Méthodes aléatoires pour l'apprentissage de données en grande dimension : application à l'apprentissage partagé

Nassara Elhadji Ille Gado

Thèse
de doctorat
de l'UTT

Nassara ELHADJI ILLE GADO

Méthodes aléatoires pour l'apprentissage de données en grande dimension - Application à l'apprentissage partagé

Spécialité :
Optimisation et Sûreté des Systèmes

2017TROY0032 Année 2017

utt
UNIVERSITÉ DE TECHNOLOGIE
TROYES

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Nassara ELHADJI ILLE GADO

le 5 décembre 2017

Méthodes aléatoires pour l'apprentissage de données en grande dimension - Application à l'apprentissage partagé

JURY

M. P. BEAUSEROY	PROFESSEUR DES UNIVERSITES	Président
M. S. CANU	PROFESSEUR DES UNIVERSITES	Rapporteur
Mme É. GRALL-MAËS	MAITRE DE CONFERENCES	Directrice de thèse
M. P. HONEINE	PROFESSEUR DES UNIVERSITES	Examinateur
Mme M. KHAROUF	MAITRE DE CONFERENCES	Directrice de thèse
M. M. SAYED-MOUCHAWEH	PROFESSEUR IMT LILLE DOUAI - HDR	Rapporteur

Dédicaces

À

♡ Ma fille chérie, **Nisrine AMADOU**, source de motivation incontestable, c'est à toi que revient toutes ces années d'acharnement, sans nulle doute tu es celle qui as souffert le plus durant ces années...

Remerciements

Je tiens à remercier vivement tous ceux ou celles qui m'ont accompagné tout le long de ces longues années d'études et plus particulièrement durant cette thèse. Ces trois dernières années ont été pleines d'expériences inédites sur un travail fastidieux, de belles opportunités d'échanges et de belles rencontres. J'adresse ici mes plus sincères remerciements :

- A mes directrices de thèse : Edith GRALL-MAËS et Malika KHAROUF, sans vous, ce travail de thèse n'aurait pu être possible, je tiens à exprimer toute ma gratitude. Je vous remercie très chaleureusement de m'avoir donné la chance de réaliser mon voeu sous votre direction. Je vous remercie également de m'avoir fait confiance et je ne saurai vous remercier pour tout l'enseignement, les conseils constructifs que vous m'avez transmis durant ces trois ans, pour me faire progresser et me propulser jusqu'au sommet. Merci encore pour votre intérêt, votre soutien, votre disponibilité et surtout pour votre patience dans les moments difficiles que nous avons traversé ensemble.
- Aux membres du jury qui m'ont fait l'honneur d'accepter d'évaluer ce travail. Je remercie particulièrement Mr. Stéphane CANU, Mr. Moamar SAYED-MOUCHAWEH, Mr Paul HONEINE et Mr. Pierre BEAUSEROY. Merci pour votre lecture attentive de ma thèse ainsi que les remarques que vous avez m'adressé lors de la soutenance afin de permettre l"amélioration de mon travail.
- L'équipe administrative de l'UTT : Mme Denis, Mme Leclercq, Mme Kazarian, Bernadette, Véronique et Patricia.
- Tous mes amis et collègues de l'UTT et d'ailleurs, au terme de ce parcours je vous remercie infiniment pour tous vos encouragements.
- Tous mes enseignants tout au long de mes études, a tous ceux qui ont participé de près ou de loin à la réalisation de ce travail et à ceux qui ont eu la pénible tâche de soulager et diminuer la souffrance encourue.

Résumé

Cette thèse porte sur l'étude de méthodes aléatoires pour l'apprentissage de données en grande dimension. Nous proposons d'abord une approche non supervisée consistant en l'estimation des composantes principales, lorsque la taille de l'échantillon et la dimension de l'observation tendent vers l'infini. Cette approche est basée sur les matrices aléatoires et utilise des estimateurs consistants de valeurs propres et vecteurs propres de la matrice de covariance. Ensuite, dans le cadre de l'apprentissage supervisé, nous proposons une approche qui consiste à, d'abord réduire la dimension grâce à une approximation de la matrice de données originale, et ensuite réaliser une LDA dans l'espace réduit. La réduction de dimension est basée sur l'approximation de matrices de rang faible par l'utilisation de matrices aléatoires. Un algorithme d'approximation rapide de la SVD, puis une version modifiée permettant l'approximation rapide par saut spectral sont développés. Les approches sont appliquées à des données réelles images et textes. Elles permettent, par rapport à d'autres méthodes, d'obtenir un taux d'erreur assez souvent optimal, avec un temps de calcul réduit. Enfin, dans le cadre de l'apprentissage par transfert, notre contribution consiste en l'utilisation de l'alignement des sous-espaces caractéristiques et l'approximation de matrices de rang faible par projections aléatoires. La méthode proposée est appliquée à des données issues d'une base de données de référence ; elle présente l'avantage d'être performante et adaptée à des données de grande dimension.

Abstract

This thesis deals with the study of random methods for learning large-scale data. Firstly, we propose an unsupervised approach consisting in the estimation of the principal components, when the sample size and the observation dimension tend towards infinity. This approach is based on random matrices and uses consistent estimators of eigenvalues and eigenvectors of the covariance matrix. Then, in the case of supervised learning, we propose an approach which consists in reducing the dimension by an approximation of the original data matrix and then realizing LDA in the reduced space. Dimension reduction is based on low–rank approximation matrices by the use of random matrices. A fast approximation algorithm of the SVD and a modified version as fast approximation by spectral gap are developed. Experiments are done with real images and text data. Compared to other methods, the proposed approaches provide an error rate that is often optimal, with a small computation time. Finally, our contribution in transfer learning consists in the use of the subspace alignment and the low-rank approximation of matrices by random projections. The proposed method is applied to data derived from benchmark database ; it has the advantage of being efficient and adapted to large-scale data.

Notations

Symbole	Signification
d	Nombre de variables
N	Nombre d'échantillons
K	Nombre de classes (ou clusters)
$\mathbf{x}_i$	$i^{ème}$ échantillon observé sur les (d) variables
$\mathbf{X}_j$	$j^{ème}$ variable décrit sur les (N) échantillons
$\mathbf{X}$	Matrice des données
a^T	Transposée de a
$\hat{a}$	Estimée de a
$\tilde{a}$	Approximée de a
$\bar{a}$	Matrice de données centrées
$\mathbf{S_b}$	Matrice variance-covariance inter classes
$\mathbf{S_w}$	Matrice variance-covariance intra classes
S_t	Matrice variance-covariance totale
$\mathbf{W}$	Matrice de similarité (affinité)
$\mathbf{D}$	Matrice diagonale des degrés du graphe
$\mathbf{L}$	Matrice du graphe Laplacien
$\mathbf{Y}$	Vecteur des étiquettes des classes
$\mathbf{y_i}$	étiquette de la classe numéro i
λ	Valeur propre
$\mathbf{\Delta}$	Matrice diagonale contenant les valeurs propres
u	Vecteur propre
$\mathbf{U}$	Matrice des vecteurs propres
$\mathbf{P}$	Matrice de projection orthogonale
$\mathbf{I}$	Matrice identité

Table des matières

1 Introduction générale **1**

1.1 Contexte général 1

1.2 Objectifs et structure de la thèse 3

2 Généralités sur l'apprentissage automatique **5**

2.1 Introduction 5

2.2 Apprentissage non supervisé 6

2.2.1 Nuées dynamiques 6

2.2.2 Partitionnement hiérarchique 8

2.2.3 Partitionnement spectral 9

2.3 Apprentissage supervisé 11

2.3.1 Plus proches voisins 13

2.3.2 Moyenne la plus proche 13

2.3.3 Analyse linéaire discriminante 14

2.3.4 Séparateurs à vaste marge 15

2.4 Réduction de dimension 17

2.4.1 Analyse en composante principale 18

2.4.1.1 Avec la covariance 19

2.4.1.2 Avec la décomposition en valeurs singulières 20

2.4.1.3 Analyse en composante principale à noyau 21

2.4.2 Réduction de dimension avec matrices aléatoires 22

2.4.2.1 Projection aléatoire 22

2.4.2.2 Approximation de la SVD 23

2.5 Apprentissage supervisé pour les données en grande dimension 25

2.6 Apprentissage partagé 25

2.6.1 Principes 26

2.6.2 Synthèse de méthodes existantes 27

2.7 Conclusion 30

3 Estimation de la covariance pour les données en grande dimension 32
3.1 Introduction . . . 32
3.2 Estimation de la covariance basée sur les matrices aléatoires . . . 33
3.2.1 Outils des matrices aléatoires . . . 34
3.2.2 Estimation (N, d)-consistante de la matrice de covariance . . . 37
3.3 Partitionnement basé sur la covariance empirique . . . 39
3.3.1 Analyse en composantes principales : nouvelle alternative . . . 39
3.3.2 Partitionnement par ACP . . . 40
3.3.3 Indicateurs de performance . . . 41
3.4 Résultats et discussion . . . 43
3.5 Conclusion . . . 45

4 Analyse discriminante linéaire pour les données en grande dimension 47
4.1 Introduction . . . 47
4.2 Description de méthodes existantes . . . 48
4.2.1 Régression spectrale . . . 48
4.2.2 Décomposition QR . . . 54
4.2.3 Projection aléatoire . . . 55
4.3 Analyse discriminante linéaire rapide . . . 56
4.3.1 Principes et motivations de la méthode proposée . . . 56
4.3.2 Approches pour la réduction de dimension à l'aide de l'approximation de la SVD . . . 58
4.3.2.1 Approximation classique . . . 58
4.3.2.2 Approximation rapide . . . 60
4.3.2.3 Approximation rapide par saut spectral . . . 60
4.3.3 Description de la méthode . . . 64
4.3.4 Complexité de la méthode . . . 64
4.4 Présentation des données . . . 65
4.4.1 Données images . . . 65
4.4.2 Données textes . . . 66
4.4.3 Normalisation des données . . . 66
4.5 Résultats d'expérimentation . . . 68
4.5.1 Implémentation et paramétrage . . . 68
4.5.2 Résultats et analyse . . . 69
4.6 Conclusion . . . 70

5 Apprentissage partagé pour les données en grande dimension 74
5.1 Introduction . . . 74

5.2 Formulation du problème 75
5.3 Adaptation par transfert partagé entre les domaines 76
5.3.1 Méthode d'alignement des sous-espaces 77
5.3.2 Approximation rapide d'alignement des sous-espaces 78
5.4 Expérimentation 79
5.4.1 Présentation des données 80
5.4.2 Méthodes de comparaison 81
5.4.3 Implémentation et paramétrage 83
5.4.4 Résultats et analyse 84
5.5 Conclusion 86

6 Conclusion et perspectives **87**
6.1 Conclusion et travaux effectués 87
6.2 Perspectives 89

Bibliographie **91**

Table des figures

2.1 Données originales 2.1(a) ; Résultats de partitionnement de l'algorithme k-means obtenus avec une seule itération 2.1(b) ; deux itérations 2.1(c) ; cinq itérations 2.1(d). 8

2.2 Exemple de classification ascendante hiérarchique. 9

2.3 Matrice d'adjacence et graphe correspondant 9

2.4 2.4(a) Données originales ; 2.4(b) avec la matrice de Laplacien normalisée symétrique avec différentes similarités : (1) voisinage $\epsilon = 2$, (2) normal KNN=3 et (3) mutuel KNN=5 ; 2.4(c) résultats partitionnement obtenus avec Laplacien normalisé et similarité totalement connectée ; 2.4(d) par k-means. . . . 12

2.5 Exemple de k plus proches voisins. Dans le cercle en pointillé, le point A dispose de 5 plus proches voisins, et dans le cercle supérieur, le point A dispose de 10 voisins. 13

2.6 Exemple d'illustration du principe de la moyenne la plus proche. Les distances d_1 et d_2 caractérisent la classe d'appartenance du point inconnue matérialisé en vert. 14

2.7 Principe de l'analyse linéaire discriminante 15

2.8 Séparateurs vaste marge. 17

3.1 Densité Marchenko-Pastur et l'histogramme des valeurs empiriques pour N=1000 et trois différentes valeurs de $y = \frac{d}{N}$. 36

3.2 Entropie et information mutuelle d'un couple de variables aléatoires (W,Z). 42

3.3 Résultats du partitionnement 45

4.1 Représentation de la liste des sommets adjacents à coté de chaque sommet du graphe. 49

4.2 Exemple de graphe de Scree. 62

4.3 Échantillons d'exemples illustratifs 67

4.4 Résultats de simulation sur les données MINST. 70

4.5 Résultats de simulation sur les données COIL20. 71

4.6 Résultats de simulation sur les données Reuters21578. 71

4.7 Résultats de simulation sur les données TDT2. 71

4.8 Résultats de simulation sur les données 20NewsGroups. 72

4.9 Résultats de simulation sur les données ORL. 72
4.10 Influence du paramètre p sur l'accuracy et le temps de calcul sur les données COIL20, avec $k = p$ fixé et TN=30%. 72
4.11 Influence du paramètre k sur l'accuracy et le temps de calcul sur les données COIL20, avec TN=30% et p=70. 73

5.1 Positionnement de l'adaptation entre les domaine au sein de l'apprentissage automatique 76
5.2 Variation du taux d'erreur en fonction de la dimension réduite sur données 20Newsgroups. Nombre de plus proches voisins $KNN = 10$. 84
5.3 Variation du taux d'erreur en fonction du nombre de plus proches voisins KNN sur données 20Newsgroups. Dimension sous-espace $k = 100$. 84
5.4 Variation du taux d'erreur en fonction de la dimension réduite sur les données Reuters. Nombre de plus proches voisins KNN=10. 85
5.5 Variation du taux d'erreur en fonction du nombre de plus proches voisins KNN sur données Reuters. Dimension sous-espace $k = 50$. 85

Liste des tableaux

3.1 Information mutuelle normalisée (m=moyenne, σ=écart-type) 44
3.2 Taux d'erreur (m=moyenne, σ=écart-type) 44

4.1 Statistique des données et valeurs des paramètres 66

5.2 Statistique des données Reuters-21578 data 81
5.1 Description des données 20Newsgroups 82
5.3 Taux d'erreur (%). $k = 20$ et $KNN = 10$. 83
5.4 Temps d'exécution pour les données images (s). $k = 20$ et $KNN = 10$. . . . 85

Chapitre 1

Introduction générale

1.1 Contexte général

Dans les différents domaines de recherches scientifiques, le développement technologique et le besoin de superviser des systèmes de plus en plus complexes nécessitent l'analyse de bases de données de taille importante (signaux, images, documents, scènes audio/vidéo, ...). A titre d'exemples, dans le domaine de la reconnaissance d'objets, du multimédia, de la vision par ordinateur et de classification de documents, près de 500 heures de fichiers vidéo sont téléchargés sur Youtube chaque minute [1], Google a répertorié plus de 1000 milliards de pages web dans le monde [2], environ 3 millions d'applications mobiles se partagent entre Google app store, Apple app store et Windows phone store [3], [1, 2]. En réponse aux difficultés d'encodage de ces volumes de données en perpétuelle croissance, de nombreux chercheurs ont récemment tourné leur attention vers l'apprentissage automatique des données en grande dimension comme un moyen de surmonter le goulet d'étranglement de leur traitement.

Toutefois, si l'on est sûr d'avoir une information assez complète lors de l'acquisition de ces données, celle-ci risque d'être "immergée" dans le lot (ou noyée dans la masse de données). Ceci pose les problèmes de la structuration des données et de l'extraction des connaissances ou d'informations. En effet, les bases de données sont en général définies par des tableaux à deux dimensions : le nombre de variables et le nombre d'échantillons. Ces deux dimensions peuvent prendre des valeurs très élevées, ce qui peut poser un problème lors du stockage, de l'exploration et de l'analyse. Pour cela, il est fondamental de mettre en place des outils de traitement de données permettant l'extraction des connaissances sous-jacentes. L'extraction de connaissances à partir des données se définit comme l'acquisition de connaissances nouvelles, intelligibles et potentiellement utiles à partir de faits cachés au sein de grandes quantités de données [3]. En fait, on cherche surtout à isoler des traits structuraux ou schémas (patterns) qui soient valides, non triviaux, utilisables et surtout compréhensibles ou explicables. L'extraction des connaissances s'effectue selon deux directions, la catégorisation des données (par regroupement en classes) ou la réduction de

1. http ://tubularinsights.com/hours-minute-uploaded-youtube/
2. http ://www.webrankinfo.com/dossiers/indexation
3. http ://www.geeksandcom.com/2015/04/15/applications-mobiles-chiffres/

la dimension de l'espace de représentation de ces données (par sélection ou extraction des variables).

La réduction de la dimension se pose comme une étape primordiale dans le processus de pré-traitement des données (compression, nettoyage, élimination des points aberrants, etc.). Son but principal est de sélectionner ou d'extraire un sous-ensemble optimal de variables pertinentes. En effet, pour des données appartenant à un espace de grande dimension, certaines variables n'apportent aucune information, d'autres sont simplement redondantes ou corrélées. Ceci rend les algorithmes de décision complexes, inefficaces, moins généralisables dans certaines situations ou présentent une interprétation assez délicate. La sélection d'un sous-ensemble permet d'éliminer les informations non-pertinentes et redondantes selon un critère défini. Les méthodes de réduction de la dimension de l'espace de représentation des données peuvent être divisées en deux parties principales : les méthodes d'extraction de variables et méthodes de sélection de variables. L'extraction d'attributs transforme l'espace d'attributs de départ en un nouvel espace formé par une combinaison linéaire ou non linéaire des attributs initiaux. La sélection d'attributs choisit les attributs les plus pertinents selon un critère donné.

La complexité du traitement des données observées diffère généralement selon leur type. On parle de base de données massive lorsque le nombre d'échantillons observés N est largement supérieur au nombre de variables (d), dans le cas contraire on parle des données à très grande dimension. On distingue les données clairsemées ou creuses (*'sparses' en anglais*) des deux autres types de données lorsque la grande majorité des valeurs prises pour les variables explicatives sont absentes ou nulles. Ces données à faible densité en information sont caractéristiques du Big Data. Le nombre d'échantillons dans ce cas et le nombre des variables sont généralement tous deux grands. Face au défi de grande dimension des données, les méthodes d'apprentissage se focalisent sur la recherche d'informations pertinentes ou des "pépites" d'informations pour l'aide à la décision et à la prévision. Elles mettent en œuvre des techniques statistiques d'apprentissage en tenant compte de la volumétrie de la base de données.

Une hypothèse majeure des méthodes traditionnelles d'apprentissage automatique est que les données d'apprentissage (*training*) et les données et de validation (*testing*) sont issues du même domaine, de sorte que l'espace des variables en entrée et la distribution des données sont les mêmes. Cependant, dans beaucoup de scénarios d'apprentissage supervisé, cette hypothèse forte n'est pas toujours vérifiée en pratique. Par exemple, très souvent lorsqu'on effectue une tâche de classification dans un domaine, on dispose généralement de données suffisamment abondantes dans un autre domaine d'intérêt différent du premier domaine. Dans ce domaine, les données peuvent avoir des variables différentes (espace de fonctionnalité différent) ou suivre une distribution différente du premier domaine. Il est donc important de développer des méthodes d'apprentissage performantes formées à partir de données plus facilement récupérables, voir simplement les seules disponibles. *L'apprentissage partagé* ou l'apprentissage par transfert des connaissances entre les domaines (*transfer learning*) donne des éléments de réponse à ce type de problématique. L'idée générale consiste à trouver un espace commun ou intermédiaire entre les domaines, dans lequel les données

d'un domaine source D_S et d'un domaine cible D_T, partagent le maximum d'informations communes ou peuvent avoir une distribution marginale assez similaire.

1.2 Objectifs et structure de la thèse

L'objectif principal de cette thèse est le développement d'algorithmes d'apprentissage automatique pour les données en grande dimension. En utilisant des méthodes aléatoires pour l'extraction des variables, le but est de contribuer à une amélioration des performance en temps de calcul des méthodes existantes. Le travail est réalisé sur différentes problématiques.

La première problématique réside sur le fait que, les méthodes traditionnelles de traitement des données peuvent donner des temps de calcul excessifs et présenter des difficultés de stockage des grandes matrices de données. En se basant sur des techniques d'apprentissage supervisé et non supervisé, le but est de proposer une approche qui permet une manipulation plus aisée de données issues d'un environnement complexe et en grande dimension. De ce fait, l'objectif d'une part, est de développer une technique de réduction de dimension qui permet la suppression d'information redondante et non informative au sein des données. D'autre part, l'application des méthodes classiques sur les grandes bases de données est parfois infaisable. Pour ce faire, la réduction de dimension de l'espace initial permet de pouvoir effectuer les méthodes d'apprentissage classique dans le nouvel espace réduit, tout en limitant la perte d'information au sein des données. La seconde problématique de cette thèse est l'utilisation des techniques d'apprentissage partagé (ou apprentissage par transfert) dans le cas des données en grande dimension. En effet, les données d'apprentissage et de validation des modèles d'apprentissage peuvent provenir de sources différentes. Dans ce genre de contexte, un modèle d'apprentissage construit sur une base de données peut être confronté à une dégradation de performance lorsqu'il est testé ou validé sur une nouvelle base de données provenant d'une autre source. Le principe de transfert de connaissance entre les domaines permet d'utiliser l'information du premier domaine pour la transférer au deuxième domaine dans le but de prédire uniquement le deuxième domaine. Ainsi, un modèle d'apprentissage construit dans un nouvel espace où les deux domaines partagent conjointement certaines caractéristiques pourrait convenablement prédire les données du deuxième domaine.

Le chapitre 2 est consacré à des généralités des méthodes d'apprentissage des données nécessaires à la suite du document. Des techniques en apprentissage supervisé et non supervisé, les plus couramment utilisées dans la littérature, ont été d'abord introduites. Ensuite des méthodes de la réduction de dimension des données sont présentées ainsi que des approches utilisées pour l'apprentissage des données en grande dimension. Puis le principe d'apprentissage partagé ainsi que des techniques développées dans cette thématique sont également présentées. Enfin, une conclusion clôture ce chapitre.

Le chapitre 3 présente une approche proposée dans ce travail de thèse pour l'analyse des données en grande dimension dans le cas de l'apprentissage non supervisé. Une technique d'analyse en composantes principales basée sur des nouveaux estimateurs de la matrice de

covariance est proposée. Le chapitre présente dans un premier temps les outils des matrices aléatoires communément utilisés dans la littérature pour l'analyse des grandes matrices de données. Le principe consiste d'utiliser ces outils des matrices aléatoires pour calculer de nouveaux estimateurs de vecteurs propres et valeurs propres afin de trouver un sous espace de projection optimal, où il est possible de calculer les composantes principales. A cet effet, une application du partitionnement spectral est réalisée dans le sous espace engendré par les vecteurs propres calculés. Pour évaluer la méthode proposée, deux indicateurs de performance sont utilisés et testés sur des données synthétiques.

Le chapitre 4 présente d'abord une description détaillée de méthodes existantes d'apprentissage de données en grande dimension basées sur la technique d'analyse linéaire discriminante. Principalement, trois méthodes utilisées par la suite sont présentées à savoir la régression spectrale, la décomposition QR et la projection aléatoire. Ensuite une nouvelle approche de l'analyse linéaire discriminante (LDA) est proposée sur la base d'une approximation de la décomposition en valeurs singulières ainsi que deux versions améliorées. Une présentation détaillée des bases de données utilisées dans les expériences est introduite. Les résultats d'expérimentation sur l'ensemble des méthodes comprenant les méthodes de comparaison ainsi que approches proposées ont été reportés.

Le chapitre 5 est consacré à l'adaptation des approches proposées dans le chapitre 4 dans le cadre de l'apprentissage partagé ou transfert de connaissance entre les domaines. Dans ce chapitre, tout d'abord le problème de transfert est introduit. Ensuite une technique permettant le transfert, afin d'adapter les domaines, basée sur l'alignement des sous espaces est présentée. Une nouvelle approche de l'alignement des sous espace pour le transfert est introduite en utilisant la technique de l'approximation rapide de la décomposition en valeurs singulières. Sont ensuite présentées les bases de données et les méthodes utilisées pour la comparaison ainsi que l'implémentation et paramétrage des méthodes. Puis les résulats sont exposés et discutés.

Une conclusion générale termine ce manuscrit en synthétisant les points forts des différents travaux réalisés ainsi que les perspectives et extensions pour des travaux futurs.

Chapitre 2

Généralités sur l'apprentissage automatique

2.1 Introduction

L'apprentissage automatique -*Machine learning*- désigne un ensemble de méthodes et d'algorithmes permettant d'extraire de l'information pertinente au sein de données ou d'apprendre un comportement à partir de l'observation d'un phénomène. Il permet aux ordinateurs d'utiliser des données préalablement recueillies afin de prévoir les comportements, les résultats et les évolutions/tendences futures. L'apprentissage automatique est considéré comme un champ d'étude de l'intelligence artificielle (IA) où les prévisions sont établies à partir de techniques d'apprentissage. Dans plusieurs domaines d'intérêt, ces techniques peuvent rendre les applications et les appareils plus intelligents. Par exemple, lorsque nous faisons nos achats en ligne, l'apprentissage automatique permet de recommander d'autres produits susceptibles de nous intéresser en fonction de nos historiques d'achats. Ou lorsqu'on utilise une carte de crédit pour effectuer une transaction, l'apprentissage automatique compare la transaction encours à une base de données de transactions et aide la banque à détecter des fraudes. Pour la reconnaissance des formes, la détection d'anomalies, et pour plein d'autres applications, l'apprentissage automatique est de nos jours devenu chose incontournable. Il se caractérise par un ensemble de règles utilisées pour résoudre les problèmes de traitement et d'analyse des données, de calcul mathématique ou de déduction automatisée. La construction d'un modèle d'apprentissage est une abstraction de la question à laquelle on essaie de répondre ou le résultat que l'on souhaite prédire. Ainsi, les méthodes d'apprentissage automatique consistent à la recherche d'information véhiculée au sein d'un ensemble d'observations recueillies sur un quelconque prototype ou système donné en construisant des modèles mathématiques à des fins prévisionnelles et/ou décisionnelles.

Ce chapitre présente différents types de méthodes d'apprentissage automatique. Nous présentons dans la section 2.2 des techniques d'apprentissage non supervisé dont entre autres, la méthode de nuées dynamiques, le partitionnement hiérarchique et le partitionnement spectral. Ensuite, nous présentons dans la section 2.3 des techniques d'apprentissage supervisé à savoir la moyenne la plus proche, l'analyse linéaire discriminante et les sépara-

teurs à vaste marge. Puis dans la section 2.4 nous présentons des techniques de réduction de dimension où nous détaillons des algorithmes basés sur l'analyse en composantes principales et sur les matrices aléatoires. Puis la section 2.5 présente des approches adaptées pour l'apprentissage des données en grande dimension. Ensuite la section 2.6 introduit le principe général de l'apprentissage partagé et la synthèse de approches existantes.

2.2 Apprentissage non supervisé

On parle d'apprentissage non supervisé lorsque l'on dispose uniquement d'un ensemble d'échantillons à partir duquel on cherche à inférer des connaissances ou des structures naturellement présentes. Le but est de trouver des relations entre les données sans disposer d'aucune information a priori sur le jeu de données. Il existe des techniques basées sur l'estimation de la densité où l'on cherche au mieux à déceler l'existence de classes ou groupes dans les données en utilisant la théorie Bayésienne sur la base des probabilités a posteriori. Dans ce travail de thèse, nous présentons de techniques de classification non supervisée qui déborde le cadre strictement exploratoire correspondant à la recherche d'une typologie ou d'une partition d'individus en classes ou catégories. Ceci est effectué en optimisant un critère visant à regrouper les individus homogènes dans la même classe et ceux qui sont distincts dans des classes différentes. La classification non supervisée se distingue des procédures de discrimination, ou encore de classement (classification en anglais) pour lesquelles une répartition est a priori connue. Nous présentons trois principales méthodes de classification non supervisée, à savoir le nuées dynamiques (K-means), le partitionnement hiérarchique et le partitionnement spectral (spectral clustering). Nous allons rappeler le principe de ces méthodes.

2.2.1 Nuées dynamiques

La méthode K-means ou nuées dynamiques [4] est une technique bien connue de classification qui propose une solution au problème d'optimisation d'un critère des moindres carrés (appelé aussi critère de variance intra-classe). Ce critère favorise les partitions dont les classes présentent une faible variance, c'est-à-dire que les objets à l'intérieur d'une même classe sont faiblement dispersés. L'algorithme s'effectue de manière itérative.

Étant donné un ensemble Ω de N échantillons de données décrites par d variables à valeurs dans $\mathbb{R}$ et $\mathcal{D}$ une distance sur $\mathbb{R}^d$, l'algorithme K-means cherche à regrouper ces données en K groupes homogènes $\Omega_1, \cdots, \Omega_K$, inconnus a priori. Il cherche à minimiser la distance $\mathcal{D}$ entre les échantillons à l'intérieur de chaque groupe $\Omega_i, i = 1, \cdots, K$. Cette méthode produit exactement K différents clusters, avec le paramètre K fixé a priori. L'idée principale est de définir K centres, chacun émane d'un cluster. Chaque échantillon est placé dans le cluster dont la distance au centre de ce cluster est la plus petite par rapport aux autres centres. On s'intéresse souvent à un critère qui correspond à la somme des inerties intra-classe des groupes. Ce critère correspond à la fonction d'optimisation de K-means qui

vise à trouver l'optimum de l'expression suivante :

$$\mathcal{J}_{\Omega}(\mathcal{V}) = \arg\min_{\mathbf{\Omega}} \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in \Omega_i} \mathcal{D}(\mathbf{x}_j, \boldsymbol{\mu}_i)^2, \tag{2.1}$$

avec $\mathcal{V} = \{\boldsymbol{\mu}_i, 1 \leq j \leq K\}$ l'ensemble des centres des K ensembles Ω_i pour $\{\Omega\}_{i=1,\ldots,K}$. Pour Ω et K donnés, plus la valeur de $\mathcal{J}_{\Omega}(\mathcal{V})$ est faible, plus les groupes sont "compacts" autour de leurs centres, et donc meilleure est la qualité du partitionnement obtenu. Trouver le minimum global de la fonction $\mathcal{J}_{\Omega}(\mathcal{V})$ est un problème difficile, mais on dispose d'algorithmes de complexité polynomiale par rapport au nombre de données N qui produisent une solution en général sous-optimale. Un tel algorithme est l'algorithme des centres mobiles décrit ci-dessous :
Initialisation : le nombre de classes K étant imposé, choisir K points aléatoirement pour constituer initialement les représentants de chaque classe.
Pour chaque point :

1. Calculer les distances entre ce point et les représentants des classes,
2. Affecter à ce point la classe pour laquelle la distance est minimale,
3. Connaissant les membres de chaque classe, on recalcule les représentants de chaque classe (centres d'inertie),
4. On redistribue les objets dans la classe qui leur est la plus proche en tenant compte des nouveaux centres de classe calculés à l'étape précédente,
5. On retourne à l'étape 3 jusqu'à ce qu'il y est convergence, c'est-à-dire jusqu'à ce qu'il n'y a plus aucun individu qui change de classe.

La valeur de $\mathcal{J}_{\Omega}(\mathcal{V})$ diminue lors de chacune des deux étapes du processus itératif (affectation de chaque donnée à un groupe, calcule des centres). Comme $\mathcal{J}_{\Omega}(\mathcal{V}) \geq 0$, le processus itératif converge. La solution obtenue est en général un minimum local, dépendant de l'initialisation, de valeur $\mathcal{J}_{\Omega}(\mathcal{V})$ pouvant être plus élevée que celle correspondant au minimum global qui est généralement inconnu. La figure 2.1 donne un exemple illustrant les étapes de l'algorithme K-means. Les données initiales sont matérialisées en point et les centres en croix. A chaque itération, on affecte chaque échantillon à un centre qui lui est proche. Ensuite, on met à jour la valeur du nouveau centre jusqu'à ce que les centres ne bougent plus.

L'algorithme K-means a l'avantage d'être une méthode extrêmement simple à appliquer, mais il est peu robuste car il est très sensible aux outliers (valeurs aberrantes). Ainsi, ajouter un élément atypique les données peut complètement modifier le partitionnement des données [5]. Il existe diverses variantes de la méthode concernant la sensibilité des résultats en fonction des conditions d'initialisation de la méthode. Différentes initialisations des centroïdes au démarrage de l'algorithme peuvent parfois influencer les résultats finaux obtenus [6], [7].

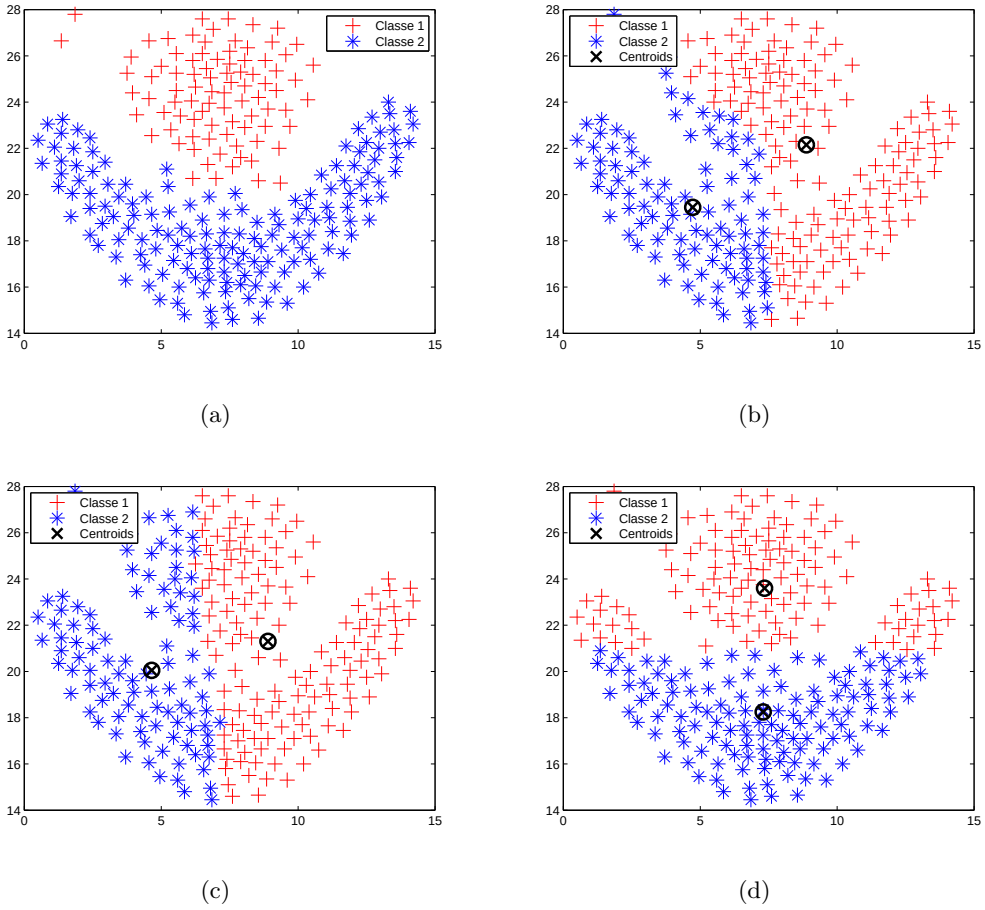


FIGURE 2.1 – Données originales 2.1(a) ; Résultats de partitionnement de l'algorithme k-means obtenus avec une seule itération 2.1(b) ; deux itérations 2.1(c) ; cinq itérations 2.1(d).

2.2.2 Partitionnement hiérarchique

On distingue deux types d'approches de classification hiérarchique : les méthodes descendantes ou divisives et les méthodes ascendantes ou agglomératives [8]. Ces méthodes peuvent s'appliquer à des tableaux de dissimilarités ou des tableaux numériques. Les algorithmes construisent des partitions emboîtées (hiérarchies) avec un nombre K de partitions variant de N à 1 pour une classification hiérarchique ascendante, ou de 1 à N pour une classification hiérarchique descendante. Le partitionnement hiérarchique vise à obtenir une agrégation de regroupements. Par rapport au partitionnement des données classiques, il fournit une information riche concernant la structure de similarité des données. La classification ascendante procède par agrégations successives de groupes. A partir de la hiérarchie de groupes résultante, le partitionnement hiérarchique permet d'observer l'ordre des agrégations de groupes, d'examiner les rapports des similarités entre groupes, ainsi que d'obtenir plusieurs partitionnements à des niveaux de similarité différents. Il existe plusieurs algorithmes pour choisir comment agréger les classes. La figure 2.2 montre un exemple d'un cas particulier de la hiérarchie de groupes (ou dendogramme) obtenue par agrégations successives à partir d'un petit ensemble de données bi-dimensionnelles :

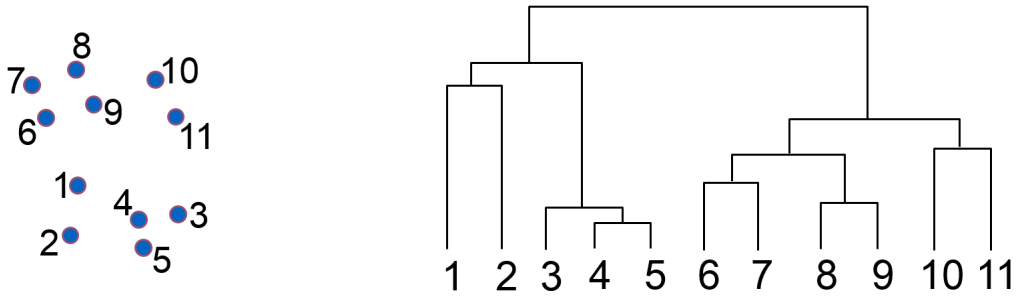


FIGURE 2.2 – Exemple de classification ascendante hiérarchique.

La méthode suppose qu'on dispose d'une mesure de dis-similarité entre les individus ; dans le cas de points situés dans un espace euclidien, on peut utiliser la distance comme mesure de dissimilarité. Le principe est de regrouper (ou d'agréger), à chaque itération, les données et/ou les groupes les plus proches qui n'ont pas encore été regroupé(e)s. Initialement, chaque individu forme une classe. On cherche à réduire itérativement le nombre de classes à $nb_{classes} < N$. À chaque étape, on fusionne deux classes, conduisant ainsi à réduire le nombre de classes. Les deux classes choisies pour être fusionnées sont celles qui sont les plus proches selon le respect d'une certaine métrique de distance. Les classes dont la dissimilarité entre elles est minimale seront fusionnées et la valeur de la dis-similarité est considérée comme indice d'agrégation. Ainsi, on rassemble d'abord les individus les plus proches donnant à la première itération un indice d'agrégation faible, puis celui-ci augmente d'itération en itération.

2.2.3 Partitionnement spectral

Étant donné un ensemble de N échantillons, il est possible d'obtenir une représentation détaillée de cet ensemble sous la forme d'un graphe pondéré, noté $\mathcal{G}(V, E, W)$. V désigne l'ensemble des N nœuds du graphe, correspondant aux échantillons, E est l'ensemble des liaisons ou arcs entre les nœuds du graphe, et $\mathbf{W}$ est la matrice de poids des arcs (ou matrice d'adjacence figure 2.3), symétrique et non négative (l'élément w_{ij} indique la similarité entre les échantillons $\mathbf{x}_i$ et $\mathbf{x}_j$).

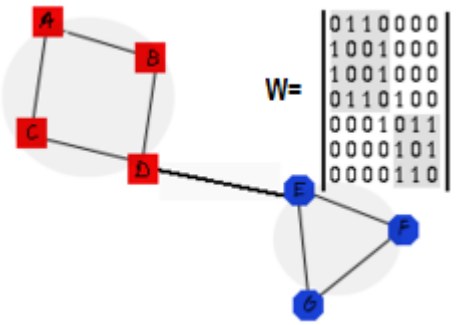


FIGURE 2.3 – Matrice d'adjacence et graphe correspondant

Le partitionnement spectral est une des techniques de partitionnement qui est basée sur la matrice de similarité entre les échantillons [9]. La technique consiste à calculer les valeurs

propres et les vecteurs propres associés de la matrice de similarité. Une partie de ces vecteurs propres forme un sous-espace de faible dimension. Le fondement du partitionnement spectral consiste à projeter les échantillons sur ce sous-espace. Dans ce sous-espace, on utilise une méthode de partitionnement (K-means le plus souvent) pour identifier les clusters.

L'objectif de la construction d'un graphe pondéré est de modéliser les relations de voisinage entre les échantillons. Il existe plusieurs façons de définir cette relation de voisinage entre les points dont :

— Graphe de voisinage ϵ : chaque sommet est relié à des sommets compris dans une balle de rayon ϵ où ϵ est une valeur réelle qui doit être accordée pour capter la structure locale des données,

— Graphe des k plus proches voisins : chaque sommet est connecté à ses k voisins les plus proches où k est un nombre entier qui contrôle les relations locales de données. Il existe deux types de similarité dans ce cas : le k-NN mutuel où la relation du voisinage est définie comme étant un "ET EXCLUSIF", c'est à dire que les k individus doivent être mutuellement voisins, et le k-NN normal où le voisinage est établi par un "OU EXCLUSIF".

