

Sélection et importance de variables en apprentissage automatique

Alex Mourer

▶ **To cite this version:**

Alex Mourer. Sélection et importance de variables en apprentissage automatique : Applications à des données d'essai de moteurs d'avions très corrélées. Statistiques [math.ST]. Paris 1 - Panthéon-Sorbonne, 2022. Français. NNT : . tel-03842117

HAL Id: tel-03842117
https://hal.science/tel-03842117

Submitted on 7 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

THÈSE DE DOCTORAT

UNIVERSITÉ PARIS 1 PANTHÉON-SORBONNE
École Doctorale Sciences Mathématiques de Paris Centre (ED 386)
Laboratoire : Statistiques, Analyse et Modélisation Multidisciplinaire (EA 4543)

Présentée par

Alex Mourer

Pour obtenir le grade de Docteur de l'Université Paris 1 Panthéon-Sorbonne
Spécialité : Mathématiques Appliquées

Sélection et importance de variables en apprentissage automatique

Applications à des données d'essai de moteurs d'avions très corrélées

---

Sous la direction de Marie Chavent, Jérôme Lacaille et Madalina Olteanu

Jury de soutenance :

Mathilde Mougeot	(PR, Université Paris Saclay)	*Rapportrice*
Charles Bouveyron	(PR, Université Côte d'Azur)	*Rapporteur*
Marie Cottrell	(PR émérite, Université Paris 1)	*Examinatrice*
Alain Celisse	(PR, Université Paris 1)	*Examinateur*
Jean-Michel Poggi	(PR, Université Paris 5)	*Examinateur*
Marie Chavent	(PR, Université de Bordeaux)	*Directrice*
Madalina Olteanu	(PR, Université Paris Dauphine)	*Directrice*
Jérôme Lacaille	(HDR, Safran Aircraft Engines)	*Directeur*

Table des matières

1 Introduction 1
- 1.1 Les différents acteurs, le contexte et les enjeux . . . 1
- 1.2 Description des concepts industriels . . . 2
- 1.3 Les principaux défis . . . 6
- 1.4 Les principales contributions de ce travail de thèse . . . 7

I Sélection de variables, importance de variables et interprétabilité 14

2 Analyses, comparaisons et critiques des algorithmes de clustering sparse 15
- 2.1 Introduction . . . 15
- 2.2 Du clustering à l'algorithme des K-means . . . 16
- 2.3 Méthodes sparses basées sur les K-means . . . 18
- 2.4 Méthodes sparses basées sur les GMM . . . 24
- 2.5 Description des packages `R` existants . . . 31
- 2.6 Analyse des schémas et des résultats de simulations . . . 33
- 2.7 Simulations : comparaison des méthodes de clustering sparse . . . 36
- 2.8 Conclusion . . . 41

3 Clustering sur des groupes de variables via les K-means sparses et application à des données mixtes 43
- 3.1 Introduction . . . 43
- 3.2 Le modèle WT-K-means et son extension avec la pénalité *group lasso* . . . 44
- 3.3 Clustering sparse de données mixtes . . . 47
- 3.4 Illustration du package `vimpclust` sur des données réelles . . . 50
- 3.5 Simulations : comparaison des méthodes de clustering sparse sur des données mixtes . . . 51
- 3.6 Conclusion . . . 52
- 3.7 Annexe . . . 52

4 Clustering et données corrélées 56
- 4.1 Introduction . . . 56
- 4.2 État de l'art . . . 58
- 4.3 ACP et K-means . . . 59
- 4.4 La solution proposée : normaliser les variables en fonction des corrélations . . . 60
- 4.5 Simulations . . . 62
- 4.6 Conclusion . . . 67

5 Sélectionner le nombre de clusters K avec un compromis de stabilité : un indice de validation interne 70
- 5.1 Introduction . . . 70
- 5.2 État de l'art . . . 72
- 5.3 Stabilité en clustering . . . 72
- 5.4 Stabilité intra et inter-classes . . . 75
- 5.5 Exemples illustratifs . . . 77
- 5.6 Expériences : *benchmark* . . . 80
- 5.7 Conclusion . . . 81

6 Une nouvelle mesure d'importance de variables basée sur le clustering de variables pour les forêts aléatoires 85
6.1 Introduction . . . 85
6.2 Définitions, état de l'art et analyses . . . 86
6.3 Comparaison des méthodes existantes à l'aide d'un exemple simulé . . . 90
6.4 Méthodologie de la solution proposée . . . 92
6.5 Simulations . . . 95
6.6 Application industrielle : flottement *fan* . . . 97
6.7 Conclusion . . . 100

II Applications industrielles 101

7 Analyse et correction des données de bancs d'essais 102
7.1 Description des données et du problème . . . 102
7.2 Estimer et retirer la tendance de production . . . 106
7.3 Détection des biais bancs . . . 108
7.4 Correction des biais des équipements de bancs . . . 110
7.5 Vérification de la correction . . . 112
7.6 Conclusion . . . 114

8 Détection automatique d'observations rares pendant les essais de production à l'aide de modèles statistiques 117
8.1 Abstract . . . 118
8.2 Introduction . . . 118
8.3 Data Analysis . . . 119
8.4 Expert Knowledge to Define Anomalies . . . 120
8.5 Anomaly Detection . . . 121
8.6 Results on the Data . . . 122
8.7 Anomaly Categorization Using Self-Organizing Map . . . 124
8.8 Conclusion . . . 127

9 Conclusion 130

1

Introduction

1.1 Les différents acteurs, le contexte et les enjeux . . . 1
1.2 Description des concepts industriels . . . 2
1.2.1 Fonctionnement du moteur 2
1.2.2 Production et réception 3
1.2.2.a Essai de réception 4
1.2.2.b Vérification et certification d'un moteur 4
1.2.2.c Correction des équipements de bancs 5
1.3 Les principaux défis . . . 6
1.3.1 Les principaux défis industriels 6
1.3.2 Les principales caractéristiques des données 6
1.3.3 Les principaux défis théoriques 6
1.4 Les principales contributions de ce travail de thèse . . . 7

Mon travail de thèse CIFRE s'inscrit dans la continuité des thèses menées par Tsirizo Rabenoro (Rabenoro, 2015), Cynthia Faure (Faure, 2018), Florent Forest (Forest, 2021). L'objectif de cette thèse est de développer une méthodologie pour comprendre et mettre en évidence des typologies spécifiques du fonctionnement des moteurs d'avion lors de tests de réception effectués sur des bancs d'essai, et d'aider les ingénieurs métier de Safran Aircraft Engines dans l'analyse des résultats. Il s'agit, d'un point de vue statistique, de sélectionner les variables importantes et de calculer leur importance dans un cadre supervisé et non supervisé. Par ailleurs, cette approche doit rester cohérente avec les modèles physiques qui contrôlent les mécanismes et le fonctionnement du moteur.

1.1 Les différents acteurs, le contexte et les enjeux

Le travail présenté dans ce mémoire de thèse est le fruit d'une collaboration entre l'équipe de Statistique, Analyse et Modélisation Multidisciplinaire (SAMM) de l'Université Paris 1 Panthéon-Sorbonne, l'équipe Datalab de l'entreprise Safran Aircraft Engines et l'équipe ASTRAL (Méthodes avancées d'apprentissage statistique et de contrôle) de l'Inria Bordeaux Sud-Ouest.

L'équipe SAMM est une équipe d'accueil (EA 4543) et une des 3 composantes de la Fédération de Recherche CNRS FR2036 FP2M qui regroupe des mathématiciens et des informaticiens. Elle a été créée le 1er janvier 2010. Les domaines de recherche présents au sein du SAMM couvrent de nombreux champs des mathématiques appliquées (analyse fonctionnelle appliquée, apprentissage statistique, contrôle optimal, équations d'évolution, probabilités et statistique), et quelques thématiques en informatique (graphes, automates cellulaires).

Safran Aircraft Engines conçoit, développe, produit, et commercialise, seul ou en coopération, des moteurs pour avions civils, pour lanceurs spatiaux et pour satellites. Safran Aircraft Engines propose également aux compagnies aériennes et aux opérateurs d'avions une gamme complète de services pour leurs moteurs aéronautiques, couvrant le cycle de vie du moteur de son entrée en service jusqu'à son démantèlement. L'équipe

du Datalab possède des compétences en matière de statistiques, d'analyse de données et de génie logiciel et offre un support à de nombreux projets de l'entreprise, toujours en coopération avec les bureaux d'étude. La collaboration avec l'équipe SAMM date de 2008 et est très fructueuse (4 thèses et 2 postdocs).

Les activités de recherche de l'équipe ASTRAL se concentrent principalement sur le développement de méthodes statistiques et probabilistes avancées pour l'analyse et le contrôle de systèmes stochastiques complexes : modélisation statistique et stochastique, estimation et calibration, contrôle et décision.

Avant de présenter les défis industriels, les caractéristiques des données et les défis théoriques, il est nécessaire de décrire le fonctionnement d'un moteur et la méthodologie de certification pour autoriser la livraison au client.

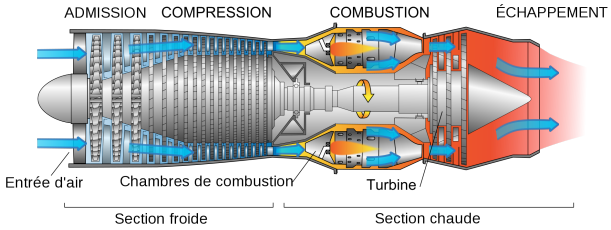
1.2 Description des concepts industriels

Une présentation brève du fonctionnement d'un turboréacteur, du rôle des aubes de soufflante, ainsi que des enjeux liés à leur conception et à leur fabrication permet de situer dans un contexte industriel la problématique de ce travail de thèse.

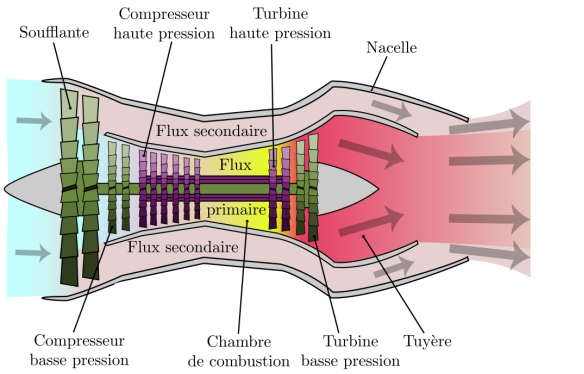
1.2.1 Fonctionnement du moteur

Les turboréacteurs équipent la majorité des avions civils assurant ainsi leur propulsion. Celle-ci est créée au moyen d'une force, appelée *poussée*, qui résulte de l'accélération d'une masse d'air en sens opposé au déplacement de l'avion.

Le turboréacteur doit produire une forte poussée en éjectant des gaz (mélange d'air et de combustibles) à une vitesse la plus élevée possible. On peut remarquer que cela est différent pour un turbopropulseur qui doit fournir une puissance certaine pour la mise en rotation d'une hélice, ce qui produit un déplacement d'air. Visuellement, comme illustré sur les schémas (a) et (b) de la Figure 1.1, le turboréacteur se distingue principalement par la présence d'une structure autour du compresseur (axial) composée d'une superposition de *roues aubagées* qui jouent le rôle d'hélices, dont la première est appelée soufflante ou *fan* en anglais (voir Figure 1.2 (a)). C'est sur ce type de moteur, notamment le modèle LEAP-1A, que portent les analyses de cette thèse.



(a) Turboréacteur (source*)



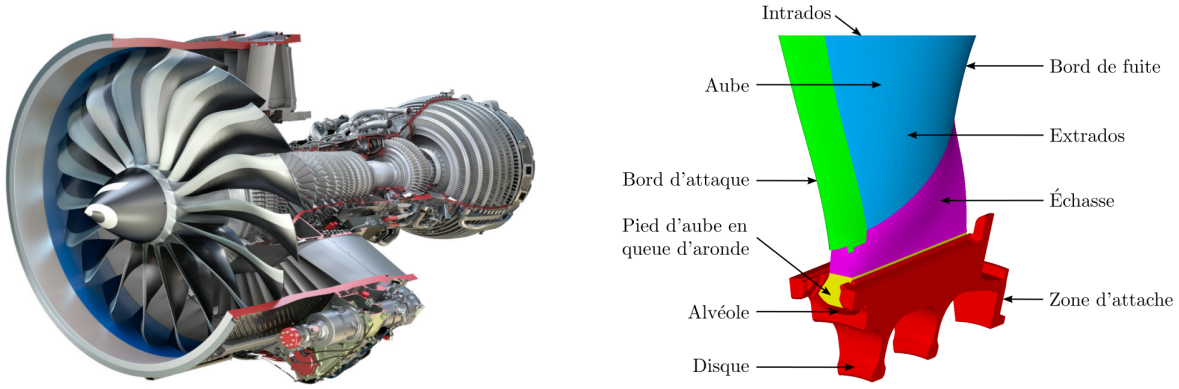
(b) Schéma double-corps double-flux (source :Mabilia (2020))

Figure 1.1 : Le graphique (a) représente un turboréacteur fonctionnant via une poussée de gaz éjecté et le graphique (b) donne un schéma d'un turboréacteur double corps double flux.

PRÉSENTATION DES TURBORÉACTEURS Un turboréacteur commence par aspirer un volume d'air, puis il le comprime et le chauffe avant que celui-ci ne s'échappe vers l'extérieur via une tuyère. Pour fournir une poussée, la vitesse d'éjection doit être supérieure à celle de l'admission. Ainsi, suivant le cycle *admission-compression-combustion-échappement* (voir Figure 1.1 (a)) : Le turboréacteur développe une poussée à l'aide de l'énergie qui est apportée par la combustion du carburant. Le flux d'air admis à l'avant du turboréacteur est d'abord accéléré puis comprimé par le compresseur et il est envoyé dans la chambre de combustion. Ensuite, l'air comprimé est mélangé à du fuel et le mélange est enflammé à l'intérieur de la chambre de combustion. Les gaz produits par la combustion sont éjectés dans la turbine et ensuite dans la *tuyère*, ce qui délivre une poussée.

*http ://www4.ac-nancy-metz.fr/ciras/cahierdubia/GTR/groupe_turbo_reacteur.html

Le cheminement du flux d'air à travers le turboréacteur peut être suivi sur les schémas de la Figure 1.1. On distingue deux parties importantes du moteur, la *nacelle* et la *tuyère*, qui sont des équipements présents dans les données étudiées dans ce manuscrit.



(a) LEAP-1A (source : Mabilia (2020))

(b) Secteur *fan* (source : Mabilia (2020))

Figure 1.2 : Le graphique (a) donne une vue en coupe du moteur LEAP-1A et de sa roue aubagée et le graphique (b) donne une description d'un secteur *fan*.

Le *fan* est le permier étage du compresseur axial de illustré sur la Figure 1.2 (a) et chaque aube du *fan* est appelée secteur *fan* (Figure 1.2 (b)). Un *secteur fan* est composé d'un bord d'attaque et d'un bord de fuite reliés ensemble par le milieu de corde ou *extrados* comme illustré sur la Figure 1.2 (b)). Les roues aubagées sont composées d'aubes qui sont en rotation et qui déplacent un grand volume d'air en accélérant le flux d'air. Les turboréacteurs séparent le flux d'air en deux flux distincts après le franchissement de la roue aubagée : le flux primaire et le flux secondaire. Le flux primaire effectue le parcours décrit ci-dessus dans la Figure 1.1 (a) et le flux secondaire circule autour du moteur tout en restant contenu dans la nacelle (voir Figure 1.1 (b)), subissant alors une accélération.

1.2.2 Production et réception

Les tests de réception de moteurs sont obligatoires avant toute mise en service. Ils sont importants tant pour l'avionneur que pour les équipes de Safran Aircraft Engines. Pour ces dernières, ils permettent d'avoir une compréhension complète du comportement de chaque moteur. Ces tests produisent plusieurs milliers de mesures et leur analyse est une tâche essentielle et difficile.

Ce travail est réalisé par l'équipe performance de Safran Aircraft Engines, au sein du département Performance et Opérabilité, dont les ingénieurs sont responsables de la préparation, du suivi et de l'analyse initiale de la performance ou de l'opérabilité des moteurs, en particulier du LEAP-1A. À ce titre, les principales missions qui leur sont confiées sont les suivantes :

- la définition des tests pour répondre aux besoins de vérification des performances (rédaction des demandes de tests et d'instrumentation) ;
- la participation à la préparation de la séquence de test avec les équipes chargées de le réaliser (prédiction du comportement du moteur, adaptation des lois de conduite) ;
- la vérification que le test a atteint ses objectifs (respect des instructions, des procédures, vérification du fonctionnement de l'instrumentation) ;
- le suivi des équipements de bancs d'essai ;
- le soutien aux équipes d'étude pour l'exploitation des résultats.

Dans ce manuscrit, nous nous sommes concentrés sur les deux derniers points avec l'aide d'un ingénieur performance, Alexandre Vasseur. L'objectif principal est de fournir des méthodes statistiques qui aideront les experts dans l'analyse, la vérification et l'exploitation des résultats des bancs d'essai.

Ces tests sont réalisés sur un ensemble de moteurs identiques sous des hypothèses de conditions de fonctionnement représentatives des conditions de fonctionnement réelles. Malheureusement, les moteurs testés ne fonctionnent pas tous sous des conditions équivalentes. Premièrement, il faut prendre en compte les variabilités

propres de la production. Ensuite, ils sont également soumis à des différences de conditions météorologiques. Les variabilités de conception font l'identité du moteur et ne sont pas corrigées. En revanche, les variabilités liées aux conditions météorologiques et atmosphériques sont corrigées à l'aide d'un modèle thermodynamique (Meqqadmi et al., 2017). Une fois les données corrigées, il existe encore des dispersions importantes qui sont dues aux équipements utilisés pour faire ces tests.

1.2.2.a Essai de réception

La production des moteurs de série est certifiée par un essai de réception défini dans des documents contractuels. Cet essai comprend deux phases principales :

1. la vérification du comportement mécanique, notamment en faisant un rodage stabilisé, un équilibrage des niveaux vibratoires et un rodage transitoire (voir Figure 1.3) ;
2. la vérification des performances et des limites fixées dans le contrat.

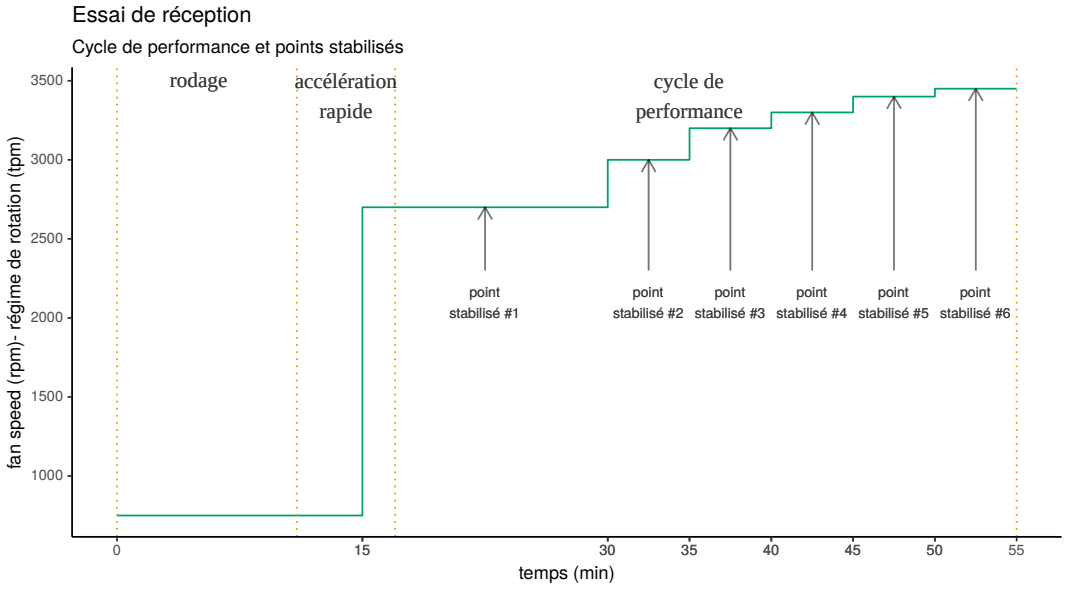


Figure 1.3 : Le graphique schématise les étapes d'un essai de réception. L'opérateur fixe un régime de rotation pendant une certaine durée et au cours de cette phase à régime constant que l'on appelle *point stabilisé*, différentes mesures sont effectuées.

En cas d'anomalie, le moteur n'est pas livré au client et il est admis en *chaîne hôpital* afin d'être analysé, opération qui est très coûteuse. Une fois le problème traité, le moteur retourne en essai de production.

1.2.2.b Vérification et certification d'un moteur

Les performances du moteur en essai ne sont pas comparables directement car elles dépendent :

1. des conditions ambiantes du jour de l'essai, c'est-à-dire de la pression atmosphérique, de la température de l'air admis par le moteur, de l'humidité de l'air et de la condensation ;
2. de l'environnement du moteur en essai et notamment des systèmes et des équipements nécessaires, tels que le banc d'essai sur lequel est attaché le moteur, la nacelle d'avionnage, la buse d'entrée d'air, la tuyère posée sur le moteur.

Les fluctuations sur les performances du moteur en test de réception ont plusieurs origines et grossièrement il est admis que 30% seulement sont liées directement aux aléas de production des moteurs. En outre, pour certifier les performances d'un moteur, il est indispensable de les ramener à des conditions connues de fonctionnement. La méthode de normalisation utilisée est la suivante :

- on corrige les résultats issus de l'essai avec des coefficients de correction qui dépendent des conditions ambiantes et des différences entre l'environnement d'essai et l'environnement d'utilisation ;
- on corrige les résultats par rapport à un fonctionnement (régime et poussée) de référence.

Ces corrections sont appliquées sur les paramètres tels que la poussée, le débit carburant, les régimes de rotation, la température des gaz d'éjection (ou *exhausting gaz temperature* EGT), etc.

1.2.2.c Correction des équipements de bancs

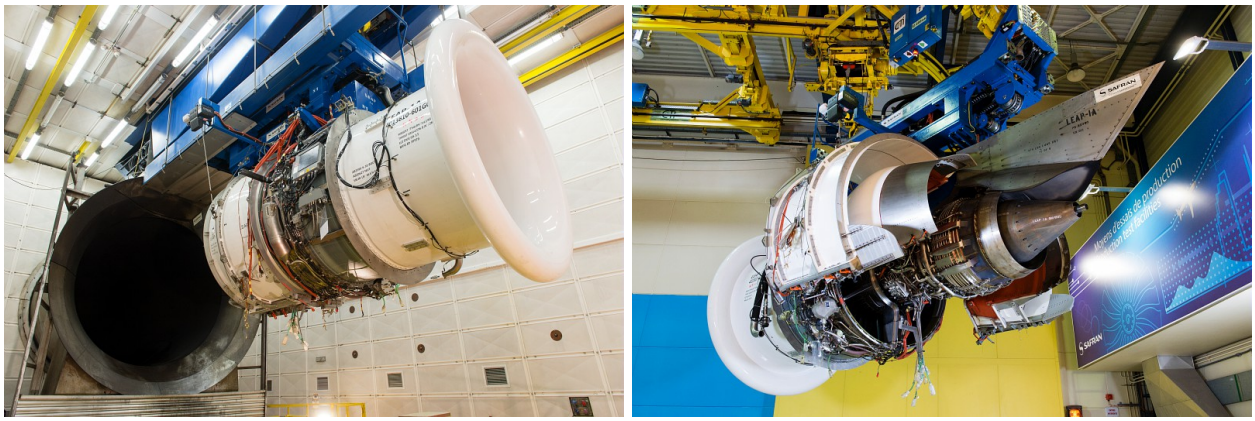
CORRECTION DE BANCS D'ESSAI Pour éviter la pollution sonore, les essais de moteur se font sur banc fermé comme illustré sur la Figure 1.4 (a). Le bruit est alors considérablement réduit par l'utilisation d'un tunnel conçu à cet effet. Malheureusement la poussée du moteur n'est plus du tout la même que lorsqu'il est à l'air libre, en banc d' essai ou en vol. En effet, cette isolation acoustique entraîne une diminution du volume d'air à absorber disponible pour le moteur et à débit de carburant identique le moteur produit moins de poussée en banc fermé. Ainsi les résultats du test doivent être corrigés par calcul pour obtenir la poussée réelle du moteur. Les correctifs apportés sont de l'ordre de 3% à 10% suivant les installations (les débits d'air à l'entrée des moteurs peuvent varier de 80 kg/s à 1600kg/s).

Un coefficient (additif) est alors déterminé pour chaque banc d'essai fermé. Il est calculé en faisant, pour un moteur de référence aussi appelé *moteur étalon*, la différence des performances obtenues sur le banc fermé étudié et sur un banc d'essai à l'air libre pour un même moteur. Ce coefficient est ensuite appliqué à chaque moteur de série qui est testé sur ce banc d'essai fermé.

CORRECTION DE NACELLES Les nacelles utilisées pour les essais de production produisent, au même titre que les bancs, des écarts dans les mesures observées entre les différentes nacelles utilisées en tests, notamment celle utilisées lors de l'étalonnage en banc d'essai à l'air libre. Les raisons de la présence de ces écarts sont différentes de celles qu'on observe sur les bancs, mais un coefficient de correction à estimer doit également être appliqué aux résultats d'essais de chaque moteur de série.

CORRECTION D'INSTRUMENTATION D'ESSAIS Les installations de mesure en essais induisent des écarts dans les résultats des tests qui doivent aussi être corrigés pour obtenir les performances *réelles* du moteur, ou des performances conformes à ce qui est attendu. Les corrections à appliquer sur les résultats d'essais sont déterminées par calcul à partir d'un modèle moteur dont on simule le fonctionnement avec et sans instrumentation. Dans ces instrumentations, la buse d'air et la tuyère primaire sont prises en compte.

DÉRIVE DE PERFORMANCES DES ÉQUIPEMENTS DE BANCS D'ESSAI Il existe une *dérive* lente des équipements d'essai qui peut être due à une usure. Il est nécessaire de déterminer un coefficient de correction dépendant du temps et du travail fourni par les équipements. En effet, l'usure est principalement due aux températures élevées auxquelles sont soumis les équipements, et donc la correction prend en compte les températures auxquelles sont exposés les équipements ainsi que le carburant utilisé. Même dans un cas d'utilisation stable des équipements, il peut exister des tendances non linéaires des mesures qui sont effectuées sur le moteur. Lorsque la dérive, que l'on appelle aussi *biais*, est trop importante, les équipements subissent une opération de maintenance. Une fois cette opération effectuée, il convient de corriger à nouveau les mesures en estimant un coefficient comme expliqué dans les paragraphes ci-dessus. Ces opérations sont coûteuses en temps, en personnel et en matériel, elles impliquent des organisations complexes de la chaîne de production et le résultat, en termes de correction, n'est pas forcément satisfaisant. Une analyse et une correction à l'aide d'outils issus de la statistique peuvent être plus appropriées.



(a) Banc fermé (source : Safran Aircraft Engines) (b) Nacelle et tuyère (source : Safran Aircraft Engines)

Figure 1.4 : Le graphique (a) représente le moteur placé en banc fermé et le graphique (b) représente une nacelle et une tuyère posées sur le moteur.

1.3 Les principaux défis

Dans cette section, nous abordons les principaux défis rencontrés dans cette thèse. Nous commençons par introduire les différents sujets auxquels nous sommes confrontés, puis décrivons les principales caractéristiques des données à notre disposition, ce qui permet enfin de motiver et de présenter les défis théoriques à résoudre.

1.3.1 Les principaux défis industriels

Ces travaux de thèse portent sur l'analyse de données d'essais de réception qui sont indispensables pour la vérification et la certification des moteurs. Les résultats de ces essais dépendent en grande partie des équipements avec lesquels ces essais sont faits. Ainsi, un des premiers objectifs est de comprendre statistiquement l'influence de ces équipements et de vérifier que la correction calculée à partir de modèles physiques permet de comparer les populations de moteurs, c'est-à-dire que les résultats corrigés des essais sont indépendants des équipements avec lesquels ils ont été effectués. Si ce n'est pas le cas, il faudra proposer une correction statistique des mesures faites sur les moteurs.

Ensuite, une fois les mesures corrigées obtenues et validées, un deuxième objectif est de les analyser pour vérifier qu'elles sont bien conformes aux exigences attendues. D'un point de vue physique, cela se fait à l'aide d'un modèle moteur, et les mesures réelles doivent être conformes aux intervalles de fonctionnement prévus par le modèle. Le problème peut être approché d'un point de vue statistique en utilisant des algorithmes de détection d'anomalies. Les paramètres à l'origine de ces anomalies peuvent être déterminés à l'aide de méthodes de calcul d'*importance de variables*.

Enfin, un dernier sujet est abordé, celui des phénomènes de vibration des aubes *fan*. De manière générale, les aubes sont soumises à diverses sources d'excitation et d'amortissement, naturelles ou intentionnelles, qu'il s'agit de connaître précisément en vue d'éviter les phénomènes de résonance et les risques d'endommagement dont le plus important est la fatigue vibratoire. Ce point sera détaillé dans le chapitre abordant cette question.

1.3.2 Les principales caractéristiques des données

Les données de bancs sont des données très structurées. Des mesures de multiples paramètres, fournissant souvent des informations assez similaires, sont effectuées suivant 6 niveaux de poussée du moteur que l'on appelle des points stabilisés dont le régime de rotation est fixé par l'opérateur. En outre, deux variables représentant un même paramètre mesuré en deux points stabilisés différents sont très corrélées. Les données de bancs ont donc par nature une structure en groupes, où les groupes sont formés par les variables représentant un même paramètre mesuré sur les 6 points stabilisés. Pour ces données, le nombre d'observations (le nombre de moteurs) est relativement faible (de l'ordre de 500 lors de l'acquisition des données en début de thèse) et il est supérieur au nombre de variables dans un ratio de 1 à 5.

Par ailleurs, on suspecte que les observations représentant les moteurs dans les données, sont partitionnées selon l'environnement et la configuration des tests, notamment selon l'instrumentation et les équipements utilisés.

Les données décrivant les phénomènes de vibration des aubes *fan* sont de grande dimension, c'est-à-dire que le nombre d'observations est très faible (de l'ordre de 100) et inférieur au nombre de variables (dans un ratio de 1 à 10). De plus, ces données sont extrêmement corrélées. En effet, le phénomène à étudier est complexe et l'une des premières approches métier a été de multiplier le nombre de paramètres mesurés pour tenter de le comprendre, donnant des blocs de variables très corrélées dont la structure est a priori inconnue (car physiquement on ne connaît pas la loi liant exactement les paramètres).

Finalement, les principales caractéristiques des données sont :

- un faible nombre d'observations ;
- un grand nombre de variables les décrivant, dont une partie n'est pas utile pour décrire les phénomènes physiques ;
- de fortes corrélations entre les variables ;
- des structures par blocs ou par groupes entre les variables.

1.3.3 Les principaux défis théoriques

Du point de vue théorique, l'enjeu principal de ce travail de thèse est de modéliser et expliquer les phénomènes physiques observés et non observés à l'aide de méthodes statistiques et d'en interpréter la cause à l'aide d'une (petite) partie des variables explicatives. Ce travail se place donc dans le cadre supervisé et non supervisé,

avec une contrainte forte, celle de construire des modèles interprétables : on cherche à indiquer la contribution de chaque variable à la construction du modèle, c'est ce que l'on appelle mesurer l'*importance des variables*. En outre, on cherche à améliorer l'interprétabilité en construisant des modèles parcimonieux ou *sparses* en anglais, c'est-à-dire tels que les variables qui ne contribuent pas à leur construction en soient exclues.

Dans le cadre non supervisé, des méthodes de partitionnement de groupes d'observations, ou méthodes de *clustering* en anglais, vont être étudiées. La solution que nous cherchons doit pouvoir s'utiliser en grande dimension, doit être interprétable et elle doit pouvoir tenir compte de la structure de groupes des variables. Ce type de méthode est connu sous le nom de méthodes de clustering sparse (clustering et sparse ne sont plus écrits en italique dans la suite du manuscrit et ce sont les seules exceptions à la règle typographique qui met en évidence les mots étrangers).

Dans le cadre supervisé, nous proposons des méthodes modélisant un phénomène décrit par des variables continues (respectivement catégorielles), c'est-à-dire des méthodes de régression (respectivement de classification). On impose les mêmes contraintes que dans le cas non supervisé (gérer des données de grande dimension, être sparse, indiquer l'importance des variables, tenir en compte des structures de groupes de variables). Par ailleurs, remarquons que nous voulons modéliser un phénomène physique dans le but d'en expliquer les principes et donc que nous ne sommes pas (directement) intéressés par les aspects prédictifs. Ce point est extrêmement important car nous verrons qu'expliquer un phénomène et expliquer la prédiction du modèle associé sont deux buts qui peuvent être contradictoires.

Une caractéristique propre aux algorithmes sparses est qu'ils dépendent d'un paramètre à ajuster (au même titre que les méthodes de clustering). Différentes valeurs de ce paramètre donnent naissance à différents modèles et il est nécessaire de choisir entre tous ces modèles. Ainsi, il est primordial de disposer d'une méthode de sélection de modèle efficace. Il faut insister sur le fait que la sélection de modèle est un défi majeur en clustering non supervisé. En effet, il n'existe pas de méthode universellement admise pour évaluer les résultats du clustering pour la raison évidente qu'il n'y a pas de vérité de terrain par rapport à laquelle les résultats pourraient être comparés. C'est aussi le cas lorsque l'on s'intéresse à la sélection de variables (modèles sparses) et à l'importance de variables, même dans le cadre supervisé, car on ne dispose jamais du *vrai* ensemble de variables et des *vraies* importances de variables définissant le phénomène sous-jacent étudié.

Pour résumer, les principaux défis théoriques sont :

- gérer des données de grande dimension ;
- procéder à une sélection des variables dans un but explicatif (et non prédictif par exemple) ;
- déterminer l'importance des variables (dans un but explicatif) ;
- prendre en compte une structure de groupes lors de la modélisation ;
- gérer les corrélations entre les variables pour répondre aux tâches indiquées ci-dessus ;
- sélectionner le modèle le plus représentatif du phénomène étudié.

1.4 Les principales contributions de ce travail de thèse

La suite du manuscrit est organisée en sept chapitres, dont les cinq premiers présentent des contributions théoriques au domaine des statistiques.

Le Chapitre 2 est un état de l'art et une analyse des algorithmes existants de clustering sparse. La contribution essentielle de ce chapitre est de proposer une étude détaillée et critique d'un certain nombre des algorithmes de clustering permettant de faire de la sélection de variables, en présentant notamment des méthodes basées sur les K-means (Lloyd, 1982) ou les modèles de mélange gaussien (McLachlan and Basford, 1988 ; McLachlan and Krishnan, 2007 ; McLachlan et al., 2019).

Le Chapitre 3 introduit une méthode de clustering sparse adaptée aux variables ayant une structure de groupes. Notre contribution (M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu), décrite dans Chavent et al. (2020), consiste à proposer une extension de l'algorithme du WT-K-means (Witten and Tibshirani, 2010), que nous appellerons Group-Sparse K-means, à des données numériques structurées en groupes de variables, en introduisant une pénalité dite *group lasso* (Yuan and Lin, 2006).

Le Chapitre 4 étudie le problème que posent les corrélations entre les variables notamment dans le contexte du clustering sparse. Nous remarquons que la présence de corrélations entre les variables de bruit impacte négativement les performances des algorithmes de clustering basés sur la distance euclidienne, nous expliquons le problème en pratique tout en donnant des intuitions théoriques et proposons une solution de normalisation basée sur l'inverse de la somme des corrélations au carré des variables deux à deux.

Le Chapitre 5 présente une méthode de sélection de modèle en clustering. Notre contribution, décrite dans Mourer et al. (2020b), consiste à proposer une nouvelle définition du clustering, basée sur la notion de stabilité (Von Luxburg, 2010) entre les clusters et au sein des clusters et nous introduisons un critère de différence de stabilité, *Stadion*, un indice interne de validation du clustering.

Le Chapitre 6 introduit une méthode permettant de calculer l'importance des variables en présence de groupes inconnus de variables très corrélées, dans le cadre de l'apprentissage supervisé et cette contribution est publiée dans Chavent et al. (2021b) (M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu). Dans le cas des modèles de forêts aléatoires (Breiman, 2001), nous expliquons le problème que pose la présence de corrélations entre les variables explicatives pour calculer l'importance des variables, et nous proposons une méthode pour réduire ces corrélations en entrée de l'algorithme qui est basée sur le clustering de variables (Chavent et al., 2011, 2021a).

Les deux derniers chapitres présentent des applications industrielles.

Le Chapitre 7 définit une méthodologie de détection, de correction et de vérification des biais introduits par les équipements de bancs lors des tests de réception. L'algorithme de détection et de vérification se fonde sur une méthodologie d'apprentissage supervisé, où un modèle tente de prédire l'équipement de bancs utilisé à partir des mesures qui y sont faites. Ce travail est un enjeu important car les résultats de ces tests permettent la certification des moteurs.

Le Chapitre 8 expose une méthode de détection d'anomalies dans les données de tests de réception. La méthodologie étend l'utilisation des valeurs de Shapley (Shapley, 1953) aux modèles d'*Isolation Forest* Liu et al. (2008). Ce chapitre a fait l'objet d'une publication dans Mourer et al. (2020b).

Ce manuscrit est accompagné de trois logiciels libres d'accès : le package `R vimpclust`* publié sur le `CRAN` implémente la méthode présentée au Chapitre 3 (et bientôt celle du Chapitre 4) ; le code `python skstab`† implémente la méthode du Chapitre 5 (la version `R` est déjà implémentée et elle sera mise en ligne très prochainement) ; le code `R SMDA`‡ implémente la méthode du Chapitre 6.

Les données et les codes permettant de traiter spécifiquement les applications du Chapitre 7 et du Chapitre 8 sont soumises à des contraintes de confidentialité. Pour le reste, le code (excepté pour le Chapitre 5) permettant de fournir les résultats, les figures, les tableaux de ce manuscrit est disponible en accès libre dans un projet§ `R`, partitionné par chapitre, où chaque fichier contient un programme qui délivre le résultat d'une figure ou d'un tableau du manuscrit.

*https ://cran.r-project.org/web/packages/vimpclust/index.html
†https ://github.com/FlorentF9/skstab
‡https ://github.com/MourerAlex/SMDA
§https ://github.com/MourerAlex/Thesis

Articles publiés et en cours de soumission

Articles publiés

[1] M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. Handling correlations in random forests : which impacts on variable importance and model interpretability ? In M. Verleysen, editor, *29th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN) : October 6-8, 2021*, pages 569–574, Online event, 2021b. European Symposium on Artificial Neural Networks (ESANN), i6doc.com (**Chapitre** 6)

[2] M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. Sparse k-means for mixed data via group-sparse clustering. In M. Verleysen, editor, *28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN) : October 2-4, 2020*, pages 235–240, Online event, 2020. European Symposium on Artificial Neural Networks (ESANN), i6doc.com (**Chapitre** 3)

[3] A. Mourer, J. Lacaille, M. Olteanu, and M. Chavent. Automatic detection of rare observations during production tests using statistical models. In *Annual Conference of the PHM Society*. PHM, 2020b (**Chapitre** 8)

Article en cours de soumission

A. Mourer, F. Forest, M. Lebbah, H. Azzag, and J. Lacaille. Selecting the number of clusters k with a stability trade-off : an internal validation criterion. *arXiv preprint arXiv :2006.08530*, 2020a (**Chapitre** 5)

Articles en cours de rédaction

M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. Handling correlations in clustering : ought the variables to be standardised not only by their variance but also by their correlations ? a (**Chapitre** 4)

M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. A new variable importance measure based on variable clustering for random forests. b (**Chapitre** 6 - [1] en version étendue)

M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. Review of sparse clustering methods. c (**Chapitre** 2)

Brevet prévu

M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. Detect, correct and verify the presence of bias in reception test data. d (**Chapitre** 7)

Logiciels libres d'accès

package `R vimpclust`* (Variable Importance in Clustering) `CRAN`

code `R SMDA`† (**Chapitre** 6)

code `python skstab`‡ (**Chapitre** 5 - version `R` disponible prochainement)

*https ://cran.r-project.org/web/packages/vimpclust/index.html

†https ://github.com/MourerAlex/SMDA

‡https ://github.com/FlorentF9/skstab

Préambule : évaluation des méthodes de clustering et des méthodes de sélection de modèles en clustering

DÉFINITION DU CLUSTERING Le clustering est l'une des tâches les plus courantes en apprentissage non supervisé. Elle consiste à mettre en évidence des groupes d'observations (également appelés *clusters* ou *classes*) dans un jeu de données. Pour l'analyse exploratoire de données, cela permet de mieux comprendre la structure d'un jeu de données et peut également être utilisé pour la classification. Le clustering peut être défini comme le partitionnement des données en groupes (ou clusters) de sorte que les éléments similaires (au sens d'une fonction de distance sous-jacente) partagent le même cluster et que tous les membres de chaque cluster soient similaires (ou, de manière équivalente, que les éléments dissimilaires soient séparés en clusters différents) (Ben-David, 2018). C'est un objectif très difficile notamment en raison de la non-transitivité de la notion de similarité : si A est similaire à B, et B est similaire à C, A n'est pas nécessairement similaire à C. Nous verrons plus en détail dans les chapitres suivant les défis qui se posent en clustering.

COMMENT ÉVALUER LES MÉTHODES DE CLUSTERING Pour évaluer une méthode de clustering, il est d'usage dans la littérature de comparer la partition obtenue avec une partition sous-jacente que l'on appelle la *vraie partition*. Ainsi, évaluer la qualité d'un clustering est équivalent à mesurer le taux d'erreur en apprentissage supervisé. Néanmoins, cela nécessite d'avoir connaissance de la vraie partition, ce qui n'est généralement pas le cas pour les données réelles. En fait, les méthodes de clustering ne peuvent être évaluées que sur des jeux de données simulées. Pour illustrer certains aspects d'une méthode, les jeux de données réelles sont indispensables, mais les ensembles de données simulées sont les seuls qui garantissent une évaluation des performances fiable. Ce point sera discuté précisément dans les paragraphes suivants.

Généralement en clustering, le nombre de clusters à trouver est inconnu et il faut l'estimer conjointement à la partition. Certains algorithmes tels que les algorithmes de clustering hiérarchiques Ward (1963), permettent d'estimer le nombre de clusters, d'autres utilisent des critères de pénalité fondés statistiquement tels que le BIC (Schwarz, 1978 ; Lebarbier and Mary-Huard, 2006) pour les modèles de mélange gaussien (GMM). Enfin, pour une grande partie des algorithmes dont l'algorithme des K-means, le choix du nombre de clusters reste une question difficile et ouverte. C'est pourquoi, lorsqu'un algorithme de clustering est couplé à une méthode fournissant une estimation du nombre de classes, ce qui sera majoritairement le cas pour les algorithmes que nous verrons dans le chapitre suivant, il est possible de les évaluer dans le même temps.

Néanmoins, cela ne permet pas d'étudier les performances de l'algorithme de clustering seul. Une bonne pratique dans ce cas est de fixer le nombre de clusters, pour évaluer l'algorithme de clustering de manière indépendante. Une deuxième évaluation suivra pour évaluer la méthodologie globale.

Supposons alors que nous connaissions la vraie partition, ce qui est le cas lorsque l'on simule des données, reste à définir une mesure pour comparer deux partitions. Si le nombre de classes des deux partitions est le même, le taux de bon classement peut être utilisé. Si le nombre de classes est grand ou si les classes sont déséquilibrées, d'autre mesures plus adéquates ont été proposées (Romano et al., 2015). Dans le cas où le nombre de classes n'est pas le même, plusieurs mesures ont été définies pour évaluer la performance de l'algorithme. Par exemple, l'*Adjusted Rand Index* (ARI) défini par Hubert and Arabie (1985b) est selon Romano et al. (2015) un choix recommandé lorsque les clusters sont majoritairement équilibrés.

Définissons formellement l'ARI, qui est largement utilisé dans ce manuscrit. Soit deux partitions $\mathcal{C}_K$ et $\mathcal{C}'_{K'}$. En considérant toutes les paires d'observations deux à deux nous pouvons définir les quatre quantités suivantes :

1. N_{11} le nombre de paires qui sont dans le même cluster dans les deux partitions $\mathcal{C}_K$ et $\mathcal{C}'_{K'}$;
2. N_{00} le nombre de paires dans des clusters différents dans les deux partitions $\mathcal{C}_K$ et $\mathcal{C}'_{K'}$;
3. N_{10} le nombre de paires dans le même cluster dans $\mathcal{C}_K$ mais pas dans $\mathcal{C}'_{K'}$;
4. N_{01} le nombre de paires dans le même cluster dans $\mathcal{C}_{K'}$ mais pas dans $\mathcal{C}'_K$.

L'*Indice de Rand Ajusté* (ARI) est une version de de l'indice de Rand qui est corrigée par la valeur attendue de l'indice. Il a été introduit sous deux variantes différentes dans Morey and Agresti (1984) et Hubert and Arabie (1985a). Dans ce travail, nous utilisons la première version de Hubert and Arabie (1985a) qui s'exprime comme suit :

$$\text{ARI} = \frac{2(N_{00}N_{11} - N_{01}N_{10})}{(N_{00} + N_{01})(N_{01} + N_{11}) - (N_{00} + N_{10})(N_{10} + N_{11})}. \tag{1.1}$$

L'ARI est toujours compris entre -1 et 1, il vaut 1 quand les partitions sont identiques et il est proche de 0 ou inférieur à 0 lorsque les accords entre les deux partitions sont aléatoires.

Si l'on cherche à évaluer plusieurs méthodes sur différents ensembles de données, calculer un ARI moyenné peut être trompeur. En effet, les ensembles de données peuvent être de difficultés différentes et, par conséquent, une simple moyenne peut induire en erreur (Demšar, 2006). La moyenne fait sens pour agréger les résultats d'une même simulation qui a été répétée, mais pas pour agréger des résultats sur différents ensembles de données.

Afin de comparer les méthodes sur plusieurs bases de données de complexité et de difficulté différentes, il est possible de faire des tests de rangs avec le test de rangs de Wilcoxon (Demšar, 2006)) ou de calculer une performance normalisée par la difficulté de l'ensemble des données (Hofmeyr, 2018), pour autant qu'on puisse l'estimer.

LE PROBLÈME DE L'ÉVALUATION SUR DES DONNÉES RÉELLES Pour utiliser une *vraie partition*, il est nécessaire de s'assurer qu'elle représente bien exactement les clusters présents dans les données. Cela nécessite une analyse approfondie du jeu de données et, quand bien même celle-ci est faite, cela ne garantit pas que la partition des données soit conforme à la réalité. En effet, il peut toujours exister des variables latentes non observées qui induisent des structures de clusters dans les données, provoquant un décalage entre la variable désignant la *vraie partition* et les vrais clusters sous-jacents. Pour autant que nous le sachions, aucune méthodologie ou analyse n'a été suggérée pour garantir avec certitude qu'un jeu de données réelles soit partitionné en K classes.

En général, les algorithmes de clustering tels que les K-means sont souvent testés sur des ensembles de données réelles et les clusters obtenus comparés à une variable catégorielle désignée comme la *vraie partition*. Il se peut que l'algorithme réussisse à retrouver les clusters définis par la *vraie partition* et que l'ARI soit proche de 1. Par exemple, c'est le cas pour la base de données sur les maladies cardiaques HDdata[§], pour lequel l'algorithme des K-means exécuté sur les variables numériques retrouvent les clusters définis par la variable qui code la présence ou l'absence de maladie cardiaque. On sait que les classes sont quasiment linéairement séparables sur la première composante principale. Cependant, comme on peut le voir sur la Figure 1.5(a), la frontière entre les classes est très dense et les classes peuvent représenter une distribution unimodale plutôt que bimodale, ce qui soulève des questions quant à l'existence de 2 clusters distincts.

Pour résumer, sur des données réelles, rien ne peut garantir qu'une variable donnée ou définie par l'utilisateur représente la vraie partition si toutefois celle-ci existe et en aucun cas le fait qu'un algorithme retrouve un partitionnement défini arbitrairement est la preuve que ce partitionnement représente la vraie partition.

COMMENT ÉVALUER LES MÉTHODES DE SÉLECTION DE MODÈLE POUR LE CLUSTERING On doit distinguer l'évaluation d'une méthode de clustering pour un nombre de classes fixé, de l'évaluation de la qualité du modèle, c'est-à-dire de l'adéquation du nombre de classes choisi avec la réalité.

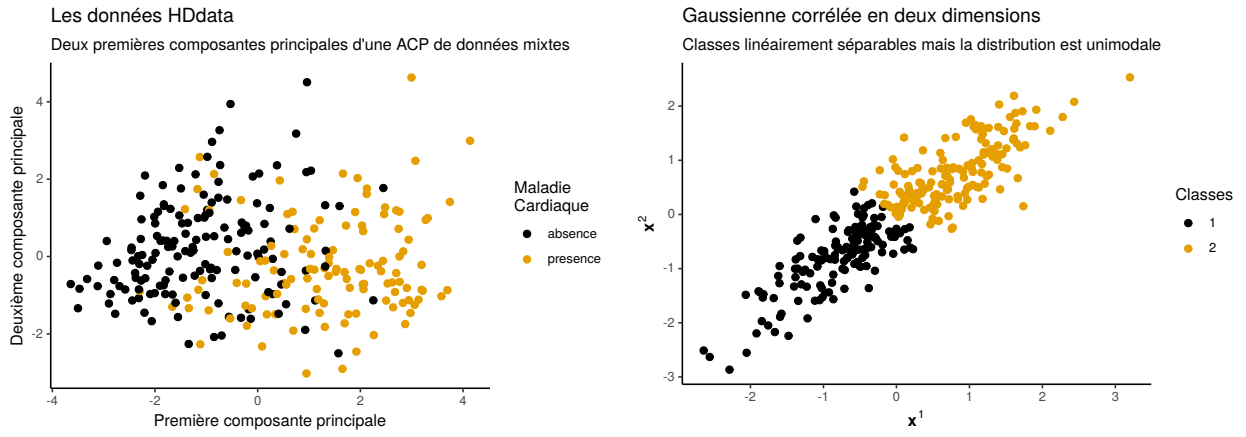
Par exemple, une distribution gaussienne unimodale non sphérique peut être divisée en deux en son centre pour former artificiellement deux classes comme c'est le cas sur la Figure 1.5(b). Dans ce cas, la partition trouvée par l'algorithme des 2-means sera la meilleure solution au sens de l'ARI, mais ne sera pas un *bon* modèle, puisqu'en réalité il n'y a pas lieu de considérer deux clusters.

Il faut donc être très prudent lors de la simulation de données destinées à évaluer les méthodes de sélection de modèles. Dans les expériences, nous éviterons au maximum le chevauchement entre les clusters en générant des clusters séparés, ce qui donnera des scénarios plus simples mais fiables.

RÉSUMÉ Pour résumer, trois points sont importants à retenir :

1. Pour évaluer des méthodes de clustering ou des méthodes de sélection de modèles en clustering, il ne suffit pas de savoir que l'on a trouvé le *bon* nombre de clusters, il faut comparer les partitions obtenues avec une mesure de dissimilarité. Pour des simulations répétées de données de difficultés comparables, les résultats peuvent être moyennés, sinon il faudra utiliser par exemple des tests de rangs.

[§]https ://archive.ics.uci.edu/ml/datasets/statlog+(heart)



(a) (b)

Figure 1.5 : Le graphique (a) représente les classes définies par la variable indiquant la présence ou l'absence d'une maladie cardiaque sur la première et deuxième composantes principales d'une ACP (sur données numériques uniquement). Le graphique (b) illustre des classes artificielles définies arbitrairement sur un jeu de données gaussien de dimension deux. Sur ces deux exemples l'algorithme des K-means a des très bons résultats.

2. L'évaluation des méthodes de clustering ou des méthodes de sélection de modèle nécessite de disposer de bases de données qui sont effectivement structurées en clusters. Même si cela parait évident, ce n'est pas si simple, car il n'y a pas de définition concrète de ce qu'est un cluster. C'est pourquoi dans la suite nous utiliserons des jeux de données simulées et des vérifications visuelles de clusters simulés.

3. Sur des données réelles, rien ne ne peut garantir qu'une variable donnée ou définie par un utilisateur représente la vraie partition si toutefois celle-ci existe et en aucun cas le fait qu'un algorithme retrouve un partitionnement défini arbitrairement n'est la preuve que celui-ci représente les clusters, les seuls clusters, la *vraie partition*.

Première partie

Sélection de variables, importance de variables et interprétabilité

2

Analyses, comparaisons et critiques des algorithmes de clustering sparse

2.1 Introduction . . . 15
2.2 Du clustering à l'algorithme des K-means . . . 16
2.2.1 Critères d'inertie 16
2.2.2 Algorithme des K-means 17
2.3 Méthodes sparses basées sur les K-means . . . 18
2.3.1 WT-K-means (1) 18
2.3.2 Choix du paramètre λ 18
2.3.3 Autres algorithmes sparses : extensions du WT-K-means 19
2.3.4 K-means sparses par *hill climbing* 21
2.3.5 Regularized K-means : pénalisation par les centres 22
2.3.6 Lasso-Weighted K-means 22
2.3.7 Résumé des travaux sur les K-means sparses 23
2.4 Méthodes sparses basées sur les GMM . . . 24
2.4.1 Les modèles de mélange gaussien 24
2.4.2 Méthodes par pénalisation de la vraisemblance 25
2.4.3 Méthodes par pénalisation des *loadings* (5) 26
2.4.4 Sélection de variables via la sélection de modèles 27
2.4.5 VSCC : La méthode de Jeffrey L. Andrews et Paul D. McNicholas (9) 30
2.4.6 Résumé 31
2.5 Description des packages `R` existants . . . 31
2.6 Analyse des schémas et des résultats de simulations . . . 33
2.6.1 Analyse des schémas de simulations 33
2.6.2 Analyse des résultats de simulations dans la littérature 36
2.7 Simulations : comparaison des méthodes de clustering sparse . . . 36
2.7.1 Résultats scénario 1 : $K = 2; n = 120; p_K = 2; d = 20; m = 1.5$ 38
2.7.2 Résultats scénario 2 : $K = 2; n = 120; p_K = 10; d = 100; m = 0.85$ 38
2.7.3 Conclusion sur les simulations 41
2.8 Conclusion . . . 41

2.1 Introduction

Il est légitime de penser que s'il existe une structure de clustering dans des ensembles de données réelles, les clusters sous-jacents ne diffèrent que selon certaines variables. Les ensembles de données réelles contiennent souvent des variables indépendantes et non informatives pour le clustering, que nous appelons variables de

bruit, et celles-ci peuvent être en très grand nombre. Considérer explicitement ces variables comme du bruit dans le modèle permet tout d'abord d'améliorer les performances de l'algorithme et ensuite de réduire le nombre de variables du modèle, qui sera ainsi plus facile à interpréter.

Malheureusement, le clustering (ou classification non supervisée) est encore aujourd'hui une tâche difficile, alors le clustering sparse peut paraître quant à lui inabordable. Déjà il y a plus de 20 ans, Gnanadesikan et al. (1995) écrivaient "*One of the thorniest aspects of cluster analysis continue to be the weighting and selection of variables*" et plus récemment Raftery and Dean (2006) commentaient que "*Less work has been done on variable selection for clustering than for classification, perhaps reflecting the fact that the former is a harder problem. In particular, variable selection and dimension reduction in the context of model-based clustering have not received much attention*". Les difficultés tant dans la formulation que dans la mise en oeuvre, font que les algorithmes de clustering sparse restent encore peu utilisés comparativement aux algorithmes de clustering standards et cela est surprenant au vu de la popularité des méthodes sparses dans le contexte supervisé. Néanmoins, nous allons voir que de nombreuses méthodes déjà exploitables existent et offrent des solutions pertinentes.

CONTRIBUTIONS DE CE CHAPITRE La contribution essentielle de ce chapitre est de proposer une étude détaillée et critique d'un certain nombre des algorithmes de clustering permettant de faire de la sélection de variables, en présentant notamment des méthodes basées sur les K-means (Lloyd, 1982) ou les modèles de mélange gaussiens (GMM) (McLachlan and Basford, 1988 ; McLachlan and Krishnan, 2007 ; McLachlan et al., 2019).

Il existe deux études comparatives intéressantes dans ce domaine : celles de Bouveyron and Brunet-Saumard (2014a) et de Fop and Murphy (2018). Ces études sont orientées plus spécifiquement sur des méthodes basées sur les GMM. Dans Bouveyron and Brunet-Saumard (2014a), on trouve une description d'algorithmes sparses et non sparses pour des données de grande dimension et les algorithmes sont illustrés sur des données réelles. Dans Fop and Murphy (2018), les auteurs ajoutent une partie sur l'analyse des classes latentes qui sont des modèles de mélange permettant de faire du clustering sur des observations décrites le plus souvent par des variables catégorielles. Ils illustrent les méthodes sur des ensembles de données réelles et font une analyse comparative des temps de calcul des algorithmes.

Notre apport réside donc dans le fait d'étendre ce type d'étude aux algorithmes sparses basés sur les K-means en plus des GMM et de proposer une analyse comparative numérique détaillée des algorithmes dont le code est disponible librement, en s'appuyant sur des schémas de simulations structurés et des indicateurs précis.

2.2 Du clustering à l'algorithme des K-means

Cette section est consacrée à des rappels sur les critères d'inertie et l'algorithme des K-means. Ces points sont nécessaires tant pour la compréhension que pour la formulation des méthodes décrites dans la suite.

2.2.1 Critères d'inertie

En clustering, on cherche à construire des classes les plus homogènes possible tout en étant les plus séparées possible au sens de la distance euclidienne. Soit la matrice $\mathbf{X} \in \mathbb{R}^{n\times p}$ de n observations $\mathbf{x}_i$ décrites par p variables $\mathbf{x}^j$. D'après le théorème de König-Huygens, pour une partition en K classes des n observations $\{C_1, ..., C_K\}$, la somme pondérée des carrés totale (somme des carrés des distances euclidiennes des observations au centre des données) se décompose en la somme pondérée des carrés inter-classes et la somme pondérée des carrés intra-classes. De cette décomposition on peut donc déduire la relation suivante :

$$\underbrace{\sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} (x_i^j - \bar{x}^j)^2}_{t_j} = \underbrace{\sum_{j=1}^{p} \frac{1}{n} \sum_{k=1}^{K} \sum_{i\in C_k} (x_i^j - \bar{x}_k^j)^2}_{v_j} + \underbrace{\sum_{j=1}^{p} \frac{1}{n} \sum_{k=1}^{K} n_k (\bar{x}_k^j - \bar{x}^j)^2}_{b_j}, \tag{2.1}$$

où

- $\bar{x}^j$ la moyenne de la variable j ;
- $\bar{\mathbf{x}} = (\bar{x}^1, ..., \bar{x}^p)^\intercal$ est le centre des données ;
- n_k le nombre d'observations dans le cluster k ;
- $\bar{x}_k^j = \frac{1}{n_k} \sum_{i\in C_k} x_i^j$;

- $\bar{\mathbf{x}}_k = (\bar{x}_k^1, ..., \bar{x}_k^p)^\intercal$ est le centre du cluster k.

Les notations t_j, v_j, b_j définies dans l'Équation (2.1) sont respectivement la variance totale, la variance intra-classes et la variance inter-classes de la variable j. La variance totale étant une quantité fixe indépendante des classes de la partition, minimiser la variance intra, c'est-à-dire rendre les classes homogènes, est équivalent à maximiser la variance inter, c'est-à-dire séparer les classes comme illustré sur la Figure 2.1. Cette remarque

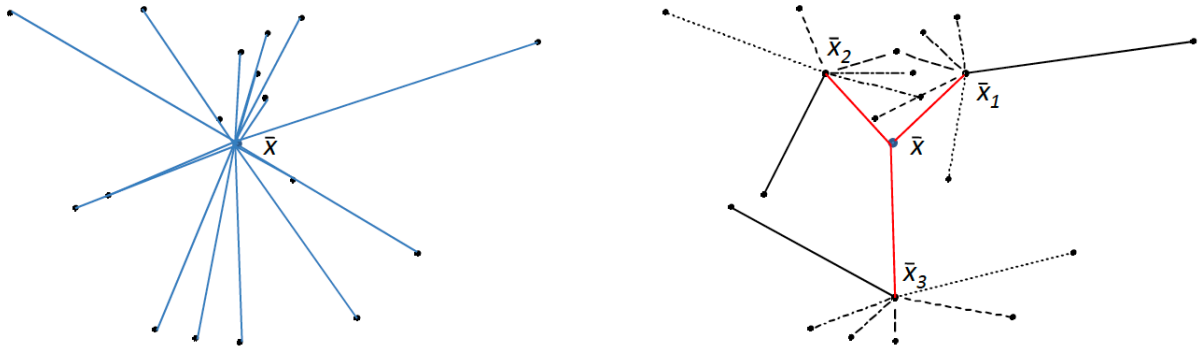


Figure 2.1 : Le graphique illustre l'Équation 2.1.

est à l'origine de la définition de l'algorithme le plus utilisé en clustering, connu comme algorithme des centres mobiles, K-moyennes ou K-means.

2.2.2 Algorithme des K-means

Comme son nom l'indique, l'algorithme des K-means (Forgy, 1965 ; MacQueen et al., 1967 ; Hartigan and Wong, 1979 ; Lloyd, 1982) se définit en fixant a priori le nombre de clusters K à construire. Le but est de minimiser la variance intra-classes ou de manière équivalente maximiser la variance inter-classes :

$$\underset{C_1,...,C_K}{\text{minimiser}} \sum_{j=1}^{p} v_j \iff \underset{C_1,...,C_K}{\text{maximiser}} \sum_{j=1}^{p} b_j. \tag{2.2}$$

L'Équation 2.2 définit un problème non convexe qui est donc difficile à optimiser. La stratégie mise en place est un algorithme itératif, fonctionnant par étapes successives (Lloyd, 1982). L'algorithme commence par initialiser les K centres notés $\mathbf{c}_1, ..., \mathbf{c}_K$, en choisissant par exemple des observations tirées aléatoirement et alterne ensuite les deux étapes suivantes jusqu'à la convergence de l'algorithme.

1. **Affectation** : chaque observation est assignée au centre le plus proche,

$$C_k \leftarrow \{i \mid \underset{k'=1,...,K}{\text{argmin}} \; ||\mathbf{x}_i - \mathbf{c}_{k'}||_2^2 = k\}.$$

2. **Minimisation** : chaque centre est mis à jour pour devenir la moyenne de son cluster,

$$\mathbf{c}_k \leftarrow \frac{1}{n_k} \sum_{i \in C_k} \mathbf{x}_i.$$

Du fait de sa non convexité, on ne peut empêcher que l'algorithme converge la plupart du temps vers un minimum local. C'est pour cela qu'il est recommandé de relancer plusieurs fois l'algorithme avec des initialisations différentes et de choisir la meilleure solution au sens de L'Équation 2.2. D'autres types d'initialisation sont aussi possibles, tel que le K-means++ (Arthur and Vassilvitskii, 2007). L'intuition pour cette approche est qu'il faut choisir les centres initiaux les plus distants possible : le premier centre de cluster est choisi uniformément au hasard parmi les observations, après quoi chaque centre est choisi parmi les points de données restants avec une probabilité proportionnelle à sa distance au carré du centre de cluster le plus proche.

VERSIONS NON SPARSES DES K-MEANS AVEC PONDÉRATION DES VARIABLES Des auteurs ont introduit des variantes de la méthode des K-means en introduisant des pondérations des variables (Friedman and Meulman, 2004 ; Huang et al., 2005 ; Jing et al., 2007 ; Domeniconi et al., 2007). Le but est d'améliorer les performances du clustering en différenciant les contributions, inconnues a priori, des différentes variables importantes ou de bruit. Ces méthodes ne conduisent pas à sélectionner des variables, mais elles sont à l'origine de certaines des méthodes K-means sparses.

2.3 Méthodes sparses basées sur les K-means

Les méthodes de clustering sparse basées sur les K-means ont connu un développement très important à la suite des travaux de Daniella Witten et Robert Tibishirani publiés en 2010 (Witten and Tibshirani, 2010). Dans cette section, nous introduisons cet algorithme qui a une place centrale dans ce manuscrit de thèse.

2.3.1 WT-K-means (1)

Le principe général de l'algorithme de Witten and Tibshirani (2010) est de réécrire le problème d'optimisation des K-means en introduisant une pondération des variables (poids) et une contrainte basée sur la norme ℓ_1 de ces poids pour obtenir de la sparsité en forçant des poids à valoir 0, ce qui exclut les variables correspondantes du modèle. On nomme cet algorithme WT-K-means et il fait partie de la famille des algorithmes K-means sparses. Précisément l'algorithme repose sur l'introduction d'un nouveau critère de variance, la variance pondérée et pénalisée. À partir de L'Équation (2.2), on pondère les variances inter-classes par variable, pour prendre en compte la contribution des variables à la séparabilité des classes en clustering, et on ajoute une pénalité dépendant de ces poids pour supprimer des variables du modèle. La variance inter-classes pondérée et pénalisée s'écrit alors :

$$\sum_{j=1}^{p} w_j b_j - \lambda h(\mathbf{w}) = \mathbf{w}^\intercal \mathbf{b} - \lambda h(\mathbf{w}) \ , \tag{2.3}$$

où

- $\mathbf{w} = (w_1, ..., w_p)^\intercal$ est le vecteur des poids,
- avec $w_j \geqslant 0, \forall j \in \{1, ..., p\}$ et $\|\mathbf{w}\|_2^2 \leqslant 1$,
- $\mathbf{b} = (b_1, ..., b_p)^\intercal$ est le vecteur des variances inter-classes par variable pour une partition donnée $C_1, ..., C_K$,
- $\lambda \geqslant 0$ est l'hyperparamètre déterminant l'intensité de la pénalisation,
- $h(\mathbf{w})$ une fonction de pénalité.

