

# **Laboratoires d'Évaluation d'algorithmes pour l'identification de visages statiques**

**SYS828 – Systèmes biométriques**

Responsable et enseignant: Eric GRANGER

Auxiliaire de laboratoire: George Ekladios

Session: A2019

## Laboratoire 1 – Extraction des caractéristiques

L’objectif de l’extraction et de la sélection de caractéristiques est d’identifier les caractéristiques importantes pour la discrimination entre classes. Après avoir choisi le meilleur ensemble de caractéristiques, il s’agit de réduire la dimensionnalité de l’ensemble des caractéristiques en trouvant un nouvel ensemble, plus petit que l’ensemble original, qui néanmoins, contient la plupart de l’information.

Contrairement à ce que l’intuition peut laisser penser, la performance des classificateurs n’augmente pas indéfiniment avec la taille du vecteur de caractéristiques. Après avoir augmenté en fonction du nombre de caractéristiques, l’efficacité atteint un plateau et quelquefois se met à décroître au-delà d’un certain nombre de caractéristiques. Ce phénomène est appelé *malédiction de la dimensionnalité*. En plus, le coût (temps + mémoire) de la classification augmente avec la taille de ce vecteur. On a toujours intérêt à réduire le nombre de caractéristiques pour qu’il corresponde à la dimensionnalité intrinsèque du problème ou pour de simples raisons pratiques.

Les méthodes d’extraction de caractéristiques à partir de l’intensité d’une image du visage peuvent être divisées en trois catégories : les méthodes globales (exploitant toute la région visage), les méthodes locales (exploitant des caractéristiques locales: yeux, nez, etc.), et les méthodes hybrides. On se limite principalement aux méthodes globales où l’image du visage au complet peut être vue comme un vecteur dans un espace ayant autant de dimensions que de pixels dans l’image, et représentant la variation d’intensité (en niveau de gris).

L’image du visage, une matrice de pixel  $\mathbf{M}$ , peut être représentée, par concaténation de colonnes, comme un vecteur  $\mathbf{x}$  de dimension  $I = \text{largeur} \times \text{hauteur}$ . Chaque visage peut donc être considéré comme un vecteur à  $I$  composantes dans un espace  $\mathbb{R}^I$ . L’ensemble des visages forme alors la matrice d’images  $\mathbf{X}$ , où chaque vecteur colonne  $\mathbf{x}_j$  représente un visage dont chaque élément  $x_{ij}$  est l’intensité du pixel  $i$ .

Dans notre cas, la dimension ( $I = 92 \times 112 = 10304$  pixels) du vecteur de caractéristiques est très grande, mais la vraie dimension de l’espace dans laquelle les visages observés sont situés peut être plus petite. En plus, le nombre d’échantillons réservés à l’apprentissage est considérablement inférieur au nombre de caractéristiques.

Parmi les techniques de réduction de dimensionnalité, les analyses statistiques linéaires consistent à transformer les données originales par une transformation linéaire (changement de base) optimisant un critère géométrique ou statistique. Pour  $J$  images, on cherche une matrice de transformation  $\mathbf{W}$  pour obtenir la projection  $\mathbb{R}^I \rightarrow \mathbb{R}^K$  de la matrice d’images  $\mathbf{X}$  :

$$\mathbf{Y}_{K \times J} = \mathbf{W}_{I \times K}^T \mathbf{X}_{I \times J}, \quad (1)$$

où  $\mathbf{Y}$  est la matrice résultante de la projection telle que  $K < I$ .

### 1.1 Analyse en composantes principales (Eigenfaces)

L’analyse en composantes principales (PCA pour “Principal Component Analysis”)<sup>1</sup> est une méthode classique de réduction de dimension. Cette méthode permet de représenter le nuage des points originel dans un sous-espace orthogonale de dimension inférieure, tout en limitant au maximum la perte d’information. Les axes principaux sont ceux qui contiennent le plus d’information

---

<sup>1</sup>Également connue sous le nom de transformée de Karhunen-Loève ou de transformée de Hotelling.

