Rapport du TP3 du Projet de Calcul Scientifique et Analyse de données

---

APPLICATION DE L'ACP : LES "EIGENFACES"

Auteurs :
M. LOTFI CHAIMAA
M. MESKINE HATIM
M. WISSAD MEHDI

15 mai 2020

Sommaire

Introduction

On dispose de n images de visages d'un ensemble d'individus. Chaque individu est photographié sous le même nombre de postures faciales (gauche, face, trois quart face, etc.). Chacune de ces n images en niveaux de gris est stockée dans une matrice bidimensionnelle de taille 480×640. Ces n images constituent les images d'apprentissage. En les vectorisant, nous pouvons donc représenter ces images par des vecteurs colonnes de R^p , où $p = 480 \times 640 = 307200$ est le nombre de pixels commun à toutes les images. Alors que dans le TP1, chaque pixel d'une image couleur constitue un point de R^3, ici c'est chaque image qui constitue un point d'un espace affine R^p de dimension très élevée. Ci-dessous, on a les 3 postures des trois premiers individus de notre base d'apprentissage :

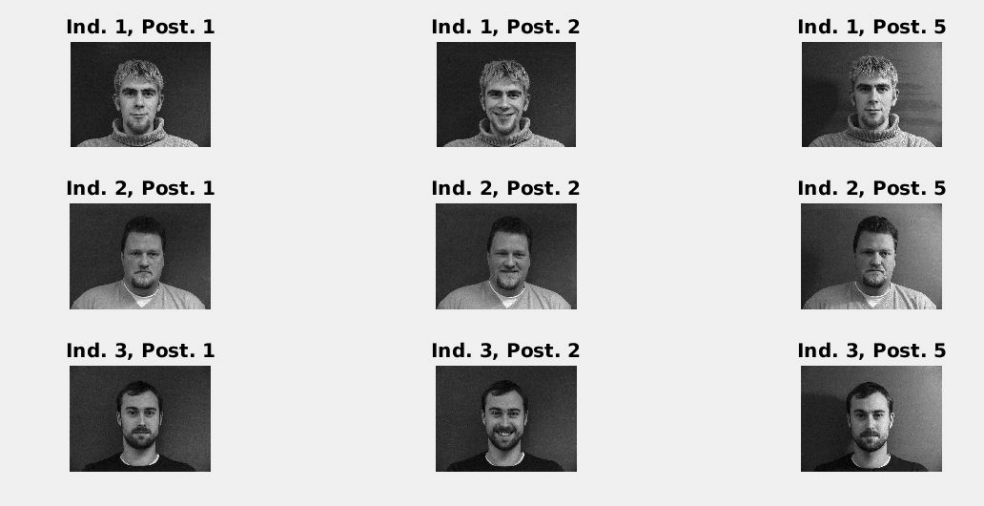

FIGURE 1 – base d'images

Exercice1 : analyse en composantes

Dans cet exercice qui vise à calculer les axes principaux des images d'apprentissage à partir des vecteurs propres associés aux n-1 valeurs propres non nulles de la matrice de variance/covariance Σ des données ,

- Calcul de l'individu moyen (c'est la moyenne X).
- Centrage des donnees : en elevant du vecteur X la moyenne de chaque l'individu .
- Etant donné la taille gigantesque de la matrice Σ : p$\times$ p (p = 307200), on commence par calculer la matrice Σ_2 de taille n x n ($n_{max} = 222$). On a montré dans le TP1 qu'il suffisait de calculer les couples propres de Σ_2 pour obtenir ceux de Σ. On a : $\Sigma_2 = \frac{1}{n} X_{centre} X_{centre}^T$
- Calcul des vecteurs/valeurs propres de la matrice Σ_2 . avec eig de matlab , on récupère alors les valeurs propres .
- Tri par ordre décroissant des valeurs propres de Σ_2 : avec la fonction sort de matlab .
- Tri des vecteurs propres de Σ_2 dans le même ordre .
- Elimination du dernier vecteur propre de Σ_2 :
- Vecteurs propres de Sigma (deduits de ceux de Σ_2) : $W = X_{centre}.V_{pt}$ avec V_{pt} sont les vecteurs propres triés .