Dans ce qui suit, K, le nombre de clusters, ainsi que λ sont supposés être fixés à l'avance. Dans l'algorithme WT-K-means introduit dans Witten and Tibshirani (2010), le terme de pénalité est choisi par analogie avec le cadre *lasso* (*least absolute shrinkage and selection operator*) (Tibshirani, 1996), et donc $h(\mathbf{w}) = \|\mathbf{w}\|_1$. Optimiser la version pondérée et pénalisée de la variance inter-classes revient à trouver des poids et des clusters. Le problème peut alors se réécrire sous la forme d'une double optimisation :

$$\underset{C_1,...,C_K,\mathbf{w}}{\text{maximiser}} \ \mathbf{w}^\intercal \mathbf{b} - \lambda \|\mathbf{w}\|_1 \ \text{ s.c. } \ \|\mathbf{w}\|_2^2 \leqslant 1, w_j \geqslant 0 \ \forall j. \tag{2.4}$$

Ainsi, lorsque l'optimisation de l'Équation (2.4) se fait par rapport à $\mathbf{w}$, d'une part les poids traduisent directement la contribution des variables au clustering et d'autre part l'on obtient une représentation sparse des variables. En effet, la pénalité ℓ_1 permet la mise à 0 de certains poids ce qui implique que les variables correspondantes n'influent plus sur l'inertie inter-classes et donc n'ont plus de pouvoir discriminatif. On remarque que, lorsque $\lambda = 0$ l'algorithme ne se réduit pas aux K-means standard mais à des K-means avec pondération car l'optimisation des poids et le fait que les variances inter-classes des variables sont le plus souvent différentes implique des résultats différents des K-means standards. Néanmoins si $\lambda = 0$ l'algorithme n'effectue pas de sélection de variable.

2.3.2 Choix du paramètre λ

Le paramètre λ peut être choisi à l'aide de critères d'évaluation de qualité du modèle. La plupart des critères d'évaluation en clustering portent sur le choix du nombre de clusters K. Ici, le paramètre λ détermine le sous-espace dans lequel les centres des clusters résident. Les deux tâches sont très différentes et les outils qui fonctionnent pour l'une ne vont pas obligatoirement fonctionner pour l'autre.

GAP STATISTIC Dans leurs travaux, Witten and Tibshirani (2010) utilisent le Gap Statistic (Tibshirani et al., 2001b) pour choisir le paramètre λ. Le Gap Statistic est une méthode utilisée pour déterminer le nombre de clusters dans un ensemble de données. Elle consiste à comparer la variance inter-classes observée à son espérance sous une distribution de référence.

Il existe plusieurs possibilités pour obtenir une distribution de référence, celle qui est préconisée par les auteurs consiste à permuter indépendamment les observations de chaque variable (Tibshirani et al., 2001b). Ce processus de permutation crée de nouvelles variables qui sont des copies indépendantes des variables originales. Le processus de permutation est répété D fois. Enfin, l'algorithme WT-K-means est appliqué à chacun de ces nouveaux ensembles de données $\{\mathbf{X}^1, ..., \mathbf{X}^D\}$ pour des valeurs fixes de (K, λ).

Notons $b_j(K, \lambda, \mathbf{X}^d)$ la variance inter-classes du clustering obtenu par le WT-K-means sur $\mathbf{X}^d$ avec les paramètres (K, λ) et $b_j(K, \lambda, \mathbf{X})$ la variance inter-classes du clustering obtenu sur l'ensemble de données original $\mathbf{X}$ pour les mêmes paramètres. Le Gap Statistic est défini comme suit :

$$Gap(K, \lambda, \mathbf{X}) = b_j(K, \lambda, \mathbf{X}) - \frac{1}{D}\sum_{d=1}^{D} \log(b_j(K, \lambda, \mathbf{X}^d)). \tag{2.5}$$

La valeur optimale $\lambda^\star$ est obtenue lorsque cette quantité est maximum :

$$\lambda^\star = \underset{\lambda}{\operatorname{argmax}}(Gap(K, \lambda, \mathbf{X})).$$

Il est possible de choisir $\lambda^\star$ dans l'Équation (2.5) autrement qu'en choisissant le maximum (Tibshirani et al., 2001b), mais ces alternatives ne sont pas présentées.

Dans Witten and Tibshirani (2010), le Gap Statistic est utilisé pour choisir le paramètre λ, mais les auteurs en nuancent les performances "*the performance of the gap statistic of Section 3.2 for selecting the tuning parameter is mixed. This is not surprising, since tuning parameter selection in the unsupervised setting is known to be a very difficult problem. This is an area in which more work is needed.*". Dans cet article, le choix de nombre K n'est pas abordé.

DÉTECTION DE RUPTURE Bien que ce chapitre soit un état de l'art (et une analyse) des méthodes de clustering sparse existantes, nous introduisons ici une contribution originale pour le choix du paramètre λ. En effet, des simulations numériques sont faites en fin de chapitre et nous souhaitons inclure la méthode que nous allons décrire dans les comparaisons, sans multiplier les tableaux de résultats par la suite.

Un des buts du WT-K-means est d'expliquer au maximum la variance des données à l'aide du partitionnement tout en réduisant au minimum le nombre de variables nécessaires. On cherche donc à observer l'évolution de $\sum_{j=1}^{p} b_j$, la variance inter-classes non pondérée, en fonction des valeurs du paramètre λ et c'est ce que l'on appelle le *chemin de variance expliquée* (voir Chapitre 3 Section 3.4 pour des exemples). Plus λ est grand, plus le nombre de variables utilisées dans le modèle est petit, et par définition plus la variance expliquée diminue.

Nous cherchons à trouver une *rupture* sur le *chemin de variance expliquée*, c'est-à-dire la plus grande différence de variance inter-classes pour deux valeurs consécutives du paramètre, λ_i et λ_{i+1} et nous choisissons λ_i comme valeur optimale $\lambda^\star$.

Pour trouver cette rupture nous utilisons la méthode `cpt.meanvar` du package `R changepoint`. Celle-ci se base sur la méthode *At Most One Changepoint* (Hinkley, 1970).

Il nous reste à approfondir toutes les propriétés de cette contribution mais d'ores et déjà nous avons observé qu'elle permet de mettre en évidence l'importance du choix du paramètre λ dans les simulations, et que les résultats semblent prometteurs.

Nous aurons, par la suite, une ample discussion sur le choix des paramètres, l'initialisation et la normalisation, qui font l'ipséité stricte de l'algorithme mais qui sont aussi communs à toute la famille des algorithmes K-means sparse.

2.3.3 Autres algorithmes sparses : extensions du WT-K-means

WT-K-MEANS ET DONNÉES ABERRANTES (2) Plusieurs extensions du WT-K-means ont été proposées. Historiquement, l'une des premières méthodes consiste à rendre l'algorithme robuste aux données extrêmes ou aberrantes et a été développée par Kondo et al. (2012). Ils proposent d'utiliser l'algorithme Trimmed K-means (Cuesta-Albertos et al., 1997) dans la procédure du WT-K-means. L'idée principale du Trimmed K-means est de remplacer l'étape d'affectation des observations dans les K-means par une nouvelle étape :

1. **Affectation*** : Après avoir affecté les observations aux clusters, on les ordonne selon leur distance au centre de leur cluster, on retire (*we trim*) un pourcentage α des observations les plus éloignées de leur centre, et on met à jour les centres des clusters sans tenir compte des observations retirées.

À chaque étape les observations retirées peuvent être différentes. Une première approche naïve pour rendre robuste le WT-K-means consiste donc à utiliser l'algorithme Trimmed K-means à la place de l'algorithme original (K-means). Finalement Kondo et al. (2012) proposent une modification de l'algorithme Trimmed-K-means pour améliorer encore les performances. L'idée est qu'à partir de la partition trouvée à chaque itération par le WT-K-means, on détermine deux ensembles d'observations à retirer, le premier à partir du sous-espace pondéré par les poids du WT-K-means et le second à partir de l'espace original. En procédant ainsi, les auteurs assurent que les poids nuls ne masqueront pas nécessairement les valeurs aberrantes dans les variables de bruit. Cette propriété peut être très utile lorsque l'on cherche à nettoyer l'ensemble de données en retirant les observations aberrantes.

Une autre extension est proposée par Brodinová et al. (2019) avec une méthodologie plus complexe. Toujours dans l'idée de faire du clustering sparse en présence de données aberrantes, les auteurs reprennent le schéma méthodologique de Kondo et al. (2012) et proposent de l'améliorer en utilisant l'algorithme ROBIN (3) (ROBust INitialization) de Al Hasan et al. (2009) pour rendre l'initialisation indépendantes aux valeurs aberrantes. De plus, dorénavant les valeurs aberrantes des deux ensembles d'observations à retirer calculés sur les données pondérées et non pondérées, sont identifiées à l'aide de l'algorithme Local Outlier Factor (LOF) (Breunig et al., 2000), appliqué indépendamment sur chaque cluster. Les auteurs indiquent que la procédure proposée est conçue de manière à ce que les paramètres requis soient sélectionnés automatiquement et que notamment cela permet de se défaire de l'hyperparamètre α. Il faut noter que LOF requiert malgré tout deux nouveaux hyperparamètres. Ces nouveaux hyperparamètres sont fixés par défaut suivant les recommandations des articles de références (Breunig et al., 2000 ; Al Hasan et al., 2009).

WT-K-MEANS ET GROUPES DE VARIABLES (4) Huo and Tseng (2017) ont proposé une extension des K-means sparses au cas où les variables qui décrivent les données sont structurées en groupe. Ils reprennent l'algorithme WT-K-means en introduisant une pénalité dite *overlapping group* (Jacob et al., 2009), qui permet le chevauchement des groupes tout en sparsifiant au sein des groupes. La combinaison de la pénalité lasso de norme ℓ_1 et de la pénalité *overlapping group* sert à faire de la sélection de variables et encourage également (mais ne force pas) les variables du même groupe à être sélectionnées ensemble. Formellement, Le problème d'optimisation s'écrit comme suit :

$$\underset{C_1,\dots,C_K,\mathbf{w}}{\text{maximiser}}\ \mathbf{w}^\intercal\mathbf{b} - \lambda\Big[\alpha\left\|\mathbf{w}\right\|_1 + (1-\alpha)g(\mathbf{w})\Big]\ \text{s.c.}\ \left\|\mathbf{w}\right\|_2^2 \leqslant 1, w_j \geqslant 0\ \forall j, \tag{2.6}$$

où

- $\alpha \in [0,1]$ est un terme contrôlant l'équilibre entre la pénalité mettant à 0 des poids dans les groupes ($\left\|\mathbf{w}\right\|_1$) et la pénalité inter-groupes ($g(\mathbf{w})$),
- $g(\mathbf{w}) = \sum_{l=1}^{L} \upsilon_l \left\|\mathbf{s}_l \circ \mathbf{w}\right\|_2$ est la pénalité *overlapping group* où $\circ$ est le produit terme à terme des deux vecteurs (produit d'Hadamard),
- L le nombre de groupes variables,
- $\mathbf{s}_l = (\mathbf{s}_{l,1}, ..., \mathbf{s}_{l,p})^\intercal$ est le vecteur binaire d'appartenance au group l,
- $\upsilon_l \geqslant 0$ est un poids associé à chaque groupe de variables.

Si $\alpha = 1$, il n'y a pas de terme de pénalité de groupes et la fonction objectif est équivalente à celle du WT-K-means et si $\alpha = 0$, seule la pénalité *overlapping group* subsiste. En outre, chaque variable appartient à un ou plusieurs groupes, éventuellement seule dans son groupe. Les auteurs préconisent de fixer $\upsilon_l \geqslant 0$ égal au nombre de variables dans le groupe l, avec coefficient 1 si la variable n'appartient qu'à ce groupe (comme dans Yuan and Lin (2006)) et $\frac{1}{q}$ si la variable appartient à q groupes.

La difficulté avec la pénalisation *overlapping group* est la méthode d'optimisation correspondante pour trouver les poids w_j. En effet, ce type de pénalisation ne permet pas en général d'avoir une solution explicite comme c'est le cas avec le WT-K-means. Ainsi, Huo and Tseng (2017) décident d'utiliser la méthode *Alternating Direction Method of Multipliers* (ADMM) (Boyd et al., 2011) pour résoudre (2.6). Cette méthode d'optimisation demande beaucoup d'itérations pour déterminer une solution avec précision, et elle a aussi besoin d'un hyperparamètre supplémentaire, en plus de K et des deux paramètres de pénalisation α et λ, pour contrôler sa convergence (pour trouver les poids).

En résumé, c'est un algorithme qui offre une grande liberté sur la modélisation de la structure des données en raison de la forme de la pénalité, mais en contrepartie il nécessite une optimisation beaucoup plus complexe. Les solutions sur le choix des hyperparamètres (Gap Statistic) et le critère d'arrêt sont celles qui ont été proposées par Witten and Tibshirani (2010).

STRUCTURED WT-K-MEANS AVEC PÉNALISATION PAR LE LAPLACIEN Dans un article récent de Gong et al. (2018), une nouvelle forme de la pénalité $h(\mathbf{w})$ est proposée pour accompagner le problème d'optimisation (2.4). La méthode exploite les informations de corrélation entre les variables via la pénalité dite *Laplacian smoothing*, c'est-à-dire en fixant $h(\mathbf{w}) = \sum_{j,l\in\{1\dots,p\}} \text{cor}(\mathbf{x}^j, \mathbf{x}^l) \times (w_j - w_l)^2$. Ainsi, le problème peut s'écrire sous la forme suivante :

$$\underset{C_1,\dots,C_K,\mathbf{w}}{\text{maximiser}} \ \mathbf{w}^\intercal \mathbf{b} - \lambda \left\|\mathbf{w}\right\|_1 - \gamma \sum_{j,l\in\{1\dots,p\}} \text{cor}(\mathbf{x}^j, \mathbf{x}^l) \times (w_j - w_l)^2 \text{ s.c. } \left\|\mathbf{w}\right\|_2^2 \leqslant 1, w_j \geqslant 0 \ \forall j.$$

où $\lambda, \gamma \geqslant 0$ sont des paramètres de pénalisation. La pénalisation associée au paramètre γ introduit une *structured sparsity*. En effet, pour γ fixé, une forte corrélation entre les variables $\mathbf{x}^j$ et $\mathbf{x}^l$ conduit à une petite différence entre les poids w_j et w_l. Par conséquent, si des variables corrélées sont considérées comme importantes pour le clustering, elles auront tendance à être sélectionnées ensemble. De même, si elles ne sont pas importantes pour le clustering, elles auront tendance à être écartées ensemble. Les variables importantes sélectionnées par la méthode partagent donc une même structure et les résultats sont plus faciles à interpréter.

Les auteurs supposent dans l'article que $\text{cor}(\mathbf{x}^j, \mathbf{x}^l) \geqslant 0$ et donc la pénalité est toujours positive, mais dans le cas contraire le modèle force les variables avec une corrélation négative à avoir des poids différents (définir une pénalité avec la corrélation au carré ou multiplier les variables par -1 sont des solutions possibles). D'autre part en régression, la pénalité *lasso* a tendance à ne sélectionner qu'une seule variable dans le groupe de variables corrélées même si un grand nombre de ces variables, voire toutes, font partie du modèle sous-jacent (Bühlmann et al., 2013). En revanche, cette difficulté n'a pas encore été observée en clustering et nous verrons dans le Chapitre 4 que le WT-K-means a plutôt tendance à produire le phénomène inverse qui est de sélectionner les variables corrélées ensemble.

SUBSPACE WT-K-MEANS Une dernière extension a été proposée cette année par Diallo et al. (2021). Les auteurs proposent de pondérer la variance inter-classes par variable et par cluster ce qui donne le critère suivant :

$$\underset{C_1,\dots,C_K,\mathbf{w}}{\text{maximiser}} \ \sum_{k=1}^{K}\sum_{j=1}^{p} w_k^j b_{k,j} - \lambda \left\|\mathbf{w}\right\|_1, \text{ s.c. } \left\|\mathbf{w}\right\|_2^2 \leqslant 1, w_{k,j} \geqslant 0 \ \forall j,k,$$

où $\mathbf{w} = (w_1^1, ..., w_K^p)^\intercal$ sont les poids par variable et par cluster et $b_{k,j} = \frac{n_k}{n}(\bar{x}_k^j - \bar{x}^j)^2$ la contribution du cluster k à la variance inter-classes de la variable j. Cette approche est intéressante car elle apporte une information supplémentaire qui est l'importance des variables par cluster, ce qui permet de savoir si une variable est discriminante uniquement pour certains clusters. Cette formulation a un défaut notable. Imaginons par exemple un ensemble de données avec une seule variable centrée qui suit un mélange de trois gaussiennes tel que les clusters sont séparés, équidistants, de même effectif, de même variance. La variable est donc centrée en 0 et son centre se confond avec le milieu d'un des trois clusters, disons le deuxième. Or par définition, la variance inter-classes du deuxième cluster est nulle ($b_{2,1} = 0$) car le centre de ce cluster se confond avec le centre des données et ainsi $w_{2,1} = 0$. Cet exemple met en évidence le fait que l'algorithme supprime, entre autres, des clusters proches du centre des données. Ce type de comportement n'est pas recherché a priori et peut même être gênant pour le clustering. Pour incorporer l'importance des variables par cluster, il faudrait considérer les clusters deux à deux et non dans leur ensemble.

2.3.4 K-means sparses par *hill climbing*

Dans Arias-Castro and Pu (2017), les auteurs proposent ce qu'ils appellent "*a simple approach to sparse clustering by hill climbing*". En pratique, la méthodologie consiste à chercher un sous-ensemble de variables de taille s fixée à l'avance (parmi tous les sous-ensembles de variables de taille s), qui maximise la variance inter-classes à l'aide de l'algorithme des K-means et d'une méthode itérative de type *hill climbing*. Le *hill climbing* s'apparente à un algorithme de type *stepwise forward-backward* en introduisant le paramètre à optimiser s, qui indique le nombre de variables à sélectionner. Le choix de s (la taille de l'ensemble des variables à sélectionner) est choisi à l'aide du Gap-statistic comme dans Witten and Tibshirani (2010). La grande différence entre les deux méthodes est que pour différents niveaux de pénalisation, le WT-K-means peut trouver une même partition et le même sous-ensemble de variables importantes. Au contraire pour que la méthode décrite dans Arias-Castro and Pu (2017) conduise au *vrai* modèle, il faut fixer à l'avance le nombre de variables à retenir.

Dans leurs simulations le cardinal du sous-ensemble des variables importantes (s) apparaît toujours dans la liste des paramètres testés, ce qui donne un clair avantage à l'algorithme. Finalement les résultats ne sont pas forcément très différents de ceux obtenus avec le WT-K-means (où le choix du paramètre λ se fait avec le Gap Statistic). Dans l'article, une analyse du temps de calcul est faite et l'algorithme semble plus rapide que le WT-K-means qui sélectionne généralement davantage de variables.

2.3.5 Regularized K-means : pénalisation par les centres

L'idée principale du Regularized K-means est d'étendre l'algorithme K-means en ajoutant un terme de pénalité de type *group lasso* sur les centres (Sun et al., 2012). Plus précisément, le Regularized K-means est formulé comme suit :

$$\underset{C_1,...,C_K,\mathbf{C}}{\text{minimiser}} \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in C_k}\sum_{j=1}^{p}(x_i^j-\bar{x}_k^j)^2+\sum_{j=1}^{p}\lambda_j\left\|\mathbf{c}^j\right\|_2, \tag{2.7}$$

où $\mathbf{c}^j=(\bar{x}_1^j,...,\bar{x}_K^j)^\intercal$ et $\mathbf{C}=[\mathbf{c}^1,...,\mathbf{c}^p]$ et où il y a un terme de régularisation λ_j par variable $j=1,...,p$. La norme ℓ_2 dans la pénalité met simultanément à 0 toutes les coordonnées des centres d'une même variable (centrée) sur toutes les classes. Cela conduit à sélectionner des variables.

C'est dans le cadre supervisé que cette pénalisation avec un terme de pénalité λ_j différent par variable a d'abord été introduite, elle se nomme *adaptative group lasso* (Wang and Leng, 2008) et elle avait été proposée en raison des problèmes théoriques que posait le *group lasso* (Yuan and Lin, 2006). Ici (en clustering) la version adaptative donne à l'utilisateur la possibilité de pénaliser chaque variable différemment. Néanmoins, l'avantage pratique d'ajouter p paramètres à estimer, qui plus est en clustering, est discutable. Les questions de sélection de modèles sont des questions difficiles à résoudre, elles seront abordées en détail dans la suite et il semble plus judicieux de fixer $\lambda_1=...=\lambda_p$ pour ce modèle.

Il est impossible de résoudre tel quel le double problème d'optimisation (2.7) où l'on optimiserait par rapport aux centres et à la partition itérativement à l'aide de l'algorithme des K-means standard. L'astuce est donc algorithmique et elle consiste à réécrire l'algorithme heuristique de Loyd en y modifiant une étape. En effet, la pénalisation s'effectue uniquement par les centres, or si on laisse converger les K-means standards entre chaque mise à jour de la pénalisation, on aboutit chaque fois à la même solution (en supposant qu'ils trouvent les même centres à chaque itération).

Ils montrent dans leurs travaux que l'Équation (2.7) équivaut à

$$\underset{C_1,...,C_K,\mathbf{C}}{\text{minimiser}} \sum_{j=1}^{p}\left[\frac{1}{n}(\mathbf{x}^j-\mathbf{Z}\mathbf{c}^j)^\intercal(\mathbf{x}^j-\mathbf{Z}\mathbf{c}^j)+\lambda_j\left\|\mathbf{c}^j\right\|_2\right], \tag{2.8}$$

où $\mathbf{Z}\in\{0,1\}^{n\times K}$ est la matrice d'appartenance aux clusters. Malheureusement, la solution de l'Équation (2.8) n'est pas donnée dans l'article. On remarque que l'Équation (2.8) n'a pas de solution dans le cas général (Hastie et al., 2019). Une solution existe seulement lorsque la matrice d'appartenance aux clusters $\mathbf{Z}$ est orthonormale. La matrice $\mathbf{Z}$ n'est pas orthonormale par définition car elle n'est pas de rang plein et ses colonnes ne sont pas de norme égale à 1, ce qui pose des difficultés lors de la résolution du problème d'optimisation qui ne sont pas discutées dans l'article. Malgré tout, si nous supposons que la solution du problème d'optimisation (2.7) suivant $\mathbf{c}^j$ existe, on peut résoudre ce dernier avec la procédure proposée par Sun et al. (2012) : **Initialisation** : les centres $\mathbf{C}$ sont initialisés à l'aide du K-means. Ensuite on répète jusqu'à ce que l'on satisfasse à un critère d'arrêt :

1. **Affectation** : chaque observation est assignée au centre le plus proche.

$$C_k\leftarrow\{i\mid \underset{k'=1,...,K}{\text{argmin}}\ ||\mathbf{x}-\mathbf{c}_{k'}||_2^2=k\}$$

2. **Minimisation** : chaque centre est mis à jour en maximisant l'Équation (2.8) par rapport à $\mathbf{C}$.

où $\mathbf{c}_{k'}=(\mathbf{c}_{k'}^1,...,\mathbf{c}_{k'}^p)^\intercal$, c'est-à-dire que P_K et $\mathbf{C}$ sont mis à jour séparément à chaque itération en supposant que l'autre est fixé. Comme critère d'arrêt, les auteurs proposent "*the iteration stops when* $\mathbf{Z}$ *does not change any more.*". Plusieurs choix sont possibles et comme aucun détail n'est donné, on ne peut pas savoir ce que les auteurs ont implémenté.

2.3.6 Lasso-Weighted K-means

Cette méthode de K-means pondérés avec pénalisation lasso développée dans Chakraborty and Das (2019) aussi appelée Lasso-Weighted K-means (LW-K-means) est différente du WT-K-means (Witten and Tibshirani, 2010) car elle se base sur le H-K-means de Huang et al. (2005). Le problème d'optimisation correspondant s'écrit comme suit :

$$\underset{C_1,...,C_K,\mathbf{w}}{\text{maximiser}}\ g(\mathbf{w},\alpha,\beta)\mathbf{b}-\lambda\left\|\mathbf{w}\right\|_1 \text{ s.c. } \left\|\mathbf{w}\right\|_2^2\leqslant 1, w_j\geqslant 0\ \forall j.$$

où $g(\mathbf{w},\alpha,\beta)=(w_1^\beta+\frac{\alpha}{p^2}|w_1|,...,w_p^\beta+\frac{\alpha}{p^2}|w_p|)^\intercal$ est une fonction de poids bénéficiant de deux nouveaux paramètres α et β.

Cette nouvelle méthode bénéficie des bons résultats du H-K-means tout en proposant une solution sparse. De plus, l'article s'appuie sur la preuve de consistance du H-K-means pour prouver celle du LW-K-means. La grande différence se trouve dans la réécriture du poids associé à une variable, ce qui implique que la fonction objectif associe deux nouveaux hyperparamètres à optimiser. Ces deux paramètres sont fixés de manière arbitraire et les auteurs ne discutent pas leur influence. Pour le paramètre de régularisation lasso λ, les auteurs indiquent "*The value of λ was chosen by performing some hand-tuned experiments.*".

Les auteurs concluent sur des données simulées que les performances de leur algorithme sont proches de celles du WT-K-means en terme de partitionnement, mais que le WT-K-means a tendance à sélectionner un grand nombre de variables de bruit, en leur attribuant un poids très faible mais non nul. De plus, ils insistent sur le fait que le WT-K-means est plus lent et que cela est dû à l'utilisation du Gap Statistic qui nécessite de multiples exécutions de l'algorithme.

2.3.7 Résumé des travaux sur les K-means sparses

À partir de ce parcours de la littérature sur le K-means sparses, quelques points intéressants peuvent être notés.

1. Performances : nombreux sont les articles comparant les performances du WT-K-means en termes de clustering sur des données simulées, en calculant l'ARI (défini Section 1.4 Équation (1.1)) entre la vraie partition et celle trouvée par l'algorithme par exemple. Sur ce point, les performances sont souvent similaires et à part des cas très atypiques, il n'y a pas de schéma précis ou de type de données (par exemple des images ou des données très corrélées) pour lesquelles le WT-K-means se comporterait moins bien que les autres algorithmes.

2. Sélection de variables : les articles développant des méthodes de sélection de variables s'intéressent souvent à des méthodes interprétables, un des objectifs étant de sélectionner uniquement les variables importantes et toutes les variables importantes, c'est-à-dire celles qui sont liées au vrai clustering sous-jacent. Les travaux soulignent que le WT-K-means sélectionne généralement trop de variables, attribuant un poids faible mais non nul à certaines variables de bruit. Cela est majoritairement dû au choix du paramètre de pénalisation λ comme le confirment les simulations de la Section 2.7.

3. Temps de calcul : certains articles comparent le temps de calcul du WT-K-means avec ceux des autres algorithmes, et ils montrent que celui-ci est globalement plus lent. Là encore, en nous fondant sur plusieurs études, il nous semble que c'est dû à l'usage du Gap Statistic (défini Section 2.3.2 Équation (2.5)) pour le choix du paramètre λ, qui requiert de nombreuses exécutions de l'algorithme pour trouver le λ optimal.

4. Le choix du paramètre λ optimal : la version originale (Witten and Tibshirani, 2010) et quasiment toutes les extensions et travaux basés dérivés (à l'exception de Sun et al. (2012) ; Chakraborty and Das (2019)) utilisent le Gap Statistic pour déterminer le paramètre de pénalisation optimal λ. Comme on l'a remarqué précédemment, cette méthode est coûteuse en temps de calcul et elle semble mettre l'accent sur les performances plutôt que sur l'interprétabilité. De plus Daniella Witten et Robert Tibshirani eux-mêmes portent un avis critique en affirmant que "*the performance of the gap statistic of Section 3.2 for selecting the tuning parameter is mixed. This is not surprising, since tuning parameter selection in the unsupervised setting is known to be a very difficult problem. This is an area in which more work is needed.*". Nous tenterons d'expliquer plus en détail par la suite les difficultés et les possibles solutions de ce problème.

5. L'importance de l'initialisation : une bonne initialisation est cruciale. Dans la version originale du WT-K-means, l'algorithme est initialisé avec tous les poids égaux $w_j = 1/\sqrt{p}$, ensuite un premier K-means est exécuté et on obtient des clusters et donc des nouveaux poids. Dans cette étape, le K-means standard est appliqué sur les variables de départ (sans poids) et donc mécaniquement ce résultat aura beaucoup d'impact pour la suite de l'algorithme. En effet, si à la suite de cette étape, l'algorithme attribue aux variables importantes un poids proche de zéro, il sera difficile voire impossible pour le WT-K-means d'échapper à ce minimum local. Plusieurs solutions sont évoquées mais nous ne ferons pas d'analyses comparatives dans ce manuscrit.

6. Convergence des algorithmes de K-means sparses : avons-nous l'assurance que les algorithmes de K-means sparses convergent vers un minimum local ? En règle générale, ce n'est pas le cas. Seuls deux publications traitent de cette question, celles de Sun et al. (2012) pour l'algorithme *Regularized* K-means et de Chakraborty and Das (2019) pour l'algorithme LW-K-means. En pratique, aussi bien dans la

littérature que dans nos simulations, les algorithmes de K-means sparses arrivent toujours à converger en un nombre fini d'opérations (moins d'une dizaine pour le WT-K-means (Witten and Tibshirani, 2010)), mais il serait important d'étudier théoriquement leurs comportements encore largement inconnus.

2.4 Méthodes sparses basées sur les GMM

Avant de discuter des différentes méthodes de modèles de mélange gaussien sparses, nous rappelons brièvement la formulation du modèle dans le cadre du clustering sans sélection de variables.

2.4.1 Les modèles de mélange gaussien

DÉFINITIONS Pour le clustering basé sur les modèles de mélange gaussien (GMM) (McLachlan and Basford, 1988 ; McLachlan and Krishnan, 2007 ; McLachlan et al., 2019), on considère que les données appartiennent à K clusters $C_1, ..., C_K$. Chaque composante C_k de ce mélange est modélisée par la distribution gaussienne $\mathcal{N}_p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, $k = 1, ..., K$, et de densité $\phi(\mathbf{x}, |\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Si on définit des poids positifs $\pi_1, ..., \pi_K$ tels que $\sum_{k=1}^{K} \pi_k = 1$ comme proportion du mélange, alors un modèle de mélange gaussien est défini par sa densité :

$$p(\mathbf{x}, \theta) = \sum_{k=1}^{K} \pi_k \phi(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.9}$$

où $\theta = (\pi_1, ..., \pi_K, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K)$ est le paramètre global du mélange. Pour un échantillon $(\mathbf{x}_1, ..., \mathbf{x}_n)$, la log-vraisemblance de ce modèle de mélange est alors :

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \log\Big(\sum_{k=1}^{K} \pi_k \phi(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \Big).$$

L'estimation des paramètres de ce modèle ne peut se faire directement par la maximisation de la vraisemblance puisque les clusters sont inconnus. Bien qu'il ne soit pas spécifiquement dédié aux modèles de mélange, l'algorithme EM (Expectation-Maximization), proposé par Dempster et al. (1977) est certainement la technique la plus utilisée pour estimer les paramètres dans ce cadre. Son fonctionnement repose sur la vraisemblance des données complétées par des variables indicatrices. Notons $\mathbf{Z} \in \{0, 1\}^{n \times K}$ la matrice indicatrice d'appartenance aux clusters dont les lignes sont définies par $\mathbf{z}_i^k = (z_i^1, ..., z_i^K)^\intercal$ avec $z_i^k = 1$ si $i \in C_k$ et 0 sinon. L'algorithme EM maximise itérativement l'espérance conditionnelle de la log-vraisemblance des données complétées :

$$\mathcal{L}_c(\theta, \mathbf{Z}) = \sum_{k=1}^{K} \sum_{i=1}^{n} z_i^k \log\Big(\pi_k \phi(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \Big).$$

DIFFÉRENTES FORMES DES MODÈLE DE MÉLANGE GAUSSIEN Généralement lorsque l'on parle des modèles de mélange gaussien (GMM), on fait le plus souvent référence au modèle complet 2.9 où les matrices de covariance $\boldsymbol{\Sigma}_k$ peuvent dépendre du cluster. Comme le font remarquer Bouveyron and Brunet-Saumard (2014b), ce type de modèle avec $p = 100$ et $K = 4$ requiert l'estimation de 20603 paramètres alors que pour l'algorithme des K-means, cela nécessite l'estimation que de 400 paramètres (les moyennes des 100 variables par cluster). Dans le but de réduire le nombre de paramètres, des formes contraintes des GMM ont été proposées simultanément par Banfield and Raftery (1993) et Celeux and Govaert (1995). L'idée est d'utiliser la décomposition spectrale des matrices de covariance par cluster, et ensuite de contraindre certains éléments de la décomposition à être égaux. Soit la décomposition spectrale de la matrice $\boldsymbol{\Sigma}_k$:

$$\boldsymbol{\Sigma}_k = s_k \mathbf{V}_k \mathbf{A}_k \mathbf{V}_k^\intercal, \tag{2.10}$$

où $s_k = \det(\boldsymbol{\Sigma}_k)^{\frac{1}{p}}$ donne le volume du cluster C_k, $\mathbf{V}_k \in \mathbb{R}^{p \times p}$ est la matrice orthogonale des vecteurs propres qui donnent l'orientation de C_k et $\mathbf{A}_k$ une matrice diagonale proportionnelle aux valeurs propres, avec $\det(\mathbf{A}_k) = 1$, qui précise la forme. Ainsi, pour réduire le nombre de paramètres, on peut par exemple poser $\forall k$, $\mathbf{V}_k = \mathbf{V}$, pour fixer l'orientation par cluster. On peut utiliser d'autres contraintes pour fixer la forme et le volume. Différentes combinaisons sont proposées (Banfield and Raftery, 1993 ; Celeux and Govaert, 1995) pour obtenir finalement 14 formes de modèles différents.

Bayesian Information Criterion Le problème du choix entre plusieurs modèles est particulièrement bien posé dans le paradigme bayésien. Le cœur de la modélisation statistique bayésienne est l'appréhension de l'incertitude par le biais de distributions préalables sur les paramètres θ, maintenant traités comme des variables latentes.

De nombreux critères ont été proposés pour choisir entre différents modèles dans le cadre non-supervisé. Parmi ces modèles, les critères bayésiens qui choisissent le modèle pour lequel la probabilité des observations est maximum sont largement utilisés dans le cas des GMM. Le critère *Bayesian Information Criterion* (BIC) (Schwarz, 1978) est certainement l'un des plus connus. Le critère BIC est constitué de deux termes : le terme de vraisemblance qui favorise la sélection d'un modèle complexe et un terme de pénalité, fonction croissante du nombre de paramètres, qui favorise la sélection d'un modèle parcimonieux. Pour simplifier un modèle, il s'agit de minimiser la quantité suivante :

$$\mathrm{BIC}(\mathcal{M}) = 2\mathcal{L} - \eta(\mathcal{M})\log(n), \tag{2.11}$$

où $\mathcal{L}$ est la log-vraissemblance, $\eta(\mathcal{M})$ est le nombre de paramètres du modèle $\mathcal{M}$ et n le nombre d'observations.

En pratique, le critère BIC est à préférer au critère *Akaike Information Criterion* (AIC) qui n'est pas consistent et qui ne pénalise pas suffisamment la complexité des modèles et a tendance à surestimer le nombre de paramètres.

Dans la suite de cette section, nous présentons les méthodes de clustering sparse basées sur les GMM.

2.4.2 Méthodes par pénalisation de la vraisemblance

Forme générale Les modèles de mélange en clustering se prêtent facilement à la sélection de variables. Plutôt que de chercher $\boldsymbol{\mu}_k$ et $\boldsymbol{\Sigma}_k$ qui maximisent la log-vraisemblance, on maximise une log-vraisemblance pénalisée pour obtenir de la sparsité. La forme générale de la fonction de log-vraisemblance pénalisée est la suivante :

$$\mathcal{L}_h(\theta) = \mathcal{L}(\theta) - h(\theta)$$

où $\mathcal{L}(\theta)$ est la log-vraisemblance et $h(\theta)$ une fonction de pénalité (pénalité qui est diminue avec le nombre de variables du modèle). Les modèles suivants se caractérisent par le choix de la fonction de pénalité h.

Pénalisation par les centres En supposant que les données sont centrées et que le modèle de mélange est gaussien avec des matrices de covariance diagonales communes, Pan and Shen (2007) proposent la pénalité suivante :

$$h(\theta) = \lambda \sum_{k=1}^{K} \sum_{j=1}^{p} |\mu_k^j|$$

où μ_k^j désigne la moyenne de la j-ième variable dans le cluster k et λ est un hyperparamètre qui représente le niveau de sparsité souhaité. Lorsque le paramètre λ est grand, certains des éléments μ_k^j sont exactement égaux à zéro. Si, pour une certaine variable j, $\mu_k^j = 0, \forall k = 1, ..., K$, alors la variable j est considérée comme exclue du modèle final, car elle ne permet en aucune façon de discriminer les clusters. Il faut noter qu'il est nécessaire que les variables soient centrées et de variance unitaire, comme l'indiquent les auteurs. Cette méthode est similaire dans sa formulation au Regularized K-means car la pénalité est fonction des centres. On peut remarquer que Wang and Zhu (2008) ont étendu la pénalisation à la norme infinie, ce qui a pour conséquence de discriminer tous les centres d'une même variable ensemble. Cela a l'avantage d'imposer une sélection de variables globalement importantes, mais en même temps une information précieuse est perdue : une variable peut être importante uniquement sur certaines composantes du mélange, certains clusters.

Pénalisation par les centres et les variances Xie et al. (2008) ont étendu le modèle de Pan and Shen (2007) en relaxant la contrainte d'égalité sur les matrices de covariance. En effet, ils considèrent le cas des matrices de covariance diagonales, mais différentes par cluster. En posant pour tout $k = 1, ..., K$, $\boldsymbol{\Sigma}_k = \mathrm{diag}(\sigma_{k,1}^2, ..., \sigma_{k,p}^2)$ cela conduit à la fonction de pénalité suivante :

$$h(\theta) = \lambda \sum_{k=1}^{K} \sum_{j=1}^{p} |\mu_k^j| + \gamma \sum_{k=1}^{K} \sum_{j=1}^{p} |\sigma_{k,j}^2 - 1|.$$

Un second terme régularisé est donc ajouté, qui porte sur la variance de la variable j de la k-ième composante soit $\sigma_{k,j}^2$. Il est là encore indispensable de normaliser les variables à moyenne 0 et variance 1. La justification de la normalisation est visible dans la forme de la pénalisation où, en effet, la deuxième partie de la pénalité augmente avec l'écart entre les variances par cluster et la variance globale (dans le cas où les variables sont normalisée à variance unitaire).

PÉNALISATION PAR LES CENTRES ET LES COVARIANCES Enfin quelques autres versions, peut-être moins connues, ont été proposées mais nous n'en introduirons qu'une seule qui a été récemment utilisée dans un travail qui nous sera utile par la suite. Cette pénalisation est plus générale et est énoncée dans le travail de Zhou et al. (2009) où des hypothèses de moins en moins fortes sont imposées à la matrice de covariance. En effet, la pénalité est écrite comme suit :

$$h(\theta) = \lambda \sum_{k=1}^{K} \sum_{j=1}^{p} |\mu_k^j| + \gamma \sum_{k=1}^{K} \sum_{j=1}^{p} \sum_{l \neq j} |(\mathbf{\Sigma}_k^{-1})_{j,l}| \tag{2.12}$$

où $(\Sigma_k^{-1})_{j,l}$ est l'élément de la j-ième colonne et de la l-ième ligne de l'inverse de la matrice de covariance du cluster k. L'algorithme *graphical lasso* Friedman et al. (2008), qui utilise une procédure de *coordinate descent* pour le lasso, est utilisé pour estimer la matrice inverse sparse $\mathbf{\Sigma}_k^{-1}$ pour tout $k = 1, ..., K$.

2.4.3 Méthodes par pénalisation des *loadings* (5)

Dans cette section sur les méthodes GMM pénalisées, introduisons une dernière méthode sparse qui se base sur l'algorithme Fisher-EM (Bouveyron and Brunet, 2012). Charles Bouveyron et Camille Brunet proposent une famille de modèles de mélange qui ajustent les données dans un sous-espace discriminant commun. Ce modèle de mélange, appelé *Discriminative Latent Mixture* (DLM), se caractérise par un sous-espace latent commun à tous les groupes et est supposé être le sous-espace le plus discriminant pour une dimension fixée, en maximisant la séparation entre les groupes. L'inférence des modèles DLM ne peut pas se faire avec l'algorithme EM en raison des caractéristiques spécifiques du sous-espace latent. Pour surmonter ce problème, Bouveyron and Brunet (2012) ont proposé un algorithme, nommé Fisher-EM, pour estimer à la fois le sous-espace discriminant et les paramètres du modèle de mélange. Cet algorithme est basé sur l'algorithme EM auquel une étape supplémentaire est ajoutée entre l'étape E et l'étape M. Cette étape supplémentaire, appelée étape F, vise à calculer la matrice de projection $\mathbf{U}$ dont les colonnes couvrent l'espace latent discriminant et cela revient à chercher, à chaque itération, une estimation $\hat{\mathbf{U}}$ telle que :

$$\hat{\mathbf{U}} = \underset{\mathbf{U}}{\text{maximiser}}\ \text{trace}\Big(\mathbf{U}^\intercal \hat{\mathbf{\Sigma}}_\mathbf{X} \mathbf{U}\Big)^{-1} \Big(\mathbf{U}^\intercal \hat{\mathbf{\Sigma}}_\mathbf{b} \mathbf{U}\Big) \text{ s.c. } \mathbf{U}^\intercal \mathbf{U} = \mathbf{I}_d, \tag{2.13}$$

où $\hat{\mathbf{\Sigma}}_\mathbf{X}$ est la matrice de covariance des données et $\hat{\mathbf{\Sigma}}_\mathbf{b}$ est la *soft between covariance matrix* définie comme :

$$\hat{\mathbf{\Sigma}}_\mathbf{b} = \frac{1}{n} \sum_{k=1}^{K} n_k (\mathbf{m}_k - \bar{\mathbf{x}})(\mathbf{m}_k - \bar{\mathbf{x}})^\intercal,$$

où $n_k = \sum_{i=1}^{n} t_{i,k}$ et $\mathbf{m}_k = \frac{1}{n_k} \sum_{i=1}^{n} t_{i,k} \mathbf{x}_i$, où $t_{i,k}$ est la probabilité a posteriori que l'observation i appartienne au cluster k et $d \leqslant K - 1$ la dimension de l'espace latent. La matrice $\hat{\mathbf{\Sigma}}_\mathbf{b}$ est donc estimée à partir des clusters et est différentes à chaque itération si la partition (et/ou les paramètres) change.

Dans le contexte de la sélection de variables, les auteurs proposent une extension de l'algorithme Fisher-EM pour réduire le nombre de variables dans l'espace latent discriminant (Bouveyron and Brunet-Saumard, 2014a). Ainsi, ils décident de pénaliser directement la matrice de projection $\mathbf{U}$, aussi appelée matrice des *loadings*. Pour réaliser la pénalisation de $\mathbf{U}$, trois solutions sont proposées dans Bouveyron and Brunet-Saumard (2014a). Les trois solutions ont un point commun, elles modifient l'étape F de l'algorithme Fisher-EM pour introduire de la sparsité dans $\mathbf{U}$. Néanmoins, cette méthode introduit de la sparsité de manière différente dans les colonnes de $\mathbf{U}$ et donc pour qu'une variable soit exclue du modèle, il faut que son poids soit mis à 0 dans chaque colonne de $\mathbf{U}$, ce qui n'est pas explicitement demandé par le modèle contrairement à des approches d'analyse discriminante linéaire munie d'une pénalité *group lasso* (Merchante et al., 2012) ou à des approches d'analyse en composantes principales sparses ayant une sparsité structurée (Mattei et al., 2016).

Dans la première approche proposée, la matrice $\hat{\mathbf{U}}$ est approximée à chaque étape par une matrice sparse à l'aide d'une régression pénalisée qui peut être calculée par l'algorithme *Least-Angle Regression* (LARS) (Efron et al., 2004).

Pour la deuxième approche, les auteurs reformulent le problème de maximisation du critère de Fisher comme un problème de régression et ils obtiennent une matrice *loadings* sparse en résolvant le problème lasso associé à ce problème de régression. Cependant, la résolution de ce problème lasso n'est pas directe dans ce cas et nécessite l'utilisation d'un algorithme itératif.

Dans la dernière approche, la maximisation du critère de Fisher est reformulée comme un problème de décomposition en valeurs singulières (SVD) de la matrice de covariance inter-classes. Une approximation sparse de la solution de ce problème est obtenue en effectuant une SVD pénalisée (Witten et al., 2009).

En ce qui concerne les détails de l'implémentation, Bouveyron and Brunet-Saumard (2014a) ont proposé d'initialiser l'algorithme Sparse Fisher-EM avec le résultat de l'algorithme Fisher-EM et de déterminer la valeur des deux paramètres de pénalisation à l'aide du critère BIC, pour lequel la complexité du modèle dépend du nombre de valeurs non nulles. Il faut noter que seule la première des trois solutions traite du cas où $n < p$ et dans ce cas, l'algorithme LARS est remplacé par une méthode de type *elastic-net* (Zou and Hastie, 2005).

L'un des grands avantages des algorithmes Fisher-EM et Sparse Fisher-EM est qu'ils permettent de visualiser les classes dans l'espace latent. En effet, offrir un outil de visualisation conjointement à celui du clustering est très intéressant, qui plus est en clustering car la vérification visuelle des classes obtenues est souvent un des seuls moyens de s'assurer de la qualité de la partition obtenue. Ce point sera abordé plus en détail dans la Section 2.6.

2.4.4 Sélection de variables via la sélection de modèles

L'idée sous-jacente des modèles présentés dans cette section est de déterminer pour chaque variable son rôle dans le clustering. Ainsi, une variable peut être déclarée comme importante, ou plus exactement dépendante du clustering, ou indépendante (Raftery and Dean, 2006). Certains travaux vont même plus loin (Maugis et al., 2009a,b) en ajoutant un sous-ensemble de variables dites redondantes.

L'APPROCHE DE RAFTERY ET DEAN Dans les méthodes de clustering sparse se basant sur la sélection de modèle, deux groupes sont généralement à trouver : le groupe des variables importantes et le groupe des variables de bruit. L'hypothèse d'indépendance des groupes fut d'abord considérée par Law et al. (2004). Ce cas correspond au cas où les variables non importantes sont indépendantes à la fois du clustering et des variables importantes. En revanche, certains auteurs considèrent cette hypothèse d'indépendance entre les groupes irréaliste (Raftery and Dean, 2006).

Raftery and Dean (2006) définissent la matrice des données $\mathbf{X}$ comme étant partitionnée en trois parties à découvrir à l'aide d'un algorithme itératif :

- $\mathbf{X}^C$ les variables importantes pour le clustering.
- $\mathbf{X}^N$ les variables indépendantes du clustering, non importantes ou aussi appelées variables de bruit. Les variables non importantes sont définies comme indépendantes du clustering mais dépendantes des variables importantes selon une relation linéaire.
- $\mathbf{X}^T = \mathbf{X} \backslash [\mathbf{X}^C, \mathbf{X}^N]$ un ensemble de variables à tester à chaque itération. Ensuite, l'algorithme détermine tour à tour pour chaque variable $\mathbf{x}^j \in \mathbf{X}^T$ si elle fait partie du groupe $\mathbf{X}^C$ ou $\mathbf{X}^N$.

La méthode de Raftery and Dean (2006) requiert la définition de plusieurs critères BIC.

1. On nomme $\text{BIC}_{clust}(\mathbf{X}^C, \mathbf{x}^j)$, le BIC donné par l'Équation (2.11) pour le modèle GMM de clustering basé sur $[\mathbf{X}^C, \mathbf{x}^j]$ en K clusters et il est défini comme :

$$\text{BIC}_{clust}(\mathbf{X}^C, \mathbf{x}^j) = \max_{\mathcal{M} \in \mathcal{M}_1, \ldots, \mathcal{M}_{14}} \text{BIC}_K \mathcal{M}(\mathbf{X}^C, \mathbf{x}^j),$$

   où les 14 modèles sont ceux dont la forme est décrite par l'Équation (2.10). Si $\mathbf{X}^C = \emptyset$, alors seulement deux modèles $\mathcal{M}_1, \mathcal{M}_2$ sont testés avec soit la variance égale par cluster, soit des variances différentes par cluster.

2. $\text{BIC}_{reg}(\mathbf{x}^j | \mathbf{X}^C)$ est le BIC d'un modèle de régression des variables $\mathbf{X}^C$ sur la variable $\mathbf{x}^j$ à prédire. Le BIC en régression se calcule de la façon suivante :

$$\text{BIC}_{reg}(\mathbf{x}^j | \mathbf{X}^C) = -n\log(2\pi) - n\log(\text{RSS/n}) - n - (\dim(\mathbf{X}^C) + 2)\log(n),$$

   où RSS est la somme des carrés résiduels dans la régression de $\mathbf{X}^C$ sur les variables $\mathbf{x}^j$ et dim est la dimension (nombre de colonnes) de la matrice en question.

3. On nomme $\text{BIC}_{noclust}(\mathbf{x}^j | \mathbf{X}^C)$ la somme de deux BIC définie par :

$$\text{BIC}_{noclust}(\mathbf{X}^C, \mathbf{x}^j) = \text{BIC}_{reg}(\mathbf{x}^j | \mathbf{X}^C) + \text{BIC}_{clust}(\mathbf{X}^C),$$

   et si $\mathbf{X}^C = \emptyset$ alors la première variable de clustering est choisie comme celle qui donne la plus grande différence entre le BIC pour le clustering (maximisé sur différentes paramétrisations) et le BIC pour aucun clustering (une structure de groupe unique maximisée sur différentes paramétrisations) où chaque variable est considérée séparément.

4. On nomme $\text{BIC}_{diff}(\mathbf{X}^C, \mathbf{x}^j)$ la différence définie par :

$$\text{BIC}_{diff}(\mathbf{X}^C, \mathbf{x}^j) = \text{BIC}_{clust}(\mathbf{X}^C, \mathbf{x}^j) - \text{BIC}_{noclust}(\mathbf{X}^C), \tag{2.14}$$

qui permet de savoir si $\mathbf{x}^j$ doit être ajouté et la variable à ajouter à chaque itération est choisie telle que $j = \underset{\mathbf{x}^j \in \mathbf{X}^T}{\text{argmax}}\ \text{BIC}_{diff}(\mathbf{X}^C, \mathbf{x}^j)$.

Une caractéristique importante de la formulation est qu'il n'est pas nécessaire pour les variables non importantes d'être indépendantes des variables de clustering, du fait de l'inclusion du critère BIC_{reg} dans les critères de sélection. Cela permet d'écarter les variables redondantes liées aux variables de clustering mais pas au clustering lui-même. Il faut noter que nous ne partageons pas ce point de vue dans ce manuscrit. Il est pour nous difficile d'imaginer deux variables identiques, dont l'une serait importante et l'autre considérée comme non importante et redondante.

Comme décrit par Raftery and Dean (2006), la mise en œuvre pratique de la méthodologie ci-dessus nécessite de pouvoir vérifier l'inclusion et l'exclusion des variables. Deux méthodologies sont proposées : un algorithme de recherche *stepwise forward-backward*, et un algorithme *headlong*. Précisément, l'algorithme *stepwise forward-backward* fonctionne de la façon suivante :

1. Sélectionner la première variable $\mathbf{x}^j \in \mathbf{X}$ comme étant celle qui donne le meilleur clustering univarié, c'est-à-dire celle qui maximise l'Équation (2.14).
2. Proposer la prochaine variable $\mathbf{x}^j \in \mathbf{X}^T$ de clustering comme étant celle qui donne le meilleur clustering multivarié incluant les variables précédentes sélectionnées et l'accepter si $\text{BIC}_{diff}(\mathbf{X}^C, \mathbf{x}^j) > 0$.
3. Proposer la variable $\mathbf{x}^l$ à supprimer de l'ensemble actuel des variables de clustering sélectionnées ($\mathbf{X}^C$) comme étant celle qui dégrade le moins les performances du clustering et retirer cette variable de l'ensemble des variables de clustering $\mathbf{X}^C$ pour la mettre dans l'ensemble $\mathbf{X}^N$ si $\text{BIC}_{diff}(\mathbf{X}^C, \mathbf{x}^l) \leqslant 0$.
4. Itérer les étapes 2 et 3 jusqu'à ce que deux étapes soient rejetées consécutivement.

Ainsi, à chaque étape d'inclusion, l'algorithme *stepwise forward-backward* considère tour à tour chaque variable qui ne fait pas partie des variables déjà sélectionnées et évalue s'il y est pertinent de l'ajouter. On fait de même à chaque étape d'exclusion. D'autre part l'algorithme de recherche *headlong* ne vérifie pas toutes les variables une par une. L'algorithme parcourt les variables jusqu'à ce qu'il trouve une variable pour laquelle le critère BIC_{diff} pour l'inclusion ou l'exclusion dépasse un seuil prédéfini. L'algorithme *headlong* est moins coûteux en temps de calcul que l'algorithme *stepwise forward-backward*, mais la recherche est non-exhaustive et elle donne lieu à de moins bonnes solutions.

EXTENSION DE L'APPROCHE DE RAFTERY ET DEAN Considérer que les variables non importantes dépendent des variables importantes par une relation linéaire est une hypothèse forte qui peut ne pas être valide dans certains cas pratiques. Concrètement, la variable $\mathbf{x}^j$ à examiner à chaque itération ne peut être lié qu'à un sous-ensemble $\mathbf{X}^R \subseteq \mathbf{X}^C$ des variables importantes. De cette manière, les auteurs évitent l'inclusion de paramètres inutiles qui pénaliseraient trop la log-vraisemblance. Une autre limitation de la procédure de Raftery et Dean est liée à leur algorithme de sélection des variables de type *stepwise forward-backward*. Ce type de procédure a des défauts, comme celui de construire des modèles emboîtés, or les modèles plus petits ne sont pas forcément des sous-ensembles des modèles plus grands (Judd et al., 2011). Pour surmonter ces limitations, Maugis et al. (2009a) relâchent les hypothèses en proposant une nouvelle formulation de BIC_{reg} avec un nouveau sous ensemble de variables importantes $\mathbf{X}^R \subseteq \mathbf{X}^C$ tel que :

$$\text{BIC}^{\star}_{noclust} = \text{BIC}_{clust}(\mathbf{X}^C) + \text{BIC}_{reg}(\mathbf{x}^j | \mathbf{X}^R \subseteq \mathbf{X}^C),$$

où $\text{BIC}_{reg}(\mathbf{x}^j | \mathbf{X}^R \subseteq \mathbf{X}^C)$ est le BIC de la régression de $\mathbf{X}^R$ sur $\mathbf{x}^j$ à prédire et où $\mathbf{X}^R$ est défini par la sélection de l'ensemble optimal des variables à prédire de $\mathbf{X}^P$. En outre, si $\text{BIC}^{\star}_{diff}(\mathbf{X}^C, \mathbf{x}^j) > 0$ (défini en remplaçant $\text{BIC}_{noclust}$ par $\text{BIC}^{\star}_{noclust}$ dans l'Équation (2.14)), alors $\mathbf{x}^j$ peut être ajoutée. Le modèle offre plus de possibilités lors de la modélisation mais un problème apparaît : $\mathbf{X}^R$ doit également être déterminée par un algorithme *stepwise backward*.

MODÈLE SRUW (6) Les auteurs vont encore plus loin dans Maugis et al. (2009b) en considérant un modèle, appelé SRUW, basé sur un nouveau critère où des variables peuvent être complètement indépendantes des variables importantes et cela se traduit par la définition d'un nouveau sous ensemble de variables importantes $\tilde{\mathbf{X}}^R \subseteq \mathbf{X}^C$ et d'un nouveau critère :

$$\text{BIC}^{\star\star}_{noclust} = \text{BIC}_{clust}(\mathbf{X}^C) + \text{BIC}_{reg}(\mathbf{x}^j | \tilde{\mathbf{X}}^R \subseteq \mathbf{X}^C).$$

Cela correspond au critère $\text{BIC}^{\star}_{noclust}$ où le modèle de régression avec $\mathbf{X}^R = \emptyset$ est autorisé tel que si $\text{BIC}^{\star\star}_{diff} > 0$ (défini en remplaçant $\text{BIC}_{noclust}$ par $\text{BIC}^{\star\star}_{noclust}$ dans l'Équation (2.14)) alors $\mathbf{x}^j$ est ajoutée. Ainsi, le modèle autorise toutes les variables de bruit à être indépendantes des variables importantes. Une procédure de type *stepwise backward* est mis en oeuvre pour le modèle de régression, plus efficace que la version *stepwise forward*, mais aussi plus lente car elle requiert de multiples évaluations de modèles incluant la quasi-totalité des variables même si en régression linéaire la sélection de variables *stepwise backward* n'est pas trop coûteuse en temps de calcul et que le nombre de variables dans l'ensemble $\mathbf{X}^C$ ne devrait pas être trop grand (Bouveyron et al., 2019). Par ailleurs la sélection de variables de type *stepwise forward*, démarrant d'un ensemble vide, doit être préférée pour le clustering car sinon la procédure peut devenir numériquement impraticable pour des ensembles de données dépassant les quelques dizaines de variables (Bouveyron et al., 2019).

EXTENSION DU MODÈLE SRUW (7) Les procédures de sélection de variables avec une, deux ou plus de procédures *stepwise* intégrées restent coûteuses en temps de calcul et des procédures alternatives sont souhaitables. Ainsi, Gilles Celeux, Cathy Maugis-Rabusseau et Mohammed Sedki proposent dans un travail récent (Celeux et al., 2019) de combiner le modèle SRUW avec la méthode de Zhou et al. (2009) définie à la Section 2.4.2. La procédure *stepwise* est remplacée par un classement des variables à l'aide du modèle décrit par l'Équation (2.12). À proprement parler, le modèle avec une pénalité *lasso* ne donne pas de classement des variables. Celeux et al. (2019) proposent de compter pour chaque valeur des paramètres λ et de γ le nombre de fois où une variable est sélectionnée, c'est-à-dire le nombre de fois où au moins une moyenne d'un cluster est différent de 0.

Formellement, les paramètres de régularisation (λ, γ) varient dans une grille de paramètres notée $\mathcal{G}_\lambda \times \mathcal{G}_\gamma$ et l'importance est définie pour chaque variable $j \in \{1, ..., p\}$ et pour un K fixé tel que :

$$\mathcal{O}_K(j) = \sum_{(\lambda,\gamma)\in\mathcal{G}_\lambda\times\mathcal{G}_\gamma} \mathcal{I}_{(K,\lambda,\gamma)}(j), \tag{2.15}$$

où

$$\mathcal{I}_{(K,\lambda,\gamma)}(j) = \begin{cases} 0 \text{ si } \forall k = 1, ..., K, \mu_{k,j} = 0, \\ 1 \text{ sinon.} \end{cases}$$

Plus $\mathcal{O}_K(j)$ est grand, plus la variable j est censée être liée au clustering et donc importante. Les variables sont ensuite classées par valeurs décroissantes de $\mathcal{O}_K(j)$. Il est écrit dans Bouveyron et al. (2019) que "*It is hoped that using this lasso-like ranking of the variables instead of stepwise algorithms would not degrade the identification of the sets S, R, U and W*". Même s'il n'est pas certain que cette méthodologie soit plus performante, elle reste néanmoins plus rapide.

Deux autres défauts de cette méthode, non soulignés dans la littérature peuvent être relevés. Premièrement, pour différentes valeurs de (λ, γ), des clustering complètement différents peuvent être obtenus. Par exemple, un groupe de variables $\mathbf{X}^1 \in \mathbf{X}$ peut être considéré comme important par l'algorithme pour des petites valeurs du paramètre λ alors qu'un deuxième groupe $\mathbf{X}^2 \in \mathbf{X}$ indépendant du premier peut être considéré comme important par l'algorithme pour des grandes valeurs du paramètre λ. Or, ici, l'importance d'une variable est définie sur l'ensemble de la grille de valeurs. En d'autres termes, pour certaines valeurs de (λ, γ), des modèles très bruités peuvent être obtenus et peuvent modifier l'estimation de $\mathcal{O}_K$. Dissocier la mesure d'importance de la partition à laquelle elle se rattache est risqué. Deuxièmement, l'importance d'une variable définie par l'Équation (2.15) dépend de l'échelle de la grille de (λ, γ). L'échelle de la grille influence les résultats en mettant l'accent sur certains intervalles de la grille, intervalles pouvant correspondre à des modèles bruités et à une représentation erronée. Par ailleurs en apprentissage supervisé, pour la régression *lasso*, il est d'usage de fixer une grille d'échelle logarithmique pour $\mathcal{G}_\lambda$ (Friedman et al., 2010). Ici, l'échelle est linéaire ou fixée par l'utilisateur, mais son influence sur $\mathcal{O}_K$ est non négligeable.

SÉLECTION DE VARIABLES PAR MAXIMISATION DU MICL (8) Dans les approches ci-dessus, à chaque étape de l'algorithme de sélection des variables, la vraisemblance d'un GMM doit être optimisée plusieurs fois. Pour éviter de telles répétitions, Marbac and Sedki (2017) proposent une procédure qui repose sur le critère ICL (Biernacki et al., 2010) et celle-ci ne nécessite pas de multiples appels de l'algorithme EM. Dans ce cadre, une variable est déclarée comme non importante pour le clustering si ses distributions marginales unidimensionnelles sont égales pour toutes les composantes du mélange. Ainsi, l'hypothèse d'indépendance des variables dans les clusters est faite.

Les auteurs introduisent une variable indicatrice de poids $\mathbf{w} = (w_1, ..., w_p)$ telle que $w_j = 1$ si la variable $\mathbf{x}^j$ est importante, 0 sinon. Dans ce contexte, $\mathbf{X}^C = \{\mathbf{x}^j \in \mathbf{X} : w_j = 1, \forall j\}$ et $\mathbf{X}^N = \{\mathbf{x}^j \in \mathbf{X} : w_j = 0, \forall j\}$

et les différents modèles sont spécifiés par $\mathbf{w}$. Ces modèles sont comparés à l'aide d'un critère basé sur la vraisemblance intégrée des données complètes :

$$p(\mathbf{X}, \mathbf{Z}|\mathbf{w}) = \int p(\mathbf{X}, \mathbf{Z}|\theta, \mathbf{w})p(\theta|\mathbf{w})d\theta,$$

où θ et $\mathbf{Z}$ sont définis dans la Section 2.4.1. Après avoir supposé l'indépendance locale et globale de toutes les variables, l'intégrale ci-dessus se réduit à :

$$p(\mathbf{X}, \mathbf{Z}|\mathbf{w}) = p(\mathbf{Z}) \prod_{j=1}^{J} p(\mathbf{x}_j|w_j, \mathbf{Z}).$$

Pour une variable non importante, $p(\mathbf{x}_j|w_j, \mathbf{Z}) = p(\mathbf{x}_j|w_j)$ puisque cette quantité ne dépend pas du clustering, alors que pour une variable importante, cette quantité prend des valeurs différentes sur les composantes du mélange. Par conséquent, la sélection des variables est effectuée en déterminant le vecteur $\mathbf{w}$ qui maximise le critère MICL (Maximum Integrated Complete-data Likelihood), donné par :

$$\text{MICL}(\mathbf{w}) = \underset{\mathbf{Z}}{\text{argmax}} \log p(\mathbf{X}, \mathbf{Z}|\mathbf{w}).$$

La maximisation du MICL($\mathbf{w}$) est effectuée de manière itérative en utilisant un algorithme qui alterne entre deux étapes d'optimisation de l'ICL : optimisation sur le clustering donné par $\mathbf{Z}$ étant donné $\mathbf{w}$, et maximisation sur $\mathbf{w}$ étant donné le clustering $\mathbf{Z}$. Selon les auteurs, cette approche est censée être rapide et adaptée aux problèmes comportant un grand nombre de variables. De plus, l'optimisation sur $\mathbf{Z}$ peut être coûteuse en temps de calcul pour des échantillons de grande taille. Les auteurs suggèrent que pour les tailles d'échantillon inférieures à 10^4, cette optimisation est encore possible.

