TP2 – *Eigenfaces*

Ce TP s'inspire d'un article intitulé *Eigenfaces for recognition*, écrit par Turk et Pentland et publié dans le *Journal of Cognitive Neuroscience* en 1991.

Description des données

Vous disposez 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, ...). Chacune de ces n images en niveaux de gris est stockée dans une matrice bidimensionnelle de même taille 480×640. En les vectorisant, vous pouvez donc représenter ces images par des vecteurs colonnes de $\mathbb{R}^p$, 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 constituait un point de $\mathbb{R}^3$, ici c'est chaque image qui constitue un point d'un espace affine de dimension p beaucoup plus élevée.

La matrice des données X, de dimension $n \times p$, contient sur chaque ligne la transposée d'une image vectorisée. Lancez le script `donnees.m` afin de créer cette matrice et de la stocker dans un fichier au format Matlab, de nom `donnees.mat` (ne recopiez pas les images sur votre compte, afin de préserver votre quota).

Différences importantes avec le TP1

- La taille de la matrice X_c des données centrées du TP1 est $n \times 3$, n étant le nombre de pixels de l'image couleur étudiée. Bien entendu, comme $n \gg 3$, $\mathrm{rg}(\mathrm{X}_c) \leqslant \min\{n, 3\} = 3$. Or, pour une image naturelle, la corrélation entre les 3 canaux RVB n'est jamais égale à 1, donc les 3 colonnes de X_c sont linéairement indépendantes. Par conséquent, $\mathrm{rg}(\mathrm{X}_c) = 3 = \min\{n, 3\}$. On dit d'une telle matrice qu'elle est de *rang maximal*. Une conséquence de ce résultat est que la matrice de variance/covariance $\Sigma = \mathrm{X}_c^\top \mathrm{X}_c / n$ est également de rang 3 (comme elle est de taille 3×3, cette matrice est inversible).

- Dans le TP2, la matrice X_c des données centrées, obtenue en retranchant à chaque ligne de X l'*individu moyen* $\overline{\mathrm{X}}$ (égal à la moyenne des colonnes de X, comme dans le TP1), a pour taille $n \times p$, où n désigne le nombre d'images et p le nombre de pixels commun à toutes ces images. Comme $p \gg n$, on en déduit que $\mathrm{rg}(\mathrm{X}_c) \leqslant \min\{n, p\} = n$. Pour que cette matrice soit de rang maximal, il faudrait que ses n lignes soient linéairement indépendantes. Or, leur somme est égale au vecteur nul, puisque $\overline{\mathrm{X}}$ a été obtenu en moyennant les n lignes de X. Pour des images naturelles, il apparaît donc que $\mathrm{rg}(\mathrm{X}_c) = n - 1 < \min\{n, p\}$. Par conséquent, la matrice de variance/covariance $\Sigma = \mathrm{X}_c^\top \mathrm{X}_c / n$ est de rang $n - 1$ (comme elle est de taille $p \times p$ et que $p \gg n$, cette matrice n'est pas inversible).

- Une autre différence avec le TP1 vient de ce que, dans le TP2, la fonction `eig` ne peut pas être directement appliquée à Σ. En effet, sa taille $p \times p$ est gigantesque ($p = 307200$). Or, pour une matrice M quelconque, $\mathrm{M}^\top \mathrm{M}$ et $\mathrm{M}\,\mathrm{M}^\top$ ont les mêmes valeurs propres non nulles. On peut donc appliquer la fonction `eig` à $\Sigma_2 = \mathrm{X}_c \, \mathrm{X}_c^\top / n$, de taille $n \times n$ beaucoup plus petite, pour calculer les valeurs propres non nulles de Σ.

- Si Y est un vecteur propre de Σ_2 associé à une valeur propre $\lambda \neq 0$, alors $(\mathrm{X}_c \, \mathrm{X}_c^\top / n) \, \mathrm{Y} = \lambda \, \mathrm{Y}$, d'où on déduit que $(\mathrm{X}_c^\top \, \mathrm{X}_c / n) \, \mathrm{X}_c^\top \, \mathrm{Y} = \lambda \, \mathrm{X}_c^\top \, \mathrm{Y}$, c'est-à-dire que $\mathrm{X}_c^\top \, \mathrm{Y}$ constitue un vecteur propre de Σ associé à λ ($\mathrm{X}_c^\top \, \mathrm{Y}$ n'est pas nul, car cela impliquerait $\lambda = 0$). Il est facile de montrer que les vecteurs $\mathrm{X}_c^\top \, \mathrm{Y}$ sont orthogonaux deux à deux. En normalisant cas vecteurs, on obtient donc une base orthonormée $\mathcal{B}$ de $\mathrm{Im}(\Sigma)$.

Exercice 1 : analyse en composantes principales

Complétez le script `exercice_1.m`, de manière à effectuer l'analyse en composantes principales (ACP) des données contenues dans la matrice X. L'objectif de cet exercice est de calculer les vecteurs propres de la matrice de variance/covariance des données, associés aux valeurs propres non nulles. Ces vecteurs propres sont appelés *eigenfaces*, par contraction des mots anglais *eigenvectors* et *faces*.

Exercice 2 : projection des images sur les *eigenfaces*

Le principal intérêt de l'ACP est de stocker la majeure partie de l'information sous une forme très condensée. Une fois connus l'individu moyen $\overline{\mathrm{X}}$ et la base $\mathcal{B}$ de $\mathrm{Im}(\Sigma)$ comportant $n-1$ *eigenfaces*, on peut calculer les coordonnées (ou *composantes principales*) des images dans ce repère. Au lieu de stocker une image de $p = 307200$ pixels, il suffit donc de stocker ses $n-1$ composantes principales, voire $q < n-1$ composantes principales correspondant aux q premières *eigenfaces* triées par ordre décroissant des valeurs propres associées, mais il faut vérifier que cette compression ne dégrade pas trop la qualité des images.

Lors du TP1, l'affichage de la matrice `D` vous a permis de savoir si les valeurs propres retournées par la fonction `eig` étaient triées par ordre croissant ou décroissant. Pour forcer cet ordre à être décroissant, il faut utiliser la fonction `sort` de Matlab, par exemple avec la syntaxe suivante :

```
[lambda,indices] = sort(diag(D),'descend');
```

où `diag(D)` est le vecteur (colonne) contenant les éléments diagonaux de `D`. Des deux paramètres de sortie `lambda` et `indices`, c'est le second qui vous permettra d'inverser l'ordre des vecteurs propres.

Complétez le script `exercice_2.m`, de manière à afficher les images reconstruites par projection sur les q premières *eigenfaces*, pour différentes valeurs de $q \in [1, n-1]$ (n'oubliez pas de leur ajouter l'individu moyen). Tracez la racine carrée de l'erreur quadratique moyenne (RMSE, pour *Root Mean Square Error*) entre les images originales et les images reconstruites, en fonction de q.

Exercice 3 : restauration d'images dégradées

Complétez le script `exercice_3.m`, de manière à restaurer des images de visages dans lesquelles un certain nombre de pixels ont été mis à 0, en l'occurrence les pixels qui se trouvent à l'intérieur d'une bande horizontale située au niveau des yeux. Calculez les $n-1$ composantes principales de cette image dégradée, afin de la restaurer (n'oubliez pas d'ajouter l'individu moyen lors de la restauration). Observez le résultat de cette restauration lorsque l'épaisseur de la bande noire croît.