

Remerciements

Je tiens à adresser mes remerciements les plus sincères aux personnes qui m'ont accompagné au cours de cette thèse.

Tout d'abord j'adresse mes sincères remerciements à mon directeur de thèse Monsieur le Professeur Kamel BOUKHETALA pour la confiance qu'il m'a accordée, pour m'avoir accueilli dans son équipe et pour m'avoir encouragé et soutenu tout au long de la thèse avec gentillesse et patience.

Je remercie tout particulièrement le grand avec qui j'ai l'honneur de collaborer Monsieur le HDR Mohamed Hanafi qui s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de cette thèse.

Je tiens à exprimer toute ma gratitude au Professeur A. TATACHAK pour avoir accepté d'être président pour le membre de jury de cet thèse. Je tiens également à remercier Madame le Professeur Ourida SADKI, Monsieur le Professeur Khaled KHALDI, Monsieur le Professeur Djamel HAMADOUCHE et Monsieur le MCA Mohamed REZAOUI de m'avoir fait l'honneur de participer au jury de ma thèse comme examinateurs.

Enfin, j'adresse mes plus sincères remerciements à mes parents, ma famille et tous mes amis, qui m'ont toujours soutenu et encouragé au cours de la réalisation de cette

Table des matières

thèse.

Merci à tous et à toutes.

Table des matières

Introduction	11
1 Préliminaires	17
1.1 Quelques notions d’algèbre linéaire	18
1.2 Optimisation	19
1.2.1 Gradient d’une fonction	20
1.2.2 Fonction convexe	20
1.2.3 Conditions d’optimalité de premier ordre : multiplicateurs de Karush-Kuhn-Tucker	20
1.3 L’approximation par factorisation à rang inférieur	21
1.3.1 Le problème d’Eckart-Young	21
1.4 La factorisation matricielle exacte et le rang usuel	25
1.4.1 Position du problème	25
1.4.2 Calcul de rang usuel	28
1.5 Méthode de classification par partitionnement k-means Clustering .	31
1.5.1 k-means par résolution d’un problème d’optimisation	32
2 La factorisation en matrices non négatives : Etat de l’art	35
2.1 Position du problème	36
2.2 La factorisation non négative exacte et le rang non négatif	39
2.2.1 Bornes pour le rang non négatif	42

Table des matières

2.2.2	La complexité de calcul	43
2.2.3	Interprétation géométrique pour le rang non négatif	46
2.3	Variations et extensions du NMF	49
2.3.1	La factorisation en matrices semi non négatives Semi-NMF	49
2.3.2	La factorisation convexe en matrices non négatives Convex-NMF	50
2.3.3	La Tri-factorisation en matrices non négatives Tri-NMF	50
2.3.4	La factorisation en matrices non négatives binaires BNMF	51
2.4	Les algorithmes de résolution du problème NMF	51
2.4.1	Les méthodes multiplicatives (MU)	52
2.4.2	La méthode des gradients projetés (PG)	53
2.4.3	La méthode des moindres carrés non négatives (NNLS)	54
2.5	L'équivalence entre le modèle NMF et le k-means Clustering	55
3	Rang semi non négatif des matrices réelles et sa connexion au rang usuel	62
3.1	Le rang semi non négatif	63
3.1.1	Position du problème	63
3.1.2	Existence et unicité	63
3.2	Bornes pour le rang semi non négatif	64
3.3	Les valeurs possibles du rang semi non négatif	67
3.4	Caractérisation des matrices pour les quelles le rang semi non négatif et le rang usuel coïncident	68
3.5	Interprétation géométrique pour la caractérisation des matrices vérifiant $r_s(X) = r_u(X)$	73
3.6	Calcul d'une factorisation semi non négative	75
4	Factorisation semi non négative : Partie algorithmique	77
4.1	Approximation matricielle à rang semi non négatif inférieur	78
4.2	Les algorithmes proposés pour le Semi-NMF	79
4.2.1	L'algorithme à mise à jour multiplicative	79
4.2.2	L'algorithme à mise à jour descente du gradient	83

Table des matières

4.3	Le problème d'initialisation pour les algorithmes Semi-NMF	84
4.4	Initialisation du Semi-NMF basée sur l'algorithme génétique	85
4.4.1	Aperçu générale sur les algorithmes génétiques	85
4.4.2	Un algorithme d'initialisation basé sur l'algorithme génétique pour la factorisation semi non-négative	92
	Conclusion	98
	Les publications	99
	Bibliographie	107

Table des figures

1.1	Les différentes étapes du k-means	33
3.1	Interprétation géométrique du théorème 3.3	75
4.1	La décroissance du critère selon la règle de mise à jour multiplicative .	80
4.2	Organigramme de l' AG	88
4.3	Roue de loterie biaisée	89
4.4	L'opérateur de croisement à 4 sites	91
4.5	Erreur d'approximation du Semi-NMF pour différentes initialisations sur les données Ionosphère, k=5	95
4.6	Erreur d'approximation du Semi-NMF pour différentes initialisations sur les données Ionosphère, k=10	95
4.7	Erreur d'approximation du Semi-NMF pour différentes initialisations sur les données Ionosphère, k=20	96

Liste des notations

\mathbb{R}	L'ensemble des nombres réels
\mathbb{R}_+	L'ensemble des nombres réels positifs
\mathbb{R}^n	L'ensemble des vecteurs réels de dimension n
$\mathbb{R}^{n \times p}$	L'ensemble des matrices réelles de dimension (n,p)
$\mathbb{R}_+^{n \times p}$	L'ensemble des matrices réelles ayant des valeurs positives de dimension (n,p)
$\{0,1\}^{n \times p}$	L'ensemble des matrices binaires de dimension (n,p)
X	Matrice
x	Vecteur
X^t	La transposée de la matrice X
X_{ij}	L'élément de la matrice X situé dans la $i^{\text{ème}}$ ligne et la $j^{\text{ème}}$ colonne
$X_{i:}$	$i^{\text{ème}}$ ligne de la matrice X
$X_{:j}$	$j^{\text{ème}}$ colonne de la matrice X
X^{-1}	La matrice inverse de la matrice X
$\dim(X)$	La dimension de la matrice X
$D(A/B)$	Une fonction de coût calculé entre les deux matrices A et B
e_i	Le $i^{\text{ème}}$ vecteur de la base canonique de \mathbb{R}^n
1_n	Vecteur dans \mathbb{R}^n où tous les éléments sont égaux à 1
I_n	La matrice identité d'ordre (n,n)
$\langle x, y \rangle$	Le produit scalaire entre x et y

Liste des notations

$\ \cdot\ _2$	La norme euclidienne
$\ \cdot\ _F$	La norme de Frobenius
TrX	La trace de la matrice X
$det(X)$	Le déterminant de la matrice X
$r_u(X)$	Le rang usuel de la matrice X
$r_+(X)$	Le rang non négatif de la matrice X
$r_s(X)$	Le rang semi non négatif de la matrice X
$A \otimes B$	Le produit d'Hadamard entre les deux matrices A et B
$\frac{A}{B}$	La division d'Hadamard entre A et B (Division terme par terme)
$[X]^+$	La partie positive de la matrice X (Projection sur l'orthant positif)
$[X]^-$	La partie négative de la matrice X

Liste des abréviations

NMF	La factorisation en matrices non négatives
SVD	La décomposition en valeurs singulières
PCA	L'analyse en composantes principales
NMF	La factorisation en matrices non négatives ou bien factorisation non négative
Semi-NMF	La factorisation semi non négative
Convex-NMF	La factorisation convexe non négative
Tri-NMF	La tri-factorisation non négative
Binary-NMF	La factorisation non négative binaire
Clustering	La classification non supervisée des données
k-means	La méthode k-moyenne de la classification non supervisée des données
MU	L'algorithme de mise à jour multiplicative
PG	L'algorithme du gradient projeté
NNLS	La méthode des moindres carrées non négatives
AG	Algorithme génétique

Introduction

Aujourd'hui avec le développement des **TIC**, des masses de données sont générées et stockées à temps réel. Cette massification de données nécessite des outils sophistiqués et adaptés à leurs traitement et analyse, tout en conservant l'information utile et de qualité. Ceci est connu aujourd'hui par ce qu'on appelle le Data mining (Fouille de données).

Le Data mining est un domaine qui est apparu avec l'explosion des quantités d'informations stockées, avec le progrès important des vitesses de traitement et des supports de stockage. Le Data mining vise à découvrir, dans les grandes quantités de données, les informations précieuses qui peuvent aider à comprendre ces données ou à prédire leurs comportements futures. Le Data mining utilise depuis son apparition plusieurs outils de statistiques et d'intelligence artificielle pour atteindre ses objectifs [78].

Si l'on dispose des données sous forme de matrices (Individus-Variables par exemple), il existe toute une série de techniques de compression, basées sur des factorisations matricielles particulières : la décomposition en valeurs singulières **SVD** [36], l'analyse en composantes indépendantes **ICA** [21], l'analyse en composantes principales **ACP** [43], etc. Malheureusement, ces techniques ne conservent pas nécessairement les propriétés des données étudiées. En particulier, pour des données positives (disant par la suite : non-négatives), ces techniques ne préservent pas cette caractéristique. Si on désire interpréter la décomposition, on doit pouvoir conserver cette propriété. Récemment, une

nouvelle technique a été proposée, il s'agit de la factorisation en matrices non négatives, dont nous utiliserons l'acronyme anglais **NMF** pour nonnegative matrix factorization [19]. Plusieurs méthodes ont été mises au point afin d'obtenir de bonnes factorisations permettant ainsi l'interprétation des facteurs.

Dans de nombreuses applications telles que le traitement d'images, la biologie et le text mining, l'analyse de données non-négatives est souvent très importante; la contrainte de non négativité appliqué sur les composantes principales et les coefficients peut améliorer le problème d'interprétabilité des résultats. En effet, la factorisation en matrices non-négatives est la technique proposé pour résoudre ce genre des problèmes [13].

En particulier, **NMF** vise à trouver les facteurs non-négatives qui peuvent être utilisés pour l'extraction des informations, la réduction de la dimensionnalité des données, l'élimination des informations redondantes, la classification des données et découvrir les modèles cachés derrière une série de vecteurs non-négatifs.

Le modèle **NMF** a été introduit pour la première fois par Lee et Sung (1999) pour décomposer des données bi-dimensionnelles non-négatives en une combinaison linéaire d'éléments dans un dictionnaire [53], et depuis ce temps, le **NMF** connaît une forte popularité dans les domaines de l'apprentissage et du signal/image, et puis dans d'autres domaines comme la biologie et la finance. Nous citons quelques domaines d'applications des cas d'exemples traités :

- **Traitement de l'image** : Représentation d'images de visages [53], [56], [76]. Classification d'images [38].
- **Analyse de textes** : Classification de documents [26]. Surveillance de messages électroniques [2]. Extraction de caractéristiques sémantiques dans des textes [53], et en spectroscopie [34].
- **Traitement du signal audio** : La transcription automatique sur partition [69],[65] , [4]. La séparation de sources [74], [17], [63]. L'alignement sur partition

Introduction

[22] ou encore la restauration audio [55].

- **Biologie** : Clustering de gènes impliqués dans le cancer [60]. Détection de l'activité neuronale pour les interfaces cerveau-machine [48].
- **Finance et économie** : La diversification de portefeuilles d'actions [28]. L'extraction des caractéristiques discriminantes dans les données de détresse financière [66] et aussi pour estimer le risque d'un portefeuille de crédit [73].

Le problème standard de la factorisation en matrices à coefficients positifs, sous la forme la plus générique possible, s'exprime comme suit :

Étant donné une matrice X de dimension $n \times p$ à coefficients non négatifs, i.e., positifs ou nuls, la factorisation en matrices non négatives, consiste à trouver une approximation :

$$X \approx AB = \widetilde{X} \quad (1)$$

telle que les matrices A et B soient à coefficients non-négatifs et de dimensions $n \times k$ et $k \times p$, respectivement. et où l'opérateur \approx désigne une « approximation » à définir. L'ordre du modèle, k , est habituellement choisi tel que $nk + kp \ll np$, de façon à réduire la dimension des données.

La factorisation 1 est généralement obtenue par résolution du problème de minimisation suivant :

$$\min_{A,B} D(X/AB) \quad \text{sous} \quad A \geq 0; B \geq 0. \quad (2)$$

où la notation $A \geq 0$ exprime la non négativité des coefficients de A , de même pour la matrice B .

$D(X/AB)$ est appelée une divergence, c'est une application dans \mathbb{R}_+ , telle que $\forall X \in \mathbb{R}_+^{n \times p}, \forall (A, B) \in \mathbb{R}_+^{n \times k} \times \mathbb{R}_+^{k \times p}$, on a

$$D(X/AB) = 0 \Leftrightarrow X = AB.$$

Une divergence ne vérifie pas nécessairement les propriétés de symétrie et d'inégalité triangulaire que doit vérifier une distance, mais une distance est bien un cas particulier

Introduction

de divergence.

La divergence utilisée en **NMF** est définie comme :

$$D(X/AB) = \sum_{i=1}^n \sum_{j=1}^p d(X_{ij}/[AB]_{ij}) \quad (3)$$

La fonction $d(x|y)$ est une mesure de similarité entre scalaires (parfois appelée fonction de coût), i.e., une fonction de $\mathbb{R}_+ \times \mathbb{R}_+$ dans \mathbb{R}_+ avec un unique minimum égal à zéro en $x = y$.

Une fonction de coût souvent considérée pour le **NMF** est la β – *divergence* [50], une famille continue de divergences dont l’expression est donnée par :

$$d_{\beta}(x|y) = \begin{cases} \frac{1}{\beta(\beta-1)}(x^{\beta} + (\beta-1)y^{\beta} - \beta xy^{\beta-1}); & \beta \in \mathbb{R} \setminus \{0, 1\} \\ x \log \frac{x}{y} - x + y; & \beta = 1 \\ \frac{x}{y} - \log \frac{x}{y} - 1; & \beta = 0 \end{cases}$$

La fonction de coût (β – *divergence*) prend comme cas particuliers : La distance euclidienne [54], la divergence de Kullback-Leibler [54], et la divergence d’Itakura-Saito [29] pour $\beta = 2, \beta = 1$ et $\beta = 0$ respectivement.

De nombreuses variantes algorithmiques développées ont été publiées et implémentées généralement en Matlab, ou en R traitent de la question, telles que :

- L’algorithme **NMF** standard avec mise à jour multiplicative [53]
- L’algorithme basé sur les moindres carrés alternés [18]
- L’algorithme de mise à jour descente du gradient [58].

Chacun de ces algorithmes peut par ailleurs être initialisé de différentes façons :

- Initialisation aléatoire de A et B

- Initialisation par la double décomposition en valeurs singulières non-négative (NNSVD) [7]
- Classification (k-means) des lignes ou des colonnes [77]
- parties positives de matrices issues d'une analyse en composantes indépendantes (ACI),...

Entre le choix de l'algorithme ou de l'une de ses variantes, le choix de l'initialisation, et afin de répondre aux besoins exigés dans les différentes applications, cela fait beaucoup d'options à comparer, et à tester. Comme toujours avec une nouvelle méthode et la pression de publication, de très nombreuses variantes apparaissent pour aboutir à des choix plus efficaces et consensuels d'options en fonction du type de données traitées. Comme objectif de notre thèse, nous avons choisi d'étudier l'une des variantes de la factorisation non négative, qui est la factorisation semi non négative (**Semi-NMF**).

Le modèle **Semi-NMF** a été proposé pour la première fois par Ding et al. en 2006 [23], et utilisé avec un grand succès dans plusieurs applications : Dans la segmentation des mouvements avec des données manquantes [62], dans la super résolution d'images [6], et dans l'hyperspectral unmixing [80].

Organisation de la thèse

Dans la suite de cette thèse, nous décrirons les travaux réalisés et les contributions apportées. Dans le chapitre 1, nous présenterons des notions préliminaires dont les outils de l'algèbre linéaire et de l'optimisation avec contraintes sont nécessaires pour aboutir à une bonne compréhension des contributions que nous effectuerons dans la suite. Le chapitre 2 vise à présenter les notions essentielles de la factorisation matricielle non négative, les différentes extensions du **NMF**, ainsi que les classes d'algorithmes permettant d'y apporter des solutions. Il se veut donc être un chapitre bibliographique. Dans le même chapitre, nous allons présenter l'extension **Semi-NMF** (la méthode qui

est l'objet de notre contribution). Nous présenterons ensuite dans le chapitre 3 notre approche exacte de factorisation Semi non négative où la notion du rang semi non négatif a permis de mettre en lumière plusieurs propriétés qui améliorent la compréhension d'une factorisation semi non négative.

Dans le chapitre 4, nous présentons en détails le problème d'approximation matricielle semi non négative, les algorithmes proposés, et nous attaquons le problème d'initialisation des algorithmes **Semi-NMF**. Enfin, nous montrerons que notre initialisation par l'algorithme génétique est particulièrement bien adaptée à l'algorithme **Semi-NMF** et elle donne des solutions plus optimales et avec moindres erreurs, en comparant avec les autres méthodes d'initialisation existaient dans la littérature. Nous terminons cette thèse par une conclusion générale qui ouvre plusieurs perspectives de nos travaux.

Chapitre **1**

Préliminaires

Sommaire

1.1	Quelques notions d’algèbre linéaire	18
1.2	Optimisation	19
1.2.1	Gradient d’une fonction	20
1.2.2	Fonction convexe	20
1.2.3	Conditions d’optimalité de premier ordre : multiplicateurs de Karush-Kuhn-Tucker	20
1.3	L’approximation par factorisation à rang inférieur	21
1.3.1	Le problème d’Eckart-Young	21
1.4	La factorisation matricielle exacte et le rang usuel	25
1.4.1	Position du problème	25
1.4.2	Calcul de rang usuel	28
1.5	Méthode de classification par partitionnement k-means Clustering	31
1.5.1	k-means par résolution d’un problème d’optimisation	32

Le but de ce chapitre est de donner un certain nombre de définitions et de concepts mathématiques de base qui sont nécessaires pour le fondement de cette thèse.

1.1 Quelques notions d'algèbre linéaire

1- Pour tout $n \in \mathbb{N}^*$, un vecteur $x \in \mathbb{R}^n$ sera noté $x = (x_1, x_2, \dots, x_n)^t$ (vecteur colonne).

2- Pour tous $x, y \in \mathbb{R}^n$, le produit scalaire de x et y est défini par :

$$\langle x, y \rangle = \sum_{i=1}^n x_i y_i$$

3- Pour tout $x \in \mathbb{R}^n$, la norme euclidienne de x est donnée par

$$\|x\|_2 = \sqrt{\langle x, x \rangle} = \sqrt{\sum_{i=1}^n x_i^2} \geq 0$$

4- Soit $X \in \mathbb{R}^{n \times p}$, la norme de Frobenius associée à la matrice X est définie comme :

$$\|X\|_F = \sqrt{\text{Tr}(X^t X)} = \sqrt{\text{Tr}(X X^t)} = \left(\sum_{i=1, j=1}^{n, p} |X_{ij}|^2 \right)^{\frac{1}{2}}$$

5- On appelle toute matrice $X^+ \in \mathbb{R}_+^{n \times p}$ **une matrice non négative**.

6- L'inverse d'une matrice X quand elle existe, est la matrice unique notée X^{-1} telle que :

$$X X^{-1} = X^{-1} X = I$$

elle existe si et seulement si $\det(A) \neq 0$.

Quelques propriétés

- * $(X^{-1})^t = (X^t)^{-1}$
- * $(XY)^{-1} = Y^{-1} X^{-1}$
- * $|X^{-1}| = \frac{1}{|X|}$

7- La trace d'une matrice

La trace de la matrice $X \in \mathbb{R}^{n \times n}$ est : $\text{Tr}(X) = \sum_{i=1}^n X_{ii}$

Propriétés

- * $Tr(X) = Tr(X^t)$
- * $Tr(X + Y) = Tr(X) + Tr(Y)$
- * $Tr(\alpha X) = \alpha Tr(X)$
- * $Tr(XY) = Tr(YX)$, en général $Tr(XY) \neq Tr(X)Tr(Y)$
- * $Tr(X) = 0$ si X est antisymétrique
- * Si X et X^t sont semblable alors : $Tr(X) = Tr(X^t)$

8- La somme des deux matrices A et B est donnée par :

$$[A + B]_{ij} = A_{ij} + B_{ij} \text{ pour } A \text{ et } B \text{ de même dimension } (n, p).$$

9- Le produit de deux matrices A et B est définie par : $[AB]_{ij} = \sum_{l=1}^k A_{il}B_{lj}$ avec A et B sont de dimensions (n, k) et (k, p) respectivement.

10- Le produit d'Hadamard entre les deux matrices A et B est défini par : $[A \otimes B]_{ij} = A_{ij}B_{ij}$, où A et B sont deux matrices de même dimension, on a les propriétés suivantes :

- * $A \otimes B = B \otimes A$
- * $(A \otimes B)^t = A^t \otimes B^t$

1.2 Optimisation

Le problème général de l'optimisation ou autrement dit de la programmation mathématique est le suivant :

Étant donné :

- 1- Une fonction objective $f : \mathbb{R}^n \rightarrow \mathbb{R}$ (fonction coût).
- 2- Un ensemble des contraintes $U \subset \mathbb{R}^n$, avec $U = \{x \in \mathbb{R}^n / h_i(x) = 0, \forall i \in E \text{ et } g_j(x) \geq 0, \forall j \in I\}$, où $h_i(x) = 0 (i = 1, 2, \dots, k)$ sont les contraintes d'égalité, et $g_j(x) \leq 0 (j = 1, 2, \dots, l)$ sont les contraintes d'inégalité.

On Cherche à minimiser f sur U , c'est à dire on cherche $x^* \in U$ tel que :

$$f(x^*) = \min_{x \in U} f(x) \tag{1.1}$$

ou équivalent

$$f(x^*) \leq f(x), \forall x \in U$$

1.2.1 Gradient d'une fonction

Soit $\Omega \subset \mathbb{R}^n$ un ensemble ouvert, et $f : \Omega \rightarrow \mathbb{R}$.

Définition 1.1

On dit que f est de classe C^m sur Ω ($f \in C^m(\Omega)$) si toutes les dérivées partielles jusqu'à l'ordre m existent et sont continues.

Définition 1.2

Pour tout $x \in \Omega$, on note (quand il existe) :

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)^t \in \mathbb{R}^n, \forall x \in \Omega,$$

le gradient de f en x .

1.2.2 Fonction convexe

Définition 1.3

Un ensemble $U \subset \mathbb{R}^n$ est dit convexe si $\forall x, y \in U$, on a $[x, y] \subset U$

Définition 1.4

Soit $U \subset \mathbb{R}^n$ est un ensemble convexe et $f : U \rightarrow \mathbb{R}$ une fonction. On dit que f est convexe sur U si

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y), \forall x, y \in U, \forall t \in [0, 1]$$

Théorème 1.1 (Caractérisation de la convexité avec " ∇ ")

Supposons que f est de classe C^1 . Alors f est convexe sur U si et seulement si :

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle, \forall x, y \in U$$

1.2.3 Conditions d'optimalité de premier ordre : multiplicateurs de Karush-Kuhn-Tucker

Dans la théorie de l'optimisation, les conditions de Karush-Kuhn-Tucker (**KKT**) présentent les conditions nécessaires pour les quelles une solution optimale existe pour un problème de programmation non linéaire.