Les auteurs étendent leur modèle aux données mixtes Marbac et al. (2020). Ils montrent que le MICL peut s'écrire exactement sans approximation sous une forme optimisable pour les modèles de mélange avec des données mixtes. Par conséquent, la sélection du modèle est effectuée par une procédure que les auteurs considèrent comme simple et rapide, et celle-ci alterne entre deux maximisations présentées dans Marbac and Sedki (2017).

L'applicabilité du modèle aux données mixtes et le fait que l'algorithme semble avoir un faible temps d'exécution sont des avantages non négligeables de la méthode. En revanche, elle repose sur des hypothèses fortes d'indépendances (locale et globale) qui pourraient être trop restrictives, comme nous le verrons Chapitre 4.

2.4.5 VSCC : La méthode de Jeffrey L. Andrews et Paul D. McNicholas (9)

Introduisons un dernier algorithme GMM sparses. La méthode VSCC (Variable selection for clustering and classification) de Andrews and McNicholas (2014) se décrit en quelques étapes.

1. Soit une constante c, itérer pour $c = 1, ..., 5$:
   (a) trouver une partition en K classes à l'aide des GMM ;
   (b) calculer la variance inter-classes v_j de chaque variable (définie Équation (2.1)) ;
   (c) sélectionner la variable ayant la plus petite variance inter-classes v_j et la placer dans l'ensemble $\mathcal{S}_c$ (c est défini en dessous) des variables sélectionnées ;
   (d) sélectionner la prochaine variable $\mathbf{x}^l$ comme étant celle qui a la plus petite variance inter-classes parmi les variables qui n'ont pas été testées, et la placer dans $\mathcal{S}_c$ si :

$$\text{cor}(\mathbf{x}^j, \mathbf{x}^l) < 1 - v_j^c, \ \forall \ \mathbf{x}^j \in \mathcal{S}_c, \tag{2.16}$$

   où v_j^c désigne v_j à la puissance c ;
   (e) répéter l'étape précédente jusqu'à ce que toutes les variables soient testées.
2. Pour chaque sous-ensemble de variables $\mathcal{S}_1, ..., \mathcal{S}_5$, construire un GMM et sélectionner le meilleur à l'aide du BIC (modifié, voir Andrews and McNicholas (2014)).

Le seuil défini par l'Équation (2.16) empêche l'algorithme de sélectionner des variables redondantes, c'est-à-dire des variables qui partagent une structure commune. Les auteurs remarquent qu'il faut normaliser les données pour que les variables aient les mêmes variances. Dans l'article, les variables ont été centrées réduites.

Nous pouvons faire plusieurs remarques au sujet de cet algorithme. Premièrement il est facile à implémenter et peu coûteux en temps de calcul. Deuxièmement, l'algorithme n'introduit pas de paramètres supplémentaires. En revanche, le choix de la contrainte (2.16) semble assez arbitraire tout comme l'est le choix du nombre d'itérations (cinq). De plus, si deux variables sont identiques, c'est-à-dire si $\text{cor}(\mathbf{x}^j, \mathbf{x}^l) = 1$, au plus une des deux pourra être sélectionnée ce qui peut être problématique si l'on souhaite sélectionner toutes les variables importantes (et donc problématique d'un point de vue de l'interprétabilité et de l'importance des variables).

2.4.6 Résumé

Deux types d'approches sont généralement utilisés pour effectuer de la sélection de variables dans le contexte des modèles de mélange en clustering sparse. D'une part, les approches basées sur la sélection de modèles présentent l'avantage évident d'être entièrement automatiques puisqu'elles sélectionnent automatiquement les variables à retenir. Leurs implémentations actuelles, basées sur des stratégies *forward-backward*, peuvent cependant empêcher leur application à des données de très haute dimension. D'autre part, les approches basées sur des pénalisations de type lasso sont généralement rapides et ont de bons résultats même sur des données de très grande dimension. Cependant, la sélection du paramètre de sparsité λ reste une question ouverte.

2.5 Description des packages `R` existants

Une partie des méthodes évoquées dans ce chapitre disposent d'un code disponible gratuitement et publiquement. Elles sont pour la majeure partie codées en `R` sous forme de package disponible sur le `CRAN`*. Certaines implémentations ont été publiées dans le Journal of Statistical Software†. Quelques unes ne sont pas disponibles sur le `CRAN` mais on peut trouver un code en accès libre sur `GitHub`‡.

La Table 2.1 décrit et compare un ensemble des packages `R` permettant de faire de la sélection de variables en clustering. Plusieurs critères sont comparés, pour savoir notamment, si l'algorithme implémenté est utilisable en grande dimension c'est-à-dire lorsque $p \gg n$ et si la fonction pondère les variables selon leur importance pour partitionner les données. Une information supplémentaire est donnée : la fonction est-elle adaptée au clustering de données mixtes, c'est-à-dire des données mélangeant des variables numériques et catégorielles. De plus, notre package `R` `vimplcust` est introduit dans le tableau. `vimplcust` permet de faire de la sélection de variables, sur des données structurées en groupes de variables ou sur des données mixtes. Une présentation de ce package est faite dans le Chapitre 3.

Des indications sur le coût algorithmique des méthodes et leur temps de calcul auraient été les bienvenues mais malheureusement cela dépend de beaucoup de paramètres, tels que le nombre d'observations et de variables et autres hyperparamètres des méthodes, qui interagissent les uns avec les autres et donc une simple évaluation unidimensionnelle peut s'avérer compliquée et réductrice. Néanmoins une appréciation est donnée dans la partie présentant les simulations.

Différentes simulations pour comprendre le comportement des différentes méthodes sont proposées dans les sections suivantes.

*https ://cran.r-project.org/
†https ://www.jstatsoft.org/index
‡https ://github.com/

Méthode	Famille	code	nom package	Fonction	Données	$p \gg n$	Importance
(1) Witten and Tibshirani (2010)	KM	R-package	sparcl	KMeansSparseCluster	num	oui	oui
(2) Kondo et al. (2012)	KM	R-package	RSKC Kondo et al. (2016)	RSKC	num	oui	oui
(3) Brodinová et al. (2019)	KM	GitHub	brodsa/wrsk	ROBIN	num	oui	oui
(4) Huo and Tseng (2017)	KM	GitHub	Caleb-Huo/IS-Kmeans	ISKmeans	num	oui	oui
Chavent et al. (2020)	KM	R-package	vimpclust	sparsewkm	mix	oui	oui
Chavent et al. (2020)	KM	R-package	vimpclust	groupsparsewkm	num	oui	oui
(5) Bouveyron and Brunet-Saumard (2014a)	GMM	R-package	FisherEM	sfem	num	oui/non	non
(6) Maugis et al. (2009b)	GMM	R-package	clustvarsel Scrucca and Raftery (2018)	clustvarsel	num	oui	non
(7) Celeux et al. (2019)	GMM	R-package	SelvarMix Sedki et al. (2014)	SelvarClustLasso	num	oui	non
(8) Marbac and Sedki (2017)	GMM	R-package	VarSelLCM	VarSelCluster	mix	oui	non
(9) Andrews and McNicholas (2014)	GMM	R-package	vscc	vscc	num	oui	oui

Table 2.1 : Liste de packages codés en R implémentant des méthodes qui effectuent du clustering sparse dont certaines permettent le clustering de données mixtes, du clustering pondérés, du clustering en grande dimension, du clustering sur des sous-espaces. Tous les algorithmes sont basés sur l'algorithme des K-means, noté KM, ou sur les modèles de mélange gaussien, noté GMM. La colonne "Données" indique si l'algorithme traite uniquement les données numériques ("num") ou le cas des données mixtes ("mix"). La colonne "$p \gg n$" indique si l'algorithme traite des données en grande dimension avec $p \gg n$. La colonne "Importance" indique si l'algorithme pondère les variables en fonction de leur importance. L'information "R-package" est indiquée dans la colonne "code" si le package est disponible sur le CRAN, sinon le code est disponible sur GitHub. L'article scientifique décrivant la méthode est indiqué dans la colonne "Méthode" et si le package est décrit dans un article, celui-ci sera cité dans la colonne "nom package". Pour accéder à la page web d'un package CRAN, suivez le lien : "https ://cran.r-project.org/web/packages/nom-du-package" et remplacez "nom-du-package" par le nom du package présent dans la colonne "nom package". Par exemple, "https ://cran.r-project.org/web/packages/vimpclust" vous donne la page web du package vimpclust. Pour accéder à la page web d'un package GitHub, suivez le lien : "https ://github.com/nom-du-package" et remplacez "nom-du-package" par le nom du package présent dans la colonne "nom package".

2.6 Analyse des schémas et des résultats de simulations

2.6.1 Analyse des schémas de simulations

Dans cette section, nous comparons et analysons les schémas de simulations proposés dans les articles évoqués dans ce chapitre portant sur l'importance et la sélection de variables. Certains articles partagent intentionnellement les mêmes schémas de simulations ce qui facilite la comparaison entre les méthodes et surtout cela permet d'éviter la défiance quant au choix du schéma de simulation.

LE MODÈLE GLOBAL Les simulations ont pour but d'évaluer et de comparer des algorithmes de clustering avec sélection de variables. Le schéma de simulation proposé est le plus utilisé dans la littérature, il a été proposé dans Witten and Tibshirani (2010), et on retrouve deux groupes de variables :

- le groupe des variables importantes qui ont une structure de clustering et dans notre cas, c'est un mélange de gaussiennes. On note p_K le nombre de variables importantes.
- le groupe des variables de bruit qui sont indépendantes du clustering, indépendantes des variables importantes et indépendantes entre elles. Elles suivent le plus souvent une distribution gaussienne (sphérique) et plus rarement une distribution uniforme. On note d le nombre de variables de bruit indépendantes.

Par conséquent on obtient $p = p_K + d$. Le modèle sous-jacent global s'écrit comme un mélange de K gaussiennes :

$$\sum_{k=1}^{K} \frac{1}{K} \mathcal{N}(\mu_k, \mathbf{I}_p) \text{ avec } \mu_k = (\mathbf{m}_k, 0, ..., 0)^\intercal \in \mathbb{R}^{p_K+d} \text{ et } \mathbf{m}_k \in \mathbb{R}^{p_K}, \tag{2.17}$$

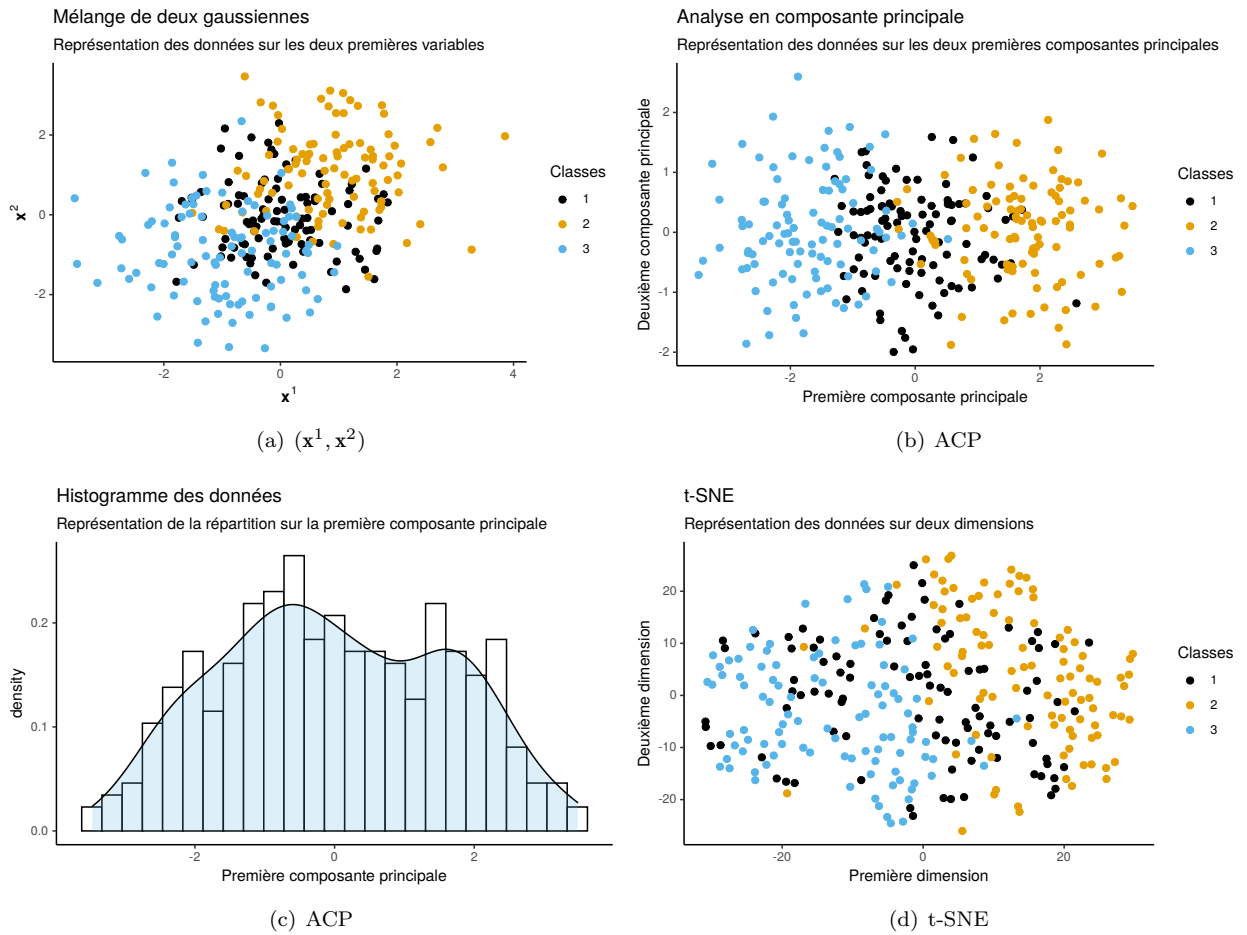
Ce mélange gaussien décrivant les variables peut être de deux types :

1. un mélange de deux gaussiennes sphériques équiprobables, $K = 2$:
   - $\sum_{k=1}^{2} \frac{1}{2} \mathcal{N}(\mu_k, \mathbf{I}_p)$;
   - $\mu_1 = -\mu_2 = (\mathbf{m}_1, 0, ..., 0)^\intercal$;
   - et donc $\mathbf{m}_1 = -\mathbf{m}_2 = (m, ..., m)^\intercal$.
2. un mélange de trois gaussiennes sphériques équiprobables, $K = 3$:
   - $\sum_{k=1}^{3} \frac{1}{3} \mathcal{N}(\mu_k, \mathbf{I}_p)$;
   - $\mu_1 = -\mu_2 = (\mathbf{m}_1, 0, ..., 0)^\intercal$;
   - $\mu_3 = \mathbf{0}_p$ et donc les clusters sont alignés sur un axe.

La matrice des n observations est alors $\mathbf{X} = [\mathbf{X}_{p_K} | \mathbf{X}_d] \in \mathbb{R}^{n \times p}$ où $\mathbf{X}_{p_K}$ est la matrice des p_K variables importantes et $\mathbf{X}_d$ la matrice des d variables de bruit. Un troisième type de mélange apparaît plus rarement dans la littérature : un mélange de quatre gaussiennes sphériques équiprobables disposées en carré lorsque $p_K = 2$ (et disposées sur le plan diagonal d'un hypercube en plus grande dimension). Les articles ne débattent pas de l'intérêt d'utiliser un schéma plutôt que l'autre. En outre, certains articles ajoutent aussi des variables dites redondantes comme ceux de Maugis et al. (2009a,b) ; Celeux et al. (2019), mais ce type de schéma ne sera pas abordé. On se restreint ici aux deux cas listés, ce qui est le schéma de simulations proposé par Witten and Tibshirani (2010) (avec $K = 3$) et c'est aussi celui qui est le plus repris dans la littérature (Arias-Castro and Pu, 2017 ; Bouveyron and Brunet-Saumard, 2014a ; Marbac and Sedki, 2017 ; Celeux et al., 2014) tandis que d'autres schémas s'en rapprochent beaucoup (Chakraborty and Das, 2019 ; Chakraborty et al., 2020). Dans Pan and Shen (2007), le schéma de simulation est similaire à celui de Witten and Tibshirani (2010) en tous points, sauf que les clusters n'ont pas le même nombre d'observations.

Par la suite, nous nommerons *schéma de simulation* (ou simplement *schéma*) le modèle global décrit par l'Équation 2.17. Par ailleurs, un *schéma de simulation* permet de simuler plusieurs *scénarios* en fixant différents paramètres p_K, d, m.

À partir du schéma 1 ($K = 2$), plusieurs scénarios ont été proposés avec par exemple $(p_K, d) = (5, 25)$, $(p_K, d) = (5, 100)$, $(p_K, d) = (50, 100)$ et $n = 30, 300$ (Bouveyron and Brunet-Saumard, 2014a ; Marbac and Sedki, 2017 ; Celeux et al., 2014) , ou encore $(p_K, d) = (50, 200)$, $(p_K, d) = (50, 500)$, (Arias-Castro and Pu, 2017) avec $n = 60, 90$, et m prend généralement ses valeurs entre 0.5 et 2.



(a) $(\mathbf{x}^1, \mathbf{x}^2)$ (b) ACP

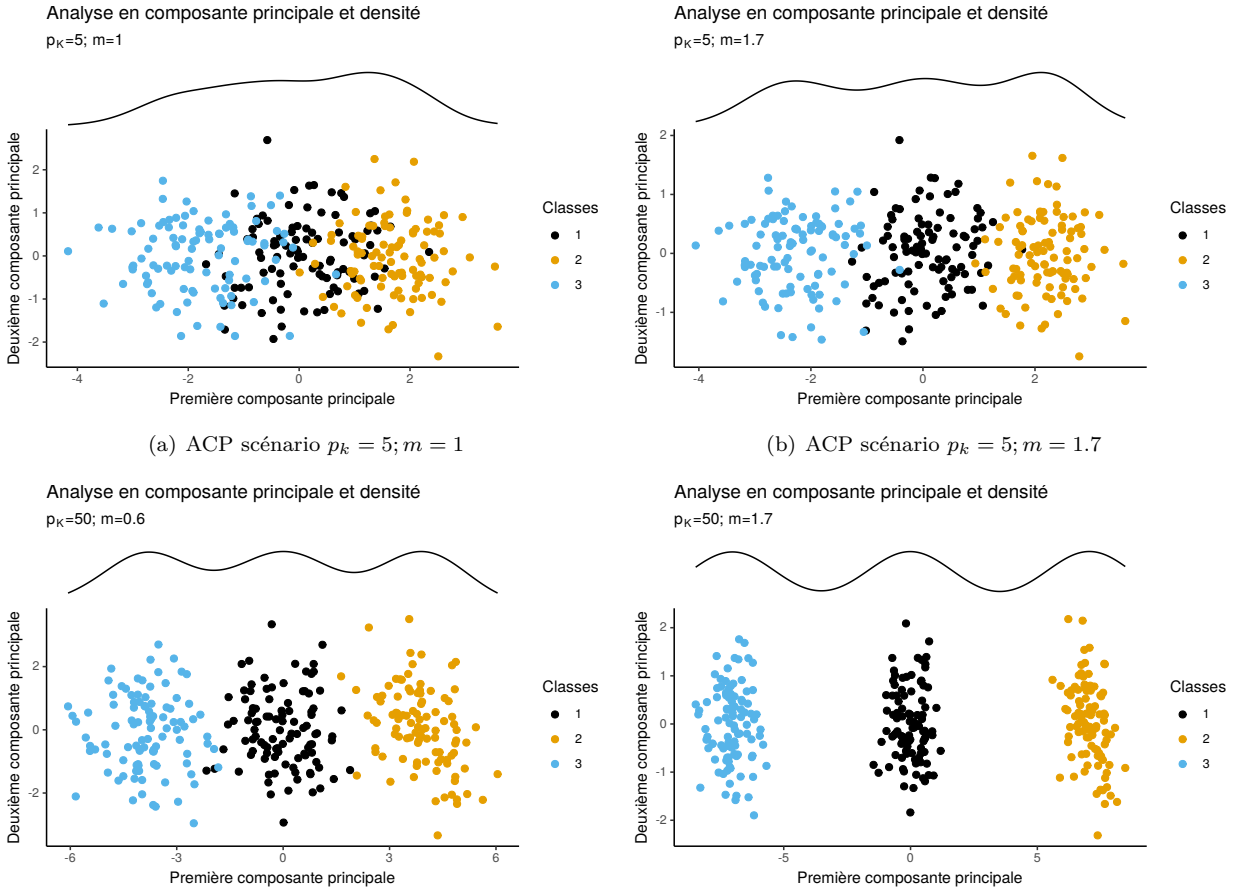
(c) ACP (d) t-SNE

Figure 2.2 : Le graphique (a) représente les deux premières variables importantes des données, avec indication des classes. Le graphique (b) les deux premières composantes principales avec indication des classes. Le graphique (c) représente la répartition des données sur la première composante principale. Le graphique (d) représente les classes dans le sous-espace trouvé pour l'algorithme t-SNE. Objectivement, les classes ne correspondent pas à des clusters.

APPRÉCIATION DU SCHÉMA DE SIMULATION Toutefois un problème demeure dans ces scénarios et il est dû au choix de la constante m. Dans la version originale (Witten and Tibshirani, 2010) $m = 0.6, 0.7, 0.8, 0.9, 1, 1.7$ puis certains auteurs ont testé des valeurs entre ces deux bornes (Arias-Castro and Pu, 2017). Regardons plus précisément un des scénarios proposés avec $n = 300, m = 0.6, p_K = 5$ mais fixons $d = 0$. Nous illustrons ce scénario sur la Figure 2.2 avec des données simulées. Il y a quatre graphiques illustrant les données. Le premier graphique (a) représente les deux premières variables importantes des données et il n'y a clairement aucune structure en clusters. Le deuxième graphique (b) représente les deux premières composantes principales d'une Analyse en Composantes Principales (ACP) effectuées sur les p variables (qui sont uniquement des variables importantes c'est-à-dire $p_K = p$). En effet, les centres des classes étant alignés sur les variables importantes, l'axe traversant les centres des trois classes sur les cinq variables importantes est la direction la plus discriminante et c'est aussi la seule. Théoriquement, c'est aussi l'axe de variance maximale car les classes induisent une corrélation entre les variables importantes en raison de l'alignement de leurs centres. Ainsi, en regardant la première composante principale, on a toute l'information disponible sur la séparabilité des classes. Nous pouvons constater que les classes se superposent et en regardant le troisième graphique (c), qui représente la distribution des données sur la première composante principale, nous ne pouvons pas distinguer les classes et la distribution semble unimodale. Le dernier graphique (d) est obtenu à l'aide de l'algorithme t-SNE (t-distributed stochastic neighbor embedding) (Van der Maaten and Hinton, 2008), qui est une technique de réduction de dimension pour la visualisation de données développée par Geoffrey Hinton et Laurens van der Maaten. Il s'agit d'une méthode non linéaire permettant de représenter un ensemble de points d'un espace à grande dimension dans un espace de deux ou trois dimensions. C'est une méthode très utilisée notamment en clustering pour voir si les données ont une structure en clusters ou même pour valider des résultats (Chakraborty and Das, 2019 ; Chakraborty et al., 2020). Encore une fois les classes se chevauchent et ne sont pas séparables. Somme toute, les données semblent ne pas avoir de structure en clusters et il s'avère

que ce scénario est similaire à celui présenté Figure 1.5 (b).

Observons maintenant des scénarios plus simples où $n = 300, d = 0$ avec $p_K = 5, 50$ et $m = 0.6, 1, 1.7$, que nous illustrons sur la Figure 2.3. Nous pouvons constater que les scénarios avec $p_K = 5$ variables induisent une structure ambiguë même pour le cas où les classes sont les plus séparées ($m = 1.7$). Les classes ne sont pas linéairement séparables, les frontières entre les classes sont denses et donc la présence de clusters est contestable. Cela rappelle une nouvelle fois le cas exposé Figure 1.5 (b). Cela n'empêche pas qu'avec $d = 0$, l'algorithme des 3-means obtiennent de très bons résultats au sens de l'ARI. Lorsque $p_K = 50$, l'information sur les classes est beaucoup plus grande et même dans le cas le plus difficile $m = 0.6$, nous pouvons distinguer des clusters, en tout cas de manière plus précise que pour $p_K = 5$.



(a) ACP scénario $p_k = 5; m = 1$

(b) ACP scénario $p_k = 5; m = 1.7$

(c) ACP scénario $p_k = 50; m = 0.6$

(d) ACP scénario $p_k = 50; m = 1.7$

Figure 2.3 : Les quatre graphiques représentent les classes sur la première composante principale et la densité des distributions. Dans le cas (a) on a $p_k = 5; m = 1$, pour (b) $p_k = 5; m = 1.7$, pour (c) $p_k = 50; m = 0.6$, pour (d) $p_k = 50; m = 1.7$.

LE PROBLÈME DE LA NORMALISATION À VARIANCE UNITAIRE Il reste un dernier point à aborder sur le schéma de simulation et il concerne la variance empirique des variables simulées. La variance des variables de bruit est par définition égale à 1 mais il n'en est pas de même pour les variables importantes. La Table 2.2 indique les variances des cinq premières variables importantes. Elles ont toutes des variances supérieures à 1, ce qui pose problème.

Table 2.2 : Table indiquant les variances des cinq premières variables importantes.

	$\mathbf{x}^1$	$\mathbf{x}^2$	$\mathbf{x}^3$	$\mathbf{x}^4$	$\mathbf{x}^5$
$m = 0.6$	1.32	1.33	1.16	1.23	1.25
$m = 1$	1.82	1.56	1.76	1.63	1.99
$m = 1.7$	2.72	2.99	3.30	2.70	2.98

Witten and Tibshirani (2010) n'indiquent pas dans leur article qu'il faut normaliser les variables pour les ramener à une même variance. De surcroît, ce n'est pas la seule méthode qui nécessite une normalisation. Il est

d'usage de normaliser les variables à variance unitaire pour les algorithmes de clustering basés sur la distance euclidienne. En effet, sans normalisation les algorithmes vont trouver des classes à partir de la structure en variance des données plutôt que de la structure en clusters car plus les variables ont de grandes variances, plus elles contribuent à la distance euclidienne et donc aux résultats de l'algorithme de clustering. La procédure itérative des algorithmes de clustering pondérés et sparses (exemple le WT-K-means) amplifie ce phénomène. L'information sur la normalisation à variance unitaire des données n'est pratiquement jamais indiquée dans la littérature sur les K-means sparses. Seuls Arias-Castro and Pu (2017) ; Chakraborty and Das (2019) ; Huo and Tseng (2017) indiquent qu'ils normalisent les données.

Le fait qu'il y ait des variances plus élevées pour les variables importantes comparativement aux variables de bruit donne un clair avantage aux méthodes nécessitant une normalisation, notamment le WT-K-means. Il n'est pas possible de savoir si, dans les simulations faites dans Witten and Tibshirani (2010) ou dans les articles se comparant à cette méthode, les données ont été normalisées en entrée de l'algorithme. Cela introduit un biais dans les simulations, car les méthodes nécessitant des données centrées réduites en entrée, révèlent la structure en variance des données plutôt que la structure en clusters.

2.6.2 Analyse des résultats de simulations dans la littérature

Les simulations dans plusieurs articles ont été étudiées (Witten and Tibshirani, 2010 ; Raftery and Dean, 2006 ; Pan and Shen, 2007 ; Maugis et al., 2009a,b ; Sun et al., 2012 ; Bouveyron and Brunet-Saumard, 2014a ; Celeux et al., 2014 ; Arias-Castro and Pu, 2017 ; Chakraborty and Das, 2019), et plusieurs points importants sont à souligner :

- les méthodes de clustering sparse présentées dans ce chapitre ont été comparées dans la littérature, aux K-means standards et aux GMM. Les performances des méthodes sparses étaient toujours équivalentes ou supérieures à celles des méthodes classiques. En outre, plus le nombre de variables de bruit augmente, notamment par rapport au nombre de variables importantes, plus les méthodes sparses sont intéressantes.
- le WT-K-means fait office de méthode de référence pour comparer les méthodes de clustering sparse. Dans l'ensemble aucune méthode n'a montré un avantage significatif en termes de partitionnement sur des données simulées.
- pour évaluer la sélection de variables, le plus souvent dans les résultats de simulations, seul le nombre de coefficients non nuls est affiché. Par conséquent, il est difficile de juger de la qualité de la sélection des variables importantes et de la discrimination des variables de bruit. Seuls Celeux et al. (2014) évalue la sélection de variables à l'aide du taux d'erreur de variables importantes sélectionnées et du nombre moyen de variables sélectionnées, ce qui est suffisamment informatif.
- le WT-K-means a de mauvais résultats en termes de sélection de variables car il sélectionne généralement trop de variables de bruit et c'est sans doute dû à un choix inapproprié du paramètre λ par la fonction Gap Statistic. Par ailleurs, le fait que le WT-K-means obtienne de bons résultats en termes de partitionnement implique que le modèle $\lambda = 0$ est souvent choisi, pour lequel les variables de bruit ont un poids non nul mais proche de 0 mais et elles n'influent pas sur le clustering.

En fait, les résultats de simulations gagneraient en clarté si des mesures d'évaluation de sélection de variables plus fines que le seul nombre total de variables sélectionnées étaient utilisées.

2.7 Simulations : comparaison des méthodes de clustering sparse

Dans cette section, nous allons comparer sept méthodes différentes de clustering sparse. En dehors de ces sept méthodes, plusieurs méthodes qui ont été décrites dans la Section 2.3 et la Section 2.4 n'ont malheureusement pas pu être évaluées notamment la méthode de Pan and Shen (2007) car il n'y a pas de code disponible implémentant la méthode. Par ailleurs, le code implémentant la méthode LW-K-means (Chakraborty and Das, 2019) (Section 2.3.6) est disponible, mais il n'y a aucune méthodologie proposée pour choisir les hyper-paramètres et particulièrement le paramètre λ et donc il est impossible d'automatiser l'analyse comparative. Enfin, il n'y a pas non plus de code disponible pour le Regularized K-means (Sun et al., 2012) et même si l'algorithme paraît facile à implémenter, la solution du problème d'optimisation n'est pas donnée dans l'article et celle-ci semble mal définie (voir 2.3.5). Les sept méthodes comparées sont les suivantes :

1. WT-K-means (sparcl) - méthode (1) : c'est l'algorithme décrit dans Witten and Tibshirani (2010) et implémenté dans `sparcl`. Le paramètre λ est choisi au moyen de la méthode du Gap Statistic (défini Section 2.3.2 Équation (2.5)). Le nombre d'itérations pour la convergence du K-means est fixé à 100,

le nombre de valeurs du paramètre λ à tester est 10 et les valeurs sont choisies automatiquement. Le nombre de permutations pour le Gap Statistic est pris par défaut à 25. Le nombre d'initialisations à tester pour le K-means est fixé à 10.

2. WT-K-means (WT-K-means rupture) : c'est l'algorithme décrit dans Witten and Tibshirani (2010) où le paramètre λ est choisi par une méthode de détection de rupture (voir Section 2.3.2). Le nombre d'itérations pour la convergence du K-means est fixé à 100 et le nombre de valeurs de λ à tester est 10. Le nombre de d'initialisation à tester pour le K-means est fixé à 10. Cette algorithme est implémenté dans une version bêta de notre package `vimpclust`.

3. Sparse Fisher-EM (SFEM) - méthode (5) : c'est l'algorithme décrit dans Bouveyron and Brunet (2012) et implémenté dans `FisherEM`. Toutes les formes de *discriminative latent mixture* sont testées. La méthode de régression est utilisée pour modéliser la matrice de projection. 10 valeurs de pénalité lasso sont testées sur une grille entre 0 et 1 (le minimum et le maximum) par pas de 0.1. La valeur de la pénalité L_2 (elastic-net) est laissée à 0. Le K-means initialise la méthode et le nombre d'initialisations à tester pour le K-means est fixé à 10. Le reste des paramètres est pris par défaut.

4. Raftery et Dean (clustvarsel) - méthode (6) : c'est l'algorithme décrit dans Maugis et al. (2009b)et implémenté dans `clustvarsel` Scrucca and Raftery (2018). La méthode *greedy* (*stepwise forward-backward*) est utilisée lorsque cela est possible et sinon on utilise la méthode *headlong*. Pour le reste on utilisera les paramètres par défaut.

5. Modèle SRUW (SelvarMix) - méthode (7) : c'est l'algorithme décrit dans Celeux et al. (2019) et implémenté dans `SelvarMix` Sedki et al. (2014). La méthode lasso est donc utilisée avec une grille de valeurs de paramètres λ et γ définies automatiquement par l'algorithme. Toutes les formes de GMM sont testées, des plus contraintes aux plus générales. Les trois formes possibles de la matrice de covariance pour le modèle de régression sont testées (sphérique, diagonale et générale). Les deux formes possibles de la matrice de covariance pour les variables de bruit sont testées (sphérique et diagonale). Le critère de sélection est le BIC.

6. Sélection de variables par maximisation du MICL (VarSelLCM) - méthode (8) : c'est l'algorithme décrit dans Marbac and Sedki (2017) et implémenté dans `VarSelLCM`. Le critère de sélection est le BIC.

7. VSCC (vscc) - méthode (9) : c'est l'algorithme décrit dans Andrews and McNicholas (2014) et implémenté dans `vscc`. Celui-ci ne requiert aucun paramètre spécifique.

Pour toutes les méthodes le nombre de clusters est donné à l'avance, égal au nombre de clusters de la partition simulée. Le schéma de simulation présenté dans la section précédente dans l'Équation (2.17) est utilisé avec $K = 2$ clusters et $n = 120$ observations mais pour p_K, d et m on choisit des paramètres différents. Deux scénarios sont testés, le premier avec $p_K = 2, d = 20, m = 1.5$, qui est un scénario avec peu de variables ce qui permet de tester les versions les plus performantes des algorithmes, très coûteuses en temps de calcul. Mais $p_K = 2$ permet aussi de visualiser directement les clusters pour s'assurer de la structure des données simulées. Pour le deuxième scénario, $p_K = 10; d = 100; m = 0.85$ et c'est donc un scénario avec plus de variables. Nous représentons les scénarios simulés à l'aide de la Figure 2.4 et les classes représentent bien des clusters séparés.

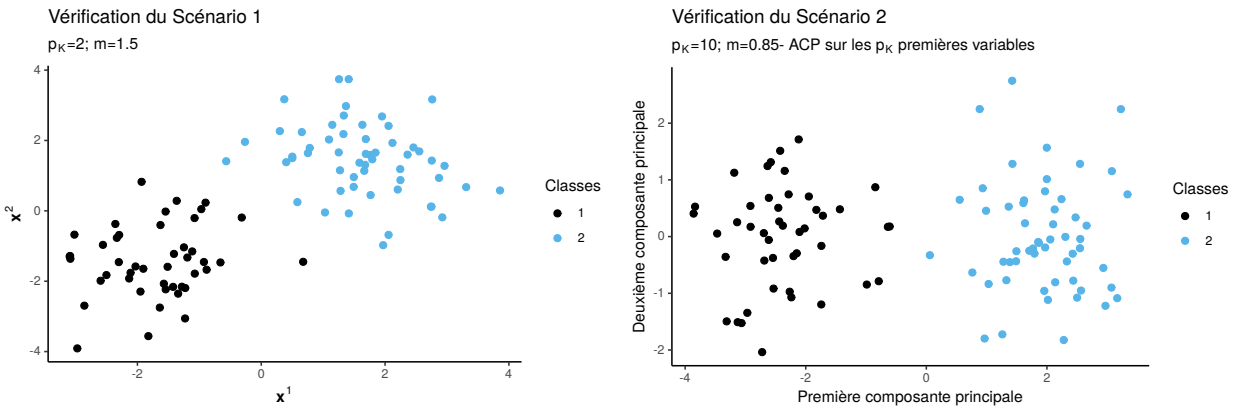
Soit $\mathcal{S}$ l'ensemble des indices des variables sélectionnées par un algorithme de clustering sparse.

- $\mathcal{S}^C = \{j : \mathbf{x}^j \in \mathbf{X}_{p_K}, j \in \mathcal{S}\}$ l'ensemble des indices des variables importantes sélectionnées par l'algorithme.
- $\mathcal{S}^N = \{j : \mathbf{x}^j \in \mathbf{X}_d, j \in \mathcal{S}\}$ l'ensemble des indices des variables de bruit sélectionnées par l'algorithme.
- $p_{\mathcal{S}^C} = \text{card}(\mathcal{S}^C)$ le nombre de variables importantes sélectionnées par l'algorithme.
- $p_{\mathcal{S}^N} = \text{card}(\mathcal{S}^N)$ le nombre de variables de bruit sélectionnées par l'algorithme.

Plusieurs mesures sont utilisées pour comparer les méthodes :

- l'ARI (défini Section 1.4 Équation (1.1)),
- le ratio de variables importantes sélectionnées $\frac{p_{\mathcal{S}^C}}{p_K}$,
- le ratio de variables de bruit sélectionnées $\frac{p_{\mathcal{S}^N}}{d}$,
- le temps de calcul des algorithmes en secondes.

20 simulations des scénarios seront testées et agrégées et la moyenne et l'écart type des résultats sont calculées.



(a) $(\mathbf{x}^1, \mathbf{x}^2)$ scénario 1 (b) ACP scénario 2

Figure 2.4 : Le graphique (a) représente les classes définies sur les deux premières variables importantes. Le graphique (b) représente les classes sur les deux premières composantes principales. Les classes représentent bien des clusters séparés.

REMARQUE SUR LE TEMPS DE CALCUL Les scénarios exacts tels que définis par Witten and Tibshirani (2010) dans leur article n'ont pas pu être testés car les méthodes sont trop coûteuses en temps de calcul. Lorsque $p_K = 2, d = 20$ c'est la version *greedy* (*forward-backward* en commençant dans la direction *forward* c'est-à-dire en commençant avec le *empty model*) du package `clustvarsel` qui a été utilisée et lorsque $p_K = 10; d = 100$ c'est la version *headlong* qui a été utilisée, car la version *greedy* était trop coûteuse en temps de calcul. Les packages `VarSelLCM`, `Selvarmix` et `clustvarsel` (version *greedy*) ont une option permettant la parallélisation des fonctions et celle-ci est utilisée uniquement pour le scénario 2 avec un nombre de cœurs fixé à 40. Cela leur donne un avantage certain et une comparaison plus précise sera faite dans des travaux futurs. La version bêta de notre package `vimpclust` dispose d'une option de parallélisation mais elle n'a pas été utilisée.

2.7.1 Résultats scénario 1 : $K = 2; n = 120; p_K = 2; d = 20; m = 1.5$

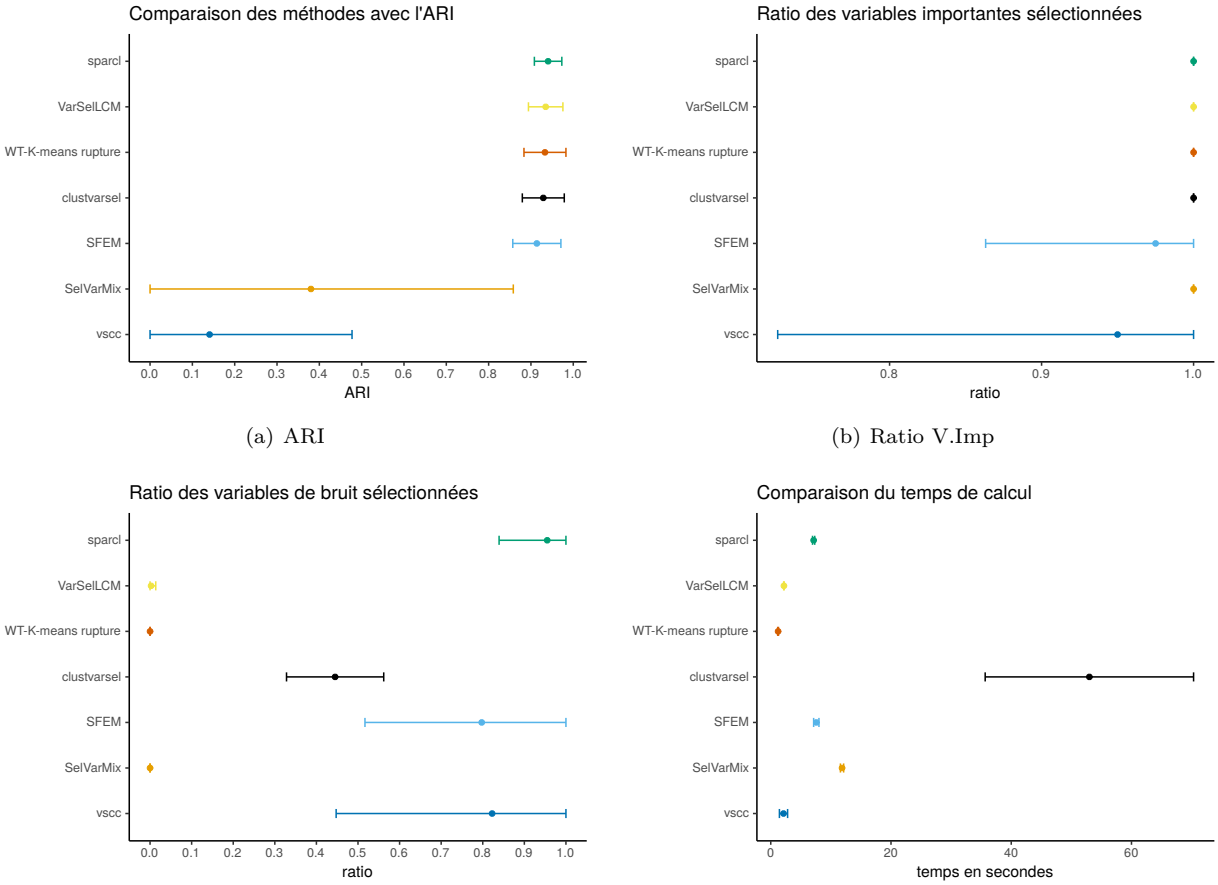
Table 2.3 : Le tableau représente les moyennes et écarts-types par méthode pour le scénario $K = 2, n = 120, p_K = 2, d = 20, m = 1.5$ sur 20 simulations de l'ARI, le ratio de variables importantes (Ratio V.Imp) et de bruit (Ratio V.Bruit) sélectionnées et le temps de calcul (Temps) en secondes.

	ARI		Ratio V.Imp		Ratio V.Bruit		Temps	
noms	moyenne	sd	moyenne	sd	moyenne	sd	moyenne	sd
sparcl	**0.94**	0.03	**1.00**	0.00	0.96	0.12	7.13	0.18
VarSelLCM	**0.93**	0.04	**1.00**	0.00	**0.00**	0.01	2.19	0.04
WT-K-means rupture	**0.93**	0.05	**1.00**	0.00	0.00	**0.00**	**1.22**	0.07
clustvarsel	**0.93**	0.05	**1.00**	0.00	0.45	0.12	53.00	17.33
SFEM	**0.91**	0.06	**0.97**	0.11	0.80	0.28	7.57	0.44
SelVarMix	0.38	0.48	**1.00**	0.00	**0.00**	0.00	11.86	0.27
vscc	0.14	0.34	**0.95**	0.22	0.82	0.38	2.12	0.68

Pour résumer, la Figure 2.5 et la Table 2.3 indiquent que les méthodes sont équivalentes pour ce scénario en termes d'ARI et de sélection de variables importantes. Par contre seules les méthodes VarSelLCM, WT-K-means rupture et SelvarMIX ne sélectionnent pas de variables de bruit ce qui leur donnent un clair avantage en termes d'interprétabilité. De plus la méthode WT-K-means rupture est en moyenne 6 fois plus rapide que sparcl, du fait de l'utilisation du Gap Statistic, 2 fois plus rapide que VarSelLCM, 6 fois plus rapide que SFEM, 10 fois plus rapide que SelvarMix et 43 fois plus rapide que clustvarsel.

2.7.2 Résultats scénario 2 : $K = 2; n = 120; p_K = 10; d = 100; m = 0.85$

Pour résumer, la Figure 2.6 et la Table 2.4 indiquent, comme pour le scénario précédent, que les méthodes sont équivalentes pour ce scénario en termes d'ARI et de sélection de variables importantes ce qui est aussi la conclusion de tous les articles comparant (en partie) les différentes méthodes. Par contre, seules les méthodes VarSelLCM et WT-K-means rupture ne sélectionnent pas de variables de bruit. Le WT-K-means combiné au Gap Statistic (sparcl) sélectionne beaucoup de variables de bruit ce qui coïncide avec les résultats des



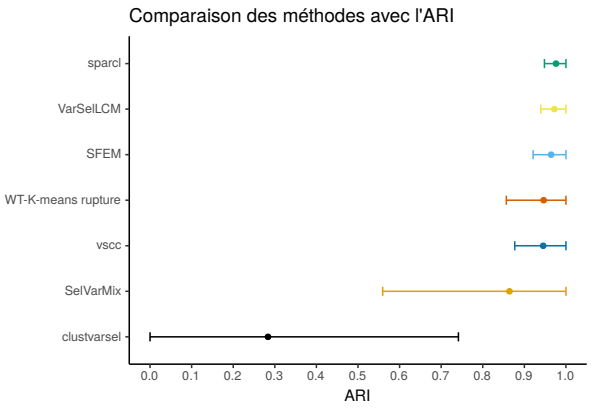
(a) ARI (b) Ratio V.Imp

(c) Ratio V.Bruit (d) Temps

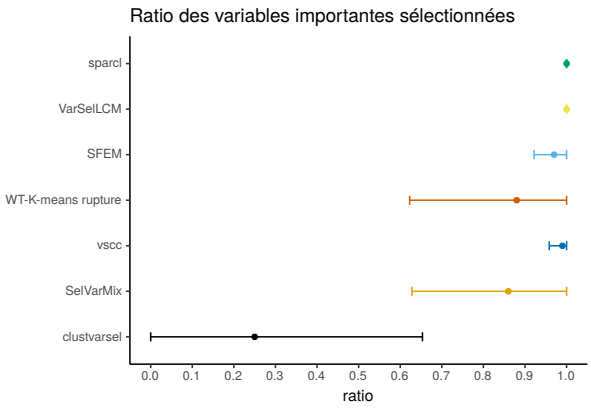
Figure 2.5 : Les quatre graphiques représentent les moyennes et écarts-types par méthode pour le scénario $K = 2, n = 120, p_K = 2, d = 20, m = 1.5$ sur 20 simulations de l'ARI (a), le ratio de variables importantes (Ratio V.Imp) (b) et de bruit (Ratio V.Bruit) (c) sélectionnées et le temps de calcul en secondes(d).

Table 2.4 : Le tableau représente les moyennes et écarts-types par méthode pour le scénario $K = 2, n = 120, p_K = 10, d = 100, m = 0.85$ sur 20 simulations de l'ARI, le ratio de variables importantes (Ratio V.Imp) et de bruit (Ratio V.Bruit) sélectionnées et le temps de calcul (Temps) en secondes.

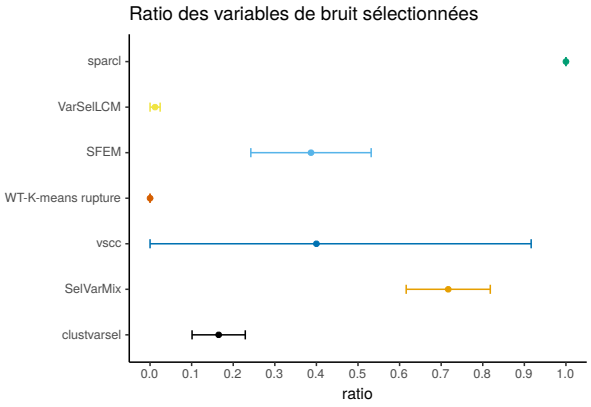
	ARI		Ratio V.Imp		Ratio V.Bruit		Temps	
noms	moyenne	sd	moyenne	sd	moyenne	sd	moyenne	sd
sparcl	**0.98**	0.03	**1.00**	0.00	1.00	0.00	5.07	0.55
VarSelLCM	**0.97**	0.03	**1.00**	0.00	**0.01**	0.01	4.73	0.37
SFEM	**0.96**	0.04	**0.97**	0.05	0.39	0.14	39.88	4.14
WT-K-means rupture	**0.95**	0.09	**0.88**	0.26	**0.00**	0.00	0.94	0.09
Vscc	**0.95**	0.07	**0.99**	0.03	0.40	0.52	**0.45**	0.13
SelVarMix	**0.86**	0.30	0.86	0.23	0.72	0.10	2.87	0.52
Clustvarsel	0.28	0.46	0.25	0.40	0.17	0.06	17.82	16.98



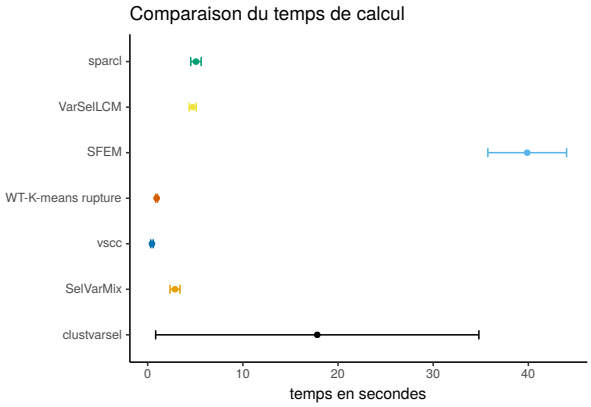
(a) ARI



(b) Ratio V.Imp



(c) Ratio V.Bruit



(d) Temps

Figure 2.6 : Les quatre graphiques représentent les moyennes et écarts-types par méthode pour le scénario $K = 2, n = 120, p_K = 10, d = 100, m = 0.85$ sur 20 simulations de l'ARI (a), le ratio de variables importantes (Ratio V.Imp) (b) et de bruit (Ratio V.Bruit) (c) sélectionnées et le temps de calcul en secondes (d).

simulations des autres articles. De plus la méthode vscc est la plus rapide, la méthode WT-K-means rupture est en moyenne deux fois et demi plus lente que vscc et toutes les autres méthodes sont dix à cent fois plus lentes que vscc.

2.7.3 Conclusion sur les simulations

Nous retrouvons globalement les conclusions faites par les auteurs des différents articles étudiés. Les méthodes ont des performances équivalentes en termes d'ARI et de sélection de variables importantes mais certaines méthodes échouent à supprimer les variables de bruit notamment le WT-K-means avec le Gap statistic (sparcl). Pour cette dernière, c'est bien entendu le choix du paramètre λ qui reste difficile. Étant donné que l'ARI est satisfaisant, on en conclut que les poids des variables de bruit sont proches de zéro, mais que le choix inapproprié du paramètre λ fait que les variables de bruit, n'ont pas été supprimées. Après inspection des résultats, le paramètre $\lambda = 0$ est souvent choisi par la méthode Gap Statistic pour sparcl ce qui explique le fait que les variables de bruit ne soient pas discriminées. VarSelLCM et WT-K-means rupture sont les méthodes les plus compétitives notamment en termes d'interprétabilité, d'importance et de sélection de variables. Parmi celles-ci, WT-K-means rupture est en moyenne cinq fois plus rapide que les autres, ce qui justifierait son utilisation sur des ensembles de données de grande taille et de grande dimension.

2.8 Conclusion

Dans ce chapitre, nous avons donc passé en revue des méthodes existantes, basées sur les K-means ou les GMM, permettant d'obtenir un clustering sparse.

Une des difficultés de ce chapitre réside dans la comparaison numérique des méthodes de clustering sparse. En effet, les résultats de simulations dépendent des schémas et des scénarios employés et il est difficile de généraliser même si des comportements semblent se dessiner.

Un des enjeux est de bien différencier l'influence de la méthode de clustering de celle de la sélection de modèle dans les résultats. Cela n'a pas été fait explicitement dans ce chapitre, notamment pour les méthodes GMM, et c'est une amélioration possible.

Ce chapitre n'a pas encore fait l'objet d'une publication, des modifications et des informations supplémentaires doivent être ajoutées dans le but d'en faire un article de *review* et de le soumettre à un journal.

3

Clustering sur des groupes de variables via les K-means sparses et application à des données mixtes

3.1 Introduction 43
3.1.1 Contexte et Motivations 43
3.1.2 Objectifs et contributions 44
3.2 Le modèle WT-K-means et son extension avec la pénalité *group lasso* 44
3.2.1 Retour sur le modèle WT-K-means 44
3.2.2 Clustering sparse de groupes de variables 46
3.3 Clustering sparse de données mixtes 47
3.3.1 Le clustering de données mixtes 47
3.3.2 Méthodes existantes pour traiter les données mixtes en clustering non sparse 48
3.3.3 Recodage des données 48
3.3.4 Group-Sparse K-means appliqué aux données mixtes 49
3.4 Illustration du package `vimpclust` sur des données réelles 50
3.5 Simulations : comparaison des méthodes de clustering sparse sur des données mixtes 51
3.6 Conclusion 52
3.7 Annexe 52
3.7.1 Solution du Group-Sparse-K-means 52

3.1 Introduction

3.1.1 Contexte et Motivations

Comme on l'a vu dans le chapitre précédent, les méthodes de clustering sparse tendent à choisir les variables les plus informatives et permettent de résoudre des tâches de clustering en diminuant le nombre de variables, ce qui améliore les performances et l'interprétabilité. Désormais, on se pose la question de généraliser encore ces méthodes lorsque les variables décrivant les données ont une structure de groupes a priori. Il semble intuitif qu'en intégrant cette information dans la sélection des variables, on pourra obtenir des clusters plus précis et plus proches de la structure sous-jacente.

De nombreux types de données présentent une structuration des variables en groupes, comme par exemple des ensembles de données d'expression génétique, où les gênes ne fonctionnent pas individuellement, mais par paquets. On peut voir à ce sujet les études de Simon et al. (2013) et Burczynski et al. (2006) qui montrent que les gênes sont regroupés en *ensemble de gênes* à l'aide des données de position cytogénétique. On peut aussi trouver dans Higuera et al. (2015), un jeu de données qui décrit les expressions des protéines chez la souris, que nous avons pris comme exemple d'application dans un document de recherche*. On peut également

*https ://cran.r-project.org/web/packages/vimpclust/vignettes/groupsparsewkm.html

évoquer des données de recensement, regroupées en variables personnelles, formation-éducation, variables socio-économiques, etc. Ou encore des données longitudinales où les différentes dates considérées structurent naturellement les variables. Nous avons aussi les données de tests de production, où des mesures sont faites pour plusieurs niveaux de puissance du moteur, formant ainsi des groupes de variables décrivant un quantité physique mesurée sur différents niveaux de poussée.

Un cas important de données ayant une structure de groupes est celui des données mixtes, c'est-à-dire mélangeant des variables numériques et catégorielles, ainsi que les données purement catégorielles. En effet, une caractéristique de ces données est qu'elles conduisent à définir naturellement des groupes de variables, en remplaçant une variable catégorielle $\mathbf{x}^j$ à c_j modalités par un groupe de c_j variables indicatrices numériques. Tous ces exemples montrent l'intérêt d'étudier les méthodes de clustering qui prennent en compte l'information de l'existence de groupes de variables.

Dans le chapitre précédent nous avons mentionné une méthode applicable à des données dont les variables ont une structure de groupe. Huo and Tseng (2017) proposent d'étendre l'algorithme WT-K-means à l'aide d'une pénalité dite *overlapping group* (Jacob et al., 2009) ce qui permet le chevauchement des groupes tout en sparsifiant au sein des groupes (Section 2.3.3). Mais si la procédure d'optimisation employée est convaincante en régression, elle n'est pas bien adaptée au clustering et selon nous il faudrait la simplifier. Une autre méthode de clustering sparse basée sur les GMM et applicable à des données mixtes a été présentée dans la Section 2.4.4 : la méthode du package `VarSelLCM` (Marbac and Sedki, 2017 ; Marbac et al., 2020), qui permet de sélectionner des variables en déterminant un vecteur de poids binaire en maximisant le critère MICL (*Maximum Integrated Complete-data Likelihood*). À notre connaissance, il n'y a pas de méthode de clustering sparse basée sur le K-means qui soit adaptée aux données mixtes.

3.1.2 Objectifs et contributions

Le but de ce chapitre est de proposer une méthode de clustering sparse basée sur les K-means et adaptée aux données dont les variables ont une structure de groupes et en particulier aux données mixtes.

Notre contribution décrite dans (Chavent et al., 2020), consiste à proposer une extension de l'algorithme du WT-K-means (Witten and Tibshirani, 2010), que nous appellerons Group-Sparse K-means, à des données numériques structurées en groupes de variables, en introduisant une pénalité dite *group lasso* (Yuan and Lin, 2006). Par ailleurs, nous définissons une transformation des données mixtes adaptée au clustering, et nous appliquons la méthode Group-Sparse K-means aux données transformées. Enfin, nous utilisons la méthode de détection de rupture présentée au chapitre précédent (Section 2.3.2) pour choisir le paramètre de pénalisation λ du Group-Sparse K-means. Un package `R` implémentant notre méthode a été développé et publié sur le `CRAN` et il est nommé `vimpclust`[†].

Par souci de clarté, les remarques et les réflexions préliminaires sur la normalisation des données, l'initialisation des algorithmes, la sélection de modèles et l'analyse de l'état de l'art (méthodes, packages, simulations, résultats) ont été présentées au chapitre précédent. Ainsi, ce chapitre est consacré à la définition de l'algorithme proposé.

3.2 Le modèle WT-K-means et son extension avec la pénalité *group lasso*

Dans cette section, on rappelle la définition de l'algorithme WT-K-means (défini dans la Section 2.3) et ensuite on présente l'extension aux variables groupées.

3.2.1 Retour sur le modèle WT-K-means

FORMULATION DU MODÈLE WT-K-MEANS Soit $\mathbf{X}$ la matrice $n \times p$ des n observations décrites par p variables numériques. Comme défini dans la Section 2.3, l'algorithme WT-K-means (Witten and Tibshirani, 2010) consiste à fixer un nombre de classes K a priori et à maximiser la variance inter-classes pondérée et pénalisée. Le problème peut s'écrire sous la forme suivante :

$$\underset{C_1,\dots,C_K,\mathbf{w}}{\text{maximiser}}\ \mathbf{w}^\intercal \mathbf{b} - \lambda \left\|\mathbf{w}\right\|_1 \ \text{ s.c. } \ \left\|\mathbf{w}\right\|_2^2 \leqslant 1, w_j \geqslant 0\ \forall j. \tag{3.1}$$

où l'on rappelle que :

- $\mathbf{w} = (w_1, ..., w_p)^\intercal$ est le vecteur des poids des variables ;
- $\mathbf{b} = (b_1, ..., b_p)^\intercal$ est le vecteur des variances inter-classes par variable pour une partition donnée $C_1, ..., C_K$;
- $\lambda \geqslant 0$ est l'hyperparamètre déterminant l'intensité de la pénalisation.

[†]https ://cran.r-project.org/web/packages/vimpclust/index.html

ALGORITHME ITÉRATIF Le problème (3.1) est un double problème d'optimisation non convexe. Ainsi, la solution envisagée pour optimiser cette équation est un algorithme itératif où les clusters sont trouvés à poids fixés et les poids sont optimisés à clusters fixés. L'algorithme itératif est résumé sous forme de diagramme dans la Figure 3.1 à K et λ fixés.

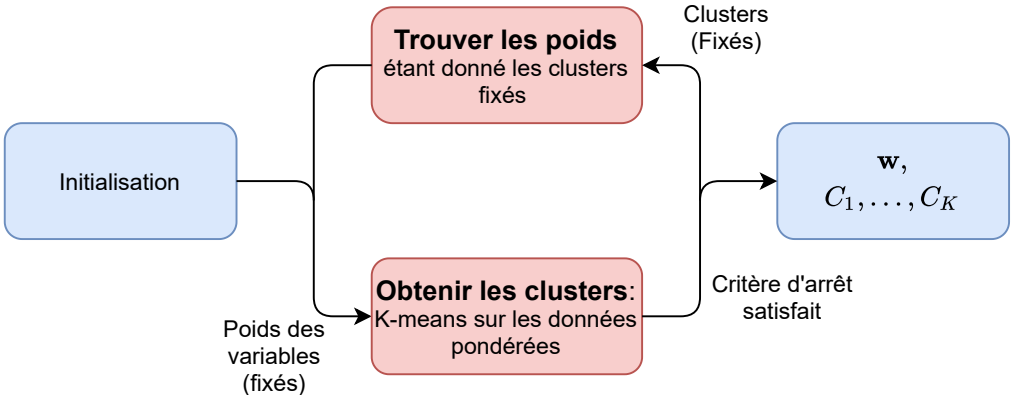


Figure 3.1 : Le graphique illustre l'algorithme itératif du WT-K-means. Les clusters sont trouvés à poids fixés et les poids sont optimisés à clusters fixés.

Pour aboutir à une solution, le double problème d'optimisation est scindé en deux comme détaillé ci-dessous.

- Initialisation : On initialise les poids tous égaux à $w_j = 1/\sqrt{p}, \forall j = 1, ..., p$ ce qui permet de satisfaire les deux contraintes et on répète l'enchaînement suivant.
  1. Obtenir les clusters : $\underset{C_1,\ldots,C_K}{\text{maximiser}}\ \mathbf{w}^\intercal \mathbf{b}$. La pénalité n'apparait plus car les poids sont fixés.
  2. Obtenir les poids : $\underset{\mathbf{w}}{\text{maximiser}}\ \mathbf{w}^\intercal \mathbf{b} - \lambda \|\mathbf{w}\|_1$, s.c. $\|\mathbf{w}\|_2^2 \leqslant 1, w_j \geqslant 0\ \forall j$. Ici les clusters sont fixés donc $\mathbf{b}$ est fixé.
- Arrêt : La procédure itère les étapes 1. et 2. jusqu'à ce que le critère d'arrêt suivant soit satisfait

$$\frac{\left\|\mathbf{w}^r - \mathbf{w}^{r-1}\right\|_1}{\left\|\mathbf{w}^{r-1}\right\|_1} < 10^{-4}, \tag{3.2}$$

où $\mathbf{w}^r = (w_1^r, \ldots, w_p^r)^\intercal$ est le vecteur des poids à l'itération r.

SOLUTIONS Le fonctionnement de l'algorithme itératif est désormais expliqué, mais il reste à donner les solutions du double problème d'optimisation. Il faut comprendre pourquoi d'une part résoudre le problème à poids fixés revient à faire un K-means sur les données pondérées et d'autre part quelle est la solution obtenue pour les poids lorsque les clusters sont fixés.

1. Trouver les clusters : pour trouver les clusters dans l'étape 1, il faut

$$\underset{C_1,\ldots,C_K}{\text{maximiser}}\ \mathbf{w}^\intercal \mathbf{b}.$$

On sait que les K-means maximisent l'inertie inter-classes, or les poids peuvent se factoriser dans l'inertie inter-classes pondérées

$$\mathbf{w}^\intercal \mathbf{b} = \sum_{j=1}^{p} \sum_{k=1}^{K} \frac{n_k}{n} (\sqrt{w_j} \times \bar{x}_k^j - \sqrt{w_j} \times \bar{x}^j)^2.$$

où

- $\bar{x}^j$ la moyenne de la variable j ;
- $\bar{x}_k^j = \frac{1}{n_k} \sum_{i \in C_k} x_i^j$;
- n_k le nombre d'observations dans le cluster k.

L'équation ci-dessus nous apprend que, optimiser l'inertie inter-classes pondérée revient à optimiser l'inertie inter-classes de l'ensemble des données où chaque variable $\mathbf{x}^j$ aurait été multipliée au préalable par $\sqrt{w_j}$. Cela revient donc à effectuer un algorithme K-means sur des variables multipliées par un facteur $\mathbf{w}^{\frac{1}{2}}$.

2. Trouver les poids : pour trouver les poids dans l'étape 2, il faut

$$\underset{\mathbf{w}}{\text{maximiser}}\ \mathbf{w}^\intercal \mathbf{b} - \lambda \left\|\mathbf{w}\right\|_1 ,\ \text{s.c.}\ \left\|\mathbf{w}\right\|_2^2 \leqslant 1, w_j \geqslant 0\ \forall j.$$

Les conditions d'optimalité s'écrivent sous la forme d'un système Karush-Kuhn Tucker (KKT). En utilisant les conditions KKT comme expliqué dans Boyd and Vandenberghe (2004) on obtient :

$$\mathbf{w}^\star = \begin{cases} \mathbf{0} \text{ si } \lambda \geqslant b_j\ \forall j = 1, \ldots, p, \\ \dfrac{S_\lambda(\mathbf{b})}{\left\|S_\lambda(\mathbf{b})\right\|_2} \text{ sinon,} \end{cases}$$

avec l'opérateur de seuillage doux $S_\lambda(\mathbf{b}) = (\mathbf{b} - \boldsymbol{\lambda})_+$, $\boldsymbol{\lambda} = (\lambda, ..., \lambda)^\intercal \in \mathbb{R}_+^p$ et $(\mathbf{x})_+ = (\max(0, x_1), \ldots, \max(0, x_p))^\intercal$ que l'on peut aussi écrire comme $S_\lambda(b_j) = (b_j - \lambda)_+$. Cela signifie qu'une variable j est supprimée du modèle si $\lambda > b_j$ et sinon que son poids est proportionnel à $b_j - \lambda$. Les détails de cette solution sont donnés dans l'annexe, Section 3.7.1.