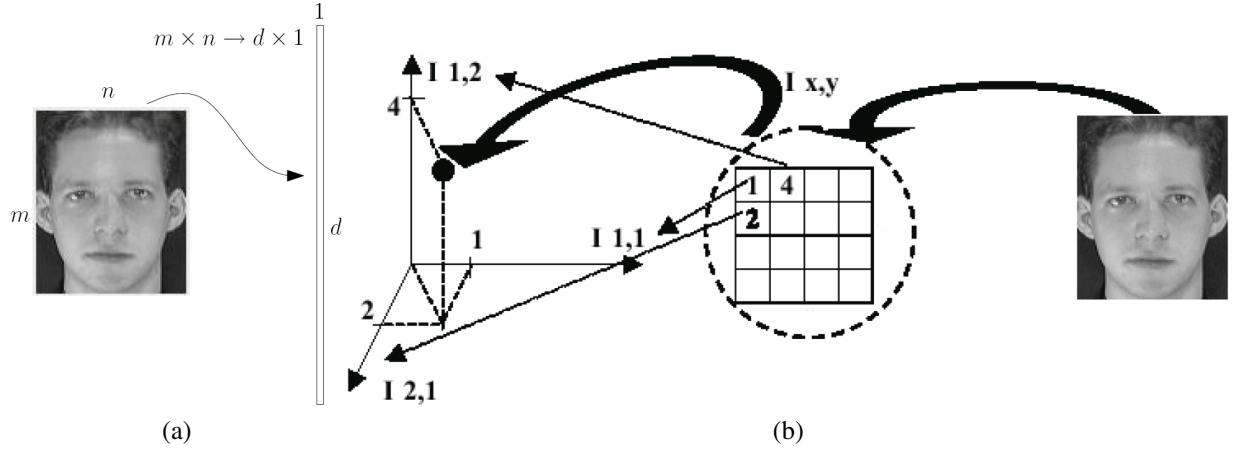


Figure 1: Représentation d'une image de visage. (a) Un vecteur image formé par concaténation de colonnes. (b) Illustration d'un visage représenté par un point dans l'espace de visages de  $K$  dimensions (en supposant  $K = 3$ )

pertinente c'est-à-dire qui maximisent l'inertie<sup>2</sup> du nuage projeté. Les composantes principales sont les vecteurs ayant pour coordonnées celles des projections orthogonales des points du nuage sur les axes principaux. La PCA fournit donc une base de représentation dans laquelle les composantes sont simplement dé-corrélées et rendant compte de la variation maximale, ce qui permet :

- une réduction des données de  $I$  à  $K$  nouvelles caractéristiques
- une visualisation des données à 2 ou 3 dimensions (si  $K = 2$  ou 3)
- une interprétation des données

Soit  $J$  images en format de vecteur colonne  $\mathbf{x}_j$  ( $j = 1, \dots, J$ ) dans  $\mathbb{R}^I$ , l'objectif est de transformer l'espace initial  $\mathbb{R}^I$  des images à un espace visage de dimensions  $\mathbb{R}^K$  ( $K < I$ ) de manière à ce que la variance observée dans chacune de ces nouvelles dimensions soit décroissante. Le calcul des composantes principales consiste à :

A) Calculer la valeur moyenne des visages (visage moyen):

$$\mu = \frac{1}{J} \sum_{j=1}^J \mathbf{x}_j \quad (2)$$

B) Centrer les images de manière à avoir une matrice de données  $\mathbf{D}$  dont la moyenne est de zéro :

$$\mathbf{D} = [\mathbf{x}_1 - \mu, \mathbf{x}_2 - \mu, \dots, \mathbf{x}_J - \mu] \quad (3)$$

<sup>2</sup>La moyenne pondérée des carrés des distances des points projetés à leur centre de gravité.

C) Calculer la matrice de covariance de la matrice centrée  $\mathbf{D}$ :

$$\mathbf{C} = \sum_{j=1}^J (\mathbf{x}_j - \mu)(\mathbf{x}_j - \mu)^T = \mathbf{D}\mathbf{D}^T \quad (4)$$

D) Décomposer  $\mathbf{C}$  en vecteurs propres  $\mathbf{v}_i$  et leurs valeurs propres  $\lambda_i$ :

$$S\mathbf{v}_i = \lambda_i \mathbf{v}_i, \quad j = 1, \dots, I \quad (5)$$

où,  $\Lambda = \text{diag}(\lambda_k)$  et  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_I]$  est la matrice de tous les  $I$  vecteurs propres. Cette opération revient à poser  $S = \mathbf{V}\Lambda\mathbf{V}^T$ .