— Graphe totalement connecté : tous les sommets (nœuds) ayant des similitudes non nulles sont connectés entre eux.

La construction de la matrice $\mathbf{W}$ est basée sur le choix de la fonction de similarité. Cette fonction dépend essentiellement du domaine de provenance des données (par exemple, fouille de documents, fouille de données web, etc.), mais également du type de données à traiter (qui peuvent être décrites par des variables numériques, catégorielles, binaires, etc.). Dans la littérature, il existe plusieurs techniques permettant d'obtenir la similarité entre deux échantillons. La matrice définie à partir du noyau représente la similarité entre les échantillons, et est donnée par :

$$w_{ij} = \exp\left(\frac{-\mathcal{D}^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right) \tag{2.2}$$

avec $\mathcal{D}$ une mesure de distance (de type Euclidienne, Manhattan, Minkowski, etc.), et σ, un paramètre d'échelle dont la valeur est fixée a priori.
Afin de détecter la structure des données, les méthodes de partitionnement utilisent les vecteurs propres d'une matrice Laplacienne $\mathbf{L}$. Pour calculer cette matrice $\mathbf{L}$, on pose $\mathbf{D}$, la matrice diagonale des degrés définie par

$$\mathbf{D} = \begin{pmatrix} d_{11} & 0 & \cdots & 0 \\ 0 & d_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{NN} \end{pmatrix}$$

où $d_{ii} = \sum_{j=1}^{N} w_{ij}$, représente le degré du $i^{ème}$ nœud du graphe $\mathcal{G}$. Il est alors possible de construire la matrice Laplacienne $\mathbf{L}$ en utilisant une normalisation parmi les différentes possibilités dans [9] à savoir :

— $\mathbf{L} = \mathbf{W}$ sans aucune normalisation,

— $\mathbf{L} = \mathbf{D}^{-1}\mathbf{W}$ avec normalisation par division,

— $\mathbf{L} = \mathbf{D}^{\frac{-1}{2}}\mathbf{W}\mathbf{D}^{\frac{-1}{2}}$ avec normalisation par division symétrique,

— $\mathbf{L} = \left(\frac{\mathbf{W}+d_{max}\mathbf{I}-\mathbf{D}}{d_{max}}\right)$ avec normalisation additive où $d_{max} = max(d_{ii})$, désignant le degré maximum de $\mathbf{D}$ et $\mathbf{I}$ étant la matrice identité.

Pour obtenir k clusters, les premiers k vecteurs propres orthogonaux associés aux k plus grandes valeurs propres de la matrice $\mathbf{L} \in \mathbb{R}^{N\times N}$ sont calculés [10]. Ces vecteurs sont rangés dans une matrice $\mathbf{U}$ telle que $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k]$. Ensuite, différentes techniques peuvent être appliquer sur la matrice $\mathbf{U}$, pour obtenir une partition. L'algorithme le plus utilisé pour le partitionnement dans le nouvel espace est de type K-means [11].

La phase de pré-traitement sur la matrice $\mathbf{L}$ permet de tirer profit des propriétés spectrales de cette matrice pour capter le maximum d'information intrinsèque au sein des données. La représentation spectrale obtenue, $\mathbf{U}$, permet de se placer dans un sous-espace où la différence entre les groupements de données est plus importante que dans l'espace initial. En considérant différents concepts de similarité entre les points à travers la matrice Laplacienne, on constate une meilleure répartition des données [11]. La figure 2.4 donne une illustration d'un exemple de résultats obtenus sur des échantillons de données synthétiques dans $\mathbb{R}^2$ avec trois différentes classes. Sur les résultats obtenus en utilisant la méthode K-means directement sur les données (2.4(a)) et en faisant la méthode de partitionnement spectral, on constate la capacité qu'a la méthode de partitionnement spectral de détecter des clusters assez complexes. En effet sur la figure 2.4(b), la méthode de partitionnement en combinaison avec l'algorithme K-means permet de détecter des clusters dont il serait difficile que l'algorithme K-means seul puisse les détecter comme le montre les résultats de la figure 2.4(d). Cependant cette méthode reste assez sensible au choix du paramètre σ de la matrice d'affinité. Différentes valeurs de σ peuvent conduire à différents résultats de partitionnement.

2.3 Apprentissage supervisé

En apprentissage supervisé, le but est de déterminer une nouvelle sortie y_i à partir d'une nouvelle entrée $\mathbf{x}_i$, connaissant un ensemble d'observations $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_{nl}, y_{nl})\}$, où pour chaque échantillon de données on donne l'indice de sa classe d'appartenance. Lorsque les y_i prennent des valeurs discrètes, on parle d'un problème de classification. En classification binaire, par exemple, on cherche à attribuer à $\mathbf{x}$ une étiquette "0" ou "1", tandis que des y_i à valeurs réelles nous placent dans le cadre de la régression. Dans ce travail de thèse, nous présentons de techniques de classification supervisée où l'objectif est d'estimer la classe d'appartenance non connue (ou étiquette) $\widehat{y_i} = f(\mathbf{x}_i)$, des échantillons non étiquetés $\{\mathbf{x}_i\}_{i=nl+1}^{nl+nu}$, avec nl et nu qui représentent respectivement le nombre d'échantillons étiquetés et non étiquetés.

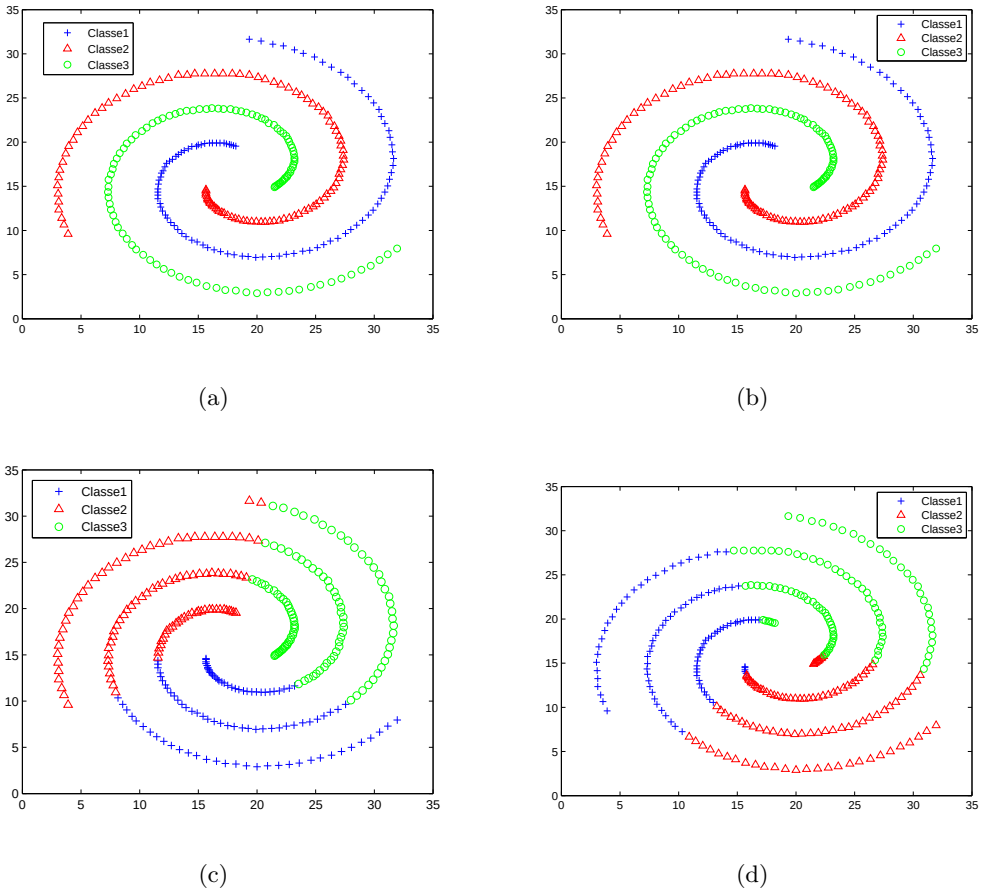


FIGURE 2.4 – 2.4(a) Données originales ; 2.4(b) avec la matrice de Laplacien normalisée symétrique avec différentes similarités : (1) voisinage $\epsilon = 2$, (2) normal KNN=3 et (3) mutuel KNN=5 ; 2.4(c) résultats partitionnement obtenus avec Laplacien normalisé et similarité totalement connectée ; 2.4(d) par k-means.

2.3.1 Plus proches voisins

La méthode des k plus proches voisins (k-PP)[12]-*k-nearest neighbors (k-NN)*- raisonne avec le principe sous-jacent : "dis moi qui sont tes amis, je te dirai qui tu es". Plus précisément, k-NN a pour but de classifier des points cibles appartenant à des classes inconnues en fonction de leurs distances par rapport à des points constituant un échantillon d'apprentissage dont la classe est connue a priori. Il s'agit d'une généralisation de la méthode du 1-plus proche voisin (NN). Elle consiste à trouver un voisinage de taille égale à k, qui représente l'ensemble des éléments les plus proches de l'échantillon à classer. Formellement, soit $\Omega = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ l'ensemble d'apprentissage où $y_i \in \{1, \cdots, K\}$ dénote la classe des différents individus. Généralement, pour estimer la sortie y associée à une nouvelle entrée $\mathbf{x}$, la méthode consiste à prendre en compte les échantillons d'apprentissage dont l'entrée est la plus proche de la nouvelle entrée, selon un critère de similarité défini. L'affectation est donnée par un vote majoritaire des échantillons les plus proches de $\mathbf{x}$ mesurés par une fonction de distance. Si $k = 1$, le cas est simplement assigné à la classe de son voisin le plus proche. Lorsque $k \geq 2$, la classe qui représente le maximum d'appartenance parmi les k plus proches voisins de $\mathbf{x}$ est sélectionnée pour la prédiction de y. La figure 2.5 donne un exemple de données issues de deux différentes classes matérialisées par les triangles et les étoiles. La décision de classement de l'échantillon "A" est basée sur le nombre de ses plus proches voisins. Le point "A" serait affecté à la classe dont la distance entre les points est la plus petite. La méthode de k plus proches voisins nécessite une capacité importante d'espace mémoire et un temps de calcul important pour réaliser les calculs des distances (afin de comparer et ne retenir que les plus petites), et ceci peut rendre la méthode assez complexe pour des grandes bases de données.

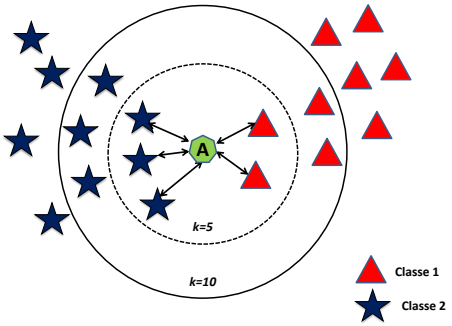


FIGURE 2.5 – Exemple de k plus proches voisins. Dans le cercle en pointillé, le point A dispose de 5 plus proches voisins, et dans le cercle supérieur, le point A dispose de 10 voisins.

2.3.2 Moyenne la plus proche

Cette méthode consiste à classer un nouvel échantillon de données dans une classe dont la distance est minimale entre cet échantillon et une des moyennes représentatives des clusters de la base d'apprentissage [13]. Considérons un jeu de données $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\}$ appartenant à K différentes classes avec $y_i \in \{1, \cdots, K\}$, où le nombre de classe K est

connu a priori. Les K ensembles sont répartis en $\{\omega_1, \omega_2, \cdots, \omega_K\}$ avec $(K \leq N)$ et ω_i un ensemble qui contient les échantillons de la classe i de taille N_k équivalente au nombre de points appartenant à un même groupe et $N = \sum_{k=1}^{K} N_k$. L'objectif de la méthode c'est d'utiliser les moyennes pour classer un nouveau échantillon. Les K moyennes des différents groupes définies telles que :

$$m_k = \frac{1}{N_k} \sum_{\substack{\mathbf{x}_i \in \omega_k \\ i=1,\cdots,N_k}} \mathbf{x}_i.$$

permettent de classer un échantillon $\mathbf{x}$ donné, dont on souhaite prédire sa classe d'appartenance. Le principe consiste simplement à calculer la distance euclidienne entre ce point et les K moyennes des groupes. Il est alors affecté au groupe de moyenne la plus proche.

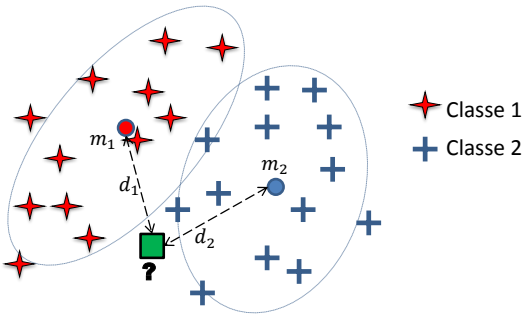


FIGURE 2.6 – Exemple d'illustration du principe de la moyenne la plus proche. Les distances d_1 et d_2 caractérisent la classe d'appartenance du point inconnue matérialisé en vert.

2.3.3 Analyse linéaire discriminante

L'analyse discriminante linéaire(LDA) [14] est une généralisation de la méthode de Fisher [15], utilisée en apprentissage automatique pour trouver une combinaison linéaire de variables qui caractérisent ou séparent deux ou plusieurs classes d'objets. La combinaison résultante peut être utilisée comme classificateur linéaire ou, plus communément, pour la réduction de la dimension avant une classification ultérieure. Elle permet d'expliquer et de prédire l'appartenance d'un individu à une classe donnée en considérant la connaissance a priori des étiquettes des échantillons d'apprentissage [16]. Le principe de la méthode de LDA est de transformer les données initiales en les projetant dans un sous-espace de dimension réduite de telle sorte que les échantillons d'une même classe soient peu dispersés et ceux d'une classe à l'autre soient éloignés. La figure 2.7 montre un exemple qui illustre le principe de la méthode.

Afin de réaliser cette tâche, la méthode se base sur la mise en évidence des matrices de variance-covariance inter-classe et intra-classe. En considérant une matrice de données $\mathbf{X} \in \mathbb{R}^{N \times d}$, qui contient un ensemble de N échantillons observés sur d variables répartis en K groupes où chaque groupe possède N_k échantillons, la LDA se base sur la recherche d'une matrice de projection $\mathbf{q}$ qui maximise le critère de Fisher défini par :

$$J(\mathbf{q}) = \underset{\mathbf{q_{opt}}}{\operatorname{argmax}} \ \frac{\det(\mathbf{q}^T S_b \mathbf{q})}{\det(\mathbf{q}^T S_w \mathbf{q})} \tag{2.3}$$

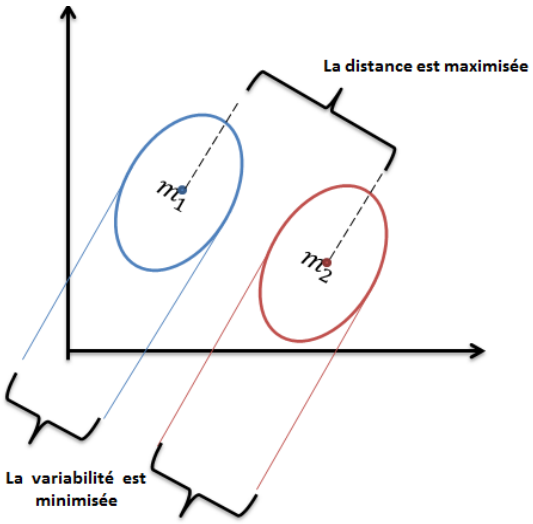


FIGURE 2.7 – Principe de l'analyse linéaire discriminante

où S_b, S_w désignent les matrices de variance inter-classe et intra-classe, respectivement définies par :

$$S_b = \sum_{k=1}^{K} N_k (m_k - m)^T (m_k - m),$$
$$S_w = \sum_{k=1}^{K} \sum_{\mathbf{x}_j \in \omega_k} (\mathbf{x}_j - m_k)^T (\mathbf{x}_j - m_k), \tag{2.4}$$

avec $m = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i)$ le vecteur contenant la moyenne totale des N échantillons, m_k le vecteur contenant la moyenne de la $k^{ème}$ classe.

La solution optimale $\mathbf{q}_{opt}$ de l'équation (2.3) est donnée par les vecteurs propres de la matrice $S_w^{-1} S_b$ [15, 17, 18]. Le rang de la matrice S_b est borné par $K-1$ [19], il en découle l'existence d'au plus $K-1$ vecteurs propres discriminants correspondant aux valeurs propres non nulles. L'obtention de la matrice de projection $\mathbf{q}$, d'une part, nécessite que la matrice de variance S_w soit non singulière pour être inversible. D'autre part, la décomposition spectrale de la matrice $S_w^{-1} S_b$ peut s'avérer complexe lorsque les données possèdent un grand nombre de variables initiales (large valeur de d). Ceci rend la réalisation de la méthode difficile, et il est fondamental de trouver des approches nécessaires afin de contourner ces problèmes.

2.3.4 Séparateurs à vaste marge

Le séparateur à vaste marge (SVM)-*machine à vecteurs de support*- est une technique d'apprentissage supervisée destinée à résoudre des problèmes de discrimination ou de régression [20, 21, 22]. La méthode SVM est bien connue pour la classification binaire où les classes sont linéairement séparables. Dans le cas où la variable de sortie compte plus de deux modalités, il existe plusieurs façon d'étendre directement le cas binaire au cas multi-classe [23]. La méthode SVM repose sur l'application d'algorithmes de recherche de règles de décision linéaires et ramène le problème de la discrimination à celui de la recherche d'un hyperplan

séparateur qui maximise la marge définie par la distance entre la frontière séparatrice et les échantillons les plus proches.

Supposons que nous disposons des échantillons de données i.i.d. dans un espace Hilbertien $\{(\mathrm{x}_i, y_i)\}_{i=1}^{N}, \mathbf{x}_i \in \mathcal{H}, y_i \in \{+1, -1\}$. Chaque hyperplan de $\mathcal{H}$ peut être écrit par

$$\{\mathbf{x} \in \mathcal{H} |\ \langle \omega, \mathbf{x} \rangle + b = 0\},\ \ w \in \mathcal{H}, b \in \mathbb{R}.$$

La surface séparatrice associée à la règle de décision correspond à l'hyperplan dont l'expression

$$\langle \omega, \mathbf{x} \rangle + b = 0 \tag{2.5}$$

est vérifiée où ω est un vecteur orthogonal à l'hyperplan et b un scalaire d'ajustement du plan discriminant. Le problème de la maximisation de la marge peut être résolu en utilisant un problème d'optimisation sous contraintes linéaires tel que :

$$\texttt{minimiser}\ \frac{1}{2}\|\omega\|^2$$

sous les contraintes : $y_i(\langle \omega, \mathbf{x} \rangle + b) \geqslant 1, \forall i = 1, \cdots, N$. Ceci peut se résoudre par la méthode classique des multiplicateurs de Lagrange, où le lagrangien est donné par

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{N} \alpha_i \left(y_i(\langle \omega, \mathbf{x}_i \rangle + b) - 1 \right),$$

avec $\alpha_i \geqslant 0$ qui sont les coefficients de Lagrange. La solution ω détermine l'orientation de l'hyperplan et est généralement donnée sous la forme

$$\omega = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i,$$

avec les coefficients α_i qui désignent les solutions du problème quadratique dual donné par :

$$\underset{\alpha \in \mathbb{R}^N}{\texttt{maximiser}}\ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

sous les contraintes : $\forall i, \alpha_i \geqslant 0$ et $\sum_{i=1}^{N} \alpha_i y_i = 0$. Les points $\mathbf{x}_i$, pour lesquels les coefficients α_i sont positifs, sont appelés vecteurs de supports. La solution générale de la surface séparatrice de l'équation (2.5) a la forme :

$$f(\mathbf{x}) = sign(\langle \omega, \mathbf{x} \rangle + b).$$

qui peut s'exprimer aussi sous forme du produit scalaire :

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{N} \alpha_i y_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b \right).$$

La figure 2.8 donne un exemple illustratif du principe de la méthode, où l'objectif est de pouvoir maximiser la marge délimitée par l'hyperplan séparateur.

Dans le cas où les données ne sont pas linéairement séparables, la méthode SVM transforme l'espace de représentation des données d'entrées en un espace de plus grande dimension dans lequel il est possible de trouver une fonction de séparation linéaire. Ceci est réalisé

grâce à l'introduction d'une certaine fonction noyau [24], qui doit respecter les conditions du théorème de Mercer [25]. Dans ce cas, le but est de faire appel a une fonction implicite non linéaire transformant l'espace de départ en un espace de plus grande dimension image de l'espace d'origine à travers la transformation $\phi(.)$. Il convient de noter que le calcul de fonction $\phi(.)$ n'est pas explicite dans la méthode, et seuls les produits scalaires $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}^T) \rangle$ sont requis dans la solution du problème. Lorsque la taille d'échantillon est très grande, la recherche de la surface séparatrice peut être assez complexe du fait que l'introduction de la fonction noyau qui peut nécessiter beaucoup d'espace mémoire pour le stockage de la matrice de Gram $\phi(\mathbf{x}_i).\phi(\mathbf{x}^T)$, ce qui rend la méthode difficile dans le cas des grandes bases de données.

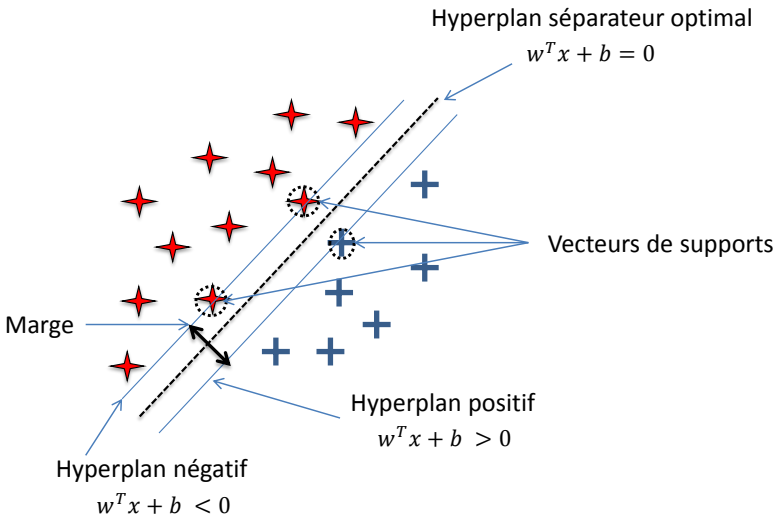


FIGURE 2.8 – Séparateurs vaste marge.

2.4 Réduction de dimension

La réduction de dimension consiste à trouver un nouvel espace constitué de combinaisons (linéaires ou non) des variables initiales. Son objectif est de sélectionner ou d'extraire un sous-ensemble optimal de caractéristiques pertinentes qui préservent une grande partie de l'information véhiculée par les données. La sélection de ce sous-ensemble de caractéristiques permet d'éliminer les informations non-pertinentes et redondantes selon le choix d'un critère donné. Cette sélection/extraction permet de réduire les variables initiales afin de se ramener dans un espace réduit dans lequel le traitement des données est plus facile à effectuer. En effet, les principaux points forts de la réduction de dimension sont :

- faciliter l'interprétation, l'analyse et la visualisation des données,
- réduire l'espace de stockage nécessaire,
- réduire le temps d'apprentissage et d'utilisation,
- identifier les facteurs informatifs/non-informatifs.

Il existe plusieurs approches qui permettent de réaliser la tâche de réduction de dimension. Ces approches peuvent être classées en deux grandes catégories qui sont : (1) les méthodes

de sélection des variables qui consistent à choisir des caractéristiques dans l'espace d'origine et (2) les méthodes d'extraction des variables qui visent à sélectionner des caractéristiques via une certaine fonction de transformation. Dans cette thèse, nous nous intéressons aux méthodes d'extraction des variables. Pour la suite de cette section, nous présentons quelques techniques couramment utilisées pour la réduction de la dimension.

2.4.1 Analyse en composante principale

L'analyse en composantes principales (ACP)-*Principal Component Analysis (PCA)*-est l'une des méthodes d'analyse multivariées les plus utilisées. Lorsque la dimension des variables est élevée, il est impossible d'appréhender la structure des données et la proximité entre les observations en se contentant d'analyser des statistiques descriptives univariées ou même une matrice de corrélation (ou de covariance). L'ACP effectue une réduction de dimension par projection des points originaux de dimension d dans un sous-espace vectoriel de dimension plus réduite k en déterminant les axes principaux qui maximisent la variance expliquée.

La solution du problème de maximisation de la variance donne à l'ACP un double sens : la projection de l'espace d'origine de dimension d dans le sous-espace de dimension k fait de l'ACP une technique de minimisation de l'erreur quadratique d'estimation et la projection inverse (du sous-espace de dimension k vers le sous-espace de dimension d) permettant d'estimer les variables initiales fait considérée l'ACP comme une technique de maximisation de la variance des projections.

Soit $\mathbf{X}$ une matrice de N données appartenant à $\mathbb{R}^d$. On suppose que $\mathbf{X}$ est centré. L'objectif de l'analyse en composantes principales est de trouver un sous-espace de dimension k ($k < d$) qui permet d'avoir une représentation réduite de $\mathbf{X}$. Pour cela, on associe un vecteur $\mathbf{z}_i \in \mathbb{R}^k$ à une observation $\mathbf{x}_i$ à travers une transformation linéaire définie par $\mathbf{U} \in \mathbb{R}^{d\times k}$ où $\mathbf{U}$ est une matrice de transformation orthogonale de $\mathbb{R}^{d\times k}$ et vérifie $\mathbf{U^T U} = \mathbf{I}$. Ceci revient donc à poser

$$\mathbf{z}_i = \mathbf{U}^T\mathbf{x}_i \quad \text{avec} \quad \mathbf{U} = [\mathbf{u_1}, \cdots, \mathbf{u_k}], \mathbf{u_i} \in \mathbb{R}^d.$$

$\mathbf{U}$ est appelée aussi matrice de changement de base où les vecteurs de la nouvelle base sont orthogonaux deux à deux, i.e, $\mathbf{u_i^T u_j} = 0$ si $i \neq j$. Les composantes $\mathbf{z}_i$, avec $(i = 1, \cdots, k)$, du vecteur caractéristique $\mathbf{z}$ représentent les composantes principales projetées du vecteur $\mathbf{x_i}$ dans le sous-espace réduit. La reconstruction de $\mathbf{x}_i$ à partir de $\mathbf{z}_i$ est donnée par :

$$\widehat{\mathbf{x}}_i = \mathbf{U}\mathbf{z}_i = \mathbf{U}\mathbf{U}^T\mathbf{x}_i.$$

Lorsque l'erreur quadratique d'estimation de $\mathbf{x}$ est minimale, on dit que la matrice de projection $\mathbf{U}$ est optimale. Ce problème de recherche des axes principaux se traduit mathématiquement en un problème d'optimisation et s'exprime par :

$$\mathbf{U}_{opt} = \underset{\mathbf{U}}{\operatorname{argmin}}\ \mathcal{J}_e(\mathbf{U}) \tag{2.6}$$

où $\mathcal{J}_e$ définit le critère d'erreur d'estimation de l'ACP. En respectant la contrainte d'orthogonalité de la matrice de projection $\mathbf{U^T U = I}$, ce critère peut être réécrit sous la forme suivante :

$$\begin{aligned}
\mathcal{J}_e &= \mathbb{E}\left[\|\mathbf{x_i} - \hat{\mathbf{x}}_\mathbf{i}\|^2\right] = \mathbb{E}\left[(\mathbf{x_i} - \mathrm{UU^T x_i})^T(\mathbf{x_i} - \mathrm{UU^T x_i})\right] \\
&= \mathbb{E}\left(\mathbf{x_i}^T\mathbf{x_i} - 2\mathbf{x_i}^T\mathbf{U}\mathbf{U}^T\mathbf{x}_i + \mathbf{x}_i^T\mathbf{U}\mathbf{U}^T\mathbf{U}\mathbf{U}^T\mathbf{x}_i\right) \\
&= \mathbb{E}\left(\mathbf{x_i}^T\mathbf{x_i} - \mathbf{x_i}^T\mathbf{U}\mathbf{U}^T\mathbf{x}_i\right) = \mathbb{E}\left(\mathbf{x_i}^T\mathbf{x_i} - \mathbf{z_i}^T\mathbf{z}_i\right) \\
&= \mathbb{E}\left(trace(\mathbf{x_i}^T\mathbf{x_i} - \mathbf{z_i}^T\mathbf{z}_i)\right) = \mathbb{E}\left(trace(\mathbf{x_i}\mathbf{x_i}^T - \mathbf{z}_i\mathbf{z_i}^T)\right) \\
&= \mathbb{E}\left[trace(\mathbf{x_i}\mathbf{x_i}^T) - trace(\mathbf{U}^T\mathbf{x_i}\mathbf{x_i}^T\mathbf{U})\right] \\
&= trace(\mathbf{\Sigma}) - trace(\mathbf{U^T\Sigma U}).
\end{aligned} \tag{2.7}$$

Minimiser l'expression de $\mathcal{J}_e$ revient simplement à maximiser le deuxième terme de $\mathcal{J}_e$ qui correspond à $trace(\mathbf{U^T\Sigma U})$, où Σ représente la matrice de covariance empirique. En conséquence, l'équivalence entre la maximisation de la variance des données projetées et la minimisation de l'erreur quadratique devient évidente et le problème de l'ACP se réduit à :

$$\mathbf{U}_{opt} = \underset{\mathbf{U}}{\operatorname{argmin}}\ \mathcal{J}_e(\mathbf{U}) = \underset{\mathbf{U}}{\operatorname{argmax}}\ trace(\mathbf{U^T\Sigma U})$$

L'estimation de la matrice de projection orthogonale U_{opt} se fait principalement de deux façons. La première technique consiste à calculer des valeurs et vecteurs propres de la matrice de covariance empirique $\mathbf{\Sigma}$ des données. La deuxième méthode est basée sur la décomposition en valeurs singulières de la matrice des données $\mathbf{X}$.

2.4.1.1 Avec la covariance

Considérons des données sous la forme d'une matrice centrée $\mathbf{X} \in \mathbb{R}^{N\times d}$, et considérons également la matrice de covariance empirique de taille $d \times d$ définie par $\mathbf{\Sigma} = \mathbf{X}^T\mathbf{X}$.

Soit $\mathbf{U} = [\mathbf{u_1}, \cdots, \mathbf{u_d}]$ avec $\mathbf{u_i}$ un vecteur unitaire de $\mathbb{R}^d$ tel que $\|\mathbf{u_i}\|^\mathbf{2} = \mathbf{u_i^T u_i} = \mathbf{1}$, suivant lequel la variance de la projection de $\mathbf{x}$ est maximale. D'un point de vue optimisation de la maximisation de la variance, la fonction objective de l'ACP est donnée par

$$\begin{aligned}
&\underset{\mathbf{u}_i}{\operatorname{argmax}} \quad trace(\mathbf{u_i}^T\mathbf{\Sigma u_i}) \\
&\qquad\qquad s.t \quad \mathbf{u_i}^T\mathbf{u_i} = 1.
\end{aligned} \tag{2.8}$$

Lorsque la matrice $\mathbf{\Sigma}$ est une matrice réelle et symétrique et $\mathbf{u_i}$ est un vecteur réel non nul, la solution du problème (2.8) est bien connue sous forme de quotient de Rayleigh $r(\mathbf{u_i})$ [26] donné par :

$$r(\mathbf{u_i}) = \frac{\mathbf{u_i^T\Sigma u_i}}{\mathbf{u_i^T u_i}}.$$

La solution qui maximise l'équation (2.8) est donnée par $\mathbf{u_i}$ qui réprésente le vecteur propre correspondant à la plus grande valeur propre de $\mathbf{\Sigma}$ définie par le scalaire $r(\mathbf{u_i})$. La diagonalisation de la matrice de covariance empirique $\mathbf{\Sigma = U\Delta U^T}$ donne les vecteurs propres $\mathbf{u_i}$ (égales aux vecteurs colonnes de $\mathbf{U}$) et leurs valeurs propres associées λ_i qui sont solution

du problème 2.8. Ainsi, les valeurs propres répresentent les variances des données projetées $\mathbf{z_i}$ sur les axes représentés par les vecteurs propres $\mathbf{u}_i, (i = 1, \ldots, d)$. La direction optimale suivant laquelle la variance de la projection du vecteur de données $\mathbf{x}$ est maximale, est représentée par le vecteur propre $\mathbf{u}_i$ correspondant à la valeur propre maximale λ_i. Le second axe qui contribue à la maximisation de la variance est orthogonal au premier axe suivant la contrainte d'orthogonalité. De façon plus générale, le sous espace vectoriel de dimension k qui assure une dispersion maximale des observations est défini par une base orthonormée formée des k vecteurs propres, communément appelés axes principaux, correspondant aux k plus grandes valeurs propres de la matrice $\mathbf{\Sigma}$.

Les valeurs propres donnent l'information véhiculée par chaque axe correspondant selon le pourcentage cumulé. L'amplitude de chaque valeur propre quantifie pour chaque axe la quantité de l'information encodée qu'il véhicule. Cela donne un intérêt considérable de la méthode de l'analyse en composante principale pour la réduction de la dimension des données. En effet, la technique permet de caractériser les directions orthogonales d'un espace de données porteuses du maximum d'information au sens de la maximisation des variances de projections. Lorsque les données sont issues d'un espace de grande dimension (d large), il est parfois difficile de passer par la diagonalisation de la matrice de covariance pour obtenir les axes principaux. La méthode de l'analyse en composantes principales devient difficilement réalisable avec un temps de calcul assez complexe. Dans ce cas il est préférable de passer par la technique de décomposition en valeurs singulières pour calculer les axes principaux.

2.4.1.2 Avec la décomposition en valeurs singulières

On appelle décomposition en valeurs singulières (SVD) [27], la décomposition d'une matrice rectangulaire de $\mathbb{R}^{N\times d}$ sous la forme

$$\mathbf{X} = \mathbf{USV^T}, \tag{2.9}$$

où $\mathbf{U}$ est une matrice orthogonale de taille $N \times N$ qui contient les vecteurs singuliers à droite, $\mathbf{S}$ une matrice semi-diagonale de taille $N \times d$ qui contient sur sa diagonale les valeurs singulières et $\mathbf{V}$ est une matrice orthogonale de taille $d \times d$ contenant les vecteurs singuliers à gauche de la matrice $\mathbf{X}$. En vertu des propriétés matricielles en algèbre linéaire, les composantes principales qui maximisent la variance et qui minimisent l'erreur de reconstruction du problème (2.7) peuvent être également déterminées par la décomposition en valeurs singulières (2.9). En effet, en réécrivant l'expression de la covariance avec la forme de la SVD, on sait que :

$$\begin{aligned} \mathbf{\Sigma} = \mathbf{X}^T\mathbf{X} &= (\mathbf{USV^T})^\mathbf{T}(\mathbf{USV^T}) \\ &= (\mathbf{VSS^TV^T}) = \mathbf{V\Delta V^T} \end{aligned} \tag{2.10}$$

La relation entre la forme diagonalisée de la matrice de covariance et celle de l'équation (2.10) montre que les vecteurs singuliers à droite de la SVD de $\mathbf{X}$ sont en fait les vecteurs propres de la matrice de covariance, ce qui justifie que la matrice $\mathbf{\Sigma}$ partage certaines

propriétés spectrales avec la matrice $\mathbf{X}$. Par ailleurs, puisque la matrice $\mathbf{V}$ est orthogonale, à partir de l'équation (2.9), les composantes principales peuvent s'exprimer par :

$$\mathbf{z_j} = \mathbf{V}^T \mathbf{x_j}.$$

L'expression de la forme tronquée de la SVD donne la matrice approximée $\mathbf{X}_k$ de $\mathbf{X}$ telle que :

$$\mathbf{X}_k = \mathbf{U_k S_k V_k^T}, \tag{2.11}$$

où uniquement k premiers vecteurs de $\mathbf{U}, \mathbf{V}$ et $\mathbf{S}$ sont considérés. En ACP, les données sont projetées dans le sous-espace engendré par les k vecteurs propres associés aux k plus grandes valeurs propres. Pour réduire la dimension des données de d à k, les k premières colonnes de $\mathbf{U}$, et $k \times k$ partie supérieure à gauche de $\mathbf{S}$ sont retenues. Ainsi, les k premières composantes principales $\mathbf{Z_k} = [\mathbf{z_1}, \cdots, \mathbf{z_k}]$ sont données par $\mathbf{Z_k} = \mathbf{X V_k} = \mathbf{U_k S_k}$. L'approximation de la matrice de données $\mathbf{X}$ donnant la meilleure approximation de rang k de $\mathbf{X}$, et est donnée par $\mathbf{X}_k$ conduisant à une erreur de reconstruction définie par :

$$\|\mathbf{X} - \mathbf{X V}_k \mathbf{V}_k^T\|^2 = \|\mathbf{X} - \mathbf{U}_k \mathbf{U}_k^T \mathbf{X}\|^2 = \sum_{i=k+1}^{d} \lambda_i^2.$$

Le calcul des composantes principales par la décomposition en valeurs singulières est une solution qui peut être utilisée pour calculer les mêmes composantes principales que dans le cas de la diagonalisation de la matrice $\mathbf{\Sigma}$. Il faut aussi noter que la matrice $\mathbf{X X^T} = \mathbf{U \Delta U^T}$ possède les mêmes vecteurs singuliers à gauche que la matrice $\mathbf{X}$. Ainsi, lorsque $N \ll d$, il est plus économique de calculer les composantes principales à travers la SVD de $\mathbf{X}$ plutôt que d'effectuer la diagonalisation de $\mathbf{\Sigma}$.

2.4.1.3 Analyse en composante principale à noyau

L'analyse en composante principale à noyau (ou Kernel PCA en anglais (KPCA)) permet de trouver des fonctions de décision non linéaires, tout en s'appuyant fondamentalement sur l'ACP linéaire. Le principe de l'ACP à noyau réside particulièrement sur le fait que N points de données ne puissent en général pas être linéairement séparable dans l'espace de dimension $d < N$. En considérant N échantillons de données observés dans l'espace $\mathbb{R}^d$, il est possible de les transformer dans un espace de dimension $\mathbb{R}^N$ via une fonction de transformation $\varphi(\mathbf{x}_i)$ telle que $\varphi : \mathbb{R}^d \mapsto \mathbb{R}^N$. Après cette transformation des données, l'ACP linéaire est effectuée sur les nouvelles données résultantes dans l'espace augmenté. Étant donné que le nouvel espace est généralement de très grande dimension, la méthode d'ACP à noyau emploie des noyaux remplissant les conditions de Mercer [25] au lieu de calculer explicitement la fonction de transformation. Ce noyau est une fonction $\mathbf{k}(\mathbf{x}, \mathbf{y})$ qui, pour toutes les données $\{\mathrm{x}_i\}_{i=1}^N$ donne lieu à une matrice positive $k_{ij} = \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$ [28]. On essaye généralement d'éviter de travailler dans l'espace des fonctions φ, et construire le noyau de taille $N \times N$ par :

$$\mathbf{K} = \mathbf{k}(\mathbf{x}, \mathbf{y}) = (\varphi(\mathbf{x}), \varphi(\mathbf{y})) = \varphi(\mathbf{x})^T \varphi(\mathbf{y})$$

où chaque colonne de $\mathbf{K}$ représente le produit scalaire d'un point de données transformé par rapport à tous les autres points transformés. La fonction la plus utilisée pour calculer

le noyau sont généralement le noyau Gaussien exprimé par :

$$\mathbf{k}(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/2\sigma^2)$$

avec σ un paramètre d'échelle ou déviation standard qui représente la largeur du noyau Gaussien. La méthode d'analyse en composante principale linéaire est réalisée sur la matrice du noyau $\mathbf{K}$. Cette méthode à noyau est très bien adaptée pour extraire les structures des données non linéaires. Cependant, lorsque l'on est en présence des données volumineuses, cela conduit à obtenir un $\mathbf{K}$ grand, et le stockage de cette matrice $\mathbf{K}$ d'une part peut devenir pratiquement impossible ou nécessite beaucoup d'espace mémoire. D'autre part, pour calculer les vecteurs propres et valeurs propres conduisant à l'obtention des composantes principales dans le nouveau espace, il faut faire la décomposition spectrale de la matrice $\mathbf{K}$. Cette décomposition est très coûteuse lorsque l'on est en présence d'une base de données volumineuse. Ce qui rend la méthode pratiquement infaisable pour les données massives. Nous nous sommes moins focalisés sur cette technique car notre travail est porté sur l'analyse des grandes bases de données et en grande dimension.

