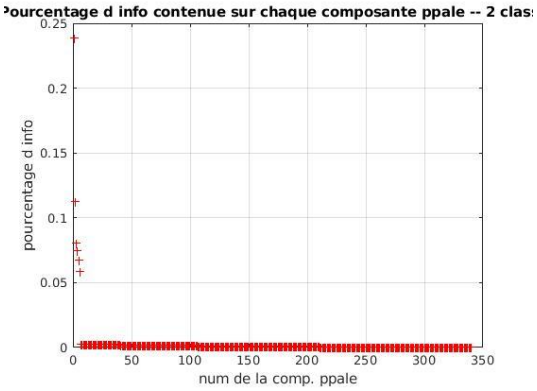


En comparant la figure ci-dessous avec la même établie précédemment, on constate que plus on a de classes à distinguer, plus on va avoir des composantes principales avec un pourcentage d'information élevé.

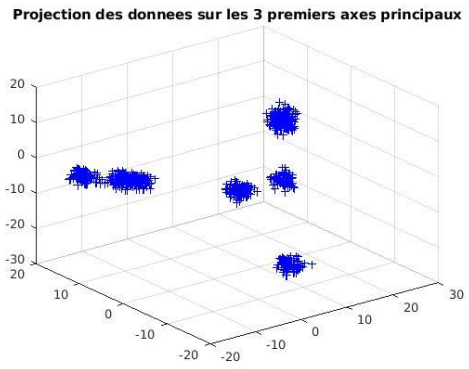


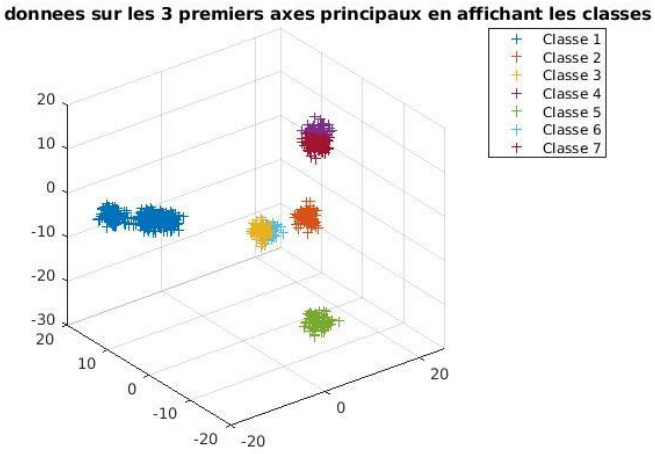
Question 6 : On peut identifier 7 classes d'individus.

On utilise la même manière utilisée dans les questions précédentes et on complète le script *classes_individus.m*. On visualise en premier le pourcentage d'information contenue dans chaque composante principale et on constate que les 6 premières contiennent plus d'information que les autres.

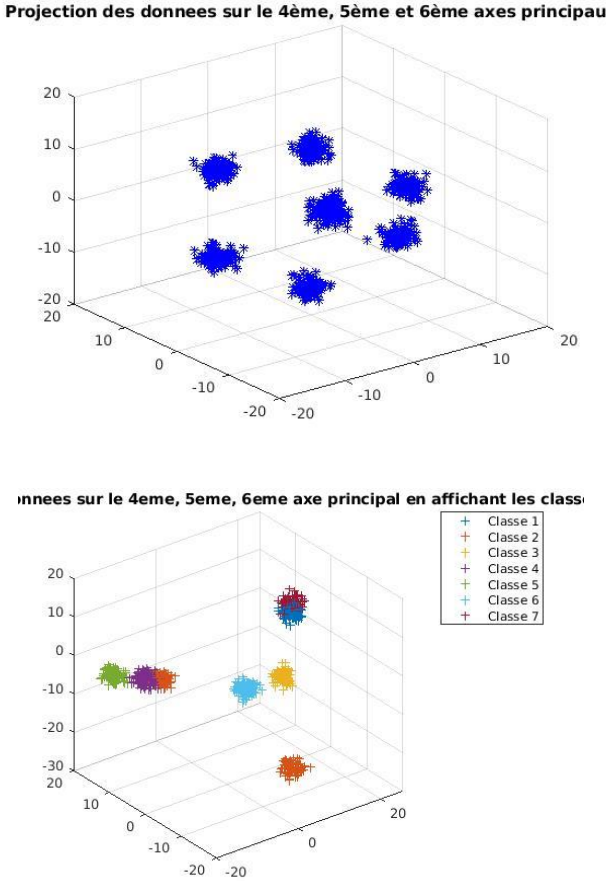


En projetant sur les 3 premières composantes, on constate qu'on a 7 classes différentes. (Une n'étant pas bien représentée dans la figure ci-dessous, on utilise la fonction kmeans et on obtient un résultat plus correct).

