Rapport du TP1 projet Calcul Scientifique et Analyse de données

Thierry Xu
Tom Bonetto
Mickaël Song

Département Sciences du Numérique - Première année
2021-2022

Table des matières

Table des figures

1 Introduction

Avec la pandémie de la Covid-19, nous avons appris à porter des masques quotidiennement. Désormais la moitié de notre visage est cachée nous laissant le regard pour seul moyen d'expression visible. Cette opération de cacher le visage s'apparente à un domaine en traitement d'images et de vidéos appelé "inpainting". En effet, les pixels correspondant à la zone du masque (modélisé par un rectangle) sont mis à 0. Et les eigenfaces permettent, entre autres, de restaurer la zone dégradée.

Nous disposons 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, on peut donc représenter ces images par des vecteurs colonnes de $\mathbb{R}^p$, où $p = 300400 = 12000$ 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×p, contient sur chaque ligne la transposée d'une image vectorisée.

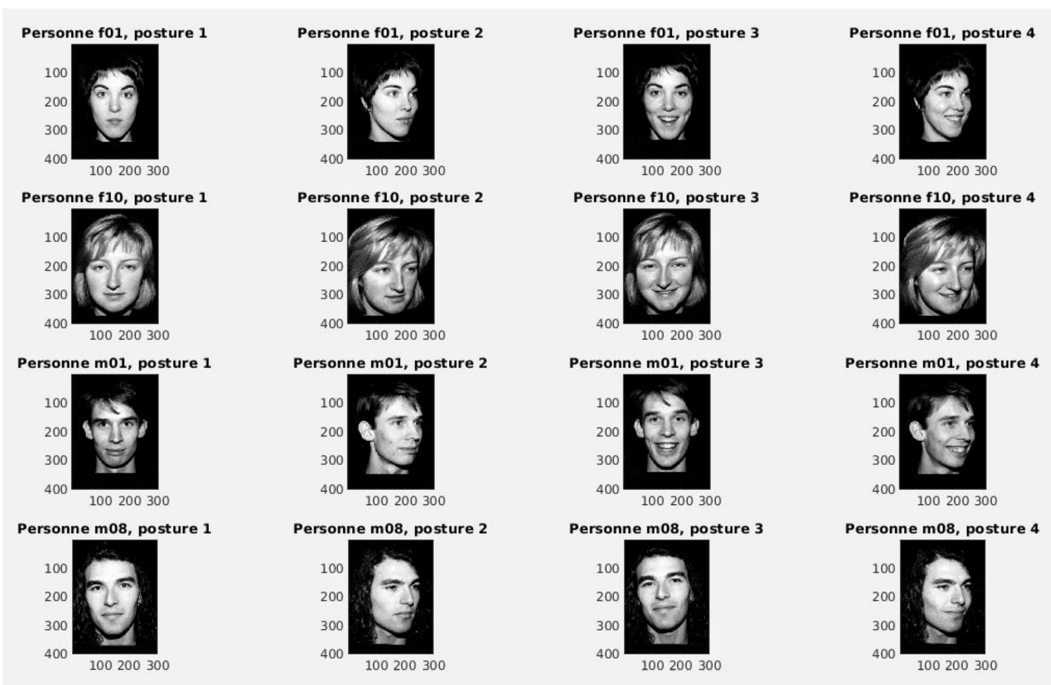

FIGURE 1 – Une base de visages

2 Les «Eigenfaces»

2.1 Analyse en Composantes Principales

Cette partie consiste à calculer et afficher les eigenfaces sans masque. Les Eigenfaces correspondent aux axes principaux des images d'apprentissage. Ce calcul se fera grâce aux vecteurs propres associés aux $n-1$ valeurs propres non nulles de la matrice de variance/covariance Σ des données.