Théorème 1.2

Soit x^* est un minimum local pour le problème (1.6).

Supposons que les fonctions f, h_i , et g_j sont continument différentiables, et $\nabla h_i(x^*)$ et $\nabla g_j(x^*)$ sont linéairement indépendants.

Alors ils existent des constantes $\mu_i (i = 1, 2, \dots, k)$ et $\lambda_j (j = 1, 2, \dots, l)$ qui sont uniques, telles que :

$$\nabla f(x^*) + \sum_{i=1}^k \mu_i \nabla h_i(x^*) + \sum_{j=1}^l \lambda_j \nabla g_j(x^*) = 0 \tag{1.2}$$

$$\lambda_j \geq 0; j = 1, 2, \dots, l \tag{1.3}$$

$$\lambda_j g_j(x^*) = 0; j = 1, 2, \dots, l \tag{1.4}$$

Ce problème avec contraintes peut s'écrire sous sa forme Lagrangienne associée :

$$L(x, \mu_1, \dots, \mu_k, \lambda_1, \dots, \lambda_l) = f(x) + \sum_{i=1}^k \mu_i h_i(x) + \sum_{j=1}^l \lambda_j g_j(x) \tag{1.5}$$

où $\mu_i (i = 1, 2, \dots, k)$ et $\lambda_j (j = 1, 2, \dots, l)$ sont les multiplicateurs de Lagrange.

1.3 L'approximation par factorisation à rang inférieur

1.3.1 Le problème d'Eckart-Young

Soit une matrice $X \in \mathbb{R}^{n \times p}$, et soit $r_u(X) = k$.

Existe-il une matrice \widetilde{X} , avec $r_u(\widetilde{X}) = \widetilde{k}$ telle que

$$F(\widetilde{X}) = \min_{\widetilde{k} \leq k} \|X - \widetilde{X}\|_F^2 \tag{1.6}$$

est atteint ?

Pour résoudre ce problème, il faut montrer d'abord l'existence de la matrice \widetilde{X} .

L'existence d'une solution pour le problème 1.6 est vérifiée si et seulement si l'ensemble des contraintes est un ensemble fermé.

Soit $E_k = \{X \in \mathbb{R}^{n \times p} / r_u(X) \leq k\}$.

E_k s'écrit aussi sous la forme :

$$E_k = \{XX^t \in \mathbb{R}^{n \times n} / r_u(X) \leq k\}$$

Si on montre que E_k est un ensemble fermé, alors, une solution du problème 1.6 existe, et pour cela, on va exploiter le lemme suivant :

Lemme 1.1

une matrice est de rang inférieur ou égal à k si et seulement si tous les déterminants d'ordres $k + 1$ sont nuls.

Preuve du lemme 1.1

[37]

Soit $B = XX^t$, on définit la matrice $\tilde{B} \in \mathbb{R}^{(k+1) \times (k+1)}$ telle que $\tilde{B} = [B_{ij}]_{i \in I, j \in J}$, avec I et J sont deux sous ensembles de $\{1, 2, \dots, n\}$ de cardinal $k + 1$.

Par l'application du lemme 1.1 sur la matrice B , on a :

$$E_k = \cap_{I,J} \{B \in \mathbb{R}^{n \times n} / \det(\tilde{B}) = 0\}.$$

Soient les deux applications f et g définies par :

$$\begin{aligned} f : \mathbb{R}^{n \times n} &\rightarrow \mathbb{R}^{(k+1) \times (k+1)} & \text{et} & & g : \mathbb{R}^{(k+1) \times (k+1)} &\rightarrow \mathbb{R} \\ B &\longmapsto \tilde{B} & & & \tilde{B} &\longmapsto \det(\tilde{B}) \end{aligned}$$

On a

$$E_k = \cap_{I,J} (g \circ f)^{-1}(\{0\}).$$

On sait que l'ensemble $\{0\}$ est un fermé, et aussi que f et g sont continues, alors E_k est une intersection d'images réciproques des ensembles fermés, par une application continue, donc E_k est un ensemble fermé.

Conclusion Une solution pour le problème 1.6 existe.

Résolution du problème

Pour résoudre le problème 1.6, Eckart et Young ont proposé d'utiliser le théorème de la décomposition en valeurs singulières, cité ci dessous :

Théorème 1.3 (Théorème de la décomposition en valeurs singulières)

Soit $X \in \mathbb{R}^{n \times p}$, avec $r_u(X) = k$ est le rang usuel de la matrice X (Voir la section suivante).

Alors

Il existent deux matrices orthogonales $U \in \mathbb{R}^{n \times n}$ et $V \in \mathbb{R}^{p \times p}$ (i.e., $U^t U = I_n$ et $V^t V = I_p$) telles que :

$$UXV^t = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \tag{1.7}$$

où D est une matrice diagonale d'éléments : $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_r > 0$, avec $r = \min(n, p)$.

les réels $\alpha_1, \alpha_2, \dots, \alpha_r$ sont appelés valeurs singulières de X , ils sont les racines carrées positives des valeurs propres non nulles de $X^t X$, les colonnes de U et de V , sont respectivement les vecteurs propres de XX^t , et de $X^t X$.

Montrons maintenant que la décomposition donnée par la SVD est une solution du problème 1.6.

Soit une matrice \widetilde{X} qui s'écrit comme :

$$\widetilde{X} = A_k M_k B_k^t$$

où M_k est une matrice d'ordre k , $A_k = \{a_1, a_2, \dots, a_k\} \in \mathbb{R}^{n \times k}$ avec $A_k^t A_k = I$, et $B_k = \{b_1, b_2, \dots, b_k\} \in \mathbb{R}^{p \times k}$ avec $B_k^t B_k = I$,

on obtient :

$$F(A_k, B_k) = \min_{A_k^t A_k = I, B_k^t B_k = I} \|X - A_k M_k B_k^t\|_F^2 \tag{1.8}$$

Soient $\varphi_k = \{A_k \in \mathbb{R}^{n \times k} / A_k^t A_k = I\}$ et $\psi_k = \{B_k \in \mathbb{R}^{p \times k} / B_k^t B_k = I\}$, alors le problème 1.6 sera de la forme :

$$F(A_k, B_k) = \min_{A_k \in \varphi_k, B_k \in \psi_k} \left\| X - A_k M_k B_k^t \right\|_F^2 \quad (1.9)$$

Pour tout couple de matrices $(A_k, B_k) \in (\varphi_k, \psi_k)$, on a :

$$\left\| X - A_k M_k B_k^t \right\|_F^2 = \|X\|_F^2 - \|M_k\|_F^2 \quad (1.10)$$

où $M_k = A_k^t X B_k$, alors,

$$\min_{A_k \in \varphi_k, B_k \in \psi_k} \left\| X - A_k M_k B_k^t \right\|_F^2$$

est équivalent à

$$\max \|M_k\|_F^2 = \max_{A_k \in \varphi_k, B_k \in \psi_k} \left\| A_k^t X B_k \right\|_F^2 \quad (1.11)$$

On utilisant 1.7, la solution du problème 1.8 est donnée par la somme de k premières valeurs singulières de la matrice X , i.e.,

$$\max_{A_k \in \varphi_k, B_k \in \psi_k} \left\| A_k^t X B_k \right\|_F^2 = \sum_{i=1}^k \alpha_i.$$

Conclusion

Une solution du problème $\max_{\tilde{k} \leq k} \left\| X - \tilde{X} \right\|^2$, est la matrice \tilde{X} donnée par :

$$\tilde{X} = \sum_{i=1}^k \alpha_i u_i v_i^t,$$

où

k est le rang de la matrice X ; α_i , u_i et v_i sont respectivement : les valeurs singulières, vecteurs singuliers à gauche, vecteurs singuliers à droite donnés par la **SVD** de la matrice X .

1.4 La factorisation matricielle exacte et le rang usuel

1.4.1 Position du problème

Étant donnée une matrice $X \in \mathbb{R}^{n \times p}$ telle que $n \leq p$, on s'intéresse au problème suivant : "Existent-ils deux matrices $A \in \mathbb{R}^{n \times k}$ et $B \in \mathbb{R}^{k \times p}$ telles que :

$$X = AB. \quad (1.12)$$

On va étudier l'existence et l'unicité d'une solution pour le problème 1.12, dont les inconnus sont les deux matrices A et B et le paramètre k .

Unicité

Si le problème 1.12 admet une solution (\tilde{A}, \tilde{B}) , alors, à partir de cette solution, on peut générer une infinité de solution de la manière suivante :

$$\hat{A} = \tilde{A}S \text{ et } \hat{B} = S^{-1}\tilde{B} \text{ pour toute matrice } S \text{ de dimension } (k, k) \text{ inversible.}$$

Par la suite, nous allons montrer l'existence d'un couple (\tilde{A}, \tilde{B}) solution du problème 1.12. Pour ce faire, plusieurs cas se présentent :

1^{er} cas : si $k = n$ ou $k = p$

$$\tilde{A} = I_k \text{ et } \tilde{B} = X \text{ ou bien } \tilde{A} = X \text{ et } \tilde{B} = I_k.$$

2^{ème} cas : $n < k < p$

On note par $O_{i,j}$ la matrice nulle d'ordre (i, j) , on a :

$X = AB$ où $A = [I_n, O_{n, k-n}] \in \mathbb{R}^{n \times k}$ et $B = \begin{bmatrix} X \\ O_{k-n, p} \end{bmatrix} \in \mathbb{R}^{k \times p}$ sont deux matrices blocs.

3^{ième} cas : $k > p$

L'interprétation de ce cas est identique à celle du cas précédent, i.e., la matrice X est écrite sous la forme :

$X = AB$ où $A \in \mathbb{R}^{n \times k}$ et $B \in \mathbb{R}^{k \times p}$ sont les deux matrices blocs : $A = [X, O_{n, k-p}]$ et $B = \begin{bmatrix} I_p \\ O_{k-p, p} \end{bmatrix}$ respectivement.

4^{ème} cas : $k < n$

Dans ce cas, on va définir la notion du rang usuel d'une matrice.

Soit $X \in \mathbb{R}^{n \times p}$, et soit J_n l'ensemble des entiers naturels défini comme :

$$J_n = \left\{ k \in \{1, 2, \dots, n-1\} / X = AB, X \in \mathbb{R}^{n \times p}, A \in \mathbb{R}^{n \times k}, B \in \mathbb{R}^{k \times p} \right\}.$$

Définition 1.5

On appelle *rang usuel* de la matrice X , l'entier minimal dans J_n , i.e.,

$$r_u(X) = \min_k \{J_n\}$$

5^{ème} cas : $r_u(X) < k < n$

Par définition du $r_u(X)$, ils existent deux matrices $\tilde{A} \in \mathbb{R}^{n \times r_u(X)}$ et $\tilde{B} \in \mathbb{R}^{r_u(X) \times p}$ telles que $X = \tilde{A}\tilde{B}$.

Alors le couple des matrices (\hat{A}, \hat{B}) avec $\hat{A} = [\tilde{A}, O_{1, k-r_u(X)}]$ et $\hat{B} = \begin{bmatrix} \tilde{B} \\ O_{1, k-r_u(X)} \end{bmatrix}$ forment aussi une solution pour le problème (1.1).

Interprétation géométrique du rang usuel

Soit une matrice $X \in \mathbb{R}^{n \times p}$ dont nous associons les systèmes $\{C_1, C_2, \dots, C_p\}$ et $\{L_1, L_2, \dots, L_n\}$ où L_i désigne la $i^{\text{ème}}$ ligne de X , et C_j est la $j^{\text{ème}}$ colonne de X .

Soit $L_X(\mathbb{R}^p)$ le sous-espace vectoriel engendré par la famille des vecteurs $\{L_1, L_2, \dots, L_n\}$, et $C_X(\mathbb{R}^n)$ le sous-espace vectoriel engendré par $\{C_1, C_2, \dots, C_p\}$.

Lemme 1.2

$$r_u(X) = \dim(C_X) = \dim(L_X).$$

Preuve du lemme 1.2

Soit k est la dimension de L_X , et $\{b_1, b_2, \dots, b_k\} \in \mathbb{R}^{n \times k}$ une base de L_X , et soit $B \in \mathbb{R}^{k \times p}$ la matrice dont les lignes sont $\{b_1^t, b_2^t, \dots, b_k^t\}$.

Préliminaires

Chaque ligne de X s'écrit comme une combinaison linéaire des lignes de B , i.e.,
 $\exists a_{ij} \in \mathbb{R}, 1 \leq i \leq n; 1 \leq j \leq k$ tels que :

$$L_1 = a_{1,1}b_1 + a_{1,2}b_2 + \dots + a_{1,k}b_k,$$

\vdots

$$L_i = a_{i,1}b_1 + a_{i,2}b_2 + \dots + a_{i,k}b_k,$$

\vdots

$$L_n = a_{n,1}b_1 + a_{n,2}b_2 + \dots + a_{n,k}b_k,$$

alors, la matrice X peut s'écrire comme

$$X = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{bmatrix} = \begin{bmatrix} a_{1,1}, a_{1,2}, \dots, a_{1,k} \\ a_{2,1}, a_{2,2}, \dots, a_{2,k} \\ \vdots \\ a_{n,1}, a_{n,2}, \dots, a_{n,k} \end{bmatrix} B = AB,$$

avec $A \in \mathbb{R}^{n \times k}$ et $B \in \mathbb{R}^{k \times p}$

Si on décompose B en colonnes, il vient

$$X = A[B_{:1}, B_{:2}, \dots, B_{:p}] = [AB_{:1}, AB_{:2}, \dots, AB_{:p}]. \quad (1.13)$$

L'équation 1.13 signifie que chaque colonne de X est une combinaison linéaire des colonnes de A , et donc

$$\dim(C_X) \leq k = \dim(L_X) \quad (1.14)$$

Pour la transposée de X , on a aussi :

$$\dim(C_{X^t}) \leq k = \dim(L_{X^t}), \quad (1.15)$$

mais $\dim(C_{X^t}) = \dim(L_X)$, et $\dim(L_{X^t}) = \dim(C_X)$,
donc, l'inégalité dans 1.15 s'écrit

$$\dim(L_X) \leq \dim(C_X). \quad (1.16)$$

Finalement, 1.14 et 1.16 nous donnent

$$\dim(L_X) = \dim(C_X).$$

Rappelons maintenant que k est le nombre de vecteurs qui forment une base pour les lignes de la matrice X , alors k est minimal, d'où k est le rang de la matrice X ou bien $k = r_u(X)$.

il est donc

$$r_u(X) = \dim(L_X) = \dim(C_X).$$

Corollaire 1.1

Pour $k < r_u(X)$, le problème 1.12 n'admet pas de solutions.

Preuve du Corollaire 1.1

Supposons qu'il existe un $k < r_u(X)$, pour lequel on a la factorisation $X = AB$, $A \in \mathbb{R}^{n \times k}$, $B \in \mathbb{R}^{k \times p}$, alors $\dim(C_X) \leq k$, mais $\dim(C_X) = r_u(X)$, d'où la contradiction.

1.4.2 Calcul de rang usuel

Le théorème suivant montre et décrit le procédé qui permet de calculer le rang usuel d'une matrice à valeurs réelles, ainsi que la factorisation associée. Pour plus de détails, on peut consulter [16].

Théorème 1.4 (Théorème de Wedderburn)

Soit $X \in \mathbb{R}^{n \times p}$, et soient a et b deux vecteurs dans \mathbb{R}^p et \mathbb{R}^n respectivement qui vérifient la propriété $w = b^t X a \neq 0$, alors

$$\text{rang}(X - w^{-1} X a b^t X) = r_u(X) - 1 \tag{1.17}$$

Il est important de remarquer que la matrice $w^{-1} X a b^t X$ est une matrice de rang 1, simplement parce que c'est le produit de deux vecteurs $X a$ et $b^t X$. L'intérêt de ce théorème réside dans le fait qu'on peut réduire le rang d'une matrice tout en substituant une matrice de rang 1 de la forme suivante : $w^{-1} X a b^t X$. Ceci étant possible pour tout choix de x et de y qui vérifie $b^t X a \neq 0$.

Une solution du problème 1.12 peut être construite itérativement en utilisant 1.17. par le procédé suivant :

$$\begin{cases} X_1 = X \\ X_{k+1} = X_k - w_k^{-1} X_k a_k b_k^t X_k \end{cases} \quad (1.18)$$

Donc, itérativement, on peut construire une factorisation de la matrice X , ceci nécessite k_0 itération, ce qui est le rang de la matrice X .

Soit k_0 le rang de X , et soient $\{a_1, a_2, \dots, a_{k_0}\}$ et $\{b_1, b_2, \dots, b_{k_0}\}$ des vecteurs tels que $w_k = b_k^t X_k a_k \neq 0$ pour tout $k \leq k_0$.

Proposition 1.1

$\forall 1 \leq k \leq k_0$

$$i) X_{k+1} = X - \sum_{i=1}^k w_i^{-1} X_i a_i b_i^t X_i \quad (1.19)$$

$$ii) r_u(X_{k+1}) = k_0 - k \quad (1.20)$$

Preuve de la proposition 1.1

Pour montrer cette proposition, on utilise le principe de récurrence, i.e.,

► Pour $k = 0$, il est claire que $X = X_1$

$$r_u(X_1) = k_0.$$

► Supposons que la proposition est vrais pour $k - 1$, il est facile de montrer que $X_k = X - \sum_{i=1}^{k-1} w_i^{-1} X_i a_i b_i^t X_i$ et $r_u(X_k) = k_0 - k + 1$, i.e.,

$$\begin{aligned} X_1 &= X & r_u(X_1) &= k_0, \\ X_2 &= X_1 - w_1^{-1} X_1 a_1 b_1^t X_1 & r_u(X_2) &= k_0 - 1, \\ X_k &= X_{k-1} - w_{k-1}^{-1} X_{k-1} a_{k-1} b_{k-1}^t X_{k-1} & r_u(X_k) &= k_0 - (k - 1), \end{aligned}$$

Par sommation, on obtient

$$X_k = X - \sum_{i=1}^{k-1} w_i^{-1} X_i a_i b_i^t X_i \quad \text{et} \quad r_u(X_k) = k_0 - k + 1.$$

Montrons que la proposition est vraie pour k ,

D'après la définition, on a

$$\begin{aligned} X_{k+1} &= X_k - w_k^{-1} X_k a_k b_k^t X_k \\ &= X - \sum_{i=1}^{k-1} w_i^{-1} X_i a_i b_i^t X_i - w_k^{-1} X_k a_k b_k^t X_k \\ &= X - \sum_{i=1}^k w_i^{-1} X_i a_i b_i^t X_i. \end{aligned}$$

Reste à montrer que $r_u(X_{k+1}) = k_0 - k$.

D'après le théorème de Wedderburn, on a

$$r_u(X_{k+1}) = r_u(X_k) - 1 = k_0 - k.$$

Corollaire 1.2

$$i) X_{k_0+1} = 0 \tag{1.21}$$

$$ii) X = \sum_{i=1}^{k_0} w_i^{-1} X_i a_i b_i^t X_i \tag{1.22}$$

Preuve du Corollaire 1.2

Par l'application directe de la proposition 1.1 pour $k = k_0$, il suffit de remarquer que

$$\text{rang}(X_{k_0+1}) = k_0 - k_0 = 0,$$

Ceci implique que $X_{k_0+1} = 0$, car une matrice de rang nul est nulle.

D'où,

$$X = \sum_{i=1}^{k_0} w_i^{-1} X_i a_i b_i^t X_i.$$

L'intérêt du corollaire réside dans le fait que le procédé mentionné ci-dessus construit une solution pour le problème 1.12.

L'équation 1.21 montre que la suite X_k est nulle à partir de k_0 , autrement dit, elle est convergente.

Le calcul effectif de X_k s'effectue par l'algorithme suivant :

Algorithme *i)* Initialiser : $X_1 = X, k_0 = 0, S = 1, k = 1$;

ii) Tant que $S \neq 0$

Choisir a_k et b_k ;

Calculer $w_k = b_k^t X a_k$;

Tant que $w_k = 0$; aller à *ii)*

iii) Calculer $X_{k+1} = X_k - w_k^{-1} X_k a_k b_k^t X_k$;

$S = \sum_{i,j} a_{i,j}$ "la somme des éléments de X "

iv) Si $S = 0$, alors : fin algorithme

Sinon, aller à *ii)*.

v) $k_0 \leftarrow$ nombre d'itérations

Fin Algorithme.

De plus, si on pose les deux matrices $\Phi = [\varphi_1, \varphi_2, \dots, \varphi_{k_0}]$ et $\Psi = [\psi_1, \psi_2, \dots, \psi_{k_0}]$ telles que

$\varphi_k = X_k a_k$ et $\psi_k = X_k^t b_k$, alors

$$X = \sum_{k=1}^{k_0} w_k^{-1} \varphi_k \psi_k^t$$

ce qui a l'écriture matricielle :

$$X = \Phi \Omega^{-1} \Psi^t \tag{1.23}$$

où Ω est une matrice diagonale de dimension (k, k) dont les éléments diagonaux sont les w_i pour $i \in \{1, 2, \dots, k\}$.

Remarque La factorisation 1.23 n'est pas unique à cause des différents choix des vecteurs a_k et b_k .

1.5 Méthode de classification par partitionnement k-means Clustering

Le *k-means*, ou bien le **K-moyennes**, est un algorithme qui partitionne les données d'un tableau $X \in \mathbb{R}^{n \times p}$ en k classes (appelées aussi clusters), (k étant un nombre

entier donné) dont la base est la minimisation de l'erreur quadratique entre la moyenne empirique d'une classe et les autres points qui appartiennent à cette classe, cette erreur quadratique est défini par :

$$EQ_{k\text{-means}}(m, C) = \sum_{i=1}^k \sum_{x_j \in C_i} d^2(x_j, m_i)$$

dont les paramètres sont : "m", l'ensemble des barycentres (centroïdes) des classes, et "C" est l'ensemble des classes, "d" est une distance ou métrique (dans le k-means standard, la distance utilisée est la distance euclidienne). On peut trouver une revue sur le k-means dans [45].

L'algorithme des K-moyennes choisit arbitrairement des centres initiaux et améliore la classification obtenue de manière itérative en alternant deux étapes fondamentales jusqu'à stabilisation. Chacune des étapes consiste à fixer l'un des paramètres de la fonction d'évaluation puis à estimer la valeur optimale de l'autre.

Algorithme k-means

- 1- Choisir k points parmi les données comme centres initiaux des classes.
- 2- Calculer les distances entre chaque point et chaque centre, et affecter chaque point (individu) au centre le plus proche, ce qui définit k classes.
- 3- Remplacer les k centres par les barycentres des k classes.
- 4- Revenir à l'étape 2, et poursuivre les itérations jusqu'à voir que les centres restent suffisamment stables, donc arrêt.

Les étapes de l'algorithme $k - means$ sont bien illustrer dans la figure 1.1.