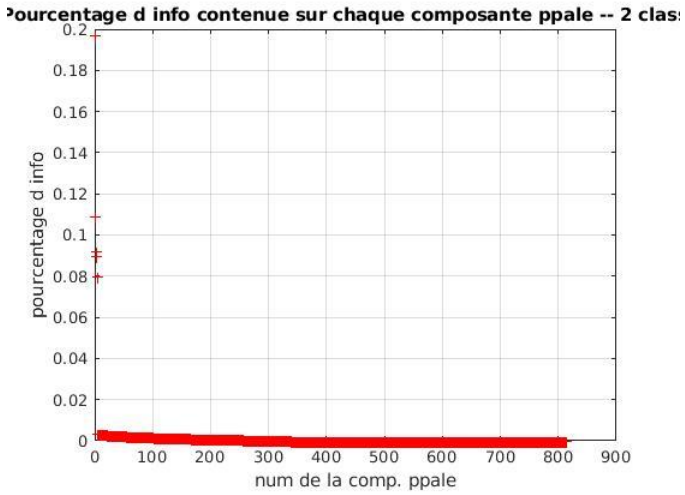


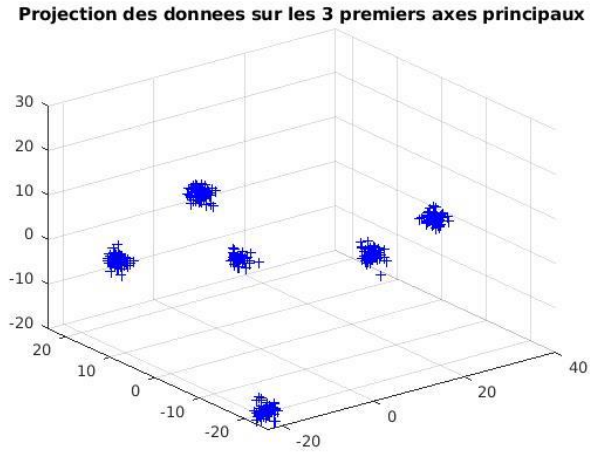


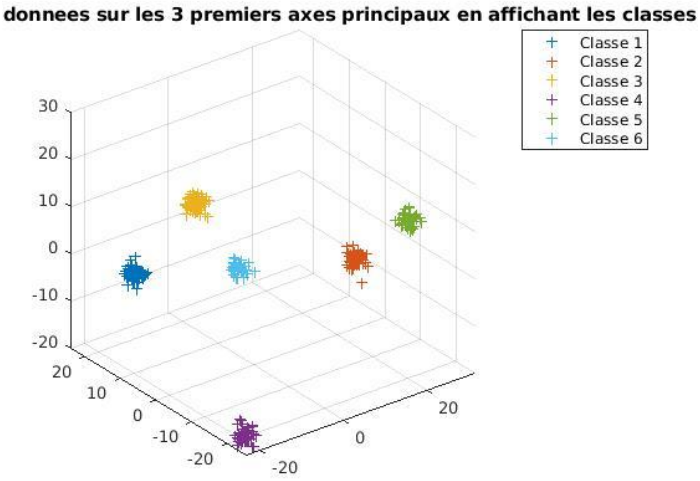
On a aussi projeté sur les 4ème, 5ème et 6ème axes principaux et on a obtenu une meilleure visualisation pour les 7 classes.



Question 7 : Cette question est similaire à la précédente. On l'a traite de la même manière en considérant maintenant la transposée de la matrice X. Cela va nous permettre de détecter le nombre de classes de variables.







On remarque qu'il y a 5 composantes principales avec un taux d'informations élevé. On projette sur les 3 premiers axes principaux et on détecte 6 classes de variables, on utilise alors la fonction kmeans pour mieux les détecter.

Partie 4 : L'ACP et la méthode de puissance itérée.

Question 8 : La matrice H dispose d'une décomposition en valeurs singulières, ces valeurs singulières au carré sont les valeurs propres non nulles des matrices HH^t et $H\ ^tH$. Puisque chaque valeur singulière vérifie la relation $Hv = \sigma u$ et $H^tu = \sigma v$ tel que σ est une valeur singulière, u est le vecteur propre associée à la valeur propre σ^2 pour la matrice HH^t et de même mais pour la matrice H^tH. On peut alors retrouver les éléments propres de la matrice HH^t en utilisant ceux de la matrice H^tH et vice versa.

Question 9 : Le script *puissance_iteree.m* contient les deux méthodes de puissance itérée, l'une sur AA^t et l'autre sur A^tA.

Résultat de l'exécution du script :

```
Erreur relative pour la methode avec la grande matrice = 9.866e-09
Erreur relative pour la methode avec la petite matrice = 9.871e-09
Ecart relatif entre les deux valeurs propres trouvees = 1.15e-10
Temps pour une ite avec la grande matrice = 4.774e-03
Temps pour une ite avec la petite matrice = 1.229e-04
```

Question 10 : En théorie, il est plus utile d'utiliser la méthode de la puissance itérée pour calculer les éléments propres de la matrice $\sum$ car avec chaque itération, on trouve la plus grande valeur propre et son vecteur propre associé et ainsi on s'arrête aux nombres des éléments nécessaires.

Question 11 : Les éléments propres de HH^t et H^tH sont les mêmes. On choisit donc la matrice avec la plus petite taille.