



TP-Projet 3 : Application de l'ACP : les "Eigenfaces"

Nom des auteurs

CAZES Noa

JAMES Christopher

MARTIN Cédric

Département Sciences du Numérique - Première année
2019-2020

Table des matières

1	Introduction	3
2	Exercice 1 : Analyse en Composantes Principales	3
3	Exercice 2 : Projection des images sur les <i>eigenfaces</i>	4
4	Exercice 3 : Application à la reconnaissance des visages	5
5	Conclusion	7

Table des figures

1	Set de données initial	3
2	Set de données après ACP	3
3	Après reconstitution	4
4	RMSE	4
5	Les clusters	5
6	Individu reconnu	5
7	Individu non reconnu	6
8	Visages avec couleurs	7

1 Introduction

De la première partie de ce projet, nous avons pu retenir que pour réduire les dimensions d'une matrice à l'aide de l'ACP nous avons seulement besoin des couples propres jugés essentiels, dans le sens où ils permettent de restaurer un maximum de l'information initialement contenue dans la matrice en question.

De la deuxième partie de ce projet, nous avons montré qu'en utilisant initialement la méthode de la puissance itérée avec déflation pour calculer les couples propres essentiels à une réduction de dimension, deux algorithmes se sont distingués. Leurs utilisations se départagent en fonction du type de la matrice initiale sur laquelle ils sont appliqués.

Dans cette partie, on va appliquer l'ACP à des données qui ne sont autres que des visages. On dispose de 37 individus et de 6 postures différentes pour chacun d'entre eux.

2 Exercice 1 : Analyse en Composantes Principales

La figure 1 représente le set de données initial :

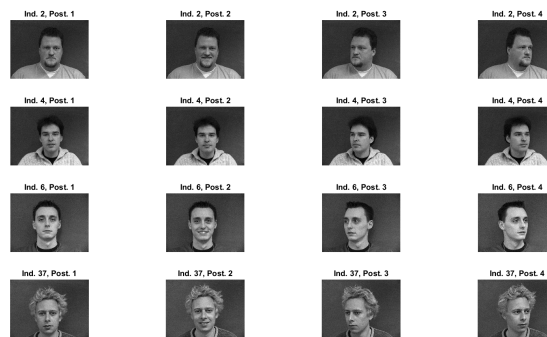


FIGURE 1 – Set de données initial

On veut diminuer les dimensions (sachant que $p = 307200$) tout en gardant un maximum d'informations. On détermine alors les axes principaux des images des données d'apprentissage à l'aide des vecteurs propres associés aux $n - 1$ valeurs propres non nulles de Σ .

On obtient alors ce set de données :

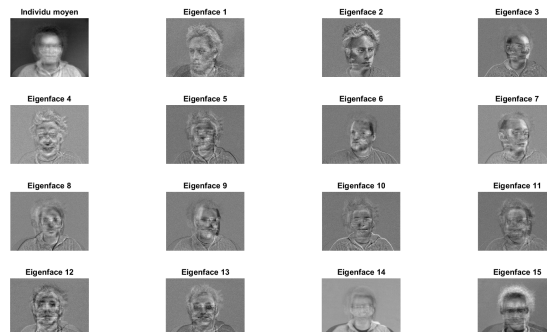


FIGURE 2 – Set de données après ACP

Sur la figure 2, le premier élément est l'individu moyen, on observe qu'une perte de l'information a bien eu lieu sur les autres images. Cependant certaines caractéristiques des individus restent observables. Elle seront certainement suffisantes pour pouvoir déterminer, à partir de ces nouvelles images, quel était l'individu initial. C'est ce que nous allons tenter de faire dans les deux autres exercices.

3 Exercice 2 : Projection des images sur les *eigenfaces*

On a ainsi obtenu les *eigenfaces* sous forme d'images. On projette alors nos données centrées sur les axes propres.

L'objectif de cet exercice est de faire varier le nombre de composantes principales pris en compte, puis de reconstituer les images avec les informations restantes (figure 3).



FIGURE 3 – Après reconstitution

On observe que plus le nombre de composantes principales pris en compte est important, plus les images sont proches des images initiales. C'est ainsi que la valeur de la *RMSE* diminue lorsque le nombre de composantes principales utilisé augmente.

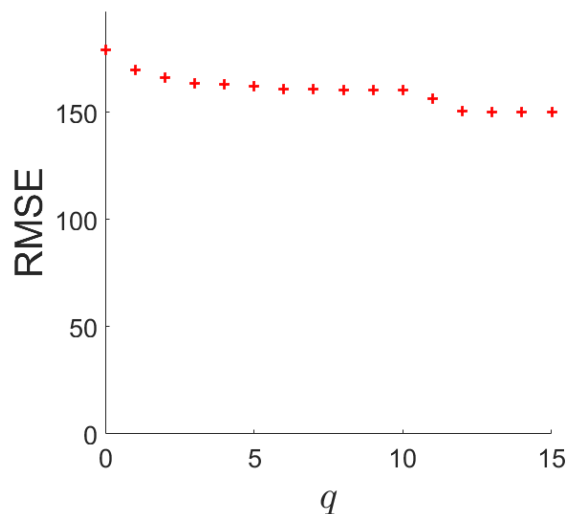


FIGURE 4 – RMSE

4 Exercice 3 : Application à la reconnaissance des visages

La figure 5 représente différents clusters réalisés sur des individus de la base d'apprentissage sous deux postures et selon leurs deux premières composantes principales.