- Calcul de l'individu moyen : moyenne de X.
- Centrage des données : on retranche à X l'individu moyen.
- Calcul de la matrice resultant du calcul commuté Σ_2. On passe par Σ_2 et pas directement par Σ à cause de sa taille giganteste $p \times p$ ($p = 120000$), cela ne pose pas de problème puisque Σ et Σ_2 ont les mêmes valeurs propres non nulles. On peut donc appliquer la fonction eig à $\Sigma_2 = X_c^T X_c / n$, de taille $n \times n$ beaucoup plus petite.
- On calcul les vecteurs/valeurs propres de Σ_2 : avec la fonction eig de matlab.
- On en déduit les vecteurs propres de Σ avec ceux de Σ_2 : $W = X_c^T W_2$.
- On trie par ordre decroissant les valeurs propres : fonction sort avec l'argument 'descend'.
- On trie les eigenfaces (vecteurs propres) dans le même ordre et on enlève la dernière eigenface qui apparitant au noyau de Σ.

On obtient le résultat de la figure 2.

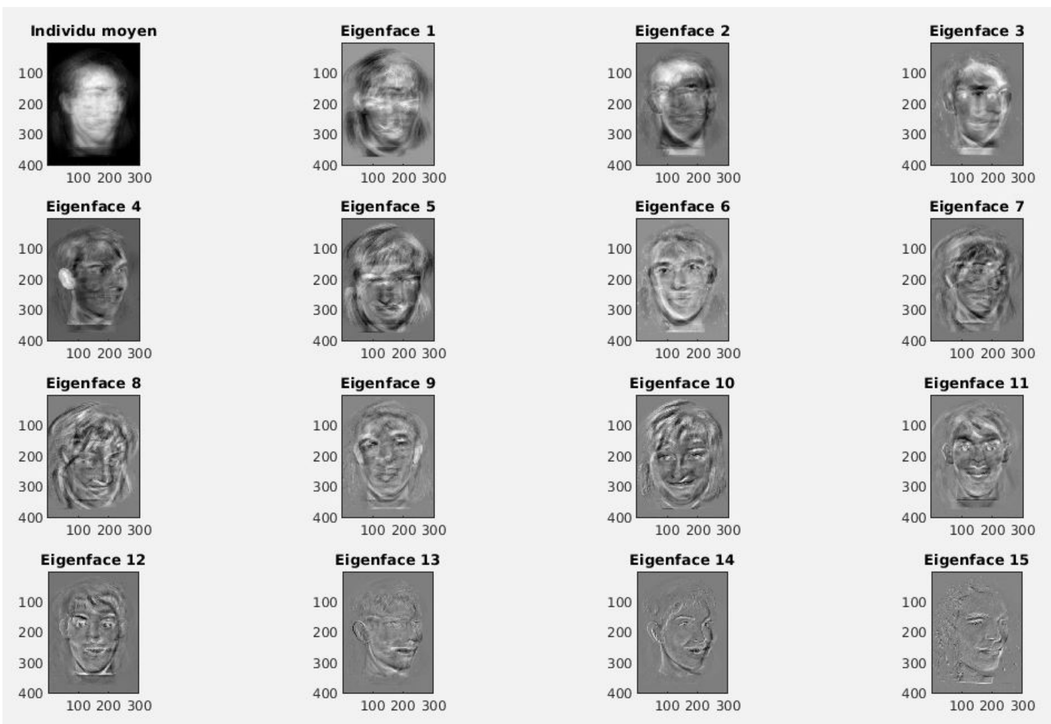

Figure 2 – Individu moyen et les Eigenfaces

2.2 Projection des images sur les eigenfaces

Cette partie a pour but d'afficher les images d'apprentissage reconstruites à l'aide des q premières eigenfaces et des q premières composantes principales, pour $q \in [0, n-1]$.
On affiche également l'évolution, en fonction de q, de la racine carrée de l'erreur quadratique moyenne (RMSE) entre les images originales et les images ainsi reconstruites.

On reconstruit les images d'apprentissage : $X_{reconstruit} = W_q C_q^T + \bar{X_n}$

Avec W_q les q premières eigenfaces et C_q les q premières composantes principales et $\bar{X_n}$ l'individu moyen. Voir figure 3

FIGURE 3 – Images d'apprentissage reconstruites

$RMSE = \sqrt{E(X - X^T_{reconstruit})^2}$, voir figure 4.