2.4.2 Réduction de dimension avec matrices aléatoires

En dehors des techniques précédemment présentées pour la réduction de dimension, il existe des alternatives permettent d'extraire des variables considérées informatifs et éliminer ceux qui sont moins informatifs ou qui véhiculent une information redondante. La réduction de dimension avec matrices aléatoires est une de ces techniques qui considère une projection basée sur des vecteurs aléatoires. Le principe consiste à transformer linéairement les données provenant d'un espace initial de grande dimension vers un espace de faible dimension où les bases de ce nouvel espace sont aléatoirement construites. Une matrice de projection $\mathbf{R}$, construite sur des principes aléatoires, permet cette transformation afin de conserver les caractéristiques au sein des données moyennant un terme d'erreur. Nous présentons dans la suite de cette section, les principales techniques qui se basent sur cette théorie.

2.4.2.1 Projection aléatoire

Une projection suivant des vecteurs aléatoires consiste à partir d'une matrice $\mathbf{R}$ de taille $d \times k$ formée par k vecteurs aléatoires, à construire une version réduite des échantillons de données $\mathbf{X}$ en dimension k par transformation linéaire $\mathbf{X}_{RP} = \mathbf{XR}$. La simplicité de cette approche réside dans le fait que les colonnes de la matrice $\mathbf{R}$ sont des vecteurs aléatoires, et dans un espace à grande dimension, cette matrice est suffisamment proche d'une matrice orthogonale et que la matrice $\mathbf{R}^T\mathbf{R}$ présente les même propriétés qu'une matrice identité [29].

Le lemme de Johnson-Lindenstrauss assure que les distances entre les échantillons peuvent être quasiment conservées lors de la réduction de dimension, en remettant à l'échelle les données transformées avec un facteur bien choisi.

Lemme 1 *[30] : Soient* $0 < \epsilon < 1/2$, N *et* p *deux entiers tels que* $p \geq p_0 = O(\log(N)/\epsilon^2)$.

Pour chaque ensemble d'échantillons de données $\mathbf{X}$ *de* N *points dans* $\mathbb{R}^d$*, il existe un ensemble de fonctions qui vérifient la transformation suivante :*

$$\begin{aligned} f : \mathbb{R}^d \to \mathbb{R}^p \quad \textit{telle que} \ : \\ (1-\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|^2 \leq (1+\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \quad \textit{avec} \ \ \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}. \end{aligned} \tag{2.12}$$

La conséquence de ce lemme garantit que pour tout ϵ défini ci-dessus et pour une métrique Euclidienne, un ensemble de N points dans $\mathbb{R}^d$ peut être plongé dans un espace de $\mathbb{R}^p$ (logarithmique en N) indépendant de la taille de l'espace initial où la transformation des données préserve la distance entre les échantillons et la distorsion des points engendrée est bornée par le terme ϵ. La fonction qui garantit cette transformation est généralement définie par la relation linéaire $f(\mathbf{x}) = \mathbf{xR}$, où $\mathbf{R}$ est une matrice de projection à caractère aléatoire dont les éléments suivent une distribution préalablement choisie. Parmi les possibilités de construction de la matrice $\mathbf{R}$, les plus communément utilisées sont les suivantes :

— Matrice aléatoire de taille $d \times p$ dont les éléments sont indépendants et identiquement distribués (i.i.d) suivant une loi normale $\mathcal{N}(0,1)$ [29].

— Matrice aléatoire de taille $d \times p$ dont les éléments sont i.i.d et prennent des valeurs égales à $\{-1, +1\}$ avec une probabilité de $\frac{1}{2}$ [31, 32].

— Matrice aléatoire de taille $d \times p$ dont les éléments sont i.i.d et prennent des valeurs égales à $\{-1, 0, +1\}$ avec probabilités respectives de $\{\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\}$ [33].

Cette technique offre un moyen simple et peu coûteux en temps de calcul afin de réduire les données de très grande dimension avec des propriétés intéressantes concernant la conservation des distances entre des paires de points. Cependant, le choix de p reste un problème délicat. Les éléments r_{ij} de la matrice $\mathbf{R}$ sont générés de manière aléatoire, ce qui conduit à obtenir des éléments différents à chaque étape de génération de $\mathbf{R}$. Ce caractère aléatoire induit généralement des résultats instables qui présentent une variance considérable sur la performance de la méthode.

2.4.2.2 Approximation de la SVD

Considérons une matrice $\mathbf{X} \in \mathbb{R}^{d \times N}$ qui possède un rang égal à ρ tel que $rang(\mathbf{X}) = \rho \leqslant \min\{N, d\}$. La valeur exacte de la décomposition en valeurs singulières de la matrice $\mathbf{X}$ est communément donnée par $\mathbf{X} = \mathbf{USV^T}$, où $\mathbf{U} \in \mathbb{R}^{d \times \rho}$, $\mathbf{S} \in \mathbb{R}^{\rho \times \rho}$ et $\mathbf{V} \in \mathbb{R}^{N \times \rho}$. D'une manière détaillée, cette décomposition s'écrit :

$$\mathbf{X}_k = \mathrm{U} \begin{pmatrix} \mathbf{S}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{V}^T = \sigma_1 u_1 v_1^T + \cdots + \sigma_k u_k v_k^T,$$

avec $\mathbf{S}_k$ la matrice contenant sur sa diagonale les éléments σ_i obtenus à partir de la SVD tronquée de $\mathbf{X}$, i.e, en conservant les k ($k < \rho$) premières colonnes de $\mathbf{S}$ et $\mathbf{0}$ est une sous matrice dont les éléments sont tous nuls. L'approximation à faible rang d'une matrice consiste à approcher la matrice de départ $\mathbf{X} \in \mathbb{R}^{d \times N}$ par une matrice de rang inférieur $\mathbf{Z} \in \mathbb{R}^{k \times N}$ de telle sorte qu'une certaine norme de l'erreur $\mathbf{X} - \mathbf{Z}$ soit minimale. Ce problème

est communément étudié pour toute norme unitaire invariante. La meilleure approximation de $\mathbf{X}$ de rang égal à k est la matrice $\mathbf{Z}$, solution du problème d'optimisation suivant [34] :

$$\underset{\text{rang}(Z)\leqslant k}{\texttt{argmin}} \ \|\mathbf{X}-\mathbf{Z}\|_F^2 = \|\mathbf{X}-\mathbf{X}_k\|_F^2, \tag{2.13}$$

où F indique la norme de Frobenius définie par $\|\mathbf{Z}\|_F = \sqrt{\sum_{i=1}^{k}\sigma_i^2(\mathbf{Z})}$. La matrice $\mathbf{A}$ qui minimise l'erreur dans l'équation (2.13) est donnée par la matrice $\mathbf{X}_k$ de la SVD tronquée [35, 36, 37]. Le calcul d'une telle matrice $\mathbf{X}_k$ a une complexité de calcul de l'ordre de $O\left(\text{Nd}\texttt{min}\{\text{N},\text{d}\}\right)$ opérations de calcul [35]. Lorsque $\texttt{min}\{\text{N},\text{d}\}$ est large, cette méthode demande un temps de calcul élevé qui rend le processus très couteux voir infaisable dans certaines conditions.

Pour optimiser le processus du calcul de la matrice $\mathbf{X}_k$, l'approximation de la SVD permet de calculer une matrice estimée de $\mathbf{X}_k$. Le principe combine la méthode de la SVD tronquée avec une projection aléatoire sur la matrice $\mathbf{X}$ afin de réduire le temps de calcul. Sarlos énonce dans [35] le théorème suivant :

Théorème 2 *: Considérant la matrice de départ $\mathbf{X}\in\mathbb{R}^{N\times d}$ et $\pi_{(.)}$ un opérateur de projection. Si le paramètre $0<\epsilon<1$ et $\mathbf{R}\in\mathbb{R}^{d\times p}$ la matrice aléatoire de Johnson-Lindenstrauss définie dans le Lemme 1 dont les éléments sont i.i.d de moyenne nulle et d'écart-type égal à 1, avec $p=O(k/\epsilon+k\log k)$, alors avec une probabilité au moins égale à 1/2, on estime que*

$$\|\mathbf{X}-\pi_{\mathbf{XR},k}(\mathbf{X})\|_F \leqslant (1+\epsilon)\|\mathbf{X}-\mathbf{X}_k\|_F.$$

Considérons un sous-espace $\mathcal{V}\leqslant\mathbb{R}^d$ avec $\mathbf{R}\in\mathcal{V}$ et posons $\mathbf{A}=\pi_{\mathcal{V}}(\mathbf{X})=\mathbf{XR}$ la matrice résultante de la projection de $\mathbf{X}$ sur $\mathcal{V}$ avec $\pi_{\mathcal{V},k}(\mathbf{A})=(\pi_{\mathcal{V}}(\mathbf{A}))_k$ qui denote la SVD tronquée de $\mathbf{A}$. Le théorème stipule que la projection de $\mathbf{X}$ sur l'espace engendré par les lignes de la matrice $\mathbf{A}$ (moyennant une orthonormalisation de $\mathbf{A}$), i.e, $\mathbf{A^TX}$, et en calculant la SVD tronquée de la matrice résultante, on obtient une approximation de la matrice $\mathbf{X}_k$ elle même avec une erreur d'approximation bornée par $1+\epsilon$. L'estimation est obtenue en procédant d'abord par le calcul de matrice $\mathbf{Q}$ telle que :

$$\mathbf{Q}=\mathbf{A}^T\mathbf{X}\in\mathbb{R}^{p\times d}.$$

Ainsi en effectuant la SVD tronquée de $\mathbf{Q}$, on aura

$$\texttt{SVD}(\mathbf{Q})=\mathbf{U_k}\mathbf{S_k}\mathbf{V_k^T}, \tag{2.14}$$

avec $\mathbf{U_k}\in\mathbb{R}^{\mathbf{p}\times\mathbf{k}}$, $\mathbf{S_k}\in\mathbb{R}^{\mathbf{k}\times\mathbf{k}}$ et $\mathbf{V_k}\in\mathbb{R}^{\mathbf{d}\times\mathbf{k}}$. La matrice $\widehat{\mathbf{X}}=\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\in\mathbb{R}^{N\times d}$ représente la meilleure approximation de $\mathbf{X}$ et $\widehat{\mathbf{X}}_k=\mathbf{X}\mathbf{V}_k\in\mathbb{R}^{N\times k}$ approxime $\mathbf{X}_k$ au sens de la norme de Frobenius. Pour calculer la forme réduite $\mathbf{X}_k$ d'une matrice de donnée $\mathbf{X}$ de grande dimension, il est plus économique de passer par le calcul de la matrice $\mathbf{V}_k^T$ qui demande une décomposition spectrale d'une matrice de taille inférieure pour obtenir $\widehat{\mathbf{X}}_k$ plutôt que de faire directement la SVD tronquée de la matrice de départ $\mathbf{X}$.

2.5 Apprentissage supervisé pour les données en grande dimension

Dans la littérature, diverses méthodes ont été développées dans le but de rendre facilement accessible l'apprentissage des grandes matrices de données. Pour ce faire, un des principes se base principalement en deux étapes : réduire la dimension et ensuite appliquer une méthode bien définie en apprentissage automatique. Parmi les travaux existants, nous pouvons citer la méthode présentée dans [38] par Bouveyron et al. où les auteurs présentent une méthode paramétrique d'un modèle de mixture Gaussien pour la classification des données en grande dimension. On peut également citer la méthode présentée par Yang et al. dans [39], où les auteurs proposent d'effectuer l'analyse linéaire discriminante (LDA) après avoir fait une première transformation par l'analyse en composante principale (ACP). Les travaux de Liu et al. dans [18] présentent une autre version de traitement des données en utilisant la réduction de dimension par projection aléatoire afin d'effectuer la méthode classique de LDA. Aussi Cai et al. dans [40] ont réalisé un modèle d'analyse spectrale discriminante régressive dans le but de maintenir une bonne classification des grandes bases de données. Ye et al. proposent dans [41] une méthode approchée de LDA en utilisant une décomposition orthogonale de type **QR** de la matrice de dispersion intra-classe. Pratiquement, toutes ces méthodes cherchent à trouver un compromis entre le temps de réalisation de l'algorithme et la performance de classification. Elles se différencient techniquement et pratiquement par leur aptitude à minimiser l'erreur de classification en utilisant un temps de calcul le plus petit possible. Nous présenterons certaines de ces méthodes dans la section 4.2 du chapitre 4.

2.6 Apprentissage partagé

Lorsqu'un modèle statistique est conçu dans un but de prédiction, une hypothèse majeure est l'absence d'évolution du phénomène modélisé entre les étapes d'apprentissage et de prédiction. Ainsi, les données d'apprentissage source et les données cibles (futures) doivent être dans le même espace caractéristique et doivent avoir la même distribution. Malheureusement, cette hypothèse peut-être fausse dans les applications du monde réel. Par exemple, les motivations industrielles pourraient conduire à classer les échantillons d'une marque donnée lorsque seuls les échantillons d'une autre marque sont disponibles pour l'apprentissage. Un modèle prédictif utilisé sur les données n'ayant pas exactement la même répartition que les données d'apprentissage utilisées pour estimer ce modèle va évidemment conduire à des résultats de prédiction dégradés. Dans ce cas, l'apprentissage par transfert est particulièrement utile lorsque nous avons des données étiquetées disponibles dans le domaine source, ce qui nécessite une exploitation des connaissances présentes dans d'autres domaines différents (cibles) mais relatifs au domaine source pour améliorer la performance de prédiction des données cibles.

L'apprentissage partagé ou apprentissage par transfert de connaissance (transfer learning) est défini comme étant l'aptitude d'un système à reconnaître et à appliquer des

connaissances et des compétences, apprises à partir de tâches antérieures, sur de nouvelles tâches ou domaines partageant quelques similitudes. La problématique qui se pose autour de l'apprentissage partagé est de pouvoir identifier ces similitudes entre une ou plusieurs tâche(s) cible(s) (target) et une ou plusieurs tâche(s) source(s), ensuite transférer la connaissance acquise de la ou des tâche(s) source(s) vers la ou les tâche(s) cible(s). La technique d'apprentissage par transfert vise à améliorer les performances d'apprentissage dans un domaine cible à l'aide de connaissances extraites des domaines sources ou des tâches connexes. Ce qui distingue l'apprentissage par transfert de l'apprentissage traditionnel, c'est que les domaines source et cible, ou les tâches cible et source, ou les deux, sont différents.

Différentes techniques ont été développées pour transférer un modèle statistique estimé d'une population source à une population cible. Trois principales tâches d'apprentissage statistique sont communément considérées : la classification probabiliste (paramétrique et semi-paramétrique), la régression linéaire et le regroupement. Dans chaque situation, le transfert de connaissance s'effectue par l'introduction des liens visant à renforcer les similitudes entre les deux domaines et à minimiser les différences. Dans de nombreuses applications typiques d'apprentissage partagé, le problème d'adaptation entre les domaines est le plus utilisé. Il peut être étudié dans des tâches semi-supervisées et non supervisées. La technique d'adaptation entre les domaines est bien adaptée à de nombreuses applications du monde réel, car dans de nombreux cas, les données de test ne contiennent pas d'échantillons étiquetés. Dans ce cas, c'est la méthode d'adaptation non supervisée qui est développée car elle ne nécessite aucune information d'étiquetage du domaine cible. Pour la suite de cette section, nous présentons le principe de la méthode d'apprentissage partagé ainsi que les travaux existants dans la littérature.

2.6.1 Principes

En apprentissage partagé, on parle généralement de domaine "source", $\mathcal{D}_{\mathcal{S}}$ et "cible", $\mathcal{D}_{\mathcal{T}}$. Un domaine est constitué de deux composantes $\mathcal{D} = \{\mathcal{X}, \mathbb{P}(\mathbf{X})\}$ avec $\mathcal{X}$ l'espace caractéristique de $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}, \mathbf{x}_i \in \mathcal{X}$ et $\mathbb{P}(\mathbf{X})$ la distribution de $\mathbf{X}$. En général, le domaine "source" contient des données étiquetées et domaine "cible" contient les données non étiquetées. Le terme "étiquette" utilisé ici fait référence à la connaissance a priori pour qu'un ensemble de données appartienne ou non à un quelconque groupe $y \in \mathcal{Y}$ (ou catégorie). Par la suite nous adaptons la notation de la matrice de données par $\mathbf{X}_{S,T}$ et l'espace des variables par $\mathcal{X}_{S,T}$ avec $\mathbf{S}$ et $\mathbf{T}$ qui sont respectivement les indices des domaines source et cible.

Le transfert partagé donne la possibilité d'apprendre une ou plusieurs tâche(s) bien définie(s) dans $\mathcal{D}_{\mathcal{S}}$ pour pouvoir prédire le comportement des données d'ensemble $\mathcal{D}_{\mathcal{T}}$. C'est à dire à partir des données sources $\mathbf{x}_i \in \mathcal{X}_{\mathcal{S}}$ et des étiquettes $y_i \in \mathcal{Y}_{\mathcal{S}}$, on construit une fonction *f(.)* qui sera utilisée pour la prédiction de y d'un nouveau échantillon $\mathbf{x}$ de $\mathbf{X_T}$.

Considérons deux ensembles d'apprentissage source $\mathcal{S}$ et cible $\mathcal{T}$ issus de deux différents espaces caractéristiques mais de même nature (classe d'appartenance ou catégorie). Le problème repose sur la construction d'une fonction de prédiction *f(.)* qui va nous permettre

de mieux classer l'ensemble $\mathcal{T}$ en utilisant seulement l'information disponibles sur $\mathcal{S}$. Plus précisément, si on suppose que le domaine $\mathcal{D}_{\mathcal{S}} = \{\mathcal{S}, P(\mathbf{X}_S)\}$ avec $\mathbf{X}_S = \{\mathbf{x}_1, \cdots, \mathbf{x}_{n_S}\}$ la matrice contenant les n_S observations de l'ensemble source de dimension d_S et le domaine $\mathcal{D}_{\mathcal{T}} = \{\mathcal{T}, \mathbb{P}(\mathbf{X}_T)\}$ où $\mathbf{X}_T = \{\mathbf{x}_{S+1}, \cdots, \mathbf{x}_{n_S+n_T}\}$ la matrice contenant les n_T observations de l'ensemble target de dimension d_T. Formellement, ces termes couramment utilisés en apprentissage partagé sont donnés par les définitions suivantes :

Définition 1 (Domaine) *Un domaine d'apprentissage $\mathcal{D} = \{\mathcal{X}, \mathbb{P}(X)\}$ est un ensemble constitué deux composantes : un espace d'attributs (feature space), $\mathcal{X}$, et une distribution de probabilité marginale, $\mathbb{P}(X)$, avec $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \in \mathcal{X}$. En général, si deux domaines sont différents, ils peuvent avoir soit différents espaces d'attributs et/ou différentes distributions de probabilité marginales.*

Définition 2 (Tâche) *Une tâche $\mathcal{T} = \{\mathcal{Y}, f(.)\}$ est composée de deux variantes : un espace d'étiquettes $\mathcal{Y}$ et une fonction objective de prédiction, $\mathrm{f}(.) = \mathbb{P}(X|Y)$, qui peut être appréhendée à partir des données d'apprentissage de l'ensemble source qui sont constituées de paires $\{\mathbf{x}_i, y_i\}$, où $\mathbf{x}_i \in \mathbf{X}$ est l'échantillon d'observation et $y_i \in \mathcal{Y}$ est l'étiquette de classe d'appartenance de $\mathbf{x}_i$. La fonction f(.) est utilisée pour la prédiction des étiquettes.*

Définition 3 (Apprentissage par transfert (transfer learning)) *Soit $\mathcal{D}_S = \{\mathcal{X}_S, \mathbb{P}(\mathbf{X}_S)\}$ et $\mathcal{T}_S$, un domaine source et la tâche d'apprentissage associée. Soit également $\mathcal{D}_T = \{\mathcal{X}_T, \mathbb{P}(\mathbf{X}_T)\}$ et $\mathcal{T}_t$, un domaine cible et la tâche d'apprentissage correspondante. Si l'on note l'ensemble des échantillons sources par $\mathbf{X}_S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_S} \sim \mathbb{P}(\mathbf{X}_S)$ et l'ensemble cible par $\mathbf{X}_T = \{\mathbf{x}_i\}_{i=1}^{n_T} \sim \mathbb{P}(\mathbf{X}_T)$; le transfer learning permet une meilleure appréhension de la fonction de prédiction $f_{\mathbb{P}_t}(.) : \mathcal{X} \rightarrow \mathcal{Y}$ en utilisant la connaissance disponible dans $\mathcal{D}_S$ et $\mathcal{T}_S$ sachant que $\mathcal{D}_S \neq \mathcal{D}_T$ ou $\mathcal{T}_S \neq \mathcal{T}_T$.*

2.6.2 Synthèse de méthodes existantes

Conformément à la définition formelle du transfer learning, différents travaux de recherche existent dans la littérature, en fonction des différentes situations possibles qui peuvent se présenter entre le domaine source et le domaine cible [42]. Plusieurs articles de synthèse sur l'apprentissage partagé et sur l'adaptation entre les domaines peuvent être trouvés dans la littérature. Nous pouvons citer entre autres les catégories du transfert partagé les plus répandus. On parle généralement de :

Transfer learning inductif [43, 44] : lorsque les tâches correspondantes du domaine source et du domaine cible sont différentes, $\mathcal{T}_S \neq \mathcal{T}_T$. Les domaines $\mathcal{D}_S$ et $\mathcal{D}_T$ peuvent être identiques ou non, et il est nécessaire dans ce cas d'avoir certaines données étiquetées disponibles dans le domaine cible pour induire un modèle de prédiction objectif. Deux cas peuvent se présenter dans cette situation selon la présence ou non des données étiquetées dans le domaine source. Dans le premier cas on fait référence à des techniques bien connues en *multi-task learning* [45, 46]. Le deuxième cas est une technique connue sous le nom de *selft-taught learning* [43, 44].

Transfer learning transductif [47, 48, 49, 50, 51] : lorsque les tâches correspondantes sont équivalentes, $\mathcal{T}_S = \mathcal{T}_T$, mais les distributions des domaines diffèrent $\mathbb{P}(\mathbf{X}_S) \neq \mathbb{P}(\mathbf{X}_T)$. Dans ce contexte, aucune donnée étiquetée n'est disponible dans le domaine cible, alors que les données disponibles dans le domaine source sont toutes étiquetées. Deux cas sont identifiables dans cette situation selon l'égalité ou non de l'espace des attributs, c'est à dire, (1) $\mathcal{X}_S = \mathcal{X}_T$, ou (2) $\mathcal{X}_S \neq \mathcal{X}_T$. Dans le premier cas, il s'agit de l'adaptation entre les domaines et le second est relatif aux techniques connues sous le non de *covariate shift* [52] ou *sample selection bias* [53, 54].

Transfer learning non supervisé [55, 56, 57, 58, 59] : dans ce cas, il est question de la résolution des problèmes d'apprentissage non supervisés dans le domaine cible $\mathcal{D}_T$ sachant que les tâches correspondantes sont différentes $\mathcal{T}_S \neq \mathcal{T}_T$. Les tâches qui sont associées le plus souvent sont relatives à l'apprentissage non supervisé telles que la réduction de la dimension, l'estimation de la densité, le clustering. Aucune donnée étiquetée n'est disponible dans ce contexte.

Transfer learning hétérogène : lorsque les données source et cible sont représentées par différents espaces de variables ($\mathcal{X}_S \neq \mathcal{X}_T$). Pour résoudre ce problème, la plupart des méthodes d'adaptation entre les domaines visent à apprendre une représentation de caractéristiques communes de sorte que les données de domaine source et cible peuvent être représentées par des caractéristiques homogènes. Formellement, on peut apprendre deux matrices de projection W_S et W_T pour transformer les données source, $\mathbf{X}_S$, et les données de domaine cible, $\mathbf{X}_T$, dans un nouveau sous-espace tel que la différence entre $\mathbf{X}_S W_S$ et $\mathbf{X}_T W_T$ soit réduite [60, 61, 62, 63], [64]. Alternativement, on peut également apprendre une transformation asymétrique G pour représenter les données d'un domaine vers l'autre de sorte que la différence entre $G\mathbf{X}_S$ et $\mathbf{X}_T$ soit minimisée ou l'alignement entre $G\mathbf{X}_S$ et $\mathbf{X}_T$ puisse être maximisé [65, 66].

Pour rapprocher les distributions respectives des domaines source et cible, plusieurs méthodes dans la littérature proposent de créer des représentations intermédiaires des données comme [56, 67, 68, 69], par exemple. Toutefois, ces représentations n'essayent pas explicitement de faire correspondre les distributions de probabilité de source et cible, ce qui peut les rendre sous-optimales pour la classification. La méthode par sélection de l'échantillon pour le transfert proposée par Li et al, dans [47], ou la pondération, l'approche de Xia et al, dans [70] tentent explicitement de faire correspondre les distributions de la source et la cible en trouvant les échantillons source les plus appropriés pour le partage de l'information. Pan et al dans [59] et [71] ont proposé une méthode qui consiste à projeter les données des deux domaines dans un espace de Hilbert à noyau reproduisant (RKHS) pour préserver certaines propriétés sur les distributions spécifiques de chaque domaine.

Certaines méthodes se basent sur la décomposition spectrale d'un noyau préalablement défini pour trouver un espace de projection qui permet de rapprocher la distribution marginale des deux domaines. Cette décomposition est coûteuse lorsque l'on dispose d'un nombre assez conséquent d'échantillons d'apprentissage (cas de matrice dense). Pour éviter les techniques basées sur la définition d'un noyau, Fernando et al, dans [56] et [69] ont présenté une méthode d'adaptation des domaines. Cette méthode permet d'utiliser l'analyse en compo-

sante principale (ACP) pour sélectionner un sous-espace intermédiaire où les distributions des deux domaines soient alignées. L'idée est d'appliquer l'ACP sur $\mathbf{X}_S$ et $\mathbf{X}_T$ séparément en choisissant un espace commun de dimension égale à d inférieur à l'espace initial. Deux matrices de projection G_S et G_T sont ainsi obtenues. Il tente ensuite d'aligner l'ensemble de données source projeté avec l'ensemble de données cible projeté dans un sous-espace commun à l'aide d'une matrice d'alignement de sous-espace $G_a = G_S G_S^T G_T$. Une fois que la source et la cible sont alignées, un classifieur est construit sur l'ensemble de données source transformé $\mathbf{X}_S G_a$ et ensuite appliqué à l'ensemble de données cible transformé $\mathbf{X}_T G_T$.

Pour résoudre ce problème, la plupart des méthodes existantes dans le cas hétérogènes visent à construire une représentation de fonctionnalité commune de sorte que les données de domaine source et cible peuvent être représentées par le même espace de dimensions (ou variables) homogènes. D'une manière formelle, on peut construire deux matrices de projection P et Q pour transformer les données de domaine source $\mathbf{X}_S$ et les données de domaine cible $\mathbf{X}_T$ dans un nouvel espace de faible dimension de sorte que la différence entre les données transformées $\mathbf{X}_S P$ et $\mathbf{X}_T Q$ soit réduite [62].

D'autres techniques appelées transfert par représentation à faible de rang d'une matrice (Low rank representation ou Sparse Coding) [51, 72, 47, 73] consistent à retrouver la structure la plus représentative d'une matrice pour éliminer les observations corrompues par le bruit de mesure pour considérer que les valeurs supposées non corrompues. Un codage de la matrice de données $\mathbf{X}$, par $\mathbf{X} = \mathbf{BS}$ permet d'apprendre une matrice $\mathbf{B}$ dont les vecteurs sont des bases qui représentent un dictionnaire commun entre les domaines et un codage sparse $\mathbf{S}$ dont les valeurs représentent les valeurs de $\mathbf{x}_i$ vraisemblablement non corrompues. Lorsque les données source et cible sont échantillonnées à partir de distributions différentes, elles peuvent être quantifiées et codées avec des représentations différentes, ce qui peut dégrader les performances de classification. Les méthodes de transfert dans ce cas visent à minimiser la divergence de distribution entre les données étiquetées et non et à incorporer ce critère dans la fonction objective du codage pour rendre les nouvelles représentations robustes à la différence de distribution des deux domaines.

Chaque méthode a ses avantages et inconvénients. Bien que ces méthodes donnent des bons résultats, elles présentent certaines limites. Bon nombre de ces méthodes utilisent des paramètres de régularisation dans leur fonction objective ou posent certaines contraintes dans le processus de minimisation de la divergence. L'optimisation de ces paramètres est généralement la tâche la plus difficile dans certaines situations. Aussi, dans le cas du transfert hétérogène, la matrice G est asymétrique, $G \in \mathcal{X}_S \times \mathcal{X}_T$, et sa taille dépend des dimensions des domaines source et cible. Dans ce cas, le coût de calcul pour estimer G peut-être extrêmement élevé, en particulier pour les données en grande dimension. La méthode proposée par Fernando et al. dans [56], utilise l'ACP pour la sélection des variables. Ce qui conduit au même problème en coût de calcul pour les grandes dimensions de données, dès lors que l'étape de l'ACP utilise la décomposition spectrale d'une grande matrice de covariance. Pour résoudre le problème de calcul, Duan et al. [60] et Kulis et al. [66] ont proposé des méthodes à noyau d'apprentissage pour obtenir un espace des fonctionnalités qui rapproche les domaines source et cible. Cependant, ces méthodes à noyau, tout comme [48, 55, 57, 59, 71]

souffrent aussi d'un coût de calcul élevé lorsque nous avons à disposition un grand volume de données (matrices denses). D'autres méthodes construisent des matrices de projection assez denses, ce qui conduit à imposer beaucoup de contraintes ou avoir besoin plus d'informations dans les fonctions d'optimisation [74, 75, 76]. Pour la classification, certaines méthodes adoptent une stratégie qui consiste à apprendre plusieurs classifieurs binaires de manière indépendante pour résoudre le problème multi-classe [77, 78]. De cette façon, la structure sous-jacente parmi les classes multiples ne parvient pas à être pleinement explorée. Par conséquent, ce régime peut limiter la capacité du transfert de connaissances dans le cas d'une classification multi-classes [79].

Dans ce travail de thèse, nous nous plaçons dans un contexte de grande dimension pour l'apprentissage partagé. Face à des bases des données issues de différentes sources, l'objectif est de pouvoir exploiter les connaissances d'un domaine cible pour aider à construire un modèle prédictif dans le domaine source en vue de mieux prédire les données cibles. La technique d'adaptation des domaines que nous avons utilisé dans ce contexte est de chercher dans un premier temps un espace de projection dans lequel $\mathbf{X}_S$ et $\mathbf{X}_T$ partagent le même espace caractéristique et ensuite minimiser la divergence entre les distributions $\mathbb{P}(\mathbf{X}_S)$ et $\mathbb{P}(\mathbf{X}_T)$. Le but est de pouvoir améliorer la complexité de calcul des méthodes existantes tout en limitant la perte d'information et présenter des bonnes performances de classification. Nous nous intéressons à l'approche d'adaptation entre les domaines, où l'on estime qu'un modèle de prédiction construit sur les connaissances de deux domaines améliore considérablement les nouveaux échantillons issus du domaine cible. Des bases de données textes et images ont été utilisées pour la validation du modèle.

2.7 Conclusion

Ce chapitre a d'abord présenté les méthodes générales d'apprentissage supervisé et non supervisé avant de présenter le contexte des réduction de dimension des données en grandes dimensions avant de présenter le principe de l'apprentissage partagé et des techniques déjà exixtantes dans ce domaine. Toutes ces méthodes bien connues dans la littérature sont généralement d'une facilité d'application dans le cas où les données sont de faibles dimensions. Comme nous l'avons introduit, ce travail de thèse vise à améliorer la complexité des méthodes existantes sur l'apprentissage des données en grandes dimension. Pour cela, nous avons considéré principalement deux cas d'études qui sont l'analyse en composante principale et l'analyse discriminante linéaire. Ces deux méthodes font partie des plus utilisées en apprentissage automatique des données pour la réduction de la dimension. Cependant leur application est très limitée lorsque les données sont de taille élevée. Le premier cas présenté dans la suite de ce document, est l'utilisation de l'analyse en composante principale afin de proposer une technique non supervisée lorsque la taille des échantillons et la dimension de l'observation sont conjointement assez larges. Cette technique non supervisée est principalement basée sur l'utilisation des outils des matrices aléatoires. Le but est de développer une nouvelle estimation de la matrice de covariance pour calculer un nouvel espace de représentation des données. Ensuite nous avons traité le cas d'apprentissage supervisé pour étudier

la méthode d'analyse discriminante linéaire dans le cas des grandes dimensions. Enfin la nouvelle méthode obtenue est adaptée dans le contexte de l'apprentissage partagé dans le cadre de l'adaptation entre les domaines.

Chapitre 3

Estimation de la covariance pour les données en grande dimension

3.1 Introduction

Les données en grande dimension apparaissent dans de nombreux domaines, et leur analyse devient de plus en plus importante dans l'apprentissage statistique. Cependant, lorsque la dimension de données tend vers l'infini, un manque d'efficacité a été observé depuis longtemps sur les méthodes statistiques bien connues dans l'analyse multivariée. Un exemple marquant a été introduit par Dempster dans [80], où il a établi l'inefficacité du test T^2 d'Hotelling. Il a soulevé la question des limites de la théorie statistique multivariée traditionnelle lorsque la dimension des données est trop importante, en proposant un *test non-exact*. Les résultats montrent que les méthodes classiques ne tiennent plus dans le cas des données en grande dimension. Bai et Saranadasa [81] ont appuyé sa théorie en montrant que les approches traditionnelles, bien qu'elles peuvent être appliquées dans certaines conditions, restent tout de même moins performantes que le test non-exact. Dès lors, cette problématique a attiré une attention particulière conduisant au développement des statistiques asymptotiques. Dans ce nouveau contexte, la dimension des données d n'est désormais plus considérée fixe, mais peut tendre vers l'infini au même ordre que la taille de l'échantillon N. Des efforts ont été réalisés visant à résoudre ces problèmes en proposant de nouvelles approches statistiques pour apporter des corrections systématiques aux méthodes classiques afin que l'effet de grande dimension soit surmonté. Beaucoup de travaux réalisés appréhendent cette problématique grâce à des outils puissants et asymptotiques empruntés à la théorie des matrices aléatoires [82, 83, 84].

Les sources historiques de la théorie des matrices aléatoires sont multiples. Parmi les plus marquantes, on retient notamment les travaux du statisticien John Wishart dans les années $1920 - 1930$ [85, 86]. Considérons un vecteur $\mathbf{x}$ de taille $1 \times d$, dont les entrées sont des variables aléatoires centrées. Soit $\mathbf{\Sigma} = \mathbb{E}(\mathbf{x}^T\mathbf{x})$ la matrice de variance-covariance du vecteur $\mathbf{x}$. Cette matrice de covariance est une matrice Hermitienne aléatoire semi-définie positive. Considérons N observations indépendantes $\mathbf{x}_1, \ldots, \mathbf{x}_N$ du vecteur $\mathbf{x}$. Si de plus la loi de $\mathbf{x}$ est Gaussienne, alors la matrice, appelée aussi matrice de covariance empirique,

$\hat{\mathbf{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i^T \mathbf{x}_i$ est une matrice de Wishart, suivant une loi de Wishart $\mathcal{W}_d(1, \mathbf{\Sigma})$. Afin de montrer les limites des méthodes statistiques classiques, dont l'ACP, dans le cas multivarié, Johnstone [87] s'est placé dans le cas où $\mathbf{\Sigma} = I_d$, que l'on appelle le cas blanc. Il a montré avec un exemple très simple que les valeurs propres de la matrice de covariance empirique sont beaucoup plus dispersées que celles de la matrice de population, qui sont toutes égales à 1. La théorie des matrices aléatoires va servir de cadre alternatif très nécessaire et bienvenu pour l'étude des propriétés spectrales des matrices de covariances empiriques afin de corriger les estimateurs classiques des valeurs propres et vecteurs propres de la matrice de covariance $\mathbf{\Sigma}$.

Dans ce chapitre, nous proposons une méthode de partitionnement des données via une analyse en composantes principales linéaire. Plus précisément, en se basant sur des outils provenant de la théorie des matrices aléatoires, nous proposons une nouvelle modélisation des composantes principales adaptée au contexte de données de grande dimension pour un partitionnement automatique. Nous notons que la méthode est valable aussi bien dans le cadre de petite dimension. Inspirés des travaux de Xavier Mestre [88], des estimateurs (N, d)-consistants des valeurs propres et vecteurs propres, quand le nombre d'échantillons N et la dimension des données d tendent conjointement vers l'infini, sont proposés.

Le chapitre est organisé comme suit : nous présentons d'abord dans la section 3.2 une brève description des outils théoriques utilisés des matrices aléatoires. Ensuite nous proposons un nouvel algorithme de partitionnement automatique, basé sur des estimateurs consistants des valeurs et vecteurs propres de la matrice de covariance dans la section 3.3. Enfin, nous montrons l'efficacité de notre algorithme par rapport à l'état de l'art, et cela à travers des indicateurs de performances, comme le NMI (Normalized Mutual Information) et le ER (Error Rate) dans la section 3.4.

3.2 Estimation de la covariance basée sur les matrices aléatoires

Les techniques et les résultats des matrices aléatoires ont beaucoup à offrir, notamment en ce qui concerne l'analyse spectrale des matrices symétriques en général, et des matrices de covariance en particulier. Un travail pionnier dans cette direction est celui de Marchenko-Pastur [89], dans lequel est étudié le comportement global des valeurs propres d'une matrice de covariance. Plus précisément, Marchenko et Pastur ont démontré la convergence vers une loi déterministe de la distribution spectrale des valeurs propres d'une matrice de Wishart dans le cas blanc où $\mathbf{\Sigma} = \sigma^2 I_d$, avec $\sigma^2 \in \mathbb{R}_+^*$, et ce quand la dimension d augmente au même rythme que le nombre d'observations N. Nous donnons un bref aperçu sur cette loi dans la sous-section suivante. Après le comportement global, certains travaux se sont concentrés sur l'étude de la plus petite/grande valeur propre dont entre autres les travaux de Geman [90] et Silverstein [91] afin de compléter le travail de Marchenko-Pastur. Depuis lors, la question de l'analyse spectrale des grandes matrices de covariance a intéressé beaucoup chercheurs [92, 93, 94].

3.2.1 Outils des matrices aléatoires

Comme en général la matrice de covariance de population $\mathbf{\Sigma}$ est inconnue, les méthodes se basent sur des observations du vecteur $\mathbf{x}$ afin de donner une estimation empirique $\hat{\mathbf{\Sigma}}$ à la matrice $\mathbf{\Sigma}$. Les valeurs propres d'une matrice sont des fonctions continues d'entrées de la matrice. Mais ces fonctions n'ont pas d'expressions explicites faciles à manipuler, notamment lorsque la dimension de la matrice est grande. Des méthodes spéciales sont nécessaires pour leur étude. Il existe diverses méthodes importantes dans cette thématique à savoir la méthode des moments, la transformation de Stieltjes et la décomposition polynomiale orthogonale de la densité exacte des valeurs propres.

Nous nous concentrons sur l'approche de la transformation de Stieltjes et nous invitons les lecteurs intéressés à consulter la référence [82] pour une description détaillée des autres méthodes. La transformée de Stieltjes constitue un outil de base dans l'étude des valeurs propres des matrices aléatoires puisqu'elle permet d'étudier la convergence en loi de la distribution spectrale. Nous donnons une description de cette méthode dans la première partie de cette section, suivie d'une présentation de la loi de Marchenko-Pastur.

