TP2 – Analyse en composantes principales

Description des données

Dans ce TP, vous disposez de 90 images de visages correspondant à 15 individus photographiés sous 6 postures (vue de gauche, vue de face, etc.). Le script `donnees` permet de sélectionner `n_ind` individus et `n_pos` postures, qui constituent un *ensemble d'apprentissage* EA comportant $n \leq 90$ images. Ces images en niveaux de gris sont toutes de même taille 480×640. Par vectorisation, il est possible de les stocker dans la *matrice des données* $\mathbf{X}$, de taille $p \times n$, où $p = 480 \times 640 = 307200$ désigne le nombre de pixels. Chaque colonne de $\mathbf{X}$ contient donc une des n images de EA. Lancez le script `donnees`, qui affiche les images de EA, crée la matrice $\mathbf{X}$ et stocke l'ensemble des variables du script dans un fichier au format Matlab, de nom `donnees.mat`. Attention : ne recopiez pas les images sur votre compte, afin de préserver votre quota !

Chaque image de visage peut être vue comme un point dans un espace affine $\mathbb{R}^p$ de très grande dimension. Or, les images de visages (pas uniquement celles de EA) présentent toutes de fortes similarités, ce qui se traduit par un nuage de points dans $\mathbb{R}^p$ dont la forme n'est pas du tout quelconque. Au contraire, ces points se situent au voisinage d'un sous-espace affine de $\mathbb{R}^p$ de très faible dimension. Les n images de EA permettent justement de caractériser un tel sous-espace de dimension $n-1$, défini par la moyenne $\overline{\mathbf{X}}$ des images de EA par une base orthonormée comportant $n-1$ vecteurs de $\mathbb{R}^p$.

Un outil classique permettant de trouver une telle base orthonormée, déjà vu en 1A, est l'*analyse en composantes principales* (ACP), qui nécessite de calculer les valeurs et vecteurs propres de la *matrice de variance/covariance* des données, définie par $\mathbf{\Sigma} = \mathbf{X}_c\mathbf{X}_c^\top/n$, où $\mathbf{X}_c$ désigne la *matrice des données centrées*, obtenue en soustrayant à chaque colonne de $\mathbf{X}$ l'image moyenne $\overline{\mathbf{X}}$.

Analyse en composantes principales

Le *rang* d'une matrice est inférieur à la plus petite de ses dimensions. La matrice $\mathbf{X}_c$ des données centrées est de taille $p \times n$. Comme $n \ll p$, on en déduit que $\mathrm{rg}(\mathbf{X}_c) \leqslant n$. Pour que cette matrice soit de rang maximal, il faudrait que ses n colonnes soient linéairement indépendantes. Or, leur somme est égale au vecteur nul de $\mathbb{R}^p$, puisque $\overline{\mathbf{X}}$ est égal à la moyenne des n colonnes de $\mathbf{X}$. On en déduit que $\mathrm{rg}(\mathbf{X}_c) = n-1$.

D'après les règles sur le rang, la matrice de variance/covariance $\mathbf{\Sigma} = \mathbf{X}_c\mathbf{X}_c^\top/n$, de taille $p \times p$, est elle aussi de rang $n-1$. D'après le théorème du rang :

$$\dim(\mathrm{Ker}(\mathbf{\Sigma})) + \dim(\mathrm{Im}(\mathbf{\Sigma})) = p \quad \Rightarrow \quad \dim(\mathrm{Ker}(\mathbf{\Sigma})) = p - \mathrm{rg}(\mathbf{\Sigma}) = p - (n-1) \tag{1}$$

ce qui signifie que, parmi les p valeurs propres de $\mathbf{\Sigma}$, seules $n-1$ sont non nulles.

Il est impossible d'appliquer la fonction `eig` directement à $\mathbf{\Sigma}$ pour calculer ses valeurs et vecteurs propres, à cause de la taille gigantesque de cette matrice (307200×307200), mais on montre que, pour une matrice $\mathbf{M}$ de taille quelconque, $\mathbf{M}\,\mathbf{M}^\top$ et $\mathbf{M}^\top\mathbf{M}$ ont les mêmes valeurs propres *non nulles*. On peut donc appliquer `eig` à $\mathbf{\Sigma}_2 = \mathbf{X}_c^\top\,\mathbf{X}_c/n$, dont la taille $n \times n$ est très inférieure à celle de $\mathbf{\Sigma}$. La matrice $\mathbf{\Sigma}_2$ étant symétrique réelle, nous savons d'après le théorème spectral qu'elle admet une base orthonormée de vecteurs propres. Si $\mathbf{Y}$ est un vecteur de cette base associé à l'une des $n-1$ valeurs propres λ non nulles de $\mathbf{\Sigma}_2$, alors par définition :

$$(\mathbf{X}_c^\top\,\mathbf{X}_c/n)\,\mathbf{Y} = \lambda\,\mathbf{Y} \quad \Leftrightarrow \quad (\mathbf{X}_c^\top/n)\,\mathbf{X}_c\,\mathbf{Y} = \lambda\,\mathbf{Y} \tag{2}$$

