Séance-Projet 1 : Application de l'ACP : les «Eigenfaces» – Méthode de la puissance

Modalités

Ce document constitue la première partie du projet de calcul scientifique et analyse de données. Ce projet s'effectue par équipe de trois étudiants issus d'un même groupe de TD, et est découpé en trois parties. Pour chacune, il y aura une séance-projet durant lequel votre enseignant vous donnera des informations pratiques et/ou théoriques pour vous aider à la réalisation des travaux en autonomie.

Une présentation de la globalité de ce projet est donné sous Moodle, dans le document "Présentation projet".

1 Les «Eigenfaces»

Ce projet 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 de personnes. Chaque personne est photographiée sous le même nombre de postures faciales (face, trois quart face, avec trois émotions). Chacune de ces n images en niveaux de gris est stockée dans une matrice bidimensionnelle de taille 300×400. Ces n images constituent les *images d'apprentissage*. En les vectorisant, vous pouvez donc représenter ces images par des vecteurs colonnes de $\mathbb{R}^p$, où $p = 300 \times 400 = 12000$ est le nombre de pixels commun à toutes les images. Alors que dans le TP1 d'Analyse de Données, chaque pixel d'une image couleur constitue un point de $\mathbb{R}^3$, ici c'est chaque image qui constitue un point d'un espace affine $\mathbb{R}^p$ de dimension très élevée.

La matrice des données X, de taille $n \times p$, contient sur chaque ligne la transposée d'une image vectorisée.

Attention, pour cette séance, seuls 4 personnes (2 femmes et 2 hommes) sur 32 (16 hommes et 16 femmes) et 4 postures sur 6 sont sélectionnées pour faire partie de la base d'apprentissage (Figure 1) ; il faudra bien entendu considérer un plus grande nombre de personnes et de postures pour les tests de performance (partie 3 du projet)

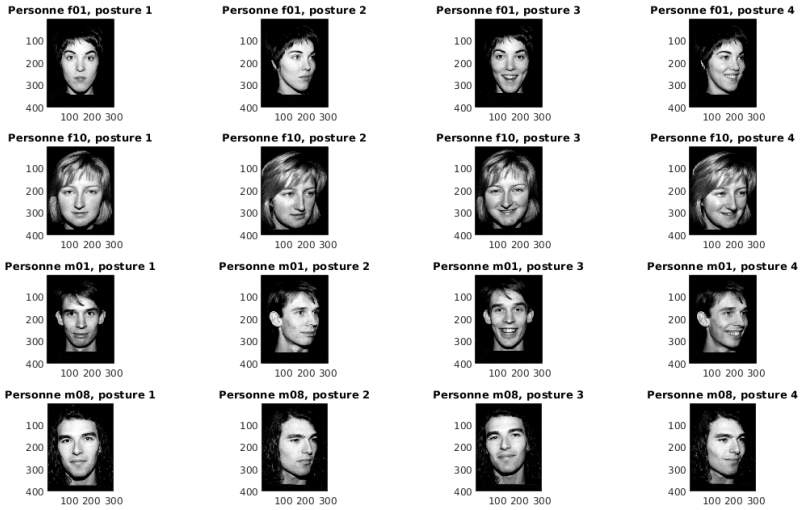

Figure 1 – Une base de visages

Exercice 1 : Analyse en Composantes Principales

Question 1 :

Complétez le script `eigenfaces.m`, 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 [1]. Ces axes principaux sont appelés *eigenfaces* (Figure 2) par Turk et Pentland, par contraction des mots anglais *eigenvectors* et *faces*.

Des explications sont données en Annexe sur le pourquoi des $n-1$ vecteurs propres.

Exercice 2 : projection des images sur les *eigenfaces*

Une fois connues les $n-1$ *eigenfaces*, on peut calculer les composantes principales.

Question 2 :

Complétez le script `projection.m`, de manière à afficher les images d'apprentissage reconstruites à l'aide des q premières *eigenfaces* et des q premières composantes principales, pour $q \in [0, n-1]$ ($q = 0$ correspond à l'individu moyen).

Attention : n'oubliez pas d'ajouter l'individu moyen pour l'affichage des images reconstruites.

Ce script doit également afficher l'évolution, en fonction de q, de la racine carrée de l'erreur quadratique moyenne (*Root Mean Square Error*, ou RMSE) entre les images originales et les images ainsi reconstruites.

