

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ ABDELHAMID BEN BADIS DE MOSTAGANEM  
FACULTÉ DES SCIENCES EXACTES ET DE L'INFORMATIQUE  
DÉPARTEMENT DE MATHÉMATIQUES ET INFORMATIQUE  
FILIERE : MATHÉMATIQUES



UNIVERSITE  
Abdelhamid Ibn Badis  
MOSTAGANEM

MÉMOIRE DE FIN D'ÉTUDES

Pour l'Obtention du Diplôme de Master en Mathématiques délivré par

Université de Mostaganem

Spécialité "Modélisation, Contrôle et Optimisation"

*présenté par :*

**Latifa SAID BEKKOUCHE**

**Factorisation matricielle non négative pour la reconnaissance faciale**

*soutenu publiquement le 26 Juin 2019 devant le jury composé de :*

<b>Président :</b>	Pr	BAHRI SIDI MOHAMED	Université de Mostaganem
<b>Examineur :</b>	Dr	ABLAOUI - LAHMAR N	Université de Mostaganem
<b>Encadrant :</b>	Pr	AMIR ABDESSAMAD	Université de Mostaganem
<b>Co-Encadrant :</b>	Dr	HENNI FOUAD	Université de Mostaganem

Année Universitaire : 2018 / 2019

M  
A  
S  
T  
E  
R

# Dédicaces

C'est avec profonde gratitude et sincères mots, que je dédie ce modeste travail de fin d'étude à mes chers parents ; qui ont sacrifié leur vie pour ma réussite et nous ont éclairé le chemin par leurs conseils judicieux. J'espère qu'un jour, je pourrais leur rendre un peu de ce qu'ils ont fait pour moi, que dieu leur prête bonheur et longue vie.

Je dédie aussi ce travail à mes chères sœurs *Nardjiss*, *Amel* et *Hind* et à mes chers frères *Islam* et *Adem* ainsi que toute ma famille.

A tous mes amis et Mes collègues de promotion avec lesquels, j'ai passé mes meilleures années d'études.

A tous mes professeurs qui m'ont enseigné.

A tous ceux qui sont chers.

Et enfin à tous ceux qui ont contribué de près ou de loin à réaliser ce modeste travail.

# Remerciements

Tout d'abord, je remercie Dieu, le Tout Puissant, de m'avoir donné la volonté et le courage afin d'arriver à la finalité de ce modeste travail.

Je tiens à remercier particulièrement Mr. Abdessamad AMIR et Mr. Fouad HENNI pour son bon encadrement, sa disponibilité et ses conseils fructueux qui n'ont cessé de nous prodiguer et ses remarques pertinentes pour la pérennité de cet écrit.

Mon attention se porte aussi sur Mlle. KAISSERLI qui n'a épargné ni temps ni effort pour m'aider et répondre à mes questions.

Que les membres de jury trouvent aussi mes hautes considérations pour avoir accepté d'évaluer mon travail dont leurs jugements me seraient certainement d'un grand profit.

Je ne pourrais clore ces remerciements sans une pensée très personnelle aux membres de ma famille respectueuse pour l'incalculable soutien moral et les nombreux encouragements qu'on a toujours reçus de leur part.

# Table des matières

<b>Liste des tableaux</b>	<b>iv</b>
<b>Table des figures</b>	<b>v</b>
<b>Liste des abréviations</b>	<b>vi</b>
<b>Index des notations</b>	<b>vii</b>
<b>Introduction</b>	<b>1</b>
<b>1 Décomposition en valeurs singulières appliquée au traitement d'image numérique</b>	<b>3</b>
1 Introduction . . . . .	3
2 La décomposition en valeurs singulières . . . . .	4
3 Approche SVD pour la compression d'image . . . . .	6
4 Approche SVD pour la reconnaissance faciale . . . . .	7
<b>2 Éléments de calcul</b>	<b>11</b>
1 Dérivée de la fonction trace . . . . .	11
2 Approximation matricielle de rang minimale . . . . .	14
3 Conclusion . . . . .	18
<b>3 FACTORISATION MATRICIELLE NON NEGATIVE (NMF)</b>	<b>19</b>
1 Introduction . . . . .	19
2 Enoncé du problème . . . . .	19
3 La Formulation du problème . . . . .	20
4 Méthodes de résolution . . . . .	21
<b>4 Expériences Numériques</b>	<b>27</b>
1 Base de Données . . . . .	27
2 La Compression d'Image . . . . .	30
3 La Reconnaissance Faciale . . . . .	33
<b>Conclusion</b>	<b>39</b>
<b>Bibliographie</b>	<b>40</b>
<b>Annexe</b>	<b>43</b>

# Liste des tableaux

4.1	Resumé des résultats de la compression d'images. . . . .	31
4.2	Resumé des résultats de la compression d'images. . . . .	33

# Table des figures

1.1	Une image en niveaux de gris. . . . .	4
1.2	Organigramme de la reconnaissance facial avec SVD . . . . .	10
3.1	Mise à jour multiplicative non croissante . . . . .	22
3.2	Alternance des moindres carrés (ALS) vs Inexact Alternance des moindres carrés (IALS) . . . . .	26
4.1	Les 40 visage de base données "ORL". . . . .	28
4.2	La 1 <sup>ère</sup> et la 35 <sup>ième</sup> personnes de "ORL". . . . .	28
4.3	La base de données "YALE". . . . .	29
4.4	La 2 <sup>ème</sup> et la 14 <sup>ième</sup> personnes de "YALE". . . . .	29
4.5	Exemples d'image utilisée pour tester la compression d'images par SVD. . .	30
4.6	Exemples d'image utilisée pour tester la compression d'images par NMF. . .	32
4.7	Image du visage moyen calculée des images de S. . . . .	34
4.8	La reconnaissance faciale par SVD d' image $\in S$ . . . . .	34
4.9	La reconnaissance faciale par SVD d'image $\notin S$ . . . . .	35
4.10	L'image à gauche est l'image originale du visage, et la 2 <sup>ième</sup> image reconstruite à l'aide d'images de base ( $W$ ) et le codage d'image correspondant ( $h_i$ ) sont affichés directement sur l'image d'origine. . . . .	36
4.11	Organigramme de la reconnaissance facial avec NMF . . . . .	37
4.12	La reconnaissance faciale par NMF d'image $\in S$ . . . . .	37
4.13	La reconnaissance faciale par NMF d'image $\notin S$ . . . . .	38

# Liste des abréviations

- SVD** : Singular Value Decomposition  
, En français Décomposition en valeur singulière.
- DIP** : Digital Image Processing  
, En français Traitement d'image numérique.
- PCA** : Principal Component Analysis  
, En français Analyse des composants principaux APC.
- MSE** : Mean Square Error  
, En français Erreur quadratique moyenne.
- $C_R$  : Compression ratio  
, En français Ratio de compression.
- NMF** : Nonnegative Matrix Factorization  
, En français Factorisation de matrice non négative.
- PCA** : Principal Component Analysis  
, En français Analyse des composants principaux.

# Index des notations

$\mathbb{R}$	: L'ensemble des nombres réels.
$\mathbb{R}_+$	: Ensemble des nombres réels positifs.
$\mathbb{1}_{m \times n}$	: Matrice de taille $m \times n$ dont tous les coefficients sont égaux à 1.
$\mathbf{I}_n$	: Matrice d'Identité ou Matrice unité est une matrice carrée avec des 1 sur la diagonale et des 0 partout ailleurs.
$x$	: vecteur.
$A = (A_{ij})$	: La matrice dont $A_{ij}$ est le terme de la $i^{\text{ème}}$ ligne et de la $j^{\text{ème}}$ colonne.
$A_{:j}$	: La $j^{\text{ème}}$ colonne de la matrice $A$ .
$A_{i:}$	: La $i^{\text{ème}}$ ligne de la matrice $A$ .
$A^T = (A_{ji})$	: La matrice transposée de $A$ dont $A_{ji}$ est le terme de la $j^{\text{ème}}$ ligne et de la $i^{\text{ème}}$ colonne.
$rg(A)$	: Le rang de la matrice $A$ définie la dimension de sous-espace vectoriel engendré par les colonne $A_{:j}$ .
$tr(A)$	: La trace d'une matrice carrée $A$ est définie comme la somme de ses coefficients diagonaux.
$Im(A)$	: L'image de la matrice $A$ .
$\ A\ _2$	: La norme 2 de la matrice $A$ .
$\ A\ _F$	: La norme de Frobenius de la matrice $A$ .
$\nabla F$	: Le gradient de la fonction multivariable $F$ .
$\nabla_x F$	: La dérivée partielle de la fonction $F$ en sa variable $x$ .
$\circ$	: Le produit matriciel de Hadamard.
$\frac{\square}{\square}$	: La division Hadamard.
$\langle A, B \rangle$	: Le produit scalaire de les matrices $A$ et $B$ .



# Introduction

À chaque seconde de cette époque moderne, des tonnes de données sont générées. Pensez au nombre de personnes en ligne qui écrivent leurs blogs, conçoivent leurs pages d'accueil et partagent leurs expériences à travers de nombreux autres supports numériques : vidéos, photos, etc. Pensez également aux données générées lors du décodage des gènes de créatures vivantes et des données acquises de l'espace ou même de notre propre planète, etc. Le développement récent de nouvelles technologies informatiques a engendré une augmentation impressionnante de l'espace disque occupé par toutes sortes de données : images, textes, sons, etc. Cela est dû principalement à l'augmentation des espaces de stockage disponibles et à la diminution importante de leurs coûts.

De nombreuses techniques existent pour le traitement de ces données. En général, on recherche des méthodes aboutissant à un certain degré de compression : en dehors de la réduction de l'espace mémoire occupé, cela permet notamment de filtrer le bruit, de faciliter la visualisation, d'accélérer l'analyse des données, etc. Identifier une personne à partir de son visage est une tâche aisée pour les humains. En est-il de même pour une machine ? Ceci définit la problématique de la reconnaissance automatique des visages, qui a engendré un grand nombre de travaux de recherche au cours des dernières années. La reconnaissance faciale est quelque chose que nous faisons naturellement tous depuis la naissance, mais la reconnaissance automatique du visage est un concept relativement nouveau, où ce dernier utilise une image ou une vidéo de la structure physique du visage d'une personne dans un but particulier. Ce concept a débuté en 1960 par Woodrow Wilson Bledsoe, qui a mis au point le premier système de reconnaissance faciale semi-automatique qui oblige l'administrateur à identifier manuellement les caractéristiques telles que les yeux, les oreilles et le nez sur les photographies.

En raison de l'intérêt accru pour les questions de sécurité mondiale, on s'intéresse de plus en plus aux systèmes informatiques pour identifier les visages et augmenter le nombre de systèmes de sécurité dans ce domaine. Le but de la reconnaissance faciale est de construire des systèmes informatiques adaptés aux personnes. La difficulté de reconnaître les visages par ordinateur varie grandement en fonction des conditions de posture. Ces systèmes sont maintenant utilisés dans divers domaines tels que les robots, la reconnaissance militaire, la télédétection, le traitement de documents et l'automatisation industrielle. Dans les dernières décennies, des informaticiens ont mis au point un certain nombre de produits commerciaux qui ont permis d'améliorer les performances des algorithmes de reconnaissance automatique du visage. Pour un certain nombre de domaines nécessitant des fonctionnalités de reconnaissance du visage, plusieurs questions ont été soulevées pour savoir dans quelle mesure l'algorithme est capable d'identifier un visage approprié.

Malheureusement, à la différence des factorisations classiques (par exemple, les fac-

torisations LU, QR et la décomposition en valeurs singulières (SVD)), il n'existe pas d'algorithme exact et robuste produisant une solution optimale. Les méthodes actuelles de résolution du problème d'optimisation trouvent au mieux un minimum local. Cependant, en pratique, les solutions renvoyées par les algorithmes actuels sont satisfaisantes. Parmi ces approximations on a la méthode de factorisation non négative ( En anglais : Non negative matrix factorisation (NMF)). Les intérêts de cette technique sont multiples. Tout comme la plupart des méthodes de factorisation, elle permet de compresser les données. De plus, grâce à la conservation de la positivité, on peut interpréter le résultat de la factorisation, contrairement aux techniques habituelles. La positivité des facteurs a un autre avantage : ceux-ci seront généralement creux et prendront donc moins d'espace en mémoire.

La reconnaissance faciale combine de nombreuses disciplines de recherche, Pour une simple introduction à ce domaine, nous avons organisé le manuscrit comme suit : Dans le premier chapitre, La technique SVD est présentée, ainsi que ses applications en compression d'image et reconnaissance faciale. Le second chapitre, englobe à la fois les outils mathématiques nécessaires pour la compréhension de la technique NMF, ainsi qu'une introduction à la grande classe des méthodes d'approximation par matrice de rang minimal. Le troisième chapitre est dédié à la factorisation NMF. Comme le problème est un problème de minimisation au sens des moindres carrés, nous caractérisons la solution par le théorème classique de Karush-Kuhn-Tucker et nous présenterons deux méthodes de résolution, à savoir l'algorithme de Lee & Seung, ainsi que la méthode des moindres carrés en alternance (ALS) . Le quatrième chapitre comporte les résultats expérimentaux réalisés avec l'environnement de calcul MATLAB, après application des deux approches SVD et NMF dans la compression d'images et la reconnaissance faciale. Notons qu'un chapitre Annex a été rajouté en dernier, pour les lecteurs non familiarisés avec les concepts mathématiques, tel que la théorie matricielle, l'algèbre linéaire et l'optimisation.

# Chapitre 1

## Décomposition en valeurs singulières appliquée au traitement d'image numérique

### 1 Introduction

Une image peut être définie comme une fonction bidimensionnelle  $f(x, y)$  (image 2D), où  $x$  et  $y$  sont des coordonnées spatiales, et l'amplitude de la fonction  $f$  pour toute paire  $(x, y)$  est le niveau de gris de l'image à ce point.

Par exemple, une image en niveaux de gris peut être représentée par :

$$f_{ij} \quad \text{où} \quad f_{ij} \equiv f(x_i, y_j)$$

Lorsque  $x, y$  et la valeur d'amplitude de  $f$ , sont des quantités finies et discrètes, l'image est appelée «image numérique». L'ensemble fini de valeurs numériques est appelé éléments d'image ou pixels (voir figure 1.1). En règle générale, les pixels sont stockés dans la mémoire de l'ordinateur sous forme de tableau de deux dimension ou de matrice à deux dimensions en nombre réel.

Les images couleur sont formées par une combinaison d'images 2D individuelles. De nombreuses techniques de traitement d'image pour les images monochromes peuvent être étendues à l'image couleur (3D) en traitant l'image à trois composants [33].

Le traitement numérique de l'image (DIP) désigne le traitement d'une image numérique au moyen d'un ordinateur. Étant donné que les données de l'image numérique sont sous forme de matrice, le DIP peut utiliser un certain nombre de techniques mathématiques. Les domaines essentiels sont l'algèbre linéaire, les transformations intégrales, les statistiques et d'autres techniques telles que, l'analyse et l'optimisation numérique. De nombreux algorithmes DIP peuvent être écrits en terme d'équation matricielle, L' algèbre linéaire devient donc un aspect important du sujet [5].

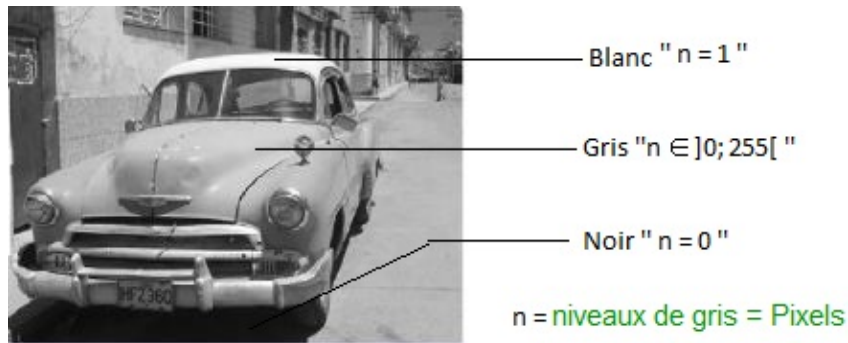


FIGURE 1.1 – Une image en niveaux de gris.

## 2 La décomposition en valeurs singulières

Etant donnée une matrice  $A \in \mathbb{R}^{m \times n}$  de rang  $r$ , alors il existe deux matrices orthogonales  $U \in \mathbb{R}^{m \times m}$  et  $V \in \mathbb{R}^{n \times n}$  telles que :

$$A = U \Sigma V^T, \quad (1.1)$$

où

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \dots & \sigma_r & \dots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix}_{m \times n}, \quad (1.2)$$

avec  $\sigma_1 \geq \sigma_2 \geq \dots \sigma_r > 0$ . La décomposition de la matrice dans l'équation (1.1) est appelée, décomposition en valeurs singulières (SVD) de  $A$ . Les scalaires positifs  $\sigma_i$  pour  $i = 1, 2, \dots, r$  dans Eq. (1.2) sont appelées les valeurs singulières de  $A$ . Les vecteurs colonnes  $U_i$  et  $V_j$  de  $U$  et  $V$  sont appelés les vecteurs singuliers gauche et droit de  $A$ , respectivement.