1.5.1 k-means par résolution d'un problème d'optimisation

Problème

Etant donnée un ensemble de vecteurs $X = \{x_1, x_2, \dots, x_p\}; x_i \in \mathbb{R}^n$.

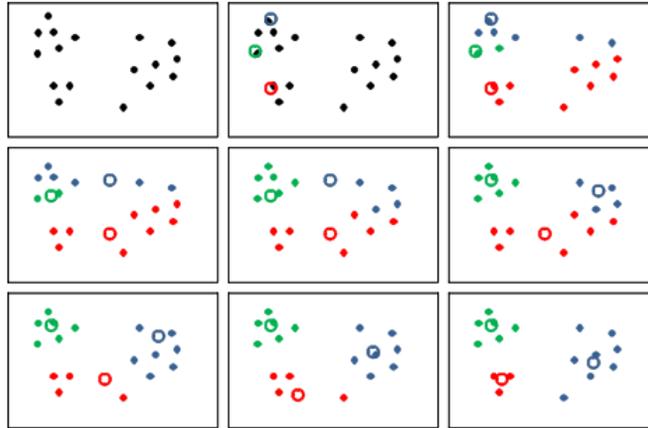


FIGURE 1.1 – Les différentes étapes du k-means

Trouver une partition $P = \{C_1, C_2, \dots, C_k\}$ où C_j sont des sous ensembles de X , qui minimise la fonction suivante :

$$I = \sum_{j=1}^k \sum_{x_i \in C_j} \|x_i - m_j\|^2 \quad (1.24)$$

où $m_j = \sum_{x_i \in C_j} \frac{x_i}{n_j}$ est le centroïde de la classe C_j , n_j est le nombre de points dans la classe C_j .

La fonction objective I représente l'inertie globale, i.e.,

$I = I_1 + I_2 + \dots + I_k$ avec I_j est la somme des distances des points x_i^j au centroïde m_j .

Il est important de remarquer que l'inconnu de la fonction défini en (1), est une partition $P = \{C_1, C_2, \dots, C_k\}$.

La procédure de l'algorithme k-means est la suivante :

Supposons k classes avec des centroïdes $\{m_1, m_2, \dots, m_k\} \in \mathbb{R}^{n \times k}$, la fonction objective I peut s'écrire sous la forme :

$$I = \sum_{j=1}^k \sum_{i=1}^n a_{ij} \|x_i - m_j\|^2 \quad (1.25)$$

Préliminaires

$$\text{où } a_{ij} = \begin{cases} 1 & \text{si } x_i \in C_j \\ 0 & \text{sinon} \end{cases} .$$

On appelle les a_{ij} des Labels ou bien indicateurs d'appartenance aux classes.

L'algorithme se déroule dans trois étapes principales :

i) Initialiser m_1, m_2, \dots, m_k en générant aléatoirement k points dans l'espace des données.

ii) Pour $m = \{m_1, m_2, \dots, m_k\}$ fixé, choisir a tel que

$$a_{ij} = \begin{cases} 1 & \text{si } j = \arg \min_l \|x_i - m_l\|^2 \\ 0 & \text{sinon} \end{cases} \quad (\text{signifie que : affecter } x_i \text{ vers la classe de}$$

centroïde le plus proche de x_i).

iii) Calculer les nouveaux centroïdes optimaux.

La factorisation en matrices non négatives : Etat de l'art

Sommaire

2.1	Position du problème	36
2.2	La factorisation non négative exacte et le rang non négatif	39
2.2.1	Bornes pour le rang non négatif	42
2.2.2	La complexité de calcul	43
2.2.3	Interprétation géométrique pour le rang non négatif	46
2.3	Variations et extensions du NMF	49
2.3.1	La factorisation en matrices semi non négatives Semi-NMF	49
2.3.2	La factorisation convexe en matrices non négatives Convex-NMF	50
2.3.3	La Tri-factorisation en matrices non négatives Tri-NMF . .	50
2.3.4	La factorisation en matrices non négatives binaires BNMF .	51
2.4	Les algorithmes de résolution du problème NMF	51
2.4.1	Les méthodes multiplicatives (MU)	52
2.4.2	La méthode des gradients projetés (PG)	53
2.4.3	La méthode des moindres carrés non négatives (NNLS) . .	54
2.5	L'équivalence entre le modèle NMF et le k-means Clustering	55

Ce chapitre vise à présenter la méthode de la factorisation en matrices non négatives, il s'agit de la formulation du problème **NMF**. On pourrait argumenter que le non Factorisation en matrices non négatives est mal dit dans certains cas, et que l'approximation par matrices non négatives devrait être utilisé à la place. Le terme "Factorisation" peut être compris comme une décomposition exacte telle que la décomposition de Cholesky, la décomposition LU,...etc, où la matrice d'entrée est exactement factorisée comme un produit d'autres matrices. Cependant, "la factorisation en matrices non négatives" est devenue tellement populaire qu'elle représente le problème de l'approximation d'une matrice non négative par un produit de deux matrices non négatives. Nous continuons à utiliser ce terme et nous référons à la factorisation exacte en matrices non négatives pour le cas exacte.

2.1 Position du problème

Le problème standard de la factorisation en matrices non négatives, sous la forme la plus générique possible, s'exprime comme suit :

Étant donné une matrice X de dimension $n \times p$ à coefficients non négatives, i.e., $X \in \mathbb{R}_+^{n \times p}$, la factorisation en matrices non négatives consiste à trouver une approximation :

$$X \approx AB = \widetilde{X}, \quad (2.1)$$

telle que les matrices A et B soient à coefficients non négatives et de dimensions $n \times k$ et $k \times p$ respectivement (i.e., $A \in \mathbb{R}_+^{n \times k}$ et $B \in \mathbb{R}_+^{k \times p}$. et où l'opérateur \approx désigne une « approximation » à définir. L'ordre du modèle, k , est habituellement choisi tel que $nk + kp \ll np$, de façon à réduire la dimension des données.

Il existe plusieurs façons pour quantifier la différence entre la matrice de données X et le modèle matriciel AB . La mesure la plus utilisée est la norme de Frobenius :

$$D(X/AB) = \|X - AB\|_F^2 = \sum_{i=1}^n \sum_{j=1}^p (X_{ij} - [AB]_{ij})^2 \quad (2.2)$$

La factorisation 2.1 est généralement obtenue par résolution du problème de minimisation suivant :

$$\min_{A,B} D(X/AB) \quad \text{sous} \quad A \geq 0; B \geq 0, \quad (2.3)$$

où la notation $A \geq 0$ exprime la non négativité des coefficients de la matrice M . On écrit le problème 2.2 comme problème d'optimisation non linéaire :

$$\min_{-A \leq 0, -B \leq 0} \|X - AB\|_F^2.$$

La fonction de Lagrange associée à ce problème d'optimisation est :

$$L(U, V, \alpha, \beta) = \|X - AB\|_F^2 - \alpha \otimes A - \beta \otimes B^t,$$

où α et β sont deux matrices de même dimension que A et B^t respectivement qui contiennent les multiplicateurs de Lagrange associés avec les contraintes de non négativité : $A \geq 0$ et $B \geq 0$. Les conditions d'optimalité **KKT** pour le problème de factorisation non négative disent que si (\tilde{A}, \tilde{B}) est un minimum local, alors il existent $\alpha_{ij} \geq 0$ et $\beta_{ij} \geq 0, (1 \leq i \leq n, 1 \leq j \leq p)$ tels que :

$$A \geq 0 \quad ; \quad B \geq 0 \quad (2.4)$$

$$\nabla L_A = 0 \quad ; \quad \nabla L_B = 0 \quad (2.5)$$

$$\alpha \otimes A = 0 \quad ; \quad \beta \otimes B = 0. \quad (2.6)$$

En développant 2.5, on a :

$$-2XB^t + 2ABB^t - \alpha = 0 \quad ; \quad -2X^tA + 2B^tA^tA - \beta = 0$$

ou bien

$$\alpha = ABB^t - XB^t \quad ; \quad \beta = B^tA^tA - X^tA \quad (2.7)$$

En combinant 2.7 avec $\alpha_{ij} \geq 0, \beta_{ij} \geq 0$ et 2.6, on obtient les conditions suivantes :

$$A \geq 0 \quad ; \quad B \geq 0 \quad (2.8)$$

$$\nabla D_A = ABB^t - XB^t \geq 0 \quad ; \quad \nabla D_B = B^tA^tA - X^tA \geq 0 \quad (2.9)$$

$$A \otimes (ABB^t - XB^t) = 0 \quad ; \quad B \otimes (B^tA^tA - X^tA) = 0 \quad (2.10)$$

Comme la distance de Frobenius n'est pas une fonction convexe par rapport aux deux variables A et B en même temps. Ceci est implicite en raison de l'existence de points critiques et de maxima. On appelle donc tous les points qui satisfont aux conditions ci-dessus, les points stationnaires.

Définition 2.1

On appelle un point stationnaire pour le problème **NMF** le couple des matrices (A, B) qui satisfait les conditions **KKT** présenté dans 2.8, 2.9, 2.10.

Question Si un point stationnaire (A, B) existe pour le problème **NMF**, est il unique ? Deux approximations (A_1, B_1) et (A_2, B_2) sont dites équivalentes si elles ont le même produit, i.e., $A_1B_1 = A_2B_2$.

En effet, d'après le point stationnaire (A, B) , on peut trouver une matrice S non négative et inversible telle que :

$$\tilde{A} = AS \geq 0 \text{ et } \tilde{B} = S^{-1}B \geq 0.$$

En vérifiant les conditions d'optimalité **KKT** pour le couple des matrices (\tilde{A}, \tilde{B}) , on a :

$$\begin{aligned} \tilde{A} \geq 0 & \quad ; \quad \tilde{B} \geq 0 \\ (ABB^t - XB^t)S^{-1} \geq 0 & \quad ; \quad S(B^tA^tA - X^tA) \geq 0 \\ (AS) \otimes [(ABB^t - XB^t)S^{-1}] = 0 & \quad ; \quad (S^{-1}B)[S(B^tA^tA - X^tA)] = 0 \end{aligned}$$

En particulier, si S est une matrice de permutation, ces conditions sont facilement vérifiées. Dans ce cas, toutes les colonnes de A et B sont retenus dans \tilde{A} et \tilde{B} , mais dans un ordre permuté, et ceci va générer essentiellement la même solution.

Pour une matrice non négative S quelconque, l'étude de l'unicité du point stationnaire n'est plus facile, elle peut être traitée dans des cas particuliers [41].

2.2 La factorisation non négative exacte et le rang non négatif

Dans cette section, nous allons examiner brièvement le problème de la factorisation exacte en matrices non négatives **Exact-NMF**, i.e., le cas où la valeur de la fonction objective 2.2 doit être égale à zéro. Dans [20], un bon traitement du problème est mis en oeuvre.

Position du problème Soit une matrice $X \in \mathbb{R}_+^{n \times p}$.

Le problème est de chercher deux matrices A et B , telles que :

$$X = AB, A \in \mathbb{R}_+^{n \times r}, B \in \mathbb{R}_+^{r \times p} \quad (2.11)$$

En plus de la recherche des matrices A et B , le paramètre r est un inconnu important du problème 2.11.

Existence et unicité d'une solution pour le problème 2.11

Existence Pour montrer l'existence d'une solution (\tilde{A}, \tilde{B}) pour le problème 2.11, plusieurs cas seront envisagés par la suite.

1^{er} cas : si $r = n$ ou $r = p$

on a :

$X = I_r X$ si $r = n$ et $X = X I_r$ si $r = p$ sont deux factorisations non négatives possibles.

2^{ème} cas : si $n < r < p$

Si on note par $O_{i,j}$ la matrice dans $\mathbb{R}^{i \times j}$ dont tout les valeurs sont nulles, on a la factorisation :

où $X = \tilde{A}\tilde{B}$
 où $\tilde{A} = [I_n, O_{n,r-n}]$ et $\tilde{B}_{r,p} = \begin{bmatrix} X \\ O_{r-n,p} \end{bmatrix}$ sont deux matrices blocs.

3^{ème} cas : si $r > p$

On a la factorisation :

$$X = \hat{A}\hat{B}$$

où \hat{A} et \hat{B} sont les deux matrices blocs définies par :

$$\hat{A} = [X, O_{n,r-p}] \text{ et } \hat{B} = \begin{bmatrix} I_p \\ O_{r-p,p} \end{bmatrix}$$

4^{ème} cas : sinon $r < n$

Soit $J_n = \{k \in \{1, 2, \dots, n-1\} \mid \text{le problème 2.11 possède une solution}\}$

Définition 2.2

$r_+(X) = \min_{r \in J_n} \{k\}$ est dit le rang non négatif de la matrice X .

Dans ce cas, nous avons le résultat particulier présenté dans la proposition suivantes :

Proposition 2.1

Si une matrice non négative est de rang 1, alors son rang non négatif est égal à 1 [20].

Preuve de la proposition 2.1

Soit X une matrice non négative de rang 1, telle qu'elle admet la décomposition $X = ab^t$, avec $a \in \mathbb{R}^n$ et $b \in \mathbb{R}^p$.

Pour $1 \leq i \leq n, 1 \leq j \leq p$, un élément de la matrice X s'écrit comme $x_{ij} = a_i b_j$, où a_i est le $i^{\text{ème}}$ élément de vecteur a , et b_j est le $j^{\text{ème}}$ élément de vecteur b ,

comme $x_{ij} \geq 0$, alors $|x_{ij}| = x_{ij}$,

d'où, $x_{ij} = |a_i| |b_j|$ pour tout $1 \leq i \leq n, 1 \leq j \leq p$.

alors, il existe une factorisation pour X sous la forme $\tilde{a}\tilde{b}^t$ dont les matrices \tilde{a} et \tilde{b} sont non négatives et d'ordre $(n, 1)$ et $(p, 1)$ respectivement.

d'où

$$r_+(X) = 1.$$

Remarques

i) Pour une matrice $X \in \mathbb{R}_+^{n \times p}$, $r_+(X)$ est le nombre minimal de matrices non négatives et de rang 1 (ou de rang non négatif égal à 1), telles que la somme de ces matrices égale à X , i.e.

$$X = \sum_{i=1}^{r_+(X)} \widetilde{X}_i,$$

telle que pour $1 \leq i \leq r_+(X)$:

$$r_+(\widetilde{X}_i) = r(\widetilde{X}_i) = 1.$$

ii) La définition du rang non négatif d'une matrice signifie que pour $r < r_+(X)$, le problème 2.11 n'admet pas de solutions.

5^{ème} cas : pour $r_+ < r < n$

Par définition de $r_+(X)$, il existe deux matrices $\widetilde{A} \in \mathbb{R}_+^{n \times r_u(X)}$ et $\widetilde{B} \in \mathbb{R}_+^{r_u(X) \times p}$ telles que $X = \widetilde{A}\widetilde{B}$.

Alors, les matrices $\widehat{A} = [\widetilde{A}, O_{1, r-r_+(X)}]$ et $\widehat{B} = \begin{bmatrix} \widetilde{B} \\ O_{1, k-r_+(X)} \end{bmatrix}$ forment aussi une solution pour le problème 2.11.

Non unicité

Définition 2.3 (Matrice monomiale)

On appelle matrice monomiale une matrice où chaque ligne et chaque colonne ne contient qu'un seul élément non nul.

Si cet élément est égal à 1, alors la matrice est dite de permutation.

Supposons qu'il existe une solution $(\widetilde{A}, \widetilde{B})$, pour le problème 2.11, alors on peut obtenir une autre solution de la forme :

$$\widehat{A} = \widetilde{A}S \quad \text{et} \quad \widehat{B} = S^{-1}\widetilde{B}$$

où S est une matrice de permutation d'ordre r .

Une matrice de permutation S peut être écrite sous la forme $S = ND$, où N est une matrice monomiale et D diagonale.

Donc, on peut caractériser S par une infinité de façon, alors le problème (2.11) admet une infinité de solutions.

2.2.1 Bornes pour le rang non négatif

Pour le rang non négatif, les résultats suivants sont bien connus et peuvent être trouvés dans [20].

Lemme 2.1

Soit une matrice $X \in \mathbb{R}_+^{n \times p}$, dont le rang non négatif est $r_+(X)$, alors

$$r_u(X) \leq r_+(X) \leq \min(n, p) \quad (2.12)$$

Preuve du lemme 2.1

Soit deux matrice $A \in \mathbb{R}_+^{n \times r_+(X)}$ et $B \in \mathbb{R}_+^{r_+(X) \times p}$ telles que $X = AB$, par définition du rang de la matrice X , on a

$$r_u(X) \leq r_+(X).$$

Pour vérifier la deuxième inégalité, on suppose que $n \leq p$, une factorisation évidente pour la matrice X est :

$$X = I_n X$$

Alors, nous avons :

$$r_+(X) \leq n.$$

On montre le cas où $p \leq n$ de la même façon.

2.2.2 La complexité de calcul

Relation entre le rang non négatif et le rang usuel d'une matrice

Cette partie nous permet d'illustrer les exemples suivants :

Exemple 1 Soit la matrice X_0 , définie par :

$$X_0 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Soient C_1, C_2, C_3 , et C_4 les colonnes de la matrice X_0 .

Remarquons que C_2, C_3 , et C_4 forment une famille libre de \mathbb{R}^4 , et en plus, C_1 s'écrit comme une combinaison linéaire de cette famille de vecteurs, d'où, on conclut que le rang de la matrice X_0 égal à 3.

Calculons son rang non négatif $r_+(X_0)$,

d'après 2.12, on a $r_+(X_0) = 3$ ou bien $r_+(X_0) = 4$.

Supposons que $r_+(X_0) = 3$, essayant d'écrire une colonne de la matrice X_0 comme une combinaison linéaire positive des trois autres colonnes.

i.e., soient $\lambda_1, \lambda_2, \lambda_3 \geq 0$ tels que

$$C_4 = \lambda_1 C_1 + \lambda_2 C_2 + \lambda_3 C_3.$$

En résolvant ce système, on trouve $\begin{cases} \lambda_2 = 1 \\ \lambda_3 = 1 \\ \lambda_1 = -\lambda_2 \end{cases}$,

ce qui contredit l'hypothèse de la positivité des λ_i .

D'après les inégalités 2.12 du lemme 2.1, on obtient :

$$r_+(X_0) = 4.$$

Exemple 2

Soit la matrice X_1 , définie par :

$$X_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

X_1 est une matrice du rang égal à 3.

D'après le lemme 1, on a

$$r_+(X_1) = 3 = r_u(X_1).$$

Pour étudier la relation entre le rang non négatif et le rang usuel, on va définir les deux ensembles J_1 , et J_2 , tels que

$$J_1 = \{X \in \mathbb{R}^{n \times p} / r_u(X) < r_+(X)\} \quad (2.13)$$

$$J_2 = \{X \in \mathbb{R}^{n \times p} / r_u(X) = r_+(X)\} \quad (2.14)$$

Caractérisation de l'ensemble J_1 Pour l'ensemble J_1 , on a le théorème définit ci-dessous pour le cas particulier des matrices de distance euclidienne :

Définition 2.4 (matrice de distance euclidienne)

Soient $a_1, a_2, \dots, a_n \in \mathbb{R}$.

On appelle une matrice de distance euclidienne la matrice D définit par $D = [d_{ij}]_{1 \leq i, j \leq n}$, avec $d_{ij} = (a_i - a_j)^2$.

Théorème 2.1

Pour n vecteurs réels dans \mathbb{R}^q , notées : a_1, a_2, \dots, a_n .

Soit la matrice $D = [d_{ij}]_{1 \leq i, j \leq n}$, avec $d_{ij} = \|a_i - a_j\|^2$, on a

$$r_u(D) = q + 2 \tag{2.15}$$

$$r_+(D) = n \tag{2.16}$$

Pour tout $p > 1$, on a

$$r_+(D) - r_u(D) \geq p \tag{2.17}$$

Preuve du théorème 2.1

La démonstration de ce théorème fait l'objet du papier [59].

Caractérisation de l'ensemble J_2

Soit $J_2 = \{X \in \mathbb{R}_+^{n \times p} / r_u(X) = r_+(X)\}$,

Si cet ensemble existe, alors la résolution du problème 2.11 est obtenue d'une manière très simple, il suffit de calculer le rang simple par une approche de théorème de Wedderburn dans le cas non négatif.

Cette approche est introduite dans [27], l'idée est basée sur la soustraction d'une matrice non négative d'après une matrice donnée non négative, et après un nombre spécifique d'itérations, on s'intéresse à la dernière matrice.

Soit $X \in \mathbb{R}_+^{n \times p}$

On pose le problème suivant :

$X_1 = X$, et pour $k \geq 2$

$$\max_{(a_k, b_k) \in \mathbb{R}^{p \times n}} \min_{X_k a_k \geq 0, b_k^t X_k \geq 0, b_k^t X_k a_k = 1} (X_k - X_k a_k b_k^t X_k) \tag{2.18}$$

Si une solution pour le problème 2.18 existe, alors on calculons les itérations : $X_{k+1} = X_k - X_k a_k b_k^t X_k$,

on obtient après k_0 itérations : $X_{k_0+1} = 0$, dans ce cas

$$r_+(X) = r_u(X) = k_0.$$

Pour plus de détails, voir [27].

2.2.3 Interprétation géométrique pour le rang non négatif

Pour étudier l'interprétation géométrique du rang non négatif, on a le théorème de Krein Millman qui nécessite quelques rappels additionnels de l'analyse convexe :

- 1) Une intersection d'ensembles convexes dans E est un convexe de E .
- 2) Soit $f : E \rightarrow F$ une application linéaire. L'image directe $f(C)$ (Respectivement l'image réciproque $f^{-1}(C)$) d'un convexe C de E (respectivement de F) par f est un convexe.

Définition 2.5

Soit C un convexe de E , un point $x \in C$ est dit extrémal s'il n'existe pas $]y_1, y_2[\subset C$, tel que $x \in [y_1, y_2]$

De façon équivalente, un point $x \in C$ est extrémal si $C \setminus \{x\}$ est convexe.

Définition 2.6

Soit $A \subset E$.

On appelle enveloppe convexe de A , l'intersection de tous les convexes de E qui contiennent A .

On note par $\text{conv}(A)$ l'enveloppe convexe de A , c'est un convexe de E , et aussi le plus petit convexe contenant A .

Définition 2.7

On dit qu'une matrice $X = [x_{ij}]_{1 \leq i, j \leq n}$ est stochastique si elle est positive et si :

$$\text{pour tout } j \in \{1, 2, \dots, n\}; \sum_{i=1}^n x_{ij} = 1.$$

Définition 2.8

Une matrice $X \in \mathbb{R}^{n \times p}$ est dite doublement stochastique si :

$$\forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, p\}; x_{ij} \geq 0, \sum_{i=1}^n x_{ij} = 1, \text{ et } \sum_{j=1}^p x_{ij} = 1.$$

Définition 2.9

Si $A \subset E$ est un ensemble fini de points, alors l'enveloppe convexe de A est appelée polytope, et ses points extrémaux sont appelés sommets.