E) Trier les valeurs propres et les vecteurs propres correspondants:

$$\begin{aligned} \lambda_1 &\geq \lambda_2 \geq \dots \geq \lambda_I \geq 0 \\ \mathbf{v}_1 &\geq \mathbf{v}_2 \geq \dots \geq \mathbf{v}_I \geq 0 \end{aligned} \quad (6)$$

F) Choisir les  $K$  premiers vecteurs propres (composantes principales):

$$\mathbf{W} = [\mathbf{v}_1, \dots, \mathbf{v}_K] \quad (7)$$

G) Finalement, projeter les données sur les  $K < I$  nouveaux axes:

$$\mathbf{Y} = \mathbf{W}^T(\mathbf{X} - \mu), \quad \mathbf{X} \in \mathbb{R}^I, \quad \mathbf{Y} \in \mathbb{R}^K \quad (8)$$

Les vecteurs propres d'un visage (Eigenfaces) associées aux plus fortes valeurs propres sont les plus pertinentes et représentent dans l'espace des images les fortes variations d'un visage. Le nombre possible de visages propres peut être donc approximé en utilisant seulement les meilleurs visages propres qui correspondent aux plus grandes valeurs propres.

## 1.2 Analyse discriminante linéaire

L'analyse discriminante linéaire (LDA pour "Linear Discriminant Analysis") est différente de PCA car il considère les étiquettes pour construire la matrice de transformation  $\mathbf{W}$ , qui devient intrinsèquement discriminative. L'objectif de LDA est de trouver la transformation qui maximise la dispersion inter-classes (Eqn. 9) et minimise au même temps la dispersion intra-classe (Eqn. 10): le critère de Fisher:

$$S_e = \sum_{i=1}^c \frac{J_i}{J} (\mu_i - \mu)(\mu_i - \mu)^T \quad (9)$$

$$S_a = \sum_{i=1}^c \frac{J_i}{J} \sum_{k=1}^{J_i} (\mathbf{x}_k - \mu_i)(\mathbf{x}_k - \mu_i)^T \quad (10)$$

où  $c$  est le nombre de classes (différents individus),  $J_i$  est le nombre d'images de la classe  $i$ ,  $\mu_i$  est la moyenne des visages de la classe  $i$ , et  $\frac{J_i}{J}$  est la probabilité a priori de la classe  $i$ . La projection

optimale peut être trouvée par la recherche des  $K$  vecteurs propres qui correspondent aux plus grandes valeurs propres de

$$\mathbf{S}_e \mathbf{W} = \mathbf{S}_a \mathbf{W} \quad (11)$$

Dans la pratique, le calcul de  $\mathbf{W}$  consiste à :

1. Calculer la valeur moyenne des visages (visage moyen)  $\mu$
2. Centrer les images de manière à avoir une matrice de données  $\mathbf{D}$  dont la moyenne est de zéro:

$$\mathbf{D} = [\mathbf{x}_1 - \mu, \mathbf{x}_2 - \mu, \dots, \mathbf{x}_J - \mu]$$

3. Calculer la moyenne de visages de chaque class ( $\mu_i$ ), et la matrice de covariance pour chaque class

$$\mathbf{C}_i = \sum_{j=1}^J (\mathbf{x}_j^i - \mu_i)(\mathbf{x}_j^i - \mu_i)^T = \mathbf{D}_i \mathbf{D}_i^T$$

où  $\mathbf{D}_i$  est la submatrice avec les visages de la class  $i$ .

4. Calculer la somme des matrices de covariance, ponderées par leurs probabilités a priori correspondants:

$$\mathbf{C} = \sum_{i=1}^c \frac{J_i}{J} \mathbf{C}_i$$

5. Décomposer  $\mathbf{C}$  en vecteurs propres  $\mathbf{v}_i$  et valeurs propres  $\lambda_i$ .
6. Choisir les  $K$  premiers vecteurs propres (triés par leurs valeurs propres), et les concaténer pour obtenir la matrice  $\mathbf{W}_a$  que minimise la dispersion intra-classe.
7. Appliquer la descomposition PCA a la matrice de covariance des moyennes  $\mathbf{M}_\mu = [\mu_1, \dots, \mu_c]^T$ :

$$\mathbf{W}_e = PCA(\mathbf{M}_\mu)$$

Cette matrice  $\mathbf{W}_e$  minimise la dispersion inter-classe.

8. Combiner les matrices pour obtenir la matrice de transformation:

$$\mathbf{W} = \mathbf{W}_a \mathbf{W}_e$$

9. Finalement, projeter les données sur les  $K < I$  nouveaux axes:

$$\mathbf{Y} = \mathbf{W}^T (\mathbf{X} - \mu), \quad \mathbf{X} \in \mathbb{R}^I, \quad \mathbf{Y} \in \mathbb{R}^K$$

Le rang de la matrice  $\mathbf{W}_e$  est inférieur ou égal à  $(c - 1)$ , donc il existe au plus  $(c - 1)$  valeurs propres pour  $c$  classes. Cela va limiter le nombre de caractéristiques qui peuvent limiter la performance de reconnaissance dans les applications réelles.