À partir de l'équation (1.1), il s'ensuit que :

$$AA^T = U \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \dots & \sigma_r^2 & \dots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix}_{m \times m} U^T,$$

et

$$A^T A = V \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \dots & \sigma_r^2 & \dots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix}_{n \times n} V^T. \quad (1.3)$$

Par conséquent, les valeurs singulières de  $A$  sont les racines carrées positives des valeurs propres non nulles de  $AA^T$  et  $A^T A$ , le vecteur singulier gauche  $U_i$  est le  $i^{\text{ème}}$  vecteur propre de  $AA^T$ , et le vecteur singulier droit  $V_j$  est le vecteur propre de  $A^T A$  [2].

## 2.1 Propriétés de la SVD

Il existe de nombreuses propriétés de la décomposition SVD, nous présenterons ici que les nécessaires propriétés à l'étude.

1. Les valeurs singulières  $\sigma_1, \sigma_2, \dots, \sigma_n$  sont uniques, mais les matrices  $U$  et  $V$  ne le sont pas.
2. La norme  $L_2$  et la norme de Frobenius d'une matrice  $A \in \mathbb{R}^{m \times n}$  de rang  $r$  sont données respectivement par :

$$\|A\|_2 = \sigma_1, \quad (1.4)$$

et

$$\|A\|_F = \left( \sum_{i=1}^r \sigma_i^2 \right)^{\frac{1}{2}}. \quad (1.5)$$

3. Si  $A$  est de rang  $r$ , alors  $V_1, \dots, V_r$  forment une base orthonormale pour l'espace  $Im(A^T)$  et  $U_1, \dots, U_r$  forment une base orthonormale pour l'espace  $Im(A)$ .
4. Le rang de la matrice  $A$  est égal au nombre de ses valeurs singulières non nulles [36].
- 5.

$$A = USV^T = \sum_{i=1}^r \sigma_i u_i v_i^T \quad (1.6)$$

## 2.2 Exemple

Soit la matrice positive  $A$ ,

$$A = \begin{bmatrix} 3 & 10 & 7 \\ 4 & 2 & 1 \end{bmatrix}.$$

Nous voulons trouver la décomposition SVD de  $A$ ,

On a

$$A^T A = \begin{bmatrix} 25 & 38 & 25 \\ 38 & 104 & 72 \\ 25 & 72 & 50 \end{bmatrix},$$

d'après (1.1) et (1.3), on obtient :

$$A^T A = V \begin{bmatrix} 168.3242 & 0 & 0 \\ 0 & 10.6758 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

où

$$V = \begin{bmatrix} -0.3024 & 0.9485 & 0.0944 \\ -0.7846 & -0.1915 & -0.5897 \\ -0.5413 & -0.2524 & 0.8021 \end{bmatrix} = [v_1 \ v_2 \ v_3].$$

D'où, les valeurs singulières non nulles de  $A$  sont :  $\sigma_1 = \sqrt{168.3242} = 12.9740$  et  $\sigma_2 = \sqrt{10.6758} = 3.2674$ .

Maintenant, d'après (1.1), nous avons  $U\Sigma = AV$  où

$$U\Sigma = [\sigma_1 u_1 \ \sigma_2 u_2 \ 0] = [12.9740 u_1 \ 3.2674 u_2 \ 0],$$

et

$$AV = \begin{bmatrix} -12.5420 & -0.8361 & 0 \\ -3.3201 & 3.1586 & 0 \end{bmatrix}.$$

Donc

$$u_1 = \frac{1}{12.9740} \begin{bmatrix} -12.5420 \\ -3.3201 \end{bmatrix} = \begin{bmatrix} -0.9667 \\ -0.2559 \end{bmatrix}, \quad u_2 = \frac{1}{3.2674} \begin{bmatrix} -0.8361 \\ 3.1586 \end{bmatrix} = \begin{bmatrix} -0.2559 \\ -0.9667 \end{bmatrix}.$$

et

$$U = [u_1 \ u_2] = \begin{bmatrix} -0.9667 & -0.2559 \\ -0.2559 & -0.9667 \end{bmatrix}.$$

On a aussi, d'après (1.4) et (1.5)

$$\|A\|_2 = \sigma_1 = 12.9740 \quad \text{et} \quad \|A\|_F = (\sigma_1^2 + \sigma_2^2)^{\frac{1}{2}} = \sqrt{179} = 13.3791.$$

### 3 Approche SVD pour la compression d'image

La compression d'image s'applique au problème de la réduction de la quantité de données requises pour représenter une image numérique. La compression est obtenue par la suppression de trois redondances de données de base :

1. La redondance du codage, qui est présente quand elle n'est pas optimale ;
2. La redondance interpixel qui résulte de la corrélation entre les pixels ;
3. La redondance psychovisuelle, due à des données ignorées par la vision humaine[1].

Le rang d'une matrice  $A$  est égal au nombre de ses valeurs singulières non nulles. Dans de nombreuses applications, les valeurs singulières d'une matrice diminuent rapidement avec l'augmentation du rang. Cette propriété nous permet de réduire le bruit ou de compresser les données de la matrice en éliminant les petites valeurs singulières dans les rangs supérieurs.

Lorsqu'une image est transformée en SVD, elle n'est pas compressée, mais les données prennent une forme dans laquelle la première valeur singulière contient une grande quantité d'informations d'image. Avec cela, nous ne pouvons utiliser que quelques valeurs singulières afin de représenter l'image avec de petites différences par rapport à l'original.

Nous illustrons le processus de compression d'image SVD comme suit :

De la 5<sup>ième</sup> propriétés. La matrice  $A$  peut être représentée par :

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_r u_r v_r^T. \quad (1.7)$$

Lors de la compression de l'image, la somme n'est pas effectuée sur les derniers valeurs singulières, les valeurs singulières avec des valeurs suffisamment petites sont supprimées (ordonnés sur la diagonale). La matrice de rang  $k$  est obtenue en tronquant ces sommes après les  $k$  premiers termes :

$$A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_k u_k v_k^T \quad (1.8)$$

La mémoire totale pour  $A_k$  sera  $k(m + n + 1)$ .

Le nombre entier  $k$  peut être bien choisi, suffisamment plus petit que  $n$ , et l'image numérique correspondante à  $A_k$  sera encore très proche de l'image d'origine. Cependant, les différents  $k$  auront une image et un stockage correspondants différents. Pour les choix typiques de  $k$ , la mémoire requise pour  $A_k$  sera inférieure à 20%.

Pour mesurer les performances de la méthode de compression d'image SVD, nous pouvons calculer le facteur de compression et la qualité de l'image compressée. Le facteur de compression de l'image peut être calculé à l'aide du taux de compression ([18]) :

$$C_R = \frac{m * n}{k(m + n + 1)},$$

Pour mesurer la qualité entre l'image d'origine A et l'image compressée  $A_k$ , la mesure de l'erreur quadratique moyenne (MSE) peut être calculée [30] :

$$MSE = \frac{1}{mn} \sum_{y=1}^m \sum_{x=1}^n (f_A(x, y) - f_{A_k}(x, y)).$$

Dans le chapitre 4, des expériences numériques seront réalisées afin de montrer l'utilité de la technique SVD en compression d'image.

## 4 Approche SVD pour la reconnaissance faciale

Au cours des dernières décennies, la reconnaissance faciale a largement attiré l'attention des chercheurs dans les domaines de la vision par ordinateur, Réseaux de neurones, Reconnaissance de formes, Apprentissage automatique, etc. L'application de la reconnaissance faciale comprend : le contrôle d'accès basé sur la reconnaissance faciale, l'interaction homme-machine, la sécurité de l'information, etc [20].

Plusieurs approches de reconnaissance faciale ont été proposées pour la reconnaissance faciale bidimensionnelle. Une grande partie du travail a été consacrée à la détection de caractéristiques individuelles telles que les yeux, le nez, la bouche et le contour de la tête, et à la définition d'un modèle de visage par la position, la taille et les relations entre ces caractéristiques [45],[21].

L'approche SVD traite un ensemble de visages connus comme des vecteurs dans un sous-espace, appelé «espace de visage», engendré par un ensemble de vecteurs visage appelé «visage de base» [17]. Comme l'Analyse en composante principale (PCA)([29]), la reconnaissance est effectuée en projetant une nouvelle image sur l'espace, puis une comparaison est faite entre ses coordonnées (position) dans cet espace et les coordonnées des visages connus. Cependant, l'approche SVD a de meilleures propriétés numériques que PCA.

On définit la matrice A en tant qu'ensemble des visages connus. Supposons que chaque image de visage a  $m \times n = M$  pixels et qu'elle est représentée par un  $M \times 1$  vecteurs colonnes  $f_i$ . Un ensemble S de N images de visage d'individus connus formes une matrice  $M \times N$  :

$$S = \{f_1, f_2, \dots, f_N\}.$$

L'image moyenne  $\bar{f}$  de l'ensemble S, est donnée par :

$$\bar{f} = \frac{1}{N} \sum_{i=1}^N f_i. \quad (1.9)$$

La soustraction de  $\bar{f}$  des visages connus donne :

$$A_{.j} = f_j - \bar{f} \quad tq \quad j = 1, 2, \dots, N.$$

Cela donne une autre matrice qu'on va appeler aussi A de taille  $M \times N$  :

$$A = [A_{.1} \quad A_{.2} \quad \dots \quad A_{.N}]. \quad (1.10)$$

Puisque  $U_{.1}, U_{.2}, \dots, U_{.r}$  forme une base orthonormale pour  $Im(A)$ , et les colonnes de  $A$  sont des images de visages,  $Im(A)$  est appelé "sous espace de visages" dans l'espace vectoriel des images de  $M$  pixels, chaque  $U_{.i}$ ,  $i = 1, 2, \dots, r$  peut être appelé un vecteur de base visage.

Soit  $x = ([x_1, x_2, \dots, x_r]^T)$  les coordonnées de toute image visage quelconque  $f$  de taille  $M$ . Alors, d'après le théorème du PCA ([37]),  $x$  est la projection scalaire de  $f - \bar{f}$  sur la base  $U_{.1}, \dots, U_{.r}$  tel que :

$$x = [U_{.1} \quad U_{.2} \quad \dots \quad U_{.r}]^T (f - \bar{f}). \quad (1.11)$$

Ce vecteur de coordonnées  $x$  est utilisé pour trouver laquelle des images visages connus décrit le mieux le visage  $f$ . C'est-à-dire trouver un visage  $f_i$ ,  $i = 1, 2, \dots, N$ , qui minimise la distance :

$$\varepsilon_i = \|x - x^i\|_2 = [(x - x^i)^T (x - x^i)]^{\frac{1}{2}}.$$

Où  $x^i$  est le vecteur de coordonnées de  $f_i$ , qui est la projection scalaire de  $f_i - \bar{f}$  sur le sous-espace vectoriel engendré par les  $U_{.i}$  (Visage de base) :

$$x_i = [U_{.1} \quad U_{.2} \quad \dots \quad U_{.r}]^T (f_i - \bar{f}). \quad (1.12)$$

Un visage  $f$  est classé comme visage  $f_i$  lorsque le minimum  $\varepsilon_i$  est inférieur à un seuil prédéfini  $\varepsilon_0$ . Si non le visage  $f$  est classée comme "visage inconnu".

Si  $f$  n'est pas un visage, sa distance au sous-espace de visage sera supérieure à 0. La projection vectorielle de  $f - \bar{f}$  sur le sous-espace engendré par le visage de base  $U_{.i}$  est donnée par :

$$f_p = [U_{.1} \quad U_{.2} \quad \dots \quad U_{.r}] x,$$

où  $x$  est donné dans (1.11)

La distance de  $f$  à l'espace des visages connus est la distance entre  $f - \bar{f}$  et la projection  $f_p$  :

$$\varepsilon_f = \|(f - \bar{f}) - f_p\|_2 = [(f - \bar{f} - f_p)^T (f - \bar{f} - f_p)]^{\frac{1}{2}}. \quad (1.13)$$

Si  $\varepsilon_f$  est supérieur à un seuil prédéfini  $\varepsilon_1$ , alors  $f$  n'est pas un visage.

#### 4.1 Étapes à suivre pour effectuer la reconnaissance faciale avec SVD

L'organigramme de la reconnaissance faciale avec SVD est présenté à la figure (1.2). Les explications de chaque étape sont définis comme suit :

1. Obtenir un ensemble des visages  $S$  avec  $N$  images de visage de personnes connues.
2. Calcul du visage moyen  $\bar{f}$  de  $S$  par (1.9).
3. Calcul de la matrice  $A$  donnée par (1.10).
4. Calculer la SVD de  $A$  par (1.6)
5. Pour chaque visage connu, calcul les coordonnées de vecteur  $x_i$  à partir de (1.12). Choisir un seuil  $\varepsilon_1$  qui définit la distance maximale autorisée à partir de l'espace. Déterminer un seuil  $\varepsilon_0$  définissant la distance maximale autorisée par rapport à tout visage connu de l'ensemble  $S$ .
6. Pour identifier une nouvelle image d'entrée  $f$ , calcul les coordonnées de son vecteur  $x$  à partir de (1.11), la projection vectorielle  $f_p$  et la distance  $\varepsilon_f$  par (1.13). Si  $\varepsilon_f > \varepsilon_1$ , l'image d'entrée n'est pas un visage.



7. Si  $\varepsilon_f < \varepsilon_1$ , calculez la distance  $\varepsilon_i$  à chaque visage connu. Si tout  $\varepsilon_i > \varepsilon_0$ , l'image d'entrée peut être classée comme visage inconnu. Si  $\varepsilon_f < \varepsilon_1$  et un certain  $\varepsilon_i < \varepsilon_0$ , classifiez l'image d'entrée en tant que visage connu associé au minimum  $\varepsilon_i$ , et cette image peut éventuellement être ajoutée à l'ensemble d'apprentissage originale [17].

Les 5 étapes peuvent être répétées. Cela peut mettre à jour le système avec plus d'instances de visages connus .

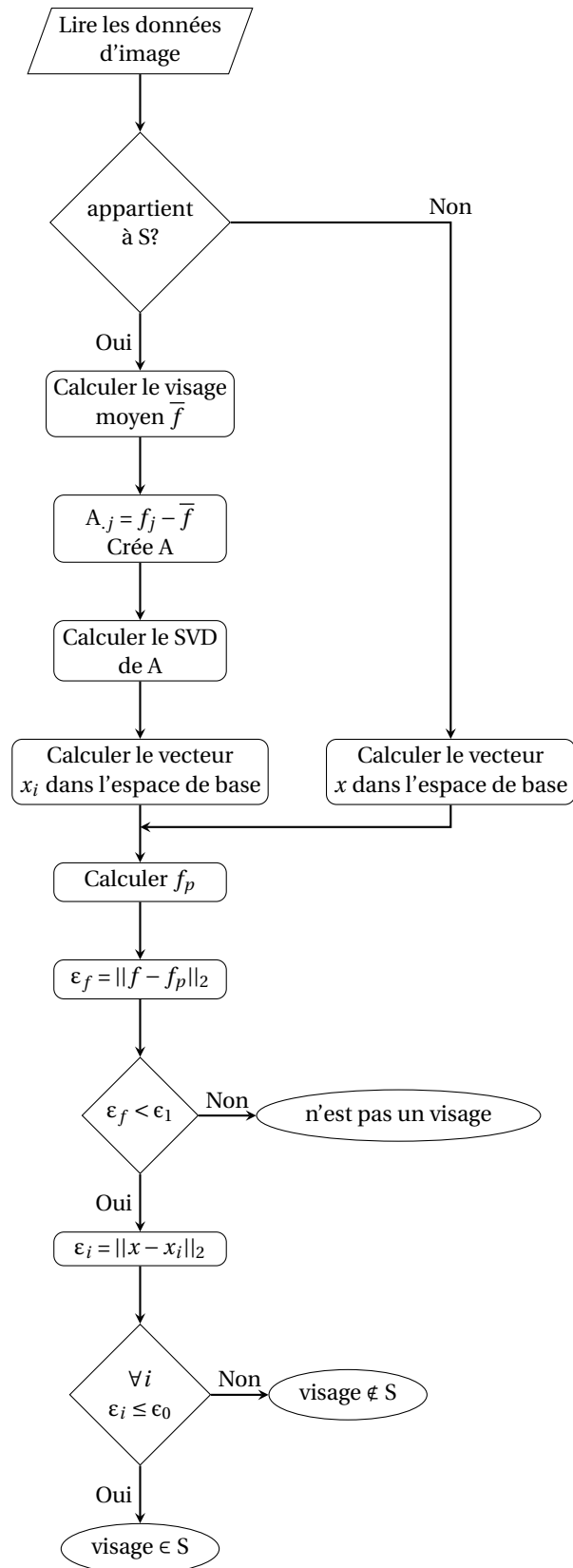


FIGURE 1.2 – Organigramme de la reconnaissance facial avec SVD

# Chapitre 2

## Éléments de calcul

### 1 Dérivée de la fonction trace

Soit  $g$  la fonction définie par :

$$\begin{aligned} g: \mathbb{R}^{n \times n} &\longrightarrow \mathbb{R}^{n \times n} \\ X &\longrightarrow g(X) \end{aligned}$$

et soit la fonction  $f$  définie par :

$$\begin{aligned} f: \mathbb{R}^{n \times n} &\longrightarrow \mathbb{R} \\ X &\longrightarrow f(x) = \text{tr}(g(X)) \end{aligned}$$

Nous voudrions calculer la dérivée de  $f$  par rapport à la matrice  $X$  :

$$\frac{\partial f}{\partial X} = ?$$

La dérivée de la fonction  $f$  par rapport à la matrice  $X$ , nous sera utile dans ce qui suit. Le but de cette section est de calculer cette dérivée.