Théorème 2.2 (Krein-Milman)

soit A une partie compacte et convexe de \mathbb{R}^n , et soit $\{s_1, s_2, \dots, s_k\}$ avec $k \leq n$, est l'ensemble des points extrémaux de A , alors

$$A = \text{conv} \{s_1, s_2, \dots, s_k\} \tag{2.19}$$

Preuve du théorème 2.2

Pour la preuve, nous pouvons consulter [51].

Proposition 2.2 ([32])

Soit $X^+ \in \mathbb{R}_+^{n \times p}$ avec $X = A^+ B^+$ une factorisation non négative où $A^+ \in \mathbb{R}_+^{n \times r_+(X)}$ et $B^+ \in \mathbb{R}_+^{r_+(X) \times p}$.

Le rang non négatif $r_+(X)$ est le nombre des points extrémaux associés à l'enveloppe convexe engendré par les colonnes de A .

Preuve de la proposition 2.2

Soit une matrice X dont on note par r_+ son rang non négatif, on a

$$X = AB,$$

où X , A et B sont des matrices colonnes stochastiques.

Notons par $\{x_1, x_2, \dots, x_p\} \in \mathbb{R}^{n \times p}$ les colonnes de la matrice X , et soit la matrice $\sigma(X)$ définie par :

$$\sigma(X) = \text{diag}\{\|x_1\|, \|x_2\|, \dots, \|x_p\|\},$$

ou $\|\cdot\|$ est la norme de Frobenius.

On définit l'opérateur ϑ par :

$$\vartheta(X) = X\sigma(X)^{-1} \tag{2.20}$$

On peut écrire la matrice X comme :

$$X = \vartheta(X)\sigma(X) = AB,$$

puisque $\sigma(A) = I$, donc

$$X = \vartheta(A)\vartheta(B)\sigma(B),$$

on obtient alors le système :

$$\begin{cases} \vartheta(X) = \vartheta(A)\vartheta(B), \\ \sigma(X) = \sigma(B) \end{cases} \tag{2.21}$$

Si $r_+ = r_+(X)$, alors $r_+ = r_+(\vartheta(X))$.

Si on note par $\tilde{A} = \vartheta(A)$ et $\tilde{B} = \vartheta(B)$, on obtient la factorisation $X = \tilde{A}\tilde{B}$ telle que \tilde{A} est une matrice stochastique.

Supposant par la suite que $X = AB$ telle que la somme de chaque colonne de la matrice B égale à 1, $r_+ = r_+(X)$.

On note par : $\{x_1, x_2, \dots, x_p\}$, $\{a_1, a_2, \dots, a_{r_+}\}$, et $\{b_1, b_2, \dots, b_p\}$ les ensembles des colonnes des matrices X, A , et B respectivement.

Pour tout $j \in \{1, 2, \dots, p\}$, on a

$$x_j = Ab_j \text{ avec } \sum_{i=1}^{r_+} b_{ij} = 1$$

alors

$$\{x_1, x_2, \dots, x_p\} \in \text{conv}\{a_1, a_2, \dots, a_{r_+}\} \tag{2.22}$$

Soit e_1, e_2, \dots, e_k sont les points extrémaux du $\text{conv}\{a_1, a_2, \dots, a_{r_+}\}$.

Il suffit de montrer que $\{e_1, e_2, \dots, e_k\} = \text{conv}\{a_1, a_2, \dots, a_{r_+}\}$ ou bien $k = r_+$.

Il est clair que $\{e_1, e_2, \dots, e_k\} \in \text{conv}\{a_1, a_2, \dots, a_{r_+}\}$, alors $k \leq r_+$.

D'autre part, $\{x_1, x_2, \dots, x_p\} \in \text{conv}\{e_1, e_2, \dots, e_k\}$, ceci est équivalent à :

pour tout $j \in \{1, 2, \dots, p\}$, $x_j = E\mu_j$ telle que $\sum_{i=1}^k \mu_{ij} = 1$, avec E est la matrice dont les vecteurs colonne sont $\{e_1, e_2, \dots, e_k\}$.

Soit M est la matrice des vecteurs : $\{\mu_1, \mu_2, \dots, \mu_k\}$, donc on a une nouvelle factorisation de X sous la forme :

$$X = EM \tag{2.23}$$

mais l'écriture 2.23 montre aussi que $k \geq r_+$.

On conclut alors que $k = r_+$

d'où, $r_+ = r_+(X)$ est le nombre de points extrémaux de l'enveloppe convexe engendré par les colonnes de la matrice A .

2.3 Variations et extensions du NMF

Dans cette section, nous allons présenter les différentes variations qui sont enracinées du NMF et proposées pour différentes perspectives.

2.3.1 La factorisation en matrices semi non négatives Semi-NMF

La factorisation en matrices semi non négatives **Semi-NMF** est une technique désigné au matrices des données qui sont de signes quelconques, i.e., le problème **Semi-NMF** peut prendre la forme suivante :

$$X \approx AB^+, X \in \mathbb{R}^{n \times p}, A \in \mathbb{R}^{n \times k} \quad \text{et} \quad B^+ \in \mathbb{R}^{k \times p} \tag{2.24}$$

La factorisation semi non négative a été introduite dans une perspective de clustering (la classification non-supervisée des données) [23]. Dans la classification des colonnes de la matrice des données X , les colonnes de la matrice A définie dans 2.24 peuvent

se voir comme les centroides des classes et les lignes de B^+ comme des indicateurs d'appartenance à ces classes. Une autre forme pour la factorisation semi non négative est : $X \approx A^+B$ si la classification est désignée au lignes de la matrice X .

2.3.2 La factorisation convexe en matrices non négatives Convex-NMF

La factorisation convexe en matrices non négatives **Convex-NMF** a été introduite pour une raison d'interprétabilité [23], on exigeons que les vecteurs de base $A = (a_1, \dots, a_k)$ soient des combinaisons convexes des colonnes de la matrice originale X , i.e., pour $l = 1, \dots, k$:

$$a_l = w_{1l}X_1 + \dots + w_{pl}X_p = XW_l$$

ou bien avec l'écriture matricielle :

$$A = XW$$

où $W \in R_+^{n \times k}$ et $\sum_{i=1}^p w_{ij} = 1, j = 1, \dots, l$. Par conséquent, le modèle **Convex-NMF** peut prendre la forme suivante :

$$X \approx XWB. \tag{2.25}$$

2.3.3 La Tri-factorisation en matrices non négatives Tri-NMF

La Tri-factorisation en matrices non négatives **Tri-NMF** a été adresser pour le problème de Co-clustering [25], c'est à dire lorsqu'on cherche à la foi la classification des lignes et des colonnes de la matrice des données $X \in \mathbb{R}^{n \times p}$. Ce modèle vise à trouver trois matrices : $A \in \mathbb{R}^{n \times k_1}, C \in \mathbb{R}^{k_1 \times k_2}$ et $B \in \mathbb{R}^{k_1 \times p}$ telles que :

$$X \approx ACB \tag{2.26}$$

sous les contraintes : $A^tA = I$ et $B^tB = I$. A et B sont les deux matrices indicateurs de lignes et des colonnes respectivement, et C est appelée le degré de liberté qui laisse l'approximation 2.26 plus stricte.

2.3.4 La factorisation en matrices non négatives binaires **BNMF**

La factorisation en matrices non négatives binaires **BNMF** est une méthode proposée dans [57] dont la perspective est la classification des données binaires. Le modèle traite les données d'une manière à conserver les caractéristiques en fonction de leurs relations d'association symétriques.

Il s'agit d'une matrice d'entrée de données binaires $X \in \{0, 1\}^{n \times p}$. Le problème est d'approximer la matrice X en deux matrices binaires A et B , i.e.,

$$X \approx AB; A \in \{0, 1\}^{n \times k} \quad \text{et} \quad B \in \{0, 1\}^{k \times p} \quad (2.27)$$

Dans le modèle de Clustering par blocs diagonales, la matrice binaire A désigne explicitement les appartenances des points aux différentes classes, et la matrice B indique les représentations des entités de chaque classe. Pour plus de détails, nous pouvons consulter [82] et [57].

2.4 Les algorithmes de résolution du problème **NMF**

Plusieurs algorithmes itératifs sont étés développés pour résoudre le problème **NMF**. Généralement, ces algorithmes peuvent être divisés en trois classes majeurs : les méthodes multiplicatifs, les méthodes des gradients projetés et les méthodes basées sur les moindres carrés non négatives **ANLS**. Chacun de ces algorithmes peut par ailleurs être initialisé de différentes façons :

- Plusieurs initialisations aléatoires de A et B , le meilleur ajustement est conservé,
- Double décomposition en valeurs singulières non négatives,
- Une classification k-means des lignes ou des colonnes,
- ...

Entre le choix de la fonction objective (fonction de Frobenius ou divergence), le choix de l'algorithme ou d'une de ces variantes, le choix de l'initialisation, etc, cela fait beaucoup d'options à comparer et tester. Comme toujours avec une nouvelle méthode et la pression de publication, de très nombreuses variantes apparaissent avant qu'une sélection n'opère pour aboutir à des choix plus efficaces et consensuels d'options en

fonction du type de données traitées. Les auteurs dans [3] décrivent les principes de ces différents algorithmes et commentent leurs propriétés : convergence, complexité.

2.4.1 Les méthodes multiplicatives (MU)

Proposés pour la première fois par Lee et Seung (1999) [53], ils ont décrit des règles de mise à jour des facteurs A et B s'exprimant sous forme multiplicative qui assurent à la fois la non négativité des coefficients à chaque itération et la décroissance monotone du critère $D(X/AB)$. L'algorithme multiplicatif est très couramment utilisé en raison de la grande simplicité d'implémentation et de la rapidité de calcul des itérations. Ces règles peuvent s'écrire de la manière suivante :

$$\left\{ \begin{array}{l} B \leftarrow B \otimes \frac{A^t X}{A^t(AB)} \\ A \leftarrow A \otimes \frac{XB^t}{(AB)B^t} \end{array} \right. ,$$

où \otimes représente le produit d'Hadamard (produit terme à terme).

Algorithme : Multiplicative Update for NMF[23] (Sous Matlab)

```

A = rand(n, k); Initialize A as random dense matrix
B = rand(k, p); Initialize B as random dense matrix
for i = 1 : maxiter
(MU) B = B .* (A^t X) ./ (A^t AB + 10^-9);
(MU) A = A .* (XB^t) ./ (ABB^t + 10^-9);
end

```

L'addition de 10^{-9} est met pour éviter la division par zéros.

Puisque les algorithmes de Lee et Seung sont les premiers connus pour résoudre le problème **NMF**, ils sont devenus une références par rapport au quel les nouveaux algorithmes sont comparées. Il a été démontré de manière répétée que les algorithmes multiplicatifs , quand ils convergent (ce qui est souvent dans la pratique), sont lent à converger, ils ont besoin de beaucoup plus d'itérations que des alternatives telles que les approches de la descente de gradient et les moindres carrées alternées décrits ci-dessous.

L'intérêt des algorithmes multiplicatifs Les algorithmes à mises à jour multiplicatives sont semble-t-ils, les plus utilisés actuellement en **NMF**. Ces algorithmes présentent en effet un certain nombre d'avantages sur des algorithmes plus classiques (Newton, Gradient projeté). Les algorithmes à mises à jour multiplicatives sont faciles à mettre en oeuvre, et ils sont relativement rapides (beaucoup plus rapides que les méthodes d'ordres 2 et plus rapides que les algorithmes de gradient avec recherche de pas optimal). Il semble que pour des données à forte dynamique, les algorithmes à mises à jours multiplicatives donnent de meilleurs résultats. Une faiblesse de ce type d'algorithmes est la lenteur de convergence à l'approche de la solution.

2.4.2 La méthode des gradients projetés (PG)

Proposé par Lin et al. (2007) [58], la méthode de la descente de gradient est généralement désignée pour les problèmes d'optimisation sous contraintes de bornitude. Le principe de cette méthode est de partir d'un point aléatoire puis de se déplacer dans la direction de la plus forte pente (c'est la direction opposée au gradient, de manière à faire décroître la fonction). En appliquant un certain nombre d'itérations, l'algorithme converge vers une solution qui est un minimum local. Dans notre problématique, il s'agit de découper le problème original en deux sous-problèmes couplés mais résolus indépendamment, c'est à dire :

$$J(A^{k+1}) = \min_{A \geq 0} \|X - AB^k\|_F^2 \quad (2.28)$$

$$J(B^{k+1}) = \min_{B \geq 0} \|X - A^k B\|_F^2 \quad (2.29)$$

Ces deux sous-problèmes peuvent être abordés par un algorithme de type gradient non négatif, c'est à dire, un algorithme du gradient dont la solution courante est projetée dans le demi-espace positif (l'orthant positif). Les nouvelles solutions sont calculée selon les règles de mise à jour suivantes :

$$\begin{aligned} A &\leftarrow [A - \alpha_A \nabla_A C]^+, \\ B &\leftarrow [B - \alpha_B \nabla_B C]^+, \end{aligned} \quad (2.30)$$

où ∇_A (resp. ∇_B) représente l'opérateur gradient par rapport à la matrice A (resp. B) et $[\cdot]^+$ représente l'opérateur de projection sur l'orthant positif.

Les pas α_A et α_B peuvent être choisis de plusieurs façons :

- (a) Par une recherche linéaire pour minimiser $D(A - \alpha_A \setminus \text{triangledown}_A C, B)$. Lorsque la divergence utilisée pour la fonction de coût n'est pas définie en dehors de l'orthant positif, le pas ne peut prendre des valeurs que dans un intervalle borné de \mathbb{R} , et par conséquent, on peut utiliser une méthode de recherche par section d'or (Forsythe et al 1976). Cette méthode est assez lente.
- (b) Par la minimisation de la parabole tangente à $C(A - \alpha_A \nabla_A C, B)$ (ce qui nécessite que cette fonction soit convexe), et ce qui est nettement plus rapide qu'une méthode de recherche exhaustive.
- (c) Par l'utilisation d'un pas vérifiant les critères de Wolf (comme dans [58] qui n'utilise en fait que la moitié des critères de Wolf : le critère d'Armijo).
- (d) Par l'utilisation d'un pas fixe, le coût de calcul d'une itération est alors très faible, mais les résultats sont assez pauvres (la convergence est généralement lente).

2.4.3 La méthode des moindres carrés non négatives (NNLS)

Le premier algorithme proposé pour résoudre le problème **NMF** (sous sa forme initiale posé par Paatero et al. 1994 [64]) est l'algorithme des moindres carrés alternées. Il est facile à comprendre que si en fixant l'une des deux matrices A ou B dans $X \approx AB$, le problème devient un problème des moindres carrés avec la contrainte de non négativité **NNLS**.

? Une forme générale d'un algorithme **NMF** basé sur la méthode **NNLS** est donnée par :

Algorithme : NMF-NNLS

Entrée : $X \in \mathbb{R}_+^{n \times p}$; $k \in \mathbb{N}^*$

Sortie : $A \in \mathbb{R}_+^{n \times k}$ et $B \in \mathbb{R}_+^{k \times p}$ telles que $(A, B) = \text{argmin} \frac{1}{2} \|X - AB\|_F^2$

1. Initialiser : A et B
2. **répéter** :
3. Résoudre $\min_{B \geq 0} \frac{1}{2} \|X - AB\|_F^2$
4. Résoudre $\min_{A \geq 0} \frac{1}{2} \|X - AB\|_F^2$
5. **Jusqu'à** : Condition d'arrêt.

Les méthodes qui peuvent étre appliquées pour résoudre un problème **NNLS** (problème défini dans les étapes 3. et 4. de l'algorithme ci-dessus) se trouvent dans [10], [30], et [52].

2.5 L'équivalence entre le modèle NMF et le **k-means** Clustering

On va montrer dans cette section que le problème **k-means** peut avoir une autre version de lecture. En effet, on peut voir le **k-means** comme une factorisation matricielle.

Rappelons que le **k-means** cherche une partition P des lignes d'une matrice de données X de dimension (n, p) , en k classes, qui minimise la fonction suivante :

$$F^{(1)} = \sum_{s=1}^k \sum_{X_t \in C_s} \|X_t - m_s\|^2, \quad (2.31)$$

avec $m_s, 1 \leq s \leq k$ sont les centroides des classes.

La norme utilisée est celle de Frobenius.

Théorème 2.3

Soit $W = X^t X$, le problème de la minimisation de 2.31 est équivalent à

$$\max_{H^t H = I, H \geq 0} F^{(2)}(H) = \text{Tr}(H^t W H) \quad (2.32)$$

avec H une matrice vérifiant $H^t H = I$ et $H \geq 0$.

Preuve du théorème 2.3

On a la formulation $F^{(1)} = \sum_{s=1}^k \sum_{X_t \in C_s} \|X_t - m_s\|^2$.

On s'intéresse à la factorisation matricielle de m_s , c'est à dire on souhaite exprimer m_s sous forme d'un vecteur h indiquant l'appartenance ou pas de X_t à C_s .

On va développer la quantité 2.31 pour $k = 2$, après on va généraliser.

Si on prends $\sum_{X_t \in C_s} \|X_t - m_s\|^2$, on aura

$$\begin{aligned} \sum_{X_t \in C_1} \|X_t - m_1\|^2 &= \sum_{X_t \in C_1} \|X_t\|^2 + \sum_{X_t \in C_1} \|m_1\|^2 - 2 \sum_{X_t \in C_1} \langle X_t, m_1 \rangle \\ &= \sum_{X_t \in C_1} \|X_t\|^2 + \sum_{X_t \in C_1} \|m_1\|^2 - 2n_1 \left\langle \sum_{X_t \in C_1} \frac{X_t}{n_1}, m_1 \right\rangle \\ &= \sum_{X_t \in C_1} \|X_t\|^2 + \sum_{X_t \in C_1} \|m_1\|^2 - 2n_1 \|m_1\|^2. \end{aligned}$$

Puisque n_1 est le nombre de points dans la classe C_1 , alors

$$\sum_{X_t \in C_1} \|X_t - m_1\|^2 = \sum_{X_t \in C_1} \|X_t\|^2 - n_1 \|m_1\|^2.$$

De la même façon, on a

$$\sum_{X_t \in C_2} \|X_t - m_2\|^2 = \sum_{X_t \in C_2} \|X_t\|^2 - n_2 \|m_2\|^2.$$

On note par X_1 la matrice dont les lignes sont les éléments de C_1 , et par X_2 la matrice dont les lignes sont les éléments de C_2 , donc

$$\sum_{s=1}^2 \sum_{X_t \in C_s} \|X_t - m_s\|^2 = Tr(X_1 X_1^t) + Tr(X_2 X_2^t) - \frac{1}{n_1} Tr(1^t X_1 X_1^t 1) - \frac{1}{n_2} Tr(1^t X_2 X_2^t 1) \quad (2.33)$$

Soit

$$\begin{cases} h_1 = \frac{1}{\sqrt{n_1}} (0, \dots, 0, \overbrace{1, \dots, 1}^{n_1}, 0, \dots, 0) \\ h_2 = \frac{1}{\sqrt{n_2}} (0, \dots, 0, \overbrace{1, \dots, 1}^{n_2}, 0, \dots, 0) \end{cases},$$

alors

$$\frac{1}{\sqrt{n_1}}1^t X_1 = h_1 X, \quad \text{et} \quad \frac{1}{\sqrt{n_2}}1^t X_2 = h_2 X,$$

2.33 sera de la forme

$$Tr(XX^t) - Tr(h_1 XX^t h_1^t) - Tr(h_2 XX^t h_2^t)$$

ce qui donne

$$Tr(XX^t) - Tr(HXX^t H^t), \quad \text{avec} \quad H = (h_1, h_2).$$

Pour une somme d'ordre k , on obtient

$$F^{(1)}(H) = Tr(X^t X) - Tr(H^t X^t X H)$$

avec H est la matrice (h_1, h_2, \dots, h_k) définit par :

$$\begin{cases} h_k = (0, \dots, 0, \overbrace{1, \dots, 1}^{n_k}, 0, \dots, 0)^t / n_k^{\frac{1}{2}} \\ h_k^t h_l = \delta_{kl} \end{cases} \quad (2.34)$$

si on note $W = X^t X$, il est claire que la minimisation de $F^{(1)}(H)$ revient à la maximisation :

$$\max_{H^t H = I, H \geq 0} F^{(2)}(H) = tr(H^t W H).$$

Reste à montrer que 2.32 vérifie la propriété de la monotonie (décroissance) du critère d'optimisation :

Soit \tilde{H} est une solution du problème 2.31, alors

$$F^{(1)}(\tilde{H}) \leq F^{(1)}(H) \quad \text{pour tout} \quad H \geq 0 \quad \text{et} \quad H^t H = I.$$

ou bien

$$Tr(XX^t) - Tr(\tilde{H}XX^t\tilde{H}^t) \leq Tr(XX^t) - Tr(HXX^t H^t)$$

ce qui est équivalent à

$$\text{Tr}(HXX^tH^t) \leq \text{Tr}(\widetilde{H}XX^t\widetilde{H}^t)$$

alors on obtient le **k-means** clustering en étudiant le problème d'optimisation 2.32.

Théorème 2.4

Le problème de maximisation 2.32 est équivalent à

$$F^{(3)}(H) = \min_{HH^t=I, H \geq 0} \|W - HH^t\|^2. \quad (2.35)$$

Preuve du théorème 2.4

Soit

$$F^{(3)}(H) = \min_{H^tH=I, H \geq 0} \{-\text{tr}(H^tWH)\} \quad (2.36)$$

Il est clair que la minimisation 2.36 est équivalente à 2.32, alors, une solution de ce problème est la factorisation $W \approx HH^t$ avec $H \geq 0$.