Dans la suite, nous définissons l'extension de cet algorithme au cas des données ayant une structure de groupes de variables.

3.2.2 Clustering sparse de groupes de variables

INTRODUCTION DE LA PÉNALISATION PAR GROUPES DE VARIABLES Nous généralisons le WT-K-means en introduisant la régularisation par groupes. Cela va permettre de prime abord de sélectionner des groupes de variables numériques.

Supposons que les p variables sont divisées en L groupes connus à l'avance, tels que $\mathbf{X} = \left[\mathbf{X}^1 | \ldots | \mathbf{X}^L\right]$, avec $\mathbf{X}^l \in \mathbf{R}^{n \times p_l}$, p_l étant la taille du groupe l, et $p_1 + ... + p_L = p$. Le vecteur de variance inter-classes $\mathbf{b}$ et le vecteur de poids $\mathbf{w}$ peuvent également être décomposés en $\mathbf{b}^\intercal = (\mathbf{b}_1, \ldots, \mathbf{b}_L)$ et $\mathbf{w}^\intercal = (\mathbf{w}_1, \ldots, \mathbf{w}_L)$.

Pour les données de groupe, on définit une pénalité de groupes ℓ_1 spécifique, qui a déjà été utilisée dans le cadre de la régression par Yuan and Lin (2006),

$$h(\mathbf{w}) = \left\|\mathbf{w}\right\|_{1,group} = \sum_{l=1}^{L} \upsilon_l \left\|\mathbf{w}_l\right\|_2 , \tag{3.3}$$

où $\boldsymbol{\upsilon}^\intercal = (\upsilon_1, ..., \upsilon_L)$ est un vecteur de poids appliqué aux groupes de variables. Dans la littérature sur la régression par *group sparse*, deux choix courants semblent émerger, soit $\upsilon_l = 1$, $\forall l = 1, ..., L$, soit $\upsilon_l = \sqrt{p_l}$ $\forall l = 1, ..., L$. Cette dernière méthode, que nous utilisons par la suite, consiste à pénaliser chaque groupe par sa taille.

NOUVEAU PROBLÈME D'OPTIMISATION Avec les notations précédentes, le nouveau problème d'optimisation s'écrit comme suit :

$$\underset{C_1, \ldots, C_K, \mathbf{w}}{\text{maximiser}}\ \mathbf{w}^T \mathbf{b} - \lambda \sum_{l=1}^{L} \sqrt{p_l} \left\|\mathbf{w}_l\right\|_2 \text{ s.c. } \left\|\mathbf{w}\right\|_2 \leq 1,\ w_j \geqslant 0\ \forall j. \tag{3.4}$$

L'Équation (3.4) représente bien la version *group sparse* du problème WT-K-means (2.4). Néanmoins, ce problème se présente naturellement sous des formes plus générales (Simon et al., 2013) où la norme ℓ_1 et la norme ℓ_2 peuvent être combinées pour former une pénalité connue sous le nom *sparse group lasso* en régression, qui est une extension de la pénalité *elastic-net* (Zou and Hastie, 2005) où la pénalité ℓ_2 est appliquée à des groupes de variables devenant ainsi une pénalité de groupes (Yuan and Lin, 2006).

Dans notre cas, la pénalité *group sparse* entre les groupes de variables donne son nom au modèle Group-Sparse K-means défini par l'Équation (3.4). On peut étendre le modèle (3.4) à la pénalité *sparse group sparse* (*sparse group lasso* en régression) combinant les normes ℓ_1 et ℓ_2 par groupes ce qui donne naissance au modèle Sparse-Group-Sparse K-means ou SGS K-means pour faire court. Ce modèle permet donc une sparsité entre les groupes et à l'intérieur des groupes de variables. Formellement, il s'écrit :

$$\underset{C_1, \ldots, C_K, \mathbf{w}}{\text{maximiser}}\ \mathbf{w}^\intercal \mathbf{b} - \lambda \Big[(1 - \alpha) \sum_{l=1}^{L} \sqrt{p_l} \left\|\mathbf{w}_l\right\|_2 + \alpha \left\|\mathbf{w}\right\|_1 \Big] \text{ s.c. } \left\|\mathbf{w}\right\|_2 \leq 1,\ w_j \geqslant 0\ \forall j, \tag{3.5}$$

où $\lambda \geqslant 0$ est le paramètre déterminant le niveau de pénalisation, $\alpha \in [0, 1]$ le paramètre qui arbitre entre une pénalisation inter-groupes ($\alpha \to 0$) ou intra-groupes ($\alpha \to 1$). En effet, dans le cas extrême où $\alpha = 0$ (respectivement $\alpha = 1$), le modèle (3.5) correspond au modèle Group-Sparse K-means (respectivement WT-K-means).

RÉSOLUTION DU PROBLÈME La résolution du problème décrit par l'Équation (3.5) est similaire à celle qui a été présentée dans la Section 3.2. En résumé, les poids sont initialisés tous égaux à $1/\sqrt{p}$ pour satisfaire les contraintes. Ensuite on itère les deux étapes de l'algorithme itératif (trouver les clusters puis les poids) jusqu'à ce que le critère de convergence défini Section 3.2 soit satisfait. La première étape consiste à trouver les clusters à poids fixés en résolvant (3.5) par rapport à $C_1, ..., C_K$, ce qui revient à exécuter les K-means sur les données pondérées par $\mathbf{w}$. La deuxième étape consiste à trouver les poids à clusters fixés en résolvant (3.5) par rapport à $\mathbf{w}$, et la solution de ce problème est :

$$\mathbf{w}^\star = \begin{cases} \mathbf{0} \text{ si } \|\tilde{S}\big((\mathbf{b}_j - \lambda\alpha)_+, \lambda\sqrt{\mathbf{p}}(1-\alpha)\big)\|_2 = 0, \\ \dfrac{\tilde{S}\big((\mathbf{b}_j - \lambda\alpha)_+, \lambda\sqrt{\mathbf{p}}(1-\alpha)\big)}{\|\tilde{S}\big((\mathbf{b}_j - \lambda\alpha)_+, \lambda\sqrt{\mathbf{p}}(1-\alpha)\big)\|_2} \text{ sinon,} \end{cases}$$

et donc naturellement la solution du problème Group-Sparse-K-means, c'est à dire lorsque $\alpha = 0$ peut s'écrire comme

$$\mathbf{w}^\star = \begin{cases} 0 \text{ si } \|\tilde{S}(\mathbf{b}, \sqrt{\mathbf{p}}\lambda)\|_2 = 0, \\ \dfrac{\tilde{S}(\mathbf{b}, \sqrt{\mathbf{p}}\lambda)}{\|\tilde{S}(\mathbf{b}, \sqrt{\mathbf{p}}\lambda)\|_2} \text{ si } \|\tilde{S}(\mathbf{b}, \sqrt{\mathbf{p}}\lambda)\|_2 \neq 0, \end{cases}$$

où

- $\sqrt{\mathbf{p}}^\intercal = (\sqrt{p_1}, ..., \sqrt{p_L})$,
- $\tilde{S}(\mathbf{b}, \sqrt{\mathbf{p}}\lambda)^\intercal = \left(S_g(\mathbf{b_1}, \sqrt{p_1}\lambda)^\intercal, ..., S_g(\mathbf{b_L}, \sqrt{p_L}\lambda)^\intercal\right) \in \mathbf{R}^p$,
- $S_g(\mathbf{b_l}, \sqrt{p_l}\lambda) = \frac{\mathbf{b_l}}{\|\mathbf{b_l}\|_2}\left(\|\mathbf{b_l}\|_2 - \sqrt{p_l}\lambda\right)_+ \in \mathbf{R}^{p_l} \ \forall l = 1, ..., L$,

et S_g est l'opérateur de seuillage doux par groupe. Cela signifie qu'une variable x^j peut être supprimée uniquement si la norme de son groupe $\|\mathbf{b_l}\|_2 < \sqrt{p_l}\lambda$. Les variables d'un même groupe sont donc sélectionnées toutes ensemble. La résolution des problèmes (2.4), (3.4) et (3.5) est donnée dans l'annexe, Section 3.7.1.

La formulation SGS K-means a été introduite par souci de complétude, d'exhaustivité et de clarté. En effet, cette méthode permet de faire le lien entre plusieurs pénalités : sparse,*group sparse*, *sparse group sparse*, *elastic-net*. Sa résolution donne une solution commune à toutes les méthodes déterminées par ces pénalités. Néanmoins, nous nous restreignons dans la suite à l'utilisation du Group-Sparse K-means ($\alpha = 0$, pénalité *group sparse*).

DISCUSSION SUR LE PARAMÈTRE λ L'opérateur de seuillage doux par groupe supprime du modèle tous les groupes dont la norme de la variance inter-classes correspondante $\mathbf{b_l}$ est inférieure au seuil fixe λ normalisé par la taille du groupe, et réduit d'autant les normes des groupes de variables restants. Une infinité de valeurs du paramètre λ peuvent être choisies mais il est d'usage de se donner une grille de valeurs comprises entre 0 et une valeur maximum λ_{max} à déterminer. Witten and Tibshirani (2010) cherchent le λ_{max} à l'aide d'une fonction de recherche dichotomique[‡] (*binary search* en anglais). Mais on peut noter que pour la version originale de l'algorithme, on a que $\lambda_{max} = \max(\mathbf{b})$ et que pour notre version group-sparse $\lambda_{max} = \max(\frac{\|\mathbf{b_1}\|_2}{\sqrt{p_1}}, ..., \frac{\|\mathbf{b_L}\|_2}{\sqrt{p_p}})$ car au-dessus de ces valeurs tous les coefficients sont mis à zéro.

3.3 Clustering sparse de données mixtes

Cette section est donc consacrée à la définition d'une méthode explicite de clustering sparse, dans le cadre de données mixtes, en utilisant un critère de pénalisation.

3.3.1 Le clustering de données mixtes

Le clustering de données mixtes est complexe, car il porte des variables de deux types différents, des variables numériques et des variables catégorielles qui ont naturellement comme on l'a vu une structure en groupes. Lorsque l'on utilise un algorithme de clustering basé sur la distance euclidienne, la première difficulté du clustering de données mixtes est de définir un type de normalisation qui permette aux variables d'avoir des contributions à la variance équitables. En effet, pour le clustering de variables numériques, si les variables ne

[‡]https ://github.com/cran/sparcl/blob/master/R/SparseClustering.R

sont pas normalisées pour être de même variance, l'algorithme de clustering met en évidence la structure en variance des données plutôt que la structure en clusters.

Des méthodes d'analyse factorielles mixtes (Chavent et al., 2012) existent et exploitent les liens entre les variables de différents types, mais ce type de méthode ne peut pas être utilisé ici. En effet, ces méthodes impliquent la création de nouvelles variables qui sont des combinaisons linéaires des variables de départ, et qui expliquent la structure en variance des données. C'est contradictoire avec le but des algorithmes sparses qui cherchent au contraire à diminuer le nombre de variables. Dans le cas d'analyse factorielle sparse (Witten et al., 2009 ; Chavent and Chavent, 2017), les combinaisons linéaires sparses obtenues pourraient ne pas correspondre à un choix optimal de sous-espace pour le clustering, sans oublier que ces méthodes requièrent le choix d'au moins deux hyperparamètres, le paramètre contrôlant la sparsité et le second fixant la dimension.

3.3.2 Méthodes existantes pour traiter les données mixtes en clustering non sparse

Il existe des méthodes de clustering non sparse de données mixtes, notamment pour les K-means. Les méthodes développées consistent soit à transformer les données en données numériques et utiliser une distance a priori, soit à utiliser un algorithme de pondération des variables catégorielles.

Une première stratégie (recodage) consiste d'abord à transformer les variables catégorielles, puis à appliquer une méthode conçue pour les données numériques. Pour une variable catégorielle $\mathbf{x}^j$ ayant c_j modalités on crée c_j variables indicatrices distinctes, une pour chacune des modalités, prenant les valeurs 0 ou $a \in \mathbb{R}$ et le problème réside dans la sélection de la constante a (Foss et al., 2016). L'approche usuelle qui consiste à fixer $a = 1$ est arbitraire et ne prend pas en compte la structure des données. Une autre approche consiste à normaliser toutes les variables à variance unitaire ce qui, pour les variables catégorielles, correspond à une normalisation par $1/\sqrt{\frac{n_{j,l}}{n} \times (1 - \frac{n_{j,l}}{n})}$ où $n_{j,l}$ est le nombre de données prenant la l-ième valeur pour la j-ième variable et donc $\frac{n_{j,l}}{n}$ représente la fréquence de la l-ième modalité pour cette variable.

D'autres méthodes de clustering de données mixtes consistent à utiliser une métrique compatible avec les données mixtes, telle que la distance de Gower (Gower, 1971), puis à utiliser une méthode de clustering qui se base sur une matrice de distances, telle que la classification ascendante hiérarchique. Cependant, pour chaque variable dans la distance de Gower, un poids spécifié par l'utilisateur qui détermine sa contribution relative à la distance est attribué, ce qui présente essentiellement le même dilemme que précédemment.

Plusieurs autres méthodologies pour le clustering de données mixtes ont ce problème d'arbitrage, car elles exigent qu'on fixe la contribution relative des variables numériques par rapport aux variables catégorielles. C'est notamment le cas pour la méthode des K-prototypes de Huang (1998) qui utilise comme métrique la distance euclidienne au carré pour les variables continues et catégorielles, où les variables catégorielles sont recodées en indicatrices unitaires et pondérées suivant une constante à optimiser.

Enfin, on peut mentionner un dernier algorithme de clustering de données mixtes. Il est décrit dans Modha and Spangler (2003) et est similaire à celui des K-prototypes. Il utilise une combinaison linéaire pondérée de la distance euclidienne au carré pour les variables numériques et de la distance cosinus pour les variables catégorielles. La pondération entre les variables numériques et catégorielles est identifiée par une recherche dite *brute force*. Cela requiert donc de multiples exécutions de l'algorithme et la sélection d'un hyperparamètre supplémentaire.

En résumé, il ne semble pas y avoir de solution idéale dans l'absolu. Il faut faire un choix entre fixer des hypothèses sur la structure des données ou ne pas le faire aux dépens de la simplicité et de la rapidité. Sachant que le Group-Sparse-K-means est pourvu de deux paramètres (K, λ) qui comme nous l'avons vu pour le WT-K-means sont compliqués à optimiser, nous faisons le choix d'un recodage à priori des données de façon à pouvoir utiliser ensuite un algorithme de clustering sur variables numériques.

3.3.3 Recodage des données

Dans cette section nous définissons le recodage des données qui transforme les variables catégorielles de façon à pouvoir utiliser des algorithmes de clustering qui utilisent la distance euclidienne directement sur les données transformées.

- Soit p_1 le nombre de variables numériques, p_2 le nombre de variables catégorielles et $p = p_1 + p_2$ le nombre total de variables. On désigne donc par $\mathbf{X}_1$ la matrice $n \times p_1$ des variables numériques centrées et réduites et $\mathbf{X}_2$ la matrice $n \times p_2$ de variables catégorielles où chaque variable $\mathbf{x}^j$ de $\mathbf{X}_2$ possède c_j catégories et on notes c le nombre de toutes les catégories $c = \sum_{j=1}^{p_2} c_j$.

- Soit $\mathbf{G}_j \in \{0, 1\}^{n \times c_j}$ la matrice indicatrice de la j-ième variable catégorielle ayant c_j catégories et soit $\mathbf{D}_j$ la matrice diagonale des fréquences des catégories de cette variable.

- Soit $\mathbf{J} = \mathbf{I}_n - \frac{\mathbf{1}_n \mathbf{1}_n'}{n}$ l'opérateur de centrage où $\mathbf{I}_n$ désigne la matrice d'identité $n \times n$ et $\mathbf{1}_n$ un vecteur de dimension n dont les termes sont égaux à 1.
- Soit $\mathbf{G} = (\mathbf{G}_1|, ..., |\mathbf{G}_{p_2})$ la matrice $n \times c$ des variables indicatrices des c catégories des p_2 variables catégorielles et soit $\mathbf{D} = \text{diag}(\mathbf{D}_1, ..., \mathbf{D}_{p_2})$ la matrice diagonale $n \times c$ des fréquences des c catégories.

Pour normaliser la matrice $\mathbf{X}_2$, nous procédons comme dans Chavent et al. (2007), en remplaçant $\tilde{\mathbf{X}}_2$ par la matrice $n \times c$ obtenue la recodant de telle sorte que

$$\tilde{\mathbf{X}}_2 = \frac{1}{\sqrt{n}} \mathbf{J}\mathbf{G}\mathbf{D}^{-\frac{1}{2}},$$

puis nous fusionnons les deux matrices pour obtenir la matrice de données $[\mathbf{X}_1, \tilde{\mathbf{X}}_2]$.

En d'autres termes, $\tilde{\mathbf{X}}_2$ doit être centrée et normalisée par $\sqrt{\frac{n_{j,l}}{n}}$, où $n_{j,l}$ est le nombre de données d'entrée prenant la l-ième valeur de la j-ième variable. La mise à l'échelle appliquée aux variables indicatrices conduit en fait à utiliser une distance de type χ^2 sur les variables catégorielles, tandis que pour les variables numériques après transformation cela revient à utiliser la distance euclidienne.

La Figure 3.2 présente le diagramme résumant les étapes nécessaire à la création de la matrice $[\mathbf{X}_1, \tilde{\mathbf{X}}_2]$. On peut alors remarquer que cette transformation est équivalente à utiliser la distance euclidienne avec la

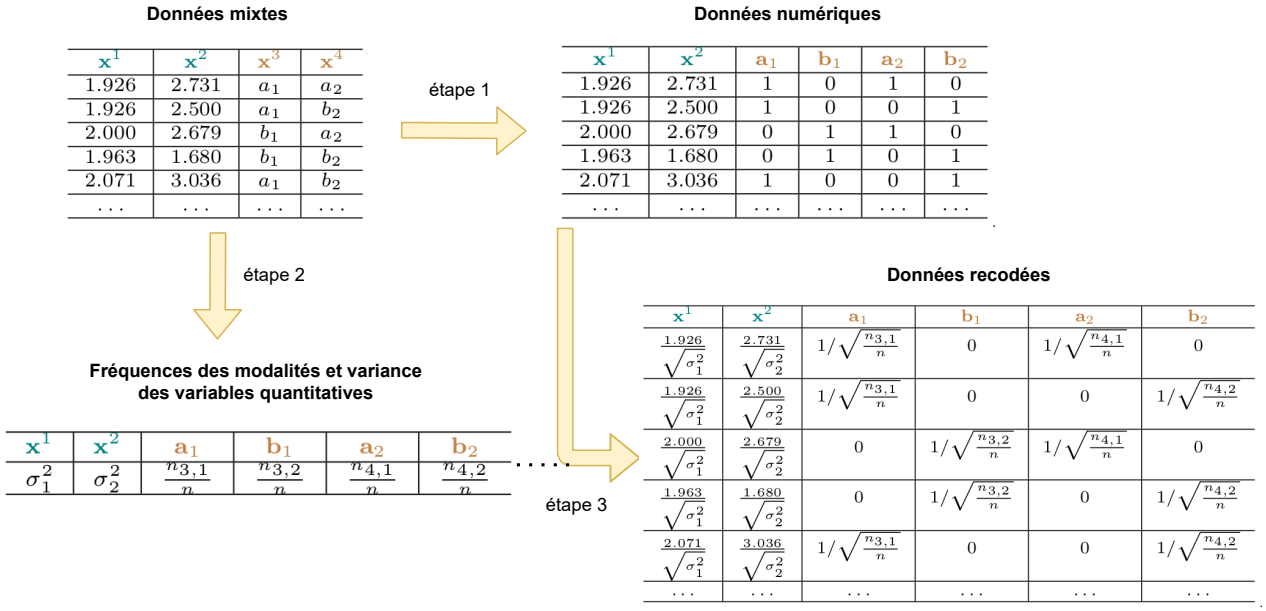


Figure 3.2 : Diagramme illustrant les étapes de recodage des données sur une matrice avec 4 variables dont deux variables numériques et deux variables catégorielles. La première étape consiste à transformer les modalités des variables catégorielles en indicatrices. La deuxième étape consiste à calculer la variance des variables numériques et la fréquence des modalités. Enfin, la troisième étape consiste à normaliser les données numériques par la racine carrée de la variance ou de la fréquence des modalités selon le type de variable.

métrique $\mathbf{M}^{-\frac{1}{2}}$ où $\mathbf{M} = \text{diag}(\sigma_1^2, \ldots, \sigma_{p_1}^2, \frac{n_{1,c_1}}{n}, \ldots, \frac{n_{p,c_p}}{n})$ sur le tableau de données $[\mathbf{X}_1, \mathbf{G}]$.

3.3.4 Group-Sparse K-means appliqué aux données mixtes

Comme on peut le constater dans cette section ou sur la Figure 3.2, le recodage des données mixtes définit naturellement des groupes de variables. Chaque variable catégorielle donne lieu à un groupe de variables indicatrices représentant la présence ou l'absence d'une modalité. Mais alors l'utilisation d'algorithmes classiques de sélection de variables tel que le WT-K-means ne va pas nécessairement sélectionner toutes les variables indicatrices associées à une variable et n'aboutira pas dans ce cas à une véritable sélection de variables. De plus, il n'est pas directement possible de comprendre l'importance d'une variable catégorielle si seulement une partie de ses modalités est sélectionnée. C'est pourquoi nous utilisons dans la suite l'algorithme de clustering Group-Sparse-K-means qui est tout à fait adapté à ce type d'application.

Les p_1 variables numériques sont seules dans leur groupe et forment p_1 groupes et les c variables réparties

en p_2 groupes définis par les variables catégorielles et ainsi :

$$\mathbf{X} = \left[\mathbf{x}^1, ..., \mathbf{x}^{p_1}, \underbrace{\tilde{\mathbf{x}}^1, ..., \tilde{\mathbf{x}}^{c_1}}_{\text{groupe 1}}, \underbrace{\tilde{\mathbf{x}}^{c_1+1}, ..., \tilde{\mathbf{x}}^{c_2}}_{\text{groupe 2}}, ..., \underbrace{\tilde{\mathbf{x}}^{c_{p_2-1}+1}, ..., \tilde{\mathbf{x}}^{c_{p_2}}}_{\text{groupe } p_2}\right] \in \mathbb{R}^{n\times(p_1+c)}.$$

Le vecteur de variance inter-classes $\mathbf{b}$ et le vecteur de poids $\mathbf{w}$ peuvent alors être décomposés en $\mathbf{b}^\intercal = (b_1, ..., b_{p_1}, \mathbf{b}_{p_1+1}, ..., \mathbf{b}_p)$ et $\mathbf{w}^\intercal = (w_1, ..., w_{p_1}, \mathbf{w}_{p_1+1}, ..., \mathbf{w}_p)$. Ensuite il faut résoudre le problème (3.4) à l'aide de l'algorithme itératif et des solutions qui ont été vues à la Section 3.2. On montre notamment que la pénalité groupe s'écrit désormais :

$$\lambda \sum_{l=1}^{p_1+c} \sqrt{p_l}\|\mathbf{w}_l\|_2 = \lambda\Big(\sum_{l=1}^{p_1} \|w_l\|_2 + \sum_{l=1}^{p_2} \sqrt{c_l}\|\mathbf{w}_l\|_2\Big).$$

3.4 Illustration du package `vimpclust` sur des données réelles

Dans cette section, nous présentons le package **vimpclust** qui implémente la méthode Group-Sparse-K-means définie dans ce chapitre (Chavent et al., 2020). Ce package dispose de deux fonctions principales, **sparsewkm** qui permet de faire du clustering sparse de données numériques et/ou catégorielles et **groupsparsewkm** qui permet de faire du clustering sparse de groupes de variables numériques. Ces méthodes généralisent l'algorithme de Witten and Tibshirani (2010). Notamment si les variables concernées par le clustering sont uniquement numériques sans structure de groupes a priori, l'algorithme de clustering se réduit à celui de Witten and Tibshirani (2010). Si certaines ou toutes les variables sont catégorielles, **sparsewkm** sparsewkm transforme les données en utilisant une étape de recodage décrite dans la Section 3.3.3 ci-avant et c'est cette fonction que nous illustrons ici.

Commençons par décrire la fonction et ses arguments. Plusieurs arguments peuvent être donnés en entrée de la fonction **sparsewkm**, mais seuls les deux premiers sont requis,

- **X** : est la matrice des données. Les données doivent être sous le format **data.frame** et les variables catégorielles en format **factor**.
- **centers** : est le nombre de clusters K à construire.

Les autres arguments sont liés au choix du paramètre de régularisation, ou au nombre d'itérations et d'initialisations aléatoires de l'algorithme. Les valeurs par défaut sont fixées pour ces paramètres et plus d'informations sont disponibles en utilisant l'aide **help(sparsewkm)**.

Pour illustrer le fonctionnement du package, nous utilisons les données sur les maladies cardiaques HDdata* qui décrivent 270 patients par six variables numériques et huit variables catégorielles.

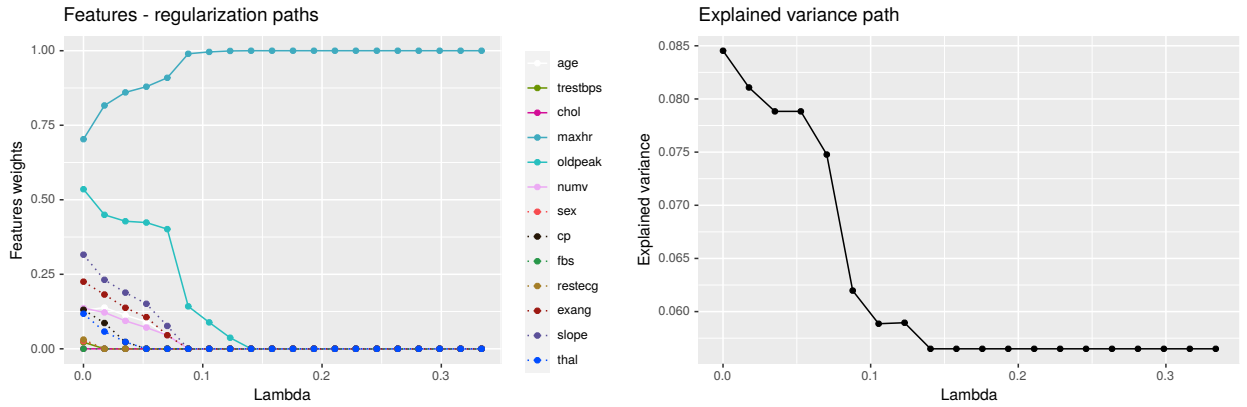
La fonction **sparsewkm** est appliquée aux données HDdata sur toutes les variables sauf la dernière, **hd**, qui code la présence ou l'absence d'une maladie cardiaque. On fixe le nombre de clusters à deux. Ci-dessous est présenté le code permettant d'exécuter la fonction et d'afficher deux graphiques, le chemin de régularisation qui montre le poids des variables en fonction des valeurs du paramètre λ et le chemin de variance expliquée des données non pondérées en fonction des valeurs du paramètre λ.

```
res <- sparsewkm(X = HDdata, centers = 2)
plot(res, what="weights.features")
plot(res, what="expl.var")
```

Les graphiques sont présentés sur la Figure 3.3 : le graphique (a) indique le chemin de régularisation c'est-à-dire le poids des variables en fonction des valeurs de λ. Lorsqu'une variable j est numérique c'est bien w_j que l'on représente et lorsqu'une variable l est catégorielle, c'est $\|\mathbf{w}_l\|_2$ qui est représenté. Le graphique (b) représente le chemin de variance expliquée en fonction des valeurs de λ. On peut voir que, lorsque λ augmente, les poids de certaines variables atteignent la valeur zéro. Par défaut, les chemins associés aux variables numériques sont affichés avec des lignes continues, tandis que ceux associés aux variables catégorielles sont affichés avec des lignes pointillées.

D'après les résultats, les variables numériques **maxhr** et **oldpeak**, et les variables catégorielles **slope** et **exang** apparaissent comme les plus discriminantes pour de petites valeurs de λ. Lorsque λ augmente, seul **maxhr** est sélectionné par l'algorithme. L'algorithme de détection de rupture choisit la huitième valeur de λ en partant de la gauche. D'autres graphiques sont disponibles, comme le chemin de régularisation des modalités qui fournit une image plus détaillée de la façon dont chaque modalité d'une variable catégorielle contribue au clustering et le nombre de variables sélectionnées en fonction des valeurs de λ.

*https ://archive.ics.uci.edu/ml/datasets/statlog+(heart)



(a) Chemin de régularisation

(b) Chemin de variance expliquée

Figure 3.3 : Le graphique (a) représente le chemin de régularisation c'est-à-dire le poids des variables en fonction des valeurs de λ, le graphique (b) représente le chemin de variance expliquée en fonction des valeurs de λ.

LA NORMALISATION DES DONNÉES La variance des variables affecte énormément les algorithmes de clustering basés sur la distance euclidienne, et davantage les algorithmes clustering sparse. Spécifiquement, pour la méthodologie destinée aux données mixtes, étant donné le recodage que l'on effectue en entrée de l'algorithme, il n'y a plus besoin de normaliser les données par la suite. Mais lorsque l'algorithme est utilisé sur des groupes de variables numériques (non recodées), a priori les données ne sont pas transformées en amont et il faut donc normaliser les variables à moyenne nulle et variance unitaire. Tous ces détails sont pris en charge automatiquement dans le package `vimpclust` qui implémente nos solutions.

INITIALISATION DE L'ALGORITHME Dans la suite de ce qui a été discuté dans le Chapitre 2, nous avons implémenté dans notre package `R` `vimpclust`, la possibilité de renseigner le nombre de fois où les K-means standards doivent être relancés (comme expliqué dans l'introduction 2.2.2) et nous avons développé dans une version bêta une initialisation au moyen de l'algorithme K-means++. Ces fonctionnalités ne sont pas disponibles dans le package de Daniela Witten `sparcl`. Cette proposition est utile car on constate que l'initialisation a un très grand impact sur les résultats de l'algorithme, et de manière générale sur les résultats des algorithmes sparses. La version comprenant l'initialisation K-means++ n'a pas été utilisé dans ce manuscrit et donc les différences observées entre les fonctions des deux packages codant le WT-K-means sont essentiellement dues au choix du paramètre λ.

3.5 Simulations : comparaison des méthodes de clustering sparse sur des données mixtes

Dans cette section nous comparons deux méthodes de clustering sparse qui traitent des données mixtes : la méthode du package `VarSelLCM` (Marbac and Sedki, 2017) décrite dans la Section 2.4.4 et le package `vimpclust` (Chavent et al., 2020) qui implémente le Group-Sparse-K-means avec détection de rupture pour le choix du paramètre λ. On reprend le même schéma de simulation que celui qui a été décrit par l'Équation 2.17, et on discrétise en variables binaires (deux catégories) la moitié des variables importantes et la moitié des variables de bruit. L'opération est simple : puisque par définition toutes les variables sont centrées et de médiane zéro, on discrétise les variables de la façon suivantes :

$$x_i'^j = \begin{cases} 1 & \text{si} \quad x_i^j > 0, \\ 0 & \text{sinon.} \end{cases}$$

Nous testons deux scénarios, avec $K = 2, p_K = 2, d = 20, m = 1.5$ où une des deux variables importantes et dix variables de bruit sont discrétisées et $K = 2, p_K = 10; d = 100$ où cinq variables importantes et cinquante variables de bruit sont discrétisées. La valeur m est indiquée dans le tableau.

La Table 3.1 résume les résultats obtenus suivant les différents scénarios. Les méthodes sont équivalentes en termes de performances, mais pas en termes de sélection de variables. En effet, `sparsewkm` a tendance à ne pas sélectionner les variables catégorielles. On peut expliquer ce fait par deux arguments : le premier c'est qu'en général `sparsewkm` attribue aux variables catégorielles des poids plus faibles que ceux attribués aux variables

Table 3.1 : Le tableau représente les moyennes et écart type par méthode pour le scénario $K = 2, p_K = 10, d = 100, m = 1.7$ sur 20 simulations de l'ARI, le ratio de variables importantes (Ratio V.Imp) et de bruit (Ratio V.Bruit) sélectionnées et le temps de calcul (Temps).

	scénario			ARI		Ratio V.Imp		Ratio V.Bruit		Temps	
noms	p_k	d	m	moyenne	sd	moyenne	sd	moyenne	sd	moyenne	sd
VarSelLCM	2	20	1.5	0.92	0.06	1.00	0.00	0.02	0.03	1.06	0.13
`sparsewkm`	2	20	1.5	0.84	0.09	0.50	0.00	0.03	0.02	0.42	0.06
VarSelLCM	10	100	0.6	0.74	0.12	0.97	0.07	0.03	0.02	4.75	0.31
`sparsewkm`	10	100	0.6	0.42	0.10	0.30	0.15	0.01	0.02	2.17	0.48
VarSelLCM	10	100	0.7	0.77	0.28	0.99	0.03	0.02	0.02	3.92	0.35
`sparsewkm`	10	100	0.7	0.57	0.15	0.35	0.14	0.01	0.02	1.95	0.50
VarSelLCM	10	100	0.8	0.93	0.04	1.00	0.00	0.02	0.02	3.54	0.18
`sparsewkm`	10	100	0.8	0.62	0.25	0.28	0.18	0.00	0.01	1.93	0.32
VarSelLCM	10	100	0.9	0.96	0.05	1.00	0.00	0.03	0.02	3.26	0.18
`sparsewkm`	10	100	0.9	0.75	0.19	0.30	0.16	0.00	0.00	1.65	0.34
VarSelLCM	10	100	1	0.98	0.04	1.00	0.00	0.03	0.01	3.16	0.15
`sparsewkm`	10	100	1	0.88	0.17	0.41	0.15	0.01	0.02	1.51	0.29

numériques. Le second est une conséquence du premier, car si les variables catégorielles ont un poids plus faible, elles ont moins d'impact sur la variance inter-classes et donc sur le chemin de variance inter-classes. Ainsi, le choix du paramètre λ se fait en grande partie à l'aide des variables numériques, ce que l'on peut vérifier dans les détails des simulations.

3.6 Conclusion

Dans ce chapitre, nous proposons une méthode de clustering sparse intégrant une structure en groupes de variables connue a priori. Cette formulation, utilisée conjointement avec le recodage des données catégorielles, permet de faire du clustering sparse de données mixtes.

Nous avons choisi d'étendre le WT-K-means pour plusieurs raisons. Premièrement, le problème d'optimisation est clair : il s'agit d'optimiser un critère d'inertie pour lequel les poids des variables s'interprètent comme leur contribution à l'inertie. Deuxièmement, l'algorithme des K-means est connu pour être peu coûteux en temps de calcul et les méthodes sparses basées sur les GMM qui ont le même avantage font souvent des hypothèses similaires à celles qui sont sous-jacentes à l'algorithme des K-means. Troisièmement, le WT-K-means est souvent choisi comme algorithme de référence, utilisé pour comparer les résultats d'autres méthodes et aucune autre n'a encore montré clairement sa supériorité. Quatrièmement, la pénalité *lasso* a une extension directe au cas des variables structurées en groupes, le *group lasso*, qui permet une écriture simple du problème. Enfin, dans le contexte supervisé, on sait que les méthodes pénalisées sont plus stables et ne souffrent pas d'autant de variabilité que les méthodes de type stepwise par exemple (Friedman et al., 2001). Nous avons voulu hériter du succès que ces méthodes ont obtenu en apprentissage supervisé, en les adaptant au cadre non supervisé du clustering.

Nous avons abordé le fait que le clustering de données mixtes est encore une question ouverte et nous avons fait le choix d'un recodage a priori des données. Toutefois, en toute honnêteté, le recodage de données mixtes présente au moins une limitation et nous pouvons le voir avec cet exemple : soit un ensemble de données décrites par une variable numérique qui a une structure en clusters, disons un mélange de deux gaussiennes, et une variable binaire. Si la variable binaire est très dépendante de la variable numérique, alors elle contient elle aussi l'information sur les clusters et a une structure en clusters. En revanche, si la variable binaire est indépendante de la variable numérique, alors rien ne peut être dit sur la structure de cette variable (elle pourrait être liée à une variable non observée, qui pourrait ou non, avoir une structure en clusters). Le raisonnement inverse peut être fait si la variable numérique n'a pas de structure en clusters. Le recodage que nous proposons est en fait une hypothèse que nous posons sur les données. On voit d'ailleurs dans les simulations, que les variables numériques et catégorielles ne semblent pas avoir la même influence sur le clustering.

3.7 Annexe

3.7.1 Solution du Group-Sparse-K-means

Nous allons dans cette section donner la solution explicite du problème Group-Sparse-K-means défini dans ce chapitre. Dans un souci de généralité et pour rendre cette section plus compacte, nous allons écrire le problème sous une forme plus générale que l'on appellera Sparse-Group-Sparse-K-means qui mélange deux

pénalités de normes ℓ_1 et ℓ_2 ce qui revient à utiliser la même forme de pénalisation que le sparse group lasso en régression Simon et al. (2013). Cette pénalisation permet de pénaliser par groupe mais aussi à l'intérieur des groupes. Malheureusement elle introduit un paramètre supplémentaire qui semble très compliqué à optimiser. Néanmoins, ce cadre offre la possibilité d'obtenir les solutions des problèmes WT-K-means, Group-Sparse-K-means et Sparse-Group-Sparse-K-means avec une seule formulation.

Supposons que les p variables soient divisées en L groupes connus à l'avance, tels que $\mathbf{X} = \Big[\mathbf{X}^1|\dots|\mathbf{X}^L\Big]$, avec $\mathbf{X}^\ell \in \mathbf{R}^{n\times p_\ell}$, p_ℓ étant la taille du groupe ℓ, et $p_1 + ... + p_L = p$. Le vecteur de variance inter-classes $\mathbf{b}$ et le vecteur de poids $\mathbf{w}$ peuvent également être décomposés en $\mathbf{b}^\intercal = (\mathbf{b}_1, \dots, \mathbf{b}_L)$ et $\mathbf{w}^\intercal = (\mathbf{w}_1, \dots, \mathbf{w}_L)$.

On définit l'algorithme Sparse-Group-Sparse-K-means comme solution du problème suivant :

$$\underset{C_1,\dots,C_K,\mathbf{w}}{\text{maximiser}}\ \mathbf{w}^\intercal\mathbf{b} - \lambda\Big[(1-\alpha)\sum_{\ell=1}^{L}\upsilon_\ell\|\mathbf{w}_\ell\|_2 + \alpha\|\mathbf{w}\|_2\Big]\ \text{s.c.}\|\mathbf{w}\|_2^2 \leq 1\ ,\ \mathbf{w} \geq \mathbf{0},$$

où ici nous supposerons $\upsilon_\ell = 1$ par souci de clarté. Rappelons que la procédure itérative est en deux étapes, où dans une étape les poids sont fixés.

SOLUTION Tout d'abord, nous pouvons réécrire le problème dans sa formulation complète avec contraintes comme un problème de minimisation :

$$\underset{C_1,\dots,C_K,\mathbf{w}}{\text{minimiser}}\ -\mathbf{w}^\intercal\mathbf{b} + \lambda\Big[(1-\alpha)\sum_{\ell=1}^{L}\|\mathbf{w}_\ell\|_2 + \alpha\|\mathbf{w}\|_1\Big]\ \text{s.c.}\ \|\mathbf{w}\|_2^2 \leqslant 1, \mathbf{w} > \mathbf{0}.$$

Soit $\mathcal{L}(\mathbf{w},\lambda,\gamma) = f(\mathbf{w}) + \gamma g(\mathbf{w})$, où $f(\mathbf{w}) = -\mathbf{w}^\intercal\mathbf{b} + \lambda\Big[(1-\alpha)\sum_{\ell=1}^{L}\|\mathbf{w}_\ell\|_2 + \alpha\|\mathbf{w}\|_1\Big]$, $g(\mathbf{w}) = \|\mathbf{w}\|_2^2 - 1$. La solution doit satisfaire aux conditions de Karush-Kuhn-Tucker (KKT) suivantes :

$$\begin{aligned}
&\frac{\partial\mathcal{L}(\mathbf{w},\lambda,\gamma)}{\partial\mathbf{w}} = 0,\\
&\gamma \geqslant 0,\\
&g(\mathbf{w}) \leqslant 0,\\
&\gamma g(\mathbf{w}) = 0,
\end{aligned}$$

où γ est un multiplicateur de Lagrange. Cependant, la norme ℓ_1 n'a pas de dérivée en 0. Ainsi, la première condition KKT s'écrit comme suit :

$$-\mathbf{b}_\ell + 2\gamma\mathbf{w}_\ell + \lambda\Big[(1-\alpha)\Gamma^2(\mathbf{w}_\ell) + \alpha\Gamma^1(\mathbf{w}_\ell)\Big] = 0, \tag{3.6}$$

où $\Gamma^1(\mathbf{w}_\ell)$ est la sous-différentielle du groupe ℓ évaluée en $\mathbf{w}_\ell$et $\Gamma^1(\mathbf{w}_\ell)$ est définie comme :

$$\Gamma^1(\mathbf{w}_\ell) = \big(\Gamma^1(w_{\ell_1}), ..., \Gamma^1(w_{\ell_{\mathbf{p}_\ell}})\big)^\intercal \in \mathbb{R}^{p_\ell} \text{ avec } \Gamma^1(w_{\ell_j}) = \begin{cases} 1 \text{ si } w_{\ell_j} > 0,\\ c \in [-1;1] \text{ si } w_{\ell_j} = 0,\\ -1 \text{ si } w_{\ell_j} < 0, \end{cases}$$

où le troisième cas n'est pas possible ici car la variable j appartient forcément au groupe ℓ, $\mathbf{x}^j \in \mathbf{X}^\ell$, et $\Gamma^2(\mathbf{w}_\ell)$ est la sous-différentielle de la norme ℓ_2 et $\Gamma^2(\mathbf{w}_\ell)$ est définie comme :

$$\Gamma^2(\mathbf{w}_\ell) = \begin{cases} \dfrac{\mathbf{w}_\ell}{\|\mathbf{w}_\ell\|_2} \text{ si } \|\mathbf{w}_\ell\|_2 \neq 0,\\ \{\mathbf{u}, \|\mathbf{u}\|_2 \leqslant 1\} \text{ si } \|\mathbf{w}_\ell\|_2 = 0. \end{cases}$$

Nous pouvons réécrire l'Équation 3.6 comme

$$2\gamma\mathbf{w}_\ell = \mathbf{b}_\ell - \lambda\Big[(1-\alpha)\Gamma^2(\mathbf{w}_\ell) + \alpha\Gamma^1(\mathbf{w}_\ell))\Big]. \tag{3.7}$$

On considère d'abord deux cas qui dépendent de la pénalité λ :

1. si $\lambda = 0$, le problème se résout directement avec $w_j = \dfrac{b_j}{\|\mathbf{b}\|_2}$. En effet,

$$\sum_{j=1}^{p} w_j^2 = 1 \iff \sum_{j=1}^{p} b_j^2 \times (2\gamma)^{-2} = 1 \iff 2\gamma = \sqrt{\sum_{j=1}^{p} b_j^2}.$$

2. si $\lambda > 0$ alors il faut encore simplifier l'Équation 3.7 sous une forme d'opérateur de seuillage. Les sous-différentielles ne sont pas évaluables en 0 et donc nous devons distinguer les deux cas suivants :

   (a) $\|\mathbf{w}_\ell\|_2 = 0$ si $\|(\mathbf{b}_\ell - \lambda\alpha)_+\|_2 \leqslant \lambda(1-\alpha)$ avec un petit travail algébrique et on obtient comme solution de l'Équation 3.7 : $\Gamma^2(\mathbf{w}_\ell) = \frac{(\mathbf{b}_\ell - \lambda\alpha)_+}{\lambda(1-\alpha)}$ et $\Gamma^1(\mathbf{w}_\ell) = \frac{\mathbf{b}_\ell - (\mathbf{b}_\ell - \lambda\alpha)_+}{\lambda\alpha}$.

   (b) $\|\mathbf{w}_\ell\|_2 \neq 0$ si $\|\mathbf{w}_\ell\|_2 = 0$ si $\|(\mathbf{b}_\ell - \lambda\alpha)_+\|_2 \geqslant \lambda(1-\alpha)$ et l'Équation 3.7 s'écrit

$$\mathbf{w}_\ell\Big(2\gamma + \frac{\lambda(1-\alpha)}{\|\mathbf{w}_\ell\|_2}\Big) = \mathbf{b}_\ell - \lambda\alpha\Gamma^1(\mathbf{w}_\ell). \tag{3.8}$$

   De cette équation nous pouvons décrire deux nouveaux cas si nous observons en particulier la variable la variable j appartenant au groupe ℓ, $\mathbf{x}^j \in \mathbf{X}^\ell$:

   i. $w_j = 0$ si $b_j \leqslant \lambda\alpha$ pour tout j dans le groupe ℓ et $\Gamma^1(w_{\ell_j}) = \frac{b_j}{\lambda\alpha}$.

   ii. $w_j > 0$ si $b_j > \lambda\alpha$ pour tout j dans le groupe ℓ et $w_j\Big(2\gamma + \frac{\lambda(1-\alpha)}{\|\mathbf{w}_\ell\|_2}\Big) = b_j - \lambda\alpha$.

   En réunissant les deux cas précédents on obtient :

$$w_j\Big(2\gamma + \frac{\lambda(1-\alpha)}{\|\mathbf{w}_\ell\|_2}\Big) = (b_j - \lambda\alpha)_+ \tag{3.9}$$

   Désormais nous avons résolu le problème de la non différentiabilité et nous pouvons décrire le problème dans le cas $\lambda > 0, \|\mathbf{w}_\ell\|_2 \neq 0, w_j > 0$. Nous pouvons ainsi repartir de l'Équation 3.9 et nous distinguons encore deux cas :

   A. si $\gamma > 0$ alors on peut réécrire l'Équation 3.9 sous forme vectorielle :

$$\mathbf{w}_\ell\Big(2\gamma + \frac{\lambda(1-\alpha)}{\|\mathbf{w}_\ell\|_2}\Big) = (\mathbf{b}_\ell - \lambda\alpha)_+ \iff 2\gamma\|\mathbf{w}_\ell\|_2 = \|(\mathbf{b}_j - \lambda\alpha)_+\|_2 - \lambda(1-\alpha) \tag{3.10}$$

   Ainsi nous pouvons injecter l'Équation 3.10 dans 3.9 et nous obtenons :

$$\mathbf{w}_\ell = \frac{1}{2\gamma}\frac{(\mathbf{b}_j - \lambda\alpha)_+}{\|(\mathbf{b}_j - \lambda\alpha)_+\|_2}\Big(\|(\mathbf{b}_j - \lambda\alpha)_+\|_2 - \lambda(1-\alpha)\Big) \tag{3.11}$$

$$= \frac{1}{2\gamma}\tilde{S}\big((\mathbf{b}_j - \lambda\alpha)_+, \lambda(1-\alpha)\big) \tag{3.12}$$

   or $\gamma > 0$ implique $\|\mathbf{w}\|_2 = 1$ donc

$$\mathbf{w}_\ell = \frac{\tilde{S}\big((\mathbf{b}_j - \lambda\alpha)_+, \lambda(1-\alpha)\big)}{\|\tilde{S}\big((\mathbf{b}_j - \lambda\alpha)_+, \lambda(1-\alpha)\big)\|_2} \tag{3.13}$$

   B. si $\gamma = 0$ on obtient directement par l'Équation 3.10 que $\mathbf{w} = 0$.

Ainsi si on résume, on a

$$\mathbf{w}^\star = \begin{cases} \mathbf{0} \text{ si } \|\tilde{S}\big((\mathbf{b}_j - \lambda\alpha)_+, \lambda(1-\alpha)\big)\|_2 = 0, \\ \dfrac{\tilde{S}\big((\mathbf{b}_j - \lambda\alpha)_+, \lambda(1-\alpha)\big)}{\|\tilde{S}\big((\mathbf{b}_j - \lambda\alpha)_+, \lambda(1-\alpha)\big)\|_2} \text{ sinon,} \end{cases}$$

avec $\frac{\tilde{S}\left((\mathbf{b}_j - \lambda\alpha)_+, \lambda(1-\alpha)\right)}{\|\tilde{S}\left((\mathbf{b}_j - \lambda\alpha)_+, \lambda(1-\alpha)\right)\|_2} = \frac{\mathbf{b}}{\|\mathbf{b}\|_2}$ si $\lambda = 0$ et γ toujours choisi supérieur à 0 pour une solution non-triviale.

Désormais, nous pouvons donner la solution du problème WT-K-means, c'est-à-dire lorsque $\alpha = 1$, qui peut s'écrire :

$$\mathbf{w}^\star = \begin{cases} \mathbf{0} \text{ si } \lambda \geqslant b_j \ \forall j = 1, \ldots, p, \\ \dfrac{(\mathbf{b}_j - \lambda)_+}{\|(\mathbf{b}_j - \lambda)_+\|_2} = \dfrac{S(\mathbf{b}_j, \lambda)}{\|S(\mathbf{b}_j, \lambda)\|_2} \text{ sinon,} \end{cases}$$

S étant l'opérateur de seuillage doux et nous pouvons donner la solution du problème Group-Sparse-K-means, lorsque $\alpha = 0$, qui peut s'écrire :

$$\mathbf{w}^\star = \begin{cases} \mathbf{0} \text{ si } \|\tilde{S}\big(\mathbf{b}_j, \lambda\big)\|_2 = 0, \\ \dfrac{\tilde{S}\big(\mathbf{b}_j, \lambda\big)}{\|\tilde{S}\big(\mathbf{b}_j, \lambda\big)\|_2} \text{ sinon.} \end{cases}$$

4

Clustering et données corrélées

4.1 Introduction . . . 56
4.1.1 Un exemple introductif 57
4.1.2 Les contributions de ce chapitre 57
4.2 État de l'art . . . 58
4.3 ACP et K-means . . . 59
4.4 La solution proposée : normaliser les variables en fonction des corrélations . . . 60
4.4.1 Présentation de la solution : pondérer les variables par leurs corrélations 60
4.4.2 Lien avec l'ACP 62
4.5 Simulations . . . 62
4.5.1 Simulations pour des algorithmes de clustering non sparse 63
4.5.2 Simulations pour les algorithmes de clustering sparse 64
4.5.2.a Résultats dans le scénario 1 : $p_K = 10; d = 50, q = 0$ 65
4.5.2.b Résultats dans le scénario 2 : $p_K = 10; d = 0; q = 50$ 65
4.5.2.c Résultats pour le scénario 3 : $p_K = 10; d = 50; q = 50$ 66
4.5.3 Détails des résultats 66
4.6 Conclusion . . . 67

4.1 Introduction

La présence de corrélations entre les variables dans des données multivariées n'est pas un cas rare. Les algorithmes de la famille des K-means, ou plus généralement ceux basés sur la distance euclidienne ne modélisent pas la variance des données comme peuvent le faire des algorithmes de modèles de mélange gaussien (GMM). Il est alors légitime de se poser les questions suivantes :

- les corrélations entre les variables posent-elles des problèmes pour les algorithmes ne les prenant pas en compte ?
- si les corrélations posent effectivement des problèmes, quels sont-ils ?

Donnons un élément de réponse avec un exemple simple : imaginez que l'ensemble de données soit décrit par trois variables, dont deux sont identiques et que l'on souhaite construire des clusters avec un algorithme qui utilise la distance euclidienne, comme les K-means. L'information apportée par les variables identiques compte double dans le calcul de la distance euclidienne entre deux observations. Par conséquent, il est probable que le clustering dépende exclusivement de ces deux variables identiques, quelle que soit la structure de l'ensemble de données.

4.1.1 Un exemple introductif

Nous simulons un exemple qui illustre le problème des données corrélées. Soit $\mathbf{X}$ une matrice de données avec $n = 120$ observations et $p = 3$ variables notées $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$, simulées selon le mélange de deux gaussiennes suivant :

$$\sum_{k=1}^{2} 0.5 \times \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_0) \text{ avec } \boldsymbol{\mu}_1 = (7, 0, 0), \boldsymbol{\mu}_2 = (0, 0, 0) \text{ et } \boldsymbol{\Sigma}_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.7 \\ 0 & 0.7 & 1 \end{pmatrix}$$

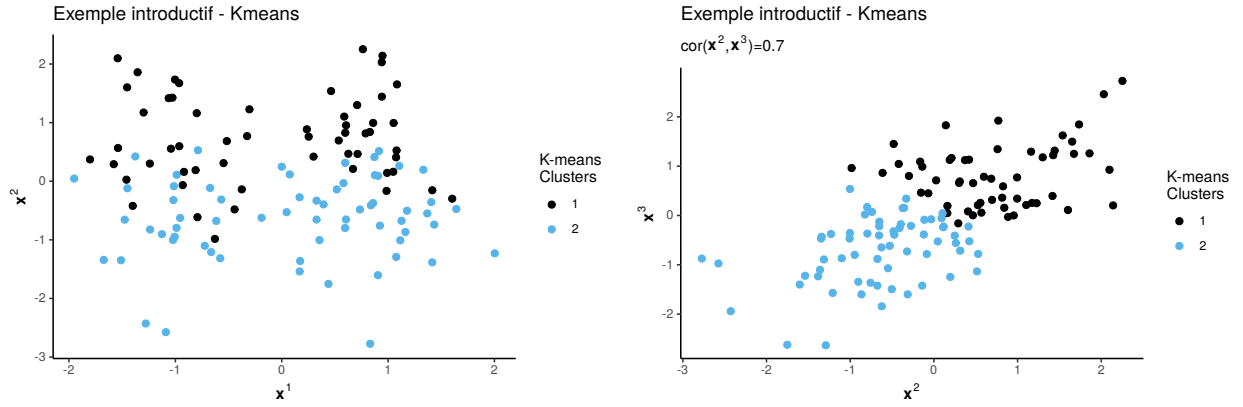
où $\boldsymbol{\mu}_k$ et $\boldsymbol{\Sigma}_0$ sont respectivement les moyennes et la matrice de covariance des deux composantes. On remarque sur la Figure 4.1 que les centres des classes simulées sont séparés uniquement sur $\mathbf{x}^1$ et que les variables $\mathbf{x}^2, \mathbf{x}^3$ sont moyennement corrélées. On peut vérifier que la matrice de covariance du mélange est

$$\boldsymbol{\Sigma}_{melange} = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 1 & 0.7 \\ 0 & 0.7 & 1 \end{pmatrix}.$$

Nous décidons de normaliser à moyenne nulle et variance unitaire car c'est généralement une opération qui est faite sur les données réelles et cela va permettre d'illustrer en détail notre propos. Ainsi nous obtenons la matrice de covariance des données (égale à la matrice de corrélation)

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{X}} \approx \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.7 \\ 0 & 0.7 & 1 \end{pmatrix}.$$

La Figure 4.1 montre la partition en deux classes obtenue par l'algorithme des K-means sur les données centrées réduites et on remarque que l'algorithme des 2-means ne parvient pas à retrouver les clusters sous-jacents. Il faut noter que si les données ne sont pas centrées réduites, l'algorithme des K-means retrouve les clusters car la variance de $\mathbf{x}^1$ est grande par rapport à celles des autres variables.



(a) Clustering représenté sur $\mathbf{x}^1, \mathbf{x}^2$

(b) Clustering représenté sur $\mathbf{x}^2, \mathbf{x}^3$

Figure 4.1 : Données simulées où $\mathbf{x}^1$ sépare bien les deux clusters et où $\mathbf{x}^2, \mathbf{x}^3$ sont légèrement corrélées ($\text{cor}(\mathbf{x}^1, \mathbf{x}^2) = 0.7$). (a) représente les clusters trouvés par les 2-means représentées sur le plan $\mathbf{x}^1, \mathbf{x}^2$; (b) représente les clusters trouvés par les 2-means représentées sur le plan $\mathbf{x}^2, \mathbf{x}^3$. Les clusters obtenus correspondent au meilleur résultat obtenu par un algorithme 2-means (sur un millier d'initialisations aléatoires). L'algorithme des K-means sépare les données suivant la direction des variables corrélées non importantes $\mathbf{x}^2$ et $\mathbf{x}^3$ et les clusters sous-jacents ne sont donc pas retrouvés.

Si on applique les K-means uniquement avec les variables $\mathbf{x}^1, \mathbf{x}^2$, on obtient les résultats attendus au départ. On voit donc que les résultats du clustering changent complètement lorsqu'on ajoute une variable corrélée non importante, ce qui est inattendu et cela illustre l'influence possible de corrélations entre les variables dans le résultat des K-means. Malgré la popularité de l'algorithme des K-means, on sait peu de choses sur son comportement lorsque les données sont corrélées.

4.1.2 Les contributions de ce chapitre

La gestion des corrélations en clustering est nécessaire, comme l'est celle de la variance des variables. En effet, nous avons vu que la présence de variables non importantes corrélées peut fausser le résultat du clustering et à notre connaissance, il ne semble pas y avoir eu de travaux spécifiques sur l'influence des variables de bruit

corrélées en clustering. Il existe néanmoins des solutions pour traiter les variables corrélées dans les données. D'une part, certains algorithmes de clustering prennent en compte la variance et la covariance des données, tels que les modèles de mélange gaussiens (GMM). D'autre part, il est possible de prétraiter les données avec des méthodes dites de *blanchiment* qui décorrèlent les données avant le clustering.

Nous présenterons ensuite dans la Section 4.3 le lien qui peut exister entre la gestion des corrélations en clustering et l'Analyse en Composantes Principales (ACP).

Comme nous souhaitons traiter le problème issu de l'existence de corrélations des variables dans un cadre général, nous proposons dans la Section 4.4 une méthode de prétraitement des données facilement implémentable, peu coûteuse en temps de calcul, adaptée au cas du clustering et du clustering sparse, utilisable en grande dimension.

4.2 État de l'art

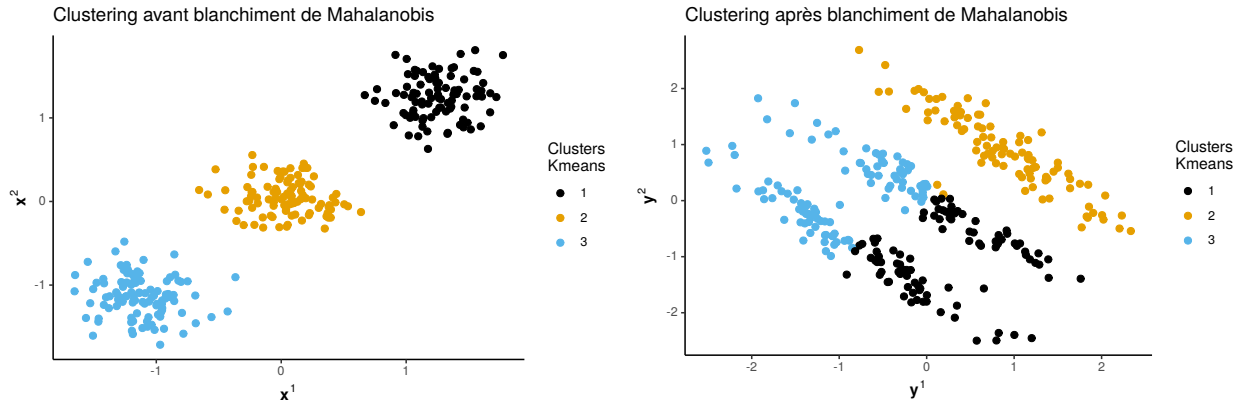
Les méthodes permettant de résoudre les problèmes liés à l'existence des corrélations entre les variables sont essentiellement des méthodes de type prétraitement, effectuées indépendamment de l'algorithme de clustering, et nous faisons une présentation non exhaustive de ces méthodes.

DÉFINITION ET EXEMPLE DE BLANCHIMENT Le blanchiment, *whitening* ou *sphering* en anglais, est une transformation linéaire qui convertit une matrice de données $\mathbf{X} \in \mathbb{R}^{n \times p}$ dont la matrice de covariance estimée est définie positive $\hat{\mathbf{\Sigma}}_{\mathbf{X}} = \frac{\mathbf{X}^\intercal \mathbf{X}}{n}$ en une nouvelle matrice de données

$$\mathbf{Y} = \mathbf{X}\mathbf{W} \text{ où } \mathbf{W}^\intercal \mathbf{W} = \hat{\mathbf{\Sigma}}_{\mathbf{X}}^{-1}, \tag{4.1}$$

de sorte que la matrice de covariance de $\mathbf{Y}$ soit $\hat{\mathbf{\Sigma}}_{\mathbf{Y}} = \mathbf{I}_p$ (la matrice identité). Après transformation, les nouvelles variables sont orthogonales deux à deux et normées. Le blanchiment est alors une généralisation de la normalisation qui prend en compte la matrice de variance-covariance. La contrainte donnée dans l'Équation (4.1) ne détermine pas de façon unique la matrice de blanchiment $\mathbf{W}$: il existe une infinité de matrices possibles $\mathbf{W}$ et chaque $\mathbf{W}$ conduit à une transformation des données en des données sphériques. Il existe donc différentes formes de blanchiment qui se différencient par le choix de la matrice $\mathbf{W}$. Avant de présenter les inconvénients des méthodes de blanchiment de manière générale, nous allons illustrer les problèmes rencontrés avec deux méthodes de blanchiment très connues.

BLANCHIMENT DE MAHALANOBIS Le blanchiment de Mahalanobis est défini par $\mathbf{W} = \hat{\mathbf{\Sigma}}_{\mathbf{X}}^{-1/2}$. Ce type de blanchiment transforme les données en données sphériques et de ce fait les distances entre les observations sont modifiées. Dans certains cas, la structure en clusters des données peut être détériorée. Reprenons le schéma de simulation décrit dans la Section 2.6 qui est le schéma défini par Witten and Tibshirani (2010) avec $K = 3, p_K = 2, \mu = 5$ et sans variable de bruit $d = 0$. On a donc trois clusters qui sont alignés sur deux variables. Les données sont comme d'habitude centrées réduites. La Figure 4.2 représente les résultats obtenus



(a) Clustering avant blanchiment

(b) Clustering après blanchiment

Figure 4.2 : Le graphique (a) représente les clusters trouvés par le K-means sur les données simulées. Le graphique (b) représente les clusters trouvés par le K-means sur les données transformées par le blanchiment de Mahalanobis. La transformation a changé la structure en clusters.

avec l'algorithme des K-means sur les données avant et après un blanchiment de Mahalanobis. Nous pouvons constater que le blanchiment détruit la structure des données.

BLANCHIMENT PAR L'ACP Une autre méthode de blanchiment est présentée ici, appelée blanchiment par ACP ou ACP normée, elle est basée sur l'analyse en composantes principales normée. Dans ce cas $\mathbf{W} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{V}^\intercal$, où $\mathbf{\Lambda} = \text{diag}(\lambda_1, ..., \lambda_p) \in \mathbb{R}^{p\times p}$ est la matrice diagonale des valeurs propres de la matrice $\frac{\mathbf{X}^\intercal\mathbf{X}}{n}$ ordonnées par valeurs décroissantes et $\mathbf{V} \in \mathbb{R}^{p\times p}$ est la matrice orthogonale dont les colonnes sont les vecteurs propres correspondants de $\frac{\mathbf{X}^\intercal\mathbf{X}}{n}$ ce qui implique que la matrice de covariance $\hat{\mathbf{\Sigma}}_\mathbf{X} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\intercal$. Par conséquent, cette transformation fait d'abord pivoter les variables en utilisant les vecteurs propres de la matrice de covariance, comme cela est fait dans l'ACP standard. On obtient ainsi des composantes principales de variance $(\lambda_1, ..., \lambda_p)$. Pour obtenir les données blanchies, il reste à diviser chaque composante principale par la racine carrée de la valeur propre associée, d'où le choix de la matrice W.

Le problème du blanchiment par ACP est un peu différent de celui présenté précédemment. Considérons un ensemble de n données dans $\mathbb{R}^p$ où les p variables sont toutes corrélées avec une corrélation à 0.95, et où les p variables sont distribuées selon un mélange de gaussiennes et ont donc une structure en clusters. Alors, toute la variance des données est représentée sur la première composante principale. Dans ce cas, les autres directions ne sont pas pertinentes pour le clustering mais la normalisation des composantes principales par $\mathbf{\Lambda}^{-\frac{1}{2}}$ implique que toutes les directions sont de même variance. Ainsi, le blanchiment par l'ACP a créé artificiellement $p-1$ variables de bruit. Autrement dit, le blanchiment par ACP peut augmenter la variance dans des directions non pertinentes, ici non redondantes, et cela peut fortement affecter les résultats du clustering.

LES INCONVÉNIENTS DU BLANCHIMENT On peut aussi mettre en évidence deux problèmes causés par les méthodes de blanchiment dans le cadre du clustering.

1. L'opération de blanchiment est impossible lorsque $p \gg n$ car les opérations de blanchiment nécessitent l'inversion de la matrice de covariance, matrice qui est singulière dans ce cas.
2. Le blanchiment ne permet pas de faire du clustering sparse car les variables transformées sont des combinaisons linéaires de toutes les variables de départ.