#### 1.1 Dérivation d'une matrice

Soit la fonction  $f$  définie sur  $\mathbb{R}^{n \times m}$  par :

$$\begin{aligned} f: \mathbb{R}^{n \times m} &\longrightarrow \mathbb{R} \\ X &\longrightarrow f(X) \end{aligned}$$

La dérivée de la fonction  $f$  par rapport à une matrice  $X \in \mathbb{R}^{n \times m}$  s'écrit :

$$\frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f}{\partial X_{11}} & \frac{\partial f}{\partial X_{12}} & \cdots & \frac{\partial f}{\partial X_{1m}} \\ \frac{\partial f}{\partial X_{21}} & \frac{\partial f}{\partial X_{22}} & \cdots & \frac{\partial f}{\partial X_{2m}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial X_{n1}} & \frac{\partial f}{\partial X_{n2}} & \cdots & \frac{\partial f}{\partial X_{nm}} \end{bmatrix}$$

Donc, le résultat est une matrice de même taille que  $X$ .

## 1.2 Dérivés de premier ordre

Il est utile de pouvoir convertir entre la notation matricielle et la notation d'indice.

Par exemple :

Soit  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{n \times r}$  et  $C \in \mathbb{R}^{r \times s}$

Les éléments du produit  $AB$  seront notés par :

$$[AB]_{ik} = \sum_{j=1}^n A_{ij} B_{jk},$$

et les éléments du produit matricielle  $ABC^T$  deviennent :

$$[ABC^T]_{il} = \sum_{j=1}^n A_{ij} [BC^T]_{jl} = \sum_{j=1}^n A_{ij} \sum_{k=1}^r B_{jk} C_{kl} = \sum_{j=1}^n \sum_{k=1}^r A_{ij} B_{jk} C_{kl}.$$

**Exemple 2.1** Soient  $A \in \mathbb{R}^{m \times n}$  et  $X \in \mathbb{R}^{m \times n}$ , définissons la fonction  $f$  :

$$f(X) = \langle A, X \rangle = \text{tr}(A^T X).$$

Nous pouvons écrire ceci en utilisant la notation d'indice :

$$f(X) = \sum_{j=1}^n [A^T X]_{jj} = \sum_{j=1}^n \sum_{i=1}^m A_{ji}^T X_{ij} = \sum_{i=1}^m \sum_{j=1}^n A_{ij} X_{ij}$$

En prenant la dérivée par rapport à  $X_{ij}$ , nous obtenons :

$$\frac{\partial f(X)}{\partial X_{ij}} = A_{ij}$$

Le résultat doit être de la même taille que  $X$ , nous savons donc que les indices des lignes et des colonnes doivent être  $i$  et  $j$ , respectivement. Cela signifie que la dérivée de  $f$  par rapport à la matrice  $X$  est :

$$\frac{\partial f(X)}{\partial X} = A$$

**Exemple 2.2** Considérons cet exemple :

Soient  $A \in \mathbb{R}^{m \times n}$  et  $X \in \mathbb{R}^{m \times n}$

$$f(X) = \langle A, X^T \rangle = \text{tr}(A^T X^T).$$

Nous pouvons écrire ceci en utilisant la notation d'indice :

$$f(X) = \sum_{j=1}^n [A^T X^T]_{jj} = \sum_{j=1}^n \sum_{i=1}^m A_{ji}^T X_{ij}^T = \sum_{j=1}^n \sum_{i=1}^m A_{ji}^T X_{ji}.$$

En prenant la dérivée par rapport à  $X_{ji}$ , nous obtenons :

$$\frac{\partial f(X)}{\partial X_{ji}} = A_{ji}^T,$$

nous savons donc que les indices des lignes et des colonnes doivent être  $i$  et  $j$ , respectivement. Cela signifie que la dérivée de  $f$  par rapport à la matrice  $X$  est :

$$\frac{\partial f(X)}{\partial X} = A^T$$

**Exemple 2.3** Soient  $A \in \mathbb{R}^{m \times n}$ ,  $X \in \mathbb{R}^{n \times r}$  et  $B \in \mathbb{R}^{r \times m}$ , définissons la fonction  $f$  :

$$f(X) = \text{tr}(AXB)$$

On peut écrire ceci en utilisant la notation d'indice :

$$\begin{aligned} f(X) &= \sum_{i=1}^m [A[XB]]_{ii} \\ &= \sum_{i=1}^m \sum_{j=1}^n A_{ij} [XB]_{ji} \\ &= \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^r A_{ij} X_{jk} B_{ki} \end{aligned}$$

En prenant la dérivée par rapport à  $X_{jk}$ , on obtient :

$$\frac{\partial f(X)}{\partial X_{jk}} = \sum_{i=1}^m A_{ij} B_{ki} = \sum_{i=1}^m B_{ki} A_{ij} = [BA]_{kj} = [BA]_{jk}^T$$

Nous savons donc que les indices des lignes et des colonnes doivent être  $j$  et  $k$ , respectivement. Cela signifie que la dérivée de  $f$  par rapport à la matrice  $X$  est :

$$\frac{\partial f(X)}{\partial X} = A^T B^T$$

**Exemple 2.4** Considérons cet exemple :

Soient  $A \in \mathbb{R}^{m \times n}$ ,  $X \in \mathbb{R}^{r \times n}$  et  $B \in \mathbb{R}^{r \times m}$  tel que :

$$f(X) = \text{tr}(AX^T B).$$

On peut écrire ceci en utilisant la notation d'indice :

$$\begin{aligned} f(X) &= \sum_{i=1}^m [A[X^T B]]_{ii} \\ &= \sum_{i=1}^m \sum_{j=1}^n A_{ij} [X^T B]_{ji} \\ &= \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^r A_{ij} X_{jk}^T B_{ki} \\ &= \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^r A_{ij} X_{kj} B_{ki} \end{aligned}$$

En prenant la dérivée par rapport à  $X_{kj}$ , on obtient :

$$\frac{\partial f(X)}{\partial X_{kj}} = \sum_{i=1}^m A_{ij} B_{ki} = \sum_{i=1}^m B_{ki} A_{ij} = [BA]_{kj}$$

Le terme  $X$  apparaissant dans (2.1) avec les indices  $k$  et  $j$ , telle que  $k$  soit l'indice de la ligne et  $j$  l'indice de la colonne. Cela signifie que la dérivée de  $f$  par rapport à la matrice  $X$  est :

$$\frac{\partial f(X)}{\partial X} = BA$$

**Exemple 2.5** *Considérons maintenant un exemple plus compliqué :*

Soient  $A \in \mathbb{R}^{m \times n}$ ,  $X \in \mathbb{R}^{n \times r}$ ,  $B \in \mathbb{R}^{r \times n}$  et  $C \in \mathbb{R}^{m \times r}$  tel que :

$$f(X) = \text{tr}([AXBXC^T]) = \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^r \sum_{l=1}^n \sum_{l=1}^r A_{ij} X_{jk} B_{kl} X_{lk} C_{ik}$$

On pose  $D = BXC^T$  et  $E = AXB$ . Alors la dérivée partielle de  $f$  par rapport à la variable  $X$  est :

$$\frac{\partial f(X)}{\partial X} = \frac{\partial \text{tr}[AXD]}{\partial X} + \frac{\partial \text{tr}[EXC^T]}{\partial X}$$

D'après l'exemple (2.3)

$$\frac{\partial f(X)}{\partial X} = A^T D^T + E^T C$$

## 2 Approximation matricielle de rang minimale

L'approximation de rang minimal est un cas particulier d'approximation matricielle au sens de moindre carré, lorsque seule une contrainte de rang est imposée. L'approximation optimale par rapport à la norme de Frobenius peut être obtenue à partir de la décomposition en SVD.

Nous examinons d'abord le problème sans la contrainte de non-négativité sur l'approximation de rang minimal. Ceci est utile pour comprendre les propriétés de l'approximation lorsque les contraintes de non-négativité sont imposées mais inactives. Nous commençons par le célèbre théorème Eckart-Young. ([13])

**Théorème 2.1 (Eckart-Young)** [13]

Soit  $A \in \mathbb{R}^{m \times n}$  ainsi que sa décomposition en SVD :

$$A = U \Sigma V^T, \Sigma = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}.$$

Où  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$  sont les valeurs singulières de  $A$ , avec  $U \in \mathbb{R}^{m \times m}$  et  $V \in \mathbb{R}^{n \times n}$  sont des matrices orthogonales.

Alors, pour  $1 \leq r \leq n$ , la matrice

$$A_r = U \Sigma_r V^T, \Sigma_r = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \dots & \sigma_r & \dots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{pmatrix}$$

est un minimum global du problème

$$\min_{B \in \mathbb{R}^{m \times n}, \text{rang}(B) \leq r} \frac{1}{2} \|A - B\|_F^2 \quad (2.1)$$

et son erreur est exactement

$$\frac{1}{2} \|A - B\|_{\mathbb{F}}^2 = \frac{1}{2} \sum_{i=r+1}^n \sigma_i^2.$$

De plus, si  $\sigma_r > \sigma_{r+1}$ , alors  $A_r$  est l'unique minimum global.

La preuve du théorème et d'autres résultats associés, peuvent être trouvés dans ([40]). Les colonnes de  $U$  et  $V$  sont appelées vecteurs singuliers de  $A$ , dans lesquels les vecteurs correspondant aux valeurs singulières les plus grandes sont appelés vecteurs singuliers dominants.

Voyons maintenant le problème modifié suivant :

$$\min_{X \in \mathbb{R}^{m \times r}, Y \in \mathbb{R}^{n \times r}} \frac{1}{2} \|A - XY^T\|_{\mathbb{F}}^2. \quad (2.2)$$

Où la contrainte de rang est implicite dans le produit  $XY^T$ , puisque les dimensions de  $X$  et  $Y$  garantissent que le  $\text{rang}(XY^T) \leq r$ . Inversement, chaque matrice de rang inférieur à  $r$  peut être réécrite en tant que produit  $XY^T$ , où  $X \in \mathbb{R}^{m \times r}$  et  $Y \in \mathbb{R}^{n \times r}$ . Par conséquent, les problèmes (2.1) et (2.2) sont équivalents. Mais même lorsque le produit  $A_r = XY^T$  est unique, les paires  $(XR^T, YR^{-1})$  avec  $R$  est inversible donnent le même produit  $XY^T$ . Pour éviter cela, nous pouvons toujours choisir  $X$  et  $Y$  tels que :

$$X = UD^{\frac{1}{2}} \text{ et } Y = VD^{\frac{1}{2}}. \quad (2.3)$$

Où  $U^T U = I_{r \times r}$ ,  $V^T V = I_{r \times r}$  et  $D$  est  $r \times r$  matrice diagonale non négative. Cela équivaut à calculer une décomposition SVD compacte du produit  $A_r = XY^T = UDV^T$ .

Comme d'habitude pour les problèmes d'optimisation, nous calculons le gradient par rapport à  $X$  et  $Y$  en utilisant les résultats de la section précédente.

On a

$$\begin{aligned} F(X, Y) &= \frac{1}{2} \|A - XY^T\|_{\mathbb{F}}^2 \\ &= \frac{1}{2} \langle A - XY^T, A - XY^T \rangle \\ &= \frac{1}{2} [\langle A, A \rangle - \langle A, XY^T \rangle - \langle XY^T, A \rangle + \langle XY^T, XY^T \rangle] \\ &= \frac{1}{2} [tr(A^T A) - tr(A^T XY^T) - tr(YX^T) + tr(YX^T XY^T)], \end{aligned}$$

et

$$\begin{aligned} \frac{\partial tr(A^T XY^T)}{\partial X} &= AY, & \frac{\partial tr(YX^T A)}{\partial X} &= AY \\ \frac{\partial tr(A^T XY^T)}{\partial Y} &= A^T X, & \frac{\partial tr(YX^T A)}{\partial Y} &= A^T X \end{aligned}$$

Maintenant pour calculer  $\frac{\partial tr(YX^T XY^T)}{\partial X}$ , on pose  $C = XY^T$  et  $E = YX^T$ , alors

$$\frac{\partial tr(YX^T XY^T)}{\partial X} = CY + E^T Y = XY^T Y + XY^T Y = 2XY^T Y,$$

et pour calculer  $\frac{\partial tr(YX^T XY^T)}{\partial Y}$ , on pose  $C = X^T XY^T$  et  $E = YX^T X$ , alors

$$\frac{\partial tr(YX^T XY^T)}{\partial Y} = C^T + E = YX^T X + YX^T X = 2YX^T X.$$

Donc

$$\begin{aligned}
 \nabla_X F(X, Y) &= \frac{1}{2}[-AY - AY + 2XY^T Y] \\
 &= \frac{1}{2}[2XY^T Y - 2AY] \\
 &= XY^T Y - AY = 0 \\
 \nabla_Y F(X, Y) &= \frac{1}{2}[-A^T X - A^T X + 2YX^T X] \\
 &= \frac{1}{2}[2YX^T X - 2A^T X] \\
 &= YX^T X - A^T X = 0
 \end{aligned}$$

Si on multiplie  $A^T$  par  $\nabla_X F$  et  $A$  par  $\nabla_Y F$ , on obtient

$$(A^T X)Y^T Y = (A^T A)Y \quad (AY)X^T X = (AA^T)X. \quad (2.4)$$

En remplaçant par  $A^T X = YX^T X$  et  $AY = XY^T Y$  dans (2.4)

$$YX^T XY^T Y = (A^T A)Y \quad XY^T YX^T X = (AA^T)X \quad (2.5)$$

En remplaçant (2.3) dans (2.5) on obtient :

$$(A^T A)VD^{\frac{1}{2}} = VDU^T UDV^T VD^{\frac{1}{2}} \quad \text{et} \quad (AA^T)UD^{\frac{1}{2}} = UDV^T VDU^T UD^{\frac{1}{2}}$$

Quand  $D$  est inversible, cela donne finalement :

$$(A^T A)V = VD^2 \quad \text{et} \quad (AA^T)U = UD^2$$

Cela montre que les colonnes de  $U$  et  $V$  sont des vecteurs singuliers et que  $D_{ii}$  sont des valeurs singulières non nulles de  $A$ . Notez que si  $D$  est singulier, on peut négliger les colonnes correspondantes de  $U$  et  $V$  et les réduire à une approximation plus petite avec les mêmes propriétés. Sans perte de généralité, nous pouvons donc nous concentrer sur des approximations du problème (2.2) qui sont de rang exacte  $r$ . Nous pouvons résumer le raisonnement ci-dessus dans le théorème suivant.