Exercice2 : projection des images sur les eigenfaces

Pour calculer les composantes principales des données d'apprentissage C, on a utilisé la formule suivante $C = W.X_{centre}$ avec W les vecteurs propres de sigma . Nous avons crée un $X_{reconstruit}$ et calculé l'erreur quadratique moyenne à l'aide de la formule suivante :

$$RMSE = \sqrt{E((X - X^T_{reconstruit})^2)}$$

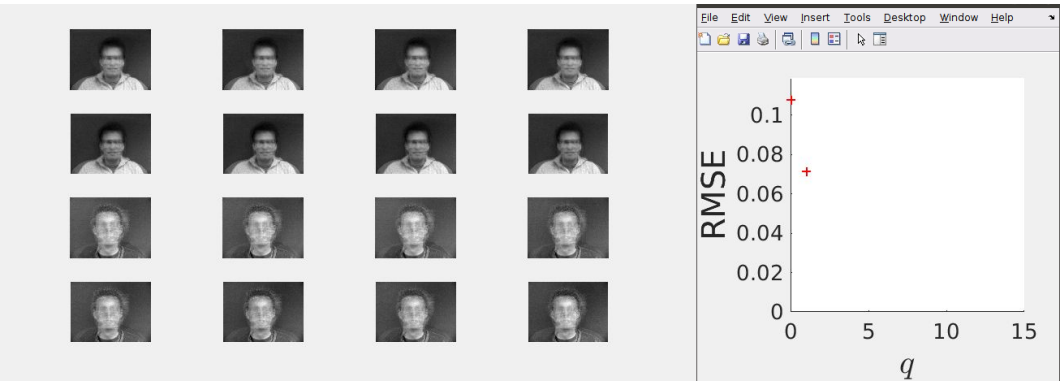

FIGURE 2 – Images originales avant l'evolution

On remarque que les images ne sont pas claires. Dans les prochaines étapes on va calculer à chaque fois une nouvelle composante principale.

FIGURE 3 – Images reconstruites après l'evolution

On remarque que les images sont devenues plus claires, grâce aux nouvelles composantes principales calculées à chaque fois, donc on remarque bien que lorsque q "nombre de composantes principales" augmente, l'erreur quadratique diminue, par conséquent les images deviennent plus claires.

Exercice3 : application à la reconnaissance faciale

Le troisième exercice est une application de l'ACP pour la reconnaissance de visages. On dispose d'une base de données qui contient 222 images (37 individus et six postures). L'objectif est de mettre en place un programme qui reconnait une image à travers une base d'apprentissage. Au début, on choist de prendre les 4 premières postures des individus 2,4,6,37 dans la base d'apprentissage (16 images).
Tout d'abord, on tire aléatoirement une image de test parmi la base de données. On calcule les N premières composantes principales des images d'apprentisssage et de l'image de test, en ayant pris 8 comme valeur de N. Ensuite, pour déterminer l'image la plus proche, on a calculé la distance euclidienne entre l'image de test et chacune des images d'apprentissages. Il suffit par la suite de comparer le minimum des distances calculées avec le seuil de reconnaissance s. En prenant s=25,on obtient les deux figures suivantes :

FIGURE 4 – Image de test reconnue

FIGURE 5 – Image de test non reconnue

On remarque que la posture n5 de l'individu a été reconnue et que l'individu 30 ne l'a pas été , ce qui était prévisible car aucune image de l'individu 30 n'était dans la base d'apprentissage.

Par la suite, on élargit les données de notre base d'apprentissage . On choisit de prendre les 3 premières postures de toutes les individus (111 images).