LE CLUSTERING DE VARIABLES CLUSTOFVAR : Une autre solution possible est de regrouper les variables selon leurs corrélations mutuelles. Cette méthodologie, qui permet notamment le clustering de variables numériques et/ou catégorielles, a été définie par Chavent et al. (2011) sous le nom de ClustOfVar. Le clustering de variables est une alternative aux méthodes de type ACP puisqu'il permet d'organiser les variables en groupes homogènes afin d'obtenir une structure informative des variables. D'un point de vue général, le clustering de variables regroupe des variables qui sont fortement corrélées les unes aux autres et qui apportent donc en grande partie la même information. Une fois les variables regroupées en clusters homogènes, le but est alors de résumer chaque groupe par une seule variable appelée variable synthétique. Pour représenter un groupe de variables, l'utilisateur peut choisir une des variables du cluster, mais d'autres choix sont possibles (comme en quantification il s'agit de déterminer une variable centrale) (Chavent et al., 2011). Ainsi, à une partition de variables est associée un ensemble de variables dites synthétiques.

Le clustering de variables a dans ce cadre deux avantages : réduire les corrélations entre les variables synthétiques et réduire le nombre de variables en entrée des algorithmes. En revanche, cette méthode présente des défauts notamment celui d'avoir un paramètre supplémentaire à optimiser, le nombre de clusters de variables. L'algorithme de clustering de variables peut aussi être très coûteux en temps de calcul, notamment lorsque l'on dépasse les quelques milliers de variables. Ces inconvénients rendent difficile l'utilisation de cette méthode en tant que prétraitement des données en entrée d'algorithmes sparses lorsqu'elles sont de très grande dimension.

4.3 ACP et K-means

LA LITTÉRATURE SUR LES LIENS ENTRE ACP ET K-MEANS Des travaux antérieurs ont déjà prouvé que l'algorithme des K-means est lié à l'ACP, notamment l'étude de Ding and He (2004) dont l'un des résultats les plus intéressants sur le sujet est le suivant :

$$\sum_{k=1}^{K} n_k \sum_{j=1}^{p} (\bar{x}_k^j - \bar{x}^j)^2 \leqslant \sum_{k=1}^{K-1} \lambda_k,$$

où

- $\bar{x}^j$, la moyenne de la variable j et $\bar{x}_k^j = \frac{1}{n_k} \sum_{i \in C_k} x_i^j$, le centre du cluster k sur la variable j d'une partition en K classes des observations ;
- n_k, le nombre d'observations dans le cluster k ;
- $\lambda_1 > ... > \lambda_{K-1}$, les $K-1$ premières valeurs propres de l'analyse en composantes principales.

Donc l'inertie inter-classes d'une partition en K classes est bornée par la somme des $K-1$ premières valeurs propres. Ding and He (2004) interprètent les résultats en concluant qu'un *bon clustering* avec les K-means est un clustering dont les K centres appartiennent au sous-espace des $K-1$ premières composantes principales. L'intuition derrière cette assertion est que les premières composantes principales sont celles qui ont les plus grandes variances et donc que la variance inter-classes sera plus grande dans ce sous espace (Meilă, 2006).

LES MÉTHODES COMBINANT ACP ET K-MEANS POUR AMÉLIORER LE CLUSTERING Sur la base de ces résultats, des études ont été réalisées pour améliorer les K-means à l'aide de l'ACP. Certains articles proposent d'initialiser les K-means à l'aide de l'ACP (Su and Dy, 2004). La procédure est la suivante : couper en son centre la première composante principale pour former deux clusters, puis choisir le cluster dont la première composante principale a la plus grande variance (pour chaque cluster on réalise une ACP et on compare les premières valeurs propres de chacune) et enfin de couper à nouveau en son centre. La procédure continue jusqu'à ce qu'une partition en K clusters soit obtenue ce qui permet de calculer les centres initiaux dans l'espace de départ. D'autres auteurs proposent d'exécuter les K-means sur les K composantes principales pour déterminer les clusters qui initialisent les centres dans l'espace de départ (Xu et al., 2015). Enfin certains auteurs proposent de combiner l'algorithme des K-means et l'ACP dans des procédures itératives (Honda et al., 2008, 2009).

LE LIEN ENTRE L'ACP ET L'ALGORITHME DES K-MEANS MET EN LUMIÈRE LE PROBLÈME DES DONNÉES CORRÉLÉES La littérature décrivant les liens entre les K-means et l'ACP nous révèle que ce qui compte le plus c'est la variance du sous-espace dans lequel résident les centres. Alors, on peut se demander quels sont les avantages de l'utilisation de la méthode des K-means par rapport à celle de l'ACP, si la première (K-means) n'est dans une certaine mesure qu'une version discrète de la seconde (ACP) qui de surcroît n'a pour objectif que de comprendre la structure en variance des données. Manifestement, si l'objectif principal est de découvrir la structure en clusters, il est nécessaire de prendre en compte les corrélations entre les variables. Étant donné que l'ACP maximise l'information restituée en représentant les données dans un sous-espace d'inertie maximale, il faut adopter un point de vue différent, arguant que les corrélations peuvent être considérées comme des redondances dans les données et peuvent nuire aux résultats*.

4.4 La solution proposée : normaliser les variables en fonction des corrélations

4.4.1 Présentation de la solution : pondérer les variables par leurs corrélations

DÉFINITION DE NOTRE SOLUTION DE NORMALISATION On propose une nouvelle normalisation des variables pour traiter le cas des variables corrélées. On a vu que les variables très corrélées contribuent beaucoup à l'inertie et peuvent être prédominantes, ce qui peut générer des erreurs si ces variables ne sont pas importantes. L'idée est donc de diminuer la variance des variables corrélées positivement ou négativement afin de diminuer leur contribution à l'inertie totale. Pour cela, on cherche à pondérer les variables en fonction de leur corrélation aux autres variables. Formellement, soit $\mathbf{X} \in \mathbb{R}^{n \times p}$ la matrice de données centrée réduite et $\text{cor}(\mathbf{x}^l, \mathbf{x}^j)$ la corrélation des variables $\mathbf{x}^l$ et $\mathbf{x}^j$. On pose alors :

$$\nu_j^2 = \sum_{l=1}^{p} \text{cor}(\mathbf{x}^l, \mathbf{x}^j)^2,$$

et ensuite on divise chaque variable par la racine de ce coefficient :

$$\forall j = 1, \ldots, p, \ \mathbf{y}_j = \frac{\mathbf{x}_j}{\sqrt{\nu_j^2}}.$$

La nouvelle matrice de données $\mathbf{Y}$ est donnée par

$$\mathbf{Y} = \mathbf{X}\mathbf{N}^{-\frac{1}{2}},$$

*correlation is not information

où

$$\mathbf{N} = \text{diag}(\nu_1^2, \ldots, \nu_p^2),$$

Cette normalisation par l'inverse de la racine carrée de la somme de toutes les corrélations carrés d'une variable avec toutes les autres est désignée par *Inverse Correlation Scaling* en anglais (ICS) et par la suite nous la nommerons *normalisation ICS*.

REMARQUE SUR LA TRANSFORMATION La normalisation ICS présente plusieurs propriétés intéressantes. Premièrement, les corrélations entre les variables ne sont pas affectées, seule la structure en variance change car la variance de chaque variable est divisée par ν_j^2. Une variable très corrélée aux autres, c'est-à-dire une variable pour laquelle la valeur de ν_j^2 est grande, aura une variance plus petite après transformation et aura moins de poids dans le calcul des distances euclidiennes. La distance euclidienne entre deux observations $\mathbf{y}_i$ et $\mathbf{y}_t$ est la distance euclidienne pondérée par les inverses des coefficients ν_j^2 entre les observations $\mathbf{x}_i$ et $\mathbf{x}_t$:

$$d^2(\mathbf{y}_i, \mathbf{y}_t) = \sum_{j=1}^{p} \frac{1}{\nu_j^2}(x_i^j - x_t^j)^2.$$

La normalisation ICS peut donc s'interpréter comme un poids qui serait attribué à chaque variable. On peut aussi noter que ce coefficient est borné comme suit :

$$1 \leq \nu_j^2 \leq p.$$

En effet, si par exemple toutes les variables sont non corrélées alors $\nu_j^2 = \text{cor}(\mathbf{x}^j, \mathbf{x}^j)^2 = 1$, et si toutes les variables sont identiques alors $\nu_j^2 = p$. Lorsque les données sont centrées réduites, cela implique :

$$\frac{1}{p} \leq \text{var}(\mathbf{y}) \leq \text{var}(\mathbf{x}) = 1. \tag{4.2}$$

DIFFÉRENCE FONDAMENTALE AVEC LE BLANCHIMENT En dehors du fait que la normalisation ICS n'entraîne pas la création de nouvelles variables comme combinaison linéaire des variables de départ, il y a une autre subtilité qui marque une différence avec les méthodes de blanchiment. Dans le cas du blanchiment par ACP, on obtient les nouvelles variables en normalisant les composantes principales par la racine carrée de leur valeur propre. Or, quand on pratique une ACP normée, sauf dans le cas sphérique, il existe nécessairement au moins une valeur propre plus petite que 1. Donc la variance augmente pour les composantes (directions) principales dont c'est le cas. Avec la normalisation ICS, la variance n'augmente pas, quelle que soit la direction (voir Équation (4.2)).

Les opérations de blanchiment modifient la structure des données de telle sorte que les distances dans certaines directions sont modifiées, augmentées ou réduites. On peut souhaiter réduire la redondance engendrée par les corrélations sans augmenter la variance dans les directions des dernières composantes principales.

RÉSULTATS SUR L'EXEMPLE INTRODUCTIF La Table 4.1 donne l'ARI entre la *vraie partition* et le clustering des K-means avant normalisation ICS (colonne K-means dans La Table 4.1) et après normalisation ICS (colonne K-means ICS dans La Table 4.1), agrégés sur 100 simulations (moyenne et écart-type), en partant du schéma de l'exemple introductif Section 4.1.1. La table montre que l'algorithme des K-means ne donne pas de bons résultats sur des données corrélées même pour un cas aussi simple.

Table 4.1 : Le tableau représente les moyennes et les écarts-types de l'ARI de chaque méthode pour l'exemple introductif avec $\text{cor}(\mathbf{x}^2, \mathbf{x}^3) = 0.7$ sur 100 simulations.

	K-means	K-means ICS
moyenne	0.01	0.90
écart-type	0.02	0.05

Comme souhaité, la normalisation ICS permet à l'algorithme des K-means de retrouver les vrais clusters sous-jacents. La variance des variables $\mathbf{y}^2, \mathbf{y}^3$ après normalisation ICS a été réduite en moyenne à 0.786 sur les simulations alors qu'évidemment celle de $\mathbf{y}^1$ est restée inchangée (autour de 1). Finalement, l'information apportée par $\mathbf{y}^2, \mathbf{y}^3$ étant la même, leur contribution à la variance s'est vue réduite.

Dans cet exemple, la normalisation ICS fonctionne car ce sont les variables non importantes qui sont corrélées. Il est légitime de se demander si en revanche, lorsque la corrélation porte sur les variables importantes, cette normalisation ne détériore pas le clustering. Cela est abordée dans la suite (Section 4.5).

4.4.2 Lien avec l'ACP

Pour étudier les propriétés de la méthode de normalisation ICS, nous analysons son lien avec l'ACP. Rappelons que les colonnes de $\mathbf{X}$ sont centrées en 0 et de variance 1 et que nous calculons le coefficient suivant :

$$\nu_j^2 = \sum_{l=1}^{p} \text{cor}(\mathbf{x}^l, \mathbf{x}^j)^2, \forall j = 1, \ldots, p.$$

Soit $\hat{\mathbf{\Sigma}}_\mathbf{X}$ la matrice de corrélation des données $\mathbf{X}$ dont le terme $(\hat{\mathbf{\Sigma}}_\mathbf{X})_{l,j} = \text{cor}(\mathbf{x}^l, \mathbf{x}^j)$. On vérifie que

$$\nu_j^2 = (\hat{\mathbf{\Sigma}}_\mathbf{X}^\intercal \hat{\mathbf{\Sigma}}_\mathbf{X})_{j,j},$$

puisque la matrice de corrélation est symétrique. On sait que $\hat{\mathbf{\Sigma}}_\mathbf{X}$ est diagonalisable et qu'elle admet une base orthonormée de vecteurs propres. On pose $\mathbf{V} \in \mathbb{R}^{p \times p}$ la matrice orthogonale dont les colonnes sont les vecteurs propres $(\mathbf{v}^1, ..., \mathbf{v}^p)$ associés aux valeurs propres $\lambda_1, ..., \lambda_p$ classées par ordre décroissant.

Soit $\mathbf{\Lambda} = \text{diag}(\lambda_1, ..., \lambda_p) \in \mathbb{R}^{p \times p}$ la matrice diagonale dont les éléments diagonaux sont les valeurs propres de $\hat{\mathbf{\Sigma}}_\mathbf{X}$. On a alors $\hat{\mathbf{\Sigma}}_\mathbf{X} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\intercal$, donc

$$\hat{\mathbf{\Sigma}}_\mathbf{X}^\intercal \hat{\mathbf{\Sigma}}_\mathbf{X} = \mathbf{V}\mathbf{\Lambda}^2\mathbf{V}^\intercal \text{ et } \nu_j^2 = (\hat{\mathbf{\Sigma}}_\mathbf{X}^\intercal \hat{\mathbf{\Sigma}}_\mathbf{X})_{j,j} = \sum_{\alpha=1}^{p} (\mathbf{v}_j^\alpha)^2 \lambda_\alpha^2.$$

Si on note $\mathbf{F}$ la matrice dont les colonnes sont les composantes principales $\mathbf{f}^1, ..., \mathbf{f}^p$ (coordonnées dans la nouvelle base des vecteurs propres). On a que $\mathbf{F} = \mathbf{X}\mathbf{V}$ et $\forall \alpha = 1, ..., p$, $\mathbf{f}^\alpha = \mathbf{X}\mathbf{v}^\alpha$. On sait que $\forall j = 1, ..., p$, $\forall \alpha = 1, ..., p$:

$$\text{cor}(\mathbf{x}^j, \mathbf{f}^\alpha) = \sqrt{\lambda_\alpha}\mathbf{v}_j^\alpha. \tag{4.3}$$

Ainsi, on obtient l'écriture suivante :

$$\nu_j^2 = \sum_{\alpha=1}^{p} \text{cor}(\mathbf{x}^j, \mathbf{f}^\alpha)^2 \lambda_\alpha.$$

Le terme de la normalisation ICS pour la variable $\mathbf{x}^j$ correspond à la somme des variances des composantes principales pondérées par les corrélations de la variable $\mathbf{x}^j$ aux composantes principales. En clustering, nous considérons que la corrélation amène une redondance, et celle-ci est quantifiée par les valeurs propres. Comme cette redondance dégrade les performances des algorithmes de clustering (basés sur la distance euclidienne) et notamment des K-means, il peut être intéressant de normaliser les variables par leur contribution à la redondance dans les données.

REMARQUE SUPPLÉMENTAIRE De l'Équation (4.3) on déduit que $\lambda_\alpha = \sum_{j=1}^{p} \text{cor}(\mathbf{x}^j, \mathbf{f}^\alpha)^2$. Définissons la corrélation totale comme la somme de tous les termes au carré de la matrice de corrélation :

$$\sum_{j=1}^{p}\sum_{l=1}^{p} \text{cor}(\mathbf{x}^j, \mathbf{x}^l)^2 = \sum_{j=1}^{p} \nu_j^2,$$

et donc

$$\sum_{j=1}^{p} \nu_j^2 = \sum_{j=1}^{p}\sum_{\alpha=1}^{p} \text{cor}(\mathbf{x}^j, \mathbf{f}^\alpha)^2 \lambda_\alpha = \sum_{\alpha=1}^{p} \lambda_\alpha^2.$$

On a que λ_α^2 est la *part* de la corrélation totale expliquée par la composante principale α et donc ν_j^2 peut s'interpréter comme la *part* de la corrélation totale expliquée par la variable j.

4.5 Simulations

LE MODÈLE GLOBAL Les simulations ont pour but ici d'évaluer et de comparer des algorithmes de clustering, sparse ou non, dans le cas où il y a des groupes de variables importantes et de variables de bruit corrélées. Ainsi, nous allons reprendre le schéma de simulation vu à la Section 2.6 du Chapitre 2 mais nous allons ajouter un nouveau groupe de variables et il y a donc trois groupes de variables :

- le groupe des variables importantes qui a une structure en clusters simulée par un mélange de gaussiennes. On note p_K le nombre de variables importantes.

- le groupe des variables de bruit, dont les variables sont indépendantes du clustering, indépendantes des variables importantes et indépendantes entre elles. Elles suivent ici une distribution gaussienne sphérique. On note d le nombre de variables de bruit indépendantes.
- le groupe des variables de bruit corrélées, dont les variables sont indépendantes du clustering, indépendantes des variables importantes et indépendantes du premier groupe de variables de bruit mais dépendantes (corrélées) entre elles. Elles suivent une distribution gaussienne. On note q le nombre de variables de bruit corrélées.

Le nombre total de variables est alors $p = p_K + d + q$. Le modèle de mélange sous-jacent est un mélange de 2 gaussiennes sphériques équiprobables qui s'écrit comme :

$$\sum_{k=1}^{2} \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \text{ avec } \boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2 = (\mathbf{m}_1, 0, ..., 0)_p^\intercal \text{ et donc } \mathbf{m}_1 = -\mathbf{m}_2 = (m, ..., m)_{p_K}^\intercal \in \mathbb{R}^{p_K}, \tag{4.4}$$

où $\boldsymbol{\Sigma}_k = \begin{pmatrix} \mathbf{I}_{p_K} & 0 & 0 \\ 0 & \mathbf{I}_d & 0 \\ 0 & 0 & \mathbf{S}_q \end{pmatrix}$ est une matrice par blocs où les 0 représentent ici des matrices de 0 avec les tailles adaptées et $\mathbf{S}_q$ a 1 sur toute sa diagonale et r ailleurs.

REMARQUE SUR LA MATRICE DE CORRÉLATION DES DONNÉES Soit $\mathbf{X}_q \in \mathbf{X}$ la matrice contenant les q variables de bruits corrélées. Nous pouvons remarquer que sa matrice de covariance associée $\hat{\boldsymbol{\Sigma}}_{\mathbf{X}_q} = \mathbf{S}_q$ car les deux gaussiennes du mélange décrit par l'Équation (4.4) se superposent (même moyenne). Il en va de même si l'on considère $\mathbf{X}_d \in \mathbf{X}$ la matrice contenant les d variables de bruits indépendantes, où $\hat{\boldsymbol{\Sigma}}_{\mathbf{X}_d} = \mathbf{I}_d$. En revanche cela est faux pour $\mathbf{X}_{p_K} \in \mathbf{X}$ la matrice des p_K variables importantes. En effet, $\hat{\boldsymbol{\Sigma}}_{\mathbf{X}_{p_K}} \neq \mathbf{I}_{p_K}$ car $\mathbf{m}_1 \neq \mathbf{m}_2$. Finalement, nous obtenons la matrice de covariance des données :

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{X}} \approx \begin{pmatrix} \mathbf{S}_{\mathbf{X}_{p_K}} & 0 & 0 \\ 0 & \mathbf{I}_d & 0 \\ 0 & 0 & \mathbf{S}_q \end{pmatrix},$$

une matrice par blocs où les 0 représentent ici des matrices de 0 avec les tailles adaptées, où les éléments diagonaux de $\mathbf{S}_q$ sont égaux à 1, alors que les éléments non diagonaux sont égaux à r. Quant à la matrice $\mathbf{S}_{\mathbf{X}_{p_K}}$, tous ses éléments sont égaux à c, à l'exception des éléments diagonaux. La valeur de c dépend uniquement (dans notre cas) des valeurs de $\mathbf{m}_1, \mathbf{m}_2$. Donnons deux exemples, que l'on utilisera par la suite, si $m = 2$ (respectivement $m = 0.85$) et dans le cas où les données centrées réduites alors $c \approx 0.8$ (respectivement $c \approx 0.42$).

4.5.1 Simulations pour des algorithmes de clustering non sparse

DESCRIPTION DES SIMULATIONS Nous reprenons le schéma de simulation décrit par l'Équation 4.4, nous allons comparer l'algorithme des K-means avec les modèles GMM. Les résultats sont présentées Table 4.2. Pour les K-means, nous comparons deux modèles :

1. l'algorithme des K-means sur des données centrées réduites (KM dans la Table 4.2) ;
2. l'algorithme des K-means sur les données normalisées par ICS (après avoir été centrées réduites) (KM (ICS) dans la Table 4.2).

Pour les GMM, nous utilisons le package `R mclust` et nous utilisons deux modèles :

1. l'algorithme des GMM où la vraie forme de matrice de covariance est connue et donnée au modèle (mclust dans la Table 4.2), c'est-à-dire que le volume des gaussiennes, leur forme et leur orientation sont égaux pour les deux clusters et que leur distribution est ellipsoïdale ce qui correspond à `modelNames ="EEE"` dans le package `mclust` ;
2. l'algorithme des GMM où toutes les formes contraintes et non contraintes de la matrice de covariance sont testées. On a donc 14 modèles gaussiens obtenus par décomposition spectrale comme expliquée Section 2.4.1, et le meilleur modèle est déterminé grâce au BIC (mclust14 dans la Table 4.2).

Il faut noter que le package `mclust` impose une initialisation résultant d'un clustering hiérarchique agglomératif basé sur des critères de vraisemblance maximale pour les modèles de mélange gaussien paramétrés par décomposition des valeurs propres (Scrucca et al., 2016).

RÉSULTATS DES SIMULATIONS Pour tous les modèles, le nombre de clusters K est donné, avec on le rappelle $K = 2$ pour toutes les simulations. Par ailleurs on fixe $n = 120, p_K = 2$, avec $r = 0.7, 0.95$ selon le scénario et $m = 2$ de sorte que les clusters soient bien séparés. Les résultats agrégés sont présentés dans la Table 4.2. Dans les scénarios nous avons introduit peu de variables et notamment peu de variables de bruit car les méthodes utilisées ne permettent pas de gérer explicitement ce type de variables.

Table 4.2 : Le tableau représente les moyennes et écarts-types de l'ARI pour chaque méthode pour le schéma de simulation décrit par l'Équation 4.4 avec $n = 120, m = 2$ et $p_K = 2$ pour 100 simulations. Les valeurs de d, q et r suivant les scénarios sont indiquées dans le tableau.

r	d	q	KM	KM(ICS)	mclust	mclust14
-	5	0	**0.98**(0.03)	**0.95**(0.06)	**0.97**(0.03)	**0.99**(0.04)
0.7	0	5	0.01(0.03)	**0.98**(0.03)	**0.61**(0.40)	**0.80**(0.51)
0.7	5	5	0.02(0.04)	**0.86**(0.24)	**0.61**(0.40)	**0.60**(0.41)
0.95	0	5	0.01(0.03)	**0.98**(0.03)	**0.51**(0.48)	**0.60**(0.42)
0.95	5	5	0.00(0.01)	**0.87**(0.25)	0.22(0.33)	**0.38**(0.42)

La Table 4.2 montre que la normalisation ICS contribue largement à améliorer les résultats de clustering de l'algorithme des K-means en présence de variables corrélées. Elle a aussi un impact négatif limité lorsque les variables corrélées sont uniquement des variables importantes (le cas $q = 0$). Par ailleurs, on constate que les GMM (package `mclust`) obtiennent de moins bons résultats pour ce schéma de simulation. En effet, soit les GMM retrouvent les clusters sous-jacents, soit la partition obtenue dépend des variables corrélées. En outre, nous observons que les GMM obtiennent des meilleurs résultats lorsque la matrice de covariance n'est pas contrainte.

En conclusion, la normalisation semble intéressante lorsque les données contiennent des variables de bruit corrélées. Néanmoins, en présence de variables de bruit il peut être plus approprié d'utiliser des algorithmes de clustering sparse et cela nous amène à étudier le comportement des algorithmes des K-means sparses sur des données normalisées avec ICS.

4.5.2 Simulations pour les algorithmes de clustering sparse

Dans cette section, nous comparons les algorithmes de clustering sparse utilisés dans les simulations présentées dans la Section 2.7. Les mêmes paramètres pour les fonctions `R` sont utilisés et nous ajoutons deux algorithmes à comparer : le WT-K-means avec détection de ruptures sur les données normalisées par ICS (après avoir été centrées réduites) que l'on note WT-KM rupture ICS et le WT-K-means avec le Gap Statistic sur les données normalisées avec ICS (après avoir été centrées réduites) que l'on note sparcl ICS. Les données ne sont pas centrées réduites en entrée des algorithmes GMM sparses sauf pour le package vscc qui se base sur des critères d'inertie. Nous reprenons le schéma de simulation décrit par l'Équation (4.4) avec $K = 2, n = 120$ et nous fixons $m = 0.85$ comme pour les simulations de la Section 2.7 et $r = 0.7$ ce qui peut donner un bon aperçu du comportement de ces algorithmes sur des données réelles. Nous reprenons le même nombre de variables de bruit que dans la Section 2.7 mais cette fois nous formons deux groupes avec des variables indépendantes et des variables corrélées. Ainsi, nous considérons trois scénarios :

1. $p_K = 10, d = 50, q = 0$ qui est le scénario le plus défavorable pour la méthode de normalisation ICS ;
2. $p_K = 10, d = 0, q = 50$ qui permet d'observer l'impact de la présence des variables de bruit corrélées par rapport aux variables de bruit indépendantes sur les algorithmes ;
3. $p_K = 10, d = 50, q = 50$ qui est le scénario le plus difficile mais aussi le plus réaliste.

Soit $\mathcal{S}$ l'ensemble des indices des variables sélectionnées par un algorithme de clustering sparse, et soit :

- $\mathcal{S}^C = \{j : \mathbf{x}^j \in \mathbf{X}_{p_K}, j \in \mathcal{S}\}$ l'ensemble des indices des variables importantes sélectionnées par l'algorithme ;
- $\mathcal{S}^N = \{j : \mathbf{x}^j \in \mathbf{X}_d, j \in \mathcal{S}\}$ l'ensemble des indices des variables de bruit sélectionnées par l'algorithme ;
- $p_{\mathcal{S}^C} = \text{card}(\mathcal{S}^C)$ le nombre de variables importantes sélectionnées par l'algorithme ;
- $p_{\mathcal{S}^N} = \text{card}(\mathcal{S}^N)$ le nombre de variables de bruit sélectionnées par l'algorithme.

Plusieurs mesures sont utilisées pour comparer les méthodes :

- l'ARI (défini Section 1.4 Équation (1.1)) ;
- le ratio de variables importantes sélectionnées $\frac{p_{\mathcal{S}^C}}{p_K}$;
- le ratio de variables de bruit sélectionnées $\frac{p_{\mathcal{S}^N}}{d}$;
- le temps de calcul des algorithmes en secondes.

20 simulations des scénarios seront testées et agrégées et la moyenne et l'écart type des résultats sont calculés.

4.5.2.a Résultats dans le scénario 1 : $p_K = 10; d = 50, q = 0$

Les résultats du scénario 1 sont présentés dans la Table 4.3. Toutes les méthodes sont équivalentes sauf celles des packages `vscc`, `clustvarsel` et `SelVarMix` qui ne parviennent pas à retrouver les clusters. Comme précédemment, nous observons que la méthode du package `sparcl` (WT-K-means avec le Gap Statistic) sélectionne toutes les variables de bruit contrairement à notre solution WT-KM_rupture. De plus, la normalisation ICS n'a pas de grandes incidences sur les résultats car les mêmes algorithmes avec et sans normalisation ICS ont des ARI, des ratios de variables importantes et de bruits et des temps de calculs qui sont équivalents et c'est ce qui était souhaité pour ce scénario.

Table 4.3 : Le tableau représente les moyennes (moyenne) et écarts-types (sd) de l'ARI par méthode pour le schéma de simulation décrit par l'Équation 4.4 avec $n = 120, m = 0.85$ sur 20 simulations. Les nombres de variables par groupe sont $p_K = 10; d = 50; q = 0$.

	ARI		Ratio V.Imp		Ratio V.Bruit		Temps (s)	
algorithmes	moyenne	sd	moyenne	sd	moyenne	sd	moyenne	sd
sparcl	**0.98**	0.03	**1.00**	0.00	1.00	0.00	7.24	0.25
SFEM	**0.98**	0.04	**1.00**	0.00	0.93	0.03	39.46	3.49
VarSelLCM	**0.98**	0.04	**1.00**	0.00	0.02	0.01	5.99	0.12
sparcl ICS	**0.97**	0.05	**1.00**	0.00	1.00	0.00	6.58	0.19
WT-KM_rupture ICS	**0.93**	0.09	**0.78**	0.24	**0.00**	0.00	**1.30**	0.05
WT-KM_rupture	**0.90**	0.11	0.71	0.25	**0.00**	0.00	**1.12**	0.08
clustvarsel	0.30	0.44	0.36	0.45	0.17	0.14	27.64	6.63
SelVarMix	0.20	0.42	0.36	0.34	0.49	0.52	3.37	0.17
vscc	0.00	0.00	1.00	0.00	1.00	0.00	**1.43**	0.28

4.5.2.b Résultats dans le scénario 2 : $p_K = 10; d = 0; q = 50$

Les résultats du scénario 2 sont présentés dans la Table 4.4. Cette fois-ci, seules les méthodes où les données d'entrée ont été normalisées avec ICS et la fonction du package `SelVarMix` donnent des résultats corrects. Nous avons constaté que les méthodes de GMM sparses ne fonctionnent pas pour le scénario 2 contrairement au scénario 1, alors que la seule différence entre les deux scénarios est la présence de corrélation entre les variables de bruit (le nombre de variables de bruit est le même). Par ailleurs, la fonction du package `sparcl` sélectionne toutes les variables de bruit ce qui n'est pas un bon résultat du point de vue de la sélection de variables.

Table 4.4 : Le tableau représente les moyennes (moyenne) et écarts-types (sd) de l'ARI par méthode pour le schéma de simulation décrit par l'Équation 4.4 avec $n = 120, m = 0.85$ et $r = 0.7$ sur 20 simulations. Les nombres de variables par groupe sont $p_K = 10; d = 0; q = 50$.

	ARI		Ratio V.Imp		Ratio V.Bruit		Temps (s)	
algorithmes	moyenne	sd	moyenne	sd	moyenne	sd	moyenne	sd
sparcl ICS	**0.97**	0.03	**1.00**	0.00	1.00	0.00	6.93	0.38
WT-KM rupture ICS	**0.92**	0.12	0.71	0.23	**0.00**	0.00	**1.69**	0.36
SelVarMix	**0.83**	0.35	0.80	0.35	0.86	0.07	4.24	0.20
clustvarsel	0.10	0.32	0.11	0.31	0.08	0.07	24.37	10.37
SFEM	0.00	0.02	0.80	0.12	0.71	0.07	67.64	2.92
sparcl	0.00	0.01	1.00	0.00	1.00	0.00	8.10	0.83
WT-KM rupture	0.00	0.01	0.00	0.00	0.12	0.09	**1.40**	0.19
VarSelLCM	0.00	0.01	0.01	0.03	0.51	0.01	6.32	0.45
vscc	0.00	0.01	0.30	0.48	0.60	0.29	**1.75**	1.70

4.5.2.c Résultats pour le scénario 3 : $p_K = 10; d = 50; q = 50$

Les résultats du scénario 3 sont disponibles dans la Table 4.5. Cette fois, seules les méthodes où les données en entrée des algorithmes ont été normalisées avec ICS fonctionnent correctement (même si les écarts-types se chevauchent pour `SelVarMix`) et finalement les résultats sont très similaires à ceux qui ont été présentés pour le scénario 2 dans la Table 4.4. Nous pouvons noter que nous obtenons de meilleurs résultats que dans la Section 2.7 pour le scénario $p_K = 10; d = 100$ décrit dans la Table 2.4. Le nombre de variables de bruit est le même mais la quantité de bruit est moindre, car les $q = 50$ variables de bruit corrélées impliquent de l'information redondante.

Table 4.5 : Le tableau représente les moyennes (moyenne) et écarts-types (sd) de l'ARI par méthode pour le schéma de simulation décrit par l'Équation 4.4 avec $n = 120, m = 0.85$ et $r = 0.7$ sur 20 simulations. Les nombres de variables par groupe sont $p_K = 10; d = 50; q = 50$.

	ARI		Ratio V.Imp		Ratio V.Bruit		Temps (s)	
algorithmes	moyenne	sd	moyenne	sd	moyenne	sd	moyenne	sd
sparcl ICS	**0.97**	0.03	**1.00**	0.00	1.00	0.00	6.58	0.28
WT-KM rupture ICS	**0.92**	0.12	0.71	0.23	**0.00**	0.00	**1.62**	0.34
SelVarMix	**0.53**	0.50	0.54	0.45	0.86	0.10	4.16	0.19
clustvarsel	0.10	0.32	0.11	0.31	0.08	0.07	24.28	10.43
SFEM	0.00	0.02	0.80	0.12	0.71	0.07	65.94	4.35
sparcl	0.00	0.01	1.00	0.00	1.00	0.00	7.10	0.62
WT-KM rupture	0.00	0.01	0.00	0.00	0.12	0.09	**1.32**	0.18
VarSelLCM	0.00	0.01	0.01	0.03	0.51	0.01	6.20	0.46
vscc	0.00	0.01	0.30	0.48	0.60	0.29	**1.72**	1.69

4.5.3 Détails des résultats

Les résultats des simulations mettent en lumière un point important : aucune des méthodes testées ne semble donner de bons résultats pour des données contenant des variables de bruit corrélées (non importantes), mis à part les méthodes avec la normalisation ICS. Les scénarios présentés sont simples et réalistes. Les algorithmes utilisant le WT-K-means sont basés sur l'algorithme des K-means (donc la distance euclidienne) et ils sont donc affectés par les corrélations entre les variables. Ainsi la normalisation ICS est utile pour ces algorithmes. Par ailleurs, le WT-K-means avec le Gap Statistic (`sparcl`) sélectionne toutes les variables importantes et toutes les variables de bruit. En fait, c'est simplement le modèle avec une pénalité nulle, correspondant à $\lambda = 0$, qui est à chaque fois choisi. Les variables de bruit ont des poids très faibles, proches de 0 et l'ARI n'est pas affecté. En un sens c'est donc un bon modèle du point de vue de l'ARI. En revanche, sur des données réelles, le groupe de variables importantes peut contenir des variables qui portent plus ou moins d'informations sur le clustering, ce qui donne des poids à $\lambda = 0$ beaucoup plus bruités. Dans la Section 2.7, sur un schéma avec plus de variables de bruit indépendantes, l'algorithme le WT-K-means avec le Gap Statistic avait déjà des difficultés pour la tâche de sélection de variables.

La fonction du package `SelVarMix` a des résultats compliqués à analyser. Après vérification, il peut arriver que le modèle donne un clustering en une seule classe alors même que $K = 2$ était fixé dans les paramètres (et que $K = 2$ est bien affiché dans le paramètre de sortie `nbcluster` : *The selected number of clusters*). L'obtention de cette partition triviale est due au fait que l'algorithme vide un cluster.

LES MODÈLES GMM SPARSES HÉRITENT DES PROBLÈMES DES GMM Dans la Section 4.5.1, on a vu que la méthode implémentée dans le package *mclust* ne parvient pas toujours à retrouver les clusters dans des cas simples de variables de bruit corrélées. De plus, le package *clustvarsel* permet de faire de la sélection de variables à l'aide d'une procédure stepwise forward-backward (Section 2.4.4) en considérant tour à tour chaque variable qui ne fait pas partie des variables déjà sélectionnées, et évaluant s'il est pertinent de les ajouter en se basant sur des modèles GMM modélisés à l'aide de `mclust`. Donc, si au départ `mclust` ne sélectionne pas le bon modèle, alors il semble cohérent que la procédure tout entière soit compromise.

De même, le package `vscc` est basé sur le package `mclust` et donc cette méthode hérite des mêmes problèmes.

LES HYPOTHÈSES DE CERTAINS MODÈLES GMM SPARSES Les mauvais résultats de certaines méthodes de GMM sparses peuvent être attribués aux hypothèses faites par ces modèles sur la structure du bruit. En effet, SFEM suppose que le bruit est isotrope, c'est-à-dire que la matrice de covariance des variables de bruit est diagonale

et que les éléments diagonaux sont tous égaux, c'est-à-dire que le bloc correspondant à la covariance du bruit s'écrit $\text{diag}(\beta_k, ..., \beta_k) \in \mathbb{R}^{p-d}$ dans le cas le plus général, où p est le nombre de variables de départ, d est la dimension de l'espace latent et β_k détermine la variance du bruit par cluster (Bouveyron and Brunet-Saumard, 2014a,b). En effet, SFEM sépare totalement le signal de clustering et le bruit dans deux sous-espaces orthogonaux de dimension d et $p-d$ respectivement. Le sous-espace discriminant, de faible dimension d, ne contient pas de bruit, tandis que son orthogonal contient un bruit isotrope de variance β_k dans les $p-d$ directions restantes, qui ne peuvent donc pas être corrélées.

Pour le modèle VarSelLCM (voir aussi la Section 2.4.4), Matthieu Marbac et Mohammed Sedki indiquent qu'ils supposent l'indépendance conditionnelle des variables dans chaque cluster, et donc que les variables ne sont pas corrélées conditionnellement aux classes (Marbac and Sedki, 2017). En outre, une variable est dite non importante lorsque ses moyennes et ses variances sont égales pour tous les clusters $\mu_{1,j} = ... = \mu_{K,j}$ et $\sigma_{1,j} = ... = \sigma_{K,j}$.

4.6 Conclusion

Finalement l'algorithme WT-K-means avec détection de rupture dont les données d'entrée sont normalisées avec ICS, offre un bon compromis en termes de performances, de sélection de variables et de coût algorithmique. Il peut être utilisé comme un outil exploratoire en grande dimension et sur des données très corrélées.

La normalisation ICS présente des avantages que nous avons listés ci-dessous.

1. Premièrement la solution est simple et rapide. En effet, elle est très facilement implémentable en une ou deux lignes de code et elle requiert un temps de calcul très faible ce qui donne un temps d'exécution supplémentaire négligeable pour la majorité des applications.

2. La normalisation ICS peut être utilisée en entrée de n'importe quel algorithme. Cela est surtout utile pour les algorithmes basés sur la distance euclidienne. En revanche, son intérêt semble plus limité pour les GMM sauf si on prend en compte leur initialisation.

3. L'algorithme peut être utilisé en grande dimension c'est-à-dire lorsque $p >> n$, ce qui n'est pas toujours le cas de toutes les autres solutions comme le blanchiment de Mahalanobis par exemple, car celui-ci nécessite alors l'inversion d'une matrice singulière.

4. La normalisation ICS permet de faire du clustering sparse. Rappelons que les solutions présentées dans l'état de l'art de ce chapitre Section 4.2 utilisent des matrices de rotation de sorte que les variables résultantes sont des combinaisons linéaires de toutes les variables de départ, ce qui ne permet pas de faire de la sélection des variables.

5. Dans les simulations, la normalisation proposée a un impact négatif très limité sur les résultats des algorithmes clustering.

Toutefois, en toute honnêteté, on doit ajouter que la normalisation par ICS présente au moins une limitation. Dans le cas où les variables corrélées sont uniquement les variables importantes, la normalisation par ICS réduit la variance et le poids de ces variables dans le clustering, ce qui est un inconvénient. Mais, il en est de même pour la normalisation à variance unitaire, dans le cas où les variables importantes ont des variances plus grandes que les variables de bruit, et pourtant cette normalisation reste très majoritairement utilisée. De plus, la présence de corrélations uniquement entre les variables importantes est possible mais improbable en pratique. Nous avons mené des expériences numériques avec l'algorithme WT-K-means sur des données normalisées avec la méthode ICS, pour tester plus précisément la robustesse de l'algorithme lorsque les corrélations portent sur les variables importantes, et la méthode est résiliente.

Dans les perspectives, nous pouvons lister trois points qui peuvent faire l'objet de travaux futurs.

1. La méthode est uniquement bien définie pour les algorithmes de clustering basés sur la distance euclidienne. Nous pensons qu'elle peut être facilement étendue à d'autres mesures de similarité si nécessaire, ce sera l'objet d'une recherche future.

2. La solution s'étend naturellement au cas des données mixtes, très utile dans les applications. Il faut alors définir une mesure de similarité entre deux variables de tout type, numérique et catégorielle. En transformant les variables catégorielles en groupes de variables binaires (Section 3.3.3) on peut utiliser l'analyse canonique des corrélations pour calculer une similarité entre deux groupes de variables (Chavent et al., 2012) et notre solution pourrait alors s'appliquer à la matrice des similarités obtenue. Cette extension n'est pas approfondie dans le cadre de cette thèse mais représente une direction de recherche future.

3. La normalisation par ICS considère les corrélations entre les variables, or les corrélations n'indiquent que des dépendances linéaires. Il n'est pas évident d'imaginer comment des dépendances non linéaires peuvent affecter la distance euclidienne et a priori on peut penser que cela sera dans une moindre mesure. En effet, si on considère deux variables identiques, elles ont une corrélation de 1 et elles comptent double dans la distance euclidienne alors que ce cas ne peut pas être décrit avec des dépendances non linéaires. De plus, pour former des groupes de variables non linéairement dépendantes, cela nécessiterait autant de relations non linéaires différentes que de variables dans le groupe (exemple $\mathbf{x}^1, (\mathbf{x}^1)^2, (\mathbf{x}^1)^3, ...$).

5

Sélectionner le nombre de clusters K avec un compromis de stabilité : un indice de validation interne

5.1 Introduction 70
5.1.1 Motivations 70
5.1.2 Résumé et contexte 71
5.2 État de l'art 72
5.3 Stabilité en clustering 72
5.3.1 Définition 72
5.3.2 Exemples et limites de la notion de stabilité 73
5.3.3 Les difficultés liées aux méthodes d'évaluation basées sur la stabilité 73
5.4 Stabilité intra et inter-classes 75
5.4.1 Surmonter les limites de la stabilité 75
5.4.2 Stadion : un nouvel indice de validité basé sur la stabilité 75
5.4.3 La mise en oeuvre pratique 76
5.5 Exemples illustratifs 77
5.5.1 Un exemple simple avec les chemins de stabilité 77
5.5.2 Les échecs des méthodes de stabilité basées sur l'échantillonnage 77
5.5.3 Un exemple où un clustering (K-means) en $K^\star$ clusters n'est pas la meilleure partition 79
5.5.4 Trouver $K = 1$: le cas d'un ensemble de données sans structure de clusters 79
5.6 Expériences : *benchmark* 80
5.7 Conclusion 81
5.7.1 Résumé 81
5.7.2 Remarques importantes 82

5.1 Introduction

5.1.1 Motivations

Dans ce chapitre, on compare plusieurs méthodes d'évaluation d'une partition obtenue à l'aide d'un algorithme de clustering et on en propose une nouvelle basée sur la notion de stabilité. De nombreuses méthodes d'évaluation de clustering portent seulement sur le choix du *bon* nombre de clusters K, alors comme nous l'avons montré dans les chapitres précédents, on doit s'intéresser au choix d'autres paramètres, comme le paramètre de régularisation λ du WT-K-means, qui détermine le niveau de la pénalisation. Une méthode permettant de trouver conjointement K et λ serait la bienvenue.

Malheureusement la grande majorité des méthodes d'évaluation en clustering reposent sur des heuristiques et sur des principes qui ne sont pas généralisables. Par exemple, de nombreuses méthodes utilisent la distance

euclidienne et sont donc dépendantes du sous-espace dans lequel on les applique. Ainsi, utiliser les méthodes telles quelles n'est pas une bonne stratégie pour sélectionner le paramètre λ et nous avons effectué quelques tests qui l'ont confirmé.

Par conséquent, nous avons décidé de développer une méthode qui se base sur un principe plus général, la stabilité, qui est l'objet d'un travail commun avec Florent Forest, Jérome Lacaille, Mustapha Lebbah et Hanane Azzag (Mourer et al., 2020a). Malgré les bonnes propriétés de notre solution lorsqu'il s'agit de sélectionner K, il a été difficile de l'étendre au choix du paramètre λ et les détails et les raisons en sont discutés dans la conclusion de ce chapitre.

5.1.2 Résumé et contexte

La difficulté de trouver un critère d'évaluation universel est une conséquence directe du fait que l'objectif d'un clustering est fondamentalement mal défini.

POURQUOI AFFIRMONS NOUS QUE LE CLUSTERING N'A PAS D'OBJECTIF CLAIR ? Comme on le sait, le clustering est une technique d'apprentissage non supervisée qui vise à découvrir la structure de données non étiquetées. Il permet de regrouper les données en clusters (ou groupes) dont les éléments sont similaires entre eux (et tous les éléments similaires sont regroupés ensemble) et dissimilaires des éléments des autres groupes Ben-David (2018). Mais cet objectif est contradictoire en raison de la non-transitivité de la notion de similarité : si A est similaire à B, et B est similaire à C, A n'est pas nécessairement similaire à C. Le clustering étant un problème mal posé, il ne peut être résolu correctement à l'aide de cette définition, et les algorithmes de clustering n'optimisent souvent qu'un seul de ses aspects. Par exemple, les K-means ne garantissent que la séparation des objets dissimilaires, en minimisant l'inertie intra-classes. On peut le voir par exemple en considérant des données issues du tirage d'une seule gaussienne multivariée, alors l'algorithme des 2-means coupe l'ensemble des données au centre de la gaussienne dans une zone de forte densité. Ainsi, de nombreux individus très similaires se retrouvent dans des clusters différents, alors que des individus très éloignés n'appartiennent pas au même cluster. Une classification ascendante hiérarchique (CAH) avec l'option du single linkage produit l'effet inverse en garantissant uniquement que les objets similaires se retrouvent dans le même cluster. Du coup, on voit bien que comparer les algorithmes et sélectionner le meilleur est un défi majeur en clustering (Ben-David, 2018)*.

LES MÉTHODES D'ÉVALUATION SONT SPÉCIFIQUES De nombreuses méthodes d'évaluation existent dans la littérature, mais elles intègrent généralement des hypothèses fortes sur la géométrie des clusters (par exemple, des clusters compacts, sphériques) ou sur la distribution sous-jacente, qui sont spécifiques à une application. Or souvent les praticiens ne connaissent pas exactement la *vraie* structure des données lorsqu'ils choisissent un algorithme de clustering, d'autant plus lorsque le clustering est utilisé comme une méthode d'exploration des données.

LA STABILITÉ COMME PRINCIPE GÉNÉRAL Dans la perspective de définir des objectifs cohérents en clustering, la stabilité est apparue comme un principe naturel et agnostique par rapport au modèle : on peut poser comme principe qu'un algorithme doit trouver des structures *stables* dans les données. Par exemple, si les données sont échantillonnées à plusieurs reprises à partir de la même distribution sous-jacente, ou si on bruite légèrement les données, un *bon* algorithme devrait trouver des partitions similaires. Cependant, il s'avère que la stabilité seule n'est pas un outil bien adapté pour déterminer le nombre de clusters. Par exemple, elle est incapable de détecter si le nombre de clusters retenus est trop petit par rapport à la structure réelle. Ainsi, nous proposons un nouveau principe : un bon clustering doit être stable, et dans chaque cluster, il ne doit pas exister de partition stable. Ce principe conduit à un nouveau critère interne de validation du clustering basé sur deux types de stabilités, entre les clusters et au sein des clusters, ce qui permet de surmonter les limites des méthodes précédentes basées sur la stabilité (entre les clusters uniquement).

RÉSUMÉ DE NOTRE CONTRIBUTION Nous proposons une méthode pour évaluer quantitativement et visuellement un clustering.

- À notre connaissance, il s'agit de la première étude empirique à grande échelle sur l'analyse de la stabilité en clustering.
- Une nouvelle définition du clustering est proposée, basée sur la stabilité entre les clusters et au sein des clusters, ainsi qu'une approche concrète pour l'implémenter pour de nombreuses familles d'algorithmes.

*Make clustering great again.

- Sur la base de cette définition, nous introduisons le critère de différence de stabilité, *Stadion*, un indice interne de validation du clustering. De plus, un outil de visualisation interprétable appelé *stability paths* ou chemin de stabilité est défini.
- Dans nos simulations, nous évaluons la capacité de Stadion à la fois de découvrir la structure en clustering et de sélectionner le bon nombre de clusters sur des ensembles de données et nous le comparons avec des indices internes largement utilisés.

5.2 État de l'art

INDICES INTERNES EN CLUSTERING Les indices internes de clustering mesurent la qualité d'un clustering lorsqu'il n'y a pas de variable cible, ce qui est principalement le cas dans l'exploration non supervisée des données. La majorité des critères internes reposent sur une combinaison de distances entre les clusters et au sein des clusters, mesurant respectivement la séparation des clusters et l'homogénéité. Malheureusement, ceci incorpore un a priori sur la géométrie des clusters (Dunn, 1974 ; Caliński and Harabasz, 1974 ; Davies and Bouldin, 1979 ; Rousseeuw, 1987 ; Ray and Turi, 1999 ; Tibshirani et al., 2001a ; Desgraupes, 2013).

LES ÉTUDES ANALYSANT LA STABILITÉ EN CLUSTERING Analyser la stabilité d'un clustering pour l'évaluer est une technique ancienne. Son origine peut être retracée dès 1973 (Strauss et al., 1973), mais les premiers travaux influents sont de Ben-Hur et al. (2002) ; Lange et al. (2004). À partir de là les méthodes de stabilité ont attiré une attention croissante (Ben-David et al., 2006, 2007 ; Ben-David and Von Luxburg, 2008 ; Von Luxburg, 2010). Ces travaux ont conclu que la stabilité n'est pas un outil bien adapté à la sélection de modèles dans le clustering Shamir and Tishby (2007). En particulier, l'analyse de la stabilité ne permet pas détecter si le nombre de clusters est trop petit par rapport à la réalité lorsque l'on utilise l'algorithme K-means (comme nous le verrons précisément à la Section 5.3.2 la Figure 5.1). Une partition avec trop peu de clusters est en effet stable, sauf pour des distributions parfaitement symétriques. Par exemple, l'algorithme des K-means avec $K = 1$ est toujours stable car les observations ne changent jamais de clusters.

Malgré des efforts théoriques importants, peu d'études empiriques complètes ont été menées. Chaque étude se concentre sur des aspects spécifiques de la stabilité du clustering. Par exemple, Levine and Domany (2001) ; Dudoit and Fridlyand (2002) ; Ben-Hur et al. (2002) ; Lange et al. (2004) ont étudié la perturbation par sous-échantillonnage aléatoire de l'ensemble de données original sans remplacement. La stabilité dans un cadre basé sur un modèle (exemple avec les modèles de mélange gaussien) a été étudiée dans Kerr and Churchill (2001). La perturbation par des projections aléatoires (Smolkin and Ghosh, 2003) et l'ajout d'un bruit aléatoire ont également été considérés (Fridlyand and Dudoit, 2001 ; Möller and Radke, 2006 ; Möller and Radke, 2006). Dans l'ensemble, les évaluations basées sur la stabilité sont en général pertinentes, mais aucune comparaison claire entre les méthodes n'a été effectuée. Comme mentionné dans Von Luxburg (2010) ; Ben-David and Reyzin (2014), une étude approfondie comparant tous les différents protocoles et une évaluation plus objective de ces résultats est nécessaire.

5.3 Stabilité en clustering

5.3.1 Définition

La stabilité des méthodes de clustering peut être définie dans le cadre de l'apprentissage statistique standard. Soit un ensemble de données $\mathbf{X} \in \mathbb{R}^{n \times p}$ formé de n observations indépendantes et identiquement distribuées (i.i.d.), générées selon une distribution de probabilité $\mathcal{P}$ définie dans un espace sous-jacent $\mathcal{X}$. Par définition, un algorithme de clustering $\mathcal{A}$ prend en entrée un ensemble de données $\mathbf{X}$, un paramètre $K \geq 1$, et produit une partition $\mathcal{C}_K = \{C_1, \ldots, C_K\}$ de $\mathbf{X}$ en K ensembles disjoints. Ainsi, un clustering peut être représenté par une fonction $\mathbf{X} \to \{1, \ldots, K\}$ attribuant une classe à chaque élément de l'ensemble de données d'entrée. Certains algorithmes sont munis d'un opérateur d'extension qui leur permet d'attribuer un numéro de classe aux données qui n'ont pas été utilisées pour construire les clusters. Cette opérateur d'extension est donc une fonction $\mathcal{X} \to \{1, \ldots, K\}$ (par exemple, pour l'algorithme des K-means, une nouvelle donnée est affectée à la classe dont le centre est le plus proche).

Si $\mathbf{X}$ et $\mathbf{X}'$ sont deux ensembles de données différents de même taille n, tirés selon la même distribution $\mathcal{P}$, on note $\mathcal{C}_K$ et $\mathcal{C}'_K$ leurs partitions correspondantes construites au moyen de l'algorithme $\mathcal{A}$. Soit s une mesure de similarité telle que $s(\mathcal{C}_K, \mathcal{C}'_K)$ mesure la concordance entre les deux partitions. Les choix possibles pour cette mesure sont détaillés ci-après. Alors la stabilité d'un algorithme de clustering $\mathcal{A}$ est définie comme

l'espérance de la similarité entre les deux partitions :

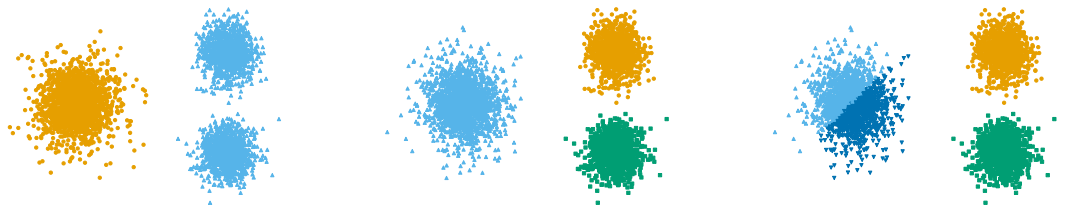
$$\mathrm{Stab}(\mathcal{A}, K) := \mathbb{E}_{\mathbf{X},\mathbf{X}' \sim \mathcal{P}^n} \left[s(\mathcal{C}_K, \mathcal{C}'_K) \right]. \tag{5.1}$$

Cette quantité ne peut être estimée qu'empiriquement, par exemple en construisant $\mathbf{X}'$ comme une version perturbée de $\mathbf{X}$, perturbée par un bruit ou par un ré-échantillonnage.

5.3.2 Exemples et limites de la notion de stabilité

MÉTHODES EXISTANTES Les premières méthodes utilisées dans la littérature sont basées sur le ré-échantillonnage de l'ensemble des données originales, avec ou sans remplacement (division en deux (Strauss et al., 1973), sous-échantillonnage (Ben-Hur et al., 2002), *bootstrapping* (Falasconi et al., 2010 ; Fang and Wang, 2012), *jackknife* (Yeung et al., 2001)). Une autre méthode consiste à ajouter du bruit aléatoire soit directement aux données (Möller and Radke, 2006), soit à la matrice de distance des données (Bilu and Linial, 2012 ; Awasthi et al., 2012 ; Vijayaraghavan et al., 2017 ; Balcan and Liang, 2016 ; Balcan et al., 2020). Pour les données de grande dimension, d'autres alternatives sont les projections aléatoires ou l'ajout ou la suppression aléatoire de variables (Strauss et al., 1973). Une fois les ensembles de données perturbées générés, il existe plusieurs façons de comparer les clustering obtenus. D'abord, avec les méthodes basées sur le bruit, il est possible de comparer la partition des données obtenues avant perturbation (partition de référence) avec les partitions obtenus sur les données perturbées. Ensuite, avec des méthodes basées sur l'échantillonnage, on peut comparer les partitions sur des individus présents dans des sous-échantillons (Falasconi et al., 2010), ou les partitions de l'ensemble des individus en utilisant par exemple un opérateur d'extension ou un algorithme supervisé. Dans ce cas on prend comme variable cible les clusters obtenus et on prédit les numéros des clusters des observations qui n'ont pas été utilisées pour construire la partition initiale Lange et al. (2004). Reste à définir un score de similarité entre deux partitions : les choix courants sont l'ARI déjà abordé dans les chapitres précédents, (Falasconi et al., 2010 ; Zhao et al., 2011), Fowlkes-Mallows, Jaccard (Ben-Hur et al., 2002), l'information mutuelle normalisée (Vinh et al., 2010), la distance minimale de correspondance (Lange et al., 2004), ou la variation d'information (Meila, 2003).

UN EXEMPLE PROBLÉMATIQUE Avant de discuter en détail les propriétés et les problèmes d'évaluation basées sur la stabilité, nous introduisons un exemple trivial qui illustre leur principal défaut : elles ne peuvent pas détecter en général si le nombre de clusters K testé est trop petit. Considérons l'exemple présenté dans la Figure 5.1



(a) $K = 2$ (stable) (b) $K = 3$ (stable) (c) $K = 4$ (instable, *jittering*)

Figure 5.1 : Ensemble de données avec trois clusters. Les classes correspondent au résultat du clustering K-means pour $K = 2, 3, 4$. K-means est stable même si le nombre de clusters est trop petit.

avec trois clusters, dont deux sont plus proches l'un de l'autre que du troisième. Dès que l'on dispose d'une quantité raisonnable de données, l'algorithme des 2-means construit toujours la solution séparant le cluster de gauche des deux clusters de droite. Par conséquent, la solution est stable malgré le fait que $K = 2$ n'est pas le nombre réel de clusters. Cette situation a été soulignée dans Ben-David et al. (2006).

5.3.3 Les difficultés liées aux méthodes d'évaluation basées sur la stabilité

LES DEUX SOURCES D'INSTABILITÉ La stabilité d'un algorithme de clustering est déterminée par le nombre de données qui changent de cluster lorsque l'on perturbe l'ensemble de données. Dans le cas des algorithmes qui minimisent une fonction objectif (par exemple, les K-means ou le clustering spectral), deux sources différentes d'instabilité ont été identifiées (Von Luxburg, 2010). Premièrement, le *sautillement* ou *jittering* en anglais, est causée par les données qui changent de côté aux frontières des clusters après perturbation. Par conséquent, un

fort *jitter* se produit lorsqu'une frontière séparant deux clusters traverse des régions de forte densité. Deuxièmement, le *jumping* fait référence au fait que l'algorithme se retrouve piégé dans différents minima locaux. La cause la plus importante du *jumping* est l'initialisation (voir la Figure 5.3). Une autre cause est l'existence de plusieurs minima globaux de la fonction objectif sur la distribution sous-jacente. Mais cela ne se produit que s'il existe des symétries parfaites dans la distribution (voir la Figure 5.2), ce qui est extrêmement improbable pour des ensembles de données réelles. Deux effets principaux conduisent au *jumping* : premièrement, les symétries dans la distribution des données, et deuxièmement, l'initialisation. Enfin, des propriétés géométriques subtiles de la distribution peuvent également provoquer des sauts Von Luxburg (2010). Un exemple de *jumping* de l'algorithme des K-means dû aux symétries est montré sur la Figure 5.2 : clairement, il y a plusieurs minima globaux, et même si l'algorithme est déterministe, de légères perturbations de la distribution (bruit ou échantillonnage) modifient complètement le résultat. La deuxième cause de *jumping* est due à l'initialisation. Comme l'illustre la Figure 5.3 pour K-means, si une seule initialisation aléatoire est utilisée, en fonction de la position initiale des centres, quatre configurations différentes apparaissent aléatoirement, même sans aucune perturbation des données. Comme nous nous plaçons dans un cadre réaliste sans symétries parfaites et avec

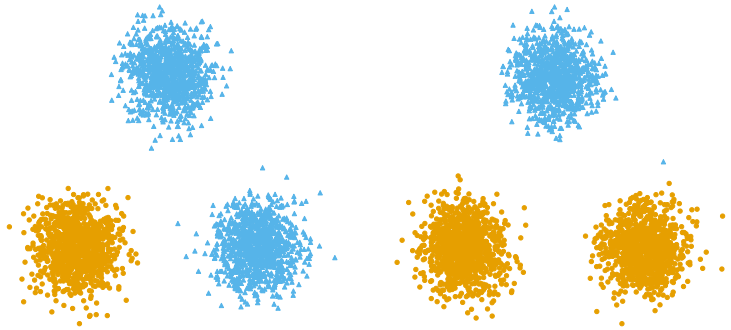
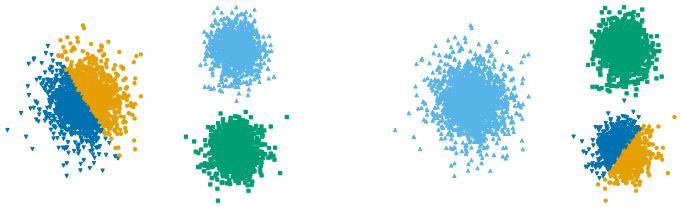


Figure 5.2 : Exemple de K-means faisant du *jumping* entre trois minima globaux pour $K = 2$ sur une distribution symétrique à trois gaussiennes, malgré une initialisation efficace (meilleur K-means++ sur 100 exécutions). Sous une légère perturbation (ici nous avons utilisé un bruit ε-AP gaussien, mais le rééchantillonnage donne des résultats identiques), l'algorithme fait du *jumping*.



(a) (b)

Figure 5.3 : Exemple de *jumping* des K-means entre trois minima locaux pour $K = 4$, lorsqu'une seule initialisation aléatoire est utilisée. Selon la configuration initiale des centres, l'algorithme peut diviser un cluster en deux (par exemple a, b) ou même diviser le cluster de gauche en trois.

une stratégie efficace d'initialisation de l'algorithme, le *jumping* n'est donc une source d'instabilité. Cela est encore plus vrai si l'on considère des indices de stabilité se basant sur des opérateurs d'extension, car la partition de départ ne change jamais et seules les nouvelles données à prédire sont perturbées.

L'IMPORTANCE DE MESURER LA DENSITÉ AUX FRONTIÈRES En fait, les utilisateurs d'algorithmes de clustering utilisent principalement des algorithmes avec des stratégies d'initialisation qui tendent à rendre les algorithmes déterministes. Par exemple, avec les K-means, nous avons déjà abordé l'importance de l'initialisation et les différentes méthodologies existantes dont les K-means++ (Arthur and Vassilvitskii, 2007). Sur des ensembles de données de taille et de dimension raisonnable, les K-means munis d'une bonne initialisation produiront généralement les mêmes résultats. Les méthodes d'initialisation abordées jusqu'ici sont différentes de celle qui est proposée par Von Luxburg (2010) ; Bubeck et al. (2012) qui produit du *jumping* chaque fois que $K > K^\star$, où $K^\star$ est le *vrai* nombre de clusters.

Tout au long de ce chapitre, nous nous plaçons dans le cas où l'on possède suffisamment d'observations, où il n'existe pas de symétries dans les données et où l'algorithme utilisé est muni d'une initialisation efficace ou déterministe. Ainsi, nous ne considérons pas le jumping comme la principale source d'instabilité même lorsque $K > K^\star$ et nous pensons plutôt que c'est le *jittering* qui joue un rôle majeur. Le *jittering* capture l'information utile de la structure du clustering, c'est-à-dire les densités aux frontières. En effet, des zones de forte densité séparées par des zones de faible densité représente l'idéal de ce que l'on recherche et donc s'intéresser aux frontières entre clusters peut nous donner l'information recherchée.

En résumé, nous cherchons à définir une méthode de perturbation qui produit du *jittering*. Malheureusement, quand n est grand, les méthodes de ré-échantillonnage deviennent trivialement stables dès qu'il y a un seul minimum global (Ben-David et al., 2006 ; Von Luxburg, 2010) et des exemples de cette assertion sont disponibles dans la Section 5.5.2.

CONCLUSION La stabilité semble être un principe élégant, mais il existe encore de sérieuses limitations qui rendent son utilisation difficile en pratique. Par exemple, évaluer une méthode de clustering au moyen de l'étude de la stabilité permet de détecter le cas où le nombre K de clusters choisi est trop grand au regard du *vrai* nombre de clusters, grâce notamment au *jittering* qui se produit lorsque l'on perturbe les données avec du bruit ; mais le *jittering* ne se produit pas en général que lorsque K est petit par rapport au *vrai* nombre de clusters car dans ce cas les solutions avec un petit nombre de clusters sont complètement stables. Afin de surmonter cette limitation de l'usage de la stabilité, nous introduisons un nouveau concept de stabilité intra-classes.

5.4 Stabilité intra et inter-classes

5.4.1 Surmonter les limites de la stabilité

Un *bon* algorithme de clustering appliqué avec les mêmes paramètres à des versions perturbées d'un même ensemble de données devrait trouver la même structure et obtenir des résultats similaires. Le bon fonctionnement du principe de stabilité décrit par l'Équation (5.1) repose sur la densité des frontières entre les clusters et nous l'appelons donc stabilité inter-classes. Par conséquent, des structures à l'intérieur des clusters ne peuvent pas être détectées de cette façon comme le montre la Figure 5.1, où l'algorithme avec $K = 2$ est stable, alors qu'un des clusters contient deux sous-clusters. De plus, on ne peut logiquement pas tester $K = 1$ comme solution ce qui pose problème car les ensembles de données réelles ne sont pas toujours structurés en clusters. C'est pour cette raison que nous introduisons une seconde notion de stabilité intra-classes : les clusters ne doivent pas être composés de plusieurs sous clusters. Cela implique l'absence de structures stables à l'intérieur de tout cluster. En d'autres termes, toute partition d'un cluster doit être instable. La combinaison de ces deux principes conduit à une nouvelle définition d'un clustering.