**Théorème 2.2** [31]

Soit  $A \in \mathbb{R}^{m \times n}$  ( $m > n$  et  $\text{rang}(A) = t$ ). Si  $A_r$  ( $1 \leq r \leq t$ ) est un point stationnaire de rang  $r$  du problème (2.2), alors il existe deux matrices orthogonales  $U \in \mathbb{R}^{m \times m}$  et  $V \in \mathbb{R}^{n \times n}$  telles que :

$$A = U \hat{\Sigma} V^T \quad \text{et} \quad A_r = U \hat{\Sigma}_r V^T$$

Où

$$\hat{\Sigma} = \begin{pmatrix} \hat{\sigma}_1 & 0 & \dots & 0 \\ 0 & \hat{\sigma}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{\sigma}_n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \quad \hat{\Sigma}_r = \begin{pmatrix} \hat{\sigma}_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \hat{\sigma}_2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \dots & \hat{\sigma}_r & \dots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{pmatrix}$$

et les  $\hat{\sigma}_i$  sont des valeurs singulières triées de  $A$ . De plus, l'erreur d'approximation est la suivante :

$$\frac{1}{2} \|A - A_r\|_F^2 = \frac{1}{2} \sum_{i=r+1}^n \hat{\sigma}_i^2.$$



Ce résultat montre que, si les valeurs singulières sont toutes différentes, il y a  $\frac{n!}{r!(n-r)!}$  points stationnaires possibles  $A_r$ . Lorsqu'il y a plusieurs valeurs singulières, il y aura une infinité de points stationnaires  $A_r$  puisqu'il y a une infinité de sous-espaces singuliers. Le prochain résultat identifiera les minima parmi tous les points stationnaires. Les autres points stationnaires sont des points selle. (voir Annexes)

**Théorème 2.3** [31] *Les seuls minima du problème (2.2) sont donnés par le théorème 2.1 et sont des minima globaux. Tous les autres points stationnaires sont des points selle.*

**Preuve.** Supposons que  $A_r$  est un point stationnaire donné par le théorème 2.2 mais non par le théorème 2.1. Alors, il existe une permutation des colonnes de  $U$  et  $V$ , ainsi que des diagonales de  $\hat{\Sigma}$  et  $\hat{\Sigma}_r$  telles que  $\hat{\sigma}_{r+1} > \hat{\sigma}_r$ . Alors, nous construisons deux points dans le voisinage  $\varepsilon$  de  $A_r$  qui donnent respectivement une augmentation et une diminution de la distance mesurée. On obtient :

$$\bar{\Sigma}_r(\varepsilon) = \begin{pmatrix} \hat{\sigma}_1 + \varepsilon & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \dots & \vdots \\ 0 & \dots & \hat{\sigma}_r & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{pmatrix}, \quad \bar{A}_r(\varepsilon) = U \bar{\Sigma}_r(\varepsilon) V^T,$$

et

$$\underline{\Sigma}_r(\varepsilon) = \begin{pmatrix} \hat{\sigma}_1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \dots & \vdots \\ 0 & \dots & \hat{\sigma}_r & \varepsilon \sqrt{\hat{\sigma}_r} & \vdots & 0 \\ 0 & \dots & \varepsilon \sqrt{\hat{\sigma}_r} & \varepsilon^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{pmatrix}, \quad \underline{A}_r(\varepsilon) = U \underline{\Sigma}_r(\varepsilon) V^T.$$

Il est clair que  $\bar{A}_r(\varepsilon)$  et  $\underline{A}_r(\varepsilon)$  sont de rang  $r$ . Evaluer la mesure de distance précédent :

$$\begin{aligned} \|A - \underline{A}_r(\varepsilon)\|_F^2 &= 2\hat{\sigma}_r \varepsilon^2 + (\hat{\sigma}_{r+1} - \varepsilon^2)^2 + \sum_{i=r+2}^T \hat{\sigma}_i^2 \\ &= \varepsilon^2 [\varepsilon^2 - 2(\hat{\sigma}_{r+1} - \hat{\sigma}_r)] + \sum_{i=r+1}^T \hat{\sigma}_i^2 \\ &< \sum_{i=r+1}^T \hat{\sigma}_i^2 = \|A - A_r\|_F^2 \end{aligned}$$

Pour tous  $\varepsilon \in (0, \sqrt{2(\hat{\sigma}_{r+1} - \hat{\sigma}_r)})$  et

$$\|A - \bar{A}_r(\varepsilon)\|_F^2 = \varepsilon^2 + \sum_{i=r+1}^T \hat{\sigma}_i^2 > \sum_{i=r+1}^T \hat{\sigma}_i^2 = \|A - A_r\|_F^2$$

Alors  $\forall \varepsilon > 0$  nous obtenons :

$$\|A - \underline{A}_r(\varepsilon)\|_F^2 < \|A - A_r\|_F^2 < \|A - \bar{A}_r(\varepsilon)\|_F^2,$$

ce qui montre que  $A_r$  est un point de selle de la mesure de distance. ■

### 3 Conclusion

Dans l'exemple (2.2), la matrice A est positive, alors que la décomposition SVD donne comme matrices X et Y les deux matrices

$$\begin{aligned} X &= US^{\frac{1}{2}} \\ &= \begin{bmatrix} -3.4820 & -0.4626 & 0 \\ -0.9218 & 1.7474 & 0 \end{bmatrix} \end{aligned}$$

et

$$\begin{aligned} Y &= S^{\frac{1}{2}}V \\ &= \begin{bmatrix} -1.0893 & 3.4164 & 0.3399 \\ -1.4182 & -0.3461 & -1.0660 \end{bmatrix} \end{aligned}$$

On remarque bien que X et Y ne sont pas positives. Dans le domaine du traitement d'image, les matrices sont positives. La technique SVD pour la décomposition 2.2 avec une contrainte de positivité n'est pas efficace, d'où la nécessité de la factorisation NMF.

# Chapitre 3

## FACTORISATION MATRICIELLE NON NEGATIVE (NMF)

### 1 Introduction

Ce chapitre est une introduction au problème de factorisation de matrices non-négatives. La formulation du problème sera présenté ainsi que la description des solutions et quelques observations. Le terme "factorisation" dans l'appellation NMF peut être trompeur car ça peut être compris comme une décomposition exacte, telle que la décomposition SVD, LU ..., où la matrice d'entrée est factorisée en tant que produit d'autres matrices.

Cependant, l'appellation " factorisation de matrice non négative " est devenue si populaire qu'elle a remplacé l'approximation d'une matrice non négative par un produit de deux matrices non négatives.

Nous continuons à utiliser ce terme et nous nous référons à la factorisation matricielle exacte non négative pour le cas exacte.

### 2 Enoncé du problème

La factorisation matricielle non négative a été introduite pour la première fois par Paa-tero et Tapper ([32]). Mais elle a gagné en popularité par les oeuvres de Lee et Seung ([22]), ils soutiennent le fait que la non-négativité est important dans les perceptions humaines et donnent également deux algorithmes simples pour trouver une représentation non négative pour des données non négatives.

Etant donné la matrice  $m \times n$  non négative  $A$  (i.e  $A_{ij} \geq 0$ ) de rang réduit  $r$  ( $r < \min(m, n)$ ), le problème de la factorisation matricielle non négative consiste à trouver deux matrices non négatives  $U \in \mathbb{R}_+^{m \times r}$  et  $V \in \mathbb{R}_+^{n \times r}$  qui approxime  $A$ , i.e

$$A \approx UV^T$$

En regardant les colonnes de  $A$ , on voit que chacun d'eux est approximée par une combinaison conique de  $r$  vecteurs de base non négatifs qui sont les colonnes de  $U$  :

$$A_{:,j} \approx \sum_{j=1}^r V_{ij} U_{:,j}$$

Nous pouvons considérer les colonnes de  $U$ , comme la base du cône  $\mathcal{U}$  complètement contenu à l'intérieur de l'orthant non négatif, chaque colonne de  $A$  est approchée par un élément de  $\mathcal{U}$ , généralement l'élément le plus proche de la colonne.

Nous pouvons également échanger le rôle de  $U$  et  $V$  pour indiquer que chaque ligne de  $A$  est approximée par un élément de  $\mathcal{V}$ , généralement l'élément le proche de la ligne, où  $\mathcal{V}$  est le cône généré par les colonnes de  $V$ .

Il existe plusieurs façons de quantifier la différence entre la matrice de données  $A$  et la matrice  $UV^T$ , mais la mesure la plus utilisée est la norme de Frobenius

$$\mathbf{F}(A, UV^T) = \frac{1}{2} \|A - UV^T\|_F^2 = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n (A_{ij} - [UV^T]_{ij})^2$$

qui est également appelé la distance Euclidienne.

Supposons que  $U$  est fixé, la fonction  $\mathbf{F}(U, V) = \frac{1}{2} \|A - UV^T\|_F^2$  peut être vu comme une composition de la norme de Frobenius et une transformation linéaire de  $V$ . donc,  $\mathbf{F}$  est convexe par rapport à  $V$ , pareillement si  $V$  est fixé  $\mathbf{F}$  est convexe par rapport à  $U$ . Le problème de la factorisation matricielle non négative sera étudié en partant de la norme de Frobenius.

### 3 La Formulation du problème

Etant donné une matrice  $m \times n$  non négative  $A$  et un entier  $r < \min(m, n)$ , résoudre

$$\min_{U \in \mathbb{R}_+^{m \times r}, V \in \mathbb{R}_+^{n \times r}} \frac{1}{2} \|A - UV^T\|_F^2$$

Où  $r$  s'appelle le rang réduit, à partir de maintenant  $m$  et  $n$  seront utilisés pour désigner la taille de la matrice  $A$  et  $r$  est le rang réduit de la factorisation.

Nous réécrivons la factorisation matricielle non négative en tant que problème standard d'optimisation non linéaire. (Voir Annexe)

$$\min_{-U \leq 0, -V \leq 0} \frac{1}{2} \|A - UV^T\|_F^2 \quad (3.1)$$

La fonction Lagrangienne associé est

$$\mathcal{L}(U, V, \mu, \nu) = \frac{1}{2} \|A - UV^T\|_F^2 - \langle \mu, U \rangle - \langle \nu, V \rangle$$

où  $\mu$  et  $\nu$  sont deux matrices de même taille de  $U$  et  $V$  respectivement, contenant les multiplicateurs de Lagrange associés aux contraintes de non-négativité  $U_{ij} \geq 0$  et  $V_{ij} \geq 0$ .

Ensuite, les conditions de Karush-Kuhn-Tucker pour le problème de la factorisation matricielle non négative disent que si  $(U, V)$  est un minimum local, alors il existe  $\mu_{ij} \geq 0$  et  $\nu_{ij} \geq 0$  tels que :

$$U \geq 0, \quad V \geq 0 \quad (3.2)$$

$$\nabla \mathcal{L}_U = 0, \quad \nabla \mathcal{L}_V = 0 \quad (3.3)$$

$$\mu \circ U = 0, \quad \nu \circ V = 0 \quad (3.4)$$

En développement (3.3) par les outils présentés au chapitre précédent nous aurons :

$$AV - UV^T V - \mu = 0 \quad , \quad A^T U - V U^T U - \nu = 0$$

ou bien

$$\mu = -(UV^T V - AV) \quad , \quad \nu = -(V U^T U - A^T U)$$

En combinant ceci avec  $\mu_{ij} \geq 0$ ,  $\nu_{ij} \geq 0$  et (3.4), on obtient les conditions suivantes :

$$U \geq 0, \quad V \geq 0 \quad (3.5)$$

$$\nabla F_U = UV^T V - AV \geq 0, \quad \nabla F_V = VU^T U - A^T U \geq 0 \quad (3.6)$$

$$U \circ (UV^T V - AV) = 0, \quad V \circ (VU^T U - A^T U) = 0 \quad (3.7)$$

où les multiplicateurs de Lagrange correspondants pour U et V sont aussi le gradient de F par rapport à U et V.

Comme la distance Euclidienne n'est pas convexe par rapport aux deux variables U et V en même temps, ces conditions ne sont que nécessaires (Voir Annexe). Nous appelons ensuite tous les points qui remplissent les conditions ci-dessus, les points stationnaires.

**Définition 3.1 (Point stationnaire NMF)** *On appelle (U, V) un point stationnaire du Problème 3.1 si et seulement si U et V satisfont aux conditions KKT (3.5), (3.6) et (3.7).*

## 4 Méthodes de résolution

Dans cette section, nous décrivons brièvement un certain nombre d'algorithmes existants pour le problème de factorisation matricielle non négative et les questions connexes telles que : l'initialisation des algorithmes, les conditions d'arrêt et la convergence.

Nous avons choisi des algorithmes typiques dans deux catégories principales : les mises à jour multiplicatives et les méthodes de calcul des moindres carrés alternatifs. Ceci a été établie en fonction de la popularité des algorithmes dans la pratique. Le deuxième algorithme est la méthode des moindres carrés alternatives proposée par Paatero [32] pour la factorisation de la matrice positive. Moins d'attention a été attribué a cette technique après l'introduction des mises à jour multiplicatives de Lee et Seung [22]. Le problème a ensuite été rebaptisé en factorisation matricielle non négative. La simplicité des mises à jour multiplicatives et l'interprétabilité des résultats ont permis d'étendre l'influence du NMF à presque tous les domaines de recherche : traitement d'image [19] [14] [24], traitement de texte [43] [34], musique. la transcription [35], l'analyse vidéo [9], la bioinformatique [15], la chimie [25]. La méthode standard du gradient projeté a été appliquée au NMF [23], qui présente certains avantages en termes de problèmes à grande taille. Récemment, une version révisée des moindres carrés alternés a été proposée dans [3], offrant une implémentation plus rapide en sacrifiant la propriété de convergence. D'autres tentatives tentent de changer la variable pour éliminer les contraintes de non-négativité.

### 4.1 Algorithme de Lee et Seung

L'algorithme le plus populaire pour le problème NMF sont les règles multiplicatives (Algorithme 1) suggérées par Lee et Seung.

Pour formuler ces règles, nous choisissons de fixer l'un des facteurs (c'est-à-dire U ou V) et essayons de minimiser la fonction de coût par rapport à l'autre facteur. Le développement suivant montre comment formuler la règle de mise à jour pour V. Nous supposons d'abord que U et V sont positifs et nous y reviendrons plus tard.

Puisque la fonction de coût peut être découplée comme suit :

$$\frac{1}{2} \|A - UV^T\|_F^2 = \frac{1}{2} \sum_{i=1}^n \|A_{:,i} - U(V_{:,i})^T\|_2^2.$$

On peut le minimiser par rapport à chacune de lignes de  $V$  séparément. Cela aboutit à la résolution d'une séquence de problèmes quadratiques comme suit :

$$\min_{v \geq 0} F(v) \quad \text{où} \quad F(v) = \frac{1}{2} \|a - Uv\|_2^2.$$

Considérons une approximation actuelle  $\bar{v} > 0$  de la solution et formulons le problème suivant :

$$\min_{v \geq 0} \bar{F}(v) = \frac{1}{2} \left[ \|a - Uv\|_2^2 + (v - \bar{v})^T H_{\bar{v}} (v - \bar{v}) \right], \quad (3.8)$$

où  $H_{\bar{v}} = D_x - U^T U$  avec  $x = \frac{[U^T U \bar{v}]}{[\bar{v}]}$ . Parce que nous pouvons prouver la semi-définitivité positive de  $H_{\bar{v}}$ , nous avons  $\bar{F}(v) \geq F(v)$  pour tout  $v$  et surtout  $\bar{F}(\bar{v}) = F(\bar{v})$ . De plus, la fonction est également convexe. Nous fixons la dérivée de  $\bar{F}(v)$  à zéro, c'est-à-dire :

$$\nabla_v \bar{F} = U^T U v - U^T a + H_{\bar{v}} (v - \bar{v}) = 0,$$

pour obtenir le minimum  $v^*$  :

$$(U^T U + H_{\bar{v}}) v^* = U^T a - H_{\bar{v}} \bar{v}.$$

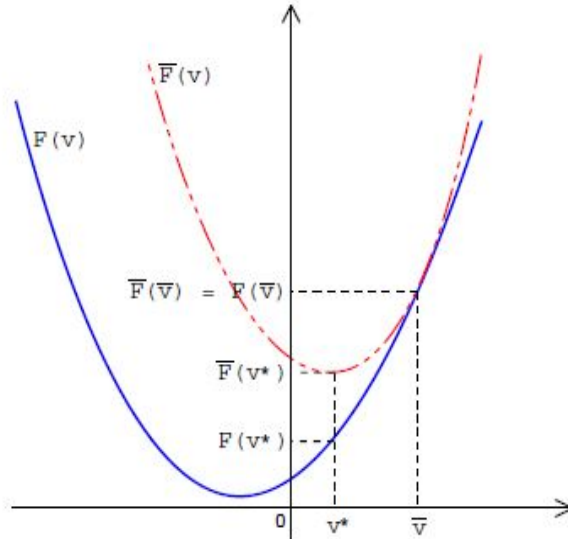


FIGURE 3.1 – Mise à jour multiplicative non croissante

Puisque  $U^T U + H_{\bar{v}} = D_{U^T U \bar{v}} D_{\bar{v}}^{-1}$  et  $H_{\bar{v}} \bar{v} = 0$ , nous concluons :

$$v^* = \bar{v} \circ \frac{[U^T a]}{[U^T U \bar{v}]}. \quad (3.9)$$

Puisque  $v^*$  est le minimum global de  $\bar{F}(v)$ , nous avons  $\bar{F}(v^*) \geq \bar{F}(\bar{v})$ . De plus,  $\bar{F}(v)$  est construit pour satisfaire  $\bar{F}(v) \leq F(v)$  pour tout  $v$ . Cela implique que  $F(v^*) \leq \bar{F}(v^*) \leq \bar{F}(\bar{v}) = F(\bar{v})$  ou nous avons une descente sur la fonction de coût. Ceci peut être résumé à la figure 3.1. La fonction  $\bar{F}$  est généralement appelée fonction auxiliaire.

La résolution pour toutes les lignes de  $V$  donne la règle de mise à jour souhaitée pour  $V$ . La règle de mise à jour pour  $U$  peut être dérivée de la même manière. Et ces mises à jour sont les deux étapes alternées de l'algorithme 1.

Le terme supplémentaire dans (3.8) peut également être considéré comme une fonction de pénalité pour : empêcher l'annulation de la solution. De plus, la matrice  $D_x - U^T U$  avec  $x = \frac{[U^T U \bar{v}]}{[\bar{v}]}$  peut être vue comme une approximation de la matrice hessienne  $U^T U$ .