4) En ce qui concerne le classifieur, on choisit de configurer le classifieur 3ppv qui calcule les 3 plus proches images de l'image du test. On obtient la figure ci-dessous qui nous montre un des résultats obtenus.
On remarque que le test a été concluant. En effet, les 3 images calculées les plus proches corres-

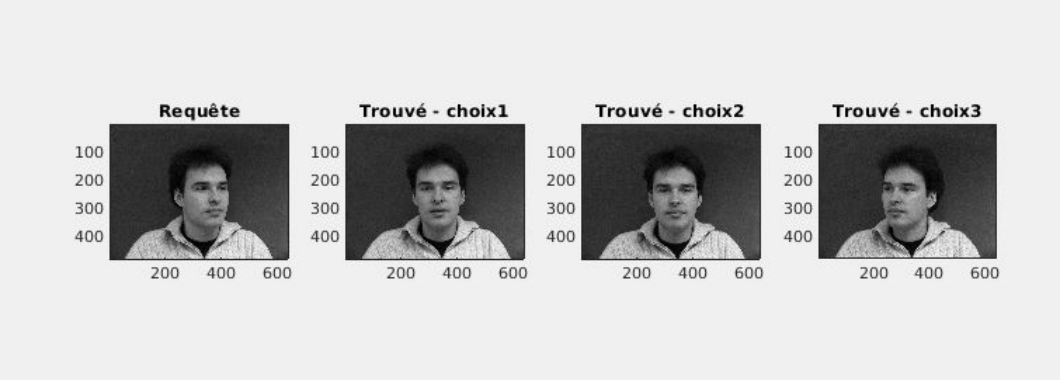

FIGURE 6 – Résultat d'une requête sur une base d'images

pondent tous à l'individu de test.

5) Pour pouvoir évaluer notre classifieur, nous avons effectué plusieurs tests et calculé la matrice de confusion. On pourra ainsi en déduire le taux d'erreur.
On rappelle que la matrice de confusion est une matrice qui mesure la qualité d'un système de classification. Dans notre cas , elle sera de taille 37×37 et son élément générique d'indice (i,j) est le nombre de tests correspondant à l'individu i qui ont été affectés aux images qui correspondant à l'individu j. Si la matrice de confusion est diagonale alors la reconnaissance est parfaite.
On choisi de teset toutes les images présentes dans la base de données. En conservant les données utilisées lors des questions précédentes , on obtient un taux d'erreur de 19%.
Il y a plusieurs facteurs qui rentrent en jeu pour améliorer notre classifieur.
On décide d'augmenter le nombre de composantes principales (la valeur de N était de 8). La nouvelle valeur de N est calculée de sorte que la taux d'information des N valeurs propres atteint les 95% . On obtient un taux d'erreur égal à 13%.

6) Compte tenu de la taille des données fournie (222 images), on peut calculer les couples propres de la matrice Σ_2 à travers la fonction eig. On rappelle que Σ_2 est de taille 111×111. On peut donc se contenter de la fonction eig plutot que de faire apple aux autres méthodes de calcul.

7) Dans le cas où on voudrait implementer notre programme pour une base de données beaucoup plus grande (en prenant par exemple 10000 individus et plusieurs postures), il serait préférable d'utiliser les algorithmes "subspace iteration" et plus précisément la méthode subspace v3 pour calculer les couples propres de la matrice de covariance. La fonction eig serait beacoup moins efficace puisqu'elle prendrait plus de temps pour calculer toutes les valeurs propres.

8) On modifie les images fournies au début en prenat en compte cette fois-ci les couleurs. On obtient la figure suivante : On remarque que parmi les 3 images retrouvées deux d'entre elles ne

FIGURE 7 – Résultat d'une requête sur une base d'images

correspondent pas à l'individu qu'on souhaite reconnaitre. On peut expliquer cela par le fait que quand utilise des images en couleurs on rajoute plus d'informations . ces dernières semblent ne pas être détéctées par l'algorithme qu'on a implémenté. Par contre, lorsqu'on diminue le seuil, on aura des images bien compatible et associée au l'image requete. De plus le taux d'erreur pour une image colorées est mieux que celui d'une image non colorées. On déduit que lorsqu'on travaille sur des requets colorées est beaucoup mieux.

Remarques concernat le code sur matlab :

— donnees.m, exercie1.m, exercice3.m (ces codes utlisient la premiere base de donnee (16 images))

— donnees_bis.m, exercice3q4, exercice3_bis, utilisent une nouvelle base d'apprentissage qui contient 111 images et qu'on utilise pour tester les performances de l'algorithme.