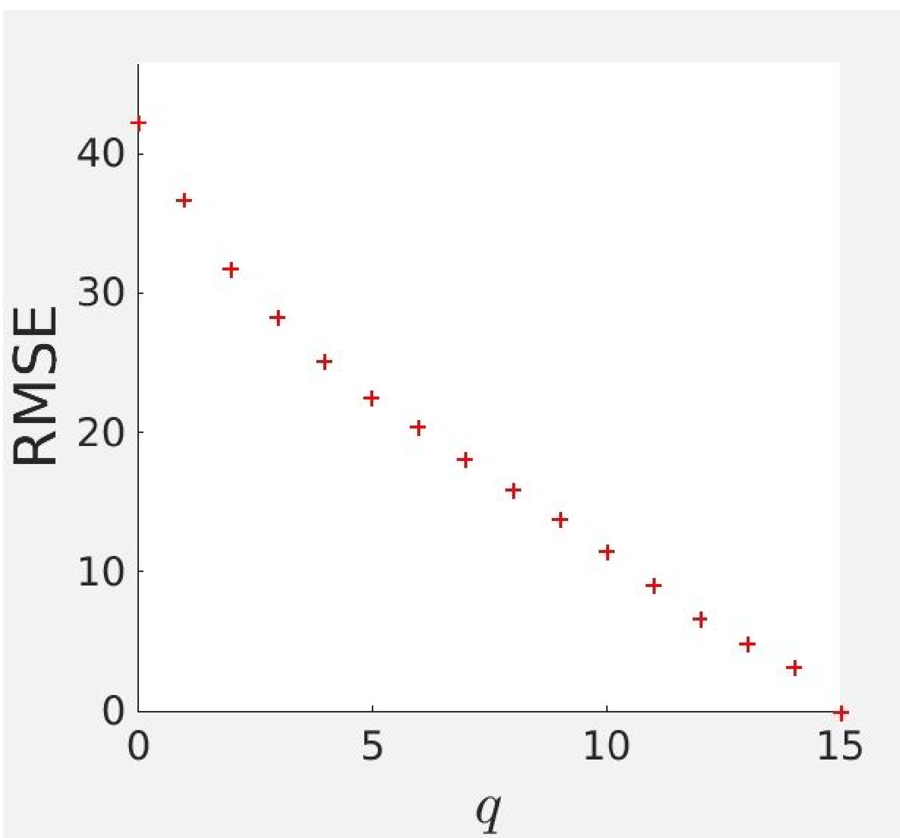

FIGURE 4 – RMSE en fonction des q composantes principales

2.3 Travail sur les visages masqués

On réeffectue le même travail mais cette fois-ci pour la base de visages masqués voir figure 5, on obtient les eigenfaces figure 6, la projection figure 7 et l'évolution de la RMSE en fonction des q composantes principales figure 8.