1. comme ici on vous demande de calculer toutes les valeurs propres de la matrice (sauf une), le choix d'utiliser la fonction *eig* va de soi

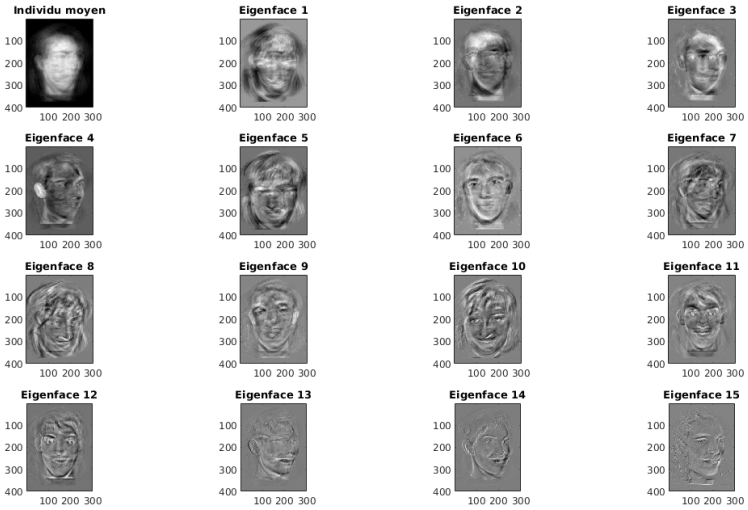

Figure 2 – Les «eigenfaces»

Exercice 3 : travail sur les visages masqués

Question 3 :

Copiez le fichier `eigenfaces.m` (resp. `projection.m`) en `eigenfaces_masque.m` (resp. `projection_masque.m`), et décommentez dans le fichier `eigenfaces_masque.m`, les lignes correspondant à la mise en place du masque.

Refaites tourner vos scripts afin de réaliser les eigenfaces sur les images masquées.

2 L'ACP et la méthode de la puissance itérée

Dans le code fourni, on récupère les éléments propres de $\mathbf{\Sigma}$ la matrice de variance/covariance par un simple appel à la fonction `eig` de Matlab, et on trie les vecteurs propres de $\mathbf{\Sigma}$ par ordre décroissant des valeurs propres associées, pour obtenir les axes principaux.

On pourrait aussi utiliser la très classique méthode de la puissance itérée avec déflation, qui renverrait les couples propres directement dans l'ordre voulu. L'algorithme de la puissance itérée pour trouver le couple propre dominant – i.e sans l'opération de déflation – est présenté ci-dessous.

Les opérations de cet algorithme consistent majoritairement en des produits matrice-vecteur. Ainsi, la taille de la matrice $\mathbf{M}$ est déterminante pour arriver rapidement à la convergence, en terme de temps de calcul.

Algorithm 1 une implémentation de l'algorithme de la puissance itérée

Données : une matrice $\mathbf{M} \in \mathbb{R}^{p \times p}$, un vecteur normé $\mathbf{x} \in \mathbb{R}^p$, une tolérance $\epsilon > 0$, it_{max} nombre max d'itérations.

Sortie : $(\lambda, \mathbf{x}) \in \mathbb{R} \times \mathbb{R}^p$ couple propre dominant de $\mathbf{M}$

Initialisation : $cv \leftarrow .FALSE.$, $i \leftarrow 0$, $\lambda \leftarrow \mathbf{x}^\top \mathbf{M}\mathbf{x}$

1. Tant que $.NOT.cv$:

$\mu \leftarrow \lambda$

$\mathbf{x} \leftarrow \mathbf{M}\mathbf{x}$

$\mathbf{x} \leftarrow \mathbf{x}/||\mathbf{x}||$

$\lambda \leftarrow \mathbf{x}^\top \mathbf{M}\mathbf{x}$

$i \leftarrow i + 1$

$cv \leftarrow \Big(\frac{|\lambda - \mu|}{|\mu|} \leq \epsilon\Big).OR.(i \geq it_{max})$

2. Retourner $(\lambda, \mathbf{x})$

Question 4 :

Soit une matrice rectangulaire $\mathbf{H} \in \mathbb{R}^{p \times n}$. Expliquer pourquoi connaître les éléments propres – i.e. les valeurs propres et les vecteurs propres – de $\mathbf{H}^\top \mathbf{H}$ permet de connaître les éléments propres de $\mathbf{H}\mathbf{H}^\top$.