Transformée de Stieltjes

Soit μ une loi quelconque sur $\mathbb{R}$. La transformée de Stieltjes de la mesure μ est une application de $\mathbb{C}_+ = \{x + iy \in \mathbb{C} \backslash x \in \mathbb{R} \text{ et } y \in \mathbb{R}_+^*\}$ à valeurs dans $\mathbb{C}$ définie par :

$$S_\mu(z) = \int_{\mathbb{R}} \frac{1}{\lambda - z} d\mu(\lambda). \tag{3.1}$$

La transformée de Stieltjes S_μ caractérise la loi μ, dans le même sens qu'une transformée de Fourier F_μ caractérise la loi μ, mais à l'aide de fonction test différente ($z \mapsto e^{-z\lambda}$ pour la transformée de Fourier et $z \mapsto \frac{1}{\lambda - z}$ pour la transformée de Stieltjes). Afin d'étudier le comportement global des valeurs propres ainsi que leur répartition, on introduit la mesure spectrale empirique associée à une matrice symétrique A, de dimension d et de valeurs propres $\lambda_1, \ldots, \lambda_d$:

$$\mu_A(d\lambda) = \frac{1}{d} \sum_{i=1}^{d} \delta_{\lambda_k}(d\lambda),$$

δ_x étant la mesure de Dirac. Autrement, la mesure spectrale de la matrice A est une loi de probabilité discrète, une loi de comptage normalisée par $\frac{1}{d}$. La transformée de Stieltjes de la mesure spectrale de la matrice A est donc donnée par :

$$S_{\mu_A}(z) = \int_{\mathbb{R}} \frac{1}{\lambda - z} d\mu(\lambda) = \frac{1}{d} \sum_{i=1}^{d} \frac{1}{\lambda_i - z}.$$

Considérons la matrice fonction du complexe z suivante : $R(z) = (A - zI_d)^{-1}$, appelée matrice résolvante associée à la matrice A. Nous pouvons remarquer que la transformée de Stieltjes de A n'est autre que la trace normalisée de la matrice résolvante de A. Autrement,

$$S_{\mu_A}(z) = \frac{1}{d} tr\left(A - zI_d\right)^{-1}. \tag{3.2}$$

Par conséquent, la question de l'étude de la convergence de la transformée de Stieltjes de μ_A revient à étudier le comportement asymptotique, quand d tend vers l'infini, de la trace normalisée de la résolvante de A. Ce lien très directe entre la transformée de Stieltjes de la mesure spectrale et la matrice résolvante a été remarqué et utilisé par Marchenko et Pastur afin de caractériser le comportement asymptotique global des valeurs propres d'une matrice de covariance empirique.

La loi de Marchenko-Pastur [89]

Rappelons le modèle de Wishart : considérons un vecteur aléatoire $\mathbf{x}$ suivant une loi gaussienne centrée $\mathcal{N}(0, \mathbf{\Sigma})$. Soient un ensemble d'observations $\mathbf{x}_i, i = 1, \ldots, N$, indépendantes et identiquement distribuées générées suivant la même loi que $\mathbf{x}$. On range habituellement les $\mathbf{x}_i$ dans une matrice $\mathbf{X} \in \mathbb{R}^{N \times d}$, où $\mathbf{x}_i$ représente la $i^{ième}$ ligne de $\mathbf{X}$. Soient $\mathbf{\Sigma}$ et $\hat{\mathbf{\Sigma}}$ qui désignent la vraie covariance de la population et la covariance empirique, respectivement. On s'intéresse précisément au comportement de la distribution empirique des valeurs spectrales de ces deux matrices telles que :

$$\mathbf{\Sigma} = \mathbb{E}(\mathbf{x}^T\mathbf{x}), \quad \hat{\mathbf{\Sigma}} = \frac{1}{N}\mathbf{X}^T\mathbf{X} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i^T\mathbf{x}_i, \ \mathbf{\Sigma} \text{ et } \hat{\mathbf{\Sigma}} \in \mathbb{S}_+^d, \tag{3.3}$$

où S_+^d est l'ensemble des matrices symétriques semi-définies positives. Considérons les décompositions spectrales respectives de $\mathbf{\Sigma}$ et $\hat{\mathbf{\Sigma}}$:

$$\mathbf{\Sigma} = \sum_{i=1}^{d}\lambda_i u_i u_i^T \text{ et } \hat{\mathbf{\Sigma}} = \sum_{i=1}^{d}\hat{\lambda}_i \hat{u}_i \hat{u}_i^T.$$

Marchenko et Pastur [89] se sont penchés sur l'étude de la mesure spectrale de la matrice de covariance empirique $\hat{\mathbf{\Sigma}}$:

$$\mu_{\hat{\mathbf{\Sigma}}}(d\lambda) = \frac{1}{d}\sum_{l=1}^{d}\delta_{\hat{\lambda}}(d\lambda),$$

sous le régime asymptotique où la taille de l'échantillon N et la dimension des données d tendent conjointement vers l'infini (tels que : $y = \lim_{N\to+\infty}\frac{d}{N} \in]0, +\infty[$). Le modèle étudié est celui de Wishart blanc, où $\mathbf{\Sigma} = \sigma^2 I_d$. Le théorème suivant décrit le résultat de cette étude.

Théorème 3 *: Soit* $\mathbf{X}$ *une matrice aléatoire de taille* $N \times d$ *dont les lignes sont des vecteurs aléatoires indépendants et identiquement distribués suivant une loi* $\mathcal{N}(0, \sigma^2 I_d)$*. Supposons que* $y = \lim_{N\to+\infty}\frac{d}{N}$ *est un réel positif. Alors, la suite des mesures spectrales* $(\mu_{\hat{\Sigma}})_N$ *converge presque sûrement, quand* N *tend vers l'infini, vers la loi de Marchenko-Pastur, dont la fonction de densité est donnée par :*

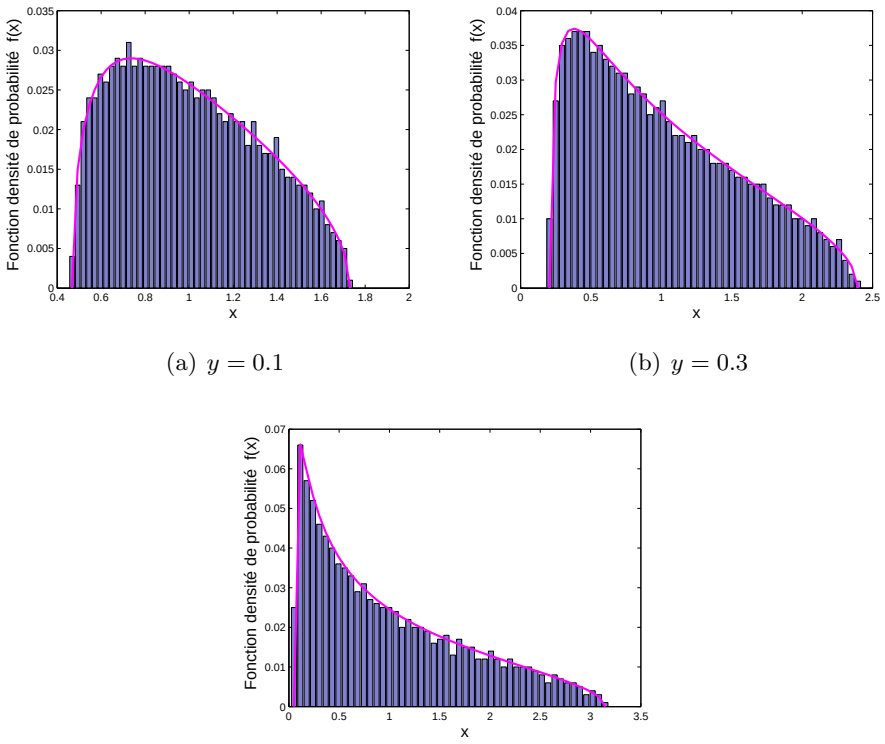
$$f_{MP}(x) = \frac{1}{2y\pi\sigma^2 x}\sqrt{(b-x)(x-a)}\ \mathbb{I}_{[a,b]}(x) + (1-y^{-1})\delta_0(x)\mathbb{I}_{y\in[1,+\infty[}\ ,$$

avec $a = \sigma^2(1-\sqrt{y})^2$ *et* $b = \sigma^2(1+\sqrt{y})^2$.

Nous présentons dans la figure 3.1 une représentation graphique de la densité de Marchenko-Pastur et de l'histogramme empirique des valeurs propres de la matrice $\hat{\mathbf{\Sigma}}$ pour différents scénario de y. Plus précisément, nous considérons N=1000 échantillons suivant la loi normale (moyenne nulle, écart-type égal 1) et trois valeurs de y, à savoir 0,1 , 0,3 et 0,6. On remarque que, à l'intérieur de l'intervalle limite $[a, b]$ qui représente le support limite des valeurs propres $\hat{\lambda}_i$, la courbe de la densité limite f épouse bien l'histogramme représentant la distribution des valeurs propres empiriques $\hat{\lambda}_i$.

Conformément à l'asymptotique classique de grands échantillons (en supposant que N = 1000 est assez large), la matrice de covariance d'échantillon $\hat{\mathbf{\Sigma}}$ devrait être proche de la matrice de covariance de la population $\mathbf{\Sigma} = I_d = \mathbb{E}(\mathbf{x}^T\mathbf{x})$. Comme les valeurs propres sont des fonctions continues des entrées matricielles, les valeurs propres des échantillons de $\hat{\mathbf{\Sigma}}$ doivent converger en 1 (valeur propre unique de I_p).

Étant donné que les valeurs propres de l'échantillon s'éloignent des valeurs propres de la population, la matrice de covariance de l'échantillon $\hat{\mathbf{\Sigma}}$ n'est plus un estimateur fiable de son homologue $\mathbf{\Sigma}$. Cette observation est en effet la raison fondamentale pour que les méthodes multivariées classiques se décomposent lorsque la dimension des données est comparable à la taille de l'échantillon.



(a) $y = 0.1$ (b) $y = 0.3$

(c) $y = 0.6$

FIGURE 3.1 – Densité Marchenko-Pastur et l'histogramme des valeurs empiriques pour N=1000 et trois différentes valeurs de $y = \frac{d}{N}$.

3.2.2 Estimation (N, d)-consistante de la matrice de covariance

Comme exposé précédemment, et grâce au travail de Marchenko-Pastur, que les valeurs propres de la matrices empirique $\hat{\mathbf{\Sigma}}$ sont des estimateurs bruités des vraies valeurs propres en grande dimension [87]. Dans le cadre d'une application au traitement statistique du signal numérique, Xavier Mestre a proposé une nouvelle approche pour estimer les valeurs propres et les vecteurs propres de $\mathbf{\Sigma}$ [88]. Mestre propose des formules analytiques basées sur les transformées de Stieltjes (3.2) des mesures spectrales des valeurs propres et vecteurs propres de $\hat{\mathbf{\Sigma}}$.

Description de la méthode de X. Mestre [88]

Rappelons la décomposition en éléments propres des matrices de covariance :

$$\mathbf{\Sigma} = \sum_{i=1}^{d} \lambda_i u_i u_i^t, \quad \text{et} \quad \hat{\mathbf{\Sigma}} = \sum_{i=1}^{d} \hat{\lambda}_i \hat{u}_i \hat{u}_i^t.$$

Supposons que chaque valeur propre λ_i de la matrice de covariance de population a une multiplicité K_m et que l'on a $\bar{d}$ valeurs propres distinctes. On a donc : $d = \sum_{m=1}^{\bar{d}} K_m$. A chaque valeur propre λ_i on lui associe l'espace propre $\mathbf{E_m}$, de dimension $d \times K_m$, constitué des K_m vecteurs propres associés à λ_i avec $\mathbf{E_m^* E_m} = \mathbf{I_{K_m}}$.

Estimation consistante des valeurs propres

Il est bien connu, que les valeurs propres des échantillons ont tendance à être plus éloignées de vraies valeurs originales. En effet, lorsque les valeurs d'un échantillon sont importantes (ou plus petites), elles tendent à surestimer (ou sous-estimer) les valeurs propres correspondantes de la vraie matrice de covariance. A travers le théorème 4 (théorème 3 dans [88]), nous présentons des estimateurs (N,d)-consistants des valeurs propres et vecteurs propres dès lors que le rapport $y = \frac{d}{N}$ reste asymptotiquement constant.

Estimation consistante des vecteurs propres

La transformation de Stieltjes donnée dans l'équation (3.1) est appropriée pour caractériser le comportement asymptotique des valeurs propres de l'échantillon, mais n'est pas suffisante pour décrire les propriétés asymptotiques des vecteurs propres (parce qu'elle dépend uniquement des valeurs propres de l'échantillon et non des vecteurs propres). Par conséquent, il est nécessaire de considérer une fonction permettant d'exprimer également les vecteurs propres. Mestre [88], propose de considérer l'estimateur des vecteurs propres à travers la fonction

$$\hat{m}_d(z) = \sum_{i=1}^{d} \frac{s_1^* \hat{u}_i \hat{u}_i^* s_2}{\hat{\lambda}_i - z} = s_1^* (\hat{\mathbf{\Sigma}} - z I_d)^{-1} s_2 \ , \tag{3.4}$$

où $\hat{u}_r$ désigne l'estimée de u_r et $s_1,\ s_2$ sont deux vecteurs déterministes de $\mathbb{R}^d$ de normes euclidiennes bornées. Il est facile de constater que l'équation (3.4) depend des valeurs propres

et des vecteurs propres. Afin de se concentrer sur l'estimation des vecteurs propres, Mestre considère le problème alternatif d'estimation consistante des formes bilinéaires de type $s_1^* u_i u_i^* s_2$ où u_i est le i^{ème} vecteur propre de la matrice de covariance de population, et cela sous le régime asymptotique où N et d tendent conjointement à l'infini. Ce qui permet de caractériser le comportement asymptotique des vecteurs propres. Un opérateur de projection est généralement défini sous la forme

$$P_m = \mathbf{E}_m \mathbf{E}_m^* \tag{3.5}$$

où $\mathbf{E}_m \in \mathbb{C}^{d \times K_m}$, est une matrice contenant les vecteurs propres associé à la valeur propre de multiplicité K_m, $\mathbf{E} = [\mathbf{E}_1, \mathbf{E}_2, \cdots, \mathbf{E}_{\bar{d}}] \in \mathbb{C}^{d \times d}, m = 1, \cdots, \bar{d}$, avec $\bar{d}$ qui représente le nombre de valeurs propres distinctes. Contrairement à la matrice $\hat{\mathbf{U}}$ contenant les vecteurs propres $\hat{u}_i$, les entrées de la matrice de projection P_m, sont toujours spécifiées dans (3.4) par les vecteurs s_1 et s_2.

L'idée de Mestre est la suivante : dans le but de proposer des estimateurs (N, d)-consistants des vecteurs propres de $\mathbf{\Sigma}$, et en rappelant que les matrices propres $\mathbf{E}$ sont stables par multiplication à droite par une matrice orthogonale, Mestre a suggéré de travailler directement les matrices de projection de type P_m, qui sont des matrices ayant les mêmes propriétés que les matrices symétriques.

Basé sur des représentations en intégrales de contour relativement facilement estimables, Mestre propose d'estimer de façon consistante des formes quadratiques de la forme :

$$\eta_m = s_1^* \mathbf{E}_m \mathbf{E}_m^* s_2, \quad m = 1, \ldots, \bar{d}$$

où s_1 et s_2 sont deux vecteurs génériques déterministes de taille $d \times 1$ et de norme euclidienne bornée. Une fois l'estimation de ces formes quadratiques réalisée, on retrouvera les estimateurs des entrées de la matrice de projection $\mathbf{E}_i$ en prenant s_1 et s_2 les vecteurs canoniques de la matrices identité $I_{d \times d}$.

L'estimateur proposé $\bar{\eta}_m$ est une somme pondérée des termes $s_1^* \hat{u}_m \hat{u}_m^* s_2$, dont les poids correspondants dépendent de toutes les valeurs propres empiriques $\hat{\lambda}_m$, $m = 1, \ldots, \bar{d}$. Plus formellement, Mestre propose les estimateurs suivants :

$$\bar{\eta}_m = \sum_{k=1}^{d} \theta_{m,k} s_1^* \hat{u}_m \hat{u}_m^* s_2 \ ,$$

où les poids $\theta_{m,k}$ sont définis dans le théorème 4.

Théorème 4 *Considérons le modèle matriciel présenté dans le théorème 1. Des estimateurs $(N, d)-$consistants des valeurs propres λ_m et des vecteurs propres u_m de la matrice de covariance de population $\mathbf{\Sigma}$ sont proposés comme suit : Pour tout $m = 1, \ldots, \bar{d}$ on a les convergences presque sûres suivantes :*

$$\lambda_m - \bar{\lambda}_m \xrightarrow[N \to +\infty, \frac{d}{N} \to y]{p.s.} 0 \qquad et \qquad \eta_m - \bar{\eta}_m \xrightarrow[N \to +\infty, \frac{d}{N} \to y]{p.s.} 0$$

$$\bar{\lambda}_m = \frac{N}{K_m} \sum_{k \in \mathcal{K}_m} \left(\hat{\lambda}_k - \beta_m \right) \qquad et \qquad \bar{\eta}_m = \sum_{k=1}^{d} \theta_m(k) s_1^* \hat{u}_m \hat{u}_m^* s_2$$

avec,

$$\theta_m(k) = \begin{cases} -\phi_m(k) & \text{if } k \neq \mathcal{K}_m \\ 1 + \psi_m(k) & \text{if } k \in \mathcal{K}_m \end{cases}$$

$$\begin{cases} \phi_m(k) = \sum_{t \in \mathcal{K}_m} \left(\frac{\hat{\lambda}_t}{\hat{\lambda}_k - \hat{\lambda}_t} - \frac{\beta_t}{\hat{\lambda}_k - \beta_t} \right) \\ \psi_m(k) = \sum_{t \notin \mathcal{K}_m} \left(\frac{\hat{\lambda}_t}{\hat{\lambda}_k - \hat{\lambda}_t} - \frac{\beta_t}{\hat{\lambda}_k - \beta_t} \right) \end{cases}$$

où $\beta_1 \leq \beta_2 \leq \cdots \leq \beta_d$ *sont les valeurs réelles des solutions de l'équation* $\frac{1}{d}\sum_{k=1}^{d}\left(\frac{\lambda_k}{\lambda_k - \beta}\right) - \frac{n}{d} = 0$ *et* $\mathcal{K}_m = \left\{\sum_{j=1}^{m-1} K_j + 1, \sum_{j=1}^{m-1} K_j + 2, \cdots, \sum_{j=1}^{m} K_j\right\}$ *est un ensemble qui contient les indices de multiplicités.*

L'estimation proposée du sous-espace défini dans $\bar{\eta}_m$ associé à la m^{ème} valeur propre $\bar{\lambda}_m$ dans ce cas utilise tous les vecteurs propres de la matrice de covariance empirique à la différence des estimateurs traditionnels qui utilisent seulement ceux associés à la valeur propre en question. L'estimateur applique un poids différent aux vecteurs propres selon qu'ils sont associés à la même valeur propre ou non. La preuve de ces résultats est basée sur des équations implicites vérifiées par les transformées de Stieltjes des valeurs propres (3.1) et des vecteurs propres (3.4) [88].

3.3 Partitionnement basé sur la covariance empirique

Pour évaluer la performance des estimateurs calculés, nous considérons une approche du partitionnement par l'analyse en composantes principales (ACP). Cette dernière bien connue est strictement basée sur le calcul des valeurs propres et vecteurs propres de la matrice de covariance empirique, pour obtenir l'espace de projection à variance maximum. Cet espace de projection est en réalité l'estimateur traditionnel des vecteurs propres. La méthode que nous présentons ici, se différencie de l'ACP traditionnelle, par le fait qu'elle substitue l'usage des estimateurs traditionnels par ces nouveaux estimateurs. Après la transformation dans le nouvel espace orthogonal, nous effectuons la méthode de partitionnement par la méthode de *K-means* sur les nouveaux échantillons obtenus après transformation.

3.3.1 Analyse en composantes principales : nouvelle alternative

Les résultats présentés dans le théorème 4 concernent le cas où les valeurs propres λ_i sont de multiplicités supérieures à 1. Dans cette application, nous considérons le cas où toutes les valeurs propres sont de multiplicité égale à 1. Notons $\hat{\lambda}_i^{new}$ et $\hat{P}_i^{new}$ les estimateurs $(N,d)-$consistants de λ_i et de l'espace de projection $U_i U_i^*$, respectivement, dans le cas de multiplicité égale à 1 :

$$\widehat{\lambda}_i^{new} = N\left(\widehat{\lambda}_i - \beta_i\right) \quad ; \quad \widehat{P}_i^{new} = \sum_{k=1}^{d} \theta_i(k)\, \widehat{u}_k \widehat{u}_k^T \qquad i = 1, \cdots, d, \tag{3.6}$$

avec,

$$\theta_i(k) = \begin{cases} \frac{\beta_i}{\hat{\lambda}_k - \beta_i} - \frac{\hat{\lambda}_i}{\lambda_k - \lambda_i} & \text{si } k \neq i \\ 1 + \sum_{t=1, t\neq i}^{d} \left(\frac{\hat{\lambda}_t}{\hat{\lambda}_i - \hat{\lambda}_t} - \frac{\beta_t}{\hat{\lambda}_i - \beta_t} \right) & \text{si } k = i \end{cases}$$

où les β_i sont décrits dans le théorème 4.

La nouvelle matrice $\hat{\mathbf{\Sigma}}^{new}$ dont les valeurs propres sont les $\hat{\lambda}_i$ et les espaces de projections propres sont $\hat{P}_i^{new}$ est donc la nouvelle matrice estimateur de $\mathbf{\Sigma}$. Nous procédons maintenant à l'extraction de ses vecteurs propres $\hat{u}_i^{new}$.

On peut écrire que :

$$\widehat{u}_i^{new}\widehat{u}_i^{new^T} - \widehat{P}_i^{new} \underset{n\to\infty}{\longrightarrow} 0,$$

donc,

$$\sum_{i=1}^{d} \widehat{u}_i^{new}\widehat{u}_i^{new^T} - \sum_{i=1}^{d} \widehat{P}_i^{new} \underset{n\to\infty}{\longrightarrow} 0,$$

En posant $P^{new} = \sum_{i=1}^{d} \widehat{u}_i^{new}\widehat{u}_i^{new^T}$, on peut remarquer que la matrice P^{new} peut être considérée comme une union des sous espaces propres calculés à travers l'équation (3.6) telle que :

$$P^{new} = \sum_{i=1}^{d} \widehat{P}_i^{new} = \sum_{i=1}^{d}\sum_{k=1}^{d} \theta_i(k)\, \widehat{u}_k \widehat{u}_k^T \tag{3.7}$$

En applicant la décomposition en valeurs singulières sur la matrice P^{new}, on peut écrire :

$$P^{new} = Q\Gamma R^T \quad avec \quad Q^T Q = I_d. \tag{3.8}$$

Nous pouvons construire l'espace de projection orthogonal à partir des vecteurs colonnes normalisés de $Q = [\widehat{u}_1^{new}, \ldots, \widehat{u}_d^{new}]$. Les valeurs spectrales obtenues, $\hat{\lambda}_i^{new}$ et Q, sont des estimateurs robustes, qui garantissent la robustesse dans un régime évolutif lorsque $(N, d) \to \infty$ au même ordre de grandeur. Ils prennent également en considération les limites des estimateurs traditionnels.

3.3.2 Partitionnement par ACP

Les composantes principales d'un ensemble d'observations suivant une distribution de probabilité $\mathbb{P}$, est un ensemble de vecteurs, qui satisfait :

$$\mathbf{u}^* = \underset{\|\mathbf{u}\|^2=1}{\texttt{argmax}} \ \mathbb{E}((\mathbf{x}\mathbf{u})^2) \tag{3.9}$$

où $\mathbf{u}^*$ est une direction le long de laquelle la variance des données est maximale, comme dans la méthode classique de l'ACP. Par ailleurs, le problème d'optimisation 3.10 est équivalent à :

$$\mathbf{u}^* = \underset{\|\mathbf{u}\|^2=1}{\texttt{argmax}} \ \mathbf{u}^T\mathbf{\Sigma}\mathbf{u}. \tag{3.10}$$

Considérons que nous allons garder que les k vecteurs propres pour l'analyse en composantes principales, et considérons un opérateur $\mathcal{P}$ de projection contenant l'ensemble des k vecteurs $\{\mathbf{u}_s^*\}_{s=1}^k \in \mathbb{R}^d$ et tels que $Q_k = [\hat{\mathbf{u}}_\mathbf{1}^\mathbf{new}, \hat{\mathbf{u}}_\mathbf{2}^\mathbf{new}, \cdots, \hat{\mathbf{u}}_\mathbf{k}^\mathbf{new}] \in \mathcal{P}$, pour chaque vecteur $\mathbf{x} \in$

$\mathbb{R}^d$, on considère son approximation par $\mathbf{x}Q$. L'erreur d'approximation est mesurée par $\|\mathbf{x} - \mathbf{x}QQ^T\|_2^2$ comme dans le cas du problème classique de l'ACP avec la matrice Q qui représente une estimation de $\mathbf{U}$. La matrice Q contient l'ensemble des vecteurs optimums qui maximise la variance des données et la matrice $\mathbf{X}Q_k$ définit les composantes principales qui minimisent l'écart de transformation après la projection.

Nous nous plaçons dans un contexte de classification des données en grande dimension où les échantillons sont issus d'une distribution multivariée avec K différentes classes. L'approche consiste d'abord à calculer la matrice de covariance empirique $\hat{\mathbf{\Sigma}}$. Ensuite, construire les nouveaux estimateurs proposés $\hat{\lambda}_i^{new}$ et la matrice Q décrite précédemment. Enfin, on applique l'algorithme *K-means* pour le partitionnement dans le nouvel espace. En effectuant le partitionnement, on évalue la consistence et la robustesse des estimateurs utilisés sur leur capacité à préserver la structure intrinsèque des données. Les grands traits de l'approche sont détaillés dans l'algorithme 3.1. Il prend en entrées, la matrice de données $\mathbf{X}$ et le nombre de classes K. La sortie de l'algorithme donne le vecteur de prédiction contenant les indices des classes d'appartenance des échantillons.

Algorithme 3.1 Nouvelle approche de l'ACP

Entrées: $\mathbf{X}$, nombre de classes K

Sorties: numéro de groupe de $\mathbf{x}_i$ tel que $\{z_i = s\}_{i=1}^N$, $s = 1, ..., K$

1: Calculer la matrice de covariance $\hat{\mathbf{\Sigma}} \in \mathbb{R}^{d\times d}$;
2: Calculer la decomposition spectrale de $\hat{\mathbf{\Sigma}}$ et obtenir $\widehat{\lambda}_i$ et $\widehat{u}_i$;
3: Calculer $\widehat{\lambda}_i^{new}$ et $\widehat{P}_i^{new}$ définies en (3.6) ;
4: Calculer P^{new} à partir de (3.7) ;
5: Calculer les k vecteurs propres par SVD tronquée de P^{new} pour obtenir Q_k ;
6: Calculer les nouvelles coordonnées des k composantes $\widetilde{\mathbf{X}}_k^{new}$ de $\mathbf{X}$ en projetant $\mathbf{X}$ sur Q_k tel que $\widetilde{\mathbf{X}}_k^{new} = Q_k^T\mathbf{X}$;
7: Appliquer K-means pour partitionner $\widetilde{\mathbf{X}}_k^{new}$ et récupérer les indices $z_i = s$, $s = 1, ..., K$.

3.3.3 Indicateurs de performance

Dans cette partie, nous utilisons le partitionnement par la méthode $Kmeans$ dans une optique de classification non supervisée. Chaque échantillon appartient à un groupe de données. Les étiquettes des échantillons ne sont pas utilisées dans l'apprentissage et sont considérées uniquement pour l'évaluation des performances de l'approche. Les résultats numériques sont évalués à l'aide de deux indicateurs de performance couramment utilisés dans la littérature [95] : l'information mutuelle normalisée (NMI) et le taux d'erreur de classification finale (ER).

L'information mutuelle normalisée (NMI)

Nous avons utilisé l'information mutuelle pour calculer la corrélation entre le vecteur contenant les indices théoriques d'appartenance à un groupe donné et le vecteur d'indices obtenus après l'application de la méthode de partitionnement. En considérant W et Z,

deux variables qui décrivent respectivement les vecteurs d'indices théoriques et pratiques, l'information mutuelle entre ces variables mesure la dépendance mutuelle de la quantité d'information apportée en moyenne par une réalisation de la variable W sur les probabilités de réalisation de Y. La présence ou non d'une information sur un phénomène aléatoire se mesure à travers l'entropie d'une distribution de probabilité réalisée sur la variable. La figure 3.2 donne le diagramme de Venn qui illustre la définition de l'information mutuelle de deux variables par rapport à leurs entropies. En ce sens, l'information mutuelle se définie par :

$$\begin{aligned} I(W,Z) &= \mathcal{H}(W) - \mathcal{H}(W|Z), \\ &= \mathcal{H}(Z) - \mathcal{H}(Z|W), \\ &= \mathcal{H}(W) + \mathcal{H}(Z) - \mathcal{H}(W,Z), \\ &= \mathcal{H}(W,Z) - \mathcal{H}(W|Z) - \mathcal{H}(Z|W) \end{aligned}$$

où $\mathcal{H}(W)$ désigne l'entropie de la variable W donnée par

$$\mathcal{H}(W) = -\sum_{i=1}^{N} \mathbb{P}(W = w_i) \times \log \mathbb{P}\left(W = w_i\right),$$

et l'entropie conjointe de W et Z est donnée par

$$\mathcal{H}(W,Z) = -\sum_{i=j}^{N}\sum_{i=1}^{N} \mathbb{P}(W = w_i, Z = z_j) \times \log \mathbb{P}\left(W = w_i, Z = z_j\right),$$

et l'entropie conditionnelle de W sachant que $Z = z_j$ est donnée par

$$\mathcal{H}(W|Z = z_j) = -\sum_{i=1}^{N} \mathbb{P}(W = w_i|Z = z_j) \times \log \mathbb{P}\left(W = w_i|Z = z_j\right).$$

L'information mutuelle est nulle si et seulement si les deux variables sont indépendantes, et croit lorsque la dépendance augmente. L'information mutuelle varie avec l'évolution du chevauchement entre les variables et l'information mutuelle normalisée (NMI) a été proposée comme une alternative qui prend en considération toutes les variations des variables [96], et est donnée par :

$$\texttt{NMI}(W,Z) = \frac{I(W,Z)}{\sqrt{(H(W) \times H(Z)}}. \tag{3.11}$$

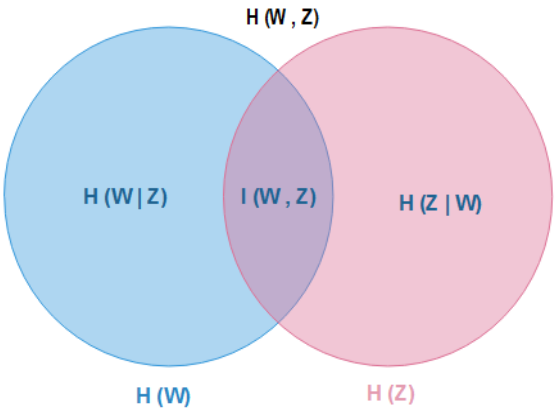


FIGURE 3.2 – Entropie et information mutuelle d'un couple de variables aléatoires (W,Z).

Taux d'erreur (ER (error rate))

Le taux d'erreur permet d'évaluer le taux de mauvais classement lorsque la prédiction ne coïncide pas avec la vraie valeur. Ce taux, rapporté à l'effectif des données, donne le pourcentage de mauvaise classification obtenu entre la répartition idéale (théorique) des données et celle obtenue dans la pratique. Il est exprimé le plus souvent en fonction de la précision de classification ou *purity* [95] et est donné par l'expression suivante :

$$\texttt{ER} = 1 - purity = 1 - \frac{1}{N}\sum_{n}\max_{j}|w_n \cap z_j|. \tag{3.12}$$

où $\{w_n\}_{n=1}^{N}$ représente l'ensemble des indices théoriques des échantillons et $\{z_j\}_{j=1}^{N}$ représente les indices obtenus. Pour améliorer le résultat de calcul, nous avons réordonné les indices obtenus après classification. Ce réarrangement est dû au fait que la fonction $Kmeans$ attribue les indices de groupes de façon aléatoire selon l'ordre de détection des clusters. Par exemple, une classe idéalement numérotée avec l'indice $n°1$, peut se voir attribuer l'indice $n°2$ ou un autre indice (parmi les groupes détectables). Ceci nécessite la réaffectation de chacune des classes avec leur indice correspondant.

3.4 Résultats et discussion

La méthode proposée a été appliquée aux jeux de données synthétiques, construits à partir de variables Gaussiennes. Nous avons généré quatre groupes $\{\Omega_i\}_{i=1}^{4}$ avec différents paramètres (moyenne et écart-type). Chaque Ω_i possède n_i échantillons ($N = \sum_i n_i$) observé sur d variables. Nous avons choisi les paramètres de la moyenne et de la variance de $\Omega_1, \Omega_2, \Omega_3, \Omega_4$ tels que $m_i = \mathbb{1}\sqrt{d}m_{i0} \in \mathbb{R}^d$, où $\mathbb{1}$ est une matrice $1 \times d$ contenant uniquement des 1, et $m_{10} = 0, m_{20} = 3, m_{30} = 4, m_{40} = 6$. La matrice de covariance diagonale dont les éléments diagonaux sont $\sigma_i^2 = d\sigma_{i0}^2$ avec $\sigma_{10} = 1,3, \sigma_{20} = 0,45, \sigma_{30} = 0,6, \sigma_{40} = 0,9$. Nous supposons que m_{i0} et σ_{i0} sont fixés et l'écart type et la moyenne sont normalisés par $\sqrt{d}$ lorsque la dimension d'observation augmente. Nous évaluons la NMI et l'ER en augmentant progressivement la taille de l'échantillon N et la dimension d. Le paramètre plus proches voisins dans la fonction de similarité de la méthode *Spectral Clust* est réglé à KNN=10%N. Nous avons utilisé le Laplacien normalisé proposé par Shi et Malick dans [9]. Toutes les méthodes utilisent l'algorithme K-means pour le partitionnement. La table 3.1 et la table 3.2 montrent les résultats obtenus de l'information mutuelle normalisée (NMI) et le taux d'erreur (ER). La colonne *NewPCA* donne les résultats de notre approche, le *Spectral Clust* les résultats de la méthode de spectral clustering, *Kmeans* pour l'algorithme Kmeans et *NormalPCA* pour la méthode PCA. Nous avons calculé la moyenne (m) et l'écart-type (σ) de NMI et de l'ER après 100 réalisations. Nous avons reporté les valeurs de m et σ de la NMI et de l'ER dans la table 3.1 et la table 3.2 pour chaque valeur de N et d donnée.

Les résultats que nous avons obtenus dans l'expérience montrent que pour des valeurs de N et d la méthode *NewPCA* présente une valeur de NMI qui augmentent progressivement. En effet, plus la dimension de l'espace augmente, plus on a de l'information au sein des données. Les résultats de la méthode *Spectral Clust* montrent le même NMI que pour *NewPCA*

TABLE 3.1 – Information mutuelle normalisée (m=moyenne, σ=écart-type)

		NewPCA		Spectral Clust		K-means		NormalPCA	
N	d	m	σ	m	σ	m	σ	m	σ
100	2	0.79	0.05	0.80	0.05	0.78	0.06	0.77	0.05
200	5	0.95	0.04	0.95	0,01	0.89	0.07	0.91	0.07
400	10	0.95	0.08	0.98	0.02	0.93	0.08	0.94	0.08
800	50	0.95	0.08	0.97	0.05	0.96	0.08	0.94	0.09
900	80	0.98	0.06	0.98	0,04	0.96	0.08	0.92	0.10
1000	100	0.98	0.06	0.98	0,04	0.92	0.10	0.96	0.08
1300	400	0.98	0.04	0.96	0.06	0.93	0.09	0.91	0.10

TABLE 3.2 – Taux d'erreur (m=moyenne, σ=écart-type)

		NewPCA		Spectral Clust		K-means		NormalPCA	
N	d	m	σ	m	σ	m	σ	m	σ
100	2	0.12	0.09	0.09	0.03	0.17	0.13	0.20	0.13
200	5	0.02	0,08	0,01	0,01	0.15	0.17	0.12	0.16
400	10	0.07	0.14	0,01	0.04	0.10	0.16	0.08	0.15
800	50	0.09	0.16	0.05	0.10	0.07	0.15	0.10	0.17
900	80	0.03	0.11	0.02	0.07	0.07	0.15	0.14	0.18
1000	100	0.03	0.11	0.02	0.07	0.14	0.18	0.07	0.15
1300	400	0.02	0.09	0.02	0.11	0.11	0.17	0.15	0.18

sauf pour $d = 2, 10, 50$ où le *Spectral Clust* est mieux et pour $d = 400$ *NewPCA* donne le meilleur NMI. La méthode *Spectral Clust* présente un meilleur taux d'erreur de classification que la méthode *NewPCA*. La méthode *Kmeans* donne une performance inférieure à celles de *NewPCA* et *Spectral Clust*. En effet, dans le jeux de données que nous avons généré, lorsque la dimension augmente, la variance de chaque groupe de données augmente aussi ($\sigma_i^2 = d\sigma_{i0}^2$), ce qui peut engendrer un chevauchement entre les groupes. Dans ce cas, l'algorithme *Kmeans*, appliqué directement sur les données, va difficilement détecter les groupes. Ce qui explique la diminution de la NMI lorsque d augmente. La méthode *NormalPCA* est celle qui présente la plus faible valeur de NMI avec un écart-type considérable avec un taux d'erreur le plus élevé par rapport aux autres méthodes. Ceci peut s'expliquer par le fait que les groupes sont interposés et les axes principaux obtenus dans ce cas présentent une faible discrimination entre les groupes.

Nous avons illustré le principe à l'aide d'un exemple de $\mathbb{R}^2$ pour faciliter l'interprétation. Nous avons généré $N = 100$ d'échantillons aléatoires dans $\mathbb{R}^2$ suivant un modèle Gaussien. Chaque Ω_i, $i = 1, \cdots, 4$, contient 25 échantillons. Les résultats de *NewPCA*, *Spectral Clust*, *Kmeans* et *NormalPCA* sont présentés dans la figure (3.3). Les résultats de partitionnement

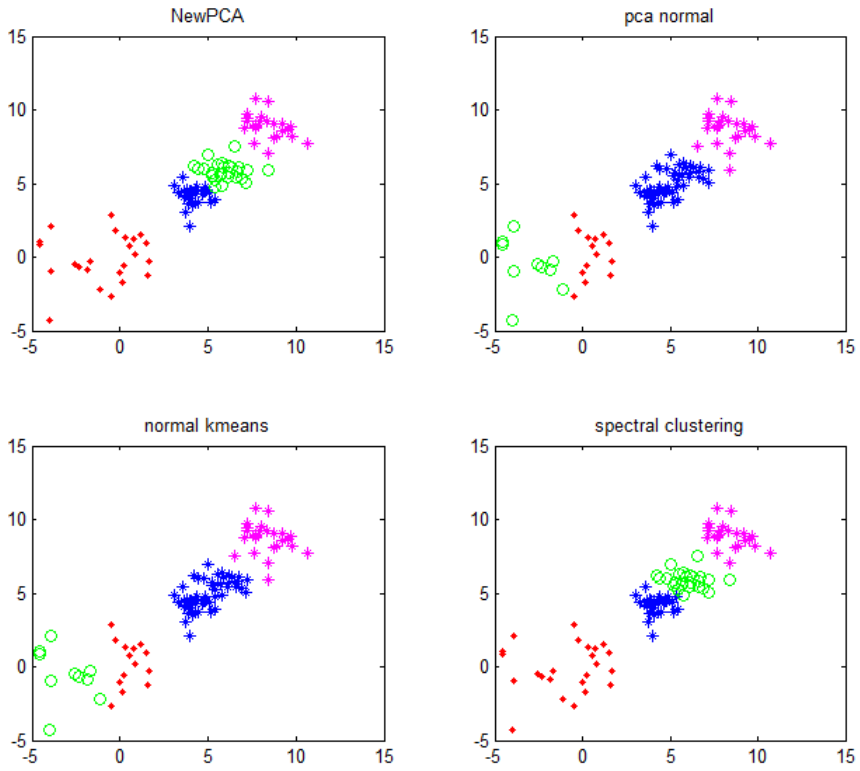


FIGURE 3.3 – Résultats du partitionnement

obtenus montrent que la méthode *NewPCA* (notre approche), dans ce cas donne des résultats assez similaires aux résultats de *Spectral Clust*, tandis que le *Kmeans* et l'ACP n'arrivent pas à détecter la bonne structure des données.

L'approche proposée dans ce cas améliore d'une part la qualité de l'ACP traditionnelle et de Kmeans puis présente des résultats comparable à ceux de la méthode de spectral clustering en présence des données en grande dimension.

3.5 Conclusion

L'analyse en composantes principales est une méthode puissante et polyvalente qui peut fournir une vue d'ensemble des données multivariées et complexes. Ce chapitre a fourni une description des concepts sur la façon d'effectuer l'ACP dans un espace multidimensionnel en utilisant la théorie des matrices aléatoires pour calculer la nouvelle décomposition spectrale de la matrice de covariance en se basant sur les estimateurs traditionnels de $\hat{\mathbf{\Sigma}}$ et les outils de matrices aléatoires. Nous avons proposé une approche pour effectuer l'analyse en composantes principales pour des données en grande dimension lorsque le nombre de variables et la taille d'échantillons tendent vers l'infini au même ordre de grandeur. L'application de la méthode d'ACP est appliquée puis une méthode de partitionnement spectral est effectuée à l'aide de la méthode *Kmeans*. Nous avons appliqué la méthode sur des données synthétiques et estimé les résultats de performance de clustering avec deux critères à savoir l'information mutuelle normalisée (NMI) et le taux d'erreur de classification (ER). L'avan-

tage de la méthode proposée est le fait que l'on n'a pas besoin de définir une fonction de noyau ou d'accorder une attention particulière sur le choix des paramètres comme c'est le cas des autres méthodes telles que le spectral clustering et les méthodes à noyau.