En développant

$$F^{(3)}(H) = \min_{H^tH=I, H \geq 0} \{-2\text{Tr}(H^tWH)\},$$

on obtient

$$F^{(3)}(H) = \min_{H^tH=I, H \geq 0} \{\|W\|^2 - \text{Tr}(H^tWH) + \|H^tH\|^2\},$$

qui est bien

$$F^{(3)}(H) = \min_{HH^t=I, H \geq 0} \|W - HH^t\|^2.$$

Théorème 2.5

Pour une matrice $H = (h_1, h_2, \dots, h_k)$, vérifiant les conditions :

$$HH^t = I, H \geq 0 \text{ et } \sum_{s=1}^k h_{ts} = 1 \text{ pour tout } 1 \leq t \leq n.$$

le problème de minimisation de 2.31 est équivalent à la minimisation de la fonction $F^{(4)}$ définie comme :

$$F^{(4)}(H) = \|X - MH^t\|^2 \quad (2.37)$$

Preuve du théorème 2.5

On a

$$\begin{aligned} F^{(4)}(H) &= \|X - MH^t\|^2 \\ &= \sum_{t=1}^n \left\| X_{t:} - \sum_{s=1}^k h_{ts} m_s \right\|^2, \end{aligned}$$

et puisque $\sum_{s=1}^k h_{ts} = 1$ pour tout $1 \leq t \leq n$

$$F^{(4)}(H) = \sum_{t=1}^n \left\| \sum_{s=1}^k h_{ts} (X_{t:} - m_s) \right\|^2.$$

En développons la quantité $\left\| \sum_{s=1}^k h_{ts} (X_{t:} - m_s) \right\|^2$, on a

$$\begin{aligned} \left\| \sum_{s=1}^k h_{ts} (X_{t:} - m_s) \right\|^2 &= \left\langle \sum_{s=1}^k h_{ts} (X_{t:} - m_s), \sum_{s=1}^k h_{ts} (X_{t:} - m_s) \right\rangle \\ &= \sum_{s=1}^k \sum_{s'=1}^k \langle h_{ts} (X_{t:} - m_s), h_{ts'} (X_{t:} - m_{s'}) \rangle \\ &= \sum_{s=1}^k \sum_{s'=1}^k h_{ts} h_{ts'} \langle X_{t:} - m_s, X_{t:} - m_{s'} \rangle \\ &= \sum_{s=1}^k h_{ts}^2 \langle X_{t:} - m_s, X_{t:} - m_s \rangle + \sum_{s=1, s'=1, s \neq s'}^k h_{ts} h_{ts'} \langle X_{t:} - m_s, X_{t:} - m_{s'} \rangle. \end{aligned}$$

Pour $s \neq s'$, on a que $h_{ts} h_{ts'} = 0$, alors

$$\begin{aligned} \left\| \sum_{s=1}^k h_{ts} (X_{t:} - m_s) \right\|^2 &= \sum_{s=1}^k h_{ts}^2 \|X_{t:} - m_s\|^2 \\ &= \sum_{s=1}^k h_{ts} \|X_{t:} - m_s\|^2, \text{ (car } h_{ts}^2 = h_{ts} \text{.)} \end{aligned}$$

D'où

$$\begin{aligned} F^{(4)}(H) &= \sum_{t=1}^n \sum_{s=1}^k h_{ts}^2 \|X_{t:} - m_s\|^2 \\ &= \sum_{s=1}^k \sum_{X_{t:} \in C_s} \|X_{t:} - m_s\|^2 \\ &= F^{(1)}. \end{aligned}$$

Théorème 2.6

La factorisation non négative de la matrice X vérifiant :

$$F^{(5)}(H) = \min_{F \geq 0, H \geq 0, HH^t = I} \|X - FH^t\|^2 \quad (2.38)$$

est équivalente au problème de la minimisation de 2.31.

Preuve du théorème 2.6

Pour effectuer la preuve de ce théorème, on va appliquer la formule de la régression multiple pour les colonnes de la matrice $\tilde{X} - HF^t$, avec $\tilde{X} = X^t$, qui sont définis comme :

$(\tilde{X}_1 - H\tilde{f}_1, \tilde{X}_2 - H\tilde{f}_2, \dots, \tilde{X}_k - H\tilde{f}_k)$, où $(\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_k)$ sont les colonnes de la matrice F^t .

Alors,

$$\begin{aligned} \|X - FH^t\|^2 &= \|X^t - HF^t\|^2 \\ &= \|\tilde{X} - HF^t\|^2 \\ &= \sum_{i=1}^k \|\tilde{X}_i - H\tilde{f}_i\|^2. \end{aligned}$$

Donc, les \tilde{f}_i peuvent être obtenus comme suit :

$$(\tilde{f}_1 = (H^t H)^{-1} H^t X_1, \tilde{f}_2 = (H^t H)^{-1} H^t X_2, \dots, \tilde{f}_k = (H^t H)^{-1} H^t X_k).$$

Avec la transposition des vecteurs \tilde{f}_i , et la condition $HH^t = I$, on peut écrire les colonnes de la matrice F comme

$$(f_1 = X_1 H, f_2 = X_2 H, \dots, f_k = X_k H)$$

d'où

$$F = XH$$

Ce qui montre que

$$\|X - FH^t\|^2 = Tr(X^tX - H^tX^tXH)$$

Alors, le problème de minimisation 2.38 est équivalent au problème

$$\max_{H^tH=I, H \geq 0} Tr(H^tX^tXH),$$

ce qui est bien le problème 2.31.

NMF et k -means Soit $X \in \mathbb{R}_+^{n \times p}$ est la matrice des données, et soient $[X_{1.}, X_{2.}, \dots, X_{n.}] = X^t$, $[X_{.1}, X_{.2}, \dots, X_{.p}] = X$ sont les matrices des lignes et des colonnes de X , et $F = [f_1, f_2, \dots, f_n]$, $G = [g_1, g_2, \dots, g_p]$ sont leurs matrices indicateurs d'appartenance aux classes respectivement.

Si on note par $W = \begin{pmatrix} 0 & X \\ X^t & 0 \end{pmatrix}$ et $H = \frac{1}{\sqrt{2}} \begin{pmatrix} F \\ G \end{pmatrix}$, on a

$$I = \max_{H^tH=I, H \geq 0} \{ Tr(H^tWH) \}$$

$$I = \max_{F^tF=I; G^tG=I; F, G \geq 0} \{ Tr(F^tXG) \} \quad (2.39)$$

Ceci est équivalent au problème d'optimisation de k - *means* clustering :

$$\min_{F^tF=I; G^tG=I; F, G \geq 0} \|X - FG^t\|^2 \quad (2.40)$$

Conclusion Une factorisation non négative d'une matrice de données X sous la forme $X \approx FG^t$ avec G est orthogonale (respectivement F est orthogonale), est identique à k - *means* clustering appliqué sur les colonnes de X (respectivement sur les lignes de X).

Chapitre 3

Rang semi non négatif des matrices réelles et sa connexion au rang usuel

Sommaire

3.1	Le rang semi non négatif	63
3.1.1	Position du problème	63
3.1.2	Existence et unicité	63
3.2	Bornes pour le rang semi non négatif	64
3.3	Les valeurs possibles du rang semi non négatif	67
3.4	Caractérisation des matrices pour les quelles le rang semi non négatif et le rang usuel coïncident	68
3.5	Interprétation géométrique pour la caractérisation des matrices vérifiant $r_s(X) = r_u(X)$	73
3.6	Calcul d'une factorisation semi non négative	75

Ce chapitre vise à présenter et résoudre le problème de la factorisation semi non négative exacte, ce qui est l'objet de notre première publication [14]. Notons par la suite que le nom **Semi-NMF** est désigné uniquement pour le problème d'approximation matricielle (voir chapitre suivant).

3.1 Le rang semi non négatif

3.1.1 Position du problème

Soit une matrice des données $X \in \mathbb{R}^{n \times p}$. Le problème de la factorisation semi non négative exacte est la recherche de deux matrices $A \in \mathbb{R}^{n \times k}$ et $B \in \mathbb{R}_+^{k \times p}$ telles que :

$$X = AB. \quad (3.1)$$

Les inconnus de ce problème sont : A, B et l'entier k .

3.1.2 Existence et unicité

L'existence d'une solution pour le problème 3.1 dépend du paramètre k . Une solution pour le problème 3.1 existe mais elle n'est pas unique, i.e., pour $k = p$, une solution est $X = XI_p$. D'autre part, si une solution (\tilde{A}, \tilde{B}) existe, alors pour toute matrice inversible $S \in \mathbb{R}_+^{k \times k}$, on a :

$$X = \tilde{A}S^{-1}S\tilde{B} = \hat{A}\hat{B} \text{ où } \hat{A} = \tilde{A}S^{-1} \text{ et } \hat{B} = S\tilde{B}.$$

La matrice $S\tilde{B}$ est non négative, alors le problème 3.1 admet une infinité de solutions.

Comme nous avons défini le rang usuel et le rang non négatif dans les chapitres précédents, on va introduire pour le problème 3.1 la notion du rang semi non négatif défini comme suit :

Définition 3.1

Soit $J = \{k / \exists A \in \mathbb{R}^{n \times k}, B \in \mathbb{R}_+^{k \times p}, X = AB\}$.

On appelle rang semi non négatif de la matrice X , la valeur minimale k pour que le problème 3.1 admette une solution, on le note par $r_s(X)$, i.e., $r_s(X) = \min_k J$.

Rang semi non négatif des matrices réelles et sa connexion au rang usuel

Deux raisons principales motivent notre intérêt pour introduire le rang semi non négatif.

La première raison est mathématique, elle réside dans le fait que le rang semi non négatif d'une matrice $X \in \mathbb{R}^{n \times p}$ n'est pas toujours égal à son rang usuel ($r_s(X) \neq r_u(X)$). De même, pour une matrice non négative $X \in \mathbb{R}_+^{n \times p}$, son rang semi non négatif est différent de son rang non négatif. Ces différences peuvent être illustrées par l'exemple ci-dessous :

Exemple 3.1

$$X_0 = \begin{pmatrix} 1 & -1 & 1 & 1 \\ 2 & -2 & 2 & 2 \\ 3 & -3 & 3 & 3 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ 2 & -2 \\ 3 & -3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$
$$X_0^+ = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Nous avons : $r_u(X_0) = 1$ et $r_s(X_0) = 2$, $r_+(X_0^+) = 4$ [20] et $r_s(X_0^+) = 3$.

Ces différences entre les rangs nous permettent de conclure que les trois rangs cités précédemment se réfèrent à des factorisations intrinsèquement différentes.

La deuxième raison est appliquée, le fait que le modèle **Semi-NMF** a été utilisé avec succès dans plusieurs applications de la classification non supervisée (clustering) ([62], [6], [80]).

3.2 Bornes pour le rang semi non négatif

Lemme 3.1 (Chouh et al., 2015)

Soit $X \in \mathbb{R}^{n \times p}$, on a

$$r_u(X) \leq r_s(X) \leq \min(p, r_u(X) + 1) \quad (3.2)$$

L'intérêt de ce lemme est de permettre de caractériser les bornes inférieure et supérieure optimales pour le rang semi non négatif.

Preuve du lemme 3.1

Rappelons que le rang usuel de $X \in \mathbb{R}^{n \times p}$ étant l'entier minimal k_1 tel qu'une factorisation : $X = AB, A \in \mathbb{R}^{n \times k_1}, B \in \mathbb{R}^{k_1 \times p}$ existe. Si on suppose que $r_s(X) = k_2$ alors $X = A^1 B^1$ avec $A^1 \in \mathbb{R}^{n \times k_2}, B^1 \in \mathbb{R}_+^{k_2 \times p}$ donc k_1 est nécessairement inférieure a k_2 , d'où

$$r_u(X) \leq r_s(X).$$

Démontrons maintenant que $r_s(X) \leq \min(p, r_u(X) + 1)$, pour ce faire on considère les deux cas suivant :

1^{er} cas $\min(p, r_u(X) + 1) = p$

Il est bien évident que $X = XI_p$ est une factorisation possible de $X \in \mathbb{R}^{n \times p}$, il en résulte que : $r_s(X) \leq p$, car $r_s(X)$ est l'entier minimal pour toute factorisation semi non négative de X .

2^{ème} cas $\min(p, r_u(X) + 1) = r_u(X) + 1$

Montrons que $r_s(X) \leq r_u(X) + 1$.

Soit $X = AB$ une factorisation de $X \in \mathbb{R}^{n \times p}$ avec $A \in \mathbb{R}^{n, r_u(X)}$ et $B \in \mathbb{R}^{r_u(X), p}$.

Soit $B = [b_{ij}]_{\substack{1 \leq i \leq r_u(X) \\ 1 \leq j \leq p}}$, on note par $\lambda = \max_{\substack{1 \leq i \leq r_u(X), 1 \leq j \leq p}} \{|b_{ij}|\}$.

Définissons la matrice \tilde{B} telle que $\tilde{B} = B + \lambda C$, où $C = 1_{r_u(X)} 1_p^t$ avec 1_n est le vecteur dans \mathbb{R}^n constitué que des 1.

Remarquons que la matrice $\tilde{B} = [\tilde{b}_{ij}]_{\substack{1 \leq i \leq r_u(X) \\ 1 \leq j \leq p}}$ est non négative. On effet, si on considère les deux matrices suivantes :

$$\hat{A} = [A, -\lambda A 1_{r_u(X)}] \in \mathbb{R}^{n \times r_u(X)+1},$$

$$\hat{B} = \begin{bmatrix} \tilde{B} \\ 1_p^t \end{bmatrix} \in \mathbb{R}_+^{(r_u(X)+1) \times p}$$

Il est facile de vérifier que :

$$\hat{A} \hat{B} = A \tilde{B} - \lambda A 1_{r_u(X)} 1_p^t = A(B + \lambda C) - \lambda AC = AB$$

Rang semi non négatif des matrices réelles et sa connexion au rang usuel

Ce qui nous laisse à conclure que :

$$r_s(X) \leq r_u(X) + 1.$$

Corollaire 3.1

Si $r_u(X) = p$, alors

$$r_s(X) = r_u(X) = p \tag{3.3}$$

Preuve du Corollaire 3.1

D'après le lemme 3.1, on a

$$r_u(X) \leq r_s(X) \leq \min(p, r_u(X) + 1)$$

donc

$$r_s(X) = p$$

La factorisation dans ce cas est obtenue le fait que $X = XI_p$, alors pour toute matrice $S \in \mathbb{R}_+^{p \times p}$ inversible, on a

$$X = XS^{-1}S,$$

donc, il suffit de choisir $A = XS^{-1}$ et $B = S$ et on obtient la factorisation $X = AB$ avec $A \in \mathbb{R}^{n \times p}$ et $B \in \mathbb{R}_+^{p \times p}$

Remarque 3.1

► Si $X \in \mathbb{R}^{n \times n}$ est une matrice inversible, alors $r_s(X) = r_u(X) = n$.

On effet, on sait que toute matrice inversible dans $\mathbb{R}^{n \times n}$ est de rang n [37], donc il n'existe pas de concept de rang semi non négatif pour les matrices carrés inversibles.

3.3 Les valeurs possibles du rang semi non négatif

Soient E^1 et E^2 deux sous-espaces de $\mathbb{R}^{n \times p}$ définis comme suit :

$$\begin{aligned} E^1 &= \{X \in \mathbb{R}^{n \times p}, r_s(X) = r_u(X)\}, \\ E^2 &= \{X \in \mathbb{R}^{n \times p}, r_s(X) = r_u(X) + 1\}. \end{aligned}$$

Nous introduisons le théorème 3.1 qui montre que quelque soit la matrice $X \in \mathbb{R}^{n \times p}$, son rang semi non négatif prend uniquement l'une des deux valeurs : $r_u(X)$ ou bien $r_u(X) + 1$.

Théorème 3.1 (Chouh et al., 2015)

$$E^1 \cup E^2 = \mathbb{R}^{n \times p} \tag{3.4}$$

Le théorème 3.1 nous permet aussi de déduire que les ensembles E^1 et E^2 sont disjoints et ils forment une partition de $\mathbb{R}^{n \times p}$.

Preuve du théorème 3.1

Nous allons montrer la double inclusion entre les deux ensembles $E^1 \cup E^2$ et $\mathbb{R}^{n \times p}$. Il est évident que

$$E^1 \cup E^2 \subset \mathbb{R}^{n \times p}, \tag{3.5}$$

pour montrer le théorème 3.1, il suffit de montrer que $\mathbb{R}^{n \times p} \subset E^1 \cup E^2$.

Soit $X \in \mathbb{R}^{n \times p}$, nous avons l'égalité suivante :

$$\min(p, r_u(X) + 1) = \begin{cases} r_u(X) & \text{si } r_u(X) = p \\ r_u(X) + 1 & \text{sinon} \end{cases} \tag{3.6}$$

(i) Supposons que $r_u(X) = p$

D'après le lemme 3.1, nous avons :

Rang semi non négatif des matrices réelles et sa connexion au rang usuel

$$p = r_u(X) \leq r_s(X) \Rightarrow r_s(X) = r_u(X) \Rightarrow X \in E^1$$

donc

$$X \in E^1 \cup E^2.$$

(ii) Sinon $r_u(X) \neq p$

Une autre fois, si nous utilisons l'égalité 3.6 et le lemme 3.1, nous avons :

$$r_u(X) \leq r_s(X) \leq r_u(X) + 1 \Rightarrow X \in E^1 \cup E^2.$$

Par conclusion de (i) et de (ii) :

$$\mathbb{R}^{n \times p} \subset E^1 \cup E^2. \quad (3.7)$$

Enfin, l'égalité du théorème 3.1 est obtenu en combinaison de 3.5 et 3.7.

3.4 Caractérisation des matrices pour les quelles le rang semi non négatif et le rang usuel coïncident

Cette sous section se concentre sur la caractérisation des matrices $X \in \mathbb{R}^{n \times p}$ qui vérifient l'équation suivante :

$$r_s(X) = r_u(X). \quad (3.8)$$

En effet, la classe particulière des matrices non négatives $X \in \mathbb{R}_+^{n \times p}$ est une solution de l'équation précédente.

Théorème 3.2 (Chouh et al., 2015)

Soit $X \in \mathbb{R}_+^{n \times p}$, alors

$$r_s(X) = r_u(X). \quad (3.9)$$

Cela signifie que pour les matrices ayant des valeurs non négatives, le rang usuel et le rang semi non négatif coïncident.

Preuve du théorème 3.2

Rang semi non négatif des matrices réelles et sa connexion au rang usuel

Soit $X \in \mathbb{R}_+^{n \times p}$ avec $r = r_u(X)$. D'après l'inégalité gauche donnée par le lemme 3.1, il est clair que $r \leq r_s(X)$.

Alors, pour démontrer le théorème 3.2, il suffit de montrer que $r_s(X) \leq r$. Cela peut être fait en construisant une factorisation $X = AB$ avec $A \in \mathbb{R}^{n \times r}$ et $B \in \mathbb{R}_+^{r \times p}$ comme suit :

Soient $\{l_1, l_2, \dots, l_n\}$ sont les lignes de la matrice X . Puisque X a un rang usuel égal à r , alors il y a dans X r lignes qui sont linéairement indépendantes. On note ces r lignes par l_1, l_2, \dots, l_r .

Considérons les deux matrices stockées en lignes comme :

$$B = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_r \end{bmatrix} \in \mathbb{R}_+^{r \times p} \text{ et } A = \begin{bmatrix} I_r \\ \tilde{A} \end{bmatrix} \in \mathbb{R}^{n \times r},$$

où $\tilde{A} \in \mathbb{R}^{(n-r) \times r}$, et les lignes de \tilde{A} sont $\{a_{r+1}, a_{r+2}, \dots, a_n\}$ solutions des équations :

$$a_i^t B = l_i \text{ pour } r \leq i \leq n.$$

Finalement, il est clair que $X = AB$, alors :

$$r_s(X) \leq r.$$

Comme il a été montré dans le théorème 3.2, l'équation 3.8 admet comme solution toute matrice réelle non négative.

D'autre part, l'intérêt de la présente section peut être formulé dans la question suivante :

Question :

Comment caractériser l'ensemble des matrices (pas nécessairement non négatives) qui réalisent l'égalité 3.8 (i.e., le rang semi non négatif et le rang usuel coïncident) ?

Le théorème suivant donne une caractérisation générale des solutions de l'équation 3.8

Théorème 3.3 (Chouh et al., 2015)

Soit $X \in \mathbb{R}^{n \times p}$, alors les deux assertions suivantes sont équivalentes :

- (i) $r_s(X) = r_u(X)$,
- (ii) $\exists \alpha \in \mathbb{R}^n$ avec $\alpha \neq 0_n$ tel que $X^t \alpha > 0_p$,

Rang semi non négatif des matrices réelles et sa connexion au rang usuel

où l'inégalité $X^t\alpha > 0_p$ veut dire que toutes les composantes du vecteur $X^t\alpha$ sont strictement positives.

La preuve du théorème 3.3 est basé sur le lemme présenté si dessous :

Étant donnée $Y \in \mathbb{R}^{n \times p}$, on note par :

$S(Y)$ le sous espace de \mathbb{R}^n engendré par les lignes de la matrice Y .

$Z_Y \in \mathbb{R}^{(n+1) \times p}$ est la matrice définie par : $Z_Y = \begin{bmatrix} Y + 1_n b^t \\ b^t \end{bmatrix}$ où $b \in S(Y)$.

$S(Z_Y)$ est le sous espace de \mathbb{R}^{n+1} engendré par les lignes de Z_Y .

Lemme 3.2

$$S(Z_Y) = S(Y). \quad (3.10)$$

Preuve du théorème 3.3

(i) \Rightarrow (ii).

Soit $X \in \mathbb{R}^{n \times p}$ avec $r = r_s(X) = r_u(X)$, et soit une factorisation semi non négative

$$X = AB, \quad (3.11)$$

avec $A \in \mathbb{R}^{n \times r}$ et $B \in \mathbb{R}_+^{r \times p}$. Il est important de remarquer que $r_u(A) = r_u(B) = r$, ce qui veut dire que la matrice $A^t A \in \mathbb{R}^{r \times r}$ est nécessairement inversible. Si on multiplions l'équation 3.11 par A^t , on obtient :

$$A^t X = A^t AB, \quad (3.12)$$

et en multipliant 3.12 par $(A^t A)^{-1}$, on a :

$$B = (A^t A)^{-1} A^t X, \quad (3.13)$$

la transposition de l'équation 3.13 et sa multiplication par 1_r à droite donne :

$$B^t 1_r = X^t A (A^t A)^{-1} 1_r. \quad (3.14)$$

Il suffit de choisir $\alpha_0 = A (A^t A)^{-1} 1_r$ pour avoir $X^t \alpha_0 = B^t 1_r \geq 0$.

Pour obtenir (ii), il reste à montrer que le vecteur $B^t 1_r$ ne peut jamais contenir une

Rang semi non négatif des matrices réelles et sa connexion au rang usuel

composante égale à zéro.

En effet, si on suppose que le vecteur $B^t 1_r$ contient une composante nulle, disant $\theta_{j_0} = 0$, ce qui est équivalent à : $\theta_{j_0} = b_{j_0}^t 1_r = 0$ avec b_{j_0} est la $j^{\text{ème}}$ colonne de B , alors $b_{j_0} = 0_r$, ce qui veut dire que la matrice $B \in \mathbb{R}_+^{r \times p}$ contient une colonne nulle, mais sa contredit l'hypothèse : $r_u(B) = r$, donc $X^t \alpha_0 = B^t 1_r > 0$.

(ii) \Rightarrow (i) :

Etant donnée une factorisation $X = AB$ avec $X \in \mathbb{R}^{n \times p}$, $A \in \mathbb{R}^{n \times r_u(X)}$ et $B \in \mathbb{R}^{r_u(X) \times p}$. On introduit les deux matrices $\tilde{A} \in \mathbb{R}^{n \times (r_u(X)+1)}$ et $\tilde{B} \in \mathbb{R}_+^{(r_u(X)+1) \times p}$ comme suit :

$$\tilde{A} = \left[A, -\delta A 1_{r_u(X)} \right] \text{ où } \delta = \max_{1 \leq i \leq r_u(X), 1 \leq j \leq p} \{|b_{ij}|\} \text{ et } B = [b_{ij}]_{1 \leq i \leq r_u(X), 1 \leq j \leq p}$$

$$\tilde{B} = \begin{bmatrix} B + \delta 1_{r_u(X)} w^t \\ w^t \end{bmatrix}, \text{ où } w = \lambda_{-1} v, v = X^t \alpha \text{ et } \lambda = \min_{1 \leq i \leq n} \{v_i\} \text{ avec } v_i \text{ est la } i^{\text{ème}} \text{ composante du vecteur } v.$$

Il est clair que :

$$\tilde{A} \tilde{B} = AB + \delta A 1_{r_u(X)} w^t - \delta A 1_{r_u(X)} w^t = AB.$$

Alors, on obtient une nouvelle factorisation de X comme :

$$X = \tilde{A} \tilde{B}. \tag{3.15}$$

Il est facile de remarquer que la matrice \tilde{B} est non négative, tout simplement parce que $w \geq 1_p$.