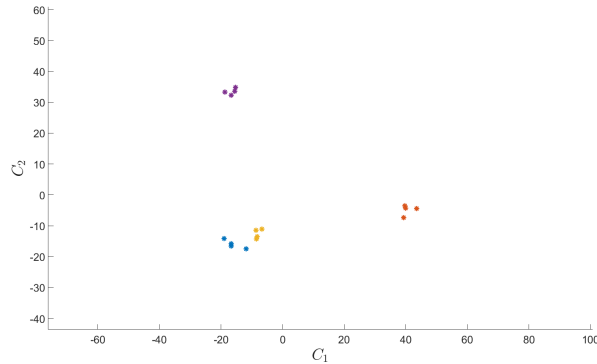


FIGURE 5 – Les clusters

C'est alors qu'il semble viable d'utiliser les *eigenfaces* pour la reconnaissance des visages. Ainsi on va conserver seulement les deux premières composantes principales d'une image appartenant à la base de tests (et pas nécessairement à la base d'apprentissage), on va déterminer de quel cluster elle est la plus proche (en fixant un seuil, avec l'algorithme des plus proches voisins...) et en déduire que cette image représente l'individu associé au cluster.

Question 4

L'algorithme des k plus proches voisins est tel qu'à partir de la base d'apprentissage, on a pu placer en clusters les individus (figure 5).

On envisage alors un individu, on le place sur le graphique de clusters grâce à ses deux composantes principales, on calcule les distances entre le point le représentant et les autres points, et on ne conserve que les k distances minimales. Les individus associés à ces k distances sont les k plus proches voisins. Au sein de cet ensemble de voisins, on compte combien d'éléments appartient à chaque cluster, et l'individu qui a le plus d'éléments appartenant au sien est associé à l'individu initial inconnu (figure 6).

Posture numero 6 de l'individu numero 2
Je reconnais l'individu numero 2



FIGURE 6 – Individu reconnu

Si l'individu initial n'était pas dans la base d'apprentissage, l'individu ne serait pas reconnu (figure 7).

Posture numero 5 de l'individu numero 3
Je ne reconnais pas cet individu !

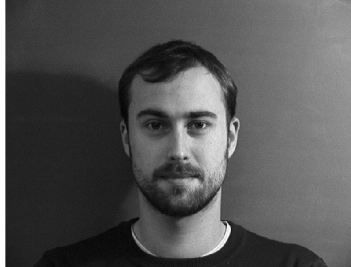


FIGURE 7 – Individu non reconnu

Question 5

On dispose de 15 composantes principales. Afin d'économiser les calculs, on décide d'utiliser seulement les N premières composantes principales, qui sont suffisantes pour avoir un taux d'erreur acceptable. Plus cette valeur est grande, plus le taux d'erreur est faible et le temps de calcul long.

De plus, si le nombre k de plus proche voisins est trop grand, la reconnaissance ne sera plus fiable car trop de point seront pris en compte, le choix de k permet donc aussi de rendre le classifieur plus optimal et exact.

Question 6

Pour le calcul des couples propres utiles, il convient d'utiliser l'algorithme de calcul des couples propres le moins couteux en calculs (le *TP2* montre qu'il n'y en a pas un d'universellement meilleur que l'autre, cela dépend du type de matrice utilisé) et de substituer les matrices de grandes dimensions par des matrices de petites dimensions, quand cela est possible (ex : usage de Σ_2 au lieu de Σ).

Question 7

A la différence de l'implémentation de Cholesky, les algorithmes de *subspace iteration* ne calculent pas l'ensemble des couples propres mais seulement ceux qui sont jugés essentiels, grâce à une condition d'arrêt basée notamment sur l'atteinte d'un certain pourcentage de la trace de la matrice initiale.

Ainsi, il vaudrait mieux utiliser les algorithmes de *subspace iteration*.

Question 8 Supplémentaire

En cas de rajout des couleurs (figure 8), la matrice à traiter serait en trois dimensions, ce qui augmenterait le temps de calcul.

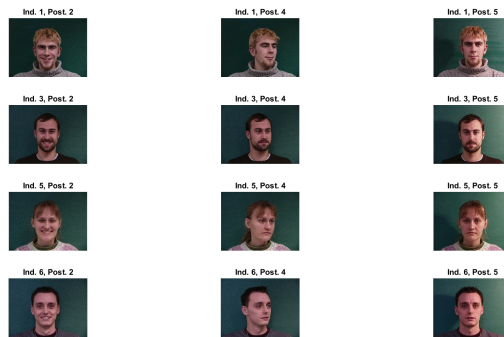


FIGURE 8 – Visages avec couleurs

De plus, après avoir fait des tests, l'usage de couleurs ne semble pas induire une meilleure reconnaissance des visages.

5 Conclusion

La reconnaissance des visages peut se faire via l'ACP associée à, par exemple, un algorithme des plus proches voisins.