Le développement ci-dessus peut être résumé dans le théorème suivant :

---

**Algorithm 1** Multiplicative Rules

---

- 1: Initialiser  $U^0, V^0$  et  $k = 0$
  - 2: **repeat**
  - 3:      $U^{k+1} = U^k \circ \frac{[AV^k]}{[U^k(V^k)^T V^k]}$
  - 4:      $V^{k+1} = V^k \circ \frac{[A^T U^{k+1}]}{[V^k(U^{k+1})^T U^{k+1}]}$
  - 5:      $k = k + 1$
  - 6: **until** Condition d'arrêt
- 

**Théorème 3.1** *La distance euclidienne  $\|A - UV^T\|_F^2$  n'augmente pas selon les règles de mise à jour de l'algorithme 1.*

Le théorème 3.1 est une version abrégée de celle de l'article de Lee et Seung [22]. Le théorème d'origine comporte une partie supplémentaire affirmant que la distance euclidienne reste inchangée selon les règles de multiplication uniquement lorsqu'elle se trouve à un point stationnaire. Ce n'est en fait pas nécessairement vrai, car si elle converge, seules les conditions (3.5) et (3.7) sont satisfaites au point fixe. Aucune preuve n'est fournie pour montrer que les conditions (3.6) peuvent être remplies. L'examen de la convergence de ces règles multiplicatives se heurte à deux obstacles principaux.

Le premier est que ces règles multiplicatives ne parviennent pas à faire une descente suffisante de la fonction de coût. Pour voir cela, nous pouvons réécrire (3.9) en tant que méthode de métrique variable :

$$\begin{aligned}
 v^* &= \bar{v} \circ \frac{[U^T a]}{[U^T U \bar{v}]} \\
 &= \bar{v} \circ \frac{[U^T a + U^T U \bar{v} - U^T U \bar{v}]}{[U^T U \bar{v}]} \\
 &= \bar{v} \circ \left(1 + \frac{[U^T a - U^T U \bar{v}]}{[U^T U \bar{v}]}\right) \\
 &= \bar{v} - \frac{[\bar{v}]}{[U^T U \bar{v}]} \circ (U^T U \bar{v} - U^T a) \\
 &= \bar{v} - D_{\bar{v}} \nabla_{\bar{v}} F
 \end{aligned} \tag{3.10}$$

où  $D_{\bar{v}}$  est une matrice diagonale positive avec  $D_{ii} = \frac{[\bar{v}]_i}{[U^T U \bar{v}]_i}$ . Avec cette mise à jour, une condition nécessaire pour une descente suffisante est que les valeurs propres de la matrice  $D_{\bar{v}}$  (c.-à-d.  $D_{ii}$ ) soient délimitées au-dessus et à l'écart de zéro. Mais ce n'est pas vrai pour  $D_{\bar{v}}$  en général. Par conséquent, les points limites de l'algorithme peuvent ne pas être stationnaires.

Le deuxième obstacle est la possibilité de zéros dans  $U$  et  $V$  lorsque l'on considère les conditions (3.6). La situation que nous voulons éviter est :

$$\nabla_{V_{ij}} F < 0 \quad \text{tandisque} \quad V_{ij} = 0 \tag{3.11}$$

pour certains  $(i, j)$  à un point limite. En supposant que les valeurs initiales  $U$  et  $V$  sont positives, ce problème pourrait ne pas se produire car si, pour certains  $k$  :

$$\nabla_{V_{ij}^k} F < 0 \quad \text{ou} \quad [V^k (U^k)^T U^k]_{ij} - [AU^k]_{ij} < 0,$$

nous avons de la mise à jour

$$V_{ij}^{k+1} = V_{ij}^k \frac{[AU^k]_{ij}}{[V^k (U^k)^T U^k]_{ij}},$$

Ceci est également soutenu par le lemme suivant :

**Lemme 3.1** *Si  $U^0$  et  $V^0$  sont positifs et que  $A$  ne contient pas ligne ou colonne nulle, alors  $U_k$  et  $V_k$  sont positifs.*

Mais dans la pratique, lors de l'utilisation d'un système numérique à précision finie, seuls les nombres dont l'amplitude est supérieure à la constante  $\epsilon_M$  (le plus petit nombre représentable d'un système numérique) peuvent être représentés par des nombres non nuls. Sinon, ils seront arrondis à zéro. Et comme nous ne pouvons pas lier  $V_{ij}^k$  à zéro, il pourrait être inférieur à  $\epsilon_M$  et donc représenté par 0. Une fois que  $V_{ij}^k = 0$ , les mises à jour multiplicatives ne le rendront pas positif à nouveau. L'algorithme sera alors probablement piégé dans un point non stationnaire.

Une façon d'éviter cela est de considérer chaque  $U_{ij} = 0$  (ou  $V_{ij} = 0$ ), si l'élément correspondant du gradient est négatif, nous pouvons définir  $U_{ij} = \epsilon$  (ou  $V_{ij} = \epsilon$ ) avec  $\epsilon > \epsilon_M$  ou faire un pas de descente progressive vers l'intérieur de l'orthant non négatif.

**Remarque 3.1** *il est possible que  $[V^k(U^k)^T U^k]_{ij} = 0$  pour certains  $(i, j)$ , ce qui entraîne des exceptions de division zéro. Nous étudions deux situations possibles suivantes :*

*bullet* Lorsque  $V_{ij}^k > 0$ ,  $[V^k(U^k)^T U^k]_{ij} = 0$  implique que

$$0 = \sum_{lt} V_{it}^k U_{lt}^k U_{lj}^k \leq V_{ij}^k U_{lj}^k U_{lj}^k = V_{ij}^k \sum_l U_{lj}^k U_{lj}^k$$

*Cela se produit uniquement lorsque  $U_{:j}^k = 0$ , ce qui est dû à une approximation sans rang et peut être corrigé en générant une substitution pour  $U_{:j}^k$ .*

*bullet* Lorsque  $V_{ij}^k = 0$ , nous avons une situation 0/0 où

$$\nabla_{V_{ij}} F = -[AU^k]_{ij} \geq 0.$$

*Nous ne devrions donc pas remplacer  $[V^k(U^k)^T U^k]_{ij}$  par  $\epsilon$  (petite constante positive avec  $\epsilon > \epsilon_M$ ) comme suggéré par de nombreux travaux, car cela gardera  $V_{ij}^{k+1} = 0$ , ce qui est défavorable pour les mises à jour multiplicatives. Définir  $V_{ij}^{k+1} = \epsilon > \epsilon_M$  est certainement un meilleur choix.*

Les règles de multiplication sont également étendues à la factorisation matricielle pondérée non négative, à la divergence généralisée de Kullback-Leibler et à une classe plus large de fonction de coût, à savoir la divergence de Bregman [11]. Beaucoup d'autres extensions peuvent être trouvées dans [19], [39], [26], [42], [10], etc.

## 4.2 Méthode des moindres carrés en alternance (ALS)

Le premier algorithme proposé pour résoudre la factorisation matricielle non négative était la méthode des moindres carrés alternatives [32]. Il est connu que, en fixant  $U$  ou  $V$ , le problème devient un problème des moindres carrés avec une contrainte de non-négativité.

Puisque les problèmes de moindres carrés de l'algorithme 2 peuvent être parfaitement découplés en problèmes plus petits correspondant aux colonnes ou aux lignes de  $A$ , nous pouvons appliquer directement des méthodes pour le problème des moindres carrés non négatifs à chacun des petits problèmes. Les méthodes pouvant être appliquées sont [44], [12], [4], etc.



Une application directe du théorème 4.4 peut montrer que, si les sous-problèmes (3.12) et (3.13) de l'algorithme 3 sont résolus de manière précise et unique, chaque limite point de l'algorithme 2 est un point stationnaire du problème de factorisation non négative de la matrice.

---

**Algorithm 2** Alternance de moindre carré (ALS)

---

Initialiser U et V

2: **repeat**

    Résoudre :

$$\min_{V \geq 0} \frac{1}{2} \|A - UV^T\|_F^2 \quad (3.12)$$

4:      Résoudre :

$$\min_{U \geq 0} \frac{1}{2} \|A^T - VU^T\|_F^2 \quad (3.13)$$

**until** Condition d'arrêt

---



---

**Algorithm 3** Inexacte moindres carrés alternés (IALS)

---

Initialiser U et V

**repeat**

3:      Résoudre pour U dans l'équation :  $UV^T V = AV$

$U = [U]_+$

        Résoudre pour V dans l'équation :  $VU^T U = A^T U$

6:       $V = [V]_+$

**until** Condition d'arrêt

---

Mais même avec l'implémentation rapide de ces algorithmes, ils ne peuvent pas correspondre aux autres méthodes en termes de temps d'exécution. Une modification a été faite en remplaçant une solution exacte du problème des moindres carrés non négatifs par la projection de la solution du problème des moindres carrés non contrainte dans l'orthant non négatif [3] comme dans l'algorithme 3. Ceci accélère l'algorithme en sacrifiant la propriété de convergence. La figure 3.2 est un exemple typique de la convergence des moindres carrés en alternance et des moindres carrés en alternance inexacts. On peut voir que même si le premier effectue toujours une mise à jour en descente, le dernier ne le fait pas. La méthode exacte produit également de meilleures erreurs d'approximation. Mais avec le même nombre d'itérations, il passe beaucoup plus de temps que la version inexacte (3.435s contre 0.02s). Notez que le solveur du problème des moindres carrés non négatifs dans cet exemple est la fonction standard de Matlab, *lsqnonneg*. Pour un solveur plus rapide tel que [4], il est rapporté que la méthode exacte est encore loin derrière en terme de temps d'exécution. En pratique, les moindres carrés alternés exacts sont rarement utilisés car ils sont très inefficaces. Et sa version inexacte ne converge généralement pas vers un point stationnaire. Il est suggéré d'utiliser la version inexacte comme phase d'initialisation d'un algorithme hybride [16].

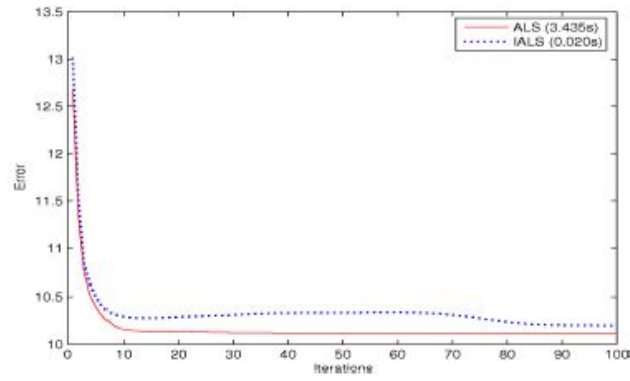


FIGURE 3.2 – Alternance des moindres carrés (ALS) vs Inexact Alternance des moindres carrés (IALS)

# Chapitre 4

## Expériences Numériques

### 1 Base de Données

Cet est consacré à la conception et réalisation de nos applications. La première partie de ce chapitre est une application de la compression d'image par la méthode SVD et la méthode NMF. La deuxième partie est dédiée à la reconnaissance faciale .

Ces applications sont réalisées sur des base données. Il y a plusieurs bases de données contenant des informations qui permettent l'évaluation des systèmes de reconnaissance de visages sont disponibles . Toutefois, ces bases de données sont généralement adaptées aux besoins de quelques algorithmes spécifiques de reconnaissance, chacune d'elle a été construite avec des conditions d'acquisition d'images de visages diverse (changements d'illumination, de pose, d'expressions faciales) ainsi que le nombre de sessions pour chaque individu.

Les bases ORL et YALE ([27],[27]), ont été les plus utilisées et ont permis de comparer plusieurs méthodes de l'état de l'art. Cependant, d'autre bases de visages sont disponibles et destinées à des évaluations adaptées à certaines variabilités du visage telles que les bases (Color FERET, FRGC, CVL, AR et IV2).

#### 1.1 La base de données "ORL"

La base de données ORL a été élaborée entre avril 1992 et 1994 par le Laboratoire AT & T à l'université de Cambridge en Angleterre [27]. La base de données contient les visages de 40 personnes voir figure 4.1, chacune étant enregistrée dans 10 vues différentes comme le montre la figure 4.2. Les images sont à niveau de gris et de  $112 \times 92$  pixels (92 pixels de colonne, 112 pixels de ligne). Pour certains sujets, les images ont été collectées à des dates différentes et avec des variations dans : les conditions d'éclairage, les expressions faciales et par port des lunettes. Toutes les images ont été recueillies sur un fond sombre. Les formes de tête ont quelques différences de profondeur par rapport à la position frontale. Cependant, ces différences ne concernent que des personnes spécifiques et sont donc irrégulières.



FIGURE 4.1 – Les 40 visage de base données "ORL".



(a)



FIGURE 4.2 – La 1<sup>ère</sup> et la 35<sup>ième</sup> personnes de "ORL".

## 1.2 La base de données "YALE"

Elle se compose de 165 images frontales en niveau de gris de 15 personnes voir figure 4.3, avec 11 images pour chacune voir figure 4.4. On trouve trois angles d'éclairage différents :

gauche, centre et droit, et il existe des images avec lunette et sans lunettes. La base offre des images incluant différentes expressions faciales : normale, triste, heureux, somnolant, étonnant, et clignotement de l'œil



FIGURE 4.3 – La base de données "YALE".



(a)

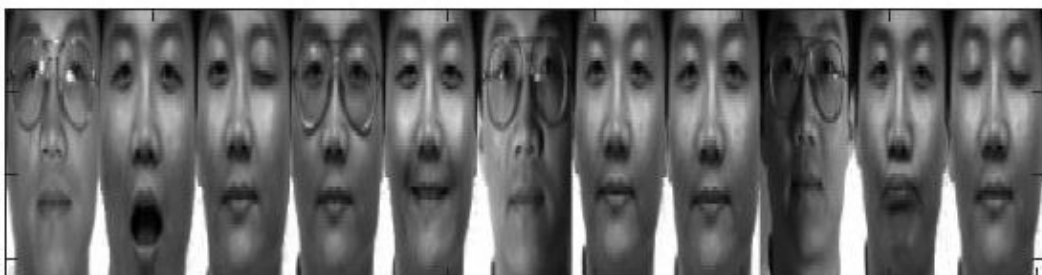


FIGURE 4.4 – La 2<sup>ème</sup> et la 14<sup>ème</sup> personnes de "YALE".



Tableau (4.1) montre un résumé des résultats obtenus par la mesure de l'espace de stockage, et les mesures d'erreur pour les images testées.

$k$	Espace de stockage (Octets)	$C_R$ Comp	MSE (Qualité)
10	2050	5.03	96.89
15	3075	3.53	53.69
20	4100	2.51	32.02
25	5125	2.01	19.90
30	6150	1.68	12.78
35	7175	1.44	8.55
40	8200	1.26	5.74
Originale	10304	1	

TABEAU 4.1 – Résumé des résultats de la compression d'images.

A partir des résultats du tableau (4.1), on remarque :

1. Si  $k$  augmente, la qualité du visage sera proche de l'originale.
2. Plus la valeur singulière est grande (plus grand  $k$ ), la mesure de qualité MSE est plus petite (meilleure qualité d'image).
3. Pour l'image de l'expérience, à partir de  $k = 30$ , on a une image acceptable avec un taux de compression  $C_R = 1.68$ .
4. L'image est proche de l'originale quand  $k = 40$ . En ce point  $C_R = 1.26$  et  $MSE = 5.74$ .

## 2.2 La Compression d'Image par NMF

### Le concept de NMF

Étant donné une matrice de données  $S = \{S_{ij}\}_{n \times m}$ , la factorisation matricielle non négative, décompose la matrice  $S$  en deux matrices  $W$  et  $H$  de taille  $n \times r$  et  $r \times m$ , respectivement, telles que

$$S = WH, \quad (4.1)$$

les éléments dans  $W$  et  $H$  sont tous positifs. À partir de cette décomposition, une représentation réduite est obtenue en choisissant  $r$  tel que  $r < \min(m, n)$ .

La reconstruction d'un objet est effectuée uniquement en ajoutant ses parties représentatives de manière collective. Chaque colonne de la matrice  $W$  est appelée une image de base et une colonne de la matrice  $H$  est appelée un codage. Une image (colonne) en  $S$  peut être reconstruite en combinant linéairement des images de base avec les coefficients d'un codage. Les codages influencent l'activation des pixels dans la matrice d'origine via des images de base.

Chaque élément de la matrice  $S$  peut être écrit sous la forme  $S_{ij} = \sum_{k=1}^r W_{ik}H_{kj}$  où  $r$  représente le nombre d'images de base et le nombre de coefficients d'un codage.

**Résultats de l'expérience**

Soit  $A$  la matrice associée à l'image originale, en utilisant la factorisation matricielle non négative (NMF) (4.1). Nous allons compresser l'image avec différentes valeurs du rang  $r$ .