De (2), on déduit que $\mathbf{X}_c\,\mathbf{Y}$ est un vecteur non nul de $\mathbb{R}^p$. En effet, cela impliquerait que $\lambda\,\mathbf{Y} = \mathbf{0}_n$, ce qui est impossible puisque $\mathbf{Y}$ est un vecteur propre, donc non nul, et que $\lambda \neq 0$ par hypothèse. De (2), il vient :

$$(\mathbf{X}_c\,\mathbf{X}_c^\top/n)\,\mathbf{X}_c\,\mathbf{Y} = \lambda\,\mathbf{X}_c\,\mathbf{Y} \tag{3}$$

Cette égalité montre que $\mathbf{X}_c\,\mathbf{Y}$ est un vecteur propre de $\mathbf{\Sigma} = \mathbf{X}_c\mathbf{X}_c^\top/n$ associé à la valeur propre λ. Il est facile de montrer que les $n-1$ vecteurs $\mathbf{X}_c\,\mathbf{Y}$ ainsi obtenus constituent, après normalisation, une base orthonormée de $\mathrm{Im}(\mathbf{\Sigma})$. Ces vecteurs de $\mathbb{R}^p$ sont appelés *eigenfaces*, par contraction des mots *eigenvectors* et *faces*.

Exercice 1 : calcul des *eigenfaces*

Le script `exercice_1` vise à calculer les valeurs et vecteurs propres de $\mathbf{\Sigma}$ selon le procédé décrit ci-dessus. Écrivez la fonction `eigenfaces`, appelée par ce script, qui doit retourner une base orthonormée de $\text{Im}(\mathbf{\Sigma})$ constituée de vecteurs propres de $\mathbf{\Sigma}$, stockée dans une matrice `W` de taille $p \times (n-1)$. Attention : n'oubliez pas de normaliser les *eigenfaces* !

Choix d'un nombre q de composantes principales

Les n points de $\mathbb{R}^p$ correspondant aux n images de EA appartiennent à un sous-espace affine de $\mathbb{R}^p$ de dimension $n-1$, dont un repère orthonormé est constitué de l'image moyenne $\overline{\mathbf{X}}$ et des $n-1$ *eigenfaces*. Les coordonnées des n images de EA dans ce repère sont appelées les *composantes principales*. Comme la fonction `eig` trie les *eigenfaces* par ordre décroissant des valeurs propres de $\mathbf{\Sigma}$, il est possible de ne conserver que les $q < n-1$ premières composantes principales : une image peut alors être réduite à un vecteur de $\mathbb{R}^q$.

Lancez le script `reconstruction`, qui utilise les q premières composantes principales, et affiche pour chaque valeur de $q \in \{0, \ldots, n-1\}$:

- Les images de EA reconstruites, qui se rapprochent de plus en plus des images originales lorsque q croît.
- L'évolution de la racine carrée de l'erreur quadratique moyenne (RMSE, pour *root mean square error*) entre les images reconstruites et les images originales, en fonction de q, qui décroît lorsque q croît, jusqu'à devenir nulle lorsque $q = n-1$.

Vous constatez que l'utilisation des $q = 3$ premières composantes principales suffit à reconnaître les individus sur les images reconstruites. Qui plus est, il est possible de visualiser un nuage de points de $\mathbb{R}^3$. Pour ces deux raisons, c'est la valeur $q = 3$ qui est retenue.

Exercice 2 : application à la reconnaissance faciale

Lancez le script `clusters`, qui affiche le nuage de points de $\mathbb{R}^3$ correspondant aux n images de EA, en utilisant pour chaque individu une couleur différente. Les différentes images correspondant à un même individu semblent constituer des *clusters*. Il est donc tentant de réaliser ainsi un système de reconnaissance faciale.

Complétez le script `exercice_2`, qui tire aléatoirement une image de visage parmi la totalité des 90 images (et non pas parmi les seules n images de EA), à l'aide de la fonction `randi` de Matlab, et affiche cette image, de manière à reconnaître l'individu à l'aide de ses trois premières composantes principales, en calculant la distance du point de $\mathbb{R}^3$ ainsi obtenu à chacun des centroïdes des clusters de EA. Si la distance minimale à l'un de ces clusters est inférieure à un seuil, alors il s'agit d'un des individu de EA. Sinon, il s'agit d'un individu « inconnu ».