Question 5 :

Le script `puissance_iteree.m` contient deux instances de la méthode de la puissance itérée : une appliquée à une matrice $\mathbf{A}^\top \mathbf{A}$, l'autre appliquée à la matrice $\mathbf{A}\mathbf{A}^\top$.

Compléter ce script Matlab, en suivant l'algorithme 1, aux endroits précisés dans le code.

La fonction `cputime` de Matlab permet de mesurer le temps d'exécution d'instructions.

Question 6 :

Lien avec l'ACP : est-il plus utile en théorie d'utiliser une fonction telle que `eig` ou la méthode de la puissance itérée pour calculer les éléments propres de $\mathbf{\Sigma}$ si le but est d'effectuer une ACP pour réduire les dimensions d'un espace ? Justifier.

Question 7 :

Si l'on choisit d'utiliser la méthode de la puissance itérée avec déflation pour calculer les éléments propres de $\mathbf{\Sigma}$, sur quelle matrice doit-on appliquer la méthode pour minimiser le temps de calcul et la mémoire utilisée ?

Annexe : Différences importantes avec le TP1 d'Analyse de Données

- Dans le TP1, la matrice des données centrées X_c est de taille $p \times 3$, où p désigne le nombre de pixels de l'image RVB. Or, le rang d'une matrice est inférieur à sa plus petite dimension. Comme $p \gg 3$, cela signifie que $\mathrm{rg}(\mathrm{X}_c) \leqslant 3$. Pour une image naturelle, le coefficient de corrélation linéaire entre les 3 canaux RVB n'est jamais exactement égal à ± 1. Cela signifie que les 3 colonnes de X_c sont linéairement indépendantes, donc que $\mathrm{rg}(\mathrm{X}_c) = 3$ (on dit que X_c est *de rang maximal*). Une conséquence de ce résultat est que la matrice de variance/covariance $\Sigma = \mathrm{X}_c^\top \mathrm{X}_c / p$ est également de rang 3. Comme elle est de taille 3×3, cette matrice est inversible.

- Pour cette séance, 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 lignes de X), est de 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 $\mathrm{rg}(\mathrm{X}_c) \leqslant n$. 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 $\overline{\mathrm{X}}$ est égal à la moyenne des n lignes de X. Pour des images naturelles, on en déduit que $\mathrm{rg}(\mathrm{X}_c) = n - 1$. La matrice de variance/covariance $\Sigma = \mathrm{X}_c^\top \mathrm{X}_c / n$ est donc elle aussi de rang $n - 1$. Comme elle est de taille $p \times p$, et que $p \gg n$, cette matrice n'est pas inversible. En l'occurrence, le noyau de Σ est de dimension $p - n + 1$.

- Une autre différence avec le TP1 vient de ce que, pour cette séance, la fonction `eig` ne peut pas être directement appliquée à Σ. En effet, sa taille $p \times p$ est gigantesque ($p = 120000$). 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 des $n - 1$ valeurs propres λ non nulles, alors par définition $(\mathrm{X}_c\,\mathrm{X}_c^\top / n)\,\mathrm{Y} = \lambda\,\mathrm{Y}$, d'où $(\mathrm{X}_c^\top\,\mathrm{X}_c / n)\,\mathrm{X}_c^\top\,\mathrm{Y} = \lambda\,\mathrm{X}_c^\top\,\mathrm{Y}$. D'autre part, $\mathrm{X}_c^\top\,\mathrm{Y}$ est un vecteur non nul : sinon, cela impliquerait que le vecteur $\lambda\,\mathrm{Y}$ est nul, ce qui est impossible puisque $\lambda \neq 0$ par hypothèse et que, étant un vecteur propre, Y ne peut pas être un vecteur nul. Par conséquent, $\mathrm{X}_c^\top\,\mathrm{Y}$ est un vecteur propre de $\Sigma = \mathrm{X}_c^\top \mathrm{X}_c / n$ associé à la valeur propre λ. Il est facile de montrer que les $n - 1$ vecteurs $\mathrm{X}_c^\top\,\mathrm{Y}$ sont orthogonaux deux à deux. En les normalisant, on obtient donc une base orthonormée $\mathcal{B}$ de $\mathrm{Im}(\Sigma)$.