Chapitre 4

Analyse discriminante linéaire pour les données en grande dimension

4.1 Introduction

L'analyse discriminante linéaire, est l'une des méthodes les plus utilisées pour la réduction de dimension en apprentissage supervisé. Elle est principalement basée sur le calcul des matrices de dispersion intra-classe et inter-classes entre les échantillons de données afin de mieux les discriminer. La matrice de dispersion intra-classes mesure la quantité de la dispersion entre les échantillons dans la même classe et représente la somme de matrices de covariance des échantillons centrés au sein de la même catégorie de classe. La matrice de dispersion inter-classes quant à elle, mesure la quantité de dispersion entre les différentes catégories de classes et représente la somme des différences entre la moyenne totale des échantillons de données et la moyenne de chaque classe. L'objectif de la méthode est de grouper ensemble les échantillons qui appartiennent à la même classe tout en éloignant le plus possible les échantillons qui appartiennent à des classes différentes. Une projection des données de l'espace de départ de dimension d dans un espace de dimension inférieur à $K-1$ (où K est le nombre de classes de données) permet au mieux de séparer les données. Pour atteindre cet objectif, la méthode se base sur la résolution d'un problème d'optimisation dont la fonction objective a pour but de trouver les directions optimales où la représentation des données soit la plus discriminante. L'ensemble de ces directions, forment en effet le sous-espace dans lequel les données sont linéairement séparables et, sont calculées en effectuant la décomposition spectrale sur les matrices de dispersion.

Cependant, lorsque la taille initiale d de la dimension des données est élevée, la méthode est souvent confrontée à des difficultés de calcul de ces matrices elles-mêmes et du sous-espace de projection qu'elles engendrent, ce qui limite en effet l'application de la méthode sur les données en grande dimension. Pour surmonter ces difficultés de calcul des grandes matrices ainsi qu'à leur décomposition, on fait généralement appel à des approches de prétraitement des données dans l'espace d'origine. L'objectif est de réduire la taille de l'espace d'origine afin de l'approximer à un nouveau sous-espace. Dans ce sous-espace intermédiaire, il est plus facile d'effectuer la méthode d'analyse linéaire discriminante dès lors que les

opérations de calcul matriciel peuvent être réalisées aisément tout en limitant la perte d'information.

Ce chapitre, présente en un premier lieu dans la section 4.2, un état de l'art sur les différentes méthodes existantes permettant d'effectuer la LDA sur des grandes matrices de données. Ensuite, nous présentons une méthode de LDA rapide basée sur l'approximation de la décomposition en valeurs singulières dans la section 4.3.1 ainsi que les algorithmes dérivés de la méthode. Puis, nous présentons explicitement dans la section 4.4 les bases de données utilisés dans les simulations. Les résultats de simulation obtenus et la discussion sur les performances comparatives avec d'autre méthodes sont présentés dans la section 4.5. Enfin le chapitre se termine par une conclusion dans la section 4.6.

4.2 Description de méthodes existantes

Dans le contexte de la reconnaissance d'objets et de la classification des textes, plusieurs méthodes ont été développées dans la littérature. Certaines techniques traitent spécifiquement la LDA pour discriminer les donner en grande dimension en adoptant une transformation linéaire ou non linéaire de l'espace d'origine [97, 98]. Dans cette section, nous présentons les principales méthodes qui nous ont servi de repère dans la suite de notre travail. Nous utilisons les même notations définies dans le chapitre 2, section 2.7.

4.2.1 Régression spectrale

La régression spectrale pour l'analyse discriminante (Spectral Regression Discriminant Analysis, SRDA) est une variante de la méthode LDA. Initialement présentée par Deng et al, dans [40], la méthode SRDA part du principe que les matrices inter-classe S_b, et intra-classe S_w, peuvent être définies en fonction d'une certaine matrice Laplacienne (Laplacian Eigenmap). En effet, l'approche de Laplacian Eigenmap [99, 100] est basée sur la théorie des graphes spectraux. Un graphe $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ d'ordre N (nombre de sommets), est la donnée d'un ensemble fini de sommets $\mathbf{V}$ et d'un ensemble d'arêtes $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$. On suppose généralement que le graphe est non orienté, c'est-à-dire que si $(\mathbf{a}, \mathbf{b})$ est une arête, alors $(\mathbf{b}, \mathbf{a})$ est également une arête. On dit qu'un sommet $\mathbf{t}$ est adjacent à $\mathbf{v}$ si $(\mathbf{v}, \mathbf{t})$ est une arête dans $\mathcal{G}$, et on note $B(\mathbf{v})$ les sommets adjacents à $\mathbf{v}$. La figure (4.1) donne un exemple illustratif d'une représentation décrivant la liste de sommets adjacents à chaque sommet du graphe. Chaque graphe est représenté par sa matrice d'adjacence $\mathbf{W}$, une matrice carrée de taille $N \times N$ dont les éléments sont définis par :

$$w_{ij} = \begin{cases} 1 & \text{si} \quad (\mathbf{v}_i, \mathbf{v}_j) \in \mathbf{E} \\ 0 & \text{sinon.} \end{cases} \tag{4.1}$$

Le nombre de liens (arêtes ou arcs) qui aboutissent à un sommet définissent le degré de ce sommet. Le degré d'un sommet est la somme de tous les sommets qui lui sont directement adjacents (ou nombre de ses voisins), i.e, $|B(\mathbf{v}_i)| = \sum_j w_{ij}$. En ce sens, la matrice de degrés

D du graphe $\mathcal{G}$ est une matrice carrée diagonale dont les éléments sont définis par :

$$d_{ij} = \begin{cases} \mid B(\mathbf{v}_i) \mid & \text{si} \quad i = j \\ 0 & \text{sinon.} \end{cases} \tag{4.2}$$

Le graphe $\mathcal{G}$ est généralement lié à une certaine métrique de distance (le plus souvent Euclidienne) par rapport aux nombres de voisins les plus proches d'un sommet. Ainsi, deux sommets p et q sont connectés par un arc, lorsque la distance définie entre p et q se trouve entre les k-plus petites distances de p à d'autres sommets de $\mathbf{V}$. La matrice associée est dite matrice d'adjacence du graphe (aussi appélée matrice de poids d'arcs) dont les éléments sont définis par :

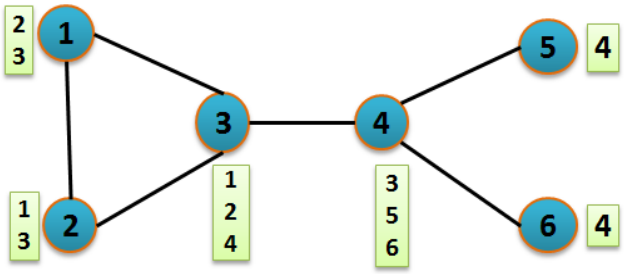


FIGURE 4.1 – Représentation de la liste des sommets adjacents à coté de chaque sommet du graphe.

$$w_{ij} = \begin{cases} 1 & \text{si} \quad \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_i) \; ou \; \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_j) \\ 0 & \text{sinon,} \end{cases} \tag{4.3}$$

où $\mathcal{N}_k(\mathbf{x}_i)$ représente l'ensemble des k points les plus proches de $\mathbf{x}_i$. En effet, si nous considérons deux vecteurs u_i et u_j de $\mathbb{R}^N$, nous pouvons écrire :

$$\begin{aligned} \sum_{(i,j)\in\mathbf{E}} (u_i - u_j)^2 &= \frac{1}{2} \sum_{i\in\mathbf{V}} \sum_{j\in\mathbf{V}:i,j\in\mathbf{E}} (u_i - u_j)^2 \\ &= \frac{1}{2} \sum_{i\in\mathbf{V}} \sum_{j\in\mathbf{V}:i,j\in\mathbf{E}} \left(u_i^2 - 2u_iu_j + u_j^2\right) \\ &= \frac{1}{2} \left(\sum_{i\in\mathbf{V}} u_i^2 d_{ii} + \sum_{j\in\mathbf{V}} u_j^2 d_{jj} \right) - \sum_{i\in\mathbf{V}} \sum_{j\in\mathbf{V}} (u_i u_j d_{ij}) \\ &= \sum_{i\in\mathbf{V}} u_i^2 d_{ii} - \sum_{i,j\in\mathbf{V}} d_{ij} u_i u_j \\ &= u^T D u - u^T W u = u^T (D - W) u \\ &= u^T L u. \end{aligned} \tag{4.4}$$

Le coefficient $1/2$ vient du fait que les arêtes du graphe sont non orientées et conduisent à une relation symétrique entre les sommets qui va dans les deux sens. Pour étudier certaines propriétés utiles du graphe, telle la similarité entre les points ou le comportement des valeurs spectrales, on se base sur les propriétés de la matrice L, appelée matrice Laplacienne donnée par $L = D - W$, où D et W représentent, respectivement, la matrice des degrés et la

matrice de poids définies précédemment. La matrice Laplacienne qui satisfait l'équation (4.4), permet de trouver la direction u qui minimise la dis-similarité entre les points, de telle sorte que les points homogènes soient dans le même groupe et ceux qui sont distincts soient dans des groupes différents. Ainsi, la solution du problème suivant :

$$\begin{aligned} &\underset{u}{\texttt{argmin}} \;\; u^T L u, \\ &s.c.\; u^T D u = 1 \end{aligned} \tag{4.5}$$

donne la direction optimale u qui minimise l'équation (4.4), où la contrainte $u^T D u = 1$ est ajoutée pour éliminer les solutions dégénérées. La résolution de ce problème est bien connue dans la littérature [101, 102].

Théorème 5 *[103] : Pour toute matrice réelle symétrique $\mathbf{Q} \in \mathbb{R}^{N \times N}$, de valeurs propres $\lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_N$, le quotient de Rayleigh qui définit une fonction d'un vecteur u telle que*

$$r(u) = \frac{u^T \mathbf{Q} u}{u^T u}, \tag{4.6}$$

admet pour extremums (maximum et minimum), les valeurs propres λ_1 et λ_N telles que

$$\lambda_1(\mathbf{Q}) \leq \frac{u^T \mathbf{Q} u}{u^T u} \leq \lambda_N(\mathbf{Q}). \tag{4.7}$$

Le théorème 5 implique que l'extremum de la fonction $r(u)$ est caractérisé par les valeurs propres de Q telles que $\mathbf{Q}u_i = \lambda_i u_i$. Ainsi, comme les matrices L et W sont par définition des matrices symétriques tout comme la matrice Q donnée dans l'énoncé du théorème 5, le problème de minimisation de l'équation (4.5) se réduit à trouver u tel que :

$$u^* = \texttt{argmin} \frac{u^T L u}{u^T D u} \Leftrightarrow \texttt{argmax} \frac{u^T W u}{u^T D u}. \tag{4.8}$$

Les solutions optimales du problème (4.8) sont données par le vecteur propre correspondant à la plus petite valeur propre du problème de diagonalisation généralisée suivant [104] :

$$Lu = \lambda D u, \tag{4.9}$$

qui est aussi équivalent à trouver le vecteur propre correspondant à la plus grande valeur propre du problème généralisé

$$Wu = \lambda D u. \tag{4.10}$$

Pour établir une liaison entre l'analyse linéaire discriminante et l'incorporation des graphes à travers la matrice Laplacienne, Deng et al procèdent par une extension linéaire de l'espace emboîtant le graphe, en choisissant une transformation linéaire $u = \mathbf{X}a$ où le vecteur $u = [u_1, \cdots, u_N]$ et $u_i = \mathbf{x}_i a$ avec $\mathbf{x}_i \in \mathbb{R}^{1 \times d}$ et $a \in \mathbb{R}^{d \times 1}$. Dans ce cas, l'équation (4.8) peut se réécrire par :

$$a^* = \texttt{argmax} \frac{u^T W u}{u^T D u} = \texttt{argmax} \frac{a^T \mathbf{X}^T W \mathbf{X} a}{a^T \mathbf{X}^T D \mathbf{X} a}. \tag{4.11}$$

La solution optimale a^* est donnée par le vecteur propre qui correspond à la plus grande valeur propre du problème de décomposition spectrale généralisé suivant :

$$\mathbf{X}^T W \mathbf{X} a = \lambda \mathbf{X}^T D \mathbf{X} a \tag{4.12}$$

Cette approche bien connue sous le nom de l'extension linéaire de l'incorporation de graphes est couramment utilisée dans plusieurs applications comme LLE (linear locally embedding), Isomap et Laplacian Eigenmap [10], la projection isométrique [105] et la projection de préservation de la localité (LPP) [106]. Pour montrer l'existence d'une liaison entre l'extension linéaire de l'équation (4.11) et la fonction objective de la LDA, on peut développer les relations à partir de la définition du principe de base de LDA. En effet, on sait que l'objectif de la LDA est de trouver a_{opt} tel que :

$$a_{opt} = \underset{a^*}{\texttt{argmax}} \frac{a^T S_b a}{a^T S_w a}, a \in \mathbb{R}^d. \tag{4.13}$$

On sait également que :

$$S_b = \sum_{k=1}^{K} N_k (m_k - m)^T (m_k - m), \tag{4.14}$$

$$S_w = \sum_{k=1}^{K} \left(\sum_{j=1}^{N_k} (\mathbf{x}_j^{(k)} - m_k)^T (\mathbf{x}_j^{(k)} - m_k) \right), \tag{4.15}$$

où nous rappelons que S_w est la matrice de dispersion intra-classes et S_b est la matrice de dispersion inter-classes, m est le vecteur de moyenne totale des échantillons, K est le nombre total des classes, m_k est le vecteur moyen de la k^{ème} classe, N_k est le nombre d'échantillons de la k^{ème} classe et $x_j^{(k)}$ est le j^{ème} échantillon dans la k^{ème} classe.

Théorème 6 *[107] : Considérons deux fonctions f et g telles que $f(\mathbf{x}) \geq 0$, $g(\mathbf{x}) \geq 0$ et $f(\mathbf{x}) + g(\mathbf{x}) > 0$, $\forall \mathbf{x} \in \mathbb{R}^d$. En définissant les fonctions h_1 et h_2 par $h_1(\mathbf{x}) = \frac{f(\mathbf{x})}{g(\mathbf{x})}$ et $h_2(\mathbf{x}) = \frac{f(\mathbf{x})}{(f(\mathbf{x})+g(\mathbf{x}))}$, alors $h_1(\mathbf{x})$ possède un maximum au point $\mathbf{x}_0$ de $\mathbb{R}^d$, si et seulement si $h_2(\mathbf{x})$ possède un maximum à ce point $\mathbf{x}_0$.*

En conséquence du théorème 6 et étant donné que les matrices S_b, S_w sont symétriques et définies positives, i.e, $a^T S_b a \geqslant 0$ et $a^T S_w a \geqslant 0$, on peut réécrire à cet effet :

$$\begin{aligned} a_{opt} &= \underset{a}{\texttt{argmax}} \frac{a^T S_b a}{a^T S_w a} = \underset{a}{\texttt{argmax}} \frac{a^T S_b a}{a^T S_b a + a^T S_w a} \\ &= \underset{a}{\texttt{argmax}} \frac{a^T S_b a}{a^T (S_b + S_w) a}. \end{aligned} \tag{4.16}$$

En posant la matrice de dispersion totale (covariance) par $S_t = S_b + S_w$, le calcul de l'optimal a_{opt} est équivalent à :

$$a_{opt} = \underset{a}{\texttt{argmax}} \frac{a^T S_b a}{a^T S_t a}.$$

En considérant les données centrées, c'est à dire $m = 0$, la matrice S_b de l'équation (4.14) peut se réécrire de la manière suivante :

$$\begin{aligned} S_b &= \sum_{k=1}^{K} N_k (m_k)^T (m_k) \\ &= \sum_{k=1}^{K} N_k \left(\frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{x}_i^{(k)} \right)^T \left(\frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{x}_i^{(k)} \right) \\ &= \sum_{k=1}^{K} (\mathbf{X}^{(k)})^T W^{(k)} \mathbf{X}^{(k)} \end{aligned} \tag{4.17}$$

où $W^{(k)}$ est une matrice de taille $N_k \times N_k$ dont tous les éléments sont égaux à $\frac{1}{N_k}$ et $\mathbf{X}^{(k)} = [\mathbf{x}_1^{(k)}, \cdots, \mathbf{x}_{N_k}^{(k)}]^T$ est la matrice qui contient l'ensemble des échantillons appartenant à la k^{ème} classe. En posant la matrice des données $\mathbf{X} = [\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(K)}]^T$, on peut également définir la matrice W de taille $N \times N$ par :

$$W = \begin{pmatrix} W^{(1)} & 0 & \cdots & 0 \\ 0 & W^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W^{(K)} \end{pmatrix}, \tag{4.18}$$

et on a

$$\begin{aligned} S_b &= \sum_{k=1}^{K} (\mathbf{X}^{(k)})^T W^{(k)} \mathbf{X}^{(k)} \\ &= \mathbf{X}^T W \mathbf{X}. \end{aligned} \tag{4.19}$$

Ce qui nous permet de réécrire l'équation (4.16) comme suit :

$$a_{opt} = \underset{a}{\texttt{argmax}} \frac{a^T S_b a}{a^T S_t a} = \underset{a}{\texttt{argmax}} \frac{a^T \mathbf{X}^T W \mathbf{X} a}{a^T \mathbf{X}^T \mathbf{X} a}. \tag{4.20}$$

De la même manière que pour l'équation (4.11), la solution optimale a_{opt} de l'équation (4.20) est donnée par le vecteur propre qui correspond à la plus grande valeur propre du problème de décomposition spectrale généralisé suivant :

$$\mathbf{X}^T W \mathbf{X} a = \lambda \mathbf{X}^T \mathbf{X} a. \tag{4.21}$$

Comme on sait que,

$$\mathbf{X}^T \mathbf{X} = S_t = S_b + S_w \text{ et que } S_b = \mathbf{X}^T W \mathbf{X},$$

on obtient la matrice S_w telle que :

$$S_w = \mathbf{X}^T (I - W) \mathbf{X} = \mathbf{X}^T L \mathbf{X}.$$

Il est clair que l'équation (4.20) peut être vue comme une extension de l'équation (4.11), d'où l'équivalence entre la LDA et l'extension linéaire de l'incorporation des graphes où

la matrice $D = I$. Pour optimiser les performances de calcul de l'ensemble de solutions du problème (4.21), Deng et al, utilisent la transformation linéaire posée par $\mathbf{X}a = u$ et ramènent le problème à la recherche du vecteur u tel que :

$$\begin{aligned} \mathbf{X}^T W \mathbf{X} a = \mathbf{X}^T W u = \mathbf{X}^T \lambda u \\ \Rightarrow W_{LDA} u = \lambda u. \end{aligned} \tag{4.22}$$

La méthode SRDA cherche un vecteur a, solution du problème (4.20), tel que $\mathbf{X}a = u$ où u est le vecteur propre associé à la valeur propre λ de la matrice W_{LDA}. Trouver u revient à résoudre le problème initial (4.8). En pratique, le calcul du vecteur u qui vérifie la condition $\mathbf{X}a = u$ n'est pas évident car cela peut conduire à des solutions dégénérées. Deng et al. proposent de relaxer ce problème de calcul de u par l'utilisation d'une méthode de régression linéaire telle que

$$a_{opt} = \underset{a^*}{\texttt{argmin}} \sum_{i=1}^{N} (\mathbf{x}_i a - u_i)^2. \tag{4.23}$$

Lorsque le nombre d'échantillons est inférieur au nombre de variables, le problème (4.23) est mal posé parce qu'il peut y avoir une infinité de solutions qui vérifient la condition de linéarité $\mathbf{X}a = u$. La technique la plus utilisée pour résoudre ce type de problème est d'imposer un critère de pénalité sur la norme de a à l'équation (4.23) conduisant à :

$$a_{opt} = \underset{a}{\texttt{argmin}} \left(\sum_{i=1}^{N} (\mathbf{x}_i a - u_i)^2 + \alpha \|a\|^2 \right), \tag{4.24}$$

La méthode pour résoudre cette équation est couramment connue sous le nom de la méthode à moindre carrée régularisée [12] (ridge regression), où $\alpha \geq 0$ est un paramètre de régularisation, qui contrôle la restriction sur a. Le problème de régularisation de l'équation (4.24) peut être reformulé sous forme matricielle par :

$$a_{opt} = \underset{a}{\texttt{argmin}} \left((\mathbf{X}a - u)^T (\mathbf{X}a - u) + \alpha a^T a \right). \tag{4.25}$$

En effectuant une dérivée de cette expression par rapport à a, et en annulant l'expression de la dérivée on trouve l'extremum dans la nouvelle expression suivante :

$$\begin{aligned} 2\mathbf{X}^T \mathbf{X} a - 2\mathbf{X}u + 2\alpha a = 0 \\ \Rightarrow (\mathbf{X}^T \mathbf{X} + \alpha I) a = \mathbf{X}u \\ \Rightarrow a = (\mathbf{X}^T \mathbf{X} + \alpha I)^{-1} \mathbf{X}u. \end{aligned} \tag{4.26}$$

Pour garantir qu'il existe un vecteur qui satisfait le système d'équation linéaire $Xa = u$, u doit être dans l'espace engendré par les vecteurs de $\mathbf{X}$. Afin de pouvoir manipuler des grandes matrices de données, la méthode SRDA exploite la définition du problème réduit de l'équation (4.22) où la matrice W est construite par bloc. En analysant ce problème, c'est à dire $Wu = \lambda u$, on sait que u est le vecteur propre de W associé à la valeur propre λ. Trouver le vecteur u revient à calculer K vecteurs $u_i^k, k = 1, \cdots, K$ tels que :

$$W^{(k)} u_i^{(k)} = \lambda_i^{(k)} u_i^{(k)}.$$

Chaque bloc de matrice $W^{(k)}$ contient $N_k \times N_k$ éléments égaux à $\frac{1}{N_k}$, comme cela a été démontré dans [9], cette matrice possède un vecteur propre $\mathbb{1}^{(k)}$ associé à la valeur propre $\mathbf{1}$ avec $\mathbb{1}^{(k)} = [1, 1, \cdots, 1]^T$. Par analogie des autres blocs, il existe exactement K vecteurs propres de W_{LDA} avec la même valeur propre égale à $\mathbf{1}$, qui sont définis par

$$u^{(k)} = [\underbrace{0, \cdots, 0}_{\sum_{i=1}^{k-1} N_i}, \underbrace{1, \cdots, 1}_{N_k}, \underbrace{0, \cdots, 0}_{\sum_{i=k+1}^{K} N_i}] \quad k = 1, \cdots, K.$$

La matrice W est une matrice diagonale définie par blocs. Les valeurs propres et vecteurs propres du bloc total correspondent à la concatenation des valeurs propres et vecteurs propres des différents blocs. Puisque tous les vecteurs u_k sont de même nature, il faudrait trouver une base dans laquelle tous les vecteurs sont orthogonaux deux à deux pour respecter la contrainte d'orthogonalité. Pour cela, on procède par le choix du vecteur $\mathbb{1}$ comme premier vecteur propre et ensuite on utilise le processus d'orthogonalisation de Gram-Schmidt pour obtenir les $K-1$ vecteurs propres orthogonaux restants et par la suite supprimer le vecteur initial, tout comme plusieurs méthodes de LDA qui ont démontré que $K-1$ vecteurs de projection sont suffisamment discriminants pour un problème de classification avec K différentes classes [108, 109, 110]. Une fois que les réponses des vecteurs u_k sont générées, la méthode SRDA permet de trouver directement les $K-1$ vecteurs discriminants a_k à travers l'expression régularisée de l'équation (4.24), qui correspondent aux $K-1$ vecteurs propres, solutions du problème (4.20).

4.2.2 Décomposition QR

La méthode LDA/QR est une variante de la LDA, initialement présentée par Ye et al, dans [41], qui applique la décomposition QR plutôt que SVD pour la réduction de la dimension. Contrairement à d'autres algorithmes basés sur LDA, cette méthode n'exige pas le stockage de toute la matrice de données dans la mémoire principale ; ce qui est souhaitable pour les grands ensembles de données. La méthode introduit l'idée d'utiliser la décomposition type $H_b = QR$ [111], où $Q \in \mathbb{R}^{K \times \rho}$ est une matrice orthogonale et $R \in \mathbb{R}^{\rho \times d}$ est une matrice triangulaire supérieure, avec ρ qui est égal au rang de la matrice H_b, ($\rho \leq K-1$). L'objectif est d'obtenir une réalisation efficace de la méthode LDA via une réduction de dimensionalité. En définissant les matrices H_w et H_b par :

$$H_w = \left[\mathbf{X}^{(1)} - \mathbb{1}_1.m_1, \cdots, \mathbf{X}^{(K)} - \mathbb{1}_K.m_K)\right] \in \mathbb{R}^{N \times d}, \tag{4.27}$$

$$H_b = \left[\sqrt{N_1}(m_1 - m), \cdots, \sqrt{N_K}(m_K - m)\right] \in \mathbb{R}^{K \times d}, \tag{4.28}$$

où $\mathbb{1}_i = [1, 1, \cdots, 1]^T \in \mathbb{R}^{N_i \times 1}$, on peut réécrire les matrices de dispersion inter-classe, S_b et intra-classe, S_w sous forme d'un produit de deux matrices. L'idée de départ repose sur la nature des matrices de dispersion, qui sont des matrices symétriques et définies positives. Par conséquent, elles peuvent être écrites sous forme $S_b = H_b^T H_b$ et $S_w = H_w^T H_w$.

Dans l'espace de dimension réduite, on peut imaginer une transformation linéaire $G \in \mathbb{R}^{d \times k}$ qui permet la transition entre l'espace d'origine et le nouvel espace réduit, à travers

l'expression :

$$(H_wG)^T(H_wG)^T = G^T H_w^T H_w G = G^T S_w G,$$
$$(H_bG)^T(H_bG)^T = G^T H_b^T H_b G = G^T S_b G.$$

La matrice de transformation optimale G, est celle qui en fait, est définie dans le problème d'optimisation de la fonction objective de la LDA et donnée par :

$$J(G) = \underset{G}{\texttt{argmax}} \ \frac{(G^T S_b G)}{(G^T S_w G)}. \tag{4.29}$$

En utilisant la décomposition QR sur la matrice $H_b = QR$, et en posant $G = QW$ pour une quelconque matrice $W \in \mathbb{R}^{\rho\times\rho}$; la fonction objective de LDA est équivalente au calcul de la matrice W telle que :

$$J(G) = \underset{G}{\texttt{argmax}} \ \frac{(G^T H_b^T H_b G)}{(G^T H_w^T H_w G)} = \underset{W}{\texttt{argmax}} \ \frac{(W^T Q^T S_b Q W)}{(W^T Q^T S_w Q W)}. \tag{4.30}$$

En posant également la matrice $\widetilde{S}_w = Q^T S_w Q$ et $\widetilde{S}_b = Q^T S_b Q$, le problème d'optimisation de LDA se réduit au calcul de W tel que :

$$J(W) = \underset{W}{\texttt{argmax}} \ \frac{(W^T \widetilde{S}_b W)}{(W^T \widetilde{S}_w W)}. \tag{4.31}$$

Après la décomposition QR, les matrices S_w et S_b sont des matrices de rang plein égal à ρ, et donc non singulières, ce qui permet d'obtenir l'ensemble des solutions optimales du problème (4.31), $w_t, t = 1, \cdots, \rho$, qui sont les vecteurs propres de la matrice $\widetilde{S}_w^{-1}\widetilde{S}_b$.

4.2.3 Projection aléatoire

La méthode par projection aléatoire proposée par Liu et al, [18], consiste d'abord à réduire la dimension de l'espace d'origine des données afin de les classifier ensuite avec des techniques de base connues, notamment la LDA. La méthode par projection aléatoire est rappelée dans le chapitre 2 section 2.4.2.1. On calcule la matrice des données réduites par la transformation

$$\widetilde{\mathbf{X}} = \mathbf{XR}, \tag{4.32}$$

où $\mathbf{X} \in \mathbb{R}^{N\times d}$ est la matrice des données initiales, et $\mathbf{R} \in \mathbb{R}^{d\times p}$ une matrice aléatoire générée dont les éléments sont donnés par $r_{ij} = \sqrt{2} \times \begin{cases} 1, & Pr = \frac{1}{4} \\ 0, & Pr = \frac{1}{2} \\ -1, & Pr = \frac{1}{4} \end{cases}$.

La matrice aléatoire $\mathbf{R}$ est une des matrices qui remplissent les conditions du théorème de Johnson-Lindenstrauss (elle peut-être également définie comme $r_{ij} \in \texttt{Uni}\{-1, +1\}$ ou $\mathcal{N}(0, 1)$). Après cette transformation linéaire des données dans l'espace réduit à travers la matrice aléatoire $\mathbf{R}$, l'étape suivante est d'appliquer directement la méthode LDA [110]. Dans ce nouvel espace, les $\mathbf{x}_i$ sont transformés en $\widetilde{\mathbf{x}}_i$. On calcule $\widetilde{S}_w$ et $\widetilde{S}_b$ à travers les $\widetilde{\mathbf{x}}_i$. Lorsque le nombre d'échantillons est plus petit que la taille de la dimension réduite, la

matrice $\widetilde{S}_w$ peut être singulière, donc non inversible. Liu et al, [18] proposent dans ce cas, une nouvelle technique de régularisation dans l'espace réduit. Cette technique consiste à imposer une transformation affine sur les valeurs propres de $\widetilde{S}_w$. Comme la décomposition spectrale de $\widetilde{S}_w$ s'écrit sous la forme :

$$\widetilde{S}_w = U_w \Delta_w U_w^T,$$

où $\Delta_w = diag\,(\lambda_1, \cdots, \lambda_\rho, 0, \cdots, 0)$ avec $\rho = rang(S_w)$, et $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_\rho > 0$. Liu et al, propose d'utiliser la méthode de régularisation sous la forme suivante :

$$\widetilde{S}_w^{reg} = U_w \widetilde{\Delta}_w U_w^T$$

à la place de $\widetilde{S}_w$, où l'exposant *reg* désigne la forme régularisée de la matrice $\widetilde{S}_w$ et $\widetilde{\Delta}_w$ est une matrice diagonale dont les éléments $\widetilde{\Delta}_{ii}$ sont définis par :

$$\widetilde{\Delta}_{ii} = \begin{cases} (\alpha\lambda_i + \beta)/M & \text{pour } \; i = 1, \cdots, \rho \\ \gamma & \text{pour } \; i = \rho+1, \cdots, d \end{cases} \tag{4.33}$$

avec $\alpha = t+1$, $\beta = \frac{\mathrm{tr}(\widetilde{S}_w)}{\rho(d-\rho)}$, $\gamma = \rho\beta$, et $0 \leq t \leq 1$. M est une constante de normalisation définie par

$$M = \frac{\alpha\,\mathrm{tr}(\widetilde{S}_w) + \rho\beta}{\mathrm{tr}(\widetilde{S}_w) - (d-\rho)\rho}.$$

Cette technique de regularisation permet en fait de déplacer affinement le support des valeurs propres via les coefficients α et β, pour pouvoir surmonter le problème de non-singularité de la matrice $\widetilde{S}_w$. Les valeurs propres nulles (correspondant aux vecteurs propres formant le sous-espace "zéro", communément appelé *nullspace*) sont régularisées par le paramètre γ. La suite consiste à résoudre le problème d'optimisation de la LDA équivalent à

$$J(G) = \underset{G}{\mathrm{argmax}} \; \frac{\det(G^T \widetilde{S}_b G)}{\det(G^T \widetilde{S}_w^{reg} G)}. \tag{4.34}$$

L'ensemble des solutions optimales, qui sont les directions de projections discriminantes de la LDA, est donné par la matrice G qui contient les vecteurs propres de la matrice $(\widetilde{S}_w^{reg})^{-1}\widetilde{S}_b$.

4.3 Analyse discriminante linéaire rapide

4.3.1 Principes et motivations de la méthode proposée

L'efficacité ou la robustesse d'une méthode se résume non seulement par sa capacité de prédiction, mais aussi, par sa rapidité en temps de calcul et par ses besoins en espace de stockage mémoire. Nous venons de présenter dans la section 4.2 quelques approches applicatives de l'analyse discriminante linéaire face au défi de la grande dimension des données. La méthode SRDA par exemple, présente une technique en évitant la décomposition spectrale des matrices denses, mais elle utilise une méthode de régularisation avec un processus

itératif de génération des vecteurs orthogonaux pour trouver l'espace de projection discriminant. Cette démarche peut rendre la méthode lente lorsque l'on est en présence de beaucoup d'échantillons de calcul. La méthode LDA/QR, quant à elle, fait de la décomposition QR comme le principe fondamental de la technique pour trouver le nouvel espace réduit de transformation. Cependant, cette méthode réduit la dimension à $q = rang(H_b) \leq K - 1$. Or, pour la plupart des problèmes réels, cette valeur est trop petite et peut engendrer une perte d'information assez conséquente sur les données d'origine. En ce qui concerne la méthode par projection aléatoire, elle utilise une réduction de dimension classique pour transformer les données et ensuite appliquer la LDA dans l'espace réduit. Cette méthode est pratiquement la plus rapide dès lors qu'elle utilise qu'une simple transformation linéaire $\mathbf{XR}$. Cependant, cette méthode présente des résultats instables à cause de la matrice de transformation $\mathbf{R}$ qui, par définition est générée aléatoirement et cela conduit à différents résultats pour chaque séquence de génération de la matrice.

Toutes ces méthodes cherchent un meilleur espace de transformation qui permet d'optimiser les performances d'analyse et de traitement des données. Cependant, il existe au moins deux problèmes qui surviennent lors de l'analyse des grandes matrices. Premièrement, les grandes matrices sont sources des problèmes de complexité en temps de calcul et en espace mémoire pour la plupart des algorithmes : pour un début, il n'est peut-être pas possible de les stocker en mémoire. Deuxièmement, pratiquement la plupart des bases de données réelles ont une grande dimension apparente (c'est-à-dire un très grand nombre de lignes et/ou de colonnes), mais ont très souvent une dimension intrinsèque beaucoup plus faible (effet des matrices creuses). Cela signifie que la majorité des variables sont effectivement redondants et obscurcissent la véritable dimension des données.

Pour éviter ces deux problèmes, les applications se concentrent sur des approximations de matrices de faible rang. Une approximation de faible rang d'une matrice $\mathbf{X}$ est une matrice $\hat{\mathbf{X}}$ pour laquelle le $rang(\hat{\mathbf{X}}) \ll rang(\mathbf{X})$, et où $\|\mathbf{X}-\hat{\mathbf{X}}\|_M$ est borné pour une certaine norme matricielle $\|.\|_M$ (le choix de la norme est généralement $\|.\|_2$ ou $\|.\|_F$). L'approximation de rang inférieur d'une matrice est généralement utilisée pour calculer $\hat{\mathbf{X}}$.

La propriété la plus attrayante de cette approximation particulière est qu'elle est optimale conformément au rang dans le sens suivant : parmi toutes les matrices de rang égal à k, la matrice donnée par la SVD tronquée $\mathbf{X}_k$ donne la plus petite distance mesurée à partir de $\mathbf{X}$, dans n'importe quelle norme $\|.\|_M$ uniformément invariante [112] [1] :

$$\|\mathbf{X} - \mathbf{X}_k\|_M = \min_{rang(\widetilde{\mathbf{X}})=k} \|\mathbf{X} - \widetilde{\mathbf{X}}\|_M.$$

Le problème avec la SVD est la complexité en temps de calcul, qui est de l'ordre de $\mathcal{O}(Nd\ \texttt{min}\{N, d\})$ pour une matrice dense de taille $N \times d$ [113]. Cette super-linéarité à la taille des données d'entrée, rend le calcul impossible sur des ensembles de données très volumineux. Nous avons présenté dans la section 2.4.2.2, la méthode détaillée de l'approximation de la SVD. Son objectif est de construire une matrice $\widetilde{\mathbf{X}}_k$ qui représente une approximation de la matrice $\mathbf{X}_k$ de telle sorte que $\mathbf{X}GG^T$ soit une matrice de rang inférieur au rang de $\mathbf{X}$

1. Une norme unitaire $\|.\|_M$ est dite invariante si $\|U\mathbf{X}V\|_M = \|\mathbf{X}\|_M$ pour tout $\mathbf{X}$ et toute matrice unitaire U, V

et qui approxime $\mathbf{X}$. La matrice G est une matrice de projection orthogonale calculée au préalable, en utilisant une approximation de la décomposition SVD. La construction d'une telle matrice de projection $G \in \mathbb{R}^{d\times k}$, a été proposée dans la littérature par divers travaux comme par exemple [108, 112, 114, 115]. L'utilisation des techniques d'approximation matricielle permet de reconstruire l'espace des variables pour réduire la taille de l'espace d'origine.

Dans la suite de cette section, nous présentons une approche en *deux phases*. La première phase consiste à introduire le calcul d'une transformation linéaire $G \in \mathbb{R}^{d\times k}$ pour réduire la dimension et la seconde phase consiste à appliquer l'analyse linéaire discriminante dans l'espace réduit. Nous proposons de calculer une matrice $\widetilde{\mathbf{X}}$ de rang inférieur de sorte que l'approximation

$$\|\mathbf{X} - \widetilde{\mathbf{X}}\|_F \approx \|\mathbf{X} - \mathbf{X}_k\|_F, \tag{4.35}$$

soit vraie et que $\widetilde{\mathbf{X}}$ soit plus rapide à calculer que $\mathbf{X}_k$. L'idée de base consiste d'utiliser une étape de pré-traitement pour obtenir une nouvelle méthode approximative de réduction de dimension. En effet, si nous appliquons d'abord une projection aléatoire, puis nous procédons à une méthode d'approximation sur la matrice résultante, nous pouvons obtenir une bonne approximation [116, 117, 118]. L'objectif est de minimiser l'erreur d'approximation de l'équation (4.35) que nous définissons de la manière suivante :

$$\delta_F(\mathbf{X}, \widetilde{\mathbf{X}}_k) = \|\mathbf{X} - \widetilde{\mathbf{X}}_k\|_F - \|\mathbf{X} - \mathbf{X}_k\|_F,$$

où $\widetilde{\mathbf{X}}_k$ est une nouvelle approximation de la SVD tronquée $\mathbf{X}_k$. Si cette erreur d'approximation est petite, cela signifie que $\widetilde{\mathbf{X}}_k$ est proche de la matrice optimale qui minimise l'erreur de reconstruction et possède un rang inférieur à celui de la matrice $\mathbf{X}$. Plusieurs résultats dans la littérature confirment l'existence d'une telle matrice $\widetilde{\mathbf{X}}_k$ avec un petit terme d'erreur $\delta_F(\mathbf{X}, \widetilde{\mathbf{X}}_k)$ dont la procédure de calcul est plus rapide que le calcul de $\mathbf{X}_k$ [119, 120, 29, 121]. Notre objectif est de développer une technique qui permet d'améliorer le temps de calcul de $\widetilde{\mathbf{X}}_k$ tout en limitant la perte d'information dans l'approximation. Pour atteindre cet objectif, nous proposons une méthode de reduction de dimension, puis nous développons deux versions modifiées de cette méthode. Chaque approche conduit à une nouvelle approximation de $\mathbf{X}_k$ via le calcul de la matrice de projection G.