Comme conséquence du théorème 3.2, il résulte que :

$$r_s(\tilde{B}) = r_u(\tilde{B}),$$

et en appliquant le lemme 3.2 avec $Y = B$ et $Z_Y = \tilde{B}$, on a

$$S(B) = S(\tilde{B}),$$

ce qui implique que

Rang semi non négatif des matrices réelles et sa connexion au rang usuel

$$r_u(B) = \dim(S(B)) = \dim(S(\tilde{B})) = r_u(\tilde{B}).$$

($\dim(\cdot)$ est la dimension de sous espace (\cdot)).

Donc, l'égalité $r_u(B) = r_u(\tilde{B})$ se réalise.

Comme conséquence, \tilde{B} peut être écrite comme :

$$\tilde{B} = \tilde{B}_1 \tilde{B}_2 \tag{3.16}$$

avec $\tilde{B}_1 \in \mathbb{R}^{(r_u(X)+1) \times r_u(X)}$ et $\tilde{B}_2 \in \mathbb{R}_+^{r_u(X) \times p}$,

par la substitution de 3.16 dans 3.15, on obtient $X = \tilde{A}_2 \tilde{B}_2$ où $\tilde{B}_2 \in \mathbb{R}_+^{r_u(X) \times p}$ et $\tilde{A}_2 = \tilde{A} \tilde{B}_1 \in \mathbb{R}^{n \times r_u(X)}$, alors

$$r_s(X) \leq r_u(B) = r_u(X).$$

Par l'utilisation de l'inégalité gauche du lemme 3.2, on obtient :

$$r_s(X) = r_u(X).$$

Preuve du lemme 3.2

Pour montrer le lemme 1.2, nous allons montrer l'inclusion dans les deux sens entre les sous ensembles $S(Y)$ et $S(Z_Y)$.

Soient l_1, l_2, \dots, l_n sont les lignes de la matrice $X \in \mathbb{R}^{n \times p}$.

(i) $S(Y) \subset S(Z_Y)$:

Soit $t \in S(Y)$, t peut s'écrire comme $t = \sum_{i=1}^n \alpha_i l_i$, où $\alpha_i \in \mathbb{R}$, pour $1 \leq i \leq n$, et soient γ et $\beta_i, 1 \leq i \leq n$ sont des scalaires définis comme :

$$\gamma = \begin{cases} -\kappa & \text{si } \kappa = -1 \\ 0 & \text{sinon} \end{cases} ; \kappa = \sum_{i=1}^n \alpha_i,$$

$$\beta_i = \begin{cases} \alpha_i(1 - \gamma) & \text{si } \kappa = -1 \\ \frac{\alpha_i}{1 + \kappa} & \text{sinon} \end{cases} ; 1 \leq i \leq n.$$

Rang semi non négatif des matrices réelles et sa connexion au rang usuel

Il est clair de vérifier que $t = \sum_{i=1}^n \beta_i(l_i + b) + \gamma b$, ce qui montre que

$$S(Y) \subset S(Z_Y)$$

$$(i) S(Z_Y) \subset S(Y).$$

Soit $t \in S(Z_Y)$ et soient $\eta_1, \eta_2, \dots, \eta_{n+1}$ sont des scalaires tels que $t = \sum_{i=1}^n \eta_i(l_i + b) + \eta_{n+1}b$.

D'autre part, $b \in S(Y)$, alors il existe $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{R}$ tels que $b = \sum_{i=1}^n \alpha_i l_i$.

Si on note $\delta = \sum_{i=1}^n \eta_i$, il suit que

$$t = \sum_{i=1}^n \eta_i l_i + (\delta + \eta_{n+1})b = \sum_{i=1}^n \eta_i l_i + (\delta + \eta_{n+1}) \sum_{i=1}^n \alpha_i l_i = \sum_{i=1}^n l_i [\eta_i + (\delta + \eta_{n+1})\alpha_i].$$

pour $\xi_i = \eta_i + (\delta + \eta_{n+1})\alpha_i, 1 \leq i \leq n$, on obtient

$$t = \sum_{i=1}^n \xi_i l_i$$

Alors

$$t \in S(Y).$$

Comme conséquence, on a

$$S(Z_Y) \subset S(Y)$$

Par la combinaison de (i) avec (ii), on conclut que

$$S(Z_Y) = S(Y)$$

3.5 Interprétation géométrique pour la caractérisation des matrices vérifiant $r_s(X) = r_u(X)$

Le théorème 3.3 permet de traduire l'égalité entre le rang usuel et le rang semi non négatif en une inégalité vectorielle (Assertion (ii) du théorème 3.3). La présente sous-section donne une interprétation géométrique des solutions de l'équation 3.8.

Soit $X \in \mathbb{R}^{n \times p}$ et soient $\{c_1, c_2, \dots, c_p\}$ les colonnes de la matrice X . L'enveloppe convexe engendrée par les colonnes de X et notée par $\text{convcol}(X)$ est définie comme suit :

Rang semi non négatif des matrices réelles et sa connexion au rang usuel

$$\text{convcol}(X) = \{\sum_{i=1}^p \lambda_i c_i, \lambda_i \geq 0, 1 \leq i \leq p, \sum_{i=1}^p \lambda_i = 1\}.$$

Le théorème 3.4 cité si-dessous permet de montrer que si les deux rangs (rang semi non négatif et rang usuel) d'une matrice X coïncident, alors le vecteur nul de l'espace vectoriel \mathbb{R}^n est bien séparé de $\text{convcol}(X)$ par un hyperplan, une illustration est présentée dans la figure 4.1.

Théorème 3.4

Les deux assertions suivantes sont équivalentes :

(i) $r_s(X) = r_u(X)$,

(ii) $0_n \notin \text{convcol}(X)$,

où 0_n est le vecteur nul dans \mathbb{R}^n .

Preuve du théorème 3.4

(i) \Rightarrow (ii) : Soit $X \in \mathbb{R}^{n \times p}$ avec $r_s(X) = r_u(X)$.

Par le théorème (1.3), il existe $\alpha \neq 0_n$ tel que $X^t \alpha > 0_p$, ce qui implique que $\forall 1 \leq j \leq p, c_j^t \alpha > 0$.

Soient $\lambda_1, \lambda_2, \dots, \lambda_p$ sont des scalaires vérifiant $\sum_{j=1}^p \lambda_j = 1$, alors :

$$\sum_{j=1}^p \lambda_j (c_j^t \alpha) > 0 \Leftrightarrow (\sum_{j=1}^p \lambda_j c_j)^t \alpha > 0,$$

donc

$$\forall x \in \text{convcol}(X), x^t \alpha > 0. \tag{3.17}$$

Alors, on conclut que :

$$0_n \notin \text{convcol}(X).$$

(ii) \Rightarrow (i) : Supposons que $0_n \notin \text{convcol}(X)$, par le théorème de la séparation des convexes [8], il existe un hyperplan défini par β tel que $x^t \beta > 0$ pour tout $x \in \text{convcol}(X)$.

En particulier, pour c_j où $1 \leq j \leq p$, nous avons $c_j^t \beta > 0$, ce qui veut dire $X^t \beta > 0_p$.

Alors par le théorème 3.3, on obtient :

$$r_s(X) = r_u(X).$$

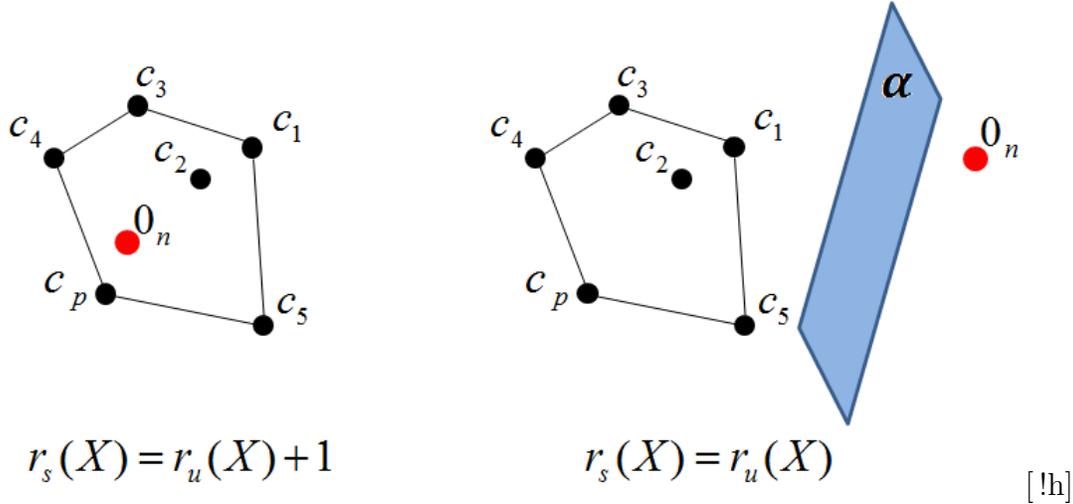


FIGURE 3.1 – Interprétation géométrique du théorème 3.3

3.6 Calcul d'une factorisation semi non négative

Le calcul d'une factorisation semi non négative se fait en trois étapes :

- Calculer une factorisation $X = AB$ avec $A \in \mathbb{R}^{n \times r_u(X)}$, $B \in \mathbb{R}^{r_u(X) \times p}$ en utilisant le procédé de Wedderburn (voir chapitre 1).
- Vérifier avec le procédé de Fourier si une solution du problème $X\alpha > 0$ existe, et si elle existe calculer α .
- Associer à la factorisation de Wedderburn une factorisation semi non négative en utilisant le lemme 2 et le théorème 2.

Le procédé de Fourier-Motzkin

On considère le système d'inéquations S de la forme $X\alpha > 0$ avec :

$X \in \mathbb{R}^{n \times p}$, n est le nombre d'inégalités du système, p est le nombre des variables, et $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_p\}^t$ est le vecteur des variables du système d'inéquations.

L'algorithme de Fourier-Motzkin est basé sur deux phases objectives : la phase de descente qui permet de tester l'existence du vecteur α , et la phase de remontée qui

Rang semi non négatif des matrices réelles et sa connexion au rang usuel

permet de calculer la solution si elle existe.

Phase de descente

On considère la première colonne de la matrice X , ainsi le premier élément du vecteur α , le système S s'écrit comme :

$$(S) \begin{cases} X_{k,1}\alpha_1 + X_{k,2}\alpha_2 + \dots + X_{k,p}\alpha_p > 0 \\ \text{avec } 1 \leq k \leq n \end{cases}$$

Si on considère le premier élément du vecteur α , le système S sera composé de trois sous systèmes : I_0, I_+ , et I_- , si on note l'inéquation d'ordre k par C_k , on a

$$(S) \Leftrightarrow \begin{cases} C_k \in I_+ \Leftrightarrow X_{k,1} > 0 \\ C_k \in I_- \Leftrightarrow X_{k,1} < 0 \\ C_k \in I_0 \Leftrightarrow X_{k,1} = 0 \end{cases}$$

L'ensemble des contraintes dans I_+ nous donne la borne inférieure de α_1 , ainsi que l'ensemble des contraintes dans I_- nous donne la borne supérieure, i.e.,

$$\alpha_1 > \max_{C_k \in I_+} \left(-\frac{X_{k,2}\alpha_2 + \dots + X_{k,p}\alpha_p}{X_{k,1}} \right) \text{ et } \alpha_1 < \max_{C_k \in I_-} \left(\frac{X_{k,2}\alpha_2 + \dots + X_{k,p}\alpha_p}{X_{k,1}} \right).$$

Dans l'ensemble I_0 , l'élément α_1 est déjà éliminé, le système S est équivalent donc à S' tel que

$$(S') \begin{cases} \max_{C_k \in I_+} \left(-\frac{X_{k,2}\alpha_2 + \dots + X_{k,p}\alpha_p}{X_{k,1}} \right) < \max_{C_k \in I_-} \left(\frac{X_{k,2}\alpha_2 + \dots + X_{k,p}\alpha_p}{X_{k,1}} \right) \\ X_{k,2}\alpha_2 + X_{k,3}\alpha_3 + \dots + X_{k,p}\alpha_p > 0, \text{ pour } C_k \in I_0 \end{cases}$$

On poursuit jusqu'à l'élimination de toutes les variables, après p itérations, on trouve soit $\alpha_p > 0$ ou bien $\alpha_p < 0$, dans ce cas la solution existe et on passe à la phase suivante, mais si on trouve la contradiction alors on dit que le système n'admet pas de solutions.

Phase de remontée

A la fin de la phase de descente, on choisit α_p vérifiant la contrainte de résultat, et on remonte en choisissant α_i pour $i = p - 1, p - 2, \dots, 1$ et on obtient le vecteur α tel que $X^t\alpha > 0$.

Factorisation semi non négative : Partie algorithmique

Sommaire

4.1	Approximation matricielle à rang semi non négatif inférieur	78
4.2	Les algorithmes proposés pour le Semi-NMF	79
4.2.1	L’algorithme à mise à jour multiplicative	79
4.2.2	L’algorithme à mise à jour descente du gradient	83
4.3	Le problème d’initialisation pour les algorithmes Semi-NMF	84
4.4	Initialisation du Semi-NMF basée sur l’algorithme génétique	85
4.4.1	Aperçu générale sur les algorithmes génétiques	85
4.4.2	Un algorithme d’initialisation basé sur l’algorithme génétique pour la factorisation semi non-négative	92

Dans ce chapitre, on va présenter le problème **Semi-NMF** comme il a été proposé la première fois, il s'agit d'un problème d'approximation matricielle sous contraintes, où deux versions d'algorithmes ont été proposées. Nous ferons le point sur le problème d'initialisation pour les algorithmes **Semi-NMF** et nous illustrerons les différentes méthodes d'initialisation existants.

Une deuxième partie de ce chapitre est consacré pour démontrer l'intérêt de l'occurrence d'un algorithme génétique pour l'initialisation du **Semi-NMF**, ce qui fait l'objectif de l'article [15].

4.1 Approximation matricielle à rang semi non négatif inférieur

La factorisation semi non négative (**Semi-NMF**) est désignée pour des matrices réelles pas nécessairement non négatives, i.e., $X \in \mathbb{R}^{n \times p}$. Dans le modèle **Semi-NMF**, la matrice X sera approximée en produit de deux matrices A et B où la matrice A peut contenir des composantes avec des signes mixtes mais la matrice B ne doit contenir que des composantes non négatives, i.e., le problème **Semi-NMF** a la forme suivante :

$$X \approx AB, A \in \mathbb{R}^{n \times k}, B \in \mathbb{R}_+^{k \times p} \quad (4.1)$$

avec k est un entier naturel donné. Ce modèle a été introduit pour la première fois par Chris Ding et al en 2006 [23] dans une perspective de la classification automatique des données (Data clustering).

Effectivement, effectuer un k-means sur une matrice des données $X \in \mathbb{R}^{n \times p}$ avec un entier naturel fixé k (souhaité le nombre des classes), consiste à chercher deux matrices $A = [a_1, a_2, \dots, a_k]$ (appelée matrices des centroides des classes) et $B = [b_1, b_2, \dots, b_k]^t$ (appelée matrice des indicateurs d'appartenance aux classes), avec $b_j = [b_{ij}]$, $b_{ij} = 1$ si la $i^{\text{ème}}$ ligne de X appartient à la classe j , et $b_{ij} = 0$ sinon, ceux qui sont solutions du problème suivant :

$$\text{minimiser } \| X - AB \|_F^2 \quad (4.2)$$

où $\| \cdot \|$ désigne la norme de Frobenius.

Pour plus de détails sur le k-means, voir préliminaires.

En effet, pour le modèle (1), lors de regroupement des colonnes de la matrice des données X , les colonnes de la matrice A peuvent être vues comme les centroïdes des classes et les colonnes de la matrice B désignent les indicateurs d'appartenance aux classes.

Par conséquent, la contrainte de non négativité sur la matrice B est relaxée telle que l'approximation AB est plus serrée et les résultats seront beaucoup plus interprétables. Le modèle **Semi-NMF** a été utilisé avec succès dans plusieurs applications telles que la segmentation des mouvements [62], la super-résolution d'images [6], et dans l'hyper-spectral unmixing [80].

4.2 Les algorithmes proposés pour le Semi-NMF

Pour résoudre le problème 1.2, il existe deux algorithmes d'optimisation qui sont basés sur des méthodes de types mise à jour multiplicative et moindres carrés non négatives, l'un de ces algorithmes est posé par C. Ding (2010) dans le papier original [23], et l'autre par N. Gillis (2014) [33]. D'une façon générale, le principe suivi par les deux algorithmes est le même, c'est de chercher la matrice A en fixant la matrice B , et chercher B en fixant A .

4.2.1 L'algorithme à mise à jour multiplicative

Algorithme 1 MU

i) Étant données : $X \in \mathbb{R}^{n \times p}$, et une initialisation $B \in \mathbb{R}_+^{k \times p}$; *ii*) Calculer les éléments de la matrice A en utilisant la règle de mise à jour :

$$A = XB^t(BB^t)^{-1}$$

iii) Calculer les éléments de la matrice B terme par terme en utilisant la procédure :

$$B_{ij} \leftarrow B_{ij} \sqrt{\frac{(X^t A)_{ij}^+ + [B(A^t A)^-]_{ij}}{(X^t A)_{ij}^- + [B(A^t A)^+]_{ij}}} \quad (4.3)$$

où

$$[Y_{ij}]^+ = (|Y_{ij}| + Y_{ij})/2; [Y_{ij}]^- = (|Y_{ij}| - Y_{ij})/2 \quad (4.4)$$

Notons que dans *ii*), $BB^t \in \mathbb{R}_+^{\overline{1} \times \overline{1}}$ est une matrice semi définie positive, donc son inverse est triviale. Dans le cas où BB^t est une matrice singulière, on prend la pseudo-inverse.

Dans la figure citée ci-dessous, on va illustrer selon la règle de mise à jour présentée dans *iii*), la décroissance du critère de la fonction objective.

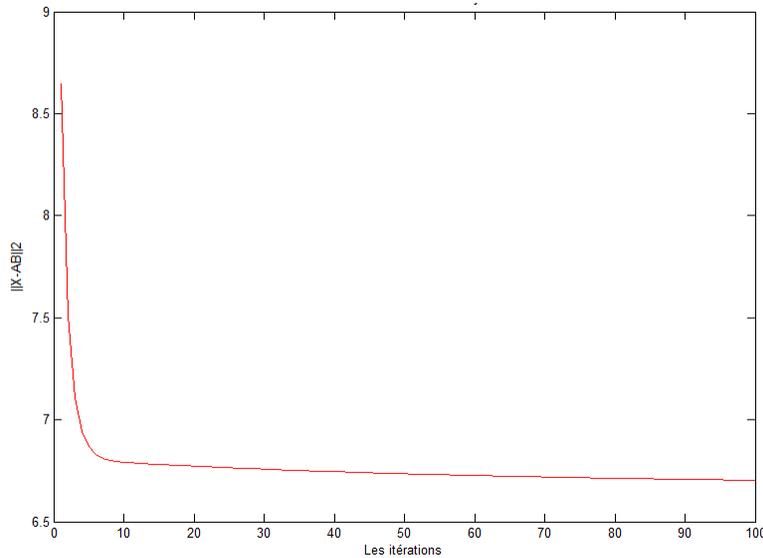


FIGURE 4.1 – La décroissance du critère selon la règle de mise à jour multiplicative

Nous allons démontrer par la suite le théorème 4.1 suivant :

Théorème 4.1

Pour le problème d'optimisation suivant :

$$J(A, B) = \min \| X - AB \|^2 \text{ avec } A \in \mathbb{R}^{n \times k}, B \in \mathbb{R}_+^{k \times p} \quad (4.5)$$

Nous avons les deux résultats suivants :

(i) Si nous fixons B, une solution optimale pour le problème 4.2 est donnée par la règle

de mise à jour de A suivante :

$$A = (XB)(B^tB)^{-1} \quad (4.6)$$

(ii) Si nous fixons A , la fonction objective J diminue de façon monotone sous la règle de mise à jour de B suivante :

$$B_{ij}^{(t+1)} = B_{ij}^{(t)} \sqrt{\frac{(X^tA)_{ij}^+ + [B(A^tA)^-]_{ij}}{(X^tA)_{ij}^- + [B(A^tA)^+]_{ij}}} \quad (4.7)$$

Preuve du théorème 4.1

(i) :

Le problème d'optimisation 4.2 peut être écrit sous la forme suivante :

$$J(A, B) = \text{Tr}(X^tX - 2X^tAB + B^tA^tAB)$$

si nous fixons B , le problème 4.2 devient un problème d'optimisation convexe et la solution peut être obtenue tout simplement par :

$$\frac{\partial J(A, B)}{\partial A} = -2XB^t + 2ABB^t = 0,$$

ce qui veut dire :

$$A = (XB^t)(BB^t)^{-1}.$$

(ii) :

Pour démontrer la partie (i), on fixe A et on résout le problème d'optimisation $J(B)$ avec la contrainte $B \geq 0$, ce qui est un problème d'optimisation avec contraintes.

La preuve se déroulera dans deux étapes : 1^{ère} **étape** :

Avant de montrer l'assertion (ii), il faut vérifier d'abord que la solution limite pour la règle de mise à jour 4.3 satisfait les conditions d'optimisation **KKT**.

En effet, la fonction de Lagrange pour le problème d'optimisation 4.2 est donnée par :

$$L(B) = \text{Tr}(-2X^tAB + B^tA^tAB - \beta B), \quad (4.8)$$

où les multiplicateurs de Lagrange $[\beta_{ij}]_{1 \leq i \leq k, 1 \leq j \leq p}$ imposent la contrainte de non négativité $b_{ij} \geq 0$ avec $B = [b_{ij}]_{1 \leq i \leq k, 1 \leq j \leq p}$.

La condition d'annulation du gradient donne :

$$\frac{\partial L}{\partial B} = -2X^t A + 2B^t A^t A - \beta = 0$$

d'où

$$-2X^t A + 2B^t A^t A = \beta \quad (4.9)$$

En multipliant l'équation 4.9 par B , nous avons pour $1 \leq i \leq k, 1 \leq j \leq p$:

$$(-2X^t A + 2B A^t A) B_{ij} = \beta_{ij} B_{ij} \quad (4.10)$$

d'après la condition complémentaire de négligence, on a :

$$(-2X^t A + 2B A^t A)_{ij} B_{ij} = \beta_{ij} B_{ij} = 0. \quad (4.11)$$

L'équation 4.11 est une équation du point fixe.