Définition. *Un* bon *clustering est une partition de données en groupes (ou clusters) de sorte que la partition est stable et qu'il n'existe aucune partition stable à l'intérieur de chaque cluster.*

Un *bon* clustering doit donc avoir une grande stabilité entre les clusters et une faible stabilité à l'intérieur des clusters. Malgré leur apparente simplicité, la mise en œuvre de ces principes est une tâche difficile. Comme nous l'avons vu dans la dernière section, la stabilité inter-classes peut être estimée de différentes manières. Cependant, toutes ne sont pas efficaces. D'autre part, la stabilité au sein d'un cluster est une quantité difficile à définir et à estimer. Dans le paragraphe suivant, nous proposons une méthode pour estimer ces deux quantités, puis nous détaillons et justifions nos choix.

5.4.2 Stadion : un nouvel indice de validité basé sur la stabilité

Soit $\{\mathbf{X}^1, \ldots, \mathbf{X}^D\}$ D versions perturbées d'un ensemble de données $\mathbf{X}$, obtenues en ajoutant un bruit aléatoire (de même loi pour chaque version) à l'ensemble original $\mathbf{X}$. La stabilité inter-classes de l'algorithme $\mathcal{A}$ avec le paramètre K est l'espérance dans l'Équation (5.1) estimée par la moyenne empirique des similarités s entre le clustering de référence $\mathcal{C}_K = \mathcal{A}(\mathbf{X}, K)$ et les clusterings obtenus à partir des ensembles de données perturbées,

$$\mathrm{Stab}_{\mathrm{B}}(\mathcal{A}, \mathbf{X}, \mathcal{C}_K, K) := \frac{1}{D} \sum_{d=1}^{D} s\left(\mathcal{C}_K, \mathcal{A}(\mathbf{X}_d, K)\right). \tag{5.2}$$

L'indice B indique qu'il s'agit d'une stabilité inter-clusters (pour *between-clusters* en anglais). Comme s est une mesure de similarité, cette quantité doit être maximisée (et minimisée si on utilisé une mesure de dissimilarité).

Afin de définir la stabilité intra-classes, nous devons évaluer la présence de structures stables à l'intérieur de chaque groupe. Nous proposons de partitionner à nouveau les données au sein de chaque cluster de $\mathcal{C}_K$. Soit $\mathbf{\Omega} \subset \mathbb{N}^*$ un ensemble de nombres de clusters. Le k-ième cluster dans le clustering de référence est noté C_k, son nombre d'éléments est n_k et $\mathcal{Q}_{K'}^{(k)} = \mathcal{A}(C_k, K')$ désigne une partition de C_k en K' clusters. La stabilité intra-classes de l'algorithme $\mathcal{A}$ est définie comme suit

$$\mathrm{Stab}_{\mathrm{W}}(\mathcal{A}, \mathbf{X}, \mathcal{C}_K, K, \mathbf{\Omega}) := \sum_{k=1}^{K} \left(\frac{1}{|\mathbf{\Omega}|} \sum_{K' \in \mathbf{\Omega}} \mathrm{Stab}_{\mathrm{B}}(\mathcal{A}, C_k, \mathcal{Q}_{K'}^{(k)}, K') \right) \times \frac{n_k}{n}. \tag{5.3}$$

Un *bon* clustering doit être instable au sein de chaque cluster et donc il s'agit de minimiser cette quantité. Par conséquent, nous proposons de construire un nouvel indice de validité combinant les stabilités intra et inter-classes. Un choix naturel est la différence entre les deux quantités. Nous appelons cet indice Stadion (*Stability difference criterion*). $\mathcal{A}$, K et $\mathbf{X}$ sont omis dans les notations pour plus de clareté :

$$\mathrm{Stadion}(\mathcal{C}_K, \mathbf{\Omega}) := \mathrm{StabB}(\mathcal{C}_K) - \mathrm{Stab}_{\mathrm{W}}(\mathcal{C}_K, \mathbf{\Omega}). \tag{5.4}$$

La même partition $\mathcal{C}_K$ est utilisée dans les deux termes de L'Équation (5.4). Ainsi, Stadion évalue la stabilité d'un algorithme par rapport à une partition de référence.

5.4.3 La mise en oeuvre pratique

COMMENT PERTURBER LES DONNÉES ? Dans notre cadre réaliste (pas de symétrie globale dans les données et une initialisation robuste), ni le *jumping* ni le *jittering* ne se produiront si les données sont perturbées par des processus d'échantillonnage, dès qu'il y a suffisamment de données. Nous montrons sur un exemple simple que les méthodes basées sur l'échantillonnage telles que celles de Ben-Hur et al. (2002) ; Lange et al. (2004) ne peuvent pas fonctionner dans le cas général (voir Section 5.5). Par conséquent, seule la perturbation basée sur le bruit est considérée ici. Parmi celles-ci, nous adoptons la perturbation additive ε (ε-AP) où le bruit est gaussien ou uniforme, en supposant que les variables sont normalisées à moyenne nulle et variance unitaire. D'après nos simulations, le nombre d'échantillons perturbés D nécessaires peut être faible, de l'ordre de 5, pour avoir de bonnes estimations.

COMMENT CHOISIR ε ? Un compromis doit être pris en compte lors de la perturbation de l'ensemble de données. Si ε-AP est trop fort, nous risquons de modifier la structure même des données. Si au contraire ε-AP est trop faible, l'algorithme de clustering obtiendra toujours des résultats identiques, ce qui conduit inévitablement à de la stabilité. Bien que le choix de cette valeur ne semble pas important selon Möller and Radke (2006) nous pensons toujours qu'il est quelque peu arbitraire et qu'il définit d'une certaine manière implicitement ce qu'est un clustering. Si ε est trop grand, les clusters proches seront fusionnés selon le principe de stabilité. Ainsi, d'une certaine manière, ε-AP définit une distance seuil en dessous de laquelle deux points de données sont similaires et devraient appartenir au même cluster. Nous proposons de contourner ce problème en ne choisissant pas une valeur unique pour le niveau de bruit ε, mais une grille de valeurs possibles. En augmentant progressivement ε de 0 à une valeur $\varepsilon_{\max}$, nous obtenons ce que nous appelons un chemin de stabilité, c'est-à-dire l'évolution de la stabilité en fonction des valeurs ε. Cette méthode a un avantage crucial : elle permet de comparer les partitions pour différentes valeurs de ε sans avoir à en choisir une. Cependant, elle présente deux inconvénients : il faut fixer à la fois la finesse et la valeur maximale de la grille. Dans nos expériences, la finesse ne joue pas un rôle majeur dans les résultats. Nous proposons une méthode simple pour fixer une valeur maximale $\varepsilon_{\max}$: la perturbation correspondant à $\varepsilon_{\max}$ est censée détruire la structure en clusters. Cela correspond à la valeur pour laquelle les données n'ont plus de structure de clusters, c'est-à-dire que $K = 1$ devient la meilleure solution par rapport à Stadion. Une première estimation de $\varepsilon_{\max} = \sqrt{p}$ (où p est la dimension des données) fonctionne bien en pratique. Nous pouvons voir l'utilité des chemins de stabilité (voir Figure 5.4) pour aider à l'interprétation des structures trouvées par un algorithme.

QUELLES DONNÉES COMPARER ? Une grande valeur de bruit ε-AP peut détruire la structure et les clusters proches peuvent fusionner plus rapidement que les autres. Par conséquent, la comparaison des données perturbées entre elles peut devenir bruitée et complexe. Ainsi, nous ne considérons que la comparaison entre les clustering obtenus sur les données de départ et les ensembles de données perturbées. Comme indiqué dans l'Équation (5.2), nous calculons les similarités entre la partition de référence et les partitions obtenus sur les ensembles de données perturbées.

COMMENT COMPARER DES PARTITIONS ? La mesure de similarité s choisie pour comparer deux partitions est l'ARI. Au total, 16 mesures de similarité et de distance différentes ont été comparées et dans nos expériences et il n'y avait pas de différences significatives entre les résultats obtenus suivant les différentes mesures. L'ARI est un choix standard pour bon nombre d'applications et nous décidons de le garder ici.

COMMENT AGRÉGER LES CHEMINS DE STABILITÉ ? Pour calculer un indice, le chemin de Stadion doit être agrégé par rapport au bruit ε pris entre 0 et $\varepsilon_{\max}$ (lorsque la solution pour $K = 1$ a le Stadion le plus élevé parmi toutes les autres solutions). Deux stratégies d'agrégation, le maximum (Stadion-max) et la moyenne (Stadion-mean), sont évaluées dans nos expériences.

La stabilité au sein des clusters est déterminée par le paramètre $\mathbf{\Omega}$. Ne sachant pas le nombre de sous clusters présents, nous calculons la moyenne pour plusieurs valeurs différentes de $\mathbf{\Omega}$. En l'absence de sous clusters, toutes les partitions seront instables car les frontières des clusters seront placées dans des régions de forte densité. En revanche, en présence de sous clusters, au moins certaines partitions auront une stabilité plus élevée, augmentant ainsi la stabilité à l'intérieur du cluster.

Une hypothèse importante nécessaire à notre implémentation du calcul de la stabilité à l'intérieur des clusters est que, pour les structures qui n'ont pas de clusters, l'algorithme doit placer les frontières des clusters dans des régions à forte densité pour produire de l'instabilité via le *jittering*. Cela concerne un large éventail d'algorithmes tels que l'algorithme des K-means, le clustering spectral ou le clustering CAH avec Ward, qui vont séparer des nuages de points en coupant à travers des zones denses. Si un algorithme ne dispose pas de cette propriété, il se peut que la méthode ne fonctionne pas. Par exemple, la CAH avec single linkage ne peut pas être évaluée de cette manière, car elle peut construire des partitions à deux clusters de taille 1 et $n - 1$, où la limite se trouve à la frontière du cluster.

Enfin, nous utilisons le même algorithme de clustering dans le calcul de la stabilité intra et inter-classes. Il serait possible d'évaluer la stabilité intra-classes en utilisant un algorithme différent de celui utilisé pour calculer la stabilité inter-classes, ou bien on pourrait aussi entraîner un algorithme supervisé avec comme variable cible les classes indiquant les clusters estimés et ensuite évaluer la stabilité du classifieur plutôt que celle de l'algorithme de clustering (Dudoit and Fridlyand, 2002 ; Lange et al., 2004 ; Tibshirani and Walther, 2005). Cependant, il est possible d'introduire un biais avec ce type d'approche.

5.5 Exemples illustratifs

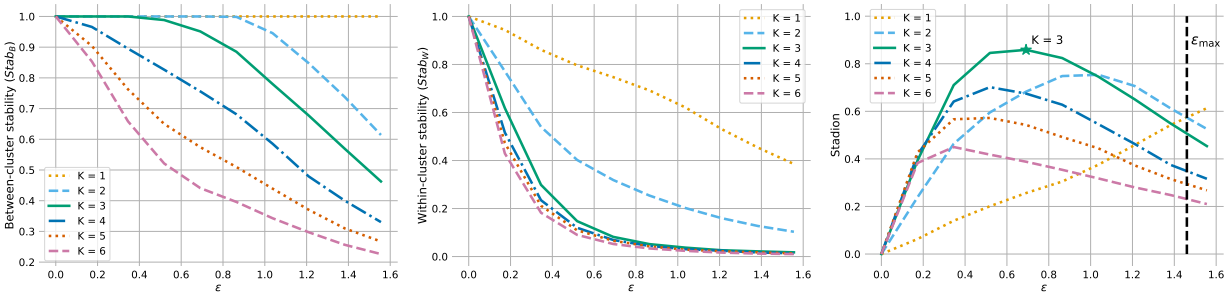
5.5.1 Un exemple simple avec les chemins de stabilité

Nous commençons cette section en illustrant notre méthode avec l'algorithme des K-means et le bruit ε-AP uniforme sur l'exemple de données discuté précédemment (voir Figure 5.1). La Figure 5.4 montre la stabilité inter-classes, la stabilité intra-classes et Stadion en fonction de l'intensité du bruit ε. Pour des quantités raisonnables de bruit, les solutions $K = 1$, $K = 2$ et $K = 3$ sont toutes parfaitement stables, montrant l'insuffisance de la stabilité inter-classes seule pour détecter que K est trop petit. Les solutions pour $K \geq 4$ coupent à travers les clusters et sont donc instables à cause du *jittering*. Cependant, les solutions pour $K = 1$ et $K = 2$ ont toutes deux une grande stabilité intra-classes, causée par la présence de sous clusters ce qui n'est pas le cas pour $K \geq 3$. Notre critère Stadion combine les informations intra et inter-classes et est capable d'indiquer le nombre correct de clusters ($K = 3$) en sélectionnant le chemin Stadion à l'aide de l'aggrégation par la moyenne et le max. Tel qu'il a été défini, Stadion mesure un compromis de stabilité. Les chemins de stabilité donnent également des indications supplémentaires sur la structure des données. Par exemple, nous pouvons lire sur le chemin de stabilité entre les clusters comment les clusters fusionnent successivement lorsque ε augmente. Des exemples supplémentaires sont fournis dans la section 5.5. Enfin, le dernier graphique représente les stabilités moyennes pour différentes valeurs du paramètre K.

5.5.2 Les échecs des méthodes de stabilité basées sur l'échantillonnage

Dans cette section, nous allons voir sur un exemple trivial pourquoi les méthodes de stabilité basées sur l'échantillonnage ne sont pas fiables pour détecter la présence de structure dans les données. Quatre méthodes sont comparées :

1. Stadion basé sur la perturbation additive ε ;
2. Stadion basé sur le *bootstrapping* (les versions bruitées sont obtenues par réechantillonage *bootstrap* ;
3. l'algorithme *Model-explorer* Ben-Hur et al. (2002) basé sur le sous-échantillonnage ;



(a) Stabilté inter-classes (b) Stabilté intra-classes (c) Stadion

Figure 5.4 : Le graphique (a) représente les chemins de stabilité entre les clusters, (b) les chemins de stabilité au sein des clusters, (c) les chemins de Stadion et (d) la courbe de compromis de stabilité pour les K-means sur l'ensemble de données Figure 5.1, pour $K \in \{1 \dots 6\}$. ε est l'amplitude de la perturbation du bruit gaussien. La meilleure solution $K = 3$ est sélectionnée soit en prenant le maximum, soit en faisant la moyenne de Stadion sur ε jusqu'à ε_{max}.

4. la méthode de *Model-order-selection* Lange et al. (2004) basée sur la division des données en deux et le transfert des classes d'une partie à l'autre à l'aide d'un algorithme des plus proches voisins.

Nous montrons que seule la première méthode réussit sur un exemple simple consistant en un mélange de deux gaussiennes corrélées, représenté sur la Figure 5.5. Les données sont normalisées à moyenne nulle et variance unitaire comme pour tout autre ensemble de données. La méthode K-means est utilisée pour construire les clusters. Comme l'illustre le graphique, l'algorithme des K-means avec $K = 2$ sépare presque parfaitement les deux gaussiennes. Toutes les autres solutions divisent les deux gaussiennes en plusieurs sous-clusters de taille égale, les frontières des clusters se trouvant dans des régions de forte densité, comme on peut le voir dans l'exemple pour $K = 4$ (où les frontières se trouvent au milieu des gaussiennes). Cependant, les méthodes basées sur l'échantillonnage ne parviennent pas à évaluer la stabilité inter-classes, puisqu'elles estiment que $K = 4$ est la solution la plus stable. Ce résultat peut s'expliquer par le fait que l'ensemble de données n'est pas symétrique et il existe un minimum global que pour chaque K, de sorte qu'aucun *jumping* ne se produit, même avec un mauvais schéma d'initialisation. Ainsi, la seule source d'instabilité possible provient du *jittering*. Comme prévu en théorie, nos expériences ont montré ici que les différents processus d'échantillonnage n'ont pas réussi à créer du *jittering*. À l'inverse, l'injection d'un bruit ε-AP a effectivement produit du *jittering*, puisque l'introduction d'une petite quantité de bruit a produit des partitions très différentes.



(a) $K = 2$ (b) $K = 4$

Figure 5.5 : Exemple d'ensemble de données constitué de deux gaussiennes corrélées, mises à l'échelle à moyenne nulle et variance unitaire. Avec l'algorithme des moyennes K, toutes les méthodes basées sur l'échantillonnage choisissent $K = 4$ ou $K = 6$, alors qu'avec la perturbation additive ε, $K = 2$ est la seule solution stable.

La méthode de sélection de *Model-order-selection* (Lange et al., 2004) sélectionne $K = 4$, suivi de $K = 6$. *Model-explorer* (Ben-Hur et al., 2002) trouve $K = 6$ comme meilleure solution, suivi de $K = 4$. Ces résultats sont cohérents quelque soit le schéma d'initialisation (aléatoire, pour K-means++, où on conserve la meilleure solution sur plusieurs exécutions). Par conséquent, l'initialisation aléatoire ne contribue pas à créer une instabilité par *jumping*. En outre, notre critère de stabilité Stadion a pu trouver $K = 2$ (voir Figure 5.6) parmi l'ensemble des valeurs testées $\{1, \dots, 6\}$ (ici avec un bruit uniforme et $\mathbf{\Omega} = \{2, \dots, 6\}$) et cela n'est pas seulement dû à l'ajout de la stabilité intra-classes. Pour preuve, nous avons remplacé ε-AP par

une perturbation *bootstrap* : Stadion avec *bootstrap* échoue également, sélectionnant $K = 1$ comme meilleure solution suivie de $K = 4$, et ce pour tous les schémas d'initialisation.

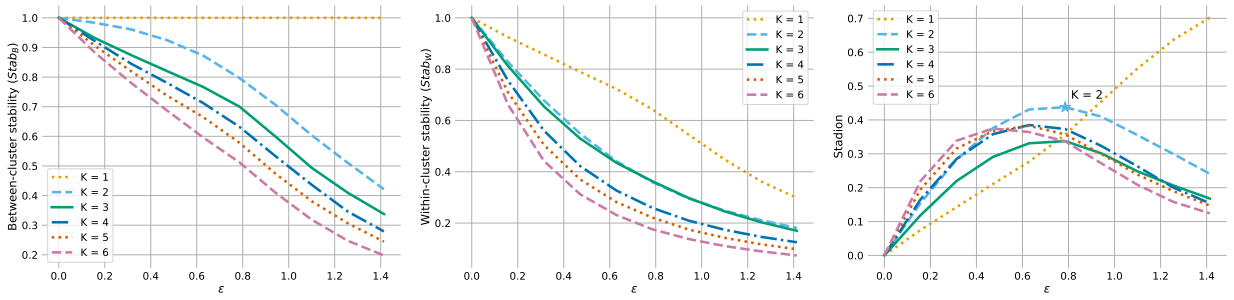


Figure 5.6 : Stabilité inter-classes, stabilité intra-classes et chemins de stabilité (Stadion avec un bruit gaussien), $= \{2, \ldots, 6\}$) sur l'exemple de deux gaussiennes corrélées où toutes les méthodes basées sur l'échantillonnage échouent à sélectionner $K = 2$. Stadion trouve $K = 2$ en prenant le maximum ou la moyenne du chemin.

5.5.3 Un exemple où un clustering (K-means) en $K^\star$ clusters n'est pas la meilleure partition

Parfois, la meilleure solution n'est pas la partition obtenue avec le vrai nombre de clusters $K^\star$, car l'algorithme est incapable de récupérer la partition de vérité. C'est le cas pour l'ensemble de données représenté sur la Figure 5.7. Bien qu'il soit évident que la meilleure solution est de séparer les quatre clusters, elle n'est pas réalisable avec l'algorithme des K-means : avec $K^\star = 4$, il coupera à travers le grand cluster au lieu de séparer les deux petits clusters. Parmi les solutions proposées, l'ARI le plus élevé (par rapport à la partition de base) est obtenu avec $K = 3$ (ARI = 0,92), suivi de $K = 2$ (0,74), $K = 5$ (0,65) et enfin $K^\star = 4$ (0,58).



(a) $K = 2$ (ARI = 0.74) - La majorité des méthodes d'évaluation

(b) $K = 3$ (ARI = 0.92) - Stadion

(c) $K^\star = 4$ (ARI = 0.58) Ben-Hur et al. (2002) ; Lange et al. (2004)

(d) $K = 5$ (ARI = 0.65)

Figure 5.7 : Partitions found by K-means on the 4clusters_corner data set for $K \in \{2, \ldots, 5\}$.

Tous les indices internes, qui seront présentés dans la prochaine Section 5.6, à l'exception du Gap Statistic, choisissent $K = 2$. Les méthodes de stabilité basées sur l'échantillonnage (Ben-Hur et al., 2002 ; Lange et al., 2004) ont sélectionné le vrai nombre de clusters $K^\star = 4$, bien qu'il s'agisse de la pire partition parmi les quatre ($K = 2, 3, 4, 5$) au sens de l'ARI. Enfin, le Stadion sélectionne la solution $K = 3$ ayant l'ARI le plus élevé. De plus, le critère produit des solutions dans le même ordre que l'ARI. Cet exemple montre clairement le compromis de stabilité qui se produit dans le Stadion : il essaie de préserver une stabilité élevée entre les clusters tout en gardant la stabilité à l'intérieur du cluster aussi faible que possible.

5.5.4 Trouver $K = 1$: le cas d'un ensemble de données sans structure de clusters

Terminons cette section illustrative avec une question : Un ensemble de données a-t-il une structure en clusters ? La stabilité entre clusters ne permet pas de répondre à cette question, car la solution avec un seul cluster ($K = 1$) est trivialement stable. Certaines méthodes de stabilité ne sont même pas définies pour $K = 1$ à cause de la normalisation (Lange et al., 2004). De plus, de nombreux indices internes utilisent la distance entre clusters et ne sont pas non plus définis pour un seul cluster. Nous avons vérifié empiriquement que notre critère donne systématiquement $K = 1$ dans le cas où l'algorithme ne trouve aucune structure de cluster. La Table 5.1 contient les résultats pour des ensembles de données simulées sans structure. Stadion trouve constamment $K = 1$ dans tous les cas de figure.

Table 5.1 : Stadion dans le cas où les ensembles de données n'ont pas de structure.

Données	n	p	Stadion
Uniform (2d)	1000	2	1
Uniform (10d)	1000	10	1
Gaussian (2d)	1000	2	1
Gaussian (10d)	1000	10	1
Golfball (Ultsch, 2005)	4002	3	1

5.6 Expériences : *benchmark*

Comme nous l'avons déjà expliqué dans l'introduction Section 1.4, les méthodes de clustering et les méthodes de sélection de modèles pour le clustering doivent être évaluées uniquement sur des ensembles de données simulées. Les ensembles de données simulées sont les seuls à garantir une évaluation cohérente des méthodes. Pour comparer les différentes méthodes d'évaluation, nous utiliserons l'ARI entre la partition choisie et la *vraie* partition.

Dans les simulations, trois algorithmes sont considérés : les K-means, les GMM et le clustering hiérarchique de Ward. Pour les K-means, deux versions de Stadion sont évaluées : la première utilise le calcul de stabilité décrit dans la Section 5.4 (appelée version standard), et la seconde utilise l'opérateur d'extension (appelée version *extended*). Comme nous l'avons vu dans la Section 5.3, un opérateur d'extension étend un clustering à de nouvelles données. L'algorithme des K-means a un opérateur d'extension car il peut affecter des nouvelles données au centre le plus proche. Ainsi, au lieu d'exécuter les K-means pour chaque perturbation des données, nous prédisons directement les affectations des clusters des points de données perturbées. Cette approximation a du sens dans notre cas puisque nous considérons le *jittering* comme la principale source d'instabilité et par conséquent nous sommes uniquement intéressés par la densité aux frontières. Or perturber les individus et observer s'ils ont changé de clusters, sans relancer l'algorithme, en calculant leurs distances aux centres permet de quantifier cette information aux frontières et cela permet aussi d'économiser du temps de calcul. Les GMM ont une extension similaire, en assignant des nouvelles données au cluster pour lequel la probabilité a posteriori est la plus élevée. C'est la seule version que nous considérons ici en raison du coût de calcul élevé des GMM qui rend la version standard trop coûteuse en temps de calcul au vu du nombre de simulations nécessaires. Bien que les premières expériences aient semblé prometteuses, la même limitation a été rencontrée pour le clustering spectral (Von Luxburg, 2007) et malheureusement il n'y a pas d'opérateur d'extension direct pour cet algorithme.

Nous évaluons les méthodes de validation du clustering sur une large collection de 73 ensembles de données artificielles, la plupart d'entre eux étant largement utilisés dans la littérature. Les ensembles de données ont été sélectionnés de manière à ce que les algorithmes puissent obtenir de bonnes partitions proche des *vrais* clusters. Nous avons également considéré différents niveaux de difficulté pour la sélection des modèles, obtenus en faisant varier le nombre de clusters, leurs tailles, leurs variances, leurs formes et la présence de bruit et de clusters proches.

La Table 5.2 résume les résultats des trois algorithmes. Nous comparons Stadion aux partitions obtenues avec le *vrai* nombre de clusters $K^\star$ pour différent indices d'évaluation de clustering (voir Desgraupes (2013) ; Hämäläinen et al. (2017)), le Gap Statistic (Tibshirani et al., 2001a) (K-means seulement), BIC (GMM seulement) et des méthodes d'évaluation de clustering à l'aide de la stabilité (Ben-Hur et al., 2002 ; Lange et al., 2004). Pour garantir une comparaison équitable, tous les indices internes ont été calculés sur la même partition, qui était également la partition de référence dans Stadion. Nous donnons pour chaque méthode le nombre de fois où $K^\star$ a été trouvé, que nous appelons le nombre de wins, ainsi que l'erreur quadratique moyenne (EQM) entre les scores ARI correspondant à K_M et à $K^\star$.

Nous incluons également les algorithmes basés sur les K-means qui sont conçus pour trouver automatiquement le paramètre K : X-means Pelleg and Moore (2000), G-means Hamerly and Elkan (2004) et SpecialK Hess and Duivesteijn (2019). Pour SpecialK, nous avons utilisé les hyperparamètres par défaut indiqués par les auteurs, mais les hypothèses faites par la méthode sont trop restrictives et la méthode échoue complètement sur 11 ensembles de données, ce qui explique les mauvais résultats. En outre, le Gap Statistic Tibshirani et al. (2001a) a été testé avec deux ensembles différents d'hyperparamètres. La fonction que nous avons utilisée est implémentée dans un package R Maechler et al. (2013), où la version (A) est la version originale proposée par les auteurs et où la version (B) utilise une distribution de référence uniforme. Pour la procédure de Hennig, le package R Hennig and Imports (2015) de l'auteur a été utilisé. Il permet d'évaluer la stabilité d'un clustering de données, lorsque les données sont ré-échantillonnées à l'aide de la méthode *bootstrap* et que la similarité de Jaccard est utilisée entre les clusters avant perturbation et les clusters après perturbations. La méthode

Table 5.2 : Résultats du benchmark sur 73 ensembles de données artificielles pour K-means, Ward et GMM. Rang moyen de l'ARI avec les *vraies* classes ($\overline{R_{\text{ARI}}}$) et nombre de fois où $K^\star$ a été sélectionné (wins).

Méthodes	K-means		Ward		GMM	
	$\overline{R_{\text{ARI}}}$	wins	$\overline{R_{\text{ARI}}}$	wins	$\overline{R_{\text{ARI}}}$	wins
$K^\star$	8.11	73	4.77	73	5.05	73
Stadion-max	**7.46**	50	**5.25**	**54**	-	-
Stadion-mean	7.70	51	5.80	49	-	-
Stadion-max (extended)	7.58	**56**	-	-	**5.59**	**56**
Stadion-mean (extended)	8.09	48	-	-	6.79	43
BIC	-	-	-	-	6.45	48
WG (Desgraupes, 2013)	8.33	53	5.40	**54**	5.77	52
Silhouette (Rousseeuw, 1987)	9.55	46	6.47	45	7.01	45
Lange (Lange et al., 2004)	10.18	45	6.53	51	6.99	48
Davies-Bouldin (Davies and Bouldin, 1979)	10.21	40	6.45	41	7.29	34
Ray-Turi (Ray and Turi, 1999)	10.28	37	6.97	40	7.68	33
Hennig Hennig (2007, 2008)	10.72	37	-	-	-	-
CH (Caliński and Harabasz, 1974)	11.44	41	7.14	39	7.43	37
Gap statistic (B) Tibshirani et al. (2001a)	11.49	29	-	-	-	-
X-means Pelleg and Moore (2000)	11.56	28	-	-	-	-
Dunn (Dunn, 1974)	13.09	26	7.77	33	7.92	34
Hofmeyr Hofmeyr (2018)	13.20	30	-	-	-	-
Xie-Beni (Xie and Beni, 1991)	13.30	22	7.61	34	8.19	28
Gap statistic (A) (Tibshirani et al., 2001a)	13.57	26	-	-	-	-
G-means Hamerly and Elkan (2004)	13.74	24	-	-	-	-
Ben-Hur (Ben-Hur et al., 2002)	14.34	20	7.86	31	8.85	28
SpecialK Hess and Duivesteijn (2019)	17.07	19	-	-	-	-

de Hofmeyr Hofmeyr (2018), correspond à une nouvelle façon d'évaluer les degrés de liberté dans K-means en utilisant le BIC qui tient compte de l'incertitude des affectations des classes dans les degrés de liberté. Malheureusement, d'autres méthodes comme Dip-means Kalogeratos and Likas (2012) ou Skinny-dip Maurus and Plant (2016) n'ont pas d'implémentations disponibles facilement utilisables et celles-ci n'ont donc pas pu être incluses. Tous les autres indices utilisés sont implémentés et décrits dans le package `R` Desgraupes (2013).

Dans toutes nos simulations, nous utilisons un ε-AP gaussien, $D = 10$, $\mathbf{\Omega} = \{2, \ldots, 10\}$, $s = \text{ARI}$ et nous évaluons les solutions pour $K \in \{1, \ldots, K_{\max}\}$ où $K_{\max}$ est égal à $K^\star + 20$ arrondis à la dizaine la plus proche.

Nous pouvons voir dans la Table 5.2, Stadion-max obtient les meilleurs résultats en général. Avec l'algorithme des K-means, il est même mieux classé que $K^\star$ en termes d'ARI. Cela s'explique par le fait que les K-means en $K^\star$ ne sont pas toujours la meilleure solution. En effet, un ensemble de donnée peut être simulé avec 3 classes de sorte que l'algorithme des 2-means obtient un résultat convenable, alors que celui des 3-means obtient de très mauvais résultats. Le deuxième indice le plus performant est celui de Wemmert-Gancarski (WG). C'est un indice très peu présent dans la littérature et peu utilisé en pratique. En toute honnêteté cet algorithme présente des avantages par rapport à notre solution : il est un peu plus rapide, même s'il a une complexité quadratique, il fonctionne très bien sur l'ensemble des simulations et il peut être utilisé sur les mêmes familles d'algorithmes que celles pour lesquelles Stadion est défini. En revanche, cet algorithme se base sur la distance euclidienne et est donc moins général en ce sens.

Stadion a obtenu des résultats inférieurs avec l'algorithme Ward. Il a été montré dans Balcan and Liang (2016) que le clustering agglomératif n'est pas robuste au bruit, ce qui explique les résultats. Par ailleurs, il faut noter que les résultats sont légèrement biaisés en faveur de tous les indices sauf Stadion, qui ne sont valables que pour $K \geq 2$, contrairement à Stadion qui choisira $K = 1$ lorsque l'algorithme de clustering ne trouve aucune solution acceptable pour $K \geq 2$.

5.7 Conclusion

5.7.1 Résumé

La stabilité est un outil très général pour évaluer la qualité des solutions obtenues par des algorithmes d'apprentissage supervisé et non supervisé. Dans cet article, nous présentons le concept de stabilité d'un clustering et les travaux connexes, avec leurs limites. Un inconvénient majeur est que lorsqu'il existe un seul minimum global de la fonction objectif, les solutions avec K trop petit sont toujours stables. Nous

démontrons empiriquement dans ce travail que même lorsque K est trop grand, l'estimation de la stabilité des clusters par échantillonnage n'est pas appropriée, puisqu'elle peut malgré tout conclure à la stabilité. Ainsi, notre contribution se résume à deux améliorations. Tout d'abord, la stabilité peut être bien estimée par une perturbation additive de ε, ce qui répond à la limitation du sous-échantillonnage en pratique. Ce principe est né du fait que la perturbation est cruciale pour mesurer les densités aux frontières des clusters et évaluer la stabilité d'une solution, dans le cadre réaliste que nous avons décrit. Deuxièmement, nous avons introduit le concept de stabilité intra-classes pour détecter les structures au sein des clusters, et Stadion (critère de différence de stabilité), un nouveau critère de validation du clustering agissant comme un compromis entre la stabilité traditionnelle et la stabilité intra-classes.

En outre, la méthode de contrôle de la quantité de perturbation que nous avons définie fournit un outil de visualisation interprétable appelé chemins de stabilité. C'est un avantage de la méthode car les indices internes de validation d'un clustering donnent uniquement un score à chaque solution. Les chemins de stabilité permettent de mieux comprendre la structure des données, la présence de sous clusters ou le fait qu'une solution mène à la séparation d'un cluster en deux.

Dans l'ensemble, la sélection de modèle reste un défi dans le clustering et il n'y a pas encore de théorie ou de méthodologie qui puisse remplir cette tâche correctement. Nous avons proposé une méthode empirique qui montre des résultats intéressants, ainsi que des indications sur un contexte théorique qui pourrait être établi dans des travaux futurs.

5.7.2 Remarques importantes

Pour conclure, nous présentons, en toute honnêteté, quelques remarques à propos de notre méthode.

Caractère non universel de la stabilité La stabilité d'un clustering se base sur des principes généraux. C'est un des seuls concepts utilisés en sélection de modèle ayant des fondements théorique en clustering. Il est donc intéressant de construire des outils d'évaluation inspirés ce principe. Malheureusement, nous nous sommes aperçus que certaines de ses caractéristiques contredisent son caractère universel :

1. premièrement, dans ce travail nous avons montré que les méthodes de mesure de stabilité basées sur l'échantillonnage ne pouvaient pas être efficaces, contrairement au cas de l'apprentissage supervisé et c'est une des raisons pour laquelle la stabilité perd son caractère universel. En effet, l'alternative est d'injecter du bruit, mais la quantité de bruit qui est nécessaire pour évaluer la stabilité dépend du sous-espace dans lequel les clusters résident. Par exemple, si $p = 1000$ le bruit nécessaire pour créer de l'instabilité est beaucoup plus grand que si $p = 2$. Cela pose problème notamment si l'on cherche à comparer des méthodes dans des sous-espaces différents et c'est peut-être une des raisons pour laquelle nous n'avons pas réussi à étendre Stadion au choix du paramètre λ pour le WT-K-means par exemple.

2. deuxièmement, la stabilité dépend nécessairement de l'algorithme de clustering utilisé. Nous nous sommes rendus compte qu'il était difficile voire impossible de comparer des chemins de stabilité, intra et inter-classes, entre des algorithmes différents (par exemple entre l'algorithme des K-means et la CAH). Dans l'idéal on ne voudrait pas simplement choisir des paramètres pour un algorithme, mais déterminer le meilleur modèle, la meilleure partition au sens d'un critère se basant sur une définition précise.

3. enfin, la troisième raison pour laquelle la stabilité perd son caractère universel a déjà en partie été discutée durant ce chapitre. Pour créer de l'instabilité, il faut que les frontières soient placées dans des zones de forte densité. Or, certains algorithmes, tel que la CAH avec single linkage, n'ont pas cette propriété, puisque cet algorithme peut créer une partition à 2 clusters en isolant simplement une observation aberrante.

C'est en partie pour ces raisons qu'une étude théorique n'a pas été menée. En effet, telle quelle, la stabilité ne semble pas offrir le cadre nécessaire à l'établissement de fondements théoriques pour la sélection de modèle en clustering. Il reste que cette méthode peut représenter une aide à la détermination du meilleur choix pour le nombre de clusters à considérer.

Temps de calcul de Stadion La procédure Stadion est assez lente comparativement à d'autres méthodes. C'est peut-être même la méthode la plus lente, dans nos simulations, par rapport à celles qui sont comparées dans ce chapitre mis à part le Gap statistic. Stadion est très coûteux en temps de calcul, surtout si le nombre de clusters K testé est grand. Or, dans nos simulations nous avons souvent testé des valeurs de 20 à 40. Néanmoins, de nombreuses méthodes ont une complexité quadratique par rapport au nombre d'observations n et dans nos simulations n était relativement petit ($n < 10^5$). Stadion lui n'a pas ce problème puisque

sa complexité est celle de l'algorithme testé par rapport au nombre d'observations. De plus, avec la version *extended* et en prenant peu de valeurs dans $\mathbf{\Omega}$, on obtient une version beaucoup plus rapide et tout aussi efficace. Malgré tout, une préoccupation importante est celle d'accélérer le calcul de Stadion. Nous pensons que la mesure de la densité des données aux frontières des clusters est suffisante pour évaluer la stabilité, elle pourrait donc être estimée uniquement en calculant les distances des points de données aux frontières lorsque celles-ci sont connues, par exemple dans les méthodes basées sur les centres, comme les K-means.

UNE VERSION LONGUE DE CE TRAVAIL Une version plus longue de ce travail a été produite (Mourer et al., 2020a) et de nombreuses analyses supplémentaires ont été effectuées :

- une analyse sur l'influence des hyperparamètres. L'influence des paramètres $\mathbf{\Omega}$, de D le nombre d'ensemble de données perturbées et de la définition de la fonction s qui mesure la similarité de deux partition a été étudiée. Nous avons conclu que le choix de s influence très peu les résultats, que Stadion peut avoir des résultats corrects avec D petit, de l'ordre de 5, et que $\mathbf{\Omega} = \{2, 3, 4\}$ était une solution qui offrait des résultats similaires à ceux obtenus avec un ensemble $\mathbf{\Omega} = \{2, 3, 4\}$ plus grand. L'analyse d'influence a été menée à l'aide d'une fANOVA (*functional analysis of variance*) Hutter et al. (2014).
- une analyse plus détaillée des résultats de simulations, avec toutes les tables de résultats par ensemble de données en fonction de chaque algorithme. Des résultats détaillés des tests de Wilcoxon-Holms et les diagrammes critiques pour les tests de rang sont aussi données. De plus, la liste de tous les ensembles de données utilisés et leurs références dans la littérature sont présentées dans la dernière partie de l'article.

6

Une nouvelle mesure d'importance de variables basée sur le clustering de variables pour les forêts aléatoires

6.1 Introduction . . . 85
6.2 Définitions, état de l'art et analyses . . . 86
6.2.1 Forêts aléatoires 86
6.2.2 MDA : Importance par permutation 87
6.2.3 MDI : Importance de Gini 87
6.2.4 Inconvénients des mesures de Breiman 88
6.2.5 Mesures alternatives 89
6.2.6 Clustering de variables 90
6.3 Comparaison des méthodes existantes à l'aide d'un exemple simulé . . . 90
6.4 Méthodologie de la solution proposée . . . 92
6.4.1 Clustering de variables et variables synthétiques 92
6.4.2 Le choix du nombre de clusters 92
6.4.3 Nouvelle mesure de l'importance des variables : Synthetic-MDA (SMDA) 94
6.5 Simulations . . . 95
6.6 Application industrielle : flottement *fan* . . . 97
6.6.1 Introduction 97
6.6.2 Analyse de l'impact du jeu radial en bout d'aube sur la marge au flottement 98
6.7 Conclusion . . . 100

6.1 Introduction

CONTEXTE Ce chapitre aborde la question de l'interprétabilité d'un modèle et du calcul de l'importance des variables dans le contexte de l'apprentissage supervisé et notamment de la régression pour des modèles de forêts aléatoires. Plus précisément, il s'agit de prendre en compte les corrélations des variables d'entrée numériques, qui peuvent dégrader les critères d'importance des variables basés sur des permutations aléatoires des observations (voir par exemple Bénard et al. (2021)) et conduire ainsi à un manque de fiabilité dans le classement des variables selon leur l'importance. Une solution est alors proposée et appliquée à un cas d'étude sur des données réelles décrivant les phénomènes de vibration des aubes *fan*.

LES OBJECTIFS DE L'IMPORTANCE DE VARIABLES Comme on l'a précisé dans l'introduction de la thèse, on sait que l'interprétation d'un modèle au moyen des méthodes d'importance des variables est une tâche utile qui aide tant à la prise de décision que pour l'analyse de données. Par ailleurs, les forêts aléatoires constituent une technique attrayante en apprentissage supervisé en raison de leurs bonnes performances empiriques, mais

elles sont souvent considérées comme des boîtes noires en raison de leur manque d'interprétabilité. Pour les forêts aléatoires, l'interprétabilité peut être évaluée en quantifiant l'importance des variables. On rappelle et on précise, que d'un point de vue statistique, le calcul de l'importance des variables en apprentissage supervisé vise à satisfaire deux objectifs :

i) estimer la contribution de chaque variable d'entrée à la prédiction ;

ii) mesurer la dépendance entre les variables d'entrée et la variable de sortie.

Cependant, il faut noter que ces objectifs peuvent être contradictoires lorsque les variables d'entrée sont corrélées. En effet, si deux variables d'entrée sont fortement corrélées, l'une d'entre elles peut être écartée sans dégrader les performances du modèle, tout en restant liée à la variable de sortie. Autrement dit, la variable écartée aura une importance estimée nulle, car elle ne contribue pas à améliorer la prédiction alors qu'elle reste dépendante de la variable de sortie. Cette contradiction est importante lorsque par exemple, l'on s'intéresse à expliquer un phénomène physique quantifié par une variable dite à expliquer à l'aide d'un ensemble de variables dites explicatives : dans ce cas le but n'est pas de maximiser la prédiction mais de modéliser au mieux les relations entre la variable à expliquer et les variables explicatives. Toutes les méthodes de calcul d'importance de variables dans les forêts aléatoires dont nous avons connaissance sont conçues pour satisfaire l'objectif i) et ne visent qu'à évaluer la contribution des variables d'entrée à la prédiction a posteriori.

CONTRIBUTIONS ET ORGANISATION DE CE CHAPITRE Ce chapitre est une version étendue d'une précédente étude (Chavent et al., 2021b) et nous nous concentrons principalement sur l'objectif ii) ci-dessus, c'est-à-dire sur la découverte de relations entre les variables d'entrée et de sortie. La performance du modèle, bien qu'essentielle, n'est pas notre préoccupation première et nous nous intéressons surtout à développer un outil d'aide à la compréhension et à la décision (objectif ii)). Nous avons identifié quatre biais possibles dus à la présence de variables corrélées : le biais de sélection (variables interchangeables), le biais de préférence (préférence pour les variables corrélées), le biais de masquage (tirage aléatoire des variables) et le biais de permutation (extrapolation du modèle sur une nouvelle distribution). Ces clarifications ont permis de mettre en lumière qu'aucune des méthodes existantes ne permettait de supprimer tous ces biais et d'atteindre l'objectif ii). Ainsi, nous introduisons un nouveau critère pour évaluer l'importance des variables dans les forêts aléatoires, basé sur l'importance par permutation (Breiman, 2001) et calculé sur un ensemble de variables synthétiques. Les variables synthétiques sont définies comme une réduction des variables d'entrée, en utilisant leur structure de corrélations et une procédure de clustering de variables, comme proposé par Chavent et al. (2021a). Le reste du chapitre est organisé comme suit : Après avoir passé en revue l'état de l'art et avoir donné un exemple introductif, la Section 6.4 contient les détails de la méthodologie proposée et énumère ses principales étapes, tandis que la Section 6.5 illustre le critère proposé et le compare à la littérature existante, pour un ensemble de données simulées, enfin une application de la méthode sur des données industrielles est présentée dans la Section 6.6.

6.2 Définitions, état de l'art et analyses

6.2.1 Forêts aléatoires

Les forêts aléatoires, inventées par Léo Breiman au début des années 2000 (Breiman, 2001) font partie des algorithmes qui restent efficaces (tant d'un point de vue computationnel que prédictif) lorsqu'ils sont appliqués à des ensembles de données de grande dimension. Leur construction repose sur les travaux fondateurs de Amit and Geman (1997) ; Dietterich (2000) et s'appuie sur le principe de diviser pour régner : la forêt est composée de plusieurs arbres (arbres CART de Breiman modifiés aussi appelés arbres CART *random inputs* (Breiman, 2001 ; Genuer and Poggi, 2017)) qui sont chacun construits avec un sous-ensemble des observations et des variables. La prédiction de la forêt est alors obtenue simplement en agrégeant les prédictions des arbres.

Rappelons que nous nous plaçons dans le cadre de modèles de régression à variables d'entrée numériques. Une forêt aléatoire prend en entrée, une matrice de données $\mathbf{X} \in \mathbb{R}^{n \times p}$ constituée de n observations $\mathbf{x}_i$ décrites par p variables $\mathbf{x}^j$, pour prédire un vecteur cible $\mathbf{y} \in \mathbb{R}^n$. On note $\{(\mathbf{x}_i, y_i)\}_{1 \leq i \leq n}$ l'ensemble de données, et $F_t(\mathbf{x}_i)$ (respectivement $F(\mathbf{x}_i)$) la sortie associée à l'observation $\mathbf{x}_i$ calculée par l'arbre t (respectivement par la forêt aléatoire). On ne rappelle pas ici la définition complète des forêts aléatoires, mais il est utile pour la suite d'expliciter les deux mécanismes de randomisation qui sont au cœur de l'algorithme. Soit T le nombre d'arbres,

- on répète les opérations suivantes pour chacun des T arbres :

  - on tire n observations dans l'ensemble des données par *bootstrap* (tirage avec remise) (Tibshirani and Efron, 1993 ; Efron, 2003) que l'on appelle échantillon *in-the-bag* pour le différencier de l'échantillon *out-of-bag* (OOB) qui est l'ensemble des observations qui ne sont pas utilisées pour l'apprentissage. Notons que l'ensemble OOB dépend de l'arbre t et pourra être noté OOB_t
  - on modélise un arbre CART modifié sur cet échantillon bootstrap. Il est modifié car à chaque nœud de l'arbre, on tire aléatoirement sans remise un certain nombre de variables (inférieur ou égal à p) et on choisit celle qui conduit à la plus grande réduction de l'erreur (variance).

- on moyenne les résultats sur les T arbres construits, c'est-à-dire que la valeur de sortie $F(\mathbf{x}_i)$ associée à chaque observation i est la moyenne arithmétique des T valeurs obtenues.

Pour chaque arbre t de la forêt, l'erreur OOB est donnée par :

$$\mathcal{E}(\text{OOB}_t, t) = \frac{1}{|\text{OOB}_t|} \sum_{\mathbf{x}_i \in \text{OOB}_t} \Big(y_i - F_t(\mathbf{x}_i)\Big)^2,$$

et l'erreur OOB de la forêt aléatoire est la moyenne arithmétique des erreurs des arbres qui la composent.

Les forêts aléatoires sont par nature des boîtes noires difficilement interprétables. En effet, les contributions de chacune des variables au modèle sont inconnues et il est impossible d'analyser la structure de milliers d'arbres à l'œil nu. Il est donc nécessaire d'utiliser des méthodes post-hoc pour les déterminer. Une mesure naïve d'importance de variables dans les forêts aléatoires consiste à compter le nombre de fois où chaque variable apparaît dans l'ensemble des arbres, mais cette mesure s'avère biaisée (Strobl et al., 2007). L'explication que nous donnons est la suivante : dans les forêts aléatoires, les arbres de régression ne sont pas *élagués* (cinq observations par feuille par défaut) et le peu d'observations disponibles proches des feuilles (nœuds terminaux) de l'arbre fait que les dernières coupures sont souvent aléatoires, tant pour le choix de la variable que pour le choix de la valeur de la coupure (Li et al., 2019 ; Duroux and Scornet, 2018), biaisant ainsi le comptage sur l'arbre. D'autres mesures plus complexes doivent alors être employées et historiquement deux ont été très largement utilisées, l'importance par permutation et l'importance de Gini introduites par Breiman (2001).

6.2.2 MDA : Importance par permutation

L'importance par permutation, ou *Mean Decrease Accuracy* (MDA), est une mesure qui a été définie en remarquant que si une variable n'est pas importante, alors la permutation de ses valeurs ne doit pas dégrader la qualité de la prédiction. La permutation rompt la relation d'une variable d'entrée avec la variable de sortie si elle existe et l'évolution de l'erreur de prédiction permet en principe de quantifier la contribution estimée de cette variable au modèle. Dans une forêt aléatoire, l'importance par permutation est calculée avec des permutations différentes par variable et par arbre.

Formellement, soit OOB_t^j l'échantillon OOB du t-ième arbre où la j-ième variable a subi une permutation de ses valeurs (propre à chaque arbre). Alors la MDA de la variable $\mathbf{x}^j$ est définie par :

$$\text{MDA}(\mathbf{x}^j) = \frac{1}{T} \sum_{t=1}^{T} \Big[\mathcal{E}(\text{OOB}_t^j, t) - \mathcal{E}(\text{OOB}_t, t)\Big]. \tag{6.1}$$

Une variable x^j est donc d'autant plus importante que la quantité $\text{MDA}(\mathbf{x}^j)$ est grande.

6.2.3 MDI : Importance de Gini

La deuxième proposition de Leo Breimann est la *Mean Deacrease Impurity* (MDI). Dans un arbre de régression le critère à optimiser est la réduction de l'erreur (variance) lorsque l'on divise un nœud en deux, ce qu'on appelle le gain. Le calcul de la MDI consiste généralise le calcul de l'importance de Gini au cadre de la régression, en additionnant le gain associé à toutes les coupures effectuées sur une variable donnée. Formellement, le gain associé à la variable $\mathbf{x}^j$ pour un nœud (cellule) A dans un arbre t est défini comme :

$$G(\mathbf{x}^j, A, t) = \sum_{i:\mathbf{x}_i \in A} \frac{1}{n_A}(y_i - \bar{\mathbf{y}}_A)^2 - \Big(\sum_{i:\mathbf{x}_i \in A_L} \frac{n_{A_L}}{n}(y_i - \bar{\mathbf{y}}_{A_L})^2 + \sum_{i:\mathbf{x}_i \in A_R} \frac{n_{A_R}}{n}(y_i - \bar{\mathbf{y}}_{A_R})^2 \Big),$$

où

- $A_L = \{\mathbf{x} \in A, \mathbf{x}^j \leqslant z\}$ est l'ensemble des observations appartenant à A qui ont une valeur inférieure ou égale à la valeur de la coupure z pour la variable j ;

- $A_R = \{\mathbf{x} \in A, \mathbf{x}^j > z\}$ est l'ensemble des observations appartenant à A qui ont une valeur supérieure à la valeur de la coupure z pour la variable j ;
- n_A, n_{A_L}, n_{A_R} sont respectivement les nombres d'observations dans les nœuds A, A_L, A_R ;
- $\bar{\mathbf{y}}_A, \bar{\mathbf{y}}_{A_L}, \bar{\mathbf{y}}_{A_R}$ sont respectivement les moyennes des observations dans le nœud A, A_L, A_R.

Ainsi, la MDI est défini comme :

$$\text{MDI}(\mathbf{x}^j) = \sum_{t=1}^{T} \sum_{A \in \mathcal{A}_t^j} G(\mathbf{x}^j, A, t), \tag{6.2}$$

où $\mathcal{A}_t^j$ est l'ensemble des nœuds dont la coupure dépend de la variable j pour l'arbre t. Étant donné que, dans une forêt aléatoire, les coupures maximisent la réduction de l'erreur, la MDI semble être un bon choix pour mesurer l'importance des variables.

Malheureusement la MDI et la MDA présentent de nombreux désavantages, notamment lorsque les variables d'entrée sont corrélées.

6.2.4 Inconvénients des mesures de Breiman

Les deux analyses théoriques suivantes Bénard et al. (2021) ; Scornet (2020) montrent respectivement que la MDA et la MDI sont bien définies lorsque les variables d'entrées sont non corrélées. En revanche, ces études ainsi que de nombreuses analyses empiriques montrent que lorsque les variables sont corrélées, la MDA et la MDI sont biaisées (Park et al., 2007 ; Archer and Kimes, 2008 ; Strobl et al., 2008 ; Nicodemus and Malley, 2009 ; Genuer et al., 2010 ; Auret and Aldrich, 2011 ; Tolo̧si and Lengauer, 2011 ; Nicodemus, 2011 ; Boulesteix et al., 2012 ; Gregorutti et al., 2017 ; Hooker and Mentch, 2019 ; Mentch and Zhou, 2020 ; Hooker et al., 2021).

Il y a beaucoup de confusion dans la littérature sur la nature du biais. Dans ce chapitre, nous distinguons quatre biais très différents que nous allons présenter dans les paragraphes ci-dessous.

LE BIAIS DE SÉLECTION Le biais de sélection provoque une diminution de l'importance des variables corrélées. En effet, lorsque deux variables discriminantes sont très corrélées, les arbres CART, dans un nœud donné, doivent choisir une des deux variables corrélées. L'autre variable n'intervenant pas dans ce nœud, elle aura un gain de 0 (MDI) et ne permutera pas d'observations (MDA) pour ce nœud. Sur la construction d'une forêt disposant de nombreux arbres, on peut espérer que les deux variables très corrélées aient la même importance, mais celle-ci sera réduite par rapport à l'importance obtenue avec l'inclusion d'une seule de ces deux variables. Cet effet est proportionnel à la force de la corrélation et à la taille du groupe des variables corrélées.

Le biais de sélection a été étudié empiriquement, en observant l'impact de variables corrélées sur l'importance des variables estimée par la MDA et la MDI. Ce comportement a d'abord été mis en évidence par Strobl et al. (2007), qui expliquent que les variables corrélées sont utilisées de manière interchangeable dans les arbres de décision des modèles de forêt aléatoires. Genuer et al. (2010) montrent à l'aide de simulations que plus on ajoute des variables corrélées plus la MDA des variables corrélées diminue. Des articles plus récents confirment ce comportement : Tolo̧si and Lengauer (2011) ; Gregorutti et al. (2017) indiquent tout d'abord que l'importance des variables corrélées les plus discriminantes ne sont pas nécessairement plus élevées que celles des variables moins discriminantes, et deuxièmement, que les importances estimées par MDA dépendent de la taille des groupes de variables corrélées.

LE BIAIS DE PRÉFÉRENCE Ce biais est observé lorsqu'il existe des corrélations entre variables d'entrée et il implique que les variables corrélées discriminantes ont une probabilité plus élevée d'être sélectionnées à chaque nœud de l'arbre. Le biais de préférence a été étudié d'abord empiriquement par Strobl et al. (2007) qui indiquent que la MDA surestime l'importance des variables discriminantes corrélées. Dans leur analyse empirique les variables discriminantes corrélées sont préférées dans les premières coupures. Récemment, une analyse théorique confirme cette assertion, Scornet (2020) montre pour la MDI dans le cas d'un modèle linéaire, qu'une corrélation positive entre deux variables discriminantes augmente la probabilité de diviser selon l'une de ces deux variables et donc que les arbres CART tendent à favoriser les variables positivement corrélées.

Le biais de préférence peut aussi se comprendre en étudiant l'importance du groupe de variables corrélées. Théoriquement, selon Scornet (2020), on s'attend à ce que la somme des contributions (à la prédiction) des variables soit égale à la contribution (à la prédiction) de l'ensemble des variables (la performance du modèle), comme c'est le cas lorsque les variables sont non corrélées. Or ce n'est pas le cas ici, car la contribution de l'ensemble des variables est plus grande que la somme des contributions. Ainsi dans un arbre CART de régression, l'inclusion de deux variables importantes (discriminantes) identiques entraîne individuellement une

diminution de l'importance de chacune d'entre elles (biais de sélection), mais une augmentation de l'importance du groupe formé par ces deux variables, par rapport à l'importance obtenue avec un arbre n'incluant qu'une seule de ces deux variables (biais de préférence). Il faut noter que l'effet du biais de préférence est inversé lorsque les variables sont corrélées négativement, contrairement à celui du biais de sélection.

LE BIAIS D'ÉCHANTILLONNAGE Le biais d'échantillonnage ne concerne que la MDA car par définition la MDI ne se base pas sur du rééchantillonnage des observations. Des travaux ont pointé le fait que l'importance des variables dans les forêts aléatoires pouvait augmenter en présence de variables corrélées. Strobl et al. (2008) indiquent que le schéma de permutation employé ne prend pas en compte les corrélations dans le calcul de la mesure d'importance de la variable. Les causes de cet effet ont été explicitées en détail dans les articles de Hooker and Mentch (2019) ; Hooker et al. (2021) qui montrent que la permutation d'une variable change la distribution des données dans le cas de données corrélées, ce qui force le modèle à extrapoler dans des régions de faible densité.

De manière théorique, cette question a été abordée dans Bénard et al. (2021) qui décomposent la MDA en trois termes, dont un correspond au biais d'échantillonnage et voit sa valeur augmenter en présence de variables corrélées. Notamment, ils ajoutent que ce terme a une valeur qui augmente avec la corrélation des variables. Ainsi, le biais d'échantillonnage provoque une augmentation de l'importance des variables corrélées, comme le biais de préférence mais via un autre mécanisme.

LE BIAIS DE MASQUAGE À proprement parler, ce biais n'est pas causé par les corrélations entre les variables d'entrée mais il est peut-être présent en grande dimension, notamment dans le cas $n \ll p$. En effet, nous rappelons que lors de la construction des arbres, à chaque nœud un sous-ensemble de variables est tiré aléatoirement. Or, en grande dimension, il est possible que des variables importantes ne soient pas tirées et donc qu'elles n'interviennent pas dans le modèle, ce qui est d'autant plus vrai lorsque l'on a peu d'observations, car il y a moins de coupures. C'est ce que l'on appelle le biais de masquage et il est dû à la procédure de randomisation appliquée à chaque nœud de l'arbre dans la forêt. Il est donc lié à la construction de la forêt aléatoire, contrairement aux biais de sélection et de préférence qui sont liés à la construction des arbres.

En pratique, on espère que dans une forêt avec de nombreux arbres profonds, toutes les variables ont été tirées et dans le cas où elles contribuent au modèle, sélectionnées.

6.2.5 Mesures alternatives

Plusieurs mesures d'importance de variables pour les forêts aléatoires ont été développées pour corriger les biais présentés ci-dessus. Les solutions proposées ne tentent en général de résoudre qu'une partie des problèmes et de fait sont encore biaisées.

MDI ET *KNOCKOFFS* Dans leur analyse théorique, Sandri and Zuccolotto (2008) ont décomposé la MDI en deux parties : le gain (explicité par l'Équation (6.2)) et un biais positif. Pour supprimer le terme de biais, l'ensemble de données $\mathbf{X}$ est augmenté de pseudo-variables, qui ne sont pas informatives mais qui partagent la même structure que $\mathbf{X}$. Pour créer ces pseudo-variables $\tilde{\mathbf{X}}$, toutes les variables de $\mathbf{X}$ sont permutées de manière identique. Algorithmiquement, la méthode fonctionne de la façon suivante :

- soit π une permutation des observations de $\mathbf{X}$ en une matrice $\tilde{\mathbf{X}}$ telle que $\forall i = 1, ..., n$, $\mathbf{x}_i = \tilde{\mathbf{x}}_{\pi(i)}$;
- $\tilde{\mathbf{X}}$ a alors la même distribution que $\mathbf{X}$ mais est indépendante de $\mathbf{X}$ et de $\mathbf{y}$;
- la forêt aléatoire est entraînée avec les variables d'entrée $[\mathbf{X}, \tilde{\mathbf{X}}]$;
- le biais positif est estimé en utilisant les pseudo-variables $\tilde{\mathbf{X}}$ puis soustrait à l'importance estimée par la MDI des variables de $\mathbf{X}$ pour obtenir une estimation débiaisée.

Nembrini et al. (2018) ont par la suite modifié cette approche afin de réduire le temps de calcul et ont fourni des procédures empiriques de test d'importance, cependant la nouvelle méthodologie reste coûteuse en temps de calcul. Cette méthode est ensuite introduite formellement dans un cadre général et ces pseudo-variables sont appelées *knockoffs* (Barber and Candès, 2015).

CONDITIONAL MDA (CMDA) La CMDA est une méthode développée par Strobl et al. (2008) et son but est de corriger le biais d'échantillonnage de la MDA. Pour ce faire, la procédure de permutation de la MDA est modifiée. Une variable $\mathbf{x}^j$ est divisée en s blocs, où s est le nombre de fois où la variable est choisie dans un nœud de l'arbre. Les intervalles des blocs sont définis par les valeurs des coupures à chaque nœud pour

cette variable. Une fois les blocs obtenus, la variable est permutée à l'intérieur des blocs. Cette extension peut effacer le biais d'échantillonnage selon les auteurs. En revanche, elle ne résout pas les problèmes des biais de sélection, de préférence et de masquage. Mais étant donné que ce sont trois biais de construction, la CMDA semble appropriée pour représenter fidèlement la contribution des variables à la prédiction pour un modèle donné.

SOBOL-MDA La Sobol-MDA se base sur l'indice de Sobol total. L'indice de Sobol total de la variable $\mathbf{x}^j$ (Sobol, 1993 ; Saltelli, 2002) en régression est la proportion de variance de réponse, expliquée par le modèle, qui est perdue lorsque la variable $\mathbf{x}^j$ est retirée du modèle. Ainsi, Bénard et al. (2021) proposent d'estimer l'indice de Sobol total dans les forêts aléatoires à l'aide d'une procédure algorithmique se basant sur la MDA. Ils appellent cette mesure, la Sobol-MDA et elle estime l'indice de Sobol total, même lorsque les variables sont dépendantes et présentent des interactions (Bénard et al., 2021).

6.2.6 Clustering de variables

L'analyse des biais, dont certains sont liés à la construction des arbres, montre que pour interpréter une forêt aléatoire à l'aide d'une mesure d'importance de variables exempte de tout biais, il faut soit modifier la méthode des forêts aléatoires en elle-même soit modifier les données en entrée. Les forêts aléatoires sont des algorithmes dont les propriétés sont longtemps restées peu analysées et ce n'est que récemment que leur fonctionnement a été clarifié, tant empiriquement (exemple Duroux and Scornet (2018)) que théoriquement (Biau and Scornet, 2016).

Dans le cadre des forêts aléatoires, il existe deux études intégrant le clustering de variables. Une première est celle de Toloşi and Lengauer (2011), où les auteurs étendent l'idée de Park et al. (2007) au modèle des forêts aléatoires, en utilisant un clustering hiérarchique de type *average-linkage* des variables basé sur la distance euclidienne. Comme les variables sont centrées et réduites, cette *distance euclidienne* est équivalente à la distance de corrélation entre les variables. Les centres des clusters sont les moyennes des variables de chaque cluster et sont ensuite utilisés comme variables synthétiques pour l'apprentissage des forêts aléatoires. Ensuite, le nombre optimal de clusters est sélectionné de manière supervisée, en visitant chaque niveau du dendrogramme et en estimant l'erreur du modèle au moyen d'une validation croisée. La méthodologie de clustering et l'apprentissage des forêts aléatoires pour chaque niveau du dendrogramme sont coûteux en temps de calcul. Park et al. (2007) proposent alors d'utiliser au préalable un filtrage univarié pour éliminer les variables non importantes. Ensuite, l'importance des variables de départ est définie comme étant égale à celle de la variable synthétique de leur cluster.

La seconde méthodologie proposée par Chavent et al. (2021a) combine le clustering de variables et la sélection de variables avec les forêts aléatoires. Le clustering hiérarchique des variables permet de construire des clusters de variables corrélées et ensuite de résumer chaque cluster par une variable synthétique comme dans Park et al. (2007). L'originalité est que l'approche de clustering peut traiter à la fois des variables numériques et des variables catégorielles, et les variables synthétiques optimisent le critère de qualité des partitions. Enfin parmi toutes les partitions possibles, les variables synthétiques les plus importantes sont sélectionnées à l'aide d'une procédure VSURF (Genuer et al., 2010), basée sur les *recursive feature elimination* (Díaz-Uriarte and De Andres, 2006), qui utilise des forêts aléatoires. La sélection de groupes de variables peut permettre d'améliorer les performances en prédiction et facilite l'interprétation des résultats. Malheureusement, l'importance de variables de départ n'est pas calculée dans cette approche et c'est l'objet de notre proposition, dans le cas de variables de départ numériques.

6.3 Comparaison des méthodes existantes à l'aide d'un exemple simulé

MÉTHODOLOGIE D'ÉVALUATION Pour motiver notre proposition de mesure de l'importance des variables, nous partons d'un exemple introductif simulé suivant un modèle linéaire pour lequel nous allons comparer la MDA, la CMDA et la Sobol-MDA. Pour comparer les différentes mesures, il faut une vérité de terrain : la *vraie* importance des variables. Mais cela pose problème car il n'y a pas de définition admise de l'importance de variables et quand bien même il y en aurait une, rien n'indique que les méthodes que nous comparons estiment cette quantité.

Par ailleurs un but commun aux méthodes de mesure de l'importance des variables est de donner un classement de l'importance qui soit cohérent avec le classement donné par les *vraies* importances des variables. Or, dans le cadre de modèles linéaires additifs simulés, le *vrai* classement de l'importance des variables (au sens de l'objectif ii)) est donné par la corrélation au carré entre les variables d'entrée et la variable de sortie ; et c'est pour cela que nous faisons ce choix de modèle.