4.3.2 Approches pour la réduction de dimension à l'aide de l'approximation de la SVD

4.3.2.1 Approximation classique

L'approximation de la SVD est un processus de recherche d'une matrice approchée X_k de rang égal à k ($k < d$), de sorte que la matrice initiale est contrainte de produire une description restreinte d'elle même. Si l'on considère la matrice contenant les données initiales $\mathbf{X} \in \mathbb{R}^{N\times d}$, sa décomposition SVD est donnée sous la forme :

$$\mathbf{X} = USV^T \tag{4.36}$$

où, $U \in \mathbb{R}^{N\times N}$, $V \in \mathbb{R}^{d\times d}$ et $S \in \mathbb{R}^{N\times d}$. Les matrices U et V sont orthogonales. La matrice S est une matrice semi-diagonale qui contient les valeurs singulières de $\mathbf{X}$, $\sigma_1 \geq \ldots \geq \sigma_s > 0$, avec $s \leq \texttt{min}\{N, d\}$. La forme de $\mathbf{X}_k$ décrite par

$$\mathbf{X}_k = U_k S_k {V_k}^T = \sum_{i=1}^{k} \sigma_i u_i {v_i}^T \tag{4.37}$$

est appelée la SVD tronquée de $\mathbf{X}$, où uniquement les k premières colonnes de $U = [u_1, \cdots, u_k]$ et $V = [v_1, \cdots, v_k]$ sont retenues ainsi que la sous matrice $S_{k\times k}$. Du fait de l'orthogonalité des matrices de U_k et V_k, le rang de la matrice $\mathbf{X}V_k{V_k}^T$ ($U_k{U_k}^T\mathbf{X}$) est au plus égal à k. Comme cela a été rappelé dans Boutsidis et al. [122], cette matrice représente l'approximation optimale de rang égal à k de $\mathbf{X}$, au sens de la norme de Frobenius, où l'erreur d'approximation est donnée par

$$\delta_F(\mathbf{X}, \mathbf{X}_k) = \|\mathbf{X} - U_k{U_k}^T\mathbf{X}\|_F = \|\mathbf{X} - \mathbf{X}V_kV_k^T\|_F$$

Le but est de trouver une nouvelle matrice $\mathbf{X}_k^*$, qui représente une approximée de $\mathbf{X}_k$ et qui serait plus rapide à calculer que $\mathbf{X}_k$ avec une faible erreur d'approximation. Pour ce faire, l'approche de l'approximation procède par une étape de pré-traitement qui consiste à réduire l'espace initial vers un espace intermédiaire via une transformation linéaire à travers une projection aléatoire [116]. En effet, si nous considérons la matrice $Z = \mathbf{X}\mathbf{R} \in \mathbb{R}^{N\times p}$, issue d'une projection de $\mathbf{X}$ sur le sous-espace de dimension p engendré par les colonnes d'une matrice aléatoire $\mathbf{R} \in \mathbb{R}^{d\times p}$, nous pouvons calculer la matrice V_k (resp. U_k) en effectuant une SVD tronquée sur la matrice $Z^T\mathbf{X} \in \mathbb{R}^{p\times d}$. Notons que cette dernière transformation permet de projeter les colonnes de Z sur la matrice $\mathbf{X}$. Pour une meilleure projection, une étape d'orthonormalisation sur la matrice Z est nécessaire conduisant à l'obtention d'une matrice $Q = \texttt{orth}(Z)$. Ensuite procéder au calcul de V_k dans ce cas est plus économique que d'effectuer la décomposition SVD de $\mathbf{X}$ dès lors que $d \gg p$ (en général $N > d$). Ainsi, si nous posons la forme économique de la SVD par

$$\texttt{SVD}(Q^T\mathbf{X})_{(k)} = U_k S_k V_k,$$

avec $U_k \in \mathbb{R}^{p\times k}$, $S_k \in \mathbb{R}^{k\times k}$ et $V_k \in \mathbb{R}^{d\times k}$, nous obtenons la matrice approximée de la SVD donnée par $\mathbf{X}_k^* = \mathbf{X}V_k$. Dans ce cas, la matrice $\mathbf{X}V_kV_k^T$ représente la nouvelle approximation de rang égal à k de $\mathbf{X}$ conduisant une erreur d'approximation telle que :

$$\begin{aligned} \delta_F(\mathbf{X}, \mathbf{X}_k^*) = \|\mathbf{X} - \mathbf{X}_k^*\|_F - \|\mathbf{X} - \mathbf{X}_k\|_F \leq \epsilon\|\mathbf{X} - \mathbf{X}_k\|_F, \\ \Rightarrow \|\mathbf{X} - \mathbf{X}_k^*\|_F \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}_k\|_F, \quad \text{et} \quad \delta_F(\mathbf{X}, \mathbf{X}_k^*) \leq \epsilon\|\mathbf{X} - \mathbf{X}_k\|_F, \end{aligned} \tag{4.38}$$

où le paramètre $0 < \epsilon < 1/2$ est défini dans le lemme 1.

L'approche décrit une double projection. La première consiste à projeter $\mathbf{X}$ de dimension d, dans un sous-espace intermédiaire de dimension p ($d \gg p$). Dans le nouvel espace, on obtient une matrice Z, issue d'une transformation intermédiaire. Il est rappelé dans [112, 117] et [35], qu'en projetant $\mathbf{X}$ sur l'espace engendré par les colonnes de Z, puis en calculant une approximation de faible rang égal à k dans le nouvel espace après projection, on obtient une bonne approximation de la matrice initiale $\mathbf{X}$. Ceci nous permet d'obtenir de manière

économique une approximation de la SVD. L'algorithme 4.1 décrit les étapes principles de la méthode. Il prend $\mathbf{X}$, deux entiers p et k comme entrées et donne une matrice de projection G.

Algorithme 4.1 Approximation de la SVD-`APX-SVD`

Entrées: $\mathbf{X}, p$ et k

Sorties: G

1: Générer $R \in \mathbb{R}^{d\times p}$ avec $r_{ij} \sim \mathcal{N}(0,1)$,
2: Calculer la matrice $Z = \mathbf{X}R$,
3: Calculer $Q = \texttt{orth}(Z)$,
4: Calculer $B = Q^T\mathbf{X}$ de taille $\in \mathbb{R}^{p\times d}$
5: Calculer, U, S, V tels que $B = USV^T$,
6: Retourner $G = V(:, 1:k)$.

4.3.2.2 Approximation rapide

L'étape 4 de l'algorithme 4.1 demande le calcul des vecteurs singuliers d'une matrice $B = Q^T\mathbf{X}$ de taille $p \times d$. A ce stade de l'algorithme, la décomposition peut être coûteuse en temps de calcul notamment si la dimension d est grande. Puisque le rang de B est au maximum égal à p, il n'est pas nécessaire de calculer tous les vecteurs propres. Pour améliorer l'efficacité de l'algorithme 4.1, nous proposons une autre manière de procéder pour aboutir à une approximation beaucoup plus rapide de calcul de G en utilisant des approches de l'algèbre linéaire. Plus précisément, à partir de l'équation (4.36), on peut facilement remarquer que la matrice B peut être décomposée en $B = U_B S_B V_B^T$ et on a

$$BB^T = (U_B S_B V_B^T)(U_B S_B V_B^T)^T = U_B S_B S_B^T U_B^T.$$

Donc les vecteurs singuliers à gauche de B, sont les vecteurs propres de BB^T. Et comme la matrice BB^T est de taille $p\times p$, il est plus rapide de calculer les vecteurs propres de BB^T, et déduire la matrice des vecteurs singuliers à droite, V_B, que l'on cherche, en utilisant

$$U_B^T B = S_B V_B^T.$$

En utilisant cette démarche, on peut calculer les vecteurs singuliers à droite de la matrice $B = Q^T\mathbf{X}$. Cette étape est très importante pour des raisons de complexité de temps de calcul, et permet de rendre le processus de décomposition spectrale beaucoup plus rapide. Le nouvel algorithme est appelé approximation rapide de la SVD. Les principales étapes sont présentées dans l'algorithme 4.2. Il prend $\mathbf{X}$, deux entiers p et k comme entrées et donne la matrice de projection optimale recherchée G.

4.3.2.3 Approximation rapide par saut spectral

Les sections précédentes décrivent les algorithmes 4.1 et 4.2 où le paramètre k est considéré comme une entrée choisie par l'utilisateur. Nous proposons dans cette section, une

Algorithme 4.2 Approximation rapide SVD-`FESVD`

Entrées: $\mathbf{X}, p$ et k

Sorties: G

1: Générer $R \in \mathbb{R}^{d\times p}$ avec $r_{ij} \sim \mathcal{N}(0,1)$,
2: Calculer la matrice $Z = \mathbf{X}R$,
3: Calculer $Q = \texttt{orth}(Z)$,
4: $B = Q^T\mathbf{X}$ de taille $\in \mathbb{R}^{p\times d}$
5: $T = BB^T \in \mathbb{R}^{p\times p}$,
6: $H\Delta_T H^T = \texttt{EIG}(T)$,
7: $\Sigma_{T_{ii}} = \sqrt{\Delta_{T_{ii}}}$,
8: $V = (\Sigma_T^{-1}H^TB)^T$,
9: Retourner $G = V(:, 1:k)$.

approche pour aider le choix du paramètre k. Le but de cette approche est de rendre la procédure de l'approximation rapide de la SVD concise et moins paramétrique. Elle permet de détecter automatiquement une valeur de k qui devrait permettre aux données dans l'espace réduit, de contenir l'information nécessaire véhiculée au sein des données. Pour ce faire, notre analyse est basée sur des approches statistiques telles que la méthode du coude ou encore appelée la méthode Scree [123]. La dimension du nouvel espace de données peut être trouvée en détectant le *coude* dans le Scree graphe, qui est un graphique présentant les valeurs propres d'une certaine matrice ordonnées de façon croissante.

Méthode de coude ou test de Scree :

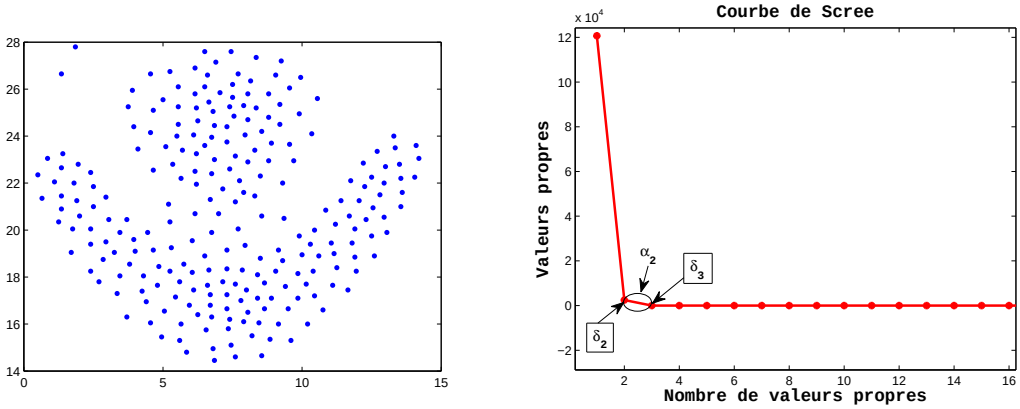
Le test Scree est un test pour déterminer le nombre de facteurs à conserver dans une analyse factorielle ou une analyse des composantes principales. Le test consiste à tracer un graphe des valeurs propres par ordre décroissant de leur amplitude par rapport à leur nombre et à déterminer l'instant où leur courbe se stabilise. La rupture entre la pente raide et le nivellement indique le nombre de facteurs significatifs, porteurs de l'information utile présente au sein de la matrice des données.

D'une manière formelle, si l'on considère par exemple une matrice donnée $\mathbf{S}$ dont les valeurs propres ordonnées sont définies par $\delta_1 \geq \cdots \geq \delta_p \geq 0$, la position du coude est détectée comme étant le plus petit écart entre les valeurs propres. Cet écart est exprimé par la distance entre les valeurs propres consécutives δ_i et δ_{i+1} et est donné par

$$\alpha_i = \delta_i - \delta_{i+1}.$$

La valeur de α_i tend vers zéro lorsque l'on atteint l'indice recherché k. Cette position $'i = k'$ définit le coude du graphe de Scree et représente le nombre des facteurs (ou la dimension des variables) suffisamment informatifs à retenir de la matrice $\mathbf{S}$. La figure 4.2 donne un exemple d'illustration du test de Scree sur un ensemble de données synthétiques $\mathbf{X} \in \mathbb{R}^{240\times 2}$, avec 240 échantillons en deux dimensions. Nous avons calculé les valeurs propres de la matrice de Gram $\mathbf{X}\mathbf{X}^T$ et tracé le graphe des valeurs propres ordonnées dans l'ordre décroissant.

Comme le montre la figure 4.2, le coude de la courbe se dresse entre la deuxième et la troisième valeur propre affichant un écart qui tend vers zéro pour les autres valeurs propres. A partir de la troisième valeur propre, la diminution régulière des valeurs propres semble se stabiliser par la suite. Le principe du test de Scree est de considérer notamment le nombre de facteurs utiles à retenir. Dans ce cas d'exemple, tout au plus trois facteurs sont discriminants pour le jeu de données. En pratique, la question du choix convenable du seuil



(a) Échantillons de données

(b) Courbe de Scree

FIGURE 4.2 – Exemple de graphe de Scree.

α est essentielle. Dans certaines situations, ce choix est dicté par les applications, mais bien souvent, il constitue un problème NP-difficile et reste assez délicat lorsque la taille des données est grande. Des travaux ont été réalisés pour proposer des techniques d'estimation du seuil α, lorsque la taille des données devient assez importante. Parmi ces travaux, la méthode communément utilisée est celle des variances isolées [124, 125].

Méthode à variances isolées

Dans l'exemple du modèle de population pointu (*spike population model*), la matrice de covariance de la population a toutes ses valeurs propres égales aux unités, à l'exception de quelques valeurs propres fixes (pointes ou *spikes*). La détermination du nombre de spikes est un problème fondamental qui apparaît dans de nombreux domaines scientifiques, y compris le traitement signal et la récupération d'information au seins des données. Des travaux récents ont proposé d'étudier le comportement asymptotique des valeurs propres de la matrice de covariance, lorsque la dimension des observations et la taille de l'échantillon augmentent vers l'infini avec un ratio qui converge vers une constante positive, c'est à dire $N \to \infty$, $\frac{p}{N} \to c > 0$, [126, 127, 128]. Passemier et al. ont montré que lorsque l'on considère les valeurs propres dans un ordre décroissant, les écarts successifs, α_i, se réduisent à des petites valeurs et tendent vers une valeur nulle lorsqu'ils s'approchent de valeurs à variances isolées.

Rappelons la définition de l'écart entre deux valeurs propres $\tilde{\delta}_i$ et $\tilde{\delta}_{i+1}$ par $\alpha_i = \tilde{\delta}_i - \tilde{\delta}_{i+1}$, et notons k l'indice recherché correspondant à $\alpha_i \to 0$ si $k \geq i$ et α_i tend vers une limite

positive si $k < i$.

Pour estimer la valeur de l'indice $i = k$ qui est égal à la taille (dimension retenue) de variables suffisamment informatives, considérons la matrice aléatoire $T \in \mathbb{R}^{p\times p}$ de l'algorithme 4.2, et dénotons ses valeurs propres ordonnées par $\tilde{\delta}_1 \geq \cdots \geq \tilde{\delta}_p \geq 0$ telles que

$$\underbrace{\tilde{\delta}_1, \cdots, \tilde{\delta}_k}_{k}, \underbrace{\tilde{\delta}_{k+1}, \cdots, \tilde{\delta}_d}_{d-k}.$$

L'estimation $\hat{k}$ de k peut être ainsi formulée par

$$\mathbb{P}\left(\hat{k} = k\right) = \mathbb{P}\left(\bigcap_{1\leq i\leq k} \{\alpha_i \geq \varepsilon\} \cap \{\alpha_{k+1} < \varepsilon\}\right). \tag{4.39}$$

L'utilisation de la probabilité ici est justifiée par le fait que les valeurs propres de la matrice T sont aléatoires. L'équation (4.39) est équivalente à l'expression suivante en termes d'événements :

$$\begin{aligned}\{\hat{k} = k\} &= \{\hat{k} = \max_i\ (\alpha_i \geq \varepsilon)\} \\ &= \{\forall i \in \{1,\ldots,k\}, \alpha_i \geq \epsilon\} \cap \{\alpha_{k+1} < \varepsilon\} \\ &= \max_i\{\forall\ j \in \{1,\ldots,i\}, \alpha_j \geq \varepsilon \text{ et } \alpha_{i+1} < \varepsilon\}, \quad i \in \{1,\ldots,p-1\}\end{aligned} \tag{4.40}$$

où, ε est un seuil soigneusement déterminé. Pour la valeur du seuil ε, Passemier et al. [128] expliquent que les valeurs propres informatives d'une matrice de données peuvent être considérées comme des variables aléatoires et sont réparties selon un taux de $N^2/3$ autour de leur moyenne. A priori, toute séquence de choix du seuil qui satisfait $\varepsilon(=\varepsilon_N) \underset{N\to\infty}{\longrightarrow} 0$ et $N^{2/3}\varepsilon(=N^{2/3}\varepsilon_N) \underset{N\to\infty}{\longrightarrow} \infty$ est admissible pour le choix de ε. En se basant sur cette hypothèse, nous adoptons le choix de la valeur du seuil par $\varepsilon = 100\frac{\sqrt{2\log(\log(N))}}{N^{\frac{2}{3}}}$. Les détails de l'approche sont présentés dans l'algorithme 4.3.

Algorithme 4.3 Approximation rapide SVD par saut spectral-`FES-GAP`

Entrées: $\mathbf{X}, p$

Sorties: G

1: Faire l'étape 1 à 4 de l'algorithme 4.2,
2: Calculer les vecteurs et valeurs propres de $T = BB^T$ tels que $T = H\Delta_T H^T$,
3: **répéter**
4: $\quad \alpha_i = \tilde{\delta}_i - \tilde{\delta}_{i+1}$, $i \in [1,\ldots,p-1]$,
5: **jusqu'à** $\alpha_i \geq \varepsilon$ et $\alpha_{i+1} < \varepsilon$
6: $k = i$,
7: $\Delta_{T_{ii}} = \sqrt{\tilde{\delta}_i}_{\{i=1,\ldots p\}}$,
8: $V = (\Delta_T^{-1} H^T B)^T \in \mathbb{R}^{d\times p}$,
9: Retourner $G = V(:, 1:k)$.

4.3.3 Description de la méthode

Une nouvelle méthode de réalisation de l'analyse linéaire discriminante pour les données en grande dimension est proposée. Cette approche est principalement constituée de deux étapes. Dans la première étape, rappelons que dans l'algorithme 4.1 une technique d'approximation de la SVD dans un environment évolutif, a été proposée. Deux versions modifiées de cet algorithme sont développées dans les algorithmes 4.2 et 4.3. Le but est de trouver une meilleure approximation des données dans l'espace initial, où nous disposons généralement d'un ensemble de points $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \in \mathbb{R}^{N\times d}$. En multipliant $\mathbf{X}$ par une matrice de projection G, c'est à dire $\widetilde{\mathbf{X}} = \mathbf{X}G$, $k \ll d$, on obtient un nouvel ensemble de points $\{\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2, \cdots, \widetilde{\mathbf{x}}_N\} \in \mathbb{R}^{N\times k}$ de taille réduite. Dans cet espace réduit, en se basant sur $\widetilde{\mathbf{X}}$, nous pouvons écrire la matrice de covariance telle que :

$$\begin{aligned}\widetilde{S}_t =& \frac{1}{N}(\widetilde{\mathbf{X}} - \widetilde{m})^T(\widetilde{\mathbf{X}} - \widetilde{m}) = \frac{1}{N}(\mathbf{X}G - mG)^T(\mathbf{X}G - mG) \\ =& \frac{1}{N}(\mathbf{X}G - mG)^T(\mathbf{X}G - mG) = G^T S G\end{aligned} \tag{4.41}$$

De la même façon nous avons :

$$\widetilde{S}_w = G^T S_w G \quad \text{et} \quad \widetilde{S}_b = G^T S_b G. \tag{4.42}$$

A partir de cette expression, nous définissons la nouvelle fonction objective de la LDA par :

$$J(\widetilde{W}) = \frac{\mathrm{tr}(\widetilde{W}^T \widetilde{S}_b \widetilde{W})}{\mathrm{tr}(\widetilde{W}^T \widetilde{S}_w \widetilde{W})},$$

où,

$$\widetilde{W}^T \widetilde{S}_b \widetilde{W} = \widetilde{W}^T G^T S_b G \widetilde{W} = (\widetilde{W}^T G^T)\ S_b\ (G\widetilde{W}) = W^T S_b W,$$

avec $W = G\widetilde{W}$. Ceci définit l'étape d'approximation d'un sous-espace de projection par l'obtention d'une matrice de projection G. Dans ce sous-espace, on preserve approximativement l'information intrinsèque au sein des données originales via une transformation linéaire $\widetilde{\mathbf{X}} = \mathbf{X}G$. L'étape suivante concerne la réalisation de la LDA pour obtenir une solution approchée donnée par une matrice $\widetilde{W}$ contenant les directions discriminantes donnée par les vecteurs propres de la matrice $\widetilde{S}_w^{-1}\widetilde{S}_b$. La matrice $\widetilde{W}$ obtenue, est considérée comme une bonne approximation de rang égal à k de la solution optimale W, dès lors que la matrice $\widetilde{\mathbf{X}}$ est considérée comme une approximation de rang k de $\mathbf{X}$. L'algorithme 4.4 donne le principe général de la méthode en utilisant les différents algorithmes de l'approche. En définissant l'erreur de reconstruction par

$$\xi \approx \|\mathbf{X} - \mathbf{X}GG^T\|_F^2 = \|\mathbf{X} - \mathbf{X}_k^*\|_F \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}_k\|_F$$

on peut remarquer en effet, plus l'approximation de la matrice G est bonne, mieux serait la prédiction et, petite serait l'erreur de reconstruction.

4.3.4 Complexité de la méthode

La complexité en temps de calcul et l'espace de stockage mémoire sont des indicateurs importants qui décrivent la performance d'une méthode. Dans cette section, nous présentons

Algorithme 4.4 LDA pour les grandes dimensions-`FLDA`

Entrées: $\mathbf{X}$, p, k and μ

Sorties: $\widetilde{W}$

1: Calculer la matrice G de taille $(d \times k)$ en utilisant l'algorithme 4.1, 4.2 ou 4.3,
2: Projeter $\mathbf{X}$ sur G pour obtenir $\widetilde{X} = \mathbf{X}G$,
3: Calculer $\widetilde{S}_w$ et $\widetilde{S}_b$ à partir de $\widetilde{\mathbf{X}}$,
4: Trouver $\widetilde{W}$, la matrice de taille $(k \times k)-$ contenant les vecteurs propres de $(\widetilde{S}_w)^{-1}\widetilde{S}_b$ si $\widetilde{S}_w$ est inversible, ou de $(\widetilde{S}_w + \mu I_k)^{-1}\widetilde{S}_b$ sinon,
5: Retourner $\widetilde{W}$.

une analyse de l'approche proposée dans le cadre général de l'algorithme 4.4. En effet, le calcul de la matrice qui contient les données réduites $\widetilde{\mathbf{X}}$, nécessite $\mathcal{O}(dN(p+k))$ opérations. Il faut compter $\mathcal{O}(k^2N)$ et $O(k^2K)$ pour calculer $\tilde{S}_w$ et $\tilde{S}_b$ respectivement. La décomposition d'une matrice dense de taille $k \times k$ prend $\mathcal{O}(k^3)$ [35, 129]. Ainsi, la complexité du temps de calcul est de l'ordre de $\mathcal{O}(dN(p+k)+k^2(N+K)+k^3)$ opérations. Le nombre d'échantillons (N) est généralement beaucoup plus grand par rapport au nombre de classes (K) et comparé à la dimension k. Ainsi, la complexité du temps de l'approche proposée peut être écrite par $\mathcal{O}(dN(p+k) + Nk^2)$, qui est linéaire en fonction de la taille d'échantillons, linéaire en fonction de la dimension choisie p et polynomiale sur k, qui est la dimension des variables réduites. Un cas particulier est lorsque nous définissons $k = p$, la complexité en temps est $\mathcal{O}(Ndp + Np^2)$ opérations de calcul. Cette valeur est polynomiale lorsque p varie. On peut remarquer que, l'étape la plus coûteuse est dominée par le terme $\mathcal{O}(Ndp)$ qui correspond à la complexité du temps de calcul de la phase de réduction de dimension. Si la matrice de données $\mathbf{X}$ est clairsemée, soit avec environ c entrées non nulles par colonne, la complexité de cette opération est de l'ordre de $O(Ncp)$ opérations. Dans les algorithmes 4.1 à 4.3, nous devons stocker la matrice Z et la matrice de données finale transformée $\tilde{\mathbf{X}}$. Par conséquent, le coût de la mémoire nécessaire est de l'ordre de $\mathcal{O}(N(p+k))$.

4.4 Présentation des données

Pour évaluer l'efficacité des algorithmes proposés, nous avons considéré trois bases de données types image et trois bases de données issues de documents/texte qui sont toutes des données réelles largement utilisées dans la littérature pour évaluer de nombreuses méthodes de classification. La statistique de l'ensemble de ces données est principalement décrite dans le tableau 4.1 ainsi que les paramètres choisis pour chaque base de données pour l'évaluation des différentes approches. Toutes les bases de données peuvent être téléchargées sur l'adresse `http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html`.

4.4.1 Données images

— **COIL20** Cette base de données contient 1440 échantillons d'images prises sur 20 différents objets. La taille de chaque image est de (32×32) pixels. La figure 4.3(b)

TABLE 4.1 – Statistique des données et valeurs des paramètres

Statistique des données				Paramètres	
data sets	samples (N)	dim (d)	# of classes (K)	p	k
MNIST	70, 000	784	10	70	40
COIL20	1,440	1, 024	20	70	50
ORL	400	4, 096	40	60	40
Reuters21578	8, 293	18, 933	65	200	120
20NewsGroups	18, 846	26, 214	20	350	150
TDT2	9, 394	36, 771	30	200	100

montre les photos de chacun des 20 objets.

— **ORL** Cette base de données contient 40 différents individus dont 10 échantillons du visage pour chaque individu. La taille de chaque image est de (64×64) pixels. La figure 4.3(a) montre un exemple illustrant l'aspect des visages de deux individus rangés sur différents angles de pose.

— **MNIST** Cette base de données contient un ensemble de 70000 échantillons de données de chiffres manuscrits ou digits (0-9). Chaque digit est de taille 28×28 pixels. La figure 4.3(c) donne un exemple d'échantillons illustratif.

4.4.2 Données textes

— **20NewsGroups** C'est une base de données qui comporte 18846 échantillons de documents provenant de 20 différentes catégories. Chaque document contient 26214 différents mots.

— **Reuters21578** Ces données ont été initialement collectées et étiquetées par *Carnegie Group, Inc. et Reuters, Ltd.* Le corpus contient 8293 documents dans 65 différentes catégories avec 18933 termes distincts.

— **TDT2** (Nist Topic Detection and Tracking corpus) Cette base de données contient environ 9394 documents dans 30 différentes catégories avec 36771 différents termes.

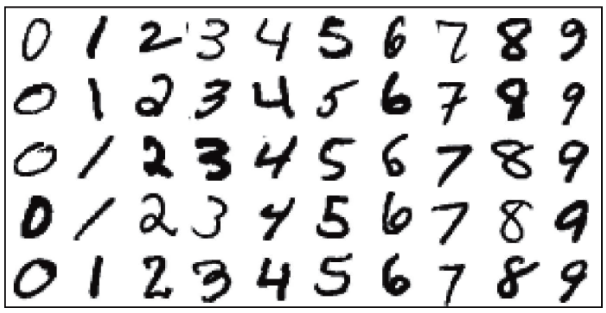
4.4.3 Normalisation des données

Il existe diverses fonctions d'évaluation communes pour le pré-traitement des données, telles que la fréquence des documents, l'information mutuelle, le gain d'information, l'entropie croisée attendue, le poids, etc. Dans notre cas, nous avons utilisé la normalisation des données images et textes sur la base de la fonction d'évaluation de la norme L_2 et document sur la base de fréquence des termes. Chaque vecteur de donnée X_j est normalisé pour avoir une norme L_2 égale à 1. Le vecteur normalisé est donné par

$$X_{j,n} = \frac{X_j}{\|X_j\|_2}.$$

(a) Images faciales ORL

(b) Images des objets Coil20



(c) Images des chiffres MNIST

FIGURE 4.3 – Échantillons d'exemples illustratifs

Pour déterminer quels mots dans un corpus de documents pourraient être plus favorables à utiliser dans une base donnée, nous avons utilisé la normalisation TFIDF (ou *terme fréquence inverse document fréquence*), pour les données textuelles [130]. Comme le terme l'indique, TF-IDF calcule des valeurs pour chaque mot dans un document par une proportion inverse de la fréquence du mot dans un document particulier au pourcentage de documents dans lesquels apparaît le mot. Les mots avec des nombres TF-IDF élevés impliquent une relation forte avec le document dans lequel ils apparaissent. La technique TFIDF permet un

repérage efficace des mots pertinents qui peuvent améliorer la récupération des catégories de différents documents présents dans un corpus. Les valeurs des éléments vectoriels $\omega_{i,j}$ pour un document sont calculées comme une combinaison des statistiques $\mathrm{TF}(i,j)$ et $\mathrm{DF}(i)$. La fréquence du terme (mot) $\mathrm{TF}(i,j)$ est le nombre de fois que le mot i apparaît dans le document j. La fréquence du document $\mathrm{DF}(i)$ est le nombre de documents dans lesquels le mot i apparaît au moins une fois [131]. La fréquence inverse du document $\mathrm{IDF}(i)$ peut être calculée à partir de la fréquence du document telle que :

$$\mathrm{IDF}(i) = \log\left(\frac{|D_{\#}|}{\mathrm{DF}(i)}\right),$$

$|D_{\#}|$ est le nombre de documents dans le corpus. La fréquence inverse du document d'un mot est faible si elle se produit dans de nombreux documents et est plus élevée si le mot se produit dans un seul. La valeur $\omega_{i,j}$ de caractéristiques i pour le document j est alors calculée comme étant le produit

$$\omega_{i,j} = \mathrm{TF}(i,j) \times \mathrm{IDF}(i)$$

Un document est représenté par un vecteur normalisé $\mathbf{\Omega}_i = \{\omega_{i,j}\}_{j=1}^{N}$ communément appelé, TFIDF où $w_{i,j}$ est le poids du $i^{\text{ème}}$ terme dans le $j^{\text{ème}}$ document, N est le nombre de de termes (mots) dans le document i, $\mathrm{TF}(i,j)$ est la fréquence de terme i dans le $j^{\text{ème}}$ document et $\mathrm{DF}(i)$ est la fréquence du $i^{\text{ème}}$ document dans le corpus. Lorsqu'il existe une large gamme de documents, le terme $\mathrm{TF}(i,j)$ est défini à l'échelle logarithmique par $1 + \log(\mathrm{TF}(i,j))$. Nous avons adapté la normalisation du corpus par

$$\omega_{i,j} = (1 + \log(\mathrm{TF}(i,j))) \times \log\left(\frac{|D_{\#}|}{\mathrm{DF}(i)}\right).$$

L'idée de base de TFIDF résulte de la théorie de la modélisation linguistique que les termes d'un document donné peuvent être considérés selon qu'un terme est ou non pertinent dans une thématique d'un document donné. L'élimination d'un terme dans un document donné peut être évaluée par la valeur du TF et celle de IDF et sert à mesurer l'importance d'un terme dans la collecte de documents.

4.5 Résultats d'expérimentation

4.5.1 Implémentation et paramétrage

Il existe trois paramètres essentiels dans la méthode proposée qui sont μ, p et k. Le paramètre μ est utilisé pour le processus de régularisation de la matrice de dispersion. La régularisation permet de palier au problème de singularité de la matrice de dispersion S_w. Dans le cas où la matrice est singulière, nous calculons $S_w + \mu I_p$ qui consiste à ajouter une perturbation sur les termes diagonaux de la matrice S_w pour s'assurer que les très petites valeurs propres sont différentes de zéro, ce qui assure la stabilité numérique lors du calcul de l'inverse de S_w. Nous avons choisis $\mu = 1$ pour toutes les expériences. Le paramètre k est la dimension de l'espace des paramètres réduit où la LDA est exécutée.

Le paramètre p est la dimension de l'espace intermédiaire où les caractéristiques d'origine sont transformées. Le choix du paramètre p est une étape sensible dans l'approche proposée. Ce paramètre devrait garantir une distorsion minimale entre les points de données après une transformation aléatoire. Dans l'espace réduit final, chaque point est représenté comme un vecteur de taille k qui conduit à un processus de classification plus rapide. Dans le tableau 4.1, nous avons donné pour chaque ensemble de données, les valeurs du paramètre correspondant p et k que nous avons utilisé pour les algorithmes 4.1 et 4.2. Pour l'algorithme 4.3, nous avons utilisé uniquement le paramètre p du tableau 4.1, k n'étant pas en entrée, l'algorithme calcule ce paramètre automatiquement.

Nous avons divisé au hasard chaque ensemble de données en deux sous-ensembles pour former un ensemble de *training* et un ensemble de *testing* en gardant le même ratio d'échantillons pour chaque classe. La taille du sous-ensemble de *training* a été définie par TN = [5 %, ..., 50 %] pour toutes à l'exception de ORL où TN = [2, 3, 4, 5, 6] puisque la base possède un petit nombre d'échantillons par classe. Toutes les expériences ont été effectuées sur une machine P7 2.7GHz Windows7 avec une mémoire de 16 Go. Nous avons utilisé le logiciel `Matlab` pour la programmation.

4.5.2 Résultats et analyse

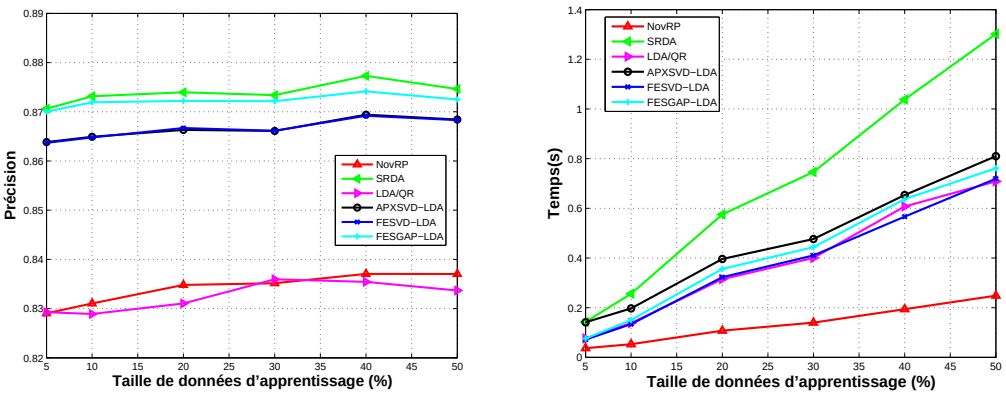
Dans cette section, nous effectuons une analyse empirique des algorithmes proposés sur les données des chiffres manuscrits, MNIST, de reconnaissance d'objet, COIL20, de reconnaissance de visage, ORL et sur les trois ensembles de données documentaires Reuters21578, 20NewsGroups et TDT2. Pour toutes les expériences, nous avons calculé en moyenne les résultats sur 50 scissions aléatoires. La précision et le temps de calcul pour toutes les approches et les ensembles de données sont affichés à partir des figures 4.4 à 4.8. En supposant que $K - 1 \leqslant k \leqslant p \ll d$, nous avons utilisé un nombre fixe de paramètres p et k donnés dans le tableau 4.1. Pour évaluer l'influence de ces paramètres, nous avons effectué également l'expérience sur COIL20 et reporté les résultats sur les figures 4.10 et 4.11. Nous nous référons aux algorithmes 4.1, 4.2 et 4.3 par `APXSVD-LDA, FESVD-LDA`, et `FESGAP-LDA` respectivement, les trois approches proposées .

L'une des observations des figures 4.5, 4.6 et 4.9 est que la précision des approches proposées est meilleure (ou compétitive, figure 4.7) par rapport aux autres méthodes de référence. Dans les figures 4.4 et 4.8, `SRDA` fournit une meilleure précision alors que son temps de calcul augmente progressivement. Dans la figure 4.8, par exemple, le temps de calcul de la `SRDA` augmente considérablement lorsque nous augmentons la taille de l'ensemble de *training*. Cela s'explique par le fait que, SRDA a besoin de plus en plus de mémoire, donc le temps d'exécution augmente considérablement. Les expériences montrent clairement comment le temps de calcul de `FESVD-LDA` et `FESGAP-LDA` améliore celui de `APXSVD-LDA` même si la précision change légèrement en fonction de la taille du *training*. Cette amélioration entraîne une amélioration considérable du temps d'exécution de la méthode.

Au vue de l'ensemble des résultats obtenus, lorsque la quantité d'échantillons dans l'ensemble de *train* varie, `FLDA` (algorithmes 4.1, 4.2 et 4.3) produisent un bon taux de détec-

tion. Plus précisément, les expériences indiquent que la précision de nos algorithmes avec un nombre relativement petit de paramètres (égal à k) surpasse les méthodes de références, sauf dans la figure 4.4 et 4.8 où la méthode `SRDA` donne des résultats de séparation beaucoup plus meilleurs, mais son temps de calcul est largement élevé. La méthode de LDA par l'approximation rapide de la SVD par saut spectral offre une meilleure précision par rapport à la LDA par l'approximation rapide de la SVD, car elle détecte automatiquement la structure intrinsèque de l'information spectrale pour obtenir la valeur raisonnable de k. Nous pouvons également voir que le temps de calcul de LDA/QR et NovRP est rapide tandis que la précision est très faible par rapport à `SRDA` et l'approche proposée.

Les figures 4.10 et 4.11 montrent l'influence des deux paramètres p et k. NovRP, `FESVD-LDA` et `FESGAP-LDA` dépendent de p car ces approches utilisent une fonction de transformation aléatoire pour la réduction de l'espace d'origine. Les autres méthodes sont invariantes lorsque p varie. Seule la méthode `FESVD-LDA` dépend de k (et `APXSVD-LDA` aussi, mais nous ne l'avons pas tracé car son temps est plus élévé). On vérifie que la précision et le temps augmentent lorsque k augmente. Au travers de toutes ces expériences, nous pouvons conclure que `FESVD-LDA` et `FESGAP-LDA` montrent des résultats encourageants de taux de prédiction et présentent un faible temps de calcul.

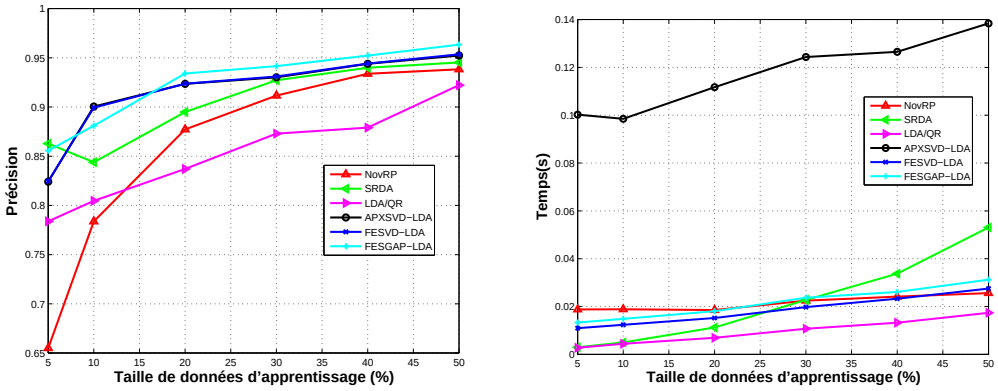


(a) (b)

FIGURE 4.4 – Résultats de simulation sur les données MINST.

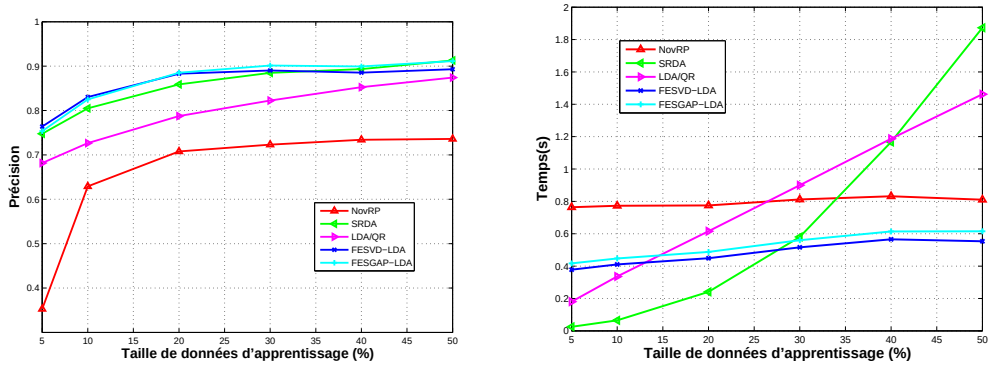
4.6 Conclusion

Dans ce chapitre, nous avons d'abord rappeler de méthodes qui traitent l'analyse linéaire discriminante (LDA) en grande dimension. Ensuite nous avons proposé une approche d'application de la LDA en utilisant l'approximation de la SVD. Deux nouvelles versions de l'approximation de la SVD ont été développées pour améliorer le temps de calcul et la précision de calcul. La complexité de calcul de la méthode a été présentée avant de décrire les conditions d'expérimentation. Des données réelles sont utilisées pour les expériences. Une discussion sur l'analyse des résultats d'expérimentation est réalisée ainsi qu'une comparaison des résultats obtenus avec les méthodes de l'état de l'art. Les résultats ont montré l'apport et l'amélioration en terme de précision et de temps de calcul sur des grandes bases



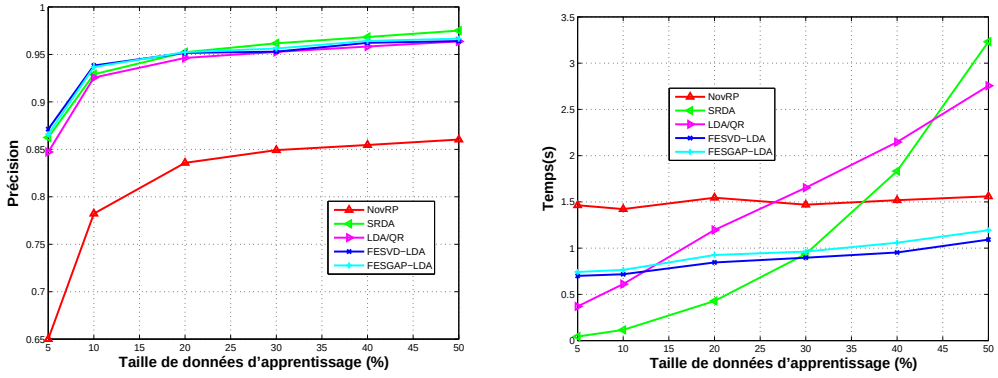
(a) (b)

FIGURE 4.5 – Résultats de simulation sur les données COIL20.