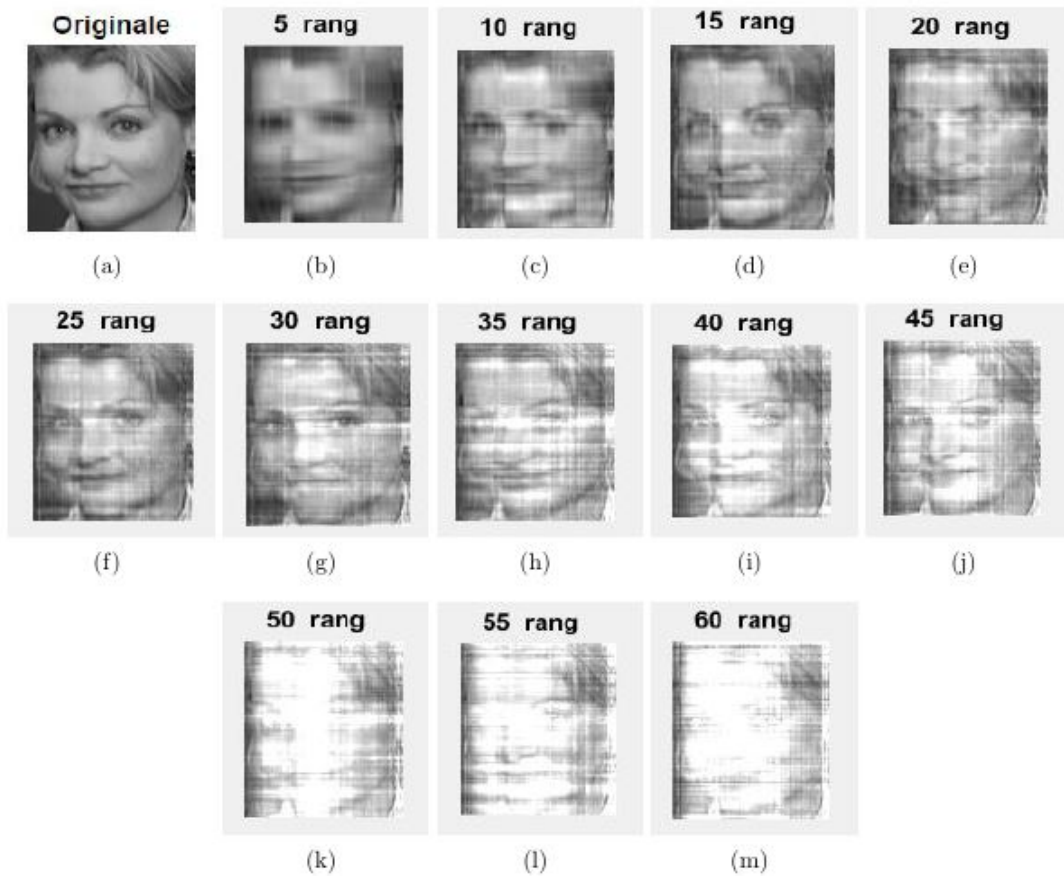


FIGURE 4.6 – Exemples d'image utilisée pour tester la compression d'images par NMF.

La figure 4.6 est les résultats obtenus en compressant l'image d'origine avec NMF avec différentes valeurs de rang  $r$ .

À travers les résultats obtenus, nous notons que lorsque la valeur de  $r$  est petite, l'image compressée est proche de l'image d'origine.



$r$	Espace de stockage (Octets)	$C_R$ Comp	MSE (Qualité)
5	1025	10.053	0.0040
10	2050	5.0263	0.0032
15	3075	3.550	0.0087
20	4100	2.513	0.0182
25	5125	2.010	0.0167
30	6150	1.675	0.0417
35	7175	1.436	0.0365
40	8200	1.256	0.0771
45	9225	1.116	0.0373
50	10250	1.005	0.0920
55	11275	0.913	0.2609
60	12300	0.837	0.274
Originale	10304	1	

TABLEAU 4.2 – Résumé des résultats de la compression d’images.

A partir des résultats du tableau (4.2), on remarque : L’image est proche de l’originale quand  $r = 10$ . En ce point  $C_R = 5.0263$  et  $MSE = 0.0032$ .

### 3 La Reconnaissance Faciale

#### 3.1 Resultats de la reconnaissance faciale par SVD

l’expérience est sous un ensemble de traitement d’image de storage  $M = 92 \times 112 = 10304$ , le nombre des visage connus  $N = 40$ . Sous différentes conditions : Toutes les inclinaisons frontales et légères de la tête, différentes expressions faciales.

Essentiellement une image du visage est de  $M$  (disons 10000) dimension. Mais l’ordre  $r$  de la matrice  $A$  est inférieur ou égal à  $N$ . Pour la plupart des applications, un nombre plus petit de visages de base à  $r$  est suffisant pour l’identification. De cette manière, la quantité de calcul est grandement réduit. Les images de ORL (4.1) l’image du visage de base  $S$ , l’image suivante montre la moyenne de l’image de l’ensemble de traitement utilisé pour cette expérience.

A partir de la base ORL, on a choisi la 247<sup>ième</sup> comme image test où le résultat était le suivant :



FIGURE 4.7 – Image du visage moyen calculée des images de S.



FIGURE 4.8 – La reconnaissance faciale par SVD d' image  $\in S$ .

This image is face : 247 Elapsed time is 21.084033 seconds.  
Après on a choisi hors de l'ORL, le résultat était comme suit :

The input image is a unknown face



FIGURE 4.9 – La reconnaissance faciale par SVD d'image  $\notin S$ .

The input image is a unknown face Elapsed time is 38.245135 seconds.

Mais l'ordre  $r$  de la matrice  $A$  est inférieur ou égal à  $N$ . Pour la plupart des applications, un nombre plus petit de visages de base à  $r$  est suffisant pour l'identification. De cette manière, la quantité de calcul est grandement réduit. Les images suivantes montrent l'image du visage de base, la moyenne de l'image de l'ensemble de traitement utilisé pour cette expérience.

### 3.2 La Reconnaissance Faciale par NMF

#### Le concept de NMF

La matrice de données  $S$  est construite de telle sorte que les images de visage de base occupent les colonnes de la matrice  $S$ . Soit l'ensemble des visages  $S = \{f_1, f_2, \dots, f_m\}$ , puis la matrice de données,  $S = [f_1 \ f_2 \ \dots \ f_m]$ . Maintenant, la décomposition de la matrice  $S$  en deux matrices  $H$  et  $W$  est fait en utilisant (4.1), soit  $W = [w_1 \ w_2 \ \dots \ w_r]$  et les codages sont  $H = [h_1 \ h_2 \ \dots \ h_m]$ . Chaque visage  $f_i$  dans  $S$  peut être approximativement reconstruit en combinant linéairement les images de base et les coefficients de codage correspondants  $h_i = (h_{1i} \ h_{2i} \ \dots \ h_{ri})^T$  (voir la figure 4.10). Il est donc possible de modéliser un visage en termes de superposition linéaire. des fonctions de base avec les codages suivants :

$$f_i = Wh_i.$$

Pour chaque visage  $f_i$  de l'ensemble  $S$ , nous calculons les coefficients de codage correspondants. Les images de base en  $W$  sont générées à partir de l'ensemble des visages  $S$ . Les encodages,  $h_i$  de chaque visage de base  $f_i$  est donné par :

$$h_i = [W^T W]^{-1} f_i.$$

Une fois formé, l'ensemble des images de visage,  $\{f_1, f_2, \dots, f_m\}$  est représenté par un ensemble de codages  $\{h_1, h_2, \dots, h_m\}$  de dimension réduite  $r$ .



FIGURE 4.10 – L'image à gauche est l'image originale du visage, et la  $2^{i\text{ème}}$  image reconstruite à l'aide d'images de base ( $W$ ) et le codage d'image correspondant ( $h_i$ ) sont affichés directement sur l'image d'origine.

### Résultats de l'expérience

Soit une image faciale  $f$ , nous pouvons trouver un codage représentatif de  $f$  comme suit :

$$h = [W^T W]^{-1} f$$

un produit scalaire est utilisé pour calculer la similarité entre les codages d'une image  $h_i \in S$  et d'une image test  $h \in$  l'ensemble test. Le cosinus de l'angle entre les deux vecteurs de données est pris comme mesure de similarité :

$$s_i = \frac{h \cdot h_i}{\|h\| \|h_i\|}.$$

La mesure de similarité si détermine le score de correspondance entre les codages  $h$  et  $h_i$  correspondant aux 2 visages  $S$  et  $S_i$ . Le codage de correspondance optimal d'une image de l'ensemble  $S$  peut être donné par  $h_{i^*}$  où

$$i^* = \underset{i}{\operatorname{argmax}} s_i$$

et  $s_{i^*} > e_1$  indiquant que le visage  $f_{i^*}$  est identifié comme étant le visage le plus proche du visage  $f$ . Par conséquent, l'image obtenue qui correspond le mieux à une image de test donnée est celle qui maximise le score  $s_i$  est supérieur à un seuil  $e_1$ . S'il n'y a pas de valeur  $h_i$  pour laquelle le score est supérieur au seuil, l'image est rejetée. Le  $e_1$  est déterminé de manière empirique comme étant le point auquel le taux d'acceptation fausse (classification erronée) et le taux de rejet erroné sont égaux .

l'organigramme 4.11 resume tous les étapes de la reconnaissance faciale.

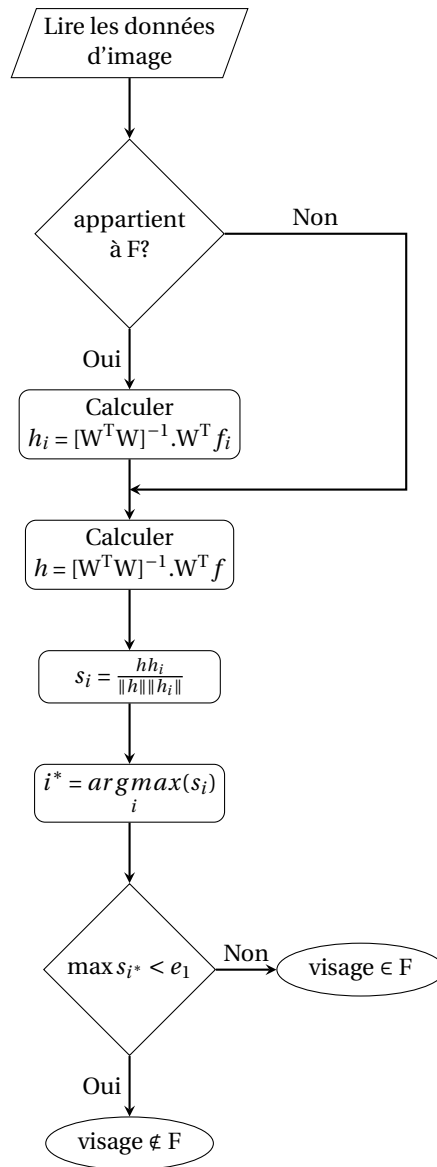


FIGURE 4.11 – Organigramme de la reconnaissance facial avec NMF

### Résultats de l'expérience

Dans cette expérience, on a pris la base "ORL" comme l'ensemble des visages de base, et on a choisi la 247<sup>ème</sup> image de même ensemble comme image test avec  $r = 400$  et  $e_1 = 0.4$  où le résultat était le suivant :



FIGURE 4.12 – La reconnaissance faciale par NMF d'image  $\in S$ .

This image is face :247 Elapsed time is 5.399564 seconds.

### 3. LA RECONNAISSANCE FACIALE

---

Après, on a pris la base "ORL" comme l'ensemble des visage de base, et on a choisi la 50<sup>ième</sup> image de "YALE" comme image test avec  $r = 400$  et  $e_1 = 0.4$  où le résultat était le suivant :

**Test Face**



FIGURE 4.13 – La reconnaissance faciale par NMF d'image  $\notin$  S.

This image is not a face in the dataset Elapsed time is 2.348423 seconds.

# Conclusion

Ce mémoire étudie un sujet relativement récent dans le domaine de la factorisation matricielle à savoir, la factorisation de matrice non négative. On a vu que cette technique permettait à la fois la compression et l'analyse de données positives tel que le visage humain. En effet, en réalisant une factorisation de rang faible, on obtient une base positive de taille réduite à partir de laquelle on peut reconstituer les données par combinaison linéaire à coefficients positifs. Par exemple, pour des images de visages, la factorisation génère les parties constituantes de celles-ci ; bouche, nez, yeux, . .

Les expériences réalisées sur des bases de données de visages, montrent l'efficacité de la technique NMF par rapport à la technique SVD, en compréhension, on a obtenu des images proche des originaux avec un rang plus petit , ainsi qu'en reconnaissance facial les résultats obtenus était très satisfaisant.

A l'issue de ce mémoire, beaucoup de questions restent encore sans réponse. Plusieurs directions de recherches peuvent être envisagées :

- ◇ Critère de décision pour le choix du rang de la factorisation.
- ◇ La possibilité de développer d'autres algorithmes de factorisation, en performant encore plus le problème d'optimisation.
- ◇ Application à d'autres problèmes de reconnaissances.

La NMF est un domaine de recherche encore fertile et en plein expansion, notamment en raison de l'usage toujours croissant de larges structures de données et des nombreuses applications possibles. On espère que ce mémoire inspirera ou incitera d'autres travaux dans le futur.

# Bibliographie

- [1] **R. Albright, J. Cox, D. Duling, A.N. Langville, and C.D. Meyer.** (2006), *Algorithms, initializations, and convergence for the nonnegative matrix factorization.*, Preprint. [6](#)
- [2] **Andreas Antoniou. and Wu-Sheng Lu** (2007), *PRACTICAL OPTIMIZATION, Algorithms and Engineering Applications*, Department of Electrical and Computer Engineering University of Victoria, Canada Springer. [4](#)
- [3] **M.W. Berry, M. Browne, A.N. Langville, V.P. Pauca, and R.J. Plemmons.** (2007), *Algorithms and applications for approximate nonnegative matrix factorization.*, **Computational Statistics and Data Analysis**, **52**(1), 155 – 173. [21](#), [25](#)
- [4] **R. Bro and S. De Jong.** (1997), *A fast non-negativity constrained least squares algorithm.*, *Journal of Chemometrics*, **11**(5) :, pages 393 – 401. [24](#), [25](#)
- [5] **Bernd Jahne.** (2002), *Digital Image Procession*, Springer. [3](#)
- [6] **A. Berman and R.J. Plemmons.** (1994), *Nonnegative matrices in the mathematical sciences.*, Siam. [48](#)
- [7] **D.P. Bertsekas.** (Année), *Nonlinear programming.*, Athena Scientific Belmont, Mass. [50](#)
- [8] **S.P. Boyd and L. Vandenberghe.** (2004), *Convex optimization.*, Cambridge University Press. [49](#)
- [9] **M. Cooper and J. Foote.** (2002), *Summarizing video using non-negative similarity matrix factorization*, In **IEEE Multimedia Signal Processing Workshop**, 25 – 28. [21](#)
- [10] **A. Cichocki, S. Amari, and R. Zdunek.** (2006), *Extended SMART algorithms for non-negative matrix factorization.*, In *Proceeding of the Eighth International Conference on Artificial Intelligence and Soft Computing*, **Volume, Numéro**, pages 548 – 562. [24](#)
- [11] **I.S. Dhillon and S. Sra.** (2005), *Generalized nonnegative matrix approximations with Bregman divergences.*, In *Proceeding of the Neural, Information Processing Systems (NIPS) Conference, Vancouver, BC* [24](#)
- [12] **V. Franc, V. Hlaváč, and M. Navara.** (2005), *Sequential coordinate-wise algorithm for the non-negative least squares problem.*, In **CAIP, Computer Analysis of Images and Patterns**, volume **3691**, pages 407 – 414, September 2005. [24](#)
- [13] **G. Golub., and C.F. Van Loan.** (1996), *Matrix computations*, Baltimore, **The Johns Univ, Press.** xxvii, 694p. [14](#), [46](#)
- [14] **D. Guillaumet, M. Bressan, and J. Vitrià.** (2001), *A weighted nonnegative matrix factorization for local representations*, In **IEEE Computer Society Conference on Compute Vision and Pattern Recognition**, 942 – 947. [21](#)
- [15] **Y. Gao and G. Church** (2005), *Improving molecular cancer class discovery through sparse non-negative matrix factorization.*, *Bioinformatics*, **21**(21), 3970– 3975. [21](#)