D'autre part, pour l'équation donnée par la règle de mise à jour 4.3, on a après la convergence : $B^\infty = B^{(k+1)} = B^{(k)} = B$, i.e.,

$$B_{ij} = B_{ij} \sqrt{\frac{(X^t A)_{ij}^+ + [B(A^t A)^-]_{ij}}{(X^t A)_{ij}^- + [B(A^t A)^+]_{ij}}} \quad (4.12)$$

Notons que $A^t A = (A^t A)^+ - (A^t A)^-$ et $A^t X = (A^t X)^+ - (A^t X)^-$, l'équation 4.12 se réduit à

$$(-2X^t A + 2B A^t A)_{ij} B_{ij}^2 = 0 \quad (4.13)$$

En effet, l'équation 4.13 est identique à l'équation 4.11.

2^{ème} étape

Elle concerne la démonstration de la convergence de l'algorithme, vous pouvez consulter [?]

4.2.2 L'algorithme à mise à jour descente du gradient

Gillis et al. ont proposé dans le papier [33] un algorithme simple qui est basé sur une descente du gradient qui optimise la fonction objective $J(A, B)$ définie dans 4.2 d'une manière alternative sur la matrice A pour une matrice B fixée, et sur la matrice B pour une matrice A fixée.

Il s'agit donc de diviser le problème 4.2 en deux sous problèmes suivants :

$$J(A) = \min_{A \in \mathbb{R}^{n \times k}} \|X - AB\|_F^2 \quad (4.14)$$

$$J(B) = \min_{B \in \mathbb{R}^{k \times p}, B \geq 0} \|X - AB\|_F^2 \quad (4.15)$$

De la même façon que Ding et al. ont procédé, le problème d'optimisation 4.14 est un problème des moindres carrés sans contraintes qui peut être résolu avec la méthode d'annulation du gradient traditionnelle.

Le problème d'optimisation 4.15 est considéré comme un problème des moindres carrés non négatives. Pour résoudre ce problème, Gillis a proposé d'utiliser une méthode de la descente du gradient (voir section dans chapitre 2) appliquée sur les lignes de la matrice B . Plus précisément, la matrice B est mise à jour d'une façon que la fonction objective $J(B)$ se réduit itérativement. i.e., dans une itération t et pour $A^{(t)}$ et $B^{(t)}$ données, on cherche $B^{(t+1)} \geq 0$ telle que :

$$J(B^{(t+1)}) = \|X - A^{(t)}B^{(t+1)}\|^2 < \|X - A^{(t)}B^{(t)}\|^2 = J(B^{(t)})$$

L'algorithme de Gillis est implémenté et il garanti la convergence vers un point stationnaire pour le problème 4.2.

Algorithme 2 PG

i) Étant donné $X \in \mathbb{R}^{n \times p}$, une initialisation $B \in \mathbb{R}_+^{k \times p}$, et un nombre maximal d'itérations (*maxiter*);

ii) pour $i = 1 : \text{maxiter}$

$$A \leftarrow \arg \min_{Y \in \mathbb{R}^{n \times k}} \|X - YB\|_F^2 \quad (= X/B \text{ en Matlab});$$

pour $i = 1 : k$

$$\begin{aligned}
 B_{(i,:)}^t &\leftarrow \underset{y \in \mathbb{R}_+^p}{\operatorname{argmin}} \|X - A(:, J)B(J, :) - A(:, i)y^t\|_F^2 \\
 &= \max \left(0, \frac{[X - A(:, J)B(J, :)]^t A(:, i)}{\|A(:, i)\|_2^2} \right)
 \end{aligned}$$

fin ;

fin ;

4.3 Le problème d’initialisation pour les algorithmes Semi-NMF

Chacun des algorithmes (proposés par Ding et Gillis) peut par ailleurs être initialisé de différentes façons :

- plusieurs initialisations aléatoires de A et B , le meilleur ajustement est conservé [23],
- non-negative double singular value decomposition (NNSVD) [33],
- une classification (k-means) des lignes ou des colonnes [23].

Nous allons illustrer dans l’exemple ci dessous simulé en Matlab comment l’algorithme proposé par Ding ou bien celui de Gillis sont des algorithmes itératives qui dépendent de la solution initiale :

Exemple 1

Soit X est la matrice dans $\mathbb{R}^{3 \times 3}$ suivante :

$$X = \begin{pmatrix} -1 & 2 & -3 \\ 6 & 2 & 6 \\ -9 & -4 & 9 \end{pmatrix}$$

On veut calculer la factorisation semi non négative à un rang $r = 2$, donc soient deux initialisations aléatoires $B_1 \in \mathbb{R}_+^{2 \times 3}$ et $B_2 \in \mathbb{R}_+^{2 \times 3}$:

$$B_1 = \begin{pmatrix} 0 & 3 & 4 \\ 4 & 0 & 1 \end{pmatrix} \text{ et } B_2 = \begin{pmatrix} 4 & 0 & 4 \\ 0 & 4 & 1 \end{pmatrix}$$

Nous tournons l'algorithme **Semi-NMF** à 1000 itérations avec les deux initialisations B_1 et B_2 et on calcul l'écart $\|\tilde{X}_1 - \tilde{X}_2\|$ où \tilde{X}_1 est la matrice d'approximation obtenue d'après l'initialisation B_1 et \tilde{X}_2 est la matrice d'approximation obtenue d'après l'initialisation B_2 , nous avons :

$$\|\tilde{X}_1 - \tilde{X}_2\|_F = 0.1546$$

d'où \tilde{X}_1 est significativement différente du \tilde{X}_2 , alors pour deux initialisations B_1 et B_2 , nous n'avons pas obtenu la même solution, ce qui veut dire que l'algorithme dépend de la solution initiale.

4.4 Initialisation du Semi-NMF basée sur l'algorithme génétique

4.4.1 Aperçu générale sur les algorithmes génétiques

La résolution d'un problème d'optimisation consiste à explorer un espace de recherche afin de maximiser (ou minimiser) une fonction donnée. Les complexités (en taille ou en structure) relatives de l'espace de recherche et de la fonction à maximiser conduisent à utiliser des méthodes de résolutions radicalement différentes. En première approximation, on peut dire qu'une méthode déterministe est adaptée à un espace de recherche petit et complexe et qu'un espace de recherche grand nécessite plutôt une méthode de recherche stochastique (recuit simulé, algorithme génétique, ...).

Les Algorithmes Génétiques (**AGs**) représentent une famille assez riche et très intéressante d'algorithmes d'optimisation stochastique fondées sur les mécanismes de la sélection naturelle et de la génétique. Les champs d'application sont fort diversifiés. On les retrouve aussi bien en théorie des graphes qu'en compression d'image numérisée ou encore en programmation automatique. Les raisons de ce nombre d'application sont claires. Lors principe est d'opérer une recherche stochastique sur un espace important à travers un ensemble \mathcal{Z} une population z de pseudo-solutions. Ces algorithmes sont

simples et très performants dans leur recherche d'amélioration. De plus, ils ne sont pas limités par des hypothèses contraignantes sur le domaine d'exploration. Ainsi, le mathématicien abordant le sujet n'a guère à se préoccuper de la continuité et de la différentiabilité de la fonction à optimiser. Les algorithmes génétiques diffèrent fondamentalement des autres méthodes d'optimisation selon quatre axes principaux :

- (1) Les **AGs** utilisent un codage des paramètres, et non les paramètres eux mêmes.
- (2) Les **AGs** travaillent sur une population de points, au lieu d'un point unique.
- (3) Les **AGs** n'utilisent que des valeurs de la fonction étudiée, pas sa dérivée, ou une autre connaissance auxiliaire.
- (4) Les **AGs** utilisent des règles de transition probabilistes, et non déterministes.

Afin de bien appréhender la dynamique des algorithmes génétiques, introduisant les principaux éléments du jargon utilisé dans la littérature.

- * Individu ou Chromosome : une solution potentielle.
- * Population : un ensemble de chromosomes ou de points de l'espace de recherche.
- * Environnement : l'espace de recherche.
- * Fitness ou Fonction d'évaluation : la fonction-positive-que l'on recherche à maximiser.

Ce petit lexique, est fait avec prudence analogie avec la biologie, a naturellement une simple fonction métaphorique. Leur emploi n'obéit qu'à un souci pragmatique. Nous les utilisons dans l'introduction de l'algorithme génétique standard, considéré désormais par maintes auteurs sous le nom d'Algorithme Génétique Canonique (**AGC**), défini par Holland (1975) [?] et redynamisés par Goldberg (1989) [35].

Algorithme génétique standard

Avant d'examiner les mécanismes et la puissance d'un algorithme génétique standard, présentons clairement l'objectif visé. Le problème des **AGs** peut être résumé de la manière suivante : On dispose d'une fonction f , non constante, définie sur le cube logique $\{0, 1\}^n$, n est un entier naturel, et on se fixe pour objectif de localiser l'ensemble f^* de maxima globaux de f , ou à défaut, de trouver le plus rapidement et le plus efficacement possible des points sous-optimaux. Formellement, il s'agit d'une optimisation

sans contraintes du type :

$$\max_{x \in \{0,1\}^n} f(x)$$

Très souvent, comme dans notre problème, on s'intéresse plutôt aux minima d'une fonction f , dans ce contexte, les résultats suivants sont utilisés :

Lemme 4.1

Soit f une fonction réelle d'une ou de plusieurs variables. Les problèmes d'optimisation suivants sont équivalents :

- (i) $\min_x f(x)$
- (ii) $\max_x (-f(x))$

De plus, la fonction optimisée par un **AG** doit avoir des valeurs positives dans l'espace de recherche. Dans le cas contraire, il est nécessaire d'ajouter aux valeurs de f une constante positive adéquate C conformément à l'équivalence des problèmes d'optimisation suivants :

- $\max_x f(x)$
- $\max_x (f(x) + C), C \in R$

En définitive, l'objectif de l'**AG** est de régler les différents paramètres d'un problème donné pour obtenir la plus grande valeur possible de la fonction objectif f .

L'organigramme dans la figure 4.2 montre clairement le fonctionnement de l'**AG** et indique ses différents opérateurs de base. Ces opérateurs sont inspirés directement des mécanismes de la sélection naturelle et des phénomènes génétiques. Ainsi, le principe de l'**AG** (ce qui fait d'ailleurs justifié son nom) est de faire évoluer une population de façon à adapter les individus à l'environnement en place, comme le ferait un ensemble d'êtres vivants. Ce principe s'apparente convenablement à celui de la théorie de Darwin [72] sur l'évolution, selon laquelle "la vie est une compétition et seuls les mieux adaptés survivent et se reproduisent".

Techniquement, comme il est montré dans la figure 4.2, une nouvelle génération s'obtient généralement au bout d'un cycle composé de trois opérateurs standards : reproduction, croisement et mutation.

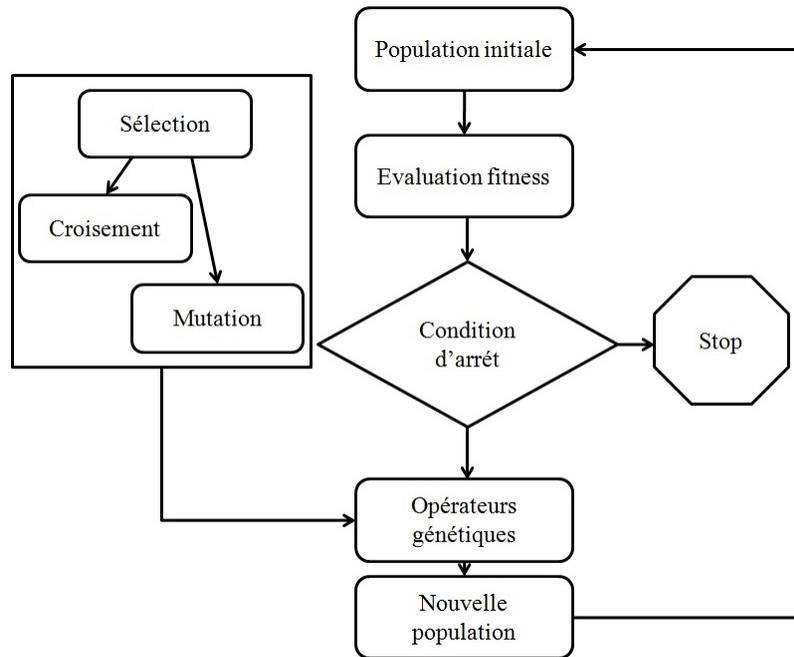


FIGURE 4.2 – Organigramme de l'AG

1) Reproduction

L'opérateur de reproduction est directement inspiré de la sélection naturelle, il consiste à dupliquer chaque individu de la population proportionnellement à sa valeur d'adaptation. Il s'agit en quelque sorte, de réaliser autant de tirages avec remise à la manière de la technique de Bootstrap, qu'il y a d'éléments dans la population. Chaque élément étant tiré avec une probabilité liée à sa valeur d'évaluation que le statisticien peut bien interpréter comme des probabilités de survie. Ainsi, les individus ayant une grande valeur de la fonction fitness ont plus de chance d'être sélectionnés pour la génération suivante. On parle alors de sélection proportionnelle.

L'opérateur de reproduction peut être mise en oeuvre sous forme algorithmique de différentes façons. La plus facile est de créer une *roue de loterie biaisée* (figure 4.3) pour laquelle chaque individu de la population courante occupe une section de roue proportionnelle à son adaptation.

Pour réaliser la reproduction, il suffit de faire tourner la roue biaisée ainsi définie autant

de fois qu'on a besoin de descendants. Le groupe ainsi obtenu constitue l'ébauche de la nouvelle génération qui est ensuite soumise à d'autres opérateurs génétiques.

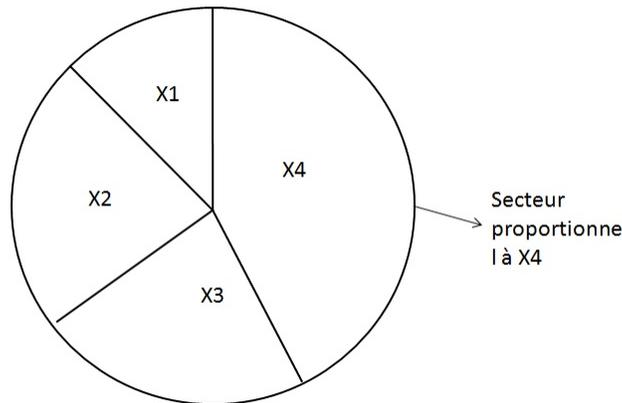


FIGURE 4.3 – Roue de loterie biaisée

Étant donné un fitness positif f et une taille de la population n , une telle reproduction peut être décrite schématiquement par le pseudo code suivant :

Algorithme : roue biaisée

Début

-Évaluer le fitness f_i de chaque individu X_i

-Calculer F le fitness total de la population : $F = \sum_{i=1}^n f_i$

Pour $i := 1 : n$

$p_i = f_i/F$ Probabilité de sélection de X_i

$q_i = 0$

Pour $j := 1 : i$

$q_i = q_i + p_j$

fin pour

Pour $i := 1 : n$

\tilde{p} réel tiré dans $[0, 1]$

Si $\tilde{p} < q_i$

Sélectionner X_i

Sinon Sélectionner $X_k, 2 \leq k \leq n$ tel que $q_{k-1} < \tilde{p} < q_k$

Fin si

Fin pour

Fin

L'inconvénient majeur de ce type de reproduction vient du fait qu'elle peut favoriser

la domination d'un individu ne représentant pas véritablement le meilleur . De plus, elle peut aussi engendrer une perte de diversité par la domination d'un super-individu. Ce qui ne donne pas une bonne population. Pour pallier cet inconvénient, on préfère très souvent des méthodes de reproduction plus justes qui n'autorisent en aucun cas de naissance d'un super-individu. Une solution de compromis peut éviter les inconvénients de l'utilisation de la sélection proportionnelle est l'utilisation de la *sélection par rangement* : Le principe de cette méthode consiste à ranger les individus de la population de la génération courante par ordre croissant de leurs évaluations (fitness), respectivement par ordre décroissant de leurs évaluation pour un problème de maximisation. Puis attribuer à tout individu i , une probabilité p_i de participation dans la génération suivante, selon son rang :

$$p_i = \frac{\varphi}{n} - (r_i - 1) \times \frac{(2\varphi - 2)}{n(n-1)},$$

tel que le nombre moyen d'enfants de l'individu i est $n \times p_i$ où :

n est la taille de la population.

r_i : le rang de l'individu i .

φ : est la pression de la sélection, $\varphi \in [1, 2]$ qui représente le nombre moyen d'individus enfants du meilleur individu.

Cette méthode de sélection évite les inconvénients de la sélection proportionnelle et représente un changement d'échelle dynamique.

Après la reproduction, un croisement permet de générer les nouveaux individus.

2) Croisement

Relativement à deux individus "parents", considérés à cette occasion comme des vecteurs, la dynamique de l'opérateur de croisement, inspiré directement du processus de multiplication des chromosomes humains, consiste à générer deux nouveaux individus "enfants" de la façon suivante :

On choisit au hasard deux parents dans la population intermédiaire p' (ayant subit la reproduction) et un site de croisement (un nombre de 1 à l , l est la longueur du chromosome), soit k l'indice du site, on obtiendra alors deux enfants en prenant d'une part les k premiers gènes du premier parent et les $(l - k)$ derniers gènes du deuxième parent. On

note que l'opérateur de croisement est appliqué avec une probabilité p_c . Cette méthode est appelée *méthode de croisement à un site*.

Il existe d'autres types de croisements comme le croisement à deux sites où on échange les segments ainsi délimités entre les deux parents. L'avantage de cet opérateur est de parcourir plus rapidement l'ensemble des solutions en permettant de modifier plus facilement les allèles (bits) des deux bouts d'un chromosome.

a) Croisement multi-sites

Plus généralement, un croisement multi-sites consiste à choisir au hasard k sites de croisement, et inter-changer les parties correspondantes des deux parents choisis aléatoirement.

Schématiquement, un opérateur de croisement à 4-sites de croisement est illustré dans la figure 4.4.

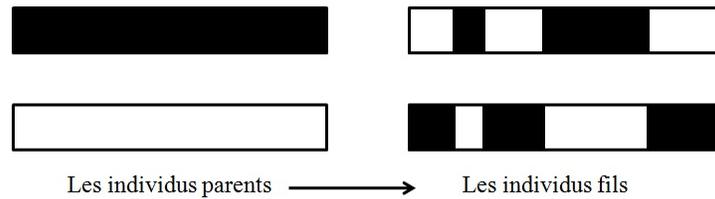


FIGURE 4.4 – L'opérateur de croisement à 4 sites

b) Croisement uniforme

Soit r_i un nombre appartenant à l'intervalle $[0, 1]$.

L'application d'un opérateur de croisement uniforme sur les deux individus parents A et B tels que : $A = (a_1, a_2, \dots, a_i, \dots, a_n)^t$, $B = (b_1, b_2, \dots, b_i, \dots, b_n)^t$ donne les deux individus fils A' , B' tels que : $A' = (a'_1, a'_2, \dots, a'_i, \dots, a'_n)^t$, $B' = (b'_1, b'_2, \dots, b'_i, \dots, b'_n)^t$ qui sont obtenus de la manière suivante :

$$a'_i = \begin{cases} a_i, & r_i \leq 0.5 \\ b_i, & r_i > 0.5 \end{cases} \quad b'_i = \begin{cases} a_i, & r_i > 0.5 \\ b_i, & r_i \leq 0.5 \end{cases}$$

3) Mutation

L'opération de mutation consiste à modifier de manière aléatoire, avec une probabilité

généralement faible, la valeur d'une composante de l'individu. Dans le cas d'un codage binaire de l'AG [71], un booléen $\omega \in \{0, 1\}$ choisi aléatoirement est remplacé par son inverse $1 - \omega$. Techniquement, la combinaison de la reproduction et du croisement est suffisante pour assurer l'évolution de la population, mais il peut arriver que certaines informations essentielles (composantes dans le cas de vecteurs) disparaissent de la population. Le rôle premier de la mutation est donc de pallier à cet inconvénient. Prise isolément, la mutation constitue une exploration aléatoire dans l'espace des chaînes. Mais une utilisation parcimonieuse avec la reproduction et le croisement constitue une police d'assurance protégeant de la perte prématurée de composantes importantes. Traditionnellement, la mutation est appliquée aux enfants avec une probabilité faible comprise entre 0.01 et 0.1 pour empêcher une convergence locale de l'algorithme génétique, c'est-à-dire pour éviter que l'algorithme génétique converge vers un minimum local.

Il existe plusieurs variantes de l'AG génétique standard. Les différences sont principalement dans : la sélection, le croisement, la mutation et la stratégie de remplacement [40].

4.4.2 Un algorithme d'initialisation basé sur l'algorithme génétique pour la factorisation semi non-négative

Comme il a été montré dans la section précédente, le problème d'initialisation dans les méthodes de factorisation non-négative a été traité dans la littérature par plusieurs chercheurs, pour la factorisation semi non-négative, trois méthodes d'initialisation sont décrits, et chaque méthode a été examinée indépendamment. Cette section présente l'algorithme d'initialisation pour le **Semi-NMF** basé sur l'algorithme génétique, ainsi qu'une étude comparative basée sur un exemple réel avec les autres initialisations.

Les algorithmes génétiques sont largement utilisés dans le domaine de la factorisation matricielle. Une version initiale des algorithmes génétiques a été adaptée pour la factorisation des matrices binaires [70]. L'article [46] présente l'application de cinq algorithmes basés sur l'évolution des populations (où l'algorithme génétique apparaît)

comme nouvelles variantes d'initialisation pour la factorisation non négative **NMF**. Et aussi, nous pouvons trouver dans [68] un algorithme de factorisation non négative amélioré basé sur l'algorithme génétique.

Le but de notre algorithme d'initialisation est de trouver heuristique-ment les points de départ optimaux pour les colonnes de la matrice à valeurs non-négatives demandée pour la factorisation semi non négative afin de minimiser la fonction objective suivante :

$$f(B) = \|X - AB\|_F^2 \quad \text{sous} \quad B \in \mathbb{R}_+^{k \times p} \quad (4.16)$$

où $X \in \mathbb{R}^{n \times p}$ et $A \in \mathbb{R}^{n \times k}$.

L'objectif est de trouver une solution avec moindre erreur lors de la convergence et/ou d'accélérer la convergence du **Semi-NMF** (c'est à dire, une erreur d'approximation plus petite pour un nombre donné d'itérations).

L'algorithme d'initialisation pour le **Semi-NMF** se déroule de la manière suivante :

Au lieu de chercher tous les éléments de la matrice B définie dans 4.16, nous optimisons séquentielle-ment chaque colonne de la matrice B . Cela permet un appel vers l'algorithme génétique standard autant de fois que le nombre des colonnes de la matrice originale X . Dans ce cas, l'**AG** va chercher dans l'espace \mathbb{R}_+^p le meilleur vecteur $B_{:j}$ qui minimise la fonction fitness suivante :

$$f(B_{:j}) = \|X_{:j} - AB_{:j}\|_F^2 \quad 1 \leq j \leq p. \quad (4.17)$$

L'algorithme présenté ci-dessous montre notre procédure d'initialisation.