(a) (b)

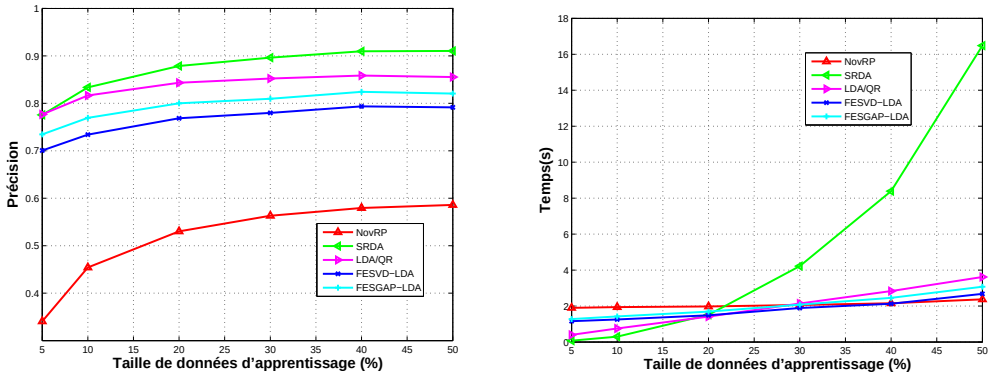
FIGURE 4.6 – Résultats de simulation sur les données Reuters21578.



(a) (b)

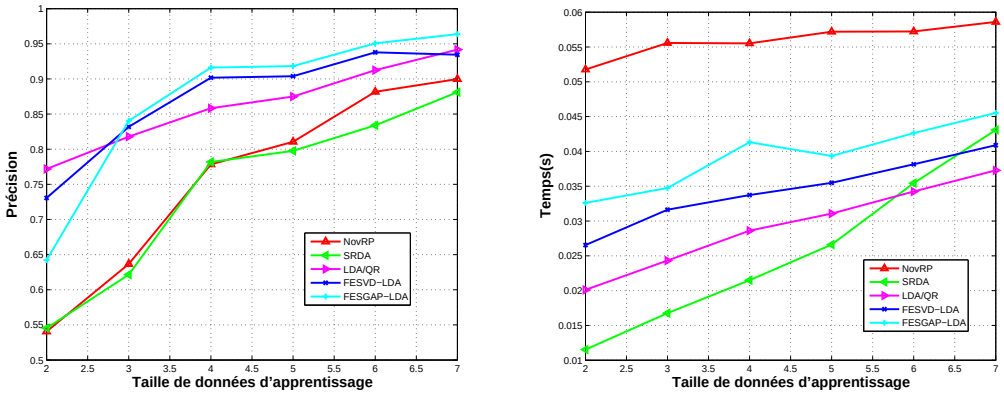
FIGURE 4.7 – Résultats de simulation sur les données TDT2.

de données. Dans cette partie, nous avons considéré que les données d'apprentissage et de validation ont la même distribution. Dans la suite de notre travail, nous allons adapté l'approche proposée dans ce chapitre avec des données qui sont issues de distribution différente dans le but d'effectuer un partage d'information au sein de données.



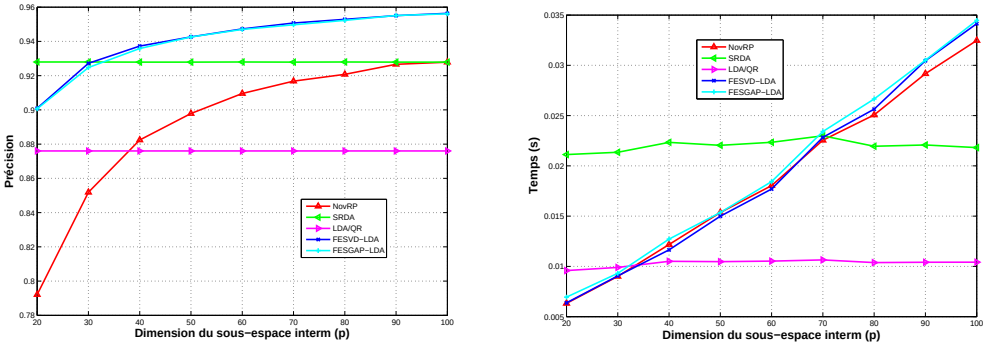
(a) (b)

FIGURE 4.8 – Résultats de simulation sur les données 20NewsGroups.



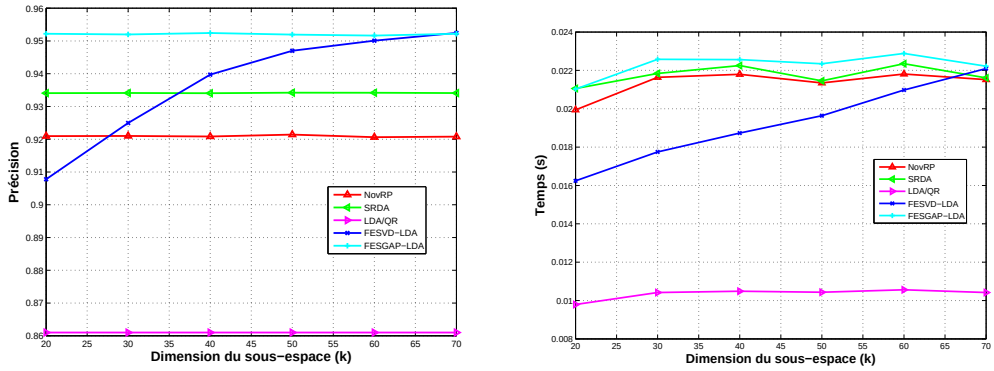
(a) (b)

FIGURE 4.9 – Résultats de simulation sur les données ORL.



(a) (b)

FIGURE 4.10 – Influence du paramètre p sur l'accuracy et le temps de calcul sur les données COIL20, avec $k = p$ fixé et TN=30%.



(a) (b)

FIGURE 4.11 – Influence du paramètre k sur l'accuracy et le temps de calcul sur les données COIL20, avec TN=30% et p=70.

Chapitre 5

Apprentissage partagé pour les données en grande dimension

5.1 Introduction

Les méthodes traditionnelles de transfert partagé entre les domaines ne parviennent pas souvent à généraliser la règle d'apprentissage pour de nouveaux échantillons de données, lorsque ces derniers proviennent d'une distribution qui diffère des échantillons d'apprentissage ou lorsqu'ils sont issus de différents espaces de variables. Lorsque cette hypothèse n'est pas satisfaite, la plupart des modèles statistiques doivent être reconstruits en utilisant de nouvelles données d'apprentissage. Cependant, il est difficile ou impossible dans certaines conditions, de construire de nouveaux modèles ou recueillir de nouvelles données [132]. Une question naturelle qui vient à l'esprit est la suivante : si un modèle est construit d'un domaine source, quelle serait sa capacité à classifier correctement des nouvelles données à partir d'un autre domaine cible dont les caractéristiques peuvent être différentes ? Dans de telles circonstances, le transfert de connaissances entre les domaines, s'il est réalisé avec succès, peut considérablement améliorer la performance de l'apprentissage en évitant les efforts d'étiquetage des données qui restent extrêmement coûteux.

L'apprentissage par transfert de connaissance entre les domaines est l'une des techniques les plus utilisées. Il tente de compenser la dégradation de la performance en transférant et en adaptant les connaissances d'un domaine source vers un domaine cible. L'idée principale consiste à rechercher un sous-espace commun ou intermédiaire entre les domaines, où les données étiquetées de l'ensemble source D_S et des données non étiquetées de l'ensemble cible D_T, partagent une quantité maximale d'informations communes ou peuvent avoir la même distribution marginale. L'adaptation entre les domaines est une méthode d'apprentissage de transfert non supervisée qui cherche un espace de projection commun, de dimension inférieure, entre les domaines source et cible et tente de rapprocher les bases des sous-espace respectifs.

Pour établir un sous-espace caractéristique commun entre les domaines, certaines méthodes se basent sur la décomposition spectrale d'une certaine fonction noyau qui permet d'approximer la distribution marginale des deux domaines. Cependant, cette décomposi-

tion peut être coûteuse lorsqu'un grand nombre d'échantillons d'apprentissage est disponible. Ainsi, les techniques d'apprentissage de transfert basées sur l'adaptation entre les domaines sont de plus en plus développées. L'alignement des sous-espaces est l'une des méthodes qui propose d'utiliser l'analyse en composantes principales (ACP) pour sélectionner un sous-espace intermédiaire où les domaines source et cible partagent une distribution marginale commune. Ainsi, la connaissance extraite des données sources est représentée sous forme d'une matrice de projection qui transforme chaque vecteur caractéristique du domaine source en un autre nouveau vecteur de représentation assez proche de celle de vecteur caractéristique cible dans le nouvel espace. L'objectif est de trouver les étiquettes inconnues pour des nouveaux échantillons dans le domaine cible en utilisant les informations disponibles dans les deux domaines. Ce type d'approche est très souvent utilisé dans diverses applications comme par exemple pour la classification des données d'origines textuelles ou images.

Ce chapitre est structuré comme suit. Dans la section 5.2, nous formulons en premier lieu la problématique du transfert ainsi qu'une présentation des travaux existants. Ensuite l'approche proposée est présentée dans la section 5.3. Les conditions d'expérimentation sont décrites dans la section 5.4, détaillant les bases de données utilisées, le choix des paramètres ainsi que les méthodes de comparaison. Une discussion sur les résultats obtenus est établie dans la section 5.4.4. Enfin, nous terminons le chapitre par une conclusion dans la section 5.5

5.2 Formulation du problème

Nous considérons le problème de l'apprentissage par transfert pour l'adaptation entre les domaines pour les problèmes de classification multi-classes. Nous définissons un espace de données par $\mathcal{X}$, un espace d'étiquettes par $\mathcal{Y}$ et une distribution par $\mathbb{P}$. Nous considérons un domaine comme une paire $D = \{\mathcal{X}, \mathbb{P}\}$ et supposons que les données proviennent de deux domaines, un domaine source ($\mathcal{D}_S$) et un domaine cible ($\mathcal{D}_T$). Les données sources sont entièrement étiquetées et définies par $\mathbf{X}_S = (\mathbf{x}_i, y_i)_{i=1}^{m}$ où $\mathbf{x}_i \in \mathbf{X}_S$ et $y_i \in \mathcal{Y}_S$. Les données cibles sont représentées par un ensemble d'échantillons non étiquetées $\mathbf{X}_T = (\mathbf{x}_i, y_i)_{i=m+1}^{m+n}$ où $\mathbf{x}_i \in \mathbf{X}_T$ et $y_i \in Y_T$. Les échantillons du domaine $\mathcal{D}_S$ sont supposés suivre une certaine distribution nommée $\mathbb{P}_S$, et ceux du domaine cible suivent une autre distribution $\mathbb{P}_T$. Dans $\mathbf{X}_T$, les étiquettes sont supposées inconnues. En apprentissage partagé, les domaines source et cible sont considérés avoir le même espace caractéristique et/ou le même ensemble d'étiquettes, mais leur distributions de probabilité marginales sont différentes, c'est-à-dire $\mathcal{X}_S = \mathcal{X}_T$ et/ou $\mathcal{Y}_S = \mathcal{Y}_T$, mais $\mathbb{P}_S \neq \mathbb{P}_T$. Dans notre cas, nous supposons que toutes les données des deux domaines sont issues du même espace caractéristique avec $\mathcal{X}_S = \mathcal{X}_T = R^d$. L'espace d'étiquette $\mathcal{Y}$ peut être adapté pour les problèmes de classification binaires ou multi-classes. L'objectif est d'apprendre une fonction de prédiction f, en se basant sur le domaine source, qui présente une faible erreur de prédiction sur le domaine cible. Un classificateur $f(\mathbf{x}_i)$, serait capable de prédire avec précision les étiquettes y_i des données cibles qui ne sont pas étiquetées avec une marge de confiance suffisamment élevée une fois que le

transfert de connaissances entre les domaines est réalisé.

Nous nous plaçons dans le cadre de l'adaptation entre les domaines(AD) [133]. C'est une méthode d'apprentissage par transfert permettant d'effectuer une tâche d'adaptation d'un système d'apprentissage d'un domaine source vers un domaine cible, (on parle aussi d'adaptation de domaine multi-sources lorsque plusieurs domaines sources sont disponibles [51, 75, 134]). La figure 5.1 donne la distinction entre l'apprentissage automatique classique et l'apprentissage par transfert. La principale différence entre ces deux thématiques réside au fait que les données disponibles dans les différents domaines d'apprentissage peuvent être complètement différents dans le cas du transfert par adaptation alors qu'elles doivent suivre la même distribution et avoir les mêmes caractéristiques pour l'apprentissage classique. L'objectif est d'apprendre une fonction de prédiction f à partir d'échantillons étiquetés ou non, issus des deux domaines $\mathcal{D}_S$ et $\mathcal{D}_T$, de telle sorte que la fonction f puisse permettre au mieux l'étiquetage de nouvelles données issues du domaine cible $\mathcal{D}_T$.

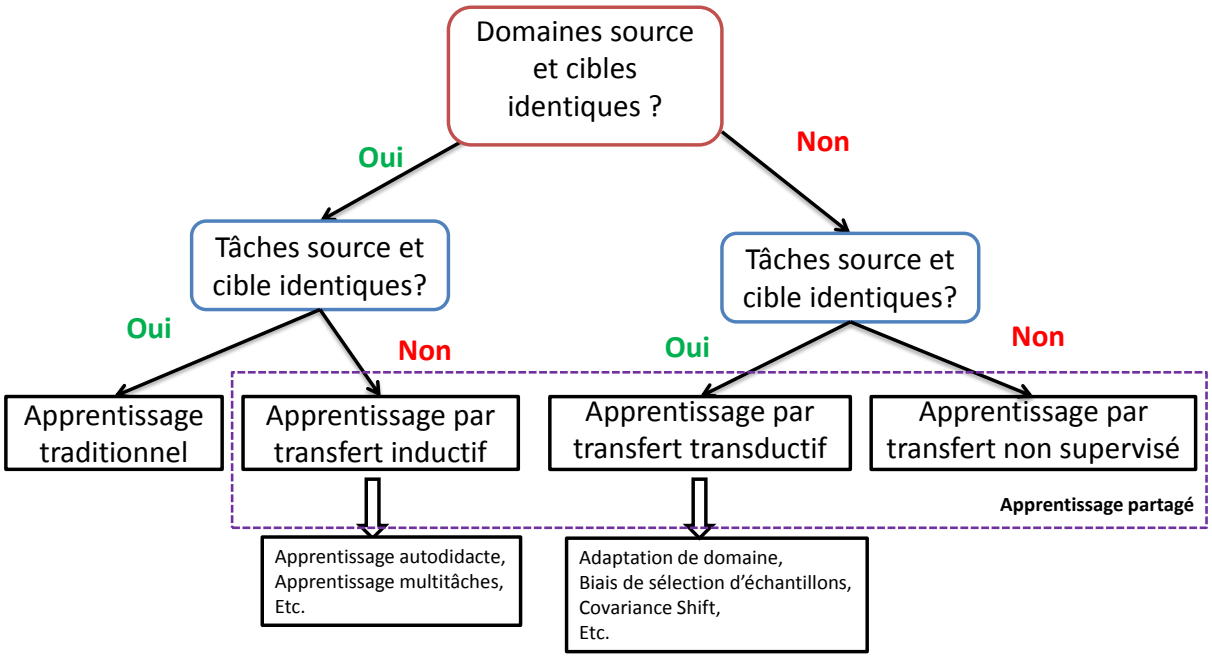


FIGURE 5.1 – Positionnement de l'adaptation entre les domaine au sein de l'apprentissage automatique

5.3 Adaptation par transfert partagé entre les domaines

D'un point de vue théorique, la performance d'un classifieur a de meilleures garanties de généralisation lorsque les distributions marginales des données du training (source) et du testing (cible) sont assez similaires [135]. Lorsque ces données proviennent de deux domaines dont les distributions marginales sont différentes, il faut évidemment trouver un moyen de maximiser la similarité (ou minimiser la dis-similarité) entre les domaines pour améliorer la performance de classification sur la base des données utilisées. Unifier ou homogénéiser

les distributions marginales des données devient une nécessité. De nombreux critères, tels que la divergence Kullback-Leibler (KL) [136], peuvent être utilisés pour optimiser le critère basé sur la distance. Cependant, beaucoup d'estimateurs sont paramétriques ou nécessitent une estimation de densité intermédiaire. Récemment, une estimation de distance non paramétrique a été conçue en intégrant des distributions dans un espace de Hilbert à noyau reproduisant (RKHS) [42]. Ces méthodes, généralement basées sur le noyau font appel à la décomposition en valeurs propres et vecteurs propres pour trouver l'espace de nouvelle représentation. Pour éviter ce type de technique assez dense, nous proposons d'utiliser l'alignement de sous-espace (SA) avec l'approximation rapide de la SVD pour une réalisation efficace du transfert entre les domaines. Pour la suite de cette section, nous présentons d'abord la méthode SA dans la section 5.3.1 puis la méthode proposée dans la section 5.3.2.

5.3.1 Méthode d'alignement des sous-espaces

La méthode d'alignement de sous-espace (SA) met l'accent sur l'utilisation du sous-espace généré par la méthode ACP afin de faire une adaptation entre les domaines. Pour une explication complète de la méthode SA, nous invitons les lecteurs à cette référence [56]. L'idée de base est d'appliquer l'ACP sur l'échantillon source, $\mathbf{X}_S$ et l'échantillon cible, $\mathbf{X}_T$ séparément en choisissant un espace de dimension commune égale à k inférieure à la dimension de l'espace d'origine, d. Cela conduit à l'obtention de deux matrices de projection G_S et G_T. Ensuite, d'aligner les données sources projetées avec les données cibles projetées dans le sous-espace commun en utilisant une matrice d'alignement sous-espace $G_a = G_S G_S^T G_T$. Pour ce faire, la méthode SA propose de réduire l'écart entre les domaines en rapprochant les sous-espaces source et cible de sorte que :

$$G^* = \underset{G}{\texttt{argmin}} \|G_S G - G_T\|_F^2, \tag{5.1}$$

où $\|.\|_F^2$ désigne la norme de Frobenius et G est la matrice de transformation qui rapproche les bases source et cible, G_S et G_T respectivement. La norme de Frobenius est invariante aux opérations orthonormales [37], et comme le sous-espace source et cible est engendré par des matrices de projection orthonormale, il en résulte que

$$\begin{aligned} G^* &= \underset{G}{\texttt{argmin}} \|G_S^T G_S G - G_S^T G_t\|_F^2 \\ &= \underset{G}{\texttt{argmin}} \|G - G_S^T G_t\|_F^2. \end{aligned} \tag{5.2}$$

De l'équation (5.2), on peut voir que la matrice de transformation optimale peut être donnée par

$$G^* = G_S^T G_T,$$

puisque les matrices de projection sont orthogonales et $G_s^T G_s = I$. Par conséquent, la matrice de projection source alignée dans le sous-espace cible est définie par

$$G_a = G_S G^* = G_S G_S^T G_T.$$

La projection des données source à travers la matrice de projection G_a, permet de transformer linéairement les données où les distributions des donnés sources et celles du

domaine cible sont alignées et minimisant la distance entre les deux domaines. Le choix du nombre de composantes d ($d \ll p$), utiles à sélectionner est détaillé dans [69]. Nous avons vu dans le chapitre 3, les limites de l'ACP lorsque la dimension est très élevée. Les composantes principales sont obtenues par décomposition en valeurs singulières de la matrice de données $\mathbf{X} \in \mathbb{R}^{n \times d}$ ou de la diagonalisation de la matrice de covariance $\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$. En terme de temps de calcul, cela devient parfois irréalisable lorsque d tend vers l'infini. Pour réduire le coût de calcul, dans le chapitre 4, nous avons présenté une méthode de réduction de dimension, *approximation rapide de la SVD (`FESVD`)*, dont le but est de réaliser de façon rapide et efficace une approximation de la SVD. Nous nous sommes inspirés de la méthode SA présentée par Fernando et al. pour proposer une méthode d'approximation d'adaptation entre les domaines à grande dimension.

5.3.2 Approximation rapide d'alignement des sous-espaces

Les approches existantes deviennent impossibles à utiliser lorsque le nombre de variables enregistrées devient trop important [137, 138]. Ici, nous suggérons d'utiliser une alternative d'apprentissage par transfert de connaissance dans un contexte de grande dimension.

Pour réaliser cette tâche, nous nous sommes inspirés des méthodes qui utilisent la réduction de dimension pour une nouvelle représentation de données. Considérons les données du domaine source et du domaine cible $\mathbf{X}_S \in \mathbb{R}^{m \times d}$ et $\mathbf{X}_T \in \mathbb{R}^{n \times d}$, respectivement, et $\mathbb{P}_S \neq \mathbb{P}_T$, nous avons utilisé la méthode de [56] pour appliquer l'alignement des sous-espaces caractéristique. L'algorithme 4.2 (`FESVD`) présenté dans le chapitre 4, permet de calculer une matrice de projection G. Le principe de la méthode de transfert est d'utiliser un alignement des sous-espaces où la matrice de projection est calculée de manière économique. Pour la suite, nous nommons la méthode :

Approximation rapide des sous-espace pour adaptation des domaines (`ASA-DA`)

Le but de cette approche est d'adapter l'approximation rapide de la SVD (`FESVD`), au lieu d'utiliser la méthode d'ACP classique [69]. Étant donné deux domaine sources et cible, les matrices de projection G_S et G_T peuvent être calculées directement avec la méthode `FESVD`. Nous pouvons ainsi évaluer l'alignement des sous-espaces à travers la matrice Ga. L'algorithme 5.1 donne les étapes principales de l'approximation de l'alignement de sous-espace proposé pour transfert entre les domaines (`ASA-DA`). L'idée derrière `ASA-DA` est de réaliser l'apprentissage par transfert sans aucun paramètre de régularisation nécessaire dans la fonction objective comme c'est imposée par beaucoup d'autres méthodes [72, 139]. La matrice S représente les nouvelles données source alignées dans un sous-espace commun aux nouvelles données cible T. Dans ce sous-espace commun au deux domaines, la divergence des distribution est minimisée. Un classifieur réalisé sur la matrice S permet de construire un modèle d'apprentissage capable de classifier les échantillons de données de la matrice T.

Algorithme 5.1 Approximation d'alignement sous-espace pour adaptation des domaines-`ASA-DA`

Entrées: : $\mathbf{X}_S$, $\mathbf{X}_T$, p et k

Sorties: : S and T

1: Calculer la matrice de projection source $G_S = \texttt{FESVD}(\mathbf{X}_S, p, k) \in \mathbb{R}^{d\times k}$;
2: Calculer la matrice de projection cible $G_T = \texttt{FESVD}(\mathbf{X}_T, p, k) \in \mathbb{R}^{d\times k}$;
3: Calculer la matrice de projection d'alignement $G_a = G_S \times G_S^T \times G_T \in \mathbb{R}^{d\times k}$;
4: Calculer les données sources alignées $S = \mathbf{X}_S \times G_a \in \mathbb{R}^{m\times k}$;
5: Calculer les données target $T = \mathbf{X}_T \times G_T \in \mathbb{R}^{n\times k}$;

Adaptation par projection de sous-espace cible-`TDA`

Pour mettre en évidence l'intérêt du transfert, nous avons présenté les algorithmes 5.2 et 5.3. L'algorithme 5.3 est en fait, la méthode classique de réduction de dimension, où l'on apprend classiquement un modèle sur l'ensemble des données source uniquement et les données cibles sont utilisées seulement pour validation du modèle. L'algorithme 5.2 propose d'apprendre une connaissance sur le domaine cible via la matrice de projection G_T. Cette matrice contient l'information contenue dans le domaine cible. La projection des données cible dans le sous-espace engendré par les colonnes de la matrice G_T conduit les deux ensembles à partager certaines caractéristiques et permet aux nouvelles représentations des données sources d'approcher les nouvelles representations des données cibles. En procédant ainsi, on minimise la divergence des sous-espaces source et cible. Les données T dans ce cas peuvent être classifiées par un classifieur construit avec les données S.

Algorithme 5.2 Adaptation par projection de sous-espace cible-`TDA`

Entrées: : $\mathbf{X}_S$, $\mathbf{X}_T$, p et k

Sorties: : S et T

1: Calculer $G_T = \texttt{FESVD}(X_T, p, k)$;
2: Calculer $S = \mathbf{X}_S \times G_T$;
3: Calculer $T = \mathbf{X}_T \times G_T$;

Algorithme 5.3 Réduction de dimension classique-`NA`

Entrées: : $\mathbf{X}_S$, $\mathbf{X}_T$, p et k

Sorties: : S and T

1: Calculer $G_S = \texttt{FESVD}(\mathbf{X}_S, p, k)$;
2: Calculer $S = \mathbf{X}_S \times G_S$;
3: Calculer $T = \mathbf{X}_T \times G_S$;

5.4 Expérimentation

Nous présentons dans cette section les résultats de simulations de la méthode. Dans la section 5.4.1, nous détaillons la composition des différentes bases de données utilisées. Puis

dans la section 5.4.2, nous présentons les méthodes de comparaison. Ensuite l'implementation et le paramétrage des méthodes sont présentés dans la section 5.4.3. Enfin, dans la section 5.4.4, nous établissons une discussion sur les résultats obtenus.

5.4.1 Présentation des données

Nous avons appliqué l'approche proposée dans un contexte de données en grande dimension pour la classification des documents, de la reconnaissance d'objets et des chiffres manuscrits. Nous avons considéré quatre bases de données construites sur la base des données présentées dans le chapitre 4. Nous avons considéré cinq bases de données qui sont largement utilisées dans les problèmes de classification pour transfert de connaissance entre les domaines. Ces bases de données sont : 20Newsgroups, Reuters-21578, COIL20, MNIST et USPS [1]. Nous avons dérivé 13 problèmes de transfert pour l'adaptation entre les domaines. Les bases de données 20Newsgroups et Reuters-21578 sont utilisées pour construire plusieurs problèmes transversales de classification binaires et les bases de données COIL20, MNIST et USPS sont utilisés pour le problème de classification multi-classes.

Les ensembles de données USPS et MNIST contiennent des images de chiffres manuscrits avec 10 classes contenant les valeurs de (0-9). La base de données USPS contient 7291 échantillons d'apprentissage et 2007 échantillons de validation avec 16×16 pixels. La base de données MNIST se compose de 60 000 échantillons d'images d'apprentissage et 10 000 échantillons d'images de validation de taille 28×28 pixels. Pour construire une base donnée de transfert entre USPS $\rightarrow$ MNIST, 1 800 échantillons ont été choisis au hasard dans la base USPS pour former un domaine source, et 2 000 ont été choisis au hasard dans MNIST pour former un domaine cible. De la même manière, pour le transfert entre MNIST $\rightarrow$ USPS on a construit une seconde base en alternant simplement les domaines source et cible. Toutes les images ont été rédimensionnées pour avoir une taille 16×16 pixels conduisant à un espace de fonctionnalité de dimension égale à 256.

La base de données COIL20 contient 20 objets avec 1 440 échantillons d'images de taille 32×32. Chaque image est prise avec une orientation de 5 degrés et 72 images sont disponibles pour chaque objet. Pour construire une base de donnée de transfert, la base de données COIL20 est séparée en deux sous-parties égales : COIL1 et COIL2. COIL1 contient des images prises dans la direction $[0°, 85°] \cup [180°, 265°]$ et COIL2 dans la direction $[90°, 175°] \cup [270°, 355°]$. Cela conduit à deux problèmes de transfert COIL1 $\rightarrow$ COIL2 et COIL2 $\rightarrow$ COIL1.

La base de données 20Newsgroups contient des données de 20 classes. Il s'agit d'une collection de textes de près de 20 000 documents dans 20 sujets différents. Certains sujets sont étroitement liés et peuvent être regroupés en une seule catégorie à un niveau supérieur, tandis que d'autres restent comme étant des catégories distinctes. Pour construire un problème de transfert, certaines sous-catégories des catégories principales sont sélectionnées pour former des échantillons du domaine source. Le domaine cible contient le reste des sous-

1. Toutes les bases de données sont disponibles à l'adresse suivante http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

catégories. Les détails des contenus pour chaque domaine de transfert sont indiqués dans le tableau 5.1.

Reuters-21578 est un autre ensemble de données de documents bien connu dans la littérature pour la catégorisation de texte. Il contient cinq catégories principales qui sont *orgs, people, places, exchanges* et *topics*. Parmi ces catégories, *orgs, people* et *places* sont les plus importantes. Pour former les applications de transfert, trois problèmes, à savoir *orgs vs. people*, *orgs vs. places* et *people vs places* sont construits. La statistique de cette base de données est donnée dans le tableau 5.2.

Tous les ensembles de données contiennent les valeurs des échantillons d'observations et les étiquettes de classes. Lorsqu'un classifieur est appliqué, les étiquettes de l'ensemble cible sont considérées comme inconnues. La vraie étiquette est utilisée uniquement pour l'estimation du taux d'erreur de classification.

TABLE 5.2 – Statistique des données Reuters-21578 data

Database	Task	# Dimension	# Samples	
			$\mathcal{D}_S$	$\mathcal{D}_T$
Reuters-21578	people vs. places	4562	1077	1077
	orgs vs. places	4415	1016	1043
	orgs vs. people	4771	1237	1208

5.4.2 Méthodes de comparaison

Dans les simulations, nous faisons référence aux algorithmes proposés par `ASA-DA` pour l'algorithme 5.1, `NA` pour l'algorithme 5.3 et `TDA` pour l'algorithme 5.2. Nous avons comparé ces algorithmes avec les méthodes suivantes :

— k(NN) : Méthode de k plus proches voisins [140] est directement appliquée sur les données sources pour construire un modèle. Les données cibles sont utilisées pour la validation du modèle.

— Adaptation de distribution jointe (`JDA`) + NN : Méthode de transfert présentée par Long et al. dans [141]. Nous avons utilisé les paramètres par défaut et appliqué KNN pour la classification.

— Alignement sous-espace (`SA`) [69] + NN : Méthode d'alignement de sous espace pour l'adaptation entre les domaines. Cette méthode utilise l'ACP pour l'unification des sous-espaces. Ensuite le classifieur NN est utilisé pour la classification.

— Analyse par transfert des composantes (`TCA`) + NN : Méthode présentée par Pan et al. dans [71]. Nous avons utilisé la même valeur de la dimension de sous-espace après décomposition, qui est égale à k dans les algorithmes proposées (algorithme 5.1, 5.3 et 5.2).

Nous disposons d'une grande quantité d'échantillons et de variables dans les données 20Newsgroups et Reuteurs. Ainsi, il n'était pas possible d'appliquer directement les méthodes `SA`,

TABLE 5.1 – Description des données 20Newsgroups

Tâche	Source|Cible	Classe 1	Classe 2	#Echantillon	# Dimension
Sci vs Talk	Source	sci.crypt sci.med	talk.politics.misc talk.religion.misc	3373	23561
	Target	sci.electronics sci.space	talk.politics.guns talk.religion.mideast	3818	
Rec vs Talk	Source	rec.autos rec.sport.baseball	talk.religion.mideast talk.politics.misc	3690	22737
	Target	rec.motorcycles rec.sport.hockey	talk.politics.guns talk.religion.misc	3525	
Rec vs Sci	Source	rec.autos rec.sport.baseball	sci.crypt sci.med	3951	22020
	Target	rec.motorcycles rec.sport.hockey	sci.electronics sci.space	3958	
Comp vs Sci	Source	comp.os.ms-windows.misc comp.sys.ibm.pc.*	sci.electronics sci.space	3911	20016
	Target	comp.graphics comp.sys.mac.*	sci.crypt sci.med	3901	
Comp vs Rec	Source	comp.graphics comp.sys.ibm.pc.*	rec.motorcycles rec.sport.baseball	3933	18825
	Target	comp.os.ms-windows.misc comp.sys.mac.*	rec.autos rec.sport.hockey	3904	
Comp vs Talk	Source	comp.os.ms-windows.misc comp.sys.ibm.pc.*	talk.religion.mideast talk.politics.misc	3911	21264
	Target	comp.graphics comp.sys.mac.*	talk.politics.guns talk.religion.misc	3904	

TABLE 5.3 – Taux d'erreur (%). $k = 20$ et $KNN = 10$.

Domaine (source → target)	Méthodes						
	NN	JDA	TCA	SA	NA	TDA	ASA-DA
USPS→ MNIST	42.90	**52.25**	33.15	44.40	28.95	50.30	48.85
MNIST→USPS	**68.88**	59.66	46.94	51.38	54.44	67.77	60.44
COIL1 → COIL2	79.86	82.36	82.77	82.77	79.86	83.61	**84.30**
COIL2 → COIL1	75.97	79.02	77.50	80.00	78.61	78.75	**81.25**

`TCA` et `JDA` en raison du temps d'exécution assez coûteux pour le calcul de leur espace de projection respectif. Pour ces bases de données, nous avons appliqué uniquement les trois algorithmes proposés. Pour les données COIL20, MNIST et USPS, nous avons évalué toutes les méthodes et comparé leurs taux d'erreur et leurs temps de calcul respectifs.

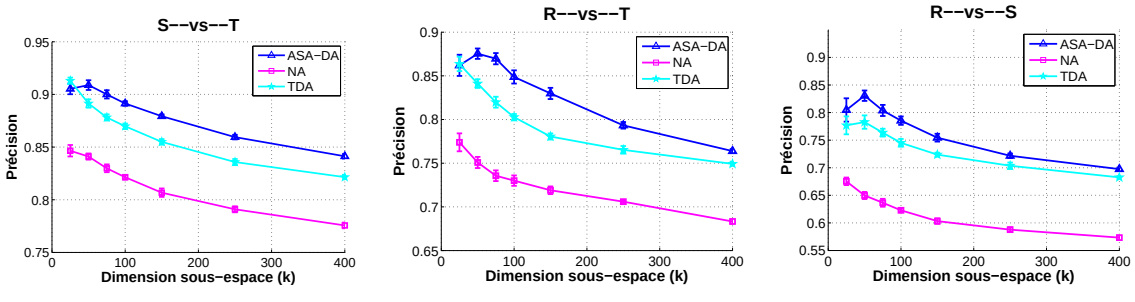
5.4.3 Implémentation et paramétrage

Toutes les méthodes ont été programmées et exécutées dans Maltab. La méthode NN a été évaluée sur les données sources étiquetées et testée sur les données cibles non étiquetées. La méthode SA a été évaluée en appliquant l'ACP pour la génération de sous-espace, puis un classifieur NN a également été utilisé. `TCA` et `JDA` sont effectuées sur tous les échantillons de données au même titre q'une procédure de réduction de dimension sur leur matrice représentative du noyau, et un classificateur NN est formé sur les données source étiquetées pour classer les données cibles non étiquetées.

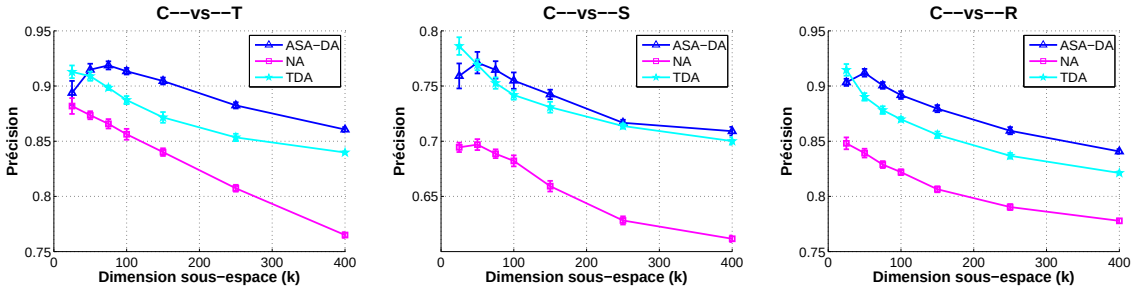
Deux paramètres principaux p et k sont utilisés dans le processus du calcul de sous-espace de l'approche proposée, puis le paramètre K, qui est égal à la valeur du plus proche voisin. Nous avons choisi $p = 2k$ et faire varier $k \in [25, 50, ..., 400]$. Toutes les méthodes ont été évaluées avec un classificateur NN. Ce paramètre a été choisi sur un intervalle pour l'évaluation des algorithmes proposés avec KNN $\in [3, 5, ..., 200]$ pour 20Newsgroups et KNN $\in [3, 5, ..., 110]$ pour Reuters-21578. Pour `TCA` et `JDA`, leur paramètre spécifique est réglé par défaut. La taille de la dimension du sous-espace est égale à k pour toutes les méthodes. Comme les étiquettes des données cibles sont disponibles, nous les avons utilisé pour évaluer la précision de la classification. Nous avons considéré le taux d'erreur sur les données cibles comme un critère d'évaluation de la méthode et est défini par

$$\texttt{Précision} = \frac{|\mathbf{x} : \mathbf{x} \in \mathbf{D}_T \wedge f(\mathbf{x}) = y|}{|\mathbf{x} : \mathbf{x} \in \mathcal{D}_T|}, \tag{5.3}$$

où $\mathcal{D}_T$ désigne le domaine cible, $f(\mathbf{x})$ affiche l'étiquette obtenu pour un point donné $\mathbf{x}$, et y est la vraie étiquette de $\mathbf{x}$. Le temps mesuré est le temps dépensé par chaque méthode pour l'apprentissage du modèle, qui est considéré comme le plus représentatif par rapport au temps de classification du domaine cible. Tous les résultats ont été calculés sur 10 essais de réalisation et la moyenne des 10 essais est reportée.

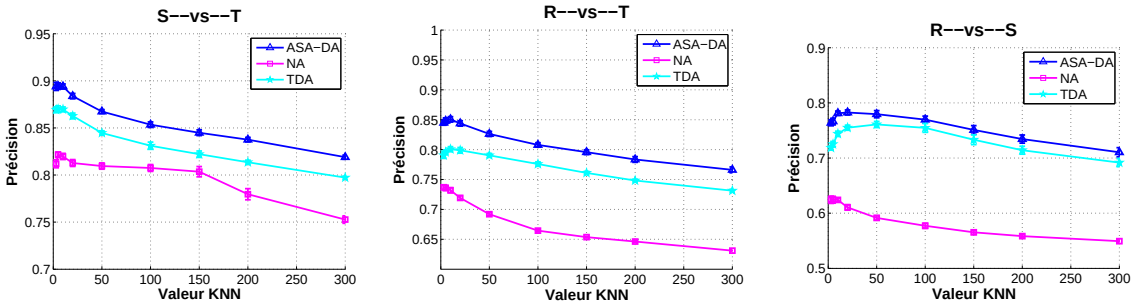


(a) Sci vs. Talk (b) Rec vs. Talk (c) Rec vs. Sci

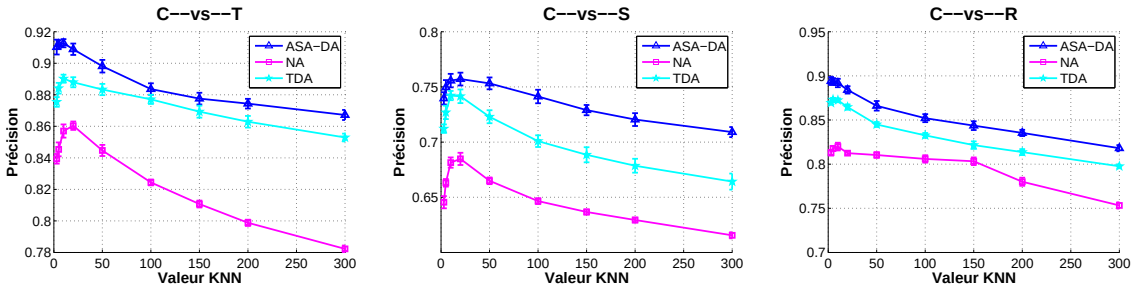


(d) Comp vs. Talk (e) Comp vs. Sci (f) Comp vs. Rec

FIGURE 5.2 – Variation du taux d'erreur en fonction de la dimension réduite sur données 20Newsgroups. Nombre de plus proches voisins $KNN = 10$.