- [16] **N. Gillis.** (2007), *Approximation et sous-approximation de matrices par factorisation positive : algorithmes, complexité et applications.*, M.S. Thesis, **Université catholique de Louvain**, 25
- [17] **Guoliang Zeng.** (2006), *Face Recognition with Singular Value Decomposition*, **CISSEL, Proceeding.** 7, 9
- [18] **Gérard Blanchet Maurice Charbit.** (2006), *Digital Signal and Image Processing using MATLAB*, ISTE. 7
- [19] **N.-D. Ho, V. Blondel, and P. Van Dooren.** (2007), *Weighted nonnegative matrix factorization and face feature extraction. Submitted to Image and Vision Computing.* 21, 24
- [20] **S.G. Kong, J. Heo, B.R. Abidi, J. Paik, and M. A. Abidi.** (2005), *Recent advances in visual and infrared face recognition a review.*, **Computer Vision and Image Understanding**, 97(1) :, pages 103 – 135. 7
- [21] **T. Kanade.** (Nov.1973), *Picture Processing System by Computer Complex and Recognition of Human Faces*, Department of Information Science, **Kyoto University.** 7
- [22] **Lee.D.D., and Seung. H.S.** (1999), *Learning the parts of objects by nonnegative matrix factorization.*, Naturel, , 401 . 19, 21, 23
- [23] **C.-J. Lin.** (2007), *Projected gradient methods for non-negative matrix factorization*, Neural Computation, To appear. 21
- [24] **S.Z. Li, X.W. Hou, H.J. Zhang, and Q.S. Cheng.** (2001), *Learning spatially localized, parts-based representation*, In **Proceedings of IEEE Conf. Computer Vision and Pattern Recognition**, 1 – 6. 21
- [25] **H.T. Gao, T.H. Li, K. Chen, W.G. Li, and X. Bi.** (2005), *Overlapping spectra resolution using non-negative matrix factorization.*, Talanta, 66(1), 65 – 73. 21
- [26] **S.Z. Li, X.W. Hou, H.J. Zhang, and Q.S.** (2001), *Learning spatially localized, parts-based representation*, In **Proceedings of IEEE Conf, Computer Vision and Pattern Recognition**, pages 1 – 6. 24
- [27] **Laboratoire.AT.** (1992), *La base données "ORL"*, L'université de Cambridge, <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html> 27
- [28] **D.G. Luenberger.** (2003), *Linear and nonlinear programming.*, Springer. 49
- [29] **Matthew A. Turk, Pentland P. Alex.** (1991), *Face Recognition using Eigenface method*, **IEEE Conference on Computer Vision and Pattern Recognition**, pages 586 – 591. 7
- [30] **Mrak M., Grgic S. and Grgic M.** (September 2003), *Picture Quality Measures in image compression systems*, **IEEE EUROCON, Ljubljana, Eslovenia.** 7
- [31] **Jorge Nocedal, Stephen J. Wright.** (2000), *Numerical Optimization*, Second Edition, Springer. 16, 17, 46
- [32] **Paatero.P and Tapper. U.** (1994), *Positive matrix factorization : a nonnegative factor model with optimal utilization of error estimates of data values*, **Environmetrics**, 5(1), 111 – 126. 19, 21, 24
- [33] **Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins.** (2006), *Digital Image Processing Using MatLab*, Prentice Hall. 3
- [34] **F. Shahnaz, M.W. Berry, V.P. Pauca, and R.J. Plemmons.** (2006), *Document clustering using nonnegative matrix factorization.*, **Information Processing and Management**, 42(2), 373 – 386. 21

- [35] **P. Smaragdis and J.C. Brown.** (2003), *Non-negative matrix factorization for polyphonic music transcription*, **IEEE Workshop on Applications of Signal Processing to Audio and Acoustics**, 177 – 180. [21](#)
- [36] **Steve J. Leon.** (1996), *Linear Algebra with Applications*, Macmillan Publishing Company, New York. [5](#)
- [37] **Smith, Lindsay I.** (2002), *A Tutorial on Principal Component Analysis*, <http://cs-net.otago.ac.nz/cosc453..> [8](#)
- [38] **Topics in matrix analysis.** (1991), *Cambridge*, University Press. viii, 607 p.. [44](#), [47](#)
- [39] **A. Vandendorpe, N.D. Ho, S. Vanduffel, and P. Van Dooren.** (2007), *On the parameterization of the CreditRisk model for estimating credit portfolio risk*. Accepted for publication in *Insurance* ; Mathematics and Economics. [24](#)
- [40] **P. Van Dooren.** (2005), *Théorie des matrices, Notes de cours*, INMA 2380, Université Catholique de Louvain-la-Neuve, 2005. [15](#)
- [41] **C.F. Van Loan** (2000), *The ubiquitous Kronecker product.*, *Journal of Computational and Applied Mathematics*, **123(1-2)**, pages 85 – 100. [44](#)
- [42] **Y.Wang, Y. Jia, C. Hu, and M. Turk.** (2004), *Fisher non-negative matrix factorization for learning local features.*, *Journal*, **In Proceedings of Asian Conference on Computer Vision**. [24](#)
- [43] **W. Xu, X. Liu, and Y. Gong.** (2003), *Document clustering based on nonnegative matrix factorization*, **In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval**, pages 267–273. ACM Press New York, NY, USA. [21](#)
- [44] **C.L. Lawson and R.J. Hanson.** (1974), *Solving least squares problems.*, Prentice-Hall Englewood Cliffs, NJ. [24](#)
- [45] **A. L. Yuille, D. S. Cohen, and P. W. Hallinan.** (June 1989), *Feature Extraction from Faces Using Deformable Templates*, **Proc. CVPR, San Diego**. [7](#)

# Annexe

Cette partie présente les résultats de base et les concepts utilisés tout au long de cette thèse. Les résultats connus ne sont indiqués que sans preuve.

## A. Théorie des matrices et algèbre linéaire

$A$  est la matrice réelle de taille  $m \times n$  tel que  $m$  le nombre des lignes et  $n$  nombre des colonnes. Nous avons une matrice carrée lorsque le nombre de lignes est égal au nombre de colonnes. L'ensemble de  $m \times n$  matrices réelles est noté  $\mathbb{R}_{m \times n}$ . Dans cette thèse, toutes les matrices sont réelles. Nous utilisons des lettres majuscules pour les matrices. La  $i^{\text{ème}}$  ligne de la matrice  $A$  est notée  $A_{i \cdot}$ . La  $j^{\text{ème}}$  colonne de la matrice  $A$  est notée  $A_{\cdot j}$ . L'élément à l'intersection de la ligne  $i$  et de la colonne  $j$  de la matrice  $A$  est noté  $A_{ij}$  ou  $[A]_{ij}$ .

Un vecteur de colonne est une matrice d'une seule colonne. De même, un vecteur ligne est une matrice d'une seule ligne. Sauf indication explicite, un vecteur est toujours un vecteur colonne. L'ensemble de tous les vecteurs de taille  $n$  est  $\mathbb{R}^n$ . Les vecteurs sont désignés par des lettres minuscules, sauf lorsqu'ils font partie d'une matrice, comme décrit dans le paragraphe précédent.

On dit qu'une matrice carrée  $A$  est symétrique si  $A_{ij} = A_{ji}$ , pour tout  $i, j$ . Une matrice diagonale  $D$  est une matrice carrée comportant des éléments non nuls uniquement sur sa diagonale principale (c'est-à-dire,  $A_{ij} = 0$  pour  $i \neq j$ ). Nous utilisons  $D_x$  pour désigner une matrice diagonale avec le vecteur  $x$  sur sa diagonale principale (c.-à-d.  $A_{ii} = x_i$ ,  $i = 1, \dots, n$ ).

Voici quelques matrices spéciales :

- Matrices dont les éléments sont tous 1 :  $\mathbb{1}_{1 \times n}$ ,  $\mathbb{1}_{m \times 1}$ ,  $\mathbb{1}_{m \times n}$ .

$$\mathbb{1}_{1 \times n} = (1, 1, \dots, 1) \quad \mathbb{1}_{m \times 1} = (1, 1, \dots, 1)^T \quad \mathbb{1}_{m \times n} = \mathbb{1}_{m \times 1} \mathbb{1}_{1 \times n}.$$

- Vecteurs base

$$e_i = (0, 0, \dots, \overbrace{1}^{i^{\text{ème position}}}, \dots, 0)^T$$

- Matrices d'identité  $I_n$  : matrice diagonale où les éléments diagonaux sont égaux à 1.
- Matrices de permutation : matrices carrées comportant sur chaque ligne et chaque colonne un seul élément non nul, égal à 1.
- Matrices de sélection : toute sous-matrices de matrices de permutation.

### A.1 Manipulation matricielle

Voici quelques opérations de base sur la matrice

- Matrice transposée  $A^T : [A^T]_{ij} := A_{ji}$ .  $A$  est une matrice symétrique  $\Leftrightarrow A^T = A$ .
- Addition matricielle  $C = A + B : C_{ij} = A_{ij} + B_{ij}$ .

- Produit matricielle  $C = A \cdot B : C_{ij} = \sum_k A_{ik} B_{kj}$ . Le point de produit est souvent supprimé.
- Vectorisation matricielle de  $A \in \mathbb{R}^{m \times n}$

$$\text{vec}(A) = \begin{pmatrix} A_{:1} \\ \vdots \\ A_{:n} \end{pmatrix} \in \mathbb{R}^{mn}.$$

- Produit de Kronecker de la matrice  $A \in \mathbb{R}^{m \times n}$  et de la matrice  $B$

$$A \otimes B = \begin{pmatrix} A_{11}B & \dots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \dots & A_{mn}B \end{pmatrix}.$$

Une relation importante entre le produit matriciel et le produit Kronecker est la suivante [41] :

$$\text{vec}(AXB^T) = (B \otimes A) \text{vec}(X).$$

On écrit  $A < B$  si  $A_{ij} < B_{ij}$  pour tous  $i, j$  et ainsi pour  $A \leq B$ ,  $A > B$  et  $A \geq B$ . On utilise  $A < \alpha$ ,  $A > \alpha$ ,  $A \leq \alpha$  et  $A \geq \alpha$ , où  $\alpha \in \mathbb{R}$ , comme abréviations de  $A < \alpha \mathbb{1}_{m \times n}$ ,  $A > \alpha \mathbb{1}_{m \times n}$ ,  $A \leq \alpha \mathbb{1}_{m \times n}$  et  $A \geq \alpha \mathbb{1}_{m \times n}$ . La matrice absolue  $|A|$  est définie comme suit :  $[|A|]_{ij} = |A_{ij}|$  pour tout  $i, j$ .

Nous définissons le produit scalaire des deux vecteurs réels  $x, y \in \mathbb{R}^n$  comme un réel :

$$\langle x, y \rangle = \sum_i x_i y_i = x^T y.$$

Les vecteurs  $x, y \in \mathbb{R}^n$  non nuls sont dits orthogonaux si leur produit scalaire est égal à zéro :

$$\langle x, y \rangle = 0.$$

On considère une matrice  $A \in \mathbb{R}^{m \times n}$  et sa vectorisation :  $\text{vec}(A) \in \mathbb{R}^{mn}$ , nous pouvons également définir le produit intérieur de deux matrices réelles de même taille :

$$\langle A, B \rangle = \text{vec}(A)^T \text{vec}(B) = \sum_{ij} A_{ij} B_{ij} = \text{tr}(A^T B),$$

où la trace de  $A$  ( $\text{tr}(A)$ ) est la somme de tous les éléments diagonaux de  $A$ . Ceci implique la relation utile suivante :

$$\langle I, ABC \rangle = \langle A^T, BC \rangle = \langle B^T A^T, C \rangle = \langle C^T B^T A^T, I \rangle = \text{tr}(ABC).$$

Une matrice carrée  $A$  est dite inversible s'il existe une matrice  $B$  telle que

$$AB = BA = I,$$

où  $B$  est appelé l'inverse de  $A$  et est noté  $B = A^{-1}$ .

La somme matricielle  $C = A + B$  est définie par  $C_{ij} = A_{ij} + B_{ij}$ . Cette opération est dite élément par élément ou par entrée puisque chaque entrée de la matrice de résultat  $C$  dépend uniquement des entrées de  $A$  et  $B$  au même endroit. Ceci est contraire au produit matriciel habituel  $C = AB$  où les relations ne sont plus locales. Un produit matriciel plus simple, élément par élément, est appelé produit Hadamard ou produit Schur  $C = A \circ B$  où  $C_{ij} = A_{ij} B_{ij}$  et  $A, B$  et  $C$  sont des matrices  $m \times n$ . Cela aide considérablement à simplifier les formules matricielles dans de nombreux cas. Voici quelques propriétés du produit Hadamard [38] :

- $A \circ B = B \circ A$
- $A^T \circ B^T = (A \circ B)^T$
- $(a \circ b)(c \circ d)^T = (ac^T) \circ (bd^T) = (ad^T) \circ (bc^T)$

Voici quelques relations du produit Hadamard avec d'autres opérations :

- $\mathbb{1}^T(A \circ B)\mathbb{1} = \langle B, A \rangle$
- $A \circ B = P^T(A \otimes B)Q$ , où P et Q sont des matrices de permutation

$$P = (e_1 \otimes e_1 \ e_2 \otimes e_2 \ \dots \ e_m \otimes e_m)$$

et

$$Q = (e_1 \otimes e_1 \ e_2 \otimes e_2 \ \dots \ e_n \otimes e_n).$$

À partir de la définition du produit Hadamard, nous pouvons définir d'autres opérations élément par élément :

- Puissance de Hadamard :  $[A^{o^r}]_{ij} = A_{ij}^r \quad r \in \mathbb{R}$
- Division de Hadamard :  $C = \frac{[A]}{[B]} = A \circ B^{o^{-1}}$ .

## A.2 Sous-espaces de vecteurs

Un sous-espace linéaire E de  $\mathbb{R}^n$  est l'ensemble de toutes les combinaisons linéaires d'un ensemble de vecteurs  $V = \{v_1, v_2, \dots, v_k\}$  de  $\mathbb{R}^n$  :

$$E = \left\{ \sum_{i=1}^k \alpha_i v_i ; \alpha_i \in \mathbb{R} \right\}$$

E est également appelé l'espace engendré par V et V est appelé un ensemble générant de E. Dans un sous-espace E, il existe de nombreux ensembles générant. Parmi eux, un ensemble dont aucun vecteur ne peut être supprimé et sans modifier l'espace engendré est dit linéaire, indépendant et base de E. Le cardinal d'une base de E est fixé et est appelée dimension de E.

Par exemple :

$$E = \text{vect}\left(\left\{(1, 2, 1)^T, (1, 0, 0)^T\right\}\right)$$

est un sous-espace de  $\mathbb{R}^3$  et  $\dim(E) = 2$  parce que  $\left\{(1, 2, 1)^T, (1, 0, 0)^T\right\}$  est linéaire indépendant. Suite à cela, le rang d'une matrice A de taille  $m \times n$  peut également être défini comme la dimension du sous-espace engendré par les colonnes de A :

$$rg(A) = \dim(\text{vect}(A_{:1}, A_{:2}, \dots, A_{:n})) \leq \min(m, n).$$

Un sous-espace linéaire est fermé sous addition et multiplication scalaire, c'est-à-dire :

$$u, v \in E \Rightarrow u + v \in E,$$

$$u \in E, \alpha \in \mathbb{R} \Rightarrow \alpha u \in E.$$

## A.3 Valeurs et Vecteurs Propres

Les concepts centraux de l'analyse matricielle sont les valeurs propres et les vecteurs propres d'une matrice carrée. Ils fournissent des informations essentielles sur la matrice. Les concepts associés aux matrices rectangulaires sont ce qu'on appelle les valeurs singulières et les vecteurs. Ils jouent un rôle crucial dans les approximations de rang minimal qui conservent les caractéristiques dominantes de la matrice d'origine.

**Définition 4.1** Le scalaire  $\lambda \in \mathbb{C}$  est une valeur propre de la matrice  $A \in \mathbb{C}^{n \times n}$  s'il existe un vecteur non nul  $x \in \mathbb{C}^n$  tel que  $Ax = \lambda x$ . Le vecteur  $x$  est appelé le vecteur propre associé à la valeur propre  $\lambda$ .

Une matrice  $A$  de taille  $n \times n$  a exactement  $n$  valeurs propres. L'ensemble de toutes les valeurs propres est noté  $\sigma(A)$ . Le module maximum de  $\sigma(A)$  est le rayon spectral de  $A$  et est noté  $\rho(A)$  :

$$\rho(A) = \max\{|\lambda| \ ; \ \lambda \in \sigma(A)\}.$$

Dans cette thèse, seules les valeurs propres et les vecteurs propres de certaines matrices symétriques sont examinés. Pour ces matrices, les résultats bien connus suivants peuvent être établis :

**Théorème 4.1 (Théorème Spectral) [31]**

Soit  $A$  une matrice réelle symétrique. Tous les vecteurs et les valeurs propres de  $A$  sont réels.

De plus, pour une matrice réelle symétrique  $A$ , si toutes les valeurs propres de  $A$  sont non négatives (respectivement non positives),  $A$  est dite semi-définie positive (respectivement semi-définie négative). Si toutes les valeurs propres sont positives (respectivement négatives),  $A$  est dite définie positive (respectivement définie négative).

La décomposition en valeurs singulières définie dans le théorème suivant est un outil très utile en analyse matricielle :

**Théorème 4.2** Pour une matrice quelconque  $A \in \mathbb{R}^{m \times n}$ , il existe deux matrices orthogonales  $U \in \mathbb{R}^{m \times m}$  et  $V \in \mathbb{R}^{n \times n}$  tel que :

$$A = U \Sigma V^T; \tag{4.2}$$

$$\Sigma = \left( \begin{array}{ccc|cc} \sigma_1 & & 0 & & \\ & \ddots & & & \\ 0 & & \sigma_r & & \\ \hline & & & O_{r \times (n-r)} & \\ O_{(m-r) \times r} & & & & O_{(m-r) \times (n-r)} \end{array} \right), \tag{4.3}$$

où les valeurs singulières  $\sigma_i$  sont des scalaires réels et décroissants :

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0. \tag{4.4}$$

La preuve et les algorithmes peuvent être trouvés dans [13]. De plus, les colonnes de  $U$  et  $V$  sont les vecteurs propres de  $A^T A$  et  $A A^T$ , respectivement.