### 1.3 Expérimentations

Utiliser le script `exp2_dimensions.m` pour expérimenter avec PCA et LDA sur des données synthétiques afin de voir les avantages et les inconvénients des deux méthodes. Voir `help pca`, `fisherm` et `map` pour bien comprendre les fonctions.

### 1.4 Préparation de la base de donnée de visages

Vous pouvez télécharger la base de visage de AT&T<sup>3</sup> du site web: [http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att\\_faces.zip](http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.zip), puis l'extraire (par exemple dans un répertoire nommé: 'att').

Les images sont organisées dans 40 répertoires (un pour chaque individu) nommé:  $s_1, \dots, s_{40}$ . Dans chacun de ces répertoires, il y a dix images de visage différentes pour chaque individu numérotées de 1 à 10. Les fichiers sont en format PGM. La taille de chaque image est de 92x112 pixels, avec 256 niveaux de gris par pixel.

Écrire un script pour :

**faces\_to\_prtools\_data.m** Charger les images de chaque individu, les transformer en PRTools 'prdatasets', et les sauvegarder en format .mat. Par exemple, `face1.mat`: (les images du premier individu dans le répertoire  $s_1$ ) doit être une 'prdataset' de: 10 images par 10304 caractéristiques de classe 1.

**read\_faces(path,subject,photo)** Charger (dans une même 'prdataset') les images déjà sauvegardées (`facexx.mat`) en choisissant le numéro de l'individu (`subject`) et le nombre des images (bien sur en gardant les étiquettes). Il est préférable si la fonction accepte des vecteurs en entrée pour faciliter la division des données en apprentissage et test (pour les séances prochaines). Par exemple,

```
% charger les 5 premières images pour les 10 premier individus dans 'Train'
Train = read_faces('./att',1:10,1:5);

% charger les 5 dernières images pour les 10 premier individus dans 'Test'
Test = read_faces('./att',1:10,6:10);
```

Fonctions utiles:

- Matlab: `imread`, `imshow`, `sprintf`, `save`, `load`, `reshape`.
- PRTools: `show`, `im2obj`, `scatterd`.

### 1.5 Application de PCA/LDA sur notre problème

On va appliquer et comparer PCA et LDA sur notre problème d'identification de visage. Dans le cadre de cette séance on ne va pas aller au-delà de la réduction de dimensions.

- Charger les images, par exemple (figure 2a): `Train = read_faces('./att',1:10,1:5)`

---

<sup>3</sup><http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>

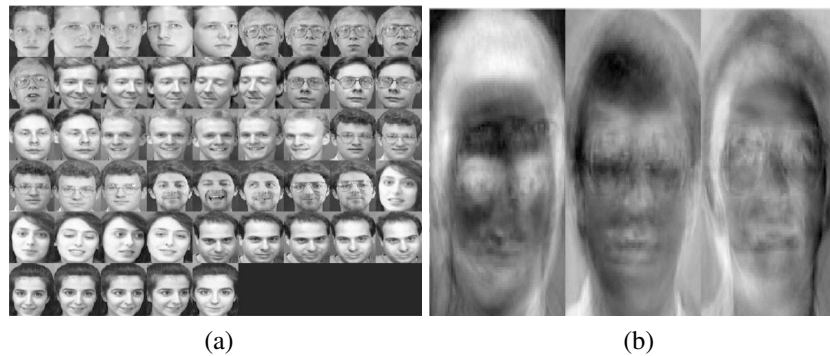


Figure 2: Les 5 premières images appartenant au 10 premiers individus de la base AT&T (2a), avec les 3 premiers 'Eigenfaces' (2b)

- Appliquer la PCA et la LDA sur la base Train
- Reconstruire quelques images (projetées dans le nouvel espace) pour visualiser l'effet de chaque méthode. Par exemple, pour les 3 premiers vecteur pour '*Eigenfaces*', vous devriez obtenir des images similaires à la figure 2b
- Comparer les dispersions de données en utilisant les 2 (ou 3) premières caractéristiques (avec la fonction `scatterd`) provenant des données brutes, de PCA et de LDA. Choisir et comparer aussi autres vecteurs caractéristiques (e.g., les derniers, etc.)
- Expérimenter en variant le nombre:
  - de composantes retenues
  - des individus