SIMULATION DE L'EXEMPLE Considérons alors le modèle linéaire suivant :

$$\mathbf{y} = \mathbf{x}^1 + \frac{2}{3}\mathbf{x}^2 + \frac{1}{3}\mathbf{x}^3 + \frac{1}{2}\mathbf{q}^1 + \boldsymbol{\varepsilon},$$

où

- $\mathbf{y} \in \mathbb{R}^n$ la variable à prédire ou variable de sortie ;
- $[\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{q}^1] \in \mathbb{R}^{n\times 4}$ sont 4 variables d'entrée simulées suivant une loi gaussienne multivariée de composantes indépendantes centrées réduites ;
- $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ simulée suivant une loi gaussienne univariée de moyenne de 0 et de variance de $0,5 \times \mathbf{I}_n$.

Le cas où les variables d'entrée sont uniquement des variables non corrélées et importantes ne peut pas illustrer les problèmes induits par les corrélations. Ainsi, 4 variables d'entrée qui n'ont pas servi à la construction de $\mathbf{y}$ sont ajoutées aux données :

1. $[\mathbf{q}^2, \mathbf{q}^3] \in \mathbb{R}^{n\times 2}$ sont 2 variables d'entrée simulées suivant une loi gaussienne multivariée de composantes centrées réduites où $\text{cor}(\mathbf{q}^j, \mathbf{q}^l) = 0.9,\ \forall j, l = 1, 2, 3\ j \neq l$ et indépendantes de $\mathbf{x}^j\ \ \forall j = 1, 2, 3$;
2. $[\mathbf{z}^1, \mathbf{z}^2] \in \mathbb{R}^{n\times 2}$ sont 2 variables d'entrée simulées suivant une loi gaussienne multivariée de composantes indépendantes centrées réduites, indépendantes de $\mathbf{y}$ et indépendantes de toutes les autres variables d'entrée.

On a donc un groupe de variables importantes indépendantes $[\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3]$, un groupe de variables importantes corrélées $[\mathbf{q}^1, \mathbf{q}^2, \mathbf{q}^3]$ et un groupe de variables de bruit $[\mathbf{z}^1, \mathbf{z}^2]$. Ainsi, nous obtenons une matrice de variables en entrée de la forêt $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{q}^1, \mathbf{q}^2, \mathbf{q}^3, \mathbf{z}^1, \mathbf{z}^2]$ et le but est d'expliquer la variable de sortie $\mathbf{y}$ à l'aide de $\mathbf{X}$ au moyen d'une forêt aléatoire et ensuite de trouver l'importance des variables de $\mathbf{X}$.

Dans ce manuscrit, nous optons pour le fait que toutes les variables $\mathbf{q}^1, \mathbf{q}^2, \mathbf{q}^3$ doivent être considérées comme importantes même si une seule d'entre elles intervient dans le modèle sous-jacent. En effet, si deux variables sont corrélées et que l'une fait partie du modèle, alors elles partagent une partie de l'information sur $\mathbf{y}$ et sont considérées toutes les deux comme importantes (cette vision est aussi en phase avec l'objectif ii)).

Le *vrai* classement est donné par les valeurs de $\text{cor}(\mathbf{y}, \mathbf{x}^j)^2$ pour toutes les variables de $\mathbf{X}$. Ensuite, le *vrai* classement pourra être comparé au classement donné par les importances estimées à l'aide de la corrélation de Spearman qui porte sur les rangs.

RÉSULTATS Une fois les données simulées avec $n = 500$, les forêts aléatoires sont entraînées à l'aide du package `R ranger` (Wright and Ziegler, 2015) avec les paramètres par défaut et le nombre d'arbres $T = 1000$. L'importance des variables va être calculée avec la MDA, la CMDA, la Sobol-MDA grâce là encore au package `R ranger` sur 100 échantillons indépendants.

La Table 6.1 compare le *vrai* classement (4ème ligne du tableau) avec les importances calculées avec la MDA, la CMDA, la Sobol-MDA. Comme on peut le constater, la MDA sous-estime l'importance des variables d'entrée corrélées avec $[\mathbf{q}^2, \mathbf{q}^3]$ et l'importance de $\mathbf{x}^3$ apparaît comme supérieure à celle de $\mathbf{q}^2$ alors qu'en réalité, il n'en est rien. La CMDA classe inexactement les variables en estimant une importance pour $\mathbf{x}^3$ supérieure à celles des variables $[\mathbf{q}^1, \mathbf{q}^2, \mathbf{q}^3]$, dont les importances estimées sont presque nulles ou nulles. La Sobol-MDA donne aussi une importance plus élevée à $\mathbf{x}^3$ qu'aux variables $[\mathbf{q}^1, \mathbf{q}^2, \mathbf{q}^3]$. De plus, $\mathbf{q}^2$ et $\mathbf{q}^3$ ont des importances calculées inférieures aux importances des variables de bruit. Enfin, la Sobol-MDA donne à $\tilde{\mathbf{x}}^1$ une importance proche de 0, suggérant, à tort, que la performance du modèle pourrait être améliorée en écartant $\mathbf{q}^1$.

Table 6.1 : Résultats des simulations. L'importance moyenne sur 100 échantillons simulés est rapportée. Les écarts types sont inférieurs à 10^{-3}.

	$\mathbf{x}^1$	$\mathbf{x}^2$	$\mathbf{x}^3$	$\mathbf{q}^1$	$\mathbf{q}^2$	$\mathbf{q}^3$	$\mathbf{z}^1$	$\mathbf{z}^2$
MDA	0.690	0.269	0.063	0.128	0.061	0.074	0.000	-0.001
Sobol-MDA	0.482	0.179	0.027	-0.002	-0.014	-0.015	-0.011	-0.011
CMDA	1.076	0.423	0.095	0.011	0.000	0.001	-0.001	-0.001
$\text{cor}(\mathbf{y}, \mathbf{x}^j)^2$	0.56	0.24	0.06	0.14	0.11	0.11	0.00	0.00

Finalement, la Table 6.1 montre bien que les objectifs i) et ii) sont contradictoires et que même sur un exemple aussi simple, il ne faut pas espérer obtenir une bonne estimation des relations entre les variables de $\mathbf{X}$ et $\mathbf{y}$ avec des méthodes satisfaisant l'objectif i).

6.4 Méthodologie de la solution proposée

L'idée est de réduire les corrélations des variables d'entrée des forêts aléatoires à l'aide d'un clustering hiérarchique de variables comme proposée dans Chavent et al. (2021a), où le choix du nombre de clusters se fait en minimisant l'erreur du modèle.

Nous proposons, en plus, une méthode de choix du nombre de clusters basée sur l'analyse graphique de l'erreur et de l'homogénéité des classes. En outre, une nouvelle mesure d'importance est définie, qui permet de calculer l'importance des variables d'origine via les variables synthétiques. Cette mesure d'importance est calculée avec la MDA sur les variables synthétiques.

Rappelons les quatre premières étapes de l'algorithme de Chavent et al. (2021a) (la cinquième étape consiste à appliquer l'algorithme VSURF et elle n'est pas utile ici) :

1. les variables d'entrée sont regroupées en fonction de leur structure de corrélation à l'aide de l'algorithme ClustOfVar Chavent et al. (2011) ;
2. chaque cluster de variables d'entrée est résumé par une variable synthétique et plus précisément par la première composante principale calculée au sein du cluster ;
3. un algorithme de forêt aléatoire est entraîné sur les variables synthétiques et le nombre de clusters K est fixé pour minimiser l'erreur de prédiction ;
4. la MDA associée des K variables synthétiques est calculée.

Cette méthodologie ne permet pas de calculer l'importance des variables d'origine. Ainsi nous proposons de prolonger la démarche de Chavent et al. (2021a) pour calculer l'importance des variables d'origine.

6.4.1 Clustering de variables et variables synthétiques

Comme défini dans Chavent et al. (2011, 2021a), une procédure de Clustering Ascendante Hiérarchique (CAH) des variables p d'entrée est utilisée pour construire des clusters de variables corrélées dans le but de réduire la redondance des informations. Dans ce chapitre nous l'utilisons uniquement dans le cas où les variables d'entrée sont numériques, mais ClustOfVar traite aussi le cas des données mixtes. On construit p partitions emboîtées des variables.

1. À l'étape 0 : on considère la partition en p clusters (c'est-à-dire une variable par cluster).
2. À chaque étape $l = 1, \ldots, p-2$: on agrège itérativement les deux clusters C_k et $C_{k'}$ de la partition précédente qui minimisent la dissimilarité $d(C_k, C_{k'})$.
3. Étape $l = p-1$: l'algorithme s'arrête lorsqu'il n'y a plus qu'un cluster.

La dissimilarité $d(C_k, C_{k'})$ entre deux clusters est définie comme :

$$d(C_k, C_{k'}) = H(C_k) + H(C_{k'}) - H(C_k \cup C_{k'}),$$

où $H(C_k)$ est l'homogénéité d'un cluster C_k qui mesure la relation entre les variables dans chaque cluster et elle est définie par :

$$H(C_k) = \sum_{\mathbf{x}_j \in C_k} \text{cor}(\mathbf{f}^k, \mathbf{x}^j)^2 = \lambda_k^1,$$

où le vecteur $\mathbf{f}^k$ est la première composante principale d'une ACP (ACP mixte dans le cas où les variables d'entrée sont numériques et catégorielles) restreinte aux variables de C_k et λ_k^1 est la première valeur propre (voir Chavent et al. (2011, 2021a) pour plus de détails).

En résumé, on choisit de représenter le cluster C_k au moyen de la première composante principale f^k (variable synthétique) d'une ACP effectuée sur les variables de ce cluster.

6.4.2 Le choix du nombre de clusters

La procédure ci-dessus établit une hiérarchie basée sur les corrélations (au carré) entre les variables d'entrée en termes d'informations redondantes, mais ne fournit pas le nombre optimal de clusters et donc le nombre optimal de variables synthétiques $\mathbf{f}^k$ à utiliser par la suite. Le choix peut s'avérer compliqué car, un mauvais choix du nombre de clusters de variables d'entrée peut induire de fortes corrélations entre les variables synthétiques, et donc une mauvaise estimation de l'importance de celles-ci. En revanche, résumer des clusters de variables peu corrélées entraîne une perte d'information et cela peut impacter les performances du modèle. Il y a donc un compromis à faire entre la performance et l'interprétabilité (objectif ii)) du modèle.

MINIMISER L'ERREUR DE PRÉDICTION Pour choisir le nombre de clusters, Park et al. (2007) ; Tolo̧si and Lengauer (2011) ; Chavent et al. (2021a) se basent uniquement sur la performance du modèle, en minimisant l'erreur. Pour chaque valeur de K et pour chaque partition de variables d'entrée associée, on entraîne une forêt aléatoire sur les seules variables synthétiques et on calcule l'erreur OOB. Ensuite, le nombre optimal de clusters de variables $K^\star$ est celui qui minimise cette erreur.

En revanche, certains ensembles de données réelles ont des structures de corrélations complexes et il peut être nécessaire de compléter la méthode par une analyse simultanée de l'erreur et de l'homogénéité des clusters de variables d'entrée.

CHEMINS D'ERREUR ET D'HOMOGÉNÉITÉ Dans certains cas, l'erreur peut être minimum pour une valeur de K voisine de p voire pour $K = p$. En revanche, il peut exister des solutions en K clusters avec $K \ll p$ qui ont des performances très proches du modèle ayant l'erreur OOB minimum. De plus, le clustering des variables réduit mais ne supprime pas complètement la corrélation puisque les variables synthétiques ne sont pas nécessairement orthogonales. Prendre en compte en même temps ces informations peut aider au choix de K.

Tout d'abord, une information sur la perte d'information dans les données (information qui peut être utile pour expliquer la variable de sortie) peut être obtenue en observant l'homogénéité des clusters de variables d'entrée en fonction de K. Pour une partition $P_K = \{C_1, \ldots, C_K\}$ en K classes, l'homogénéité de la partition P_K, que l'on note $\mathcal{H}(P_K)$, se calcule comme la somme des homogénéités des clusters :

$$\mathcal{H}(P_K) = \sum_{k=1}^{K} H(C_k) = \sum_{k=1}^{K} \sum_{j \in C_k} \text{cor}(\mathbf{f}^k, \mathbf{x}^j)^2 = \sum_{k=1}^{K} \lambda_k^1 \leqslant p.$$

Pour obtenir le chemin d'homogénéité en fonction de K, il faut afficher les valeurs du vecteur $(\mathcal{H}(P_1)/p, ..., \mathcal{H}(P_K)/p, ..., \mathcal{H}(P_p))^\intercal$ en fonction de $(1, ..., K, ..., p)^\intercal$.

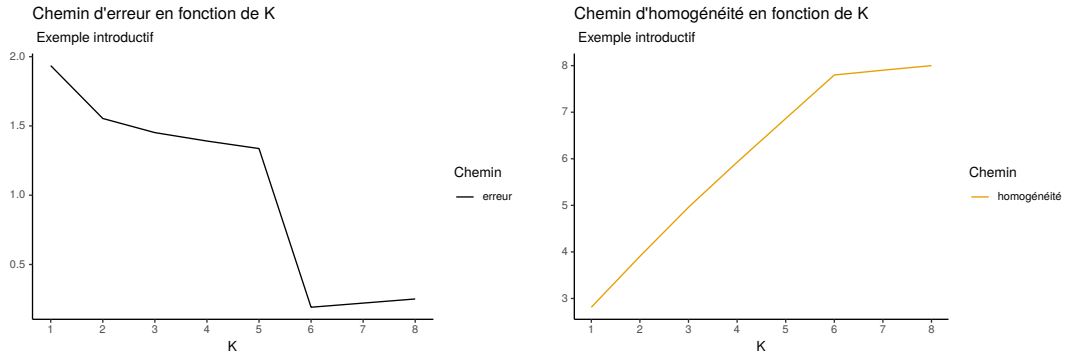
Deuxièmement, Une information de la réduction des corrélations entre les variables synthétiques en entrée de la forêt peut être obtenue en observant la corrélation totale, c'est-à-dire la somme de tous les termes au carré de la matrice de corrélations des variables synthétiques, et cela en fonction de K. Cette partie est encore inachevée et sera examinée dans des travaux postérieurs à ce travail thèse.

L'interprétation des deux chemins (erreur, homogénéité) est donnée ci-dessous :

1. l'erreur doit être minimisée, car un modèle peu performant implique une mauvaise estimation des relations entre les variables d'entrée et la variable de sortie.

2. l'homogénéité doit être maximisée, car cela implique qu'il y a peu de perte d'information, information qui peut être utile pour expliquer la variable de sortie.

Une analyse simultanée du graphe d'erreur OOB et du chemin d'homogénéité permet alors de déterminer le nombre de clusters des variables d'entrée réalisant ainsi le compromis cherché entre performance et interprétabilité.



(a) Chemin d'erreur (b) Chemin d'homogénéité

Figure 6.1 : Le graphique (a) représente le chemin d'erreur en fonction de K et le graphique (b) représente le chemin d'homogénéité en fonction de K. L'erreur OOB est minimum lorsque $K = 6$ et donc le clustering de variables améliore les performances par rapport au modèle sans cluster ($K = 8$). Lorsque $K = 6$, l'homogénéité atteint un plateau. L'analyse jointe de ces deux graphiques nous informe que le modèle entraîné sur la partition en 6 classes est le meilleur du point de vue de la performance et de l'interprétabilité.

ILLUSTRATION SUR L'EXEMPLE INTRODUCTIF Reprenons l'exemple décrit dans la Section 6.3 de ce chapitre. Nous rappelons que pour cet exemple, nous disposons d'un ensemble de données $\mathbf{X}$ avec 3 variables importantes non corrélées $[\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3]$, 3 variables importantes corrélées $[\mathbf{q}^1, \mathbf{q}^2, \mathbf{q}^3]$ (avec des corrélations par paire égales à 0.9) et 2 variables de bruit non corrélées $[\mathbf{z}^1, \mathbf{z}^2]$. Le clustering de variables est effectué à l'aide du package `R ClustOfVar`.

La Figure 6.1 représente pour un échantillon les trois chemins en fonction de K. L'exemple introductif est schématique avec une structure en corrélations très simple. Mais malgré la simplicité de l'exemple, le clustering de variables améliore les résultats de prédiction et on remarque que l'erreur OOB est minimum lorsque $K = 6$. L'erreur minimum est atteinte pour la partition en 6 clusters, et les groupes formés sont $\{\mathbf{x}^1\}, \{\mathbf{x}^2\}, \{\mathbf{x}^3\}, \{\mathbf{z}^1\}, \{\mathbf{z}^2\}$ seules et $\{\mathbf{q}^1, \mathbf{q}^2, \mathbf{q}^3\}$ ensemble ce qui correspond effectivement à la structure en corrélations simulée. Le dendrogramme des partitions est visible sur la Figure 6.2.

En outre, avec l'analyse graphique, on cherche l'entier K le plus petit (car cela facilite l'interprétabilité que d'avoir peu de clusters), tel que l'erreur et l'homogénéité soient toutes les deux respectivement minimum et maximum.

Sur certains ensembles de données avec des structures en corrélations plus complexes, le chemin d'erreur OOB peut être strictement décroissant en fonction de K (ou pire, former une droite) et une analyse des chemins peut mener à un choix plus judicieux que celui qui est fait par la procédure automatique.

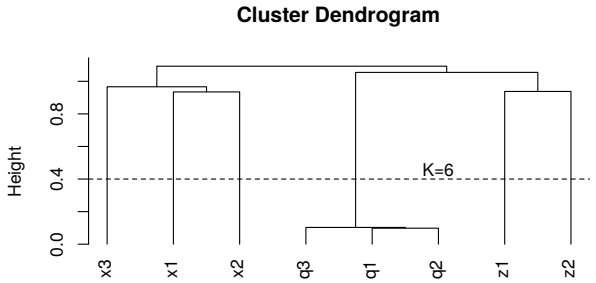


Figure 6.2 : Le graphique représente le dendrogramme des partitions obtenu avec le package `R ClustOfVar` sur un échantillon simulé suivant le schéma de l'exemple introductif. L'analyse du dendrogramme suggère une partition en $K = 6$ clusters de variables avec $\{\mathbf{q}^1, \mathbf{q}^2, \mathbf{q}^3\}$ regroupées ensemble, ce qui correspond à la structure en corrélation simulée.

6.4.3 Nouvelle mesure de l'importance des variables : Synthetic-MDA (SMDA)

La procédure présentée ci-dessus permet de calculer la MDA pour les variables synthétiques uniquement et n'évalue pas l'importance des variables d'origine. Pour combler cette lacune, on définit donc un nouveau critère que l'on nomme Synthetic-MDA (SMDA). La mesure SMDA conditionnellement à une partition de variables d'entrée en $K^\star$ clusters est définie comme suit :

$$\text{SMDA}_{K^\star}(\mathbf{x}^j) = \text{MDA}(\mathbf{f}^{k(j)}) \times \text{cor}(\mathbf{f}^{k(j)}, \mathbf{x}^j)^2,$$

où $k(j)$ est le numéro du cluster auquel appartient la variable d'entrée $\mathbf{x}^j$, $\mathbf{f}^{k(j)}$ est la variable synthétique résumant ce même cluster et $K^\star$ est le nombre de variables synthétiques choisi pour construire la forêt. Le premier terme $\text{MDA}(\mathbf{f}^{k(j)})$ nous donne l'importance des variables synthétiques et le deuxième terme $\text{cor}(\mathbf{f}^{k(j)}, \mathbf{x}^j)^2$ mesure le degré de similarité entre la variable $\mathbf{f}^{k(j)}$ et la variable $\mathbf{x}^j$. Ainsi :

1. si $\text{MDA}(\mathbf{f}^{k(j)})$ est grand (respectivement petit) et $\mathbf{x}^j$ est corrélée à $\mathbf{f}^{k(j)}$ alors $\mathbf{x}^j$ sera importante (respectivement peu importante) ;
2. si $\mathbf{x}^j$ n'est pas corrélée à $\mathbf{f}^{k(j)}$ alors $\mathbf{x}^j$ ne sera pas importante.

Deux remarques peuvent être faites. Tout d'abord, on souhaite que la corrélation entre les variables $\mathbf{f}^{k(j)}$, $k = 1, ..., K^\star$ soit faible et que donc le calcul de la MDA ne soit pas biaisé (cette information peut être donnée par la corrélation totale). Deuxièmement, on espère que $\text{cor}(\mathbf{f}^{k(j)}, \mathbf{x}^j)^2$ soit proche de 1, pour $k(j) = 1, ..., K^\star$, $j = 1, ..., p$. En effet, si c'est le cas, alors le clustering de variables d'entrée a entraîné peu de perte d'information, information qui peut être utile pour expliquer la variable de sortie. Par exemple, une variable importante peut être regroupée avec des variables de bruit corrélées et on peut se retrouver dans le cas numéro 2 de la liste ci-dessus. C'est un défaut notable de la méthode, mais il est à nuancer car si une variable est importante, les variables auxquelles elle est très corrélée sont aussi importantes (car elles partagent la même information sur la variable de sortie) et donc elles seront regroupées ensemble. Cette perte d'information peut s'analyser avec l'homogénéité de la partition et, plus globalement, avec le chemin d'erreur du modèle associé.

IMPLÉMENTATION DE LA SMDA Pour calculer la SMDA, nous utilisons le package `R ClustOfVar` (Chavent et al., 2011) qui construit le clustering de variables puis le package `R ranger` pour l'apprentissage des forêts aléatoires et pour le calcul de la MDA des variables synthétiques. Le code implémentant l'ensemble de notre méthodologie est disponible sous forme de code `R` sur GitHub*.

ILLUSTRATION SUR L'EXEMPLE INTRODUCTIF Reprenons l'exemple décrit dans la Section 6.3 de ce chapitre. Nous rappelons que pour cet exemple, nous disposons d'un ensemble de données $\mathbf{X}$ avec 3 variables importantes non corrélées, 3 variables importantes corrélées (avec des corrélations par paire à 0.9) et 2 variables de bruit non corrélées.

Table 6.2 : Résultats agrégés sur 100 échantillons de l'exemple introductif avec $n = 500$. L'importance moyenne sur 100 échantillons simulés est rapportée. Les écarts types sont inférieurs à 10^{-3}.

	$\mathbf{x}^1$	$\mathbf{x}^2$	$\mathbf{x}^3$	$\mathbf{q}^1$	$\mathbf{q}^2$	$\mathbf{q}^3$	$\mathbf{z}^1$	$\mathbf{z}^2$
MDA	0.690	0.269	0.063	0.128	0.061	0.074	0.000	-0.001
Sobol-MDA	0.482	0.179	0.027	-0.002	-0.014	-0.015	-0.011	-0.011
CMDA	1.076	0.423	0.095	0.011	0.000	0.001	-0.001	-0.001
SMDA	0.802	0.311	0.064	0.151	0.151	0.150	0.001	0.001
$\text{cor}(\mathbf{y}, \mathbf{x}^j)^2$	0.56	0.24	0.06	0.14	0.11	0.11	0.00	0.00

La Table 6.2 présente les résultats agrégés de 100 simulations pour les mesures MDA, Sobol-MDA, CMDA et SMDA (en utilisant la procédure de sélection automatique de K). Les résultats des trois premières mesures ont été discutés dans la Section 6.3 et leurs résultats étaient mitigés lorsque l'on s'intéressait à l'objectif d'interprétabilité ii). Par ailleurs, la SMDA fournit une bonne estimation du classement, et les valeurs calculées apparaissent comme significatives. Pour la procédure SMDA, sur les 100 simulations, l'algorithme a sélectionné à chaque fois $K^\star = 6$, et a regroupé les variables d'entrée $\{\mathbf{q}^1, \mathbf{q}^2, \mathbf{q}^3\}$ ensemble.

6.5 Simulations

LE MODÈLE GLOBAL Dans cette partie, nous allons effectuer de nouvelles simulations. Pour ce faire, nous considérons le modèle linéaire suivant :

$$\mathbf{y} = \sum_{j=1}^{p_1} \beta_j \times \mathbf{x}^j + 0.5 \times \mathbf{q}^1 + \boldsymbol{\varepsilon}, \text{ avec } \beta_j = \frac{p_1 - j + 1}{p_1}, \ j = 1, \ldots, p_1,$$

où

- $\mathbf{y} \in \mathbb{R}^n$ la variable à prédire ou variable de sortie ;
- $[\mathbf{x}^1, ..., \mathbf{x}^{p_1}, \mathbf{q}^1] \in \mathbb{R}^{n \times (p_1+1)}$ sont $p_1 + 1$ variables d'entrée simulées suivant une loi gaussienne multivariée de composantes indépendantes centrées réduites ;
- $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ est simulée suivant une loi gaussienne univariée de moyenne de 0 et de variance de $0,5 \times \mathbf{I}_n$.

Comme précédemment, des variables d'entrée qui n'ont pas servi à la construction de $\mathbf{y}$ sont ajoutées aux données :

1. $[\mathbf{q}^2, ..., \mathbf{q}^{p_2}] \in \mathbb{R}^{n \times (p_2-1)}$ sont $p_2 - 1$ variables d'entrée simulées suivant une loi gaussienne multivariée de composantes centrées réduites où $\text{cor}(\mathbf{q}^j, \mathbf{q}^l) = 0.9, \ \forall j, l = 1, ..., p_2, \ j \neq l$ et indépendantes des $\mathbf{x}^j, \ \forall j = 1, ..., p_1$;
2. $[\mathbf{c}^1, ..., \mathbf{c}^{d_1}] \in \mathbb{R}^{n \times d_1}$ sont d_1 variables d'entrée simulées suivant une loi gaussienne multivariée de composantes centrées réduites où $\text{cor}(\mathbf{c}^j, \mathbf{c}^l) = 0.9, \ \forall j, l = 1, ..., p_2 \ j \neq l$, indépendantes de $\mathbf{y}$ et indépendantes de toutes les autres variables d'entrée ;
3. $[\mathbf{z}^1, ..., \mathbf{z}^{d_2}] \in \mathbb{R}^{n \times d_2}$ sont d_2 variables d'entrée simulées suivant une loi gaussienne multivariée de composantes indépendantes centrées réduites, indépendantes de $\mathbf{y}$ et indépendantes de toutes les autres variables d'entrée.

On a donc quatre groupes de variables en entrée de la forêt aléatoire :

*https ://github.com/MourerAlex/SMDA

1. $\mathbf{X}_{p_1} = [\mathbf{x}^1, ..., \mathbf{x}^{p_1}]$ des variables importantes indépendantes ;
2. $\mathbf{Q}_{p_2} = [\mathbf{q}^1, ..., \mathbf{q}^{p_2}]$ des variables importantes corrélées ;
3. $\mathbf{Z}_{d_1} = [\mathbf{z}^1, ..., \mathbf{z}^{d_1}]$ des variables de bruit indépendantes.
4. $\mathbf{C}_{d_2} = [\mathbf{c}^1, ..., \mathbf{c}^{d_2}]$ des variables de bruit corrélées ;

Ainsi, nous obtenons une matrice de variables en entrée de la forêt $\mathbf{X} = [\mathbf{X}_{p_1}|\mathbf{Q}_{p_2}|\mathbf{Z}_{d_1}|\mathbf{C}_{d_2}]$ et le but est d'expliquer la variable de sortie $\mathbf{y}$ avec les variables de $\mathbf{X}$ au moyen d'une forêt aléatoire et ensuite de trouver l'importance des variables de $\mathbf{X}$ et de les classer selon leur importance. La SMDA est utilisée avec la procédure automatique de choix du nombre de clusters.

Nous simulons 100 échantillons de taille $n = 500$ et nous fixons $p_1 = 10, d_1 = 25$ (variables non corrélées), pour différents scénarios suivant p_2 et d_2 comme décrit dans le Table 6.3. La comparaison des mesures d'importance des variables a été étendue ici à la MDI et la MDI avec *knockoffs* appelée MDICor dans la Table 6.3, cette dernière étant censée gérer la corrélation, selon Wright and Ziegler (2015). Le classement obtenu avec la *vraie* importance, donnée par $\text{cor}(\mathbf{y}, \mathbf{x}^j)^2$, a été comparé au classement obtenu avec l'importance calculée par les différents critères en utilisant la corrélation de Spearman. Le pourcentage de variables importantes parmi les variables d'entrée $p_1 + p_2$ classées en premier est également indiqué.

Table 6.3 : Corrélation de Spearman (Sp) entre le *vrai* classement et le classement donné par les importances estimées, et le pourcentage de variables importantes dans les $p_1 + p_2$ premières variables d'entrée classées (%sel). $p_1 = 10, d_1 = 25$. Les valeurs moyennes sur 100 échantillons sont indiquées. Les écarts-types sont indiqués entre parenthèses. La SMDA est utilisé avec la procédure automatique de choix du nombre de clusters.

	$p_2 = 1; d_2 = 50$		$p_2 = 50; d_2 = 0$		$p_2 = 50; d_2 = 50$	
	Sp	%sel	Sp	%sel	Sp	%sel
MDICor	**0.41**(.089)	**86**(7.3)	0.64 (.085)	91(4.1)	0.68(.053)	87(6.1)
MDI	**0.42**(.045)	**87**(5.9)	0.13(.182)	61(3.2)	0.59(.072)	65(4.5)
MDA	**0.27**(.054)	**73**(5.7)	**0.75**(.026)	**98**(1.1)	0.69(.051)	88(4.1)
CMDA	0.21(.078)	**81**(6.4)	0.15(.152)	81(3.2)	0.16(.087)	43(5.2)
Sobol-MDA	**0.27**(.086)	**82**(7.1)	0.11(.101)	67(2.9)	0.25(.099)	31(4.5)
SMDA	**0.42**(.093)	**89**(6.8)	**0.77**(.031)	**98**(1.4)	**0.81**(.035)	**98**(0.9)

Nous expliquons les résultats des mesures existantes pour les trois scénarios dans la liste ci-dessous.

1. Le cas $p_2 = 1, d_2 = 50$ devrait être le plus facile pour tous les critères, puisque toutes les variables importantes sont indépendantes, les autres étant des variables de bruit et en effet on observe que la plupart des critères sont équivalents. De plus, le biais de sélection (variables interchangeables) et le biais de préférence (préférence pour les variables corrélées) impliquent une réduction de l'importance des variables corrélées, qui sont ici les variables de bruit, ce qui avantage les différentes mesures. En revanche, le biais de permutation (extrapolation du modèle sur une nouvelle distribution) implique une augmentation de l'importance des variables corrélées ce qui explique pourquoi la MDA a de moins bons résultats. La CMDA et la Sobol-MDA sont censées éviter ce biais de permutation et on voit, à l'aide de la colonne %sel, qu'elles parviennent un peu mieux que la MDA à différencier les variables importantes des variables de bruit.

2. Lorsque les variables corrélées sont uniquement importantes ($p_2 = 50, d_2 = 0$), toutes les mesures sauf la MDA et la SMDA obtiennent de mauvais résultats. Le biais de permutation est en faveur de la MDA, augmentant l'importance des variables corrélées importantes et contrebalançant donc les biais de sélection et de préférence.

3. Pour le dernier cas, la présence de variables importantes corrélées et de variables de bruit corrélées ($p_2 = 50, d_2 = 50$) détériore le résultat de toutes les mesures, sauf la SMDA. Les méthodes MDI, MDICor, CMDA et Sobol-MDA sont encore sujettes aux biais de préférence et de sélection des variables importantes (discriminantes). Pour la MDA, le biais de permutation est aussi présent sur les variables de bruit rendant la tâche plus compliquée.

Globalement, la SMDA a de bons résultats. Le nombre de clusters choisis est presque équivalent dans les trois contextes et est en moyenne égal à 37 avec un écart-type de 5. Ce nombre est cohérent sachant qu'il y a $p_1 + d_1 = 35$ variables non corrélées et 2 groupes (un lorsque $d_2 = 0$) de variables corrélées. En particulier, selon les simulations, il est rare que les variables non corrélées soient regroupées et les groupes de variables corrélées peuvent avoir été séparés en deux clusters.

6.6 Application industrielle : flottement *fan*

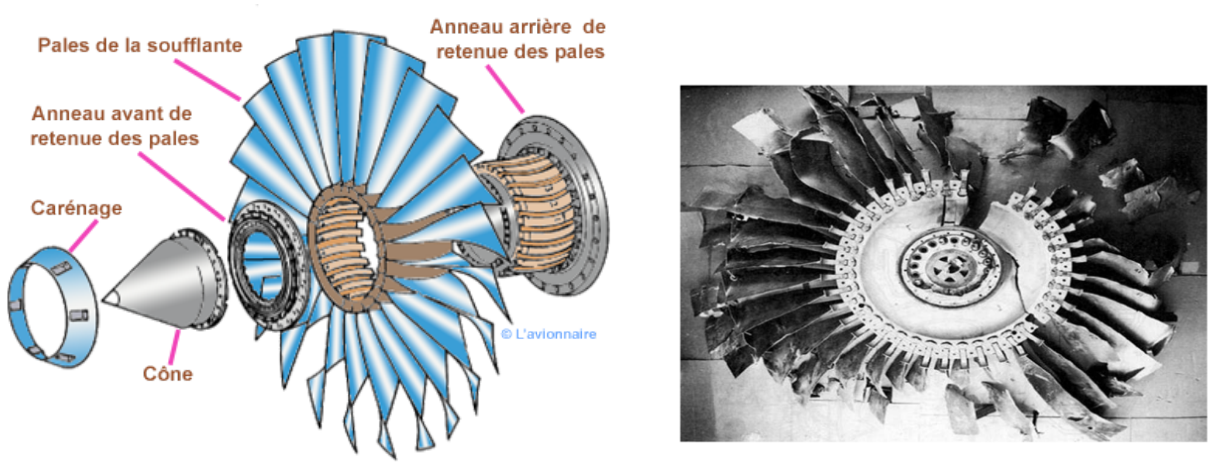
6.6.1 Introduction

PRÉSENTATION Dans les turboréacteurs (moteurs *turbofan*), les aubes *fan* (aussi appelés pales de soufflante ou aubages ; voir la Figure 6.3 (a)) sont soumises à de nombreuses sollicitations statiques et dynamiques d'origine aérodynamique ou mécanique. Ces phénomènes sont nombreux et difficiles à anticiper d'où l'existence de nombreuses marges (pompage, flottement) calculées à partir d'outils spécifiques et de modèles théoriques. Les excitations varient fortement suivant les positions étudiées dans le moteur et celles liées aux aubes *fan* peuvent être classées en deux catégories : les phénomènes aéroélastiques et les phénomènes mécaniques.

Baignées dans un flux d'air, les aubes *fan* sont soumises à des sollicitations vibratoires d'origine aérodynamique. Ces excitations ont deux effets principaux : elles peuvent provoquer une instabilité des aubes *fan* (phénomène de flottement) ou une résonance de la structure sur un (ou plusieurs) de ses modes de vibration, ce qui génère des contraintes susceptibles de provoquer de la fatigue vibratoire (voir la Figure 6.3 (b)).

Afin de réduire ces niveaux vibratoires et augmenter la résistance des aubes *fan*, la première stratégie consiste à apporter de l'amortissement par ajout de matériau viscoélastique.

Néanmoins, pour tenter de corriger le problème à la source il faudrait une meilleure compréhension des phénomènes physiques définissant ces phénomènes vibratoires. Les modèles théoriques physiques n'étant pas suffisants, nous allons mener une analyse statistique du problème.



(a) Aubes *fan* (pales de la soufflante) (source : Sénéchal (2011))

(b) Fatigue vibratoire (source : Sénéchal (2011))

Figure 6.3 : Le graphique (a) schématise les pales de la soufflante ou aubes *fan* et le graphique (b) montre une roue aubagée reconstruite suite à une rupture de disque due aux phénomènes de fatigue.

DÉFINITIONS De manière plus précise, les objectifs assignés à une aube *fan* peuvent se décliner autour de quatre thèmes majeurs :

1. la performance, mesurée par le débit et le rendement aérodynamique ;
2. la sécurité, certifiée par des tests de décrochement d'aubes et d'aspiration (sable, gravier/cailloux, objets en plastique, reproduction d'oiseaux artificiels, glace) ;
3. l'acoustique, notamment le bruit émis par les aubes *fan* ;
4. l'opérabilité, attestée par la marge au pompage et la marge au flottement.

Le phénomène de flottement est un terme générique, qui désigne les phénomènes d'instabilité aéroélastique. Le flottement se caractérise, du point de vue de la structure, par un amortissement aérodynamique élevé (Figure 6.4), de telle sorte que l'énergie apportée par le fluide est supérieure à celle dissipée dans la structure. Définissons alors formellement ce qu'est la *marge au flottement*.

Définition. *La marge au flottement mesure, à un débit de carburant donné, l'écart de pression entre la ligne de flottement (ligne rouge sur la Figure 6.4) et la ligne de fonctionnement (ligne noire sur la Figure 6.4) lorsque l'on fait varier la position de la tuyère primaire, ce que l'on appelle Variable Fan Nozzle (VFN).*

Lors du cycle de conception, et afin d'éviter la découverte tardive de ce problème au cours des essais moteur, la marge au flottement est calculée théoriquement. Mais ce calcul théorique est complexe et ainsi il est nécessaire de conduire une analyse statistique devant déterminer les facteurs influents de la marge. Plus d'informations sur le flottement *fan* et la réduction de vibrations sont disponibles dans les thèses suivantes Sénéchal (2011) ; Mabilia (2020).

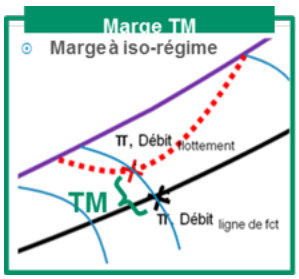


Figure 6.4 : Le graphique (source : Safran Aircraft Engines) schématise la marge au flottement (notée marge TM sur le graphique). L'axe des ordonnées correspond à un rapport de compression et l'axe des abscisses correspond au débit de carburant utilisé. Les lignes bleues correspondent aux différents régimes de rotation, la ligne noire à la ligne de fonctionnement (théorique), la ligne en pointillé rouge à l'apparition d'un flottement et la ligne violette au point de rupture.

CONTEXTE ET MOTIVATIONS En 2019 et 2020, des essais de caractérisation de la marge au flottement réalisés sur des moteurs ont mis en évidence l'impact du jeu radial (jeu perpendiculaire à l'axe) en bout d'aube sur la marge au flottement. Cet effet a été mis en évidence par la réalisation d'un certain nombre d'essais avec différentes tailles de jeu radial, réalisé par un ajout ou un enlèvement d'une épaisseur de matériaux abradable[†]. Un impact du jeu radial sur la marge au flottement a ensuite été déterminé sur la base de quatre essais réalisés, en supposant une évolution linéaire de la marge au flottement en fonction du le jeu radial, pouvant dépendre du régime de rotation.

D'après les analyses des essais réalisées par les bureaux d'étude, il y a donc un impact fort du jeu radial sur la marge au flottement impact qui est également dépendant du régime de rotation. Par ailleurs, les moteurs en service sont soumis à de la dispersion de production (longueur d'aube, épaisseur de la cartouche d'abradable, dimension du carter *fan*, etc.) qui se traduisent par une dispersion du jeu radial en bout d'aube lors de la livraison des moteurs. Par la suite, en service, ils sont également soumis à une détérioration (perte de matière en tête d'aube, usure de l'abradable, etc.) qui conduit à une ouverture du jeu radial. Ces deux aspects peuvent avoir un impact sur la marge au flottement. La marge au flottement calculée par des calculs physiques théoriques sur les moteurs étant quasiment nulle, il est nécessaire de réaliser une analyse plus approfondie pour comprendre l'impact observé du jeu radial.

Une question majeure doit ainsi être étudiée : l'impact du jeu radial en bout d'aube sur la marge au flottement est-il significatif par rapport à celui des variables liées à la poussée du moteur ?

6.6.2 Analyse de l'impact du jeu radial en bout d'aube sur la marge au flottement

DESCRIPTION DES DONNÉES Nous disposons de données qui sont telles que :

- d'une matrice $\mathbf{X} \in \mathbb{R}^{n \times p}$ de variables d'entrée avec $n = 121$ moteurs et $p = 1284$ variables explicatives, correspondant à des mesures effectuées sur le moteur ou des caractéristiques de celui-ci. Toutes les mesures ont été faites avec les mêmes équipements de bancs. Parmi ces 1284 variables, 200 décrivent les jeux des aubes, 30 de ces 200 décrivent le jeu radial et 7 de ces 30 le jeu radial en bout d'aubes.
- d'une variable à expliquer ou variable de sortie $\mathbf{y} \in \mathbb{R}^{121}$ qui représente la marge au flottement et qui est donc calculée en prenant la *distance* entre le point d'apparition des niveaux vibratoires et une ligne de fonctionnement de référence.

DESCRIPTION DE LA MÉTHODE Pour cette analyse, nous allons utiliser un modèle un peu plus complexe que les forêts aléatoires appelé CoV/VSURF (Chavent et al., 2021a). La méthode CoV/VSURF est quasiment en tous points équivalente à la méthode utilisée dans ce chapitre : on utilise ClustOfVar (CoV) (Chavent et al., 2011) pour obtenir un clustering de variables d'entrée, chaque cluster est résumé par une variable synthétique

[†]Les contacts entre le stator (la partie fixe d'une machine rotative) et le rotor (la partie rotative d'une machine, qui tourne dans le stator) des roues aubagées provoquent des phénomènes d'instabilités vibratoires. Afin de pallier ces problèmes, certaines machines sont pourvues de couches de matériau abradable dans les zones où le contact est susceptible de survenir.

(première composante principale), le nombre de clusters est choisi pour minimiser l'erreur OOB avec des forêts aléatoires et on obtient $K^\star$ variables synthétiques sur lesquelles on applique la procédure VSURF (Genuer et al., 2015). Ainsi la seule différence est l'utilisation de la procédure VSURF pour la dernière étape à la place d'une simple forêt aléatoire. La procédure VSURF est une procédure par étape qui se base sur les forêts aléatoires et sur une stratégie d'élimination récursive des variables. Techniquement, l'algorithme fonctionne en trois étapes :

1. la première consiste en un tri des variables sur la base de la mesure MDA des variables synthétiques obtenue avec les forêts aléatoires. On élimine les variables inutiles par un seuillage adaptatif. Le seuil est fixé à l'aide de l'écart-type (estimée) de la MDA d'une variable sans importance.

2. la seconde commence avec les variables précédemment conservées (étape 1) et applique une stratégie d'introduction de variables de manière ascendante (de type *forward*). La forêt qui atteint le taux d'erreur OOB minimum est alors sélectionnée et l'ensemble de variables sur lequel elle est basée est appelé *ensemble d'interprétation*.

3. la troisième étape consiste à éliminer de la redondance des variables d'interprétation et conduit à un ensemble plus petit de variables appelé *ensemble de prédiction*. Elle consiste en une stratégie ascendante par étapes, qui vérifie à chaque étape que la variable suivante (à introduire) contribue à diminuer suffisamment l'erreur OOB, selon un seuil fixé automatiquement (voir Genuer et al. (2015) pour plus de détails).

Ainsi, la méthode VSURF, qui permet de faire de la sélection de variables tout en réduisant la dimension de l'espace d'entrée avec le clustering de variables, va nous être très utile car nous sommes dans un cas de grande dimension ($n \ll p$) avec des variables très corrélées.

Enfin, étant donné que la méthode CoV/VSURF conduit à un modèle de forêt aléatoire entraîné sur des variables synthétiques (elles-mêmes construites avec la méthode ClustOfVar) nous pouvons calculer l'importance des variables des clusters avec la SMDA.

DESCRIPTION DES RÉSULTATS Nous appliquons la méthologie CoV/VSURF à l'aide du code `R` disponible sur GitHub[‡]. Les paramètres sont laissés par défaut et le nombre d'arbres fixé à 1000.

Table 6.4 : Variables sélectionnées par le modèle CoV/VSURF. Trois groupes ressortent : une variable seule dans son groupe qui donne le point de fonctionnement du fan (colonne PFF), un groupe de variables liées à la poussée du moteur (colonne Poussée), un groupe de variables décrivant le jeu radial (colonne Jeu radial). Les corrélations au carré entre la variable synthétique et les variables du groupe sont données ainsi que la SMDA. La MDA des variables synthétiques est donnée en dernière ligne.

	PFF	Poussée			Jeu radial		
	nom	noms	$\text{cor}(\mathbf{f}^k, \mathbf{x}^j)^2$	SMDA	noms	$\text{cor}(\mathbf{f}^k, \mathbf{x}^j)^2$	SMDA
	VFN	P15Q12	0.98	0.147	F01 bord d'attaque S90D 1H30	0.99	0.108
		P15	0.98	0.147	TIPGAP S90D 10H30	0.99	0.108
		XN12R	0.96	0.144	TIPGAP S90D 10H30	0.99	0.108
		T15	0.92	0.138	TIPGAP S180D 4H30	0.98	0.107
		T15Q12	0.92	0.138	TIPGAP S180D 10H30	0.98	0.107
		W2AR	0.90	0.135	TIPGAP S90D 1h30	0.98	0.107
					TIPGAP S90D 7H30	0.97	0.106
					TIPGAP S180D 1H30	0.97	0.106
					F01 bord d'attaque S180D 10H30	0.97	0.106
MDA	1	0.15			0.11		

Dans la Table 6.4, les trois groupes de variables, regroupées et sélectionnées par CoV/VSURF sont présentés. La position VFN est par définition liée à la marge au flottement. C'est une variable très liée à la marge au flottement et donc il est normal qu'elle soit sélectionnée. Les MDA relatives (divisées par le max) des variables synthétiques sont données dans le tableau et on voit que La position VFN est la plus influente et c'est ce qui était attendu. Ensuite, les variables liées à la poussée du moteur apparaissent. Enfin un groupe décrivant le jeu radial a été sélectionné. Il est uniquement formé de variables décrivant le jeu radial en bout d'aube, mais toutes les variables (décrivant le jeu radial en bout d'aube) ne sont pas présentes dans les groupes. Les mesures TIPGAP sont les plus représentées, et elles sont calculées comme étant un jeu radial pondéré à 3/5 en bord d'attaque, 1/5 en milieu de corde 1/5 en bord de fuite, ce qui est quand même très représentatif de l'information en bout d'aube.

[‡]https ://github.com/robingenuer/CoVVSURF

Finalement, seulement une partie des variables décrivant le jeu radial en bout d'aube a été sélectionnée. Mais la petite taille de l'échantillon, la grande dimension des données et la présence de nombreux groupes de variables, de différentes tailles, très corrélées font que les tâches de clustering de variables et de sélection de variables sont très compliquées. En revanche, le modèle a quand même sélectionné un groupe de variables lié à l'information du jeu radial en bout d'aube ce qui montre que celui-ci à effectivement un impact significatif sur la marge au flottement.

6.7 Conclusion

Ce chapitre développe un critère d'importance de variables basé en partie sur une méthodologie existante mais les motivations et les objectifs (notamment l'objectif ii) d'interprétabilité) de la méthode nous semblent quant à eux primordiaux et encore peu abordés. Ainsi, nous avons proposé la méthode SMDA, destinée à estimer les relations entre les variables d'entrée et la variable de sortie.

Nous avons vu que deux biais sont liés à la construction des arbres et un à celui de la forêt. Ainsi, deux solutions pour pallier ces biais sont : de modifier la construction de la forêt ou de modifier les données d'entrée. Cela contraste avec l'idée que les méthodes d'importance post-hoc devraient représenter fidèlement le modèle, montre que tous les modèles ne sont pas intrinsèquement interprétables et qu'il est nécessaire de développer conjointement des modèles et des mesures d'importance de variables lorsque l'on s'intéresse à l'objectif ii).

La méthodologie présente les avantages suivants :

- le clustering de variables permet de réduire les corrélations entre les variables d'entrée et donc d'effacer en partie les biais ;
- l'analyse de l'homogénéité du clustering de variables d'entrée permet de rendre compte de la qualité de la réduction des corrélations ;
- associé au graphe de l'erreur, le chemin d'homogénéité permet un choix du nombre de classes en adéquation avec l'objectif ii) ;
- regrouper les variables d'entrée donne une information supplémentaire utile à l'analyse de données.

En revanche et en toute honnêteté, la SMDA a les inconvénients que nous listons ci-dessous.

- Le clustering avec la CAH est coûteux en temps de calcul. En effet, le clustering de variables d'entrée avec la CAH a une complexité quadratique en nombre de variables.
- Le clustering de variables d'entrée est fait indépendamment de la prédiction. Dans le cadre de la régression linéaire, une littérature abondante sur le clustering de variables est disponible (Park et al., 2007 ; Bondell and Reich, 2008 ; Bühlmann et al., 2013 ; Sharma et al., 2013 ; Witten et al., 2014 ; Reid and Tibshirani, 2016). Nous n'allons pas nous étendre sur ces méthodes mais il est intéressant de noter que certaines d'entre elles proposent que le clustering ne soit pas fait indépendamment de la prédiction (Bondell and Reich, 2008 ; Sharma et al., 2013 ; Witten et al., 2014).
- La réduction de dimension via le clustering entraîne une perte d'information. Ainsi, par exemple, des variables d'entrée d'un même cluster ayant la même corrélation à la variable synthétique correspondante ont la même importance estimée par la SMDA. Ceci est une conséquence implicite du point précédent.

Donnons des perspectives et des possibles généralisations de la méthode :

- le clustering de variables d'entrée peut se faire avec la méthode des K-means comme précisé dans Chavent et al. (2011, 2021a) ;
- la méthode de clustering de variables d'entrée est applicable aux données mixtes Chavent et al. (2011, 2021a) ;
- le calcul de la SMDA peut se faire avec n'importe quel modèle de régression et de classification. Des extensions à des cas non supervisés sont aussi envisageables ;
- un clustering de variables d'entrée prenant en compte les dépendances non linéaires est aussi envisageable et dépend du choix de la dissimilarité.

Deuxième partie

Applications industrielles

7

Analyse et correction des données de bancs d'essais

7.1 Description des données et du problème . . . 102
7.2 Estimer et retirer la tendance de production . . . 106
7.3 Détection des biais bancs . . . 108
7.4 Correction des biais des équipements de bancs . . . 110
7.5 Vérification de la correction . . . 112
7.6 Conclusion . . . 114

7.1 Description des données et du problème

PRÉSENTATION, MOTIVATIONS ET OBJECTIFS Les tests de réception sur bancs d'essai sont obligatoires avant de livrer des moteurs d'avions car les mesures effectuées pendant les essais déterminent si le moteur répond bien aux exigences. Ils sont importants tant pour l'avionneur que pour les équipes de Safran Aircraft Engines. Pour ces dernières, cela leur permet d'avoir une compréhension complète du comportement de chaque moteur. Ces tests produisent de nombreuses mesures et leur analyse est une tâche difficile.

L'objectif de l'étude présentée dans ce chapitre est de détecter d'éventuels biais introduits par l'instrumentation permettant les mesures sur les moteurs, notamment les équipements de bancs, lors des essais. Ensuite nous proposons une méthode de correction de ces biais. Enfin, il faudra vérifier que les données sont effectivement corrigées, c'est-à-dire que les mesures ne dépendent plus du banc sur lesquelles elles ont été effectuées. Cette analyse est la suite des travaux menés par Mohammed Meqqadmi, Pierre-Etienne Mosser, Thierry Brichler et Jérôme Lacaille et présentés dans Meqqadmi et al. (2017).

DESCRIPTION DES DONNÉES Commençons par présenter les caractéristiques principales des données de bancs d'essai. Pour cette étude, nous avons à notre disposition :

- $n = 591$ moteurs ;
- Une variable $\mathbf{t} \in \mathbb{R}^{591}$ pour identifier les observations,
- 14 variables numériques mesurées lors des essais.
  - XN12R : Régime tour plan 12 réduit (RPM).
  - XN25R : Régime tour plan 25 réduit (RPM).
  - WF36 : Débit massique plan 36 (lbm/hr).
  - FNIN1 : Poussée / performance (lbf).
  - W2AR : Débit massique plan 2A réduit (lbm/s).

– CV19 : Coefficient de vitesse de la tuyère secondaire.
– CF18 : Coefficient de débit de la tuyère secondaire.
– P15Q12 : Rapport des pressions entre le plan 15 et le plan 12.
– P18QSC : Pression plan 18 normalisée par les conditions standards).
– T25 : Température au plan 25 (R).
– T49C : Température plan 49 (°C) (où aussi appelé *exhausting gaz temp* (EGT) en anglais).
– T3 : Température au plan 3 (R).
– P3 : Pression plan 3 (psia).
– DRDT48INPB : Ecart EGT calculé et mesuré (R).

Un schéma expliquant où sont mesurés les variables est illustré Figure 8.1 dans le Chapitre 8 Section 8.3. Quand il y a lieu, l'unité de mesure est donnée entre parenthèses. Le "R" qui suit les noms signifie réduit et normalisé aux conditions de fonctionnement (pression et température). Les variables considérées sont des variables mesurées sur le moteur et certaines sont corrigées par un modèle thermodynamique en fonction du contexte météorologique.

- 4 variables catégorielles qui définissent les *équipements de bancs*. Elles désignent le matériel utilisé pour effectuer les mesures sur le moteur. Parmi elles, il y a la variable *bancs* qui est un équipement particulier des équipements de bancs. Chaque équipement de bancs est donc représenté par une variable catégorielle ayant plusieurs *modalités*, que l'on appellera aussi *classes*. Par exemple, dans cette étude, l'équipement bancs à deux modalités (classes) 1 et 2 qui sont les bancs utilisés a Safran-Villaroche (d'autres existent par exemple à Cincinnati). Les quatre variables catégorielles d'équipements de bancs sont :

  1. les nacelles que l'on note $na \in \{1, 2, 3\}$;
  2. les bancs que l'on note $ba \in \{1, 2\}$;
  3. les buses d'air que l'on note $bu \in \{1, 2, 3\}$;
  4. les tuyères primaires que l'on note $tu \in \{1, 2, 3\}$;

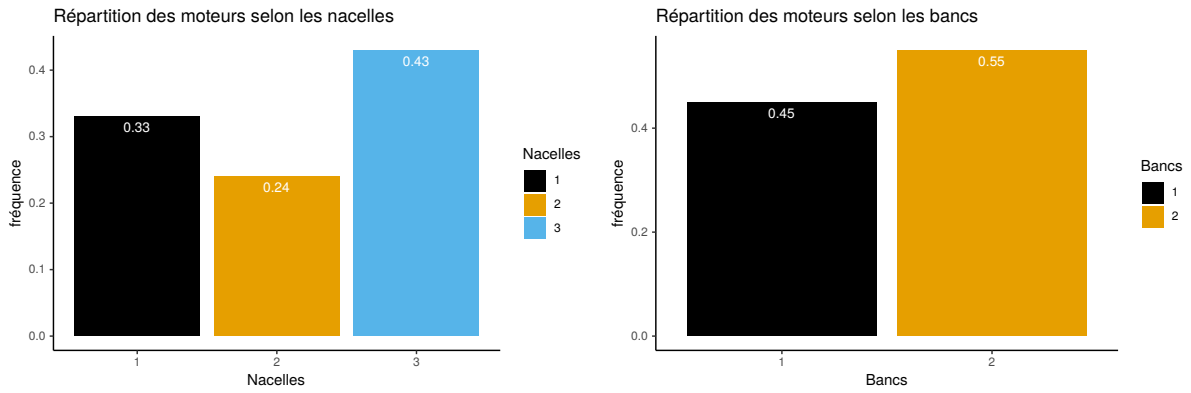
  La répartition des variables catégorielles décrivant les équipements de bancs est donnée dans la Figure 7.1. Le nombre de moteurs dans chaque classe est suffisant pour réaliser des estimations statistiques.

- Lors d'un essai sur banc, l'opérateur fixe certaines conditions de fonctionnement de manière à pouvoir comparer les différents moteurs en les plaçant dans des conditions identiques. Pour cela, il fixe un niveau de poussée ou de puissance fixe pour lequel tous les moteurs sont testés et des mesures sont prises, c'est ce qu'on appelle un *point stabilisé*. On prend en compte ici 6 points stabilisés. On les considère dans l'ordre croissant, c'est-à-dire que le point stabilisé 1 correspond au point de puissance le plus bas. On accole le suffixe k au nom de la variable pour indiquer en quel point stabilisé elle a été mesurée. Par exemple, XN12R_1 est le régime tour plan 12 réduit mesuré au premier point stabilisé .

Nous disposons donc de 84 (6×14) mesures numériques pour un même moteur, chaque variable étant mesurée 6 fois.

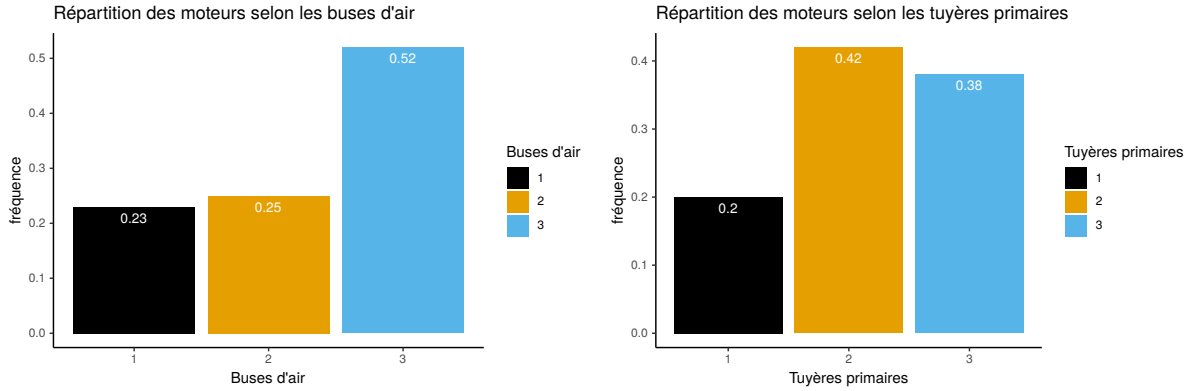
ANALYSE DESCRIPTIVE DES DONNÉES Pour chaque partition associée à l'une des variables catégorielles (équipement de bancs), on calcule les 4 variances inter-classes des 84 variables numériques. On ordonne la liste des noms de variables selon les valeurs décroissantes de la variance inter-classes associée aux classes définies par les modalités de la variable catégorielle nacelles, et on représente sur la Figure 7.2 les 4 variances inter-classes de toutes les variables. Nous pouvons constater que la variable nacelle présente le plus grand biais, comme s'y attendaient les experts métier de Safran Aircraft Engines, au moins pour une partie des variables. En revanche, la présence de biais sur les autres équipements de bancs n'était pas connue. De plus, on observe que les différents équipements de bancs induisent des biais sur des groupes de variables différents.

En outre, nous nous sommes aperçus qu'il n'y avait pas simplement un biais, mais que l'on observe aussi une tendance au cours du temps, comme pour la variable CF18_P6, représentée sur la Figure 7.3. Notons aussi qu'il existe des périodes pour lesquelles certaines modalités des équipements de bancs ne sont pas utilisées. Par exemple, pour la variable CF18_P6 (Figure 7.3), la Nacelle 9 n'est pas utilisée en 2018. Cette information sera importante par la suite.



(a) Nacelles

(b) Bancs



(c) Buses d'air

(d) Tuyères primaires

Figure 7.1 : Les quatre graphiques représentent les distributions des fréquences relatives des nacelles (a), bancs (b), buses d'air (c) et tuyères primaires (d).

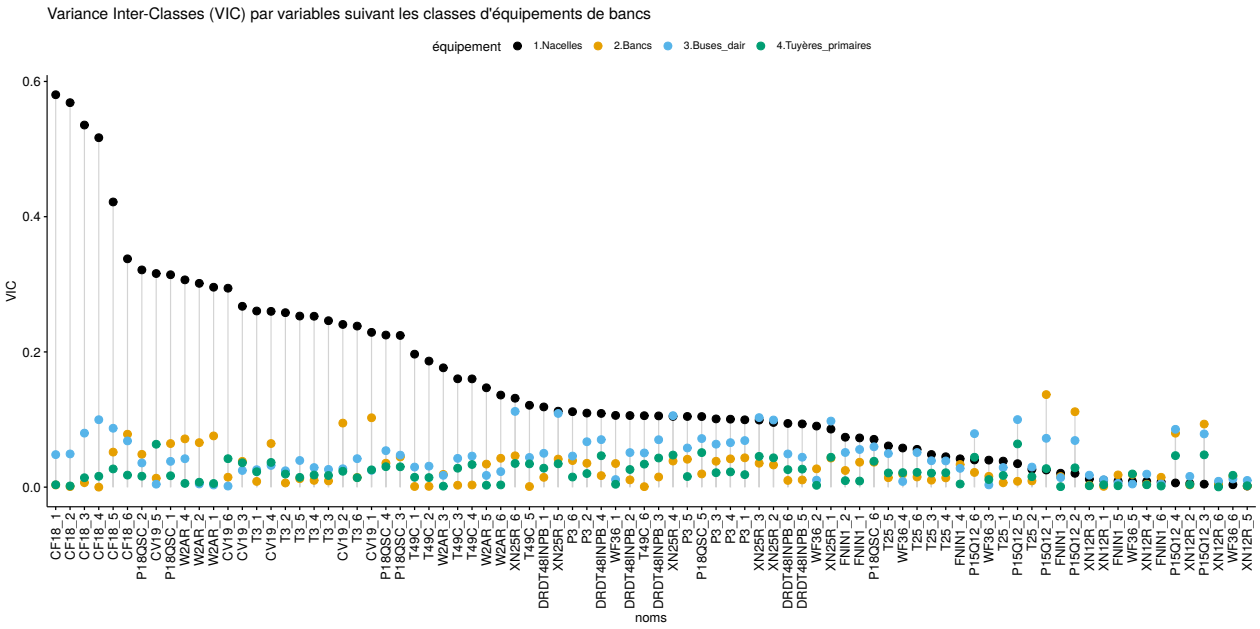


Figure 7.2 : Le graphique représente les Variances Inter-Classes (VIC) par variable suivant les classes définies par les équipements de bancs.

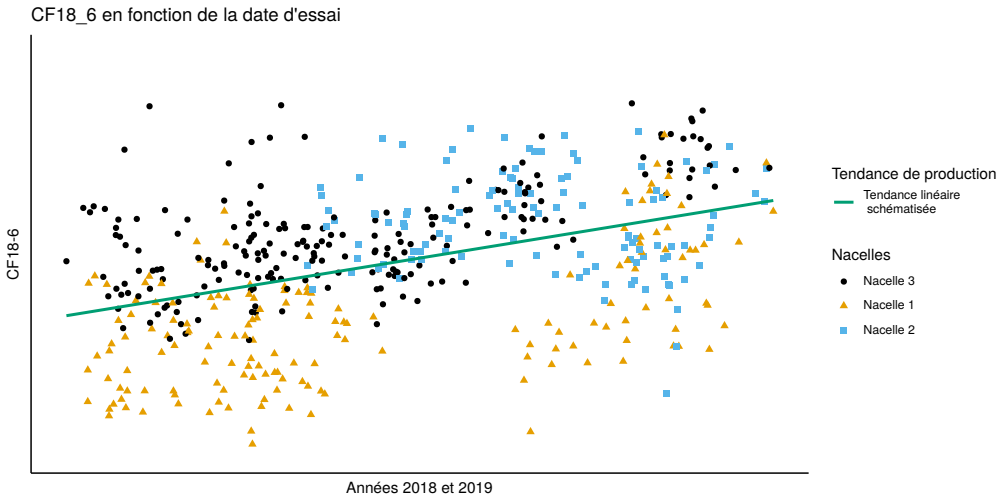


Figure 7.3 : Le graphique représente la variable CF18 au point stabilisé 6 en fonction des dates des essais et les couleurs des observations indiquent sur quelles nacelles les moteurs ont été testés. La tendance de production est schématisée de manière linéaire.

HYPOTHÈSES SUR LES DONNÉES Plusieurs hypothèses nous sont données au préalable par les équipes de Safran Aircraft Engines et elles découlent de constatations empiriques faites par les experts métier. Elles nous guideront pour la suite de l'étude.

- Les biais des équipements de bancs, aussi appelé *biais bancs*, sont des effets univariés et additifs : les données ont subi une simple translation dépendant de l'équipement de bancs. Les mesures des moteurs sont décalées d'un coefficient à estimer (voir Section 1.2.2.c pour plus d'informations).
- Les biais dépendent du temps : la valeur du biais est différente suivant la date de l'essai. Cela est dû à des effets d'usure des équipements que l'on appelle *dérive* des équipements. De plus, l'utilisation des équipements nécessite une maintenance et celle-ci impacte aussi les résultats des tests. Lorsqu'une modalité d'équipement de bancs est en maintenance, elle n'est pas utilisée pendant plusieurs mois. Les effets de maintenance sont à différencier de la tendance de production qui est un effet dû à la fabrication, à la réalisation, à la conception du moteur et non pas à un défaut dans la mesure des données.
- Il n'y a pas d'interaction entre les équipements de bancs : les équipements de bancs faussent, indépendamment les uns des autres, les mesures effectuées. La présence d'interactions entre équipements est un phénomène incohérent du point de vue métier. D'abord, les modalités d'équipements de bancs sont utilisées les unes avec les autres en fonction de leur disponibilité. Ensuite, si un effet d'interaction est visible, il est dû à la temporalité des biais et au fait que les modalités d'équipements ne sont pas utilisées sur les mêmes périodes. Ainsi, si un moteur est testé sur le banc 1 avec la nacelle 1, il suffit d'appliquer les coefficients de correction estimés sur le banc 1 et la nacelle 1 indépendamment, et il n'y a pas besoin d'estimer un nouveau coefficient pour ce sous-groupe (banc 1, nacelle 1).