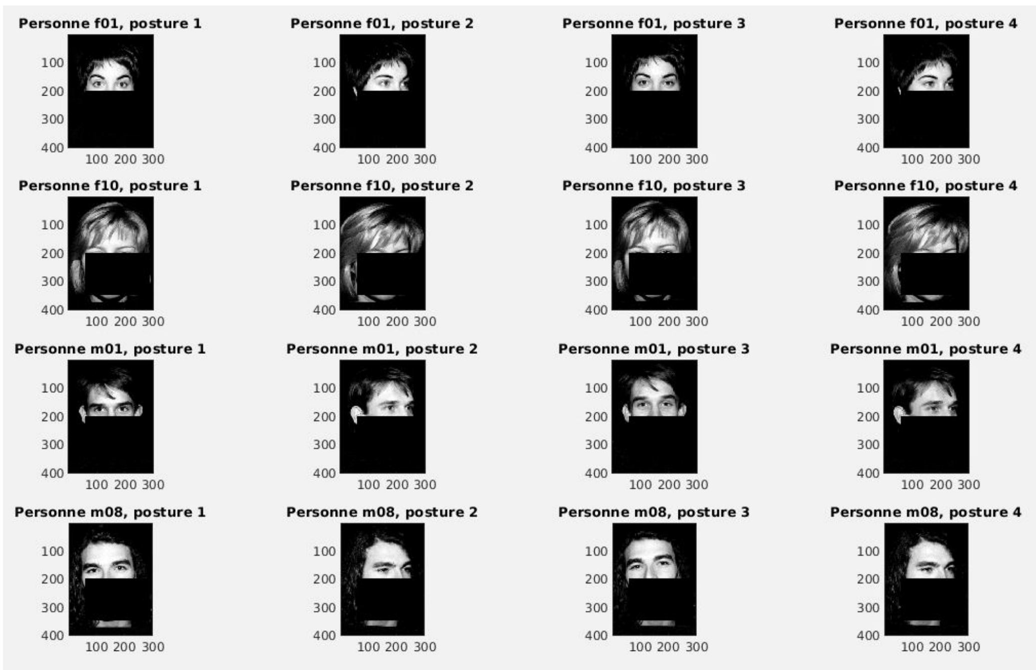

FIGURE 5 – Une base de visages avec masque

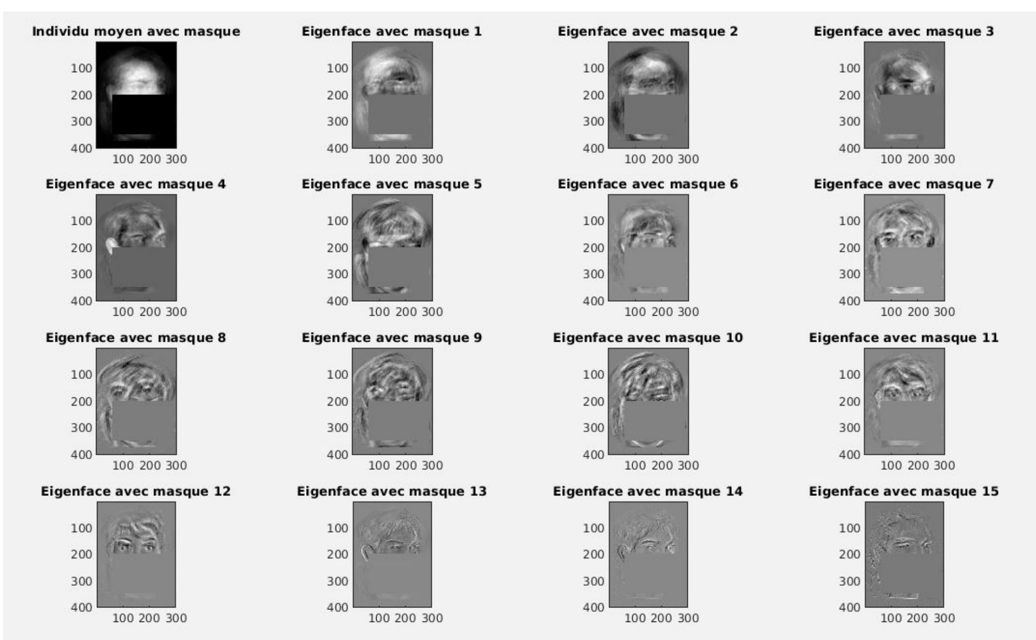

FIGURE 6 – Individu moyen et les Eigenfaces avec masque

Figure 7 – Images d'apprentissage avec masque reconstruites

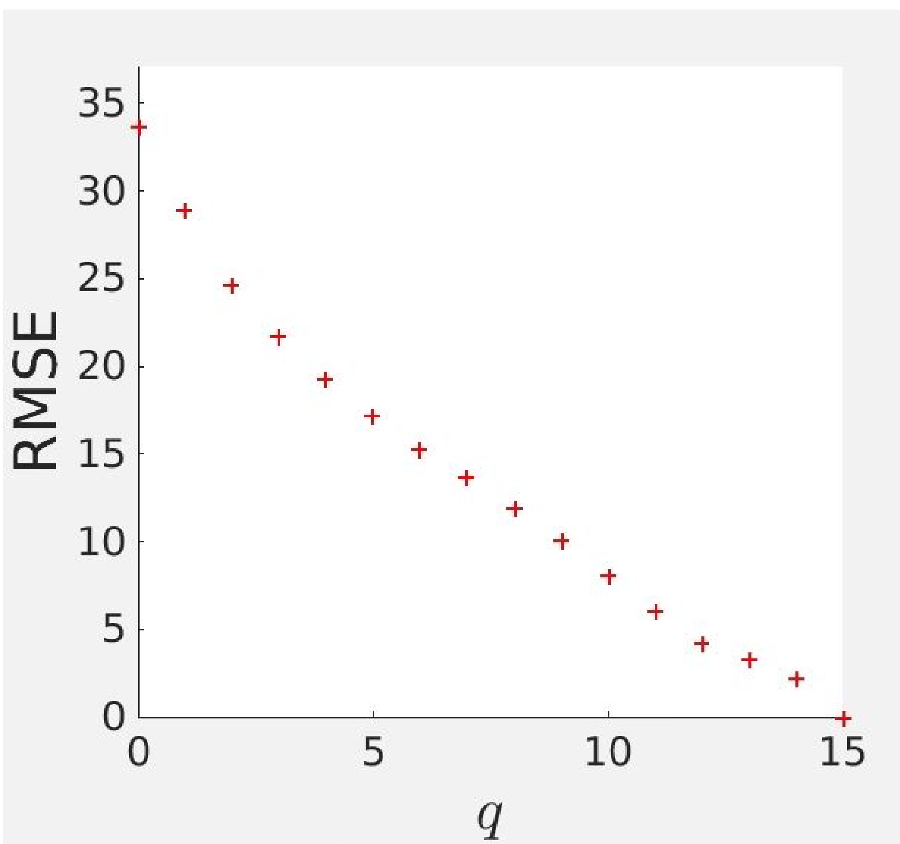

Figure 8 – RMSE en fonction des q composantes principales

3 L'ACP et la méthode de la puissance itérée

Dans cette partie on s'intérresse au lien entre l'ACP et l'algorithme de la puissance itérée, algorithme qui permet de trouver le couple propre dominant d'une matrice M.

Q4 - Tout d'abord on a : $H^T H \in M_n(R)$.

Considérons λ une valeur propre de $H^T H$ non nulle et x $\in R^n$ un vecteur propre associé à cette valeur propre. On a donc :

$$H^T H x = \lambda x$$

Puis en multipliant par H, puisque le produit matriciel est associatif :

$$(HH^T)Hx = \lambda Hx$$

D'autre part, Hx est un vecteur non nul : sinon, cela impliquerait que le vecteur λx est nul, ce qui est impossible puisque $\lambda \neq 0$ par hypothèse et que x est non nul car c'est un vecteur propre. Ainsi, Hx est un vecteur propre de HH^T associé à la valeur propre λ.

Q5 - On obtient :

Erreur pour la methode avec la grande matrice = 9.871e-09
Erreur pour la methode avec la petite matrice = 9.859e-09
Ecart relatif entre les deux valeurs propres trouvées = 7.25e-10
Temps pour une ite avec la grande matrice = 6.206e-03
Temps pour une ite avec la petite matrice = 1.884e-04
Valeur propre dominante (methode avec la grande matrice) = 9.228e+04
Valeur propre dominante (methode avec la petite matrice) = 9.228e+04
Valeur propre dominante (fonction eig) = 9.228e+04

Q6 - Il est plus efficace d'utiliser la méthode de la puissance itérée pour calculer les éléments propres de Σ car sa compléxité est en O(n) (dû au produit matriciel) alors que celui de la fonction eig de Matlab est en O(n^2).

Q7 - D'après la question 5, on trouve que le temps d'itération de $H^T H$ (la petite matrice) est inférieur à celui de HH^T (la grande matrice). De plus, la matrice $H^T H$ utilise moins de mémoire car elle est de taille plus petite. On appliquera donc la méthode de la puissance itérée avec déflation sur la matrice $H^T H$.