(a) Sci vs. Talk (b) Rec vs. Talk (c) Rec vs. Sci

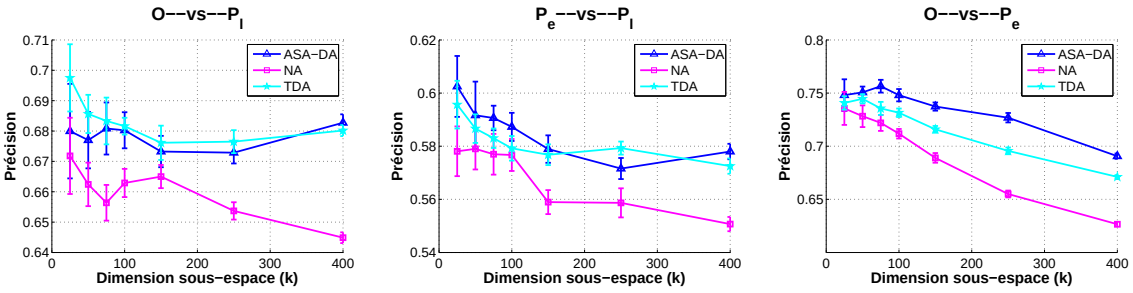


(d) Comp vs. Talk (e) Comp vs. Sci (f) Comp vs. Rec

FIGURE 5.3 – Variation du taux d'erreur en fonction du nombre de plus proches voisins KNN sur données 20Newsgroups. Dimension sous-espace $k = 100$.

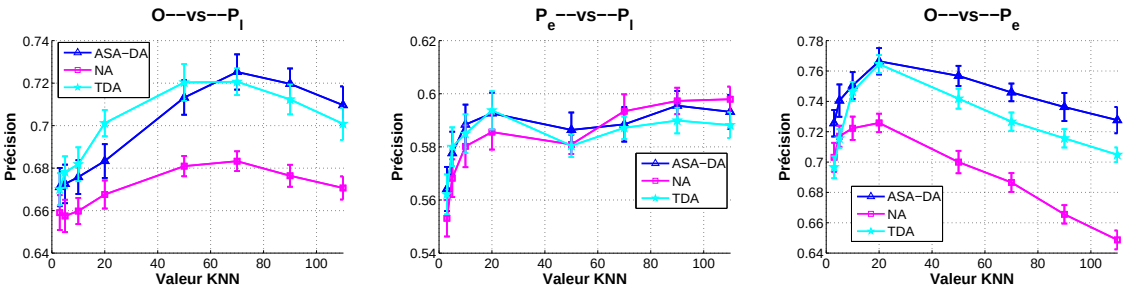
5.4.4 Résultats et analyse

Le taux d'erreur de la classification des méthodes `ASA-DA`, `NA` et `TDA` est représenté sur les figures 5.2 et 5.3 pour les données 20Newsgroups. La figure 5.2 montre la variation du



(a) Orgs vs. Places (b) People vs. Places (c) Orgs vs. People

FIGURE 5.4 – Variation du taux d'erreur en fonction de la dimension réduite sur les données Reuters. Nombre de plus proches voisins KNN=10.



(a) Orgs vs. Places (b) People vs. Places (c) Orgs vs. People

FIGURE 5.5 – Variation du taux d'erreur en fonction du nombre de plus proches voisins KNN sur données Reuters. Dimension sous-espace $k = 50$.

taux d'erreur en fonction de la dimension de sous-espace avec une valeur du plus proches voisins K = 10. La figure 5.3 montre la variation du taux d'erreur en fonction du paramètre du plus proches voisins lorsque la dimension du sous-espace choisie est fixée à $k = 100$. Sur la figure 5.2(a) à 5.2(f), lorsque la dimension du sous-espace augmente, la précision de `ASA-DA` augmente légèrement pour atteindre un maximum et diminue par la suite. La précision maximale est atteinte autour de la dimension du sous-espace $k = 50$. En ce qui concerne les méthodes `NA` et `TDA`, la précision diminue progressivement lorsque la valeur de la dimension du sous-espace augmente. Cela signifie qu'une petite valeur de sous-espace suffit à discriminer ces types de données. Le même comportement est observé lorsque le

TABLE 5.4 – Temps d'exécution pour les données images (s). $k = 20$ et $KNN = 10$.

Domaine (source → cible)	Méthodes						
	NN	JDA	TCA	SA	NA	TDA	ASA-DA
USPS→ MNIST	1.11	1.25	79.71	0.021	0.022	**0.019**	0.075
MNIST→USPS	0.917	0.92	88.64	0.054	**0.014**	0.026	0.025
COIL1 → COIL2	0.730	1.65	6.65	0.700	0.025	**0.019**	0.037
COIL2 → COIL1	0.562	1.57	6.78	0.676	**0.018**	**0.018**	0.039

nombre de plus proches voisins augmente. La performance de précision observée pour `NA`, `TDA` et `ASA-DA` dans la figure 5.2(a) à 5.2(f) montre une différence considérable en faveur de l'approche `ASA-DA` et en défaveur de `NA`. Pour les données de Reuters (figures 5.4 et 5.5), cette analyse n'est valable que pour l'adaptation de transfert entre orgs contre people. Pour people contre places, `NA`, `TDA` et `ASA-DA` donnent presque les mêmes résultats, alors que pour orgs contre places, la précision de `NA` est la plus faible et les résultats de`TDA` et `ASA-DA` restent assez proches.

Dans les Tables 5.3 et 5.4, nous avons rapporté respectivement la précision et le temps de calcul pour chaque méthode. Pour le transfert de COIL1 → COIL2 et COIL2 → COIL1, la méthode proposée `ASA-DA` présente la meilleure valeur de précision alors que, pour le transfert entre les méthodes USPS → MNIST et MNIST → USPS, `JDA` et `NN`, obtiennent respectivement la meilleure valeur de précision qui est très proche de celle de `TDA`. La méthode `TCA` présente un temps d'exécution élevé, en raison de la décomposition de la matrice du noyau dont la taille est égale au nombre d'échantillons totale (source plus cible). Le temps de calcul de l'approche `TDA` est le plus efficace et très proche de celui de `NA`, mais légèrement supérieur à celui de `ASA-DA`. `TCA` est la méthode la plus coûteuse, suivie par `JDA` par rapport aux temps d'exécution des autres méthodes.

En résumé, les résultats de simulation obtenus montrent l'intérêt de transférer les connaissances entre les domaines source et cible. La précision de la classification dépend fortement de la taille du sous-espace choisi et du nombre de plus proches voisins pour le classifieur `NN`. `ASA-DA` et `TDA` sont rapides et donnent des résultats fiables par rapport à d'autres méthodes et montrent l'utilité de la représentation des caractéristiques de faible dimension pour un apprentissage rapide dans un contexte de données en grande dimension.

5.5 Conclusion

Ce chapitre propose une approche d'apprentissage de transfert pour des données en grande dimension. L'approche définie utilise l'alignement de sous-espace et une représentation de sous-espace efficace basée sur des matrices de représentation de faible dimension. Notre approche vise à adapter les distributions marginales des domaines source et cible à l'aide de la technique de réduction de la dimensionnalité. En alignant les sous-espace des domaines source et cible, les résultats expérimentaux montrent que les approches proposées `ASA-DA` et `TDA` sont efficaces et surpassent la méthode `NA`. La performance de l'approche `TDA` proposée montre également l'avantage de transférer les connaissances du domaine cibles dans le domaine source. Les résultats de simulation obtenus sur `ASA-DA` montrent qu'en utilisant la méthode d'approximation rapide de la SVD pour la génération de sous-espace, nous obtenons de bons résultats de précision de la classification avec un faible temps d'exécution. Nous avons appliqué notre approche dans un contexte de dimension élevée où l'évaluation de certaines méthodes est pratiquement impossible en raison du temps de calcul et de l'espace de stockage mémoire.

Chapitre 6

Conclusion et perspectives

6.1 Conclusion et travaux effectués

Le travail présenté dans cette thèse est basé sur l'utilisation de méthodes aléatoires pour l'apprentissage des données en grande dimension. Nous avons dans ce cadre exploré trois grandes directions :

Notre première contribution était basée sur l'apprentissage non supervisé de données en grande dimension. Nous nous sommes intéressés à l'étude de l'analyse en composantes principales (ACP) lorsque les données sont en grande dimension. La procédure a utilisé la théorie des matrices aléatoires pour proposer un nouvel algorithme de calcul des composantes principales. Lorsque la taille de l'échantillon (N) et la dimension de l'observation (d) tendent vers l'infini, nous avons proposé une approche qui permet de calculer de nouveaux estimateurs N, d-consistants de valeurs propres et de vecteurs propres. Ces nouveaux estimateurs ont conduit à calculer un nouvel sous-espace dans lequel les données sont transformées. Une méthode de partitionnement a été appliquée sur la nouvelle représentation de données dans l'espace construit à partir de ces nouveaux estimateurs. Des expériences ont été réalisées sur des données synthétiques avec différentes valeurs de N et d. Les résultats obtenus ont été comparés avec la méthode de l'ACP classique, de partitionnement spectral et de $Kmeans$. Les performances de l'approche proposée montrent que les résultats sont comparables à ceux de la méthode du partitionnement spectral et meilleurs que ceux obtenus pour la méthode de $Kmeans$ et ACP classique. Le chapitre 3 présente le contexte dans lequel les demarches de l'approche ont été effectuées. La méthode proposée a donné des résultats qui ont fait l'objet d'une publication d'un article dans une conférence internationale [142].

Notre seconde contribution deuxième porte sur l'apprentissage supervisé des données en grande dimension à l'aide de l'analyse linéaire discriminate. Dans cette partie, nous avons proposé deux nouvelles versions adaptées aux grandes dimension, de l'analyse linéaire discriminate. Nous avons proposé une nouvelle approche qui consiste à réaliser la LDA en deux étapes. Tout d'abord, une réduction de la dimension des données est effectué grâce à un algorithme d'approximation de la matrice de données originale, et ensuite une LDA est réalisée dans l'espace réduit. L'étape de réduction de dimension est basée sur l'approxima-

tion de matrices de rang faible par l'utilisation de matrices aléatoires. Nous avons proposé un algorithme d'approximation rapide de la SVD, puis une version modifiée permettant l'approximation rapide de la SVD par saut spectral. Les approches ont été appliquées à des données réelles de type images et textes. Les expériences réalisées ont montré l'efficacité de nos méthodes par rapport à l'état de l'art, notamment en terme de taux d'erreur et de temps de calcul. Un premier résultat a été publié dans une conférence internationale [143], puis un second article a été établit portant sur l'amélioration des résultats du temps de calcul de la méthode d'approximation de la SVD. Ce article a été accepté dans une conférence internationale [144]. Une troisième version a été proposée donnant ainsi un panorama d'approches d'évaluation de la LDA en grande dimension qui a conduit à l'établissement d'une revue internationale (soumise en mars 2017). Le chapitre 4 présente l'ensemble des différents algorithmes des travaux réalisés sur cette partie.

Enfin notre troisième contribution porte sur l'apprentissage partagé entre les domaines. Il a été question d'adaptation des méthodes proposées dans le cadre d'apprentissage partagé pour les données en grande dimension. Lorsque les données d'apprentissage ne sont pas issues du même domaine d'intérêt, le transfert d'information d'un domaine d'apprentissage source vers un domaine de validation cible est nécessaire pour pouvoir prédire les données présentes dans le domaine cible. Pour ce faire, nous avons adapté l'algorithme proposé pour l'approximation rapide de la SVD pour générer des sous-espaces caractéristiques des différents domaines. En utilisant la technique d'alignement de sous-espaces, nous avons proposé une nouvelle méthode de transfert pour l'adaptation entre les domaines. Cette méthode a l'avantage d'être utilisée sur des données en grande dimension. Nous avons montré l'intérêt de l'approche proposée par l'utilisation des grandes bases de données, en particulier pour des bases de documents textuelles, pour la classification d'objects, des chiffres manuscrits ainsi que des visages. Les résultats d'expériences ont montré que l'adaptation entre les domaines améliore les performances de classification. Comparée avec d'autres méthodes, l'approche proposée donne de bonnes performances avec un temps de calcul réduit. Cette partie a conduit a fait également l'objet d'une publication d'un article dans une conférence internationale (accepté mais pas encore disponible en ligne) et une revue est actuellement en cours de rédaction. Le chapitre 5 présente les détails des travaux réalisés dans sur cette partie.

Les approches proposées dans notre travail ont l'avantage d'être appliqués dans le cas des données en grande dimension. L'approximation de matrice de rang faible est très utile dans le sens où elle permet de compresser l'information présente au sein d'une base de donnée à dimension élevée à une base à faible dimension où il est possible de réaliser une analyse et une interprétation des données assez facilement. Cependant les approches proposées présentent également certaines faiblesses. Trouver le meilleur compromis entre le temps de calcul et la taille adéquate de la dimension de l'espace réduit reste un choix difficile.

6.2 Perspectives

Les travaux réalisés ont permis de développer des techniques d'apprentissage en grande dimension avec une intention d'interprétation dans un nouvel espace qui permet de mieux analyser les données. Au-delà des contributions soulignées ci-dessus, un certain nombre de perspectives peuvent être dégagées. Pour la suite des travaux, nous planifions de poursuivre les points suivants.

- L'optimisation du paramètre de régularisation μ de la matrice de dispersion inter-classe S_w dans le chapitre 4. Lorsque cette matrice est singulière, ce paramètre est choisi pour régulariser le processus de l'inversion de la matrice S_w. Cependant, dans notre travail, et dans beaucoup de travaux de l'état de l'art d'ailleurs, le choix de ce paramètre se fait de façon arbitraire. Pour éviter cela, nous souhaitons approfondir l'étude de ce paramètre afin d'automatiser l'algorithme proposé. Il pourrait être optimisé par exemple en adaptant la technique proposée par [145].
- L'estimation de la dimension p du sous-espace intermédiaire issu de la projection aléatoire et du sous-espace final k doit être développée dans le cas de l'algorithme de l'approximation rapide de la SVD dans le chapitre 4.
- L'intégration l'information mutuelle entre les domaines est nécessaire entre les espaces caractéristiques pour renforcer le critère de minimisation de la divergence des distributions entre les sous-espaces source et cible dans le cas de l'apprentissage partagé en Chapitre 5.
- Nous pouvons considérer l'approche proposée dans le cadre du transfert dans le chapitre 5 dans un environnement évolutif où les données source arrivent séquentiellement (online) en considérant que le nombre de variables (ou d'observations) augmente progressivement. Dans ce cas, on pourrait ainsi utiliser l'information mutuelle pour sélectionner les variables source qui partagent plus d'information avec le domaine cible pour l'adaptation au transfert. Dans ce cas, on pourrait avoir l'accès en temps réel à des bases de données géantes (big data) tout en évitant potentiellement l'acquisition simultanée (stockage de grandes bases) qui pourrait nécessiter beaucoup d'espace de mémoire [2].
- Réaliser l'adaptation entre les domaines dans le cas où l'on dispose de plusieurs sources disponibles pour effectuer le transfert avec un seul domaine cible. Dans ce cas, on cherche parmi les sources disponibles, celles qui apportent plus d'informations à transférer afin d'améliorer les performances de classification du domaine cible [146].

Liste des publications

- **Journaux internationaux référencés**

  1. Nassara Elhadji Ille Gado, Edith Grall-Maës, and Malika Kharouf. "Fast-LDA : a Linear Discriminant Analysis for large scale data." (Article soumis au journal IEEE Transactions Knowledge and Data Engineering).
  2. Nassara Elhadji Ille Gado, Edith Grall-Maës, and Malika Kharouf. "Transfer Learning for Large Scale Data" (Version journal en cours de rédaction)

- **Conférences internationales avec articles étendus**

  1. Nassara Elhadji Ille Gado, Edith Grall-Maës, and Malika Kharouf. "Linear KernelPCA and K-Means Clustering Using New Estimated Eigenvectors of the Sample Covariance Matrix." In : 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE, 2015. Miami, FL, USA.
  2. Nassara Elhadji Ille Gado, Edith Grall-Maës, and Malika Kharouf. "Linear Discriminant Analysis for Large-Scale data : Application on Text and Image data." In : 2016 IEEE 15th International Conference on Machine Learning and Applications (ICMLA). IEEE, 2016. Anaheim,CA,USA.
  3. Nassara Elhadji Ille Gado, Edith Grall-Maës, and Malika Kharouf. "Linear Discriminant Analysis based on Fast Approximate SVD." In : 2017 SCITEPRESS 6th International Conference on Pattern Recognition, Applications and Methods. SCITEPRESS, 2017. Porto, Portugal.
  4. Nassara Elhadji Ille Gado, Edith Grall-Maës, and Malika Kharouf. "Transfer Learning for Large Scale Data using Subspace Alignment." In IEEE 16th International Conference on Machine Learning and Applications, ICMLA, IEEE 2017, Cancun, Mexico (Accepté).

Bibliographie

[1] Nathan Marz and James Warren. *Big Data : Principles and best practices of scalable realtime data systems*. Manning Publications Co., 2015.

[2] Kui Yu, Xindong Wu, Wei Ding, and Jian Pei. Scalable and accurate online feature selection for big data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(2) :16, 2016.

[3] Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, et al. Knowledge discovery and data mining : Towards a unifying framework.

[4] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[5] Vaishali R Patel and Rupa G Mehta. Impact of outlier removal and normalization approach in modified k-means clustering algorithm. *IJCSI International Journal of Computer Science Issues*, 8(5) :331–336, 2011.

[6] Paul S Bradley and Usama M Fayyad. Refining initial points for k-means clustering. In *ICML*, volume 98, pages 91–99, 1998.

[7] Ludmila I Kuncheva and Dmitry P Vetrov. Evaluation of stability of k-means cluster ensembles with respect to random initialization. *IEEE transactions on pattern analysis and machine intelligence*, 28(11) :1798–1808, 2006.

[8] A Reinert. Une méthode de classification descendante hiérarchique : application à l'analyse lexicale par contexte. *Les cahiers de l'analyse des données*, 8(2) :187–198, 1983.

[9] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4) :395–416, 2007.

[10] Yoshua Bengio, Jean-françcois Paiement, Pascal Vincent, Olivier Delalleau, Nicolas L Roux, and Marie Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Advances in neural information processing systems*, pages 177–184, 2004.

[11] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering : Analysis and an algorithm. *Advances in neural information processing systems*, 2 :849–856, 2002.

[12] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[13] Dennis F Sinclair. On tests of spatial randomness using mean nearest neighbor distance. *Ecology*, 66(3) :1084–1085, 1985.

[14] Reinhold Haeb-Umbach and Hermann Ney. Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 13–16. IEEE, 1992.

[15] Ethel S Gilbert. The effect of unequal variance-covariance matrices on fisher's linear discriminant function. *Biometrics*, pages 505–515, 1969.

[16] Kari Torkkola. Linear discriminant analysis in document classification. In *IEEE ICDM Workshop on Text Mining*, pages 800–806. Citeseer, 2001.

[17] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pages 41–48. IEEE, 1999.

[18] Hui Liu and Wen-Sheng Chen. A novel random projection model for linear discriminant analysis based face recognition. In *2009 International Conference on Wavelet Analysis and Pattern Recognition*, pages 112–117. IEEE, 2009.

[19] Deng Cai, Xiaofei He, and Jiawei Han. Training linear discriminant analysis in linear time. In *2008 IEEE 24th International Conference on Data Engineering*, pages 209–217. IEEE, 2008.

[20] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3) :273–297, 1995.

[21] Edgar Osuna, Robert Freund, and Federico Girosi. Support vector machines : Training and applications. 1997.

[22] Thorsten Joachims. Text categorization with support vector machines : Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[23] Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, May, 1998.

[24] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[25] CS Withers. Mercer's theorem and fredholm resolvents. *Bulletin of the Australian Mathematical Society*, 11(3) :373–380, 1974.

[26] Beresford N Parlett. The rayleigh quotient iteration and some generalizations for nonnormal matrices. *Mathematics of Computation*, 28(127) :679–693, 1974.

[27] Andreas Hoecker and Vakhtang Kartvelishvili. Svd approach to data unfolding. *arXiv preprint hep-ph/9509307*, 1995.

[28] Saburou Saitoh. *Theory of reproducing kernels and its applications*, volume 189. Longman, 1988.

[29] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction : applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM, 2001.

[30] Peter Frankl and Hiroshi Maehara. The johnson-lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory, Series B*, 44(3) :355–362, 1988.

[31] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of the johnson-lindenstrauss lemma. *International Computer Science Institute, Technical Report*, pages 99–006, 1999.

[32] Dimitris Achlioptas. Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 274–281. ACM, 2001.

[33] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3) :253–263, 2008.

[34] Nicholas J Higham. *Matrix nearness problems and applications*. University of Manchester. Department of Mathematics, 1988.

[35] Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 143–152. IEEE, 2006.

[36] Aditya Krishna Menon and Charles Elkan. Fast algorithms for approximating the singular value decomposition. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2) :13, 2011.

[37] Christos Boutsidis, Anastasios Zouzias, Michael W Mahoney, and Petros Drineas. Randomized dimensionality reduction for-means clustering. *IEEE Transactions on Information Theory*, 61(2) :1045–1062, 2015.

[38] Charles Bouveyron and Stephane Girard. Classification supervisée et non supervisée des données de grande dimension. *La revue de Modulad*, 40 :81–102, 2009.

[39] Jian Yang and Jing-yu Yang. Why can lda be performed in pca transformed space ? *Pattern recognition*, 36(2) :563–566, 2003.

[40] Deng Cai, Xiaofei He, and Jiawei Han. Srda : An efficient algorithm for large-scale discriminant analysis. *IEEE transactions on knowledge and data engineering*, 20(1) :1–12, 2008.

[41] Jieping Ye and Qi Li. Lda/qr : an efficient and effective dimension reduction algorithm and its theoretical foundation. *Pattern recognition*, 37(4) :851–854, 2004.

[42] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10) :1345–1359, 2010.

[43] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning : transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007.

[44] Xiaofeng Zhu, Zi Huang, Yang Yang, Heng Tao Shen, Changsheng Xu, and Jiebo Luo. Self-taught dimensionality reduction on the high-dimensional small-sized data. *Pattern Recognition*, 46(1) :215–229, 2013.

[45] A Evgeniou and Massimiliano Pontil. Multi-task feature learning. *Advances in neural information processing systems*, 19 :41, 2007.

[46] Zhihao Zhang and Jie Zhou. Multi-task clustering via domain adaptation. *Pattern Recognition*, 45(1) :465–473, 2012.

[47] Xiao Li, Min Fang, Ju-Jie Zhang, and Jinqiao Wu. Sample selection for visual domain adaptation via sparse coding. *Signal Processing : Image Communication*, 44 :92–100, 2016.

[48] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012.

[49] Szymon Zaręba, Marcin Kocot, and Jakub M Tomczak. Domain adaptation for image analysis : An unsupervised approach using boltzmann machines trained by perturbation. In *International Conference on Systems Science*, pages 14–22. Springer, 2016.

[50] Adrien Gaidon and Eleonora Vig. Online domain adaptation for multi-object tracking. *arXiv preprint arXiv :1508.00776*, 2015.

[51] JianWen Tao, Shiting Wen, and Wenjun Hu. Robust domain adaptation image classification via sparse and low rank representation. *Journal of Visual Communication and Image Representation*, 33 :134–148, 2015.

[52] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(May) :985–1005, 2007.

[53] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114. ACM, 2004.

[54] Francis Vella. Estimating models with sample selection bias : a survey. *Journal of Human Resources*, pages 127–169, 1998.

[55] Marzieh Gheisari and Mahdieh Soleymani Baghshah. Unsupervised domain adaptation via representation learning and adaptive classifier learning. *Neurocomputing*, 165 :300–311, 2015.

[56] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013.

[57] Cheng-An Hou, Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Unsupervised domain adaptation with label and structural consistency. *IEEE Transactions on Image Processing*, 25(12) :5552–5562, 2016.

[58] Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and David W Aha. Unsupervised and transfer learning challenge. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 793–800. IEEE, 2011.

[59] Sinno Jialin Pan, James T Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682, 2008.

[60] Lixin Duan, Dong Xu, and Ivor Tsang. Learning with augmented features for heterogeneous domain adaptation. *arXiv preprint arXiv :1206.4660*, 2012.

[61] Chang Wang and Sridhar Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1541, 2011.

[62] Xiaoxiao Shi, Qi Liu, Wei Fan, and Philip S Yu. Transfer across completely different feature spaces via spectral embedding. *Knowledge and Data Engineering, IEEE Transactions on*, 25(4) :906–918, 2013.

[63] Qingyao Wu, Hanrui Wu, Xiaoming Zhou, Mingkui Tan, Yonghui Xu, Yuguang Yan, and Tianyong Hao. Online transfer learning with multiple homogeneous or heterogeneous sources. *IEEE Transactions on Knowledge and Data Engineering*, 29(7) :1494–1507, 2017.

[64] Wanchen Sui, Xinxiao Wu, Yang Feng, and Yunde Jia. Heterogeneous discriminant analysis for cross-view action recognition. *Neurocomputing*, 191 :286–295, 2016.

[65] Wenyuan Dai, Yuqiang Chen, Gui-Rong Xue, Qiang Yang, and Yong Yu. Translated learning : Transfer learning across different feature spaces. In *Advances in neural information processing systems*, pages 353–360, 2009.

[66] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get : Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1785–1792. IEEE, 2011.

[67] Jun Gao, Rong Huang, and Hanxiong Li. Sub-domain adaptation learning methodology. *Information Sciences*, 298 :237–256, 2015.

[68] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation : A survey of recent advances. *IEEE signal processing magazine*, 32(3) :53–69, 2015.

[69] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Subspace alignment for domain adaptation. *arXiv preprint arXiv :1409.5241*, 2014.

[70] Rui Xia, Chengqing Zong, Xuelei Hu, and Erik Cambria. Feature ensemble plus sample selection : domain adaptation for sentiment classification. *IEEE Intelligent Systems*, 28(3) :10–18, 2013.

[71] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2) :199–210, 2011.

[72] JianWen Tao, Dawei Song, Shiting Wen, and Wenjun Hu. Robust multi-source adaptation visual classification using supervised low-rank representation. *Pattern Recognition*, 61 :47–65, 2017.

[73] Mingsheng Long, Guiguang Ding, Jianmin Wang, Jiaguang Sun, Yuchen Guo, and Philip S. Yu. Transfer sparse coding for robust image representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[74] Jianhui Chen, Lei Tang, Jun Liu, and Jieping Ye. A convex formulation for learning shared structures from multiple tasks. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 137–144. ACM, 2009.

[75] Jianhui Chen, Lei Tang, Jun Liu, and Jieping Ye. A convex formulation for learning a shared predictive structure from multiple tasks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(5) :1025–1038, 2013.

[76] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in neural information processing systems*, pages 41–48, 2007.

[77] Luo Jie, Tatiana Tommasi, and Barbara Caputo. Multiclass transfer learning from unconstrained priors. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1863–1870. IEEE, 2011.

[78] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200. ACM, 2007.

[79] Joey Tianyi Zhou, Ivor W Tsang, Sinno Jialin Pan, and Mingkui Tan. Heterogeneous domain adaptation for multiple classes. In *Artificial Intelligence and Statistics*, pages 1095–1103, 2014.

[80] Arthur P Dempster. A high dimensional two sample significance test. *The Annals of Mathematical Statistics*, pages 995–1010, 1958.

[81] Zhidong Bai and Hewa Saranadasa. Effect of high dimension : by an example of a two sample problem. *Statistica Sinica*, pages 311–329, 1996.

[82] ZD Bai. Methodologies in spectral analysis of large dimensional random matrices, a review. *Statistica Sinica*, pages 611–662, 1999.

[83] Zhidong Bai and Jack W Silverstein. *Spectral analysis of large dimensional random matrices*, volume 20. Springer, 2010.

[84] Shurong Zheng et al. Central limit theorems for linear spectral statistics of large dimensional f-matrices. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 48, pages 444–476. Institut Henri Poincaré, 2012.

[85] John Wishart and AR Clapham. A study in sampling technique : the effect of artificial fortilisers on the yield of potatoes. *The Journal of Agricultural Science*, 19(4) :600–618, 1929.

[86] John Wishart. A problem in combinatorial analysis giving the distribution of certain moment statistics. *Proceedings of the London Mathematical Society*, 2(1) :309–321, 1929.

[87] Iain M Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Annals of statistics*, pages 295–327, 2001.

[88] Xavier Mestre. Improved estimation of eigenvalues and eigenvectors of covariance matrices using their sample estimates. *Information Theory, IEEE Transactions on*, 54(11) :5113–5129, 2008.

[89] Vladimir A Marčenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4) :457, 1967.

[90] Stuart Geman. A limit theorem for the norm of random matrices. *The Annals of Probability*, pages 252–261, 1980.

[91] Jack W Silverstein. The smallest eigenvalue of a large dimensional wishart matrix. *The Annals of Probability*, pages 1364–1368, 1985.

[92] Antonia M Tulino, Sergio Verdú, et al. Random matrix theory and wireless communications. *Foundations and Trends® in Communications and Information Theory*, 1(1) :1–182, 2004.

[93] Radoslaw Adamczak et al. On the marchenko-pastur and circular laws for some classes of random matrices with dependent entries. *Electronic Journal of Probability*, 16 :1065–1095, 2011.

[94] Terence Tao. *Topics in random matrix theory*, volume 132. American Mathematical Society Providence, RI, 2012.

[95] Wenhao Jiang and Fu-lai Chung. Transfer spectral clustering. In *Machine Learning and Knowledge Discovery in Databases*, pages 789–803. Springer, 2012.

[96] Colin Studholme, Derek LG Hill, and David J Hawkes. An overlap invariant entropy measure of 3d medical image alignment. *Pattern recognition*, 32(1) :71–86, 1999.

[97] Alain Biem, Shigeru Katagiri, and B-H Juang. Pattern recognition using discriminative feature extraction. *IEEE Transactions on Signal Processing*, 45(2) :500–504, 1997.

[98] Navin Goel, George Bebis, and Ara Nefian. Face recognition experiments with random projection. In *Defense and Security*, pages 426–437. International Society for Optics and Photonics, 2005.

[99] Serge Belongie, Charless Fowlkes, Fan Chung, and Jitendra Malik. Spectral partitioning with indefinite kernels using the nyström extension. In *European conference on computer vision*, pages 531–542. Springer, 2002.

[100] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nystrom method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2) :214–225, 2004.

[101] Ahmed H Sameh and John A Wisniewski. A trace minimization algorithm for the generalized eigenvalue problem. *SIAM Journal on Numerical Analysis*, 19(6) :1243–1259, 1982.

[102] Boyu Wang, Joelle Pineau, and Borja Balle. Multitask generalized eigenvalue program. In *AAAI*, pages 2115–2121, 2016.

[103] Roger A Horn. Cr johnson matrix analysis. *Cambridge UP, New York*, 1985.

[104] Paul Van Dooren. A generalized eigenvalue approach for solving riccati equations. *SIAM Journal on Scientific and Statistical Computing*, 2(2) :121–135, 1981.

[105] Deng Cai, Xiaofei He, Jiawei Han, et al. Isometric projection. In *AAAI*, pages 528–533, 2007.

[106] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in neural information processing systems*, pages 153–160, 2004.

[107] Ke Liu, Yong-Qing Cheng, Jing-Yu Yang, and Xiao Liu. An efficient algorithm for foley–sammon optimal set of discriminant vectors by algebraic method. *International Journal of Pattern Recognition and Artificial Intelligence*, 6(05) :817–829, 1992.

[108] Christos Boutsidis, Anastasios Zouzias, Michael W Mahoney, and Petros Drineas. Randomized dimensionality reduction for k-means clustering. *arXiv preprint arXiv :1110.2897*, 2011.

[109] Juwei Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. Face recognition using lda-based algorithms. *IEEE Transactions on Neural networks*, 14(1) :195–200, 2003.

[110] Kari Torkkola. Linear discriminant analysis in document classification. In *IEEE ICDM Workshop on Text Mining*, pages 800–806. Citeseer, 2001.

[111] SS Kim and MJ Vanderploeg. Qr decomposition for state space representation of constrained mechanical dynamic systems. *ASME J. Mech. Trans*, 108(2) :183–188, 1986.

[112] Aditya Krishna Menon and Charles Elkan. Fast algorithms for approximating the singular value decomposition. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2) :13, 2011.

[113] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[114] Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. Random projections for k-means clustering. In *Advances in Neural Information Processing Systems*, pages 298–306, 2010.

[115] Thierry Urruty, Chabane Djeraba, and Dan A Simovici. Clustering by random projections. In *Advances in Data Mining. Theoretical Aspects and Applications*, pages 107–119. Springer, 2007.

[116] Dimitris Achlioptas. Database-friendly random projections : Johnson-lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4) :671–687, 2003.

[117] Santosh S Vempala. *The random projection method*, volume 65. American Mathematical Soc., 2005.

[118] Genevieve Gorrell. Generalized hebbian algorithm for incremental singular value decomposition in natural language processing. In *EACL*, volume 6, pages 97–104, 2006.

[119] Robert J Durrant and Ata Kabán. Compressed fisher linear discriminant analysis : Classification of randomly projected data. In *Proceedings of the 16th ACM SIGKDD*

international conference on Knowledge discovery and data mining, pages 1119–1128. ACM, 2010.

[120] Robert J Durrant and Ata Kaban. Random projections as regularizers : learning a linear discriminant from fewer observations than dimensions. *Machine Learning*, 99(2) :257–286, 2015.

[121] Bojun Tu, Zhihua Zhang, Shusen Wang, and Hui Qian. Making fisher discriminant analysis scalable. In *International Conference on Machine Learning*, pages 964–972, 2014.

[122] Christos Boutsidis, Anastasios Zouzias, Michael W Mahoney, and Petros Drineas. Randomized dimensionality reduction for k-means clustering. *IEEE Transactions on Information Theory*, 61(2) :1045–1062, 2015.

[123] Hubert Mia, J. Rousseeuw Peter, and Verboven Sabine. A fast method for robust principal components with applications to chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 60 :101–111, 2002.

[124] Jinho Baik and Jack W Silverstein. Eigenvalues of large sample covariance matrices of spiked population models. *Journal of Multivariate Analysis*, 97(6) :1382–1408, 2006.

[125] Damien Passemier and Jian-Feng Yao. On determining the number of spikes in a high-dimensional spiked population model. *Random Matrices : Theory and Applications*, 1(01) :1150002, 2012.

[126] Shashanka Ubaru, Yousef Saad, and UMN EDU. Fast methods for estimating the numerical rank of large matrices. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 468–477, 2016.

[127] Jinho Baik and Jack W Silverstein. Eigenvalues of large sample covariance matrices of spiked population models. *Journal of Multivariate Analysis*, 97(6) :1382–1408, 2006.

[128] Damien Passemier and Jian-Feng Yao. On determining the number of spikes in a high-dimensional spiked population model. *Random Matrices : Theory and Applications*, 1(01) :1150002, 2012.

[129] P Raghavan and M Henzinger. Computing on data streams. In *Proc. DIMACS Workshop External Memory and Visualization*, volume 50, page 107, 1999.

[130] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[131] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3) :2758–2765, 2011.

[132] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10) :1345–1359, 2010.

[133] Rajhans Samdani and Wen-tau Yih. Domain adaptation with ensemble of feature groups. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1458, 2011.

[134] Daoqiang Zhang, Dinggang Shen, Alzheimer's Disease Neuroimaging Initiative, et al. Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in alzheimer's disease. *NeuroImage*, 59(2) :895–907, 2012.

[135] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[136] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide. Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7893–7897. IEEE, 2013.

[137] Hastie Trevor, Tibshirani Robert, and Friedman Jerome. The elements of statistical learning : data mining, inference and prediction. *New York : Springer-Verlag*, 1(8) :371–406, 2001.

[138] Yong-Qua Yin, Zhi-Dong Bai, and Pathak R Krishnaiah. On the limit of the largest eigenvalue of the large dimensional sample covariance matrix. *Probability theory and related fields*, 78(4) :509–521, 1988.

[139] Yu Zhang and Dit-Yan Yeung. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv :1203.3536*, 2012.

[140] Pascal Soucy and Guy W Mineau. A simple knn algorithm for text categorization. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 647–648. IEEE, 2001.

[141] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013.

[142] Nassara Elhadji Ille Gado, Edith Grall-Maës, Malika Kharouf, et al. Linear kernelpca and k-means clustering using new estimated eigenvectors of the sample covariance matrix. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 386–389. IEEE, 2015.

[143] Nassara Elhadji Ille Gado, Edith Grall-Maës, and Malika Kharouf. Linear discriminant analysis based on fast approximate svd. In *ICPRAM*, pages 359–365, 2017.

[144] Elhadji Ille Gado Nassara, Edith Grall-Maës, and Malika Kharouf. Linear discriminant analysis for large-scale data : Application on text and image data. In *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*, pages 961–964. IEEE, 2016.

[145] Cheng Wang, Guangming Pan, Tiejun Tong, and Lixing Zhu. Shrinkage estimation of large dimensional precision matrix using random matrix theory. *Statistica Sinica*, pages 993–1008, 2015.

[146] Shiliang Sun, Honglei Shi, and Yuanbin Wu. A survey of multi-source domain adaptation. *Information Fusion*, 24 :84–92, 2015.

Nassara ELHADJI ILLE GADO

Doctorat : Optimisation et Sûreté des Systèmes

Année 2017

Méthodes aléatoires pour l'apprentissage de données en grande dimension - Application à l'apprentissage partagé

Cette thèse porte sur l'étude de méthodes aléatoires pour l'apprentissage de données en grande dimension. Nous proposons d'abord une approche non supervisée consistant en l'estimation des composantes principales, lorsque la taille de l'échantillon et la dimension de l'observation tendent vers l'infini. Cette approche est basée sur les matrices aléatoires et utilise des estimateurs consistants de valeurs propres et vecteurs propres de la matrice de covariance. Ensuite, dans le cadre de l'apprentissage supervisé, nous proposons une approche qui consiste à, d'abord réduire la dimension grâce à une approximation de la matrice de données originale, et ensuite réaliser une LDA dans l'espace réduit. La réduction de dimension est basée sur l'approximation de matrices de rang faible par l'utilisation de matrices aléatoires. Un algorithme d'approximation rapide de la SVD, puis une version modifiée permettant l'approximation rapide par saut spectral sont développés. Les approches sont appliquées à des données réelles images et textes. Elles permettent, par rapport à d'autres méthodes, d'obtenir un taux d'erreur assez souvent optimal, avec un temps de calcul réduit. Enfin, dans le cadre de l'apprentissage par transfert, notre contribution consiste en l'utilisation de l'alignement des sous-espaces caractéristiques et l'approximation de matrices de rang faible par projections aléatoires. La méthode proposée est appliquée à des données de référence ; elle présente l'avantage d'être performante et adaptée à des données de grande dimension.

Mots clés : méthode de projection aléatoire - analyse en composantes principales - analyse discriminante - transfert d'apprentissage – algorithmes d'approximation.

Random Methods for Machine Learning of High Dimensional Data: Application to Transfer Learning

This thesis deals with the study of random methods for learning large-scale data. Firstly, we propose an unsupervised approach consisting in the estimation of the principal components, when the sample size and the observation dimension tend towards infinity. This approach is based on random matrices and uses consistent estimators of eigenvalues and eigenvectors of the covariance matrix. Then, in the case of supervised learning, we propose an approach which consists in reducing the dimension by an approximation of the original data matrix and then realizing LDA in the reduced space. Dimension reduction is based on low–rank approximation matrices by the use of random matrices. A fast approximation algorithm of the SVD and a modified version as fast approximation by spectral gap are developed. Experiments are done with real images and text data. Compared to other methods, the proposed approaches provide an error rate that is often optimal, with a small computation time. Finally, our contribution in transfer learning consists in the use of the subspace alignment and the low-rank approximation of matrices by random projections. The proposed method is applied to data derived from benchmark database; it has the advantage of being efficient and adapted to large-scale data.

Keywords: random projection methods - PCA - dimension reduction - SVD - LDA - transfer learning - approximation algorithms.

Thèse réalisée en partenariat entre :

Grand Est
ALSACE CHAMPAGNE-ARDENNE LORRAINE

utt
UNIVERSITÉ DE TECHNOLOGIE
TROYES

Ecole Doctorale "Sciences et Technologies"