Algorithme 3

Étant donnés : $X \in \mathbb{R}^{n \times p}$ et $k \ll \min(n, p)$

(i) Initialiser A_0 aléatoirement ;

(ii) Pour $i = 1$ à p

Utiliser l'**AG** pour minimiser $\|X_{:j} - A_0 B_{:j}\|_F^2$;

$B(:, j) = B_{:j}$;

Finpour

Finalgorithme

Les paramètres d'entrées pour l'algorithme génétique sont : X_j (la $j^{\text{ème}}$ colonne de la matrice originale X) et A_0 (une matrice générée aléatoirement dans $\mathbb{R}^{n \times k}$). le paramètre de sortie est B_j (la $j^{\text{ème}}$ colonne de la matrice à initialiser B). Pour conserver la non négativité des éléments de la matrice à initialisé B , l'algorithme génétique prend des bornes supérieurs et inférieurs pour l'espace de recherche.

Expérience Numérique

Nous présentons dans cette sous-section les résultats d'une expérience simple sur les données Ionosphere UCI [83] (il s'agit d'une matrice de dimension 34×351 qui contiennent des valeurs positives et négatives dans l'intervalle $[-1,1]$) qui vise à vérifier que l'utilisation de l'algorithme génétique dans l'initialisation de l'algorithme **Semi-NMF** jette de lumière supplémentaire à l'algorithme **Semi-NMF**. Nous avons été adaptés à l'implémentation GA existant dans Matlab (ga) pour faire l'algorithme d'initialisation tout en respectant les propriétés de la norme de Frobenius dans la fonction fitness de l'**AG**.

Notre objectif est d'illustrer sur des données réelles, les observations sur l'approximation des erreurs pour trois initialisations **Semi-NMF** (initialisation aléatoire, initialisation k-means et initialisation GA).

Les résultats de notre expérience sont donnés dans les figures 4.5 , 4.6 , et 4.7.

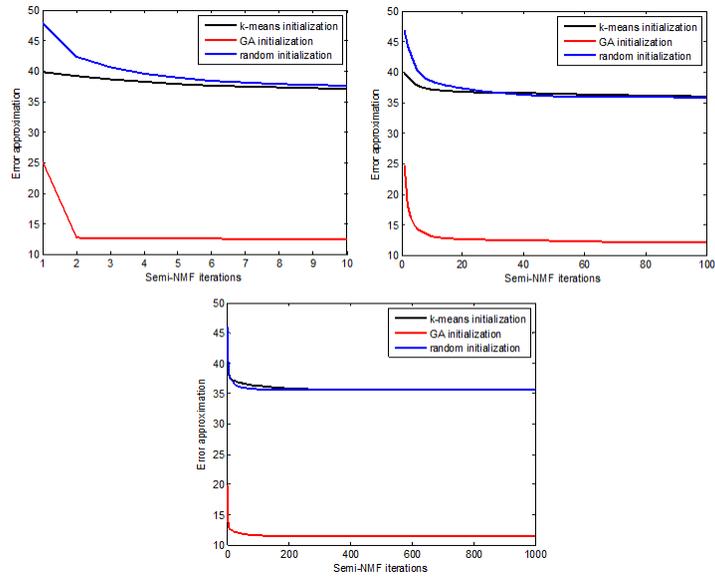


FIGURE 4.5 – Erreur d’approximation du **Semi-NMF** pour différentes initialisations sur les données Ionosphère, $k=5$

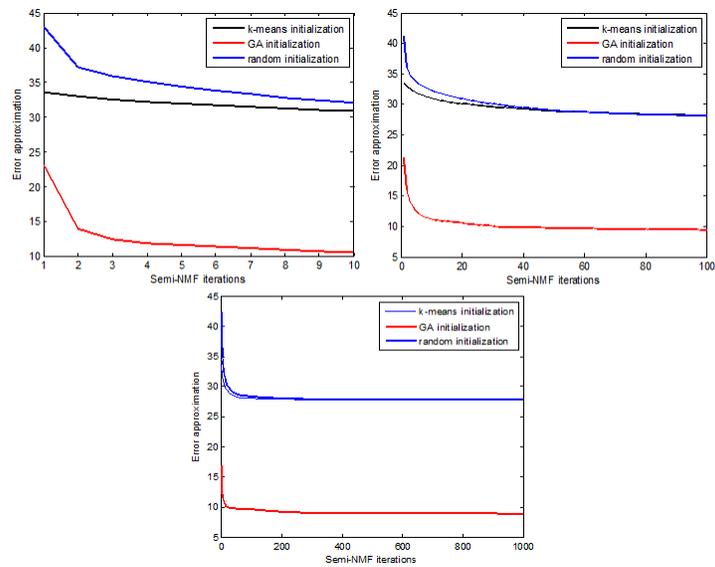


FIGURE 4.6 – Erreur d’approximation du **Semi-NMF** pour différentes initialisations sur les données Ionosphère, $k=10$

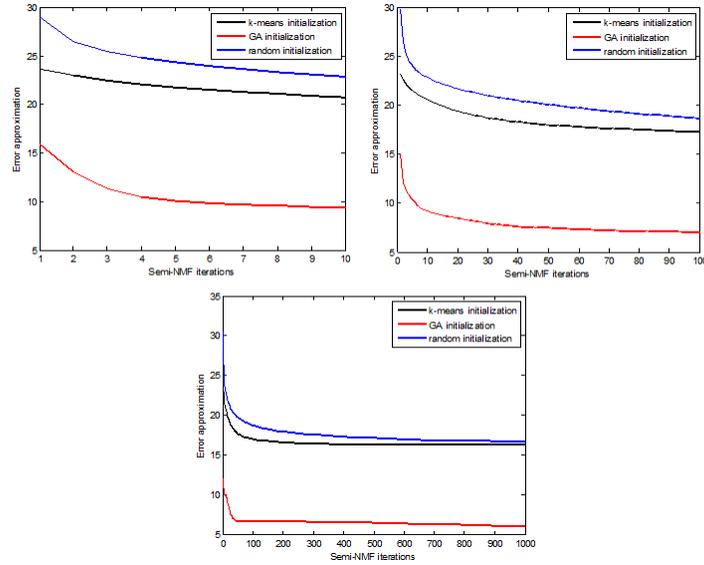


FIGURE 4.7 – Erreur d’approximation du **Semi-NMF** pour différentes initialisations sur les données Ionosphère, $k=20$

Interprétations et commentaires

Nous remarquons que les valeurs de l’erreur d’approximation obtenue par le **Semi-NMF** initialisé par l’algorithme génétique sont les plus petites durant les itérations (Graphes en rouge dans les figures 4.5, 4.6, 4.7), nous avons montré alors qu’une initialisation avec l’algorithme génétique pour la variante de mise à jour multiplicative de l’algorithme **Semi-NMF** donne une fonction d’erreur d’approximation plus faible que les initialisations aléatoires et k-means.

Conclusion Générale

Ce mémoire a exploré un sujet relativement récent : la factorisation en matrices non négatives. On a vu que cette technique permettait à la fois la compression et l'analyse de données positives (images, textes, ...). En effet, en réalisant une approximation à rang inférieur, on obtient une base positive de taille réduite à partir de laquelle on peut reconstituer les données par combinaison linéaire à coefficients positifs.

Le problème de factorisation non négative exacte a été abordé plus en détail : on a montré les résultats théoriques existaient dans la littérature tout en visualisant l'intérêt majeur du concept de rang non négatif associé à une matrice non négative. Puis, nous avons présenté le problème **NMF** comme une approximation à rang non négatif inférieur, qui est un problème d'optimisation sous contraintes dont les conditions d'optimalité de premier ordre **KKT** sont vérifiées, d'où une solution locale existait et représenté par un point stationnaire. Ensuite, nous avons présenté les différents algorithmes qui permettent de calculer une factorisation **NMF**.

Dans cette thèse, un cas particulier des méthodes **NMF** a été pris en charge, nous avons donc présenté le modèle **Semi-NMF** selon les deux aspects : théorie et algorithmes.

Comme contribution de notre thèse, nous avons réussi à introduire un nouveau concept qui est le rang semi non négatif, c'est un concept situé entre le rang non négatif (qui

est un problème pas encore résolu), et le rang usuel (qui fait la base de la méthode d'analyse des données la plus efficace **SVD**). Nous avons établi une relation explicite entre le rang semi non négatif et le rang usuel par le biais d'une condition nécessaire et suffisante. Ces résultats théoriques ouvrent la perspective de la recherche de nouvelles méthodes d'analyse des données qui sont similaires à la **SVD** d'un part, et répondent aux exigences d'une contrainte de non négativité d'un autre part.

Sur l'aspect algorithmique, nous avons choisi à utiliser la globalité des solutions de l'algorithme génétique standard pour faire le plus dans l'initialisation de l'algorithme **Semi-NMF**, il est bien connu que l'algorithme génétique avec le codage réel n'avait pas de preuves purement théoriques, et que nous ne pouvons pas vérifier les conditions d'optimisation pour un algorithme génétique (puisque'il ne tient pas compte de la complexité de la fonction objective), mais dans la réalité, l'algorithme génétique donne de meilleures solutions et il est préférable par les utilisateurs.

Les résultats pratiques que nous avons réussi au cours de notre expérience numérique nous laissent à réfléchir pour une version stochastique pour le problème **Semi-NMF** où nous pouvons définir des opérateurs explicites.

Les publications

M. Chouh, M. Hanafi and K. Boukhetala. Semi-nonnegative rank for real matrices and its connection to the usual rank. *Linear Algebra and its Applications* 466, 27-37, 2015. (Cité 2 fois)

M. Chouh and K. Boukhetala. Semi-nonnegative Matrix Factorization Algorithm Based on Genetic Algorithm Initialization. *International Journal of Machine Learning and Computing* vol. 6, no. 4, pp. 231-234, 2016.

Bibliographie

- [1] L. B. Beasley, T. J. Laffey. Real rank versus nonnegative rank. *Linear Algebra and its applications*, V :432, Iss : 12, pp.2330-2335, 2009.
- [2] M.W Berry and M. Browne. Email surveillance using non-negative matrix factorization. *Computational and Mathematical Organization Theory*, 11 :249-264, octobre 2005.
- [3] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca et R. J. Plemmons, Algorithms and applications for approximate nonnegative matrix factorization, *Computational Statistics and Data Analysis* 52, no 1, 155-173, 2007.
- [4] N. Bertin, R. Badeau and G. Richard. Blind signal decompositions for automatic transcription of polyphonic music : NMF and K-SVD on the benchmark. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages I-65-I-68, Honolulu, Hawaii, USA, Avril 2007.
- [5] N. Bertin, R. Badeau, and E. Vincent. Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3) :538-549, 2010.
- [6] M. Bevilacqua, A. Roumy, C. Guillemot, M. L. Morel. Neighbor embedding based single-image super-resolution using Semi-nonnegative matrix factorization. In : *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1289-1292, 2012.

Bibliographie

- [7] C. Boutsidis, E. Gallopoulos. On SVD-based initialization for nonnegative matrix factorization. Technical Report HPCLAB-SCG-6/08-05, University of Patras, Patras, Greece, 2005.
- [8] S. Boyd, L. Vandenberghe. Convex optimization. Cambridge university press, 2004.
- [9] P. S. Bradley and U. M. Fayyad. Refining initial points for k-means clustering. In Proceedings of the Fifteenth International Conference on Machine Learning, pp. 91–99. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1998.
- [10] R. Bro and S. De Jong. A fast non-negativity constrained least squares algorithm. Journal of Chemometrics, 11(5) :393–401, 1997.
- [11] J. Brunet, P. Tamayo, T.R. Golub, and J.P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. Proceedings of the National Academy of Sciences, 101(12) :4164–4169, 2004.
- [12] S. L. Campbell, G. D. Poole. Computing nonnegative rank factorizations. Linear algebra and App. pp.175-182, 1981.
- [13] D. Chen, R. J. Plemmons. Nonnegativity constraints in numerical analysis. In : Proceedings of the Symposium on the Birth of Numerical Analysis, Leuven Belgium, pp. 109-140, 2009.
- [14] M. Chouh, M. Hanafi, K. Boukhetala. Semi-nonnegative rank for real matrices and its connection to the usual rank. Linear Algebra and its Applications, 466, pp. 27-37, 2015.
- [15] M. Chouh and K. Boukhetala. Semi-nonnegative Matrix Factorization Algorithm Based on Genetic Algorithm Initialization. International Journal of Machine Learning and Computing vol. 6, no. 4, pp. 231-234, 2016.
- [16] M. T. Chu, R. E. Funderlik, and G. H. Golub. A rank-one Reduction Formula and its Applications to Matrix Factorizations. SIAM Rev, 37, pp.512- 530, 1995.
- [17] A. Cichocki, R. Zdunek and S. I. Amari : New algorithms for nonnegative matrix factorization in applications to blind source separation. In International Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 5, pages 621–625, Toulouse, France, Mai 2006.

- [18] A. Cichocki and P. A. Huy. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 92(3) :708-721, 2009.
- [19] A. Cichocki, R. Zdunek, A. H. Phan, S. Amari. *Nonnegative matrix and tensor factorizations : applications to exploratory multi-way data analysis and blind source separation*. Wiley, New York, 2009.
- [20] J. E. Cohen and U. G. Rothblum. Nonnegative Ranks, Decompositions, and Factorizations of Nonnegative Matrices. *Linear Algebra and its applications*, V :190, pp.149-168, 1993.
- [21] P. Comon. Independent component analysis, a new concept. *Signal Processing special issue Higher-Order Statistics*, 36 :287-314, Avril 1994.
- [22] A. Cont. Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical HMMs. In *IEEE International Conference in Acoustics and Speech Signal Processing (ICASSP)*. Toulouse, Mai 2006. URL <http://cosmal.ucsd.edu/arshia/papers/ICASSP06/>.
- [23] C. Ding, T. Li, M. I. Jordan. Convex and Semi- nonnegative matrix factorizations. *Pattern Analysis and Machine Intelligence*. *IEEE V* : 32, Iss : 1, pp. 45-55, 2010.
- [24] Ding, X. He, and H.D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. *Proc. SIAM Data Mining Conf*, 2005.
- [25] C. Ding, T. Li, W. Peng, H. Park. : Orthogonal nonnegative matrix tri-factorizations for clustering. In : *KDD,06 : Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, ACM, 126-135, 2006.
- [26] C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics and Data Analysis*, 52(8) :3913-3927, April 2008.
- [27] B. Dong, M. Lin and M. T. Chu. Nonnegative rank factorization via rank reduction. preprint, 2008

Bibliographie

- [28] K. Drakakis, S. Rickard, R. de Frein, A., Cichoki. Analysis of financial data using non-negative matrix factorization. *International Mathematical Forum* 3(38), 1853-1870, 2008.
- [29] C. Févotte, N. Bertin, and J. Durrieu. Nonnegative matrix factorization with the itakura-saito divergence : With application to music analysis. *Neural Computation*, 21(3) :793-830, 2009.
- [30] V. Franc, V. Hlavac, and M. Navara. Sequential coordinate-wise algorithm for the non-negative least squares problem. In *CAIP 2005 : Computer Analysis of Images and Patterns*, volume 3691, pages 407-414, September 2005.
- [31] J. Gareth. Genetic and evolutionary algorithms. In Paul von Rague, editor, *encyclopedia of computational Chemistry*, John Wiley and Sons, 1998.
- [32] N. Gillis, F. Glineur. On the Geometric Interpretation of the Nonnegative Rank. *Linear Algebra and its applications*, V :437, Iss11, pp.2685-2712, 2012.
- [33] N. Gillis and A. Kumar, Exact and Heuristic Algorithms for Semi-Nonnegative Matrix Factorization, *SIAM J. on Matrix Analysis and Applications* 36 (4), pp. 1404-1424, 2015.
- [34] C. Gobinet, E. Perrin and R. Huez. Application of non-negative matrix factorization to fluorescence spectroscopy. In *European Signal Processing Conference (EUSIPCO)*, Vienne, Autriche, Septembre 2004.
- [35] D.E. Goldberg, B. Korb, and K. Deb. Messy Genetic Algorithms : Motivation, Analysis, and First Results. *Complex Systems* 3, 493-530, 1989.
- [36] G. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5) :403-420, 1970.
- [37] G. Golub and C. Van Loan. *Matrix computations*. Johns Hopkins University Press Baltimore, MD, USA, 1996.
- [38] D. Guillamet, B. Schiele, and J. Vitrià. Color histogram classification using NMF. *Rapport Technique 057*, Centre de Visió Per Computador, Universit  autonome de Barcelone, Octobre 2001.

Bibliographie

- [39] J.A. Hartigan and M.A. Wong. A K-means Clustering Algorithm. *Applied Statistics*, 28 :100{108},1979.
- [40] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, 2nd ed. John Wiley and Sons, 2005.
- [41] N. D., Ho. *Nonnegative matrix factorization algorithms and applications* (Doctoral dissertation, ÉCOLE POLYTECHNIQUE), 2008.
- [42] J. H. Holland. *Adaptation in natural and artificial systems*. Ann Arbor, MI : University of Michigan Press, 1975.
- [43] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6) :471-441, Septembre 1933.
- [44] P.O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research (JMLR)*, 5 :1457-1469, 2004.
- [45] A. K. Jain. *Data Clustering : 50 years beyond K-means*. *Pattern Recognition Letters*, V : 31, Iss : 8, pp. 651-666, 2010.
- [46] A. Janecek and Y. Tan. Using population based algorithms for initializing nonnegative matrix factorization. *Advances in Swarm Intelligence*, pp. 307-316, 2011.
- [47] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.
- [48] S.P. Kim, Y.N. Rao, D. Ergodmus, J.C. Sanchez, M.A.L. Nicolelis, et J.C. Principe. Determining patterns in neural activity for reaching movements using nonnegative matrix factorization. *EURASIP Journal on Applied Sig. Proc., Special Issue of Trends in Brain Computer Interfaces*, 2005(19) :3113-3121, 2005.
- [49] Kim, H. Park. *Sparse nonnegative matrix factorization for clustering*. Georgia institute of technology. Technical reports, GT-CSE-08-01, 2008.
- [50] R. Kompass. A generalized divergence measure for nonnegative matrix factorization. *Neural Comput.* 19(3), 780-791, 2007.
- [51] M. Krein, and D. Milman : On extreme points of regular convex sets. *Studia Mathematica* 9.1, pp : 133-138, 1940.

Bibliographie

- [52] C.L. Lawson and R.J. Hanson. Solving least squares problems. Prentice-Hall Englewood Cliffs, NJ, 1974.
- [53] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755) :788–791, 1999.
- [54] D. Lee, Seung, H.Seung. Algorithms for non-negative matrix factorization. In : Annual Conference on Neural Information Processing Systems, 556-562, 2000.
- [55] J. Le Roux, H. Kameoka, N. Ono, A. de Cheveigné and S. Sagayama. Computational auditory induction by missing-data non-negative matrix factorization. In ITRW on Statistical and Perceptual Audio Processing, Brisbane, Australie, Septembre 2008b.
- [56] S.Z. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. Dans Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 207-212, Hawaii, USA, 11-13 décembre 2001.
- [57] T. Li. A general model for clustering binary data. In SIGKDD. pp. 188-197, 2005.
- [58] C. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10) :2756-2779, 2007.
- [59] M. M., Lin and M. T. Chu. On the nonnegative rank of Euclidean distance matrices. *Linear algebra and its applications*, 433(3), 681-689, 2010.
- [60] W. Liu and K. Yuan. Sparse p-norm nonnegative matrix factorization for clustering gene expression data. *International Journal of Data Mining and Bioinformatics*, 2(3) :236-249, juillet 2008.
- [61] C. Meyer, D. Carl. *Matrix Analysis and Applied Linear Algebra*, SIAM, p. 514, ISBN 978-0-89871-454-8, 2000.
- [62] Q. Mo, B., Draper. Semi-nonnegative matrix factorization for motion segmentation with missing data. In : Computer Vision-ECCV 2012, pp. 402-415, 2012.
- [63] A. Ozerov and C. Févotte. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Audio, Speech and Language Processing*, 18(3) :550-563, 2010.

- [64] P. Paatero and U. Tappe. Positive matrix factorization : A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2), 111-126, 1994.
- [65] J. Paulus, T. Virtanen. Drum transcription with non-negative spectrogram factorization. In *European Signal Processing Conference (EUSIPCO)*, Antalya, Turquie, Septembre 2005.
- [66] B. Ribeiro, C., Silva. Extracting discriminative features using nonnegative matrix factorization in financial distress data. In : *Proceedings of the 9th International Conference on Adaptive and Natural Computing Algorithms, ICANNGA 2009*, 537-547, 2009.
- [67] F. Shahnaz, M. Berry, V. Pauca, and R. Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing and Management*, 42(2) :373–386, 2006.
- [68] Z. Sheng, Y. Zhi, and W. Can. An improved non-negative matrix factorization algorithm based on genetic algorithm. in *Proc. ICCSET, Int. Conf. on Computer Science and Electronic Technology*, pp. 395-398, 2014.
- [69] P. Smaragdis and J.C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177-180. IEEE, 2003.
- [70] V. Sná ?el, J. Plato ?, P. Krömer. Developing genetic algorithms for boolean matrix factorization. *Databases, Texts*, 61, 2008.
- [71] M. Srinivas, L. M. Patnaik. Genetic algorithms : A survey. *computer*, 27(6), 17-26, 1994.
- [72] L. Stebbins, F. Ayala. L'évolution du Darwinisme. *Pour la Science Paris N°95*, pp.48-58, 1985.
- [73] A. Vandendorpe, N. D. Ho, S. Vanduffel, P. V. Dooren. : On the parameterization of the CreditRisk+ model for estimating credit portfolio risk. *Insurance : Mathematics and economics* 42, 736-745, 2008.

Bibliographie

- [74] T. Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3) :1066-1074, Mars 2007.
- [75] F. Wang, CH. Tan, P. Li, AC. Konig. Efficient document clustering via online non-negative matrix factorizations. *Proceedings of the SIAM International Conference on Data Mining*, 2011.
- [76] Y. Wang, Y. Jia, C. Hu, and M. Turk. Fisher non-negative matrix factorization for learning local features. Dans *Asian Conference on Computer Vision*, Jeju, Corée, 24-27 janvier 2004.
- [77] S. Wild. Seeding non-negative matrix factorization with the spherical k-means clustering, M.S. Thesis, University of Colorado, 2002.
- [78] I. H. Witten, E. Frank, M. A. Hall. *Data Mining : Practical Machine Learning Tools and Techniques*, Third Edition (Morgan Kaufmann Series in Data Management Systems). Elsevier. 2011.
- [79] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *SIGIR'03 : Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 267-273, New York, NY, USA, ACM Press, 2003.
- [80] N. Yokoya, J. Chanussot, A. Iwasaki. Generalized bilinear model based nonlinear unmixing using Semi-nonnegative matrix factorization. In : *IEEE Int. Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1365-1368, 2012.
- [81] Z. Y. Zhang. *Nonnegative matrix factorization : Models, Algorithms and Applications*. Springer, 2012.
- [82] Z. Y. Zhang, T. Li, C. Ding, X.S Zhang. : Binary matrix factorization with applications in Data Mining, *IEEE International Conference*, 391-400, 2007.
- [83] <https://archive.ics.uci.edu/ml/datasets/Ionosphere>.