## A.4 Les Normes

Une norme est utilisée pour mesurer la magnitude d'un vecteur ou d'une matrice. Une norme sur  $\mathbb{R}^n$  (ou  $\mathbb{R}^{m \times n}$ ) on la note  $\|\cdot\|$  qui vérifie les quatre conditions suivantes :

1.  $\|x\| \geq 0, \forall x \in \mathbb{R}^n$  (ou  $\mathbb{R}^{m \times n}$ );
2.  $\|x\| = 0 \Leftrightarrow x = 0$ ;
3.  $\|\alpha x\| = |\alpha| \|x\|, \forall x \in \mathbb{R}^n$  (ou  $\mathbb{R}^{m \times n}$ ) et  $\forall \alpha \in \mathbb{R}$ ;

$$4. \|x + y\| \leq \|x\| + \|y\|, \forall x, y \in \mathbb{R}^n \text{ (ou } \mathbb{R}^{m \times n}\text{)}.$$

La norme la plus courante est la norme Euclidienne ou la norme de Frobenius dérivée du produit scalaire :

$$\|x\|_F = \sqrt{\langle x, x \rangle},$$

où  $x$  peut être un vecteur ou une matrice. Cette norme joue un rôle central dans les problèmes des moindres carrés, où l'on essaie de minimiser une erreur mesurée par cette norme.

Les normes populaires sont des exemples des normes de Hölder (norme  $p$ ) :

$$\|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, \quad p = 1, 2, \dots$$

où les plus couramment utilisés sont  $p = 1$ ,  $p = 2$  et  $p = \infty$  :

$$\begin{aligned} \text{Norme1} & : \quad \|x\|_1 = |x_1| + |x_2| + \dots + |x_n| \\ \text{Norme2} & : \quad \|x\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \dots + |x_n|^2} \\ \text{Norme}\infty & : \quad \|x\|_\infty = \max_i |x_i|. \end{aligned}$$

Pour les vecteurs, la norme 2 ( $\|\cdot\|_2$ ) est également la norme de Frobenius ( $\|\cdot\|_F$ ). Mais ce n'est plus vrai pour les normes  $p$  de la matrice, qui sont induites par les normes  $p$  du vecteur :

$$\|A\|_p = \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}.$$

Il est prouvé que [38]

$$\begin{aligned} \text{Norme1} & : \quad \|A\|_1 = \max_j \sum_i |A_{ij}| \\ \text{Norme2} & : \quad \|A\|_2 = [\rho(A^T A)]^{\frac{1}{2}} \\ \text{Norme}\infty & : \quad \|A\|_\infty = \max_i \sum_j |A_{ij}|. \end{aligned}$$

Puisque le problème principal traité dans cette thèse est un problème de moindres carrés contraints, la norme de Frobenius sera largement utilisée. D'autres normes seront également utilisées pour ajouter plus de contraintes au problème principal.

## A.5 Cône convexe et cône polyédrique

Un ensemble  $C \subset \mathbb{R}^n$  est appelé un cône convexe s'il est fermé sous l'addition et la multiplication scalaire non négative, c.-à-d.

$$u, v \in C \Rightarrow u + v \in C,$$

$$u \in C, \alpha \geq 0 \Rightarrow \alpha u \in C.$$

Un cône polyédral est un cône convexe non négatif généré par un ensemble fini de vecteurs  $V = \{v_1, v_2, \dots, v_k\}$  de  $\mathbb{R}^n$  :

$$C = \left\{ \sum_{i=1}^k \alpha_i v_i ; \alpha_i \in \mathbb{R}_+ \right\}$$

Dans cette relation,  $C$  est également appelé l'espace engendré par  $V$  et  $V$  est appelé un ensemble générant de  $C$ . Il existe un ensemble  $\bar{V} \subset V$  qu'est une base génératrice non négative de  $C$  et dont aucun vecteur ne peut être retiré sans changer le cône.  $\bar{V}$  s'appelle le cadre de  $C$  et son cardinal s'appelle la dimension de  $C$ .

### 3.3 Matrices non négatives

Les matrices dont tous les éléments sont non négatifs sont appelées matrices non négatives. Nous utilisons  $\mathbb{R}_+^n$  et  $\mathbb{R}_+^{m \times n}$  pour désigner l'ensemble des vecteurs non négatifs à  $n$  dimensions et l'ensemble des  $m \times n$  matrices non négatives, respectivement. En effet, ces sous-ensembles sont des cônes polyédriques et sont généralement appelés orthant non négatifs.

Une matrice non négative est appelée "ligne admissible" si elle n'a pas de ligne nul. De la même manière, une matrice non négative est appelée "colonne admissible" si elle n'a pas de colonne nul. Une matrice non négative est dite stochastique colonne (ligne) si toutes les sommes de colonne (ligne) sont égales à 1. Une matrice non négative est dite doublement stochastique si elle est stochastique en colonne et stochastique en ligne.

Le résultat le plus important pour les matrices non négatives est le suivant :

**Théorème 4.3 (Perron-Frobenius, voir [6])** *Soit  $A$  une matrice carrée non négative. Il existe une plus grande valeur propre de module de  $A$  qui est non négative et un vecteur propre non négatif qui lui correspond.*

Ce vecteur est généralement appelé le vecteur de Perron de la matrice non négative. Pour une matrice non négative rectangulaire, des résultats similaires peuvent être établis pour la plus grande valeur singulière et ses vecteurs singuliers correspondants.

Étant donné un sous-ensemble  $V \subset \mathbb{R}^{m \times n}$  et une matrice  $A \in \mathbb{R}^{m \times n}$ , l'élément le plus proche de  $V$  à  $A$  (par rapport à une distance) est appelé la projection de  $A$  sur  $V$ , notée  $P_V(A)$ . Lorsque le sous-ensemble  $V$  est l'orthant non négatif et que la distance considérée est la distance Euclidienne, la projection de  $A$  est notée  $[A]_+$  et définie comme suit :

$$[[A]_+]_{ij} = \begin{cases} A_{ij} & \text{Si } A_{ij} > 0 \\ 0 & \text{Sinon} \end{cases} = \max(0, A_{ij}).$$

## B. Optimisation

Avant de présenter quelques résultats de base sur l'optimisation, nous passons en revue le concept des ensembles convexes et des fonctions convexes.

### 3.4 Ensemble convexe et fonction convexe

**Définition 4.2 (Ensemble convexe)** *Un ensemble  $\Omega$  est dit convexe si et seulement si pour chaque  $u, v \in \Omega$ , on a*

$$\alpha u + (1 - \alpha)v \in \Omega, \quad \forall \alpha \in [0, 1].$$

Il est clair que les cônes convexes et le cône polyédrique sont, par construction, des ensembles convexes, ce qui implique que l'ensemble des  $m \times n$  matrices non négatives (l'orthant non négatif  $\mathbb{R}_+^{m \times n}$ ) est également un ensemble convexe. L'ensemble  $\mathbb{R}_+^{m \times n}$  est l'un des principaux objets utilisés dans cette thèse.

**Définition 4.3 (Fonction convexe)** *Une fonction  $f$  définie sur un ensemble convexe  $\Omega$  est dite convexe si, pour tout  $u, v \in \Omega$  et tout  $\alpha \in [0, 1]$ , alors :*

$$f(\alpha u + (1 - \alpha)v) \leq \alpha f(u) + (1 - \alpha)f(v).$$

*Si pour chaque  $\alpha \in (0, 1)$  et  $u \neq v$ , alors :*

$$f(\alpha u + (1 - \alpha)v) < \alpha f(u) + (1 - \alpha)f(v),$$

*donc la fonction est dite strictement convexe.*



Pour plus de détails sur les ensembles convexes et les fonctions convexes, voir [8].

## A.6 Conditions d'optimalité

Maintenant, nous résumons quelques résultats de base sur le problème d'optimisation

$$\min_{x \in \Omega} f(x),$$

où  $f$  est une fonction à valeur réelle sur l'ensemble  $\Omega \in \mathbb{R}^n$ .

Nous distinguons deux types de minimum.

**Définition 4.4 (Minimum local)** On dit qu'un point  $x^* \in \Omega$  est un minimum local de  $f$  sur  $\Omega$  s'il existe un voisinage ouvert  $N(x^*)$  de  $x^*$  tel que pour tout  $x \in N(x^*) \cap \Omega$ ,  $f(x) \geq f(x^*)$ . On dit que  $x^*$  est un minimum local strict si pour tout  $x \in N(x^*) \cap \Omega$  et  $x \neq x^*$ ,  $f(x) > f(x^*)$ .

**Définition 4.5 (Minimum global)** On dit qu'un point  $x^* \in \Omega$  est un minimum global de  $f$  sur  $\Omega$  si pour tout  $x \in \Omega$ ,  $f(x) \geq f(x^*)$ . On dit que  $x^*$  est un minimum local strict si pour tout  $x \in \Omega$  et  $x \neq x^*$ ,  $f(x) > f(x^*)$ .

Généralement, à moins que  $f$  ne possède des propriétés de convexité, la recherche du minimum global est une tâche très difficile qui nécessite une connaissance globale de la fonction  $f$ . D'autre part, la recherche de minima locaux ne nécessite que la connaissance du voisinage. Les conditions nécessaires aux minimum locaux peuvent également être facilement déduites par calcul différentiel. Cela explique pourquoi, dans notre problème de minimisation, nous allons essayer de trouver un minimum local, au lieu d'un minimum global.

Afin de mettre en place les conditions nécessaires satisfaites par les minimum locaux, l'idée de base est de considérer un point en utilisant le concept de directions réalisables. À partir d'un point  $x \in \Omega$ , un vecteur  $d$  est une direction réalisable s'il existe un  $\bar{\alpha} > 0$  tel que  $x + \alpha d \in \Omega$  pour tout  $\alpha \in [0, \bar{\alpha}]$ . Nous avons les conditions nécessaires de premier ordre suivantes :

### Proposition 4.1 ([28])

Soit  $\Omega$  un sous-ensemble de  $\mathbb{R}^n$  et  $f$  une fonction continuellement différentiable sur  $\Omega$ . Si  $x$  est un minimum local de  $f$  sur  $\Omega$ , alors pour chaque direction  $d$  en  $x$ , nous avons

$$(\nabla f(x^*))^T d \geq 0. \quad (4.5)$$

Inversement, chaque point qui vérifie la condition (4.5) est appelé un *point stationnaire*. Lorsque  $x$  est un point intérieur de  $\Omega$ , chaque vecteur  $d$  est une direction réalisable et (4.5) implique  $\nabla f(x^*) = 0$ .

Si  $\Omega$  est convexe, pour créer toutes les directions possibles, on peut utiliser les vecteurs  $d = x - x^*$ , pour chaque  $x \in \Omega$ . Cela générera toutes les directions possibles en  $x$ , car à partir de la convexité de  $\Omega$ , avoir

$$x^* + \alpha d = x^* + \alpha(x - x^*) = \alpha x + (1 - \alpha)x^* \in \Omega, \quad \forall \alpha \in [0, 1].$$

Par conséquent, un point  $x$  est dit être un *point stationnaire* s'il satisfait

$$(\nabla f(x^*))^T (x - x^*) \leq 0, \quad \forall x \in \Omega.$$

Dans le cas particulier où  $f$  et  $\Omega$  sont convexes, chaque minimum local est également un minimum global. De plus, l'ensemble de tous ces minimum est convexe. Pour plus de résultats et d'implications, voir [28].

### 3.5 Conditions de Karush-Kuhn-Tucker

Considérons le problème d'optimisation sous contrainte suivant :

$$\min_{\substack{h_i(x)=0 \\ g_j(x)\leq 0}} f(x),$$

où  $h_i(x) = 0$ , ( $i = 1, \dots, k$ ) sont  $k$  contraintes d'égalité et  $g_j(x) \geq 0$  ( $j = 1, \dots, m$ ) sont  $m$  contraintes d'inégalité. Ce qui suit est connu sous le nom de conditions nécessaires de Karush-Kuhn-Tucker (ou conditions KKT) :

**Proposition 4.2** ([7])

Soit  $x$  le minimum local du problème ci-dessus. Supposons que  $f$ ,  $h_i$  et  $g_j$  soient des fonctions continuellement différentiables de  $\mathbb{R}^n$  à  $\mathbb{R}$  et que  $\nabla h_i(x)$  et  $\nabla g_j(x)$  soient linéairement indépendants. Il existe alors des constantes uniques  $\mu_i$  ( $i = 1, \dots, k$ ) et  $\lambda_j$  ( $j = 1, \dots, m$ ), telles que :

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^k \mu_i \nabla h_i(x^*) + \sum_{j=1}^m \lambda_j \nabla g_j(x^*) &= 0, \\ \lambda_j &\geq 0, \quad j = 1, \dots, m \\ \lambda_j g_j(x^*) &= 0, \quad j = 1, \dots, m. \end{aligned}$$

Ce problème contrainté est souvent écrit dans la fonction de Lagrange associée :

$$\mathcal{L}(x, \mu_1, \dots, \mu_k, \lambda_1, \dots, \lambda_m) = f(x) + \sum_{i=1}^k \mu_i \nabla h_i(x) + \sum_{j=1}^m \lambda_j \nabla g_j(x)$$

où  $\mu_i$  ( $i = 1, \dots, k$ ) et  $\lambda_j$  ( $j = 1, \dots, m$ ) sont identiques à ceux des conditions KKT et sont appelés *multiplicateurs de Lagrange*.

### 3.6 Algorithme de descente de coordonnées sur un ensemble convexe

Nous décrivons une méthode pour résoudre le problème suivant

$$\min_{x \in \Omega} f(x),$$

où  $\Omega \subset \mathbb{R}^n$  est un produit cartésien d'ensembles convexes fermés  $\Omega_1, \Omega_2, \dots, \Omega_m$ , où  $\Omega_i \subset \mathbb{R}^{n_i}$  ( $i = 1, \dots, m$ ) et  $\sum_i n_i = n$ . La variable  $x$  est également partitionnée en conséquence comme

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix},$$

où  $x_i \in \Omega_i$ . L'algorithme 4 s'appelle l'algorithme de descente. Si nous supposons que l'étape 4.6 de l'algorithme 4 peut être résolue exactement et le minimum est uniquement atteint, alors nous avons le résultat suivant.

**Théorème 4.4 (Méthode de convergence des coordonnées par descente)** *Supposons que  $f$  soit une fonction continuellement différentiable sur l'ensemble  $\Omega$  décrit ci-dessus. De plus, supposons que pour chaque  $i$  et  $x \in \Omega$ , la solution de*

$$\min_{\xi \in \Omega_i} f(x_1, \dots, x_{i-1}, \xi, x_{i+1}, \dots, x_m)$$

*soit uniquement atteinte. Soit  $\{x^k\}$  un ensemble généré par l'algorithme 4. alors, chaque point limite est un point stationnaire.*

---

**Algorithm 4** Descent

---

1: Initialiser  $x_i$

2: **repeat**

3:     for  $i = 1 : m$      do

4:         Résoudre

$$x_i = \underset{\xi \in \Omega_i}{\operatorname{argmin}} f(x_1, \dots, x_{i-1}, \xi, x_{i+1}, \dots, x_m) \quad (4.6)$$

5:     end for

6: **until** Condition d'arrêt

---

## **Factorisation matricielle non négative pour la reconnaissance faciale**

**Résumé :** La reconnaissance faciale est un domaine de la vision par ordinateur consistant à reconnaître automatiquement une personne à partir d'une image de son visage. La reconnaissance de visage a de nombreuses applications en vidéosurveillance, biométrie, robotique, indexation d'images et de vidéos, recherche d'images par le contenu, etc. Plusieurs méthodes et approches sophistiquées ont été développées afin d'obtenir les meilleurs résultats de reconnaissance en utilisant certaines bases de données de visages spécifiques. Dans ce projet de fin d'études, nous abordons le problème de la reconnaissance de faces frontales capturées dans différentes conditions d'illumination et contenant des occlusions naturelles telles que des individus portant des lunettes de soleil et / ou des écharpes. L'approche adoptée est basée sur la technique de factorisation de matrice non négative (NMF), vue son habilité à traiter les occlusions naturelles. La décomposition en valeurs singulières (SVD) sera appliquée dans ce contexte pour une comparaison directe.

**Mots-Clés.** NMF, SVD, PCA, Reconnaissance faciale, Traitement d'image.

---

### **Nonnegative Matrix Factorisation for Facial Recognition**

**Abstract :** Facial recognition is a field of computer vision that automatically recognizes a person from an image of their face. Face recognition has many applications in video surveillance, biometrics, robotics, image and video indexing, image search by content, etc. Several sophisticated methods and approaches have been developed to obtain the best recognition results using some specific face databases. In this final project, we address the problem of recognition of frontal faces captured under different illumination conditions and containing natural occlusions such as individuals wearing sunglasses and / or scarves. The adopted approach is based on the non-negative matrix factorization technique (NMF), given its ability to deal with natural occlusions. The singular value decomposition (SVD) will be applied in this context for a direct comparison.

**Key Words.** NMF, SVD, PCA, Facial Recognition, Image Processing.