LE MODÈLE THÉORIQUE Des hypothèses qui sont données, nous pouvons déduire un modèle théorique. Formellement, nous disposons d'une matrice de données $\mathbf{X} \in \mathbb{R}^{n \times p}$ avec $n = 591$ observations et $p = 84$ variables et nous notons $\mathbf{t} \in \mathbb{R}^n$ la variable indiquant la date de l'essai pour chaque moteur ($\mathbf{t} \notin \mathbf{X}$). Nous disposons aussi de quatre équipements de bancs **na**, **ba**, **bu**, **tu** qui sont respectivement les nacelles, les bancs, les buses d'air et les tuyères primaires. Le modèle théorique peut s'écrire de la manière suivante :

$$\mathbf{x}_i = \boldsymbol{\mu}_i + f(t) + \alpha(na_i, t) + \beta(ba_i, t) + \gamma(bu_i, t) + \delta(tu_i, t), \tag{7.1}$$

où :

- $\boldsymbol{\mu}_i \in \mathbb{R}^p$ est une constante représentant les p valeurs réelles, non biaisées, des tests de réception pour le moteur i ;
- $f(t) \in \mathbb{R}^p$ est l'effet fixe de la tendance de production (qui n'est pas un biais de mesure) et il dépend uniquement de la date de l'essai (t) et cet effet a p valeurs, une pour chaque variable mesurée ;
- $\alpha(na_i, t) \in \mathbb{R}^p$ est l'effet fixe des nacelles, il dépend de la nacelle utilisée pour le moteur i (na_i), de la date de l'essai (t) et cet effet a p valeurs, une pour chaque variable mesurée ;

- $\beta(ba_i, t) \in \mathbb{R}^p$ est l'effet fixe des bancs, il dépend du banc utilisé pour le moteur i (ba_i), de la date de l'essai (t) et cet effet a p valeurs, une pour chaque variable mesurée ;
- $\gamma(bu_i, t) \in \mathbb{R}^p$ est l'effet fixe des buses d'air, il dépend de la buse d'air utilisée pour le moteur i (bu_i), de la date de l'essai (t) et cet effet a p valeurs, une pour chaque variable mesurée ;
- $\delta(tu_i, t) \in \mathbb{R}^p$ est l'effet fixe des tuyères, il dépend de la tuyère utilisée pour le moteur i (tu_i), de la date de l'essai (t) et cet effet a p valeurs, une pour chaque variable mesurée.

Les effets fixes $\alpha, \beta, \gamma, \delta$ pourraient être décomposés en deux parties, une constante et un effet temporel dû notamment à la dérive des équipements (et aussi aux maintenances). Mais statistiquement cela revient au même d'estimer les deux en même temps et en pratique cela ne sert à rien d'isoler la fonction représentant la dérive des équipements.

MÉTHODOLOGIE GLOBALE La méthodologie consiste en trois étapes, dont la première est de détecter la présence de biais dû aux équipements de bancs. En théorie, on suppose qu'il y en a un, mais on aimerait le quantifier pour savoir s'il est utile d'effectuer une correction. En d'autres termes, on aimerait savoir si $\mathbf{x}_i \approx \boldsymbol{\mu}_i + f(t)$ pour tous les moteurs $i = 1, ..., n$. Il faut aussi réussir à communiquer les résultats de manière simple et donc il nous faut une métrique claire et transparente. Ainsi, nous voulons réduire le biais dans les données à un score qui nous indiquerait le degré de séparation des classes formées par les équipements. L'analyse univariée nous a donné quelques intuitions, mais la variance expliquée par chacun des équipements de bancs est une mesure qui peut être difficile à interpréter notamment par des ingénieurs métier. Nous proposons de prédire, tour à tour, les modalités de chaque équipement de bancs à partir des mesures effectuées ($\mathbf{X}$) à l'aide de (quatre) modèles de classification supervisée. La qualité de la classification indiquera le degré de dépendance entre les mesures et les équipements. Malheureusement, à certaines dates, certaines modalités d'équipements ne sont pas utilisées, comme c'est le cas de la nacelle 2 en 2018 (voir Figure 7.3). Ainsi, il y a des différences de distributions qui sont dues aux effets de la tendance de production et à l'utilisation des équipements à des périodes différentes. Il faut donc estimer et retirer l'effet de la tendance de production $f(t)$ des données observées avant d'apprendre un modèle de classification supervisée.

La deuxième étape consiste à estimer $\alpha, \beta, \gamma, \delta$ pour tout $i = 1, ..., n$ et corriger les données.

Enfin, la dernière étape consiste à vérifier si les biais sont effectivement corrigés, et cela pourra se faire avec un modèle de classification supervisée appris sur les données corrigées et sans la tendance de production.

Ainsi, la méthodologie globale de correction, de détection et de vérification est la suivante :

1. estimer et retirer la tendance de production ;
2. détecter les biais des équipements de bancs de manière supervisée ;
3. estimer et corriger les biais des équipements de bancs ;
4. détecter d'éventuels biais des équipements de bancs de manière supervisée sur les données corrigées ;
5. rajouter la tendance de production aux données corrigées.

7.2 Estimer et retirer la tendance de production

PRÉSENTATION Empiriquement on constate l'existence d'une tendance de production qui est due à des changements dans la fabrication, la conception et la réalisation des moteurs. Elle est indépendante des équipements de bancs et concerne tous les moteurs produits. Pour la suite de l'étude, il faut donc séparer l'étude des biais de l'étude de la tendance de production en retirant la tendance de production des données.

CLASSES DE RÉFÉRENCES Pour retirer la tendance de production, il faut choisir un quadruplet de modalités d'équipement de référence. En effet, si toutes les modalités d'équipements sont biaisées, la moyenne globale des données peut avoir été modifiée et il en va de même pour la tendance. C'est pourquoi, avec les ingénieurs métier, nous fixons pour chaque équipement de bancs une classe de référence. Nous listons ci-dessous les modalités correspondantes :

1. nacelle 3 ;
2. banc 2 ;
3. buse d'air 3 ;

4. tuyère primaire 2.

On remarquera aussi que ce sont les classes ayant le plus d'observations par équipement. Ainsi, il y a 66 moteurs réunissant ces quatre caractéristiques, c'est-à-dire que l'on note :

$$D^\star = \{(\mathbf{x}_i, t_i) : i \in \{1, ..., n\},\ na_i = 3,\ ba_i = 2,\ bu_i = 3,\ tu_i = 2\},$$

l'ensemble des observations non biaisées et on a $\text{card}(D^\star) = 66$.

REMARQUE SUR LA MÉTHODOLOGIE Si l'on souhaite simplement détecter les biais des équipements de bancs, il n'est pas nécessaire de disposer d'un ensemble de référence. Celui-ci est indispensable uniquement si l'on désire corriger les biais. En effet, on peut d'abord supposer qu'il n'y a pas de biais, puis on peut estimer la tendance de production sur l'ensemble des données et la retirer et appliquer la méthodologie de détection. Néanmoins, dans un souci de clarté, nous décrivons directement la méthodologie dans le cas où il faut corriger les biais.

La méthodologie de correction des biais des équipements n'est pas directement généralisable car il est nécessaire de disposer d'un ensemble de référence $D^\star$. Les experts métier peuvent, avoir accès à des mesures d'un moteur étalon, qu'ils considèrent comme la norme, tous les semestres ou dans le meilleur des cas tous les mois. Dans ce cas précis (accès aux mesures d'un moteur étalon), l'hypothèse que la tendance de production est linéaire par morceaux (entre les points de références) peut être faite. Cette hypothèse n'est pas trop restrictive car une année relève du temps court pour ce type de production industrielle, et les effets de productions sont lisses, réguliers et pas abrupts. En revanche pour notre étude, nous n'avons pas besoin de cette hypothèse car nous avons pu définir l'ensemble de référence $D^\star$.

MÉTHODOLOGIE DE CORRECTION La tendance de production est une dépendance non linéaire entre la date $\mathbf{t}$ et les variables $\mathbf{x^j} \in \mathbf{X}$, $j = 1, ..., 84$. Pour estimer cette relation, nous utilisons des fonctions de spline cubique lissées ou *cubic smoothing spline*. Une spline d'ordre k est une fonction polynomiale par morceaux de degré k, qui est continue et a des dérivées continues d'ordres $1, \ldots, k-1$ à chacun de ses nœuds (qui définissent les morceaux), et le cas cubique correspond à $k = 3$. Donc pour chaque variable $\mathbf{x}^j$, on estime une spline cubique lissées pour prédire $\mathbf{x}^j$ à partir de la variable indiquant la date de l'essai $\mathbf{t}$ qui est le prédicteur. Précisemment, il s'agit d'optimiser le critère suivant :

$$\underset{f_j \in \mathcal{F}}{\text{minimiser}} \sum_{i=1}^{n} (x_i^j - f_j(t_i))^2 + \lambda \int f_j^{''}(h)^2 dh, \tag{7.2}$$

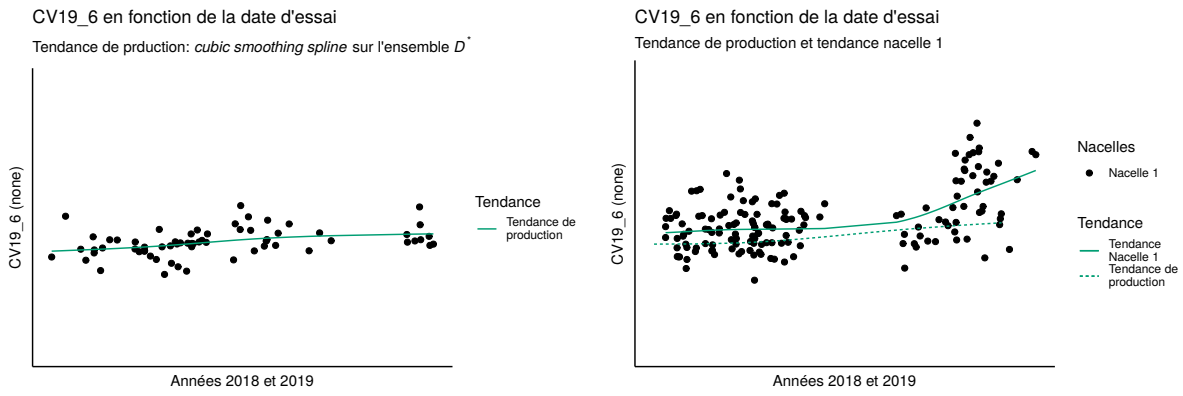
- où $\mathcal{F}$ est l'ensemble des fonctions splines de degré 3 ;
- f_j est la fonction à estimer parmi l'ensemble des fonctions $\mathcal{F}$ et $f_j^{''}$ sa dérivée seconde ;
- $\lambda \geqslant 0$ est un paramètre de lissage.

Le paramètre de lissage λ contrôle le degré d'ondulation de la fonction. Il est choisi par validation croisée dans notre cas. Lorsque le paramètre λ vaut 0, la fonction f n'est pas contrainte et donc non lisse. À l'inverse, lorsque le paramètre λ tend vers l'infini, f est contrainte et linéaire. Nous avons fait ce choix de modèle car ce type de méthode est très utilisé en dimension 1 pour ajuster une courbe à des données, nous n'avons pas d'a priori sur la forme de la tendance de production et il y a assez d'observations sur la période observée. Dans le cas contraire, on peut restreindre le nombre de nœuds ou même les choisir de manière à ce qu'ils soient équidistants sur la période.

Après avoir estimé le terme $\hat{f}_j(t_i)$,on le retranche de chaque variable pour chaque individus, ce qui revient à recentrer les variables en prenant en compte la date de production. Les étapes de la méthodologie sont résumées ci-dessous :

1. on estime la tendance de production pour chaque $j = 1, ..., 84$ en optimisant le critère (7.2) sur l'ensemble $D^\star$;
2. on obtient pour chaque observation $i = 1, ..., 591$ la prédiction $\hat{f}_j(t_i)$;
3. on retire la tendance de production en calculant $z_i^j = x_i^j - \hat{f}_j(t_i)$, $\forall i = 1, ..., 591$, $\forall j = 1, ..., 84$.

On obtient une nouvelle matrice de données $\mathbf{Z}$ avec 591 observations décrites par 84 mesures pour lesquelles la tendance de production a été retirée. Désormais nous allons travailler uniquement sur cette matrice $\mathbf{Z}$ pour étudier les biais des équipements de bancs.



(a) Tendance de production

(b) Biais Nacelle 1

Figure 7.4 : Le graphique (a) représente la tendance de production estimée par spline cubique sur l'ensemble $D^\star$ et le graphique (b) cette même tendance mais aussi une tendance estimée sur la nacelle 1. Une fois la tendance de production retirée des mesures des individus de la nacelle 1, il reste un biais à corriger.

7.3 Détection des biais bancs

PRÉSENTATION Formellement, nous disposons d'une matrice de données $\mathbf{Z} \in \mathbb{R}^{n\times p}$ avec $n = 591$ observations et $p = 84$ variables pour lesquelles la tendance de production a été retirée. L'on rappelle que l'on dispose également d'équipements de bancs $\mathbf{na} \in \{1,2,3\}^n$, $\mathbf{ba} \in \{1,2\}^n$, $\mathbf{bu} \in \{1,2,3\}^n$, $\mathbf{tu} \in \{1,2,3\}^n$ étant respectivement les nacelles, les bancs, les buses d'air et les tuyères primaires. Nous voudrions savoir si les équipements de bancs ont une influence sur les valeurs de $\mathbf{Z}$ et nous voudrions aussi connaître le degré de cette dépendance. Nous cherchons par exemple alors à estimer $p(\mathbf{na}|\mathbf{Z})$, la probabilité que le moteur ait été testé sur chaque nacelle sachant qu'on a observé $\mathbf{Z}$. Ce problème se résume à un problème de classification supervisé. Prendre le problème dans ce sens peut paraître contre-intuitif. Il serait en effet logique d'expliquer les mesures à l'aide des équipements de bancs. Cela peut se faire avec un modèle ou avec une analyse de la variance (ANOVA). Nous développerons quelque peu ce point là dans la conclusion.

Dans cette optique nous allons utiliser deux modèles de classification : les forêts aléatoires et l'analyse discriminante linéaire (LDA). La LDA est un modèle linéaire qui cherche à estimer la meilleure séparation linéaire entre les populations de moteurs définies par chaque équipement de bancs. Ainsi, si l'hypothèse des biais additifs n'est pas vérifiée, on pourrait s'attendre à ce que des modèles non-linéaires tels que les forêts aléatoires fonctionnent mieux. En revanche, les biais peuvent ne pas être additifs et il peut quand même y avoir des classes linéairement séparables dans les cas simples.

MÉTHODOLOGIE Nous allons retirer la variable indiquant la date de l'essai. Cette variable est trop discriminante et elle pourrait fausser les résultats (s'il y a des biais qui dépendent du temps). Pour évaluer les modèles, on découpe l'ensemble des données aléatoirement plusieurs fois, en un ensemble d'entraînement contenant 70% des observations et un ensemble de test contenant 30% des observations, grâce à une fonction qui équilibre les distributions des classes à l'intérieur des ensembles pour chaque équipement de bancs. Ainsi, pour les nacelles :

- soit $\{(\mathbf{z}_i, na_i)\}_{1\leq i\leq n}$ l'ensemble de données ;
- soit E et T l'ensemble des observations d'entraînement et de test de tailles n_E et n_T telles que $n_E + n_T = n$, $n_E/n \approx 0.7$, $n_T/n \approx 0.3$;
- soit $n(\mathbf{na} = k)$ le nombre de moteurs qui ont été testés sur la nacelle k, soit $n_E(\mathbf{na} = k)$ le nombre de moteurs qui ont été testés sur la nacelle k dans l'ensemble E et de même pour $n_T(\mathbf{na} = k)$, alors :

$$\frac{n_E(\mathbf{na} = k)}{n_E} = \frac{n_T(\mathbf{na} = k)}{n_T} = \frac{n(\mathbf{na} = k)}{n}.$$

Les mêmes calculs sont effectués pour les autres équipements.

Deux mesures de qualité de nos modèles sont utilisées : l'*accuracy* qui est le taux d'erreur standard c'est-à-dire le nombre d'observations bien classées divisé par le nombre total d'observations, et une version multi-classes de *Area Under the Receiver Operating Characteristic Cuve* soit l'aire sous la courbe ROC que l'on

nomme plus simplement multi-classes AUC (Hand and Till, 2001). On rappelle que l'AUC pour $K = 2$ classes se calcule ainsi :

$$\text{AUC}(1|2) = \sum_{i,j:i\neq j} \mathbb{1}\big(p(1|\mathbf{z}_i) > p(1|\mathbf{z}_j)|y_i = 1, y_j = 2\big)/n(n-1),$$

où $p(k|\mathbf{z}_i)$ est la probabilité estimée par le modèle que l'observation $\mathbf{z}_i$ appartienne à la classe k pour $k = 1, 2$. Notons que dans ce cas ($K = 2$) $\text{AUC}(1|2) = \text{AUC}(2|1)$. Dans Hand and Till (2001), les auteurs généralisent l'AUC au cas multi-classes ($K > 2$). Pour ce faire, ils posent :

$$\text{AUC}(k|k') = \sum_{i,j:i\neq j} \mathbb{1}\big(p(k|\mathbf{z}_i) > p(k|\mathbf{z}_j)|y_i = k, y_j = k'\big)/n(n-1).$$

En revanche lorsque $K > 2$, nous obtenons que $\text{AUC}(k|k') \neq \text{AUC}(k'|k)$. Ainsi, les auteurs proposent de définir $\text{AUC}(k, k') = \big(\text{AUC}(k|k') + \text{AUC}(k'|k)\big)/2$ et :

$$\text{M-AUC} = \frac{2}{K(K-1)} \sum_{k<k'} \text{AUC}(k, k').$$

Ainsi la M-AUC peut s'interpréter comme la moyenne des AUC sur les classes prises deux à deux.

Les algorithmes LDA et forêts aléatoires sont entraînés respectivement avec les packages `R MASS` et `ranger`, où pour ce dernier les paramètres sont laissés par défaut et le nombre d'arbres est fixé à 1000. Le découpage (entraînement / test) et l'apprentissage sont répétés cent fois et les résultats agrégés sont présentés dans la Table 7.1.

Table 7.1 : *accuracy* et M-AUC pour la LDA et les forêts aléatoires avec 1000 arbres sur 100 découpages entraînement/test. Les moyennes et les écarts-types de l'*accuracy* et de la M-AUC sont donnés pour les deux modèles.

		accuracy		multi-classes AUC	
	K^q	moyenne	écarts-types	moyenne	écarts-types
forêts aléatoires - nacelles ($q = 1$)	3	0.9068	0.0272	0.9721	0.0111
forêts aléatoires - bancs ($q = 2$)	2	0.8589	0.0254	0.9378	0.0212
forêts aléatoires - buses d'air ($q = 3$)	3	0.5366	0.0451	0.7318	0.0310
forêts aléatoires - tuyères primaires ($q = 4$)	3	0.4372	0.0227	0.6404	0.0262
LDA - nacelles ($q = 1$)	3	0.9959	0.0038	0.9983	0.0031
LDA - bancs ($q = 2$)	2	0.8384	0.0371	0.9038	0.0210
LDA - buses d'air ($q = 3$)	3	0.6303	0.0572	0.8026	0.0369
LDA - tuyères primaires ($q = 4$)	3	0.4359	0.0255	0.6237	0.0132

Les résultats montrent qu'il existe bel et bien un lien entre les équipements de bancs et les mesures des moteurs. La LDA obtient des résultats équivalents ou meilleurs que ceux obtenus par les forêts aléatoires suggérant que les classes sont linéairement séparables et de covariance commune. L'avantage de l'utilisation du M-AUC par rapport à l'*accuracy* se trouve dans l'interprétation du score. En effet, pour un problème de classification binaire en deux classes de taille égale, une accuracy de 0.5 correspond au minimum soit le plus mauvais modèle, mais cela n'est plus vrai lorsque $K > 2$ et/ou lorsque les classes ne sont pas de tailles égales. En outre, pour la M-AUC, 0.5 correspond au minimum pour n'importe quelles valeurs de $K > 2$ et lorsque les tailles des classes ne sont pas trop déséquilibrées (ce qui est notre cas). De plus, lorsque M-AUC>0.6, la classification est considérée comme acceptable.

La LDA cherche un espace d'au plus $K - 1$ dimensions qui permet de séparer linéairement les K classes à prédire. Donc en plus de la prédiction, cette méthode offre une représentation des classes dans le sous-espace dit discriminant. Des représentations des classes, formées par les variables bancs et buses d'air, sur les axes discriminants sont représentées sur la Figure 7.5.

IMPORTANCE DES VARIABLES DES MODÈLES LDA ET FORÊTS ALÉATOIRES Nous cherchons à calculer l'importance des variables de $\mathbf{Z}$ pour les tâches de classification supervisée décrites ci-dessus. Cela nous permet de comprendre quelles sont les variables les plus biaisées. Pour éviter la multiplication des résultats, nous présentons uniquement le cas où nous prédisons la variable bancs.

Tout d'abord avec la LDA, il est possible de comprendre l'importance des variables en observant leurs contributions aux axes. Cela est donné par le carré de la corrélation entre les axes et les variables. Dans la prédiction de la variable bancs, nous n'avons qu'un seul axe discriminant.

Ensuite, nous allons appliquer la méthode présentée dans le Chapitre 6 dans le cadre de la classification. On construit un clustering hiérarchique des variables de $\mathbf{Z}$ et pour chaque partition obtenue, on entraîne une

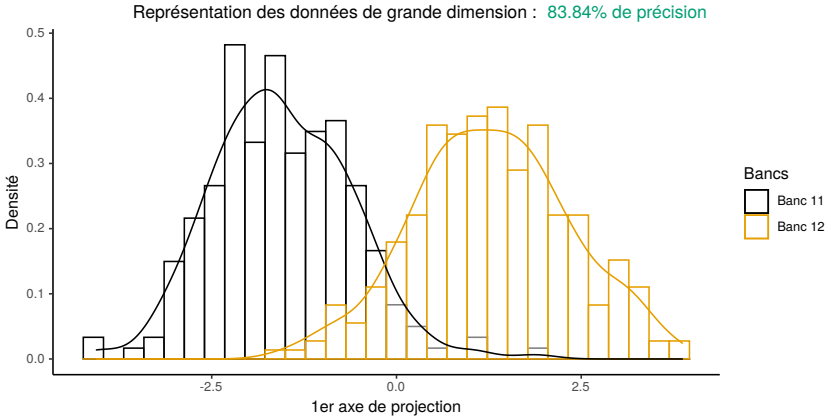


Figure 7.5 : Le graphique représente la densité des données projetées sur le premier axe discriminant. Les classes sont bien séparées.

forêt aléatoire. Le dendrogramme du clustering est présenté sur la Figure 7.6. On observe que les variables sont regroupées par point stabilisé (par exemple {CF18_1,...,CF18_6}) ce qui correspond à la structure en corrélation des données. En effet, les mesures sont assez similaires d'un point stabilisé à un autre.

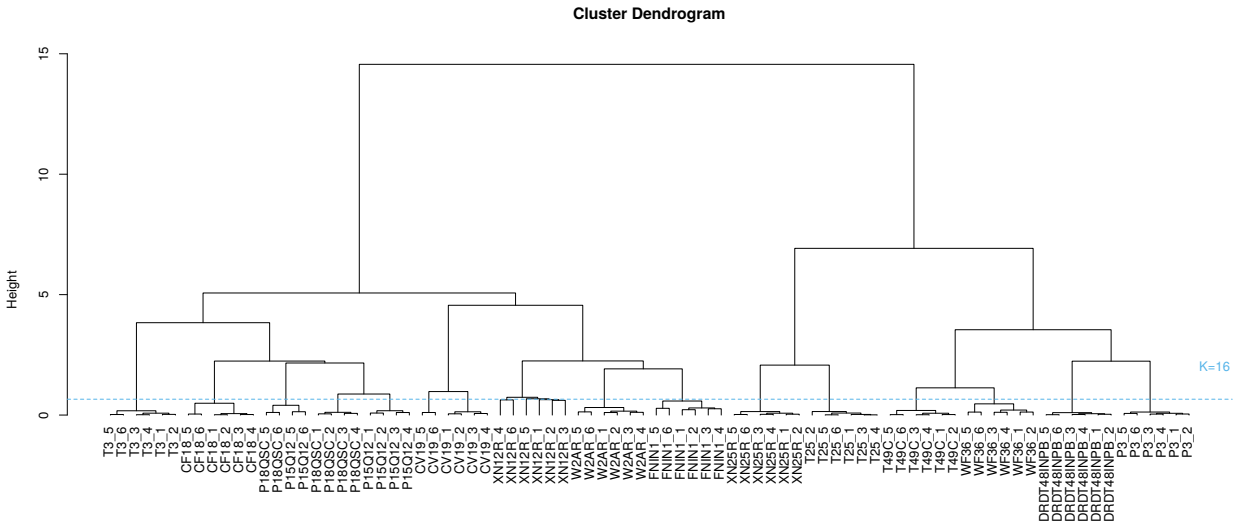


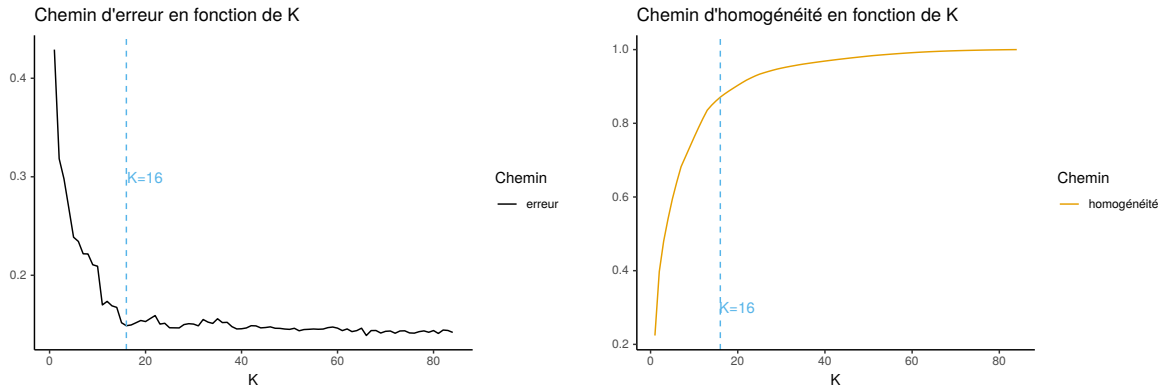
Figure 7.6 : Le graphique représente le dendrogram obtenu à l'aide du package `R ClustOfVar`. Les variables sont regroupées par point stabilisé ce qui correspond bien à la structure en corrélation des données.

Nous affichons ensuite les chemins d'erreur et d'homogénéité en fonction de K sur la Figure 7.7. Après analyse des chemins, $K = 16$ semble un bon choix, car K est petit, l'erreur est faible et l'homogénéité semble atteindre un plateau.

Enfin, les résultats de la SMDA (pour $K = 16$) et la contribution à l'axe discriminant pour la LDA sont disponibles dans la Table 7.2. Pour la LDA, on observe beaucoup de valeurs nulles, ce qui ne correspond pas a ce qui a été observée dans les analyses descriptives de ce chapitre. Pour la SMDA, les valeurs des importances sont nettement moins diluées que pour la LDA et on observe une homogénéité des valeurs par groupe. Les résultats SMDA sont cohérents avec ceux observés Figure 7.2 c'est-à-dire que les groupes de variables CV19, P15Q12 et W2AR semblent les plus importants. Le clustering de variables n'a pas regroupé ensemble toutes les variables représentant les points stabilisés de CV19. Néanmoins dans l'ensemble la structure des données est bien visible et les résultats sont cohérents avec les analyses métier (Meqqadmi et al., 2017).

7.4 Correction des biais des équipements de bancs

Pour corriger les biais des équipements de bancs il faut utiliser les hypothèses que l'on a sur les biais. Pour rappel, les hypothèses sont que les biais sont additifs, les biais dépendent de la date d'essai des moteurs, les équipements de bancs sont indépendants.



(a) Chemin d'erreur

(b) Chemin d'homogénéité

Figure 7.7 : Le graphique (a) représente le chemin d'erreur en fonction de K et le graphique (b) représente le chemin d'homogénéité en fonction de K. $K = 16$ semble un bon choix, car K est petit, l'erreur est faible et l'homogénéité semble atteindre un plateau.

Table 7.2 : *accuracy* et M-AUC pour la LDA et les forêts aléatoires avec 1000 arbres sur 100 découpages entraînement/test. Les moyennes et les écarts-types de l'*accuracy* et de la M-AUC sont donnés pour les deux modèles.

Variables	SMDA	Contrib	Variables	SMDA	Contrib
CV19_1	1.00	0.10	WF36_6	0.20	0.00
CV19_2	1.00	0.17	P18QSC_1	0.20	0.57
CV19_3	0.99	0.07	P18QSC_2	0.20	0.59
CV19_4	0.98	0.28	P18QSC_3	0.19	1.00
P15Q12_1	0.48	0.43	P18QSC_4	0.19	0.51
P15Q12_2	0.48	0.26	FNIN1_6	0.19	0.00
P15Q12_3	0.48	0.89	DRDT48INPB_1	0.18	0.00
P15Q12_4	0.48	0.81	DRDT48INPB_2	0.18	0.00
P15Q12_5	0.43	0.13	DRDT48INPB_3	0.18	0.00
P18QSC_5	0.41	0.54	DRDT48INPB_4	0.18	0.00
P15Q12_6	0.41	0.18	DRDT48INPB_5	0.18	0.00
P18QSC_6	0.41	0.13	DRDT48INPB_6	0.17	0.00
W2AR_1	0.33	0.00	P3_1	0.15	0.00
W2AR_2	0.33	0.00	P3_2	0.15	0.00
W2AR_3	0.32	0.00	P3_3	0.15	0.00
W2AR_4	0.32	0.00	P3_4	0.15	0.00
CV19_5	0.32	0.34	P3_5	0.15	0.00
CV19_6	0.32	0.08	P3_6	0.14	0.00
W2AR_5	0.31	0.00	T49C_1	0.09	0.00
W2AR_6	0.30	0.00	T49C_2	0.09	0.00
CF18_1	0.28	0.01	T49C_3	0.09	0.00
CF18_2	0.27	0.08	T49C_4	0.09	0.00
CF18_3	0.26	0.01	T49C_5	0.09	0.00
CF18_4	0.26	0.03	T49C_6	0.09	0.00
CF18_5	0.25	0.07	T25_1	0.08	0.00
FNIN1_1	0.24	0.00	T25_2	0.08	0.00
WF36_1	0.24	0.00	T25_3	0.08	0.00
WF36_2	0.23	0.00	T25_4	0.08	0.00
FNIN1_2	0.23	0.00	T25_5	0.08	0.00
FNIN1_3	0.23	0.00	T25_6	0.07	0.00
FNIN1_4	0.23	0.00	T3_1	0.03	0.00
XN25R_1	0.22	0.00	T3_2	0.03	0.00
XN25R_2	0.22	0.00	T3_3	0.03	0.00
XN25R_3	0.22	0.00	T3_4	0.03	0.00
XN25R_4	0.22	0.00	T3_5	0.03	0.00
CF18_6	0.22	0.01	T3_6	0.03	0.00
XN25R_5	0.22	0.00	XN12R_1	0.02	0.00
WF36_3	0.21	0.00	XN12R_2	0.02	0.00
WF36_4	0.21	0.00	XN12R_3	0.02	0.00
WF36_5	0.21	0.00	XN12R_4	0.02	0.00
XN25R_6	0.21	0.00	XN12R_5	0.02	0.00
FNIN1_5	0.21	0.00	XN12R_6	0.01	0.00

Les causes de la dépendance des biais aux dates d'essais ont déjà été abordées et il s'agit des effets de maintenances et des effets de dérive. On peut noter que les biais ne sont pas forcément des tendances linéaires. D'ailleurs on a bien souvent une dépendance non linéaire des variables par rapport aux dates des essais.

MÉTHODOLOGIE DE CORRECTION Nous décrivons ici comment estimer les effets fixes $\alpha, \beta, \gamma, \delta$ (voir Équation (7.1)) afin de les retrancher aux données. Nous allons procéder variable par variable. Les biais dépendent non linéairement des dates des essais, et donc nous allons devoir estimer cette relation. Pour corriger les variables, nous estimons pour chaque variable une *cubic smoothing spline* pour chaque modalité de chaque équipement de bancs avec comme prédicteur la date de production, et nous retranchons à chaque individu sa prédiction, ce qui revient à recentrer les variables par groupe, en prenant en compte la date de production. Les effets de maintenance sont des effets abrupts, mais lorsque les équipements sont en maintenance ils ne sont pas utilisés pendant quelques mois (au moins un) et donc les splines restent adaptés. Les étapes de la méthodologie de correction sont données ci-dessous.

1. On retire les biais nacelles :
   (a) pour $k = 1, 2, 3$ et pour chaque variable de $\mathbf{z}^j$, $j = 1, ..., 84$ on optimise le critère (7.2) sur l'ensemble suivant $\{(z_i^j, t_i, na_i) : i \in \{1, ..., n\},\ na_i = k\}$;
   (b) on obtient pour chaque observation $i = 1, ..., 591$ la prédiction $\hat{\alpha}(na_i, t)$, (où $\hat{\alpha} \in \mathcal{F}$ estimée via (7.2)) ;
   (c) on retire les biais en appliquant $v_i^{j,\mathbf{na}} = z_i^j - \hat{\alpha}(na_i, t)$, $\forall i = 1, ..., 591$, $\forall j = 1, ..., 84$.
   (d) on obtient $\mathbf{V^{na}}$ la matrice sans la tendance de production et sans les biais des nacelles.
2. On retire les biais bancs :
   (a) pour $k = 1, 2, 3$ et pour chaque variable de $\mathbf{v}^{j,\mathbf{na}}$, $j = 1, ..., 84$ on optimise le critère (7.2) sur l'ensemble suivant $\{(v_i^{j,\mathbf{na}}, t_i, na_i) : i \in \{1, ..., n\},\ na_i = k\}$;
   (b) on obtient pour chaque observation $i = 1, ..., 591$ la prédiction $\hat{\beta}(ba_i, t)$ (où $\hat{\beta} \in \mathcal{F}$ estimée via (7.2)) ;
   (c) on retire les biais en appliquant $v_i^{j,\mathbf{ba}} = v_i^{j,\mathbf{na}} - \hat{\beta}(ba_i, t)$, $\forall i = 1, ..., 591$, $\forall j = 1, ..., 84$.
   (d) on obtient $\mathbf{V^{ba}}$ la matrice sans la tendance de production et sans les biais des nacelles et sans les biais bancs.
3. On retire les biais des équipements buses d'air et tuyères primaire de la même manière.
   Ainsi on corrige les biais successivement et indépendamment des uns des autres. On obtient donc une nouvelle matrice de données $\mathbf{V^{tu}}$ avec $n = 591$ observations décrites par $p = 84$ mesures dont les biais ont été retirés.
4. On rajoute la tendance de production : en appliquant $u_i^j = v_i^{j,\mathbf{tu}} + \hat{f}_j(t_i)$, $\forall i = 1, ..., 591$, $\forall j = 1, ..., 84$ où $f_j^\star$ est estimé comme expliqué dans la Section 7.2. Finalement, nous obtenons une matrice débiaisée mais incluant la tendance de production $\mathbf{U} \in \mathbb{R}^{591 \times 84}$.

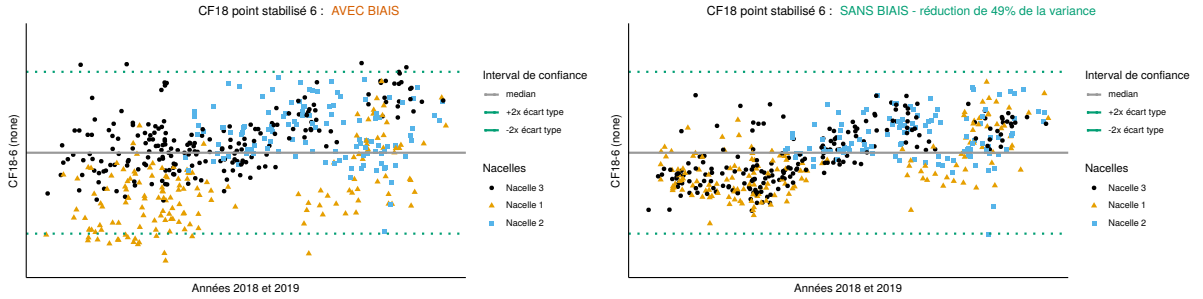
On notera que pour les observations de l'ensemble $D^\star$ (Section 7.2), cette opération a déjà été faite lorsque l'on a retiré la tendance de production mais dans un souci de clarté nous n'avons pas différencié les ensembles dans cette section.

Nous donnons une illustration de la correction en affichant les données obtenues après correction, c'est-à-dire la matrice $\mathbf{U}$. Les graphiques de la Figure 7.8 montrent la variable CF18-P6 avant et après correction. On observe que les 3 classes des nacelles ne se différencient plus, que la variance de la variable a été réduite et que la tendance de production a été conservée.

7.5 Vérification de la correction

Dans cette section, nous expliquons la procédure de vérification des corrections effectuées, c'est-à-dire lorsque les effets fixes $\alpha, \beta, \gamma, \delta$ (voir Équation (7.1)) ont bien été estimés et retirés des données. En outre, vérifier la correction permet en partie de vérifier les hypothèses.

La dérive des équipements est explicite dans les données et il est assuré que les effets fixes dépendent de la date de l'essai. En revanche les hypothèses d'additivité et d'indépendance des biais des équipements de bancs sont plus difficiles à confirmer. Nous pouvons tenter de vérifier a posteriori les hypothèses en utilisant



(a) Avant correction (matrice $\mathbf{X}$)

(b) Après correction (matrice $\mathbf{U}$)

Figure 7.8 : Le graphique (a) représente la variable CF18 au point stabilisé 6 avant que les corrections ne soient appliquées (matrice $\mathbf{X}$) et le graphique (b) représente cette même variable après correction (matrice $\mathbf{U}$). Les échelles sont les mêmes pour les deux graphiques et on observe une réduction de la variance de presque 50%.

la méthodologie de prédiction de la Section 7.3. En effet, nous considérons que l'hypothèse d'additivité est vérifiée si, après correction, les équipements de bancs ne sont plus distinguables au sens de la prédiction (comme vu précédemment). Nous considérons que l'hypothèse d'indépendance est vérifiée si, la correction d'un équipement de bancs de bancs n'est pas détériorée par la correction d'un autre équipement de bancs. En d'autres mots, si corriger successivement le biais des nacelles puis des bancs ne recrée pas un décalage sur les nacelles. Pour vérifier cela, il suffit entre chaque étape de la correction, d'observer l'*accuracy* et la M-AUC des modèles ré-entraînés sur les données corrigées. Si leurs performances sont modifiées au cours des corrections successives, on peut envisager une dépendance.

MÉTHODOLOGIE DE VÉRIFICATION Nous entraînons des forêts aléatoires à chaque étape de la correction. Les étapes de la méthodologie sont expliquées ci-dessous :

1. on part de la matrice $\mathbf{Z}$ c'est-à-dire la matrice sans la tendance de production mais dont les équipements de bancs n'ont pas été corrigés. Quatre modèles de forêt aléatoire sont utilisés pour prédire $\mathbf{na}, \mathbf{ba}$, $\mathbf{bu}$, $\mathbf{tu}$ à partir de $\mathbf{Z}$.

2. ensuite, on continue avec la matrice $\mathbf{V^{na}}$ c'est-à-dire la matrice sans la tendance de production où seuls les équipements de nacelles ont été corrigés. Quatre modèles de forêt aléatoire sont utilisés pour prédire $\mathbf{na}, \mathbf{ba}$, $\mathbf{bu}$, $\mathbf{tu}$ à partir de $\mathbf{V^{na}}$.

3. ensuite, on continue avec la matrice $\mathbf{V^{ba}}$ c'est-à-dire la matrice sans la tendance de production où seuls les équipements de nacelles et des bancs ont été corrigés successivement. Quatre modèles de forêt aléatoire sont utilisés pour prédire $\mathbf{na}, \mathbf{ba}$, $\mathbf{bu}$, $\mathbf{tu}$ à partir de $\mathbf{V^{ba}}$.

4. enfin, on répète l'étape précédente pour les matrices $\mathbf{V^{bu}}$ et $\mathbf{V^{tu}}$.

Finalement, on obtient 20 modèles de forêts aléatoires. On choisit les forêts aléatoires car cela permet de déterminer des séparations non linéaires entre les classes. En effet, comme nous avons centré les classes, elles ne sont plus linéairement séparables. En revanche elles peuvent être, au moins en partie, distinguables si la correction n'est pas appropriée. Nous ne construisons pas de modèle à partir de la matrice $\mathbf{U}$ (matrice débiaisée avec la tendance de production) car comme expliqué précédemment et comme on peut le voir sur la Figure 7.8, l'addition de la présence d'une tendance de production avec l'utilisation de modalités d'équipements de bancs à des périodes différentes implique des différences de distribution par modalité d'équipements de bancs, qui permettent de les distinguer.

Comme précédemment les modèles de forêts aléatoires sont entraînés à l'aide du package `R ranger` avec les paramètres laissés par défaut et en fixant le nombre d'arbres à 1000. Les résultats agrégés de l'*accuracy* et de la M-AUC sur 100 découpages entraînement/test sont présentés dans la Table 7.3. La Table 7.3 contient elle-même quatre (sous) tables. Chacune présente les résultats agrégés de modèles de forêt aléatoires, suivant un équipement de bancs à prédire, sur les ensembles $\mathbf{Z}, \mathbf{V^{na}}, \mathbf{V^{ba}}, \mathbf{V^{bu}}, \mathbf{V^{tu}}$. Ainsi, la première table montre les résultats agrégés des forêts aléatoires prédisant les nacelles sur $\mathbf{Z}, \mathbf{V^{na}}, \mathbf{V^{ba}}, \mathbf{V^{bu}}, \mathbf{V^{tu}}$. La ligne au milieu de chaque table indique le moment où la variable à prédire a été corrigée.

La Table 7.3 nous montre qu'après débiaisage, les modèles de forêts aléatoires ne distinguent plus les équipements de bancs, sauf dans une certaine mesure les nacelles. On remarque aussi que le débiaisage d'un équipement de bancs n'implique pas de différences dans les performances des modèles prédisant les autres équipements de bancs. Ce n'est pas une preuve formelle que les équipements de bancs aient des biais additifs

Table 7.3 : Résultats agrégés sur 100 découpages entraînement/test de modèles de forêts aléatoires sur des ensembles de données avant et après correction de chaque équipement de bancs. Chaque sous-table présente les résultats agrégés de modèles de forêt aléatoires, suivant un équipement de bancs à prédire, sur les matrices $\mathbf{Z}, \mathbf{V}^{\mathbf{na}}, \mathbf{V}^{\mathbf{ba}}, \mathbf{V}^{\mathbf{bu}}, \mathbf{V}^{\mathbf{tu}}$. Ainsi, la première table montre les résultats agrégés des forêts aléatoires prédisant les nacelles sur $\mathbf{Z}, \mathbf{V}^{\mathbf{na}}, \mathbf{V}^{\mathbf{ba}}, \mathbf{V}^{\mathbf{bu}}, \mathbf{V}^{\mathbf{tu}}$. La ligne au milieu de chaque table indique le moment où la variable à prédire a été corrigée. La moyenne et les écarts-types des *accuracy* et des M-AUC sur les 100 échantillons sont présentés.

variable à prédire : **na**, les nacelles

	accuracy		multi-classes AUC	
	moyenne	écarts-types	moyenne	écarts-types
$\mathbf{Z}$	0.9173	0.0223	0.9751	0.0093
$\mathbf{V}^{\mathbf{na}}$	0.4853	0.0293	0.6316	0.0305
$\mathbf{V}^{\mathbf{ba}}$	0.4616	0.0283	0.6155	0.0319
$\mathbf{V}^{\mathbf{bu}}$	0.4662	0.0353	0.6171	0.0346
$\mathbf{V}^{\mathbf{tu}}$	0.4671	0.0305	0.6166	0.0288

variable à prédire : **ba**, les bancs

	accuracy		multi-classes AUC	
	moyenne	écarts-types	moyenne	écarts-types
$\mathbf{Z}$	0.8482	0.0293	0.9267	0.0215
$\mathbf{V}^{\mathbf{na}}$	0.8254	0.0290	0.9050	0.0208
$\mathbf{V}^{\mathbf{ba}}$	0.5352	0.0338	0.5258	0.0432
$\mathbf{V}^{\mathbf{bu}}$	0.5407	0.0390	0.5318	0.0462
$\mathbf{V}^{\mathbf{tu}}$	0.5468	0.0305	0.5332	0.0389

variable à prédire : **bu**, les buses d'air

	accuracy		multi-classes AUC	
	moyenne	écarts-types	moyenne	écarts-types
$\mathbf{Z}$	0.5377	0.0322	0.7386	0.0236
$\mathbf{V}^{\mathbf{na}}$	0.5221	0.0215	0.7173	0.0307
$\mathbf{V}^{\mathbf{ba}}$	0.5287	0.0234	0.7053	0.0292
$\mathbf{V}^{\mathbf{bu}}$	0.4910	0.0220	0.5406	0.0261
$\mathbf{V}^{\mathbf{tu}}$	0.4959	0.0213	0.5416	0.0319

variable à prédire : **tu**, les tuyères primaires

	accuracy		multi-classes AUC	
	moyenne	écarts-types	moyenne	écarts-types
$\mathbf{Z}$	0.4523	0.0323	0.6487	0.0284
$\mathbf{V}^{\mathbf{na}}$	0.4406	0.0327	0.6255	0.0274
$\mathbf{V}^{\mathbf{ba}}$	0.4432	0.0331	0.6201	0.0313
$\mathbf{V}^{\mathbf{bu}}$	0.4495	0.0348	0.6209	0.0312
$\mathbf{V}^{\mathbf{tu}}$	0.4130	0.0312	0.5704	0.0301

ou qu'ils soient indépendants. Néanmoins, cela permet au moins de s'assurer que les classes d'équipements de bancs se confondent (conditionnellement au modèle utilisé).

De plus, cette analyse supervisée permet d'observer que les biais induits par les nacelles ne sont pas complètement corrigés. En effet une M-AUC>0.6 peut s'interpréter comme le fait que le résultat du modèle n'est pas aléatoire et donc qu'il existe, dans une certaine mesure, une séparation entre certaines observations des classes. On observe aussi que la correction des tuyères primaires n'implique qu'une faible réduction de la performance de nos modèles.

7.6 Conclusion

Dans ce chapitre, nous avons présenté une méthodologie de détection, de correction et de vérification de la correction, pour les biais induits par les équipements de bancs. Cette méthodologie se base sur des hypothèses que nous avons admises au départ. Nous avons tout de même proposé une vérification de ces hypothèses, notamment en vérifiant si la correction était conforme aux résultats attendus.

Les hypothèses de dépendance à la date des essais et d'indépendance des équipements de bancs sont irréfutables du point de vue métier. L'hypothèse de dépendance à la date des essais peut être attestée par les graphes. L'hypothèse d'indépendance des équipements de bancs a été étudiée un peu plus en détail d'un point de vue statistique à l'aide d'une analyse de la variance (ANOVA) à deux facteurs. Étant donné le nombre de moteurs dans chaque sous groupe il n'était pas possible de prendre en compte plus de deux facteurs et l'interaction entre équipements de bancs était à chaque fois non significative. L'hypothèse d'additivité est quant à elle plus contestable, et c'est peut-être pour cette raison qu'il reste un biais sur les nacelles. Une possibilité aurait été de corriger la covariance des données. Une correction des covariances nécessite l'estimation de matrices de covariance pour chaque modalité d'équipements de bancs puis l'utilisation d'un blanchiment de Mahalanobis des données (Kessy et al., 2018) (voir la définition Section 4.2), avant de les retransformer à l'aide de la *vraie* matrice de covariance estimée sur les observations de références non biaisées $D^{\star}$.

Par ailleurs, une correction plus compliquée (par exemple par la moyenne et la variance ou encore par la covariance entre les variables) n'aurait pas été utilisée par les ingénieurs métier dans la pratiques. En outre, en corrigeant uniquement par une moyenne dépendant du temps, on limite les risques d'introduire un bruit supplémentaire dans les données. Ainsi, les avantages de la méthodologie résident dans sa simplicité.

Dans les perspectives, nous pouvons noter deux points intéressants. Le premier est que la méthode de détection des biais des équipements de bancs peut être utilisée de manière *glissante*, sur des nouveaux moteurs et au cours du temps, pour détecter d'éventuels nouveaux décalages. Le deuxième est la généralisation de la méthodologie (détection, correction et vérification) à d'autres usages, notamment la méthodologie de détection et de vérification. Par exemple, pour d'autres usages de production industrielle, la détection de dépendance entre les éléments produits et les outils de production peut se faire à l'aide d'une méthode supervisée. De

plus, lors des deux étapes (détection et vérification), la méthodologie d'importance de variables développée dans le Chapitre 6 a été utilisée pour découvrir les variables biaisées. En effet, on s'intéresse à expliquer la dépendance entre les équipements de production et les mesures, et non pas à expliquer la prédiction. Néanmoins nous avons décidé de corriger toutes les variables car s'il existe un biais, il devrait être présent sur l'ensemble des variables selon les ingénieurs métier. De plus, lors de la correction des variables une à une, nous avons entraîné un modèle pour chaque correction, c'est-à-dire que l'étape de vérification a été faite après la correction de chaque variable (pour chaque équipement) et on a observé une détérioration des performances des modèles pour toutes les variables. La méthode de correction utilisée est dépendante de notre application et donc est plus difficilement généralisable.

8

Détection automatique d'observations rares pendant les essais de production à l'aide de modèles statistiques

8.1 Abstract . . . 118
8.2 Introduction . . . 118
8.3 Data Analysis . . . 119
8.3.1 Structure of the Data 119
8.4 Expert Knowledge to Define Anomalies . . . 120
8.4.1 Define Anomalies in Production Tests Data 120
8.5 Anomaly Detection . . . 121
8.5.1 Definition of the Method 121
8.5.2 Model Interpretability with Shapley Values 122
8.6 Results on the Data . . . 122
8.6.1 Average Shapley Values 122
8.6.2 Single Prediction Explanation 123
8.7 Anomaly Categorization Using Self-Organizing Map . . . 124
8.7.1 Definition of the Method 124
8.7.2 Choose a Subset of Variables with the Help of Group-Sparse Weighted K-means 125
8.7.3 Data Representation with SOM 126
8.8 Conclusion . . . 127

RÉSUMÉ EN FRANÇAIS Les moteurs sont vérifiés au cours de tests de production avant d'être livrés aux clients. Au cours de ces essais, de nombreuses mesures sont prises sur différentes parties du moteur, en tenant compte de multiples paramètres physiques. Des mesures inattendues peuvent être observées. Ainsi, il est important d'évaluer si ces observations atypiques sont statistiquement significatives.

La détection d'anomalies est un problème difficile en apprentissage non supervisé. La raison évidente est que, contrairement à la classification supervisée, il n'existe pas de vérité de terrain par rapport à laquelle nous pourrions évaluer les résultats. Par conséquent, nous proposons une méthodologie basée sur deux algorithmes statistiques indépendants pour vérifier les résultats : i) Isolation Forest pour estimer le score d'anomalie et les valeurs de Shapley pour interpréter ces scores ; ii) Des Self-Organizing Map (SOM) sur un sous-ensemble de variables obtenu avec l'algorithme Group-Sparse K-means.

Les contributions de ce travail sont donc :

- l'extension des valeurs de Shapley au cas non supervisé et notamment pour interpréter le modèle Isolation Forest ;
- l'utilisation de SOM (et du Group-Sparse K-means) pour valider les méthodes de détection des anomalies.

Contrairement aux Isolation Forest, les SOM fournissent des visualisations et des catégorisations des anomalies, ce qui donne des informations supplémentaires pour mieux comprendre et vérifier les anomalies estimées.

Les deux méthodes donnent des résultats similaires. En effet, les moteurs détectés comme anomalies via les Isolation Forest coïncident avec les moteurs qui ont les plus grandes distances estimées avec les SOM. Les résultats sont interprétables et exploitables par les experts métier. Cet article a fait l'objet d'une publication dans Mourer et al. (2020b).

REMARQUE Le travail du Chapitre 7 a été réalisé après celui de ce chapitre. Les données n'ont pas été corrigées selon les équipements de bancs avec la méthodologie présentée au Chapitre 7, mais elles ont été centrées par modalité d'équipement. La méthodologie de ce chapitre reste générale et peut être réutilisée sur de nouveaux ensembles de données.

8.1 Abstract

Engines are verified through production tests before delivering them to customers. During those tests, numerous measures are taken on different parts of the engine, considering multiple physical parameters. Unexpected measures can be observed. For this very reason, it is important to assess if these unusual observations are statistically significant.

However, anomaly detection is a difficult problem in unsupervised learning. The obvious reason is that, unlike supervised classification, there is no ground truth against which we could evaluate results. Therefore, we propose a methodology based on two independent statistical algorithms to double check the results. One approach is the Isolation Forest model which is specific to anomaly detection and able to handle a large number of variables. The goal of the algorithm is to find rare items, events or observations which raise suspicions by differing significantly from the majority of the data and, at the same time, it discriminates non-informative variables to improve estimation. One main issue of Isolation Forest is its lack of interpretability. Within this scope, we extend the Shapley values, interpretation indicators, to the unsupervised context to interpret the model outputs.

The second approach is the Self-Organizing Map (SOM) model which has nice properties for data mining by providing both clustering and visual representation. The performance of the method and its interpretability depend on the chosen subset of variables. In this respect, we first implement a sparse-weighted K-means to reduce the input space, allowing the SOM to give an interpretable discretized representation.

We apply both methodologies on aircraft engines data. Both approaches show similar results which are easily interpretable and exploitable by the experts.

8.2 Introduction

As an aircraft engines manufacturer, Safran verifies all individual engines before delivering to the customer during production tests. Those bench test operations generate lots of measures for different parts of the engines, resulting in multiple physical parameters acquisitions. As we may encounter unexpected measures, it is important to detect their causes and relevance. We build statistical methods to reach this goal.

Variations between performances of engines are common. Nevertheless, the production tests that verify essential engine functions before delivering it to an airline company are done in different bench test cells, under different ambient conditions, etc. A thermodynamic model is applied to compensate for context variations but there still exist some second level residuals we may have to compensate to enhance the quality of the measurements. They essentially depend on test bench components like slave cowls, but also sites and suppliers.

Therefore, one of the objectives is to take into account test bench components effects. Furthermore, there is no universally admitted way to evaluate unsupervised anomaly detection algorithms results. Hence, we proposed a new methodology based on two different algorithms.

- Large number of variables (>50) make statistical estimation challenging, especially w.r.t. the small number of engines (591). Therefore, a specific algorithm for anomaly detection, named Isolation Forest (Liu et al., 2008), is proposed. Isolation-based methods measures the probability to be isolated and anomalies are those that have the highest probability. In randomly generated binary trees, where instances are recursively partitioned, the trees produce noticeable shorter paths for anomalies. In fact, regions occupied by anomalies are low density regions, which result in a smaller number of partitions (shorter paths). Furthermore anomalies have, by definition, distinct feature-values and thus they are more likely to be separated early in the partitioning process.

- Then, another unsupervised algorithm named, Self-Organizing Map (SOM) is applied (Kohonen, 1982). It acts as an extension of the k-means algorithm that preserves as much as possible the topological structure of the data. Moreover, SOM has an intrinsic distance between prototypes and their direct neighbours. This latter representation can validate the estimation of Isolation Forest if the results coincide. Finally, SOM gives a discretized representation of the input space, which categorize anomalies. The categorization helps to understand the origin of the problem.

In addition to the rare events detection task, we need to provide explanations of the different models. Moreover, important parameters must be discovered at a local level to figure out flaws in a single engine and at a global level to discern origins of unexpected variations and inherent bias.

8.3 Data Analysis

8.3.1 Structure of the Data

The characteristics of the test bench data used in the analysis are as follows :

- 14 variables are chosen in an expert-manner by domain experts of the performance team of Safran. They are not generally interested in other variables and thus we limit ourselves to this subset of variables. However, in future works we will consider a larger set of variables.
- 591 engines are observed.
- 4 stabilized points are considered.

A stabilized point, is a fixed level of performance for which all engines are tested and measurements acquired. They are ordered from the lowest to the highest level of performance. In the database, six are available, but we do not consider the two first because they are taken at low engine speeds where there is a lot of variance in measures which makes them difficult to analyze. The measures of interest are listed below. Nine of them are numerical variables :

- FNIN1 : Thrust (FN : performance).
- XN12R : LP spool speed (N1 : fan speed).
- XN25R : HP spool speed (N25 : core speed).
- WF36 : Fuel flow.
- W2AR : Engine corrected air flow.
- P3 : HP compressor discharge pressure.
- T49C : LP turbine inlet temperature (EGT : exhausting gas temp).
- T3 : HP compressor discharge temperature.
- P18QSC : Pressure section 18 normalized by standard conditions.

where HP and LP stand for High pressure and Low pressure respectively. The variables listed above are described in Figure 8.1.

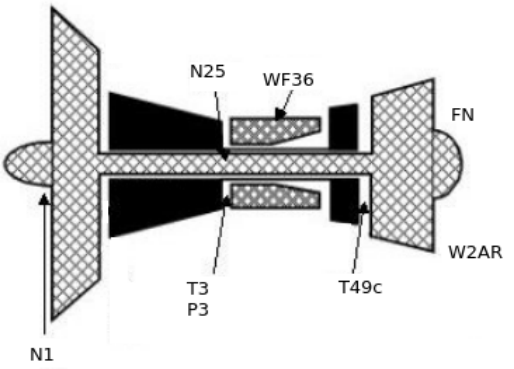


Figure 8.1 : Simplified diagram of a turbofan engine where the different measured variables are specified.

8.4 Expert Knowledge to Define Anomalies

Some pre-treatments are needed before detecting unusual case in the data. As said before, it is normal to have variance in production tests data. Those fluctuations do not necessarily represent unexpected behavior. In practice, the behavior of an engine is not defined with measurements taken independently, but it is defined between pairs of measurements. Thus, an engine has a normal level of functioning if it has a "constant ratio" between some defined pairs of variables across the stabilized points considered.

In other words we do not define an anomaly considering the observed values, but we construct a new data set from the observed one where each variable is constructed considering the dependence between pairs of measurements.

In the new data set, the dependence between the pairs of variables is of importance, and especially the evolution of these dependencies across the different stabilized points. At each stabilized point, a physical equation describes the relationship for each pair of variables and reveals the expected behavior of engines.

Figure 8.2 shows an example of the physical equation (red line) for the pair of variables (FNIN1, W2AR) and the observed values for the set of engines at the fourth stabilized point. The line gives one important information, that is the expected relationship between the thrust (FNIN1) and the mass flow rate (W2AR) ; which means that for a certain value of thrust we expect a certain mass flow rate.

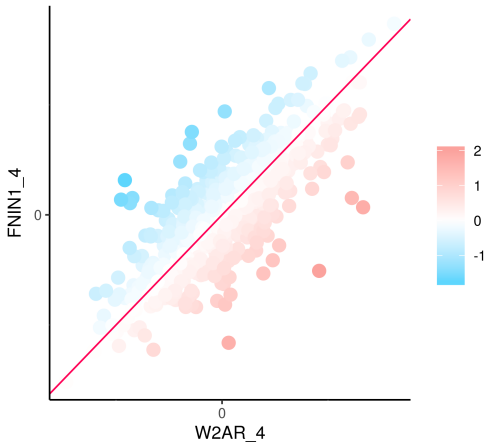


Figure 8.2 : Representation of the functioning line for the pair of variables (FNIN1, W2AR) on the stabilized point four. The value of each engine is represented by the colors in the orthogonal space estimated with the RPCA.

However, the equation of the functioning line (red line) is unknown and its estimation can be done with the help of a Robust-Principal component analysis (PCA) as detailed in Candès et al. (2011). PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. Therefore, the first axe of the PCA will model the functioning line, and the second axe will represent the space where the new variable will lie. The Robust PCA (RPCA) is an extension of PCA that is less sensible to extreme values, thus the slope of the functioning line will not depend on outliers and let them easier to detect.

8.4.1 Define Anomalies in Production Tests Data

For different stabilized points, we expect that an engine keeps a constant ratio between pairs of variables, i.e. their relationships do not change over the stabilized points. Therefore, we consider the engines values projected on the second axis obtained by the RPCA for each stabilized point. Then, we define a normal engine behavior as follow : *An engine has a normal behavior if it has similar projected values across the stabilized points.*

Formally, let us consider an engine $i \in \{1, \dots, n\}$, a pair of variables $j \in \{1, \dots, J\}$ and a stabilized points $p \in \{1, \dots P\}$. Let $y^j_{i,p}$ be the projection of the engine i on the second axis of the RPCA for the pair of variables j at the stabilized point p. Then, the mean value over the stabilized points for an engine is

$$m^j_i = \frac{1}{P} \sum_{p=1}^{P} y^j_{i,p}. \tag{8.1}$$

The difference of an engine from its mean value for a stabilized point p is

$$x_{i,p}^j = y_{i,p}^j - m_i^j. \tag{8.2}$$

Thus given j, $\forall p$ a new variable $\mathbf{x}_p^j = (x_{1,p}^j, \ldots, x_{n,p}^j)^T$ is created. Then, $x_{i,p}^j$ represents the deviation of the engine i at a specific point p compared to its mean value m_i^j to the pair j. Thus, small values of $x_{i,p}^j$ imply small deviations thus normal behavior, meanwhile large values lead to anomalies.

Nine pairs of measurements are defined in an expert manner are : (FNIN1, W2AR), (XN12R, W2AR), (P18QSC, W2AR) are the thrust, the LP spool speed and the pressure at section 18 given the engine corrected air flow. (FNIN1, WF36), (T49C, WF36) are the thrust and the exhausting gaz temperature given the fuel flow. The core speed given the fan speed, the pressure and the temperature at section 3 (HP) are also considered (XN25R, XN12R), (XN25R,T3), (XN25R, P3). Finally, the thrust function of the exhausting gaz is taken into consideration (FNIN1, T49C).

Each pair of variables is observed over four stabilized points, which give 36 new variables.

Note that, to keep an understanding of the new variables obtained from a pair, they are named as follows : "first variable name _ second variable name _ stabilized point ". For example, the variable created from T49C and FNIN1 on the fourth stabilized point will be called T49C_FNIN1_6 (the 4 stabilized point levels are listed from 3 to 6).

8.5 Anomaly Detection

8.5.1 Definition of the Method

Engines, in most of the cases, have solid and adequate measures. Thus, checking for unusual values can be seen as a statistical problem of outlier detection. For our purpose, a statistical method that is both efficient and interpretable is required. Density-based techniques are the most competitive and among these, Isolation Forest in Liu et al. (2008) showed best results on various studies (Goix, 2016). We employ this method on the new data to detect anomalies.

Isolation Forest is similar in principle to Random Forest (Breiman, 2001) and is built on the basis of decision trees. It identifies anomalies or outliers rather than profiling normal data points. Isolation Forest isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of that selected feature. Then, if an observation lies in a high-density region, the probability to isolate it is small because the values of the splits must be very close. On the other hand, if an observation lies in a low-density region, then many values of splits can isolate it, thus it has a higher probability to be isolated by a random split. Random partitioning produces noticeably shorter paths for anomalies. When a forest of random trees collectively produces shorter path lengths for particular samples, they are highly likely to be anomalies.

Let $h_t(\mathbf{x})$ the path length of $\mathbf{x}$ in the tree t and $h(\mathbf{x}) = \sum_{t=1}^{T} h_t(\mathbf{x})$ the average value of $h(\mathbf{x})$ over the trees with T is the total number of trees in the forest. The number of splits required to isolate an observation is influenced by the number of samples n in the data. To account for this a normalized anomaly score, relying on a property of Binary Search Trees (BST)) Liu et al. (2008), is defined as

$$f(\mathbf{x}, n) = 2^{\frac{-h(\mathbf{x})}{c(n)}}, \tag{8.3}$$

with $c(n)$ defined as

$$c(n) = \begin{cases} 2H(n-1) - \dfrac{2n-1}{n} & \text{for } n > 2, \\ 1 & \text{for } n = 2, \\ 0 & \text{otherwise,} \end{cases} \tag{8.4}$$

where n is the size of data set and H is the harmonic number. The value of $c(n)$ above represents the average of $h(\mathbf{x})$ given n, so we can use it to normalise $h(\mathbf{x})$ and get an estimation of the anomaly score for a given instance $\mathbf{x}$. Note that $f \in [0; 1]$ with value closer to 1 indicates that the observation is more likely to be an anomaly.

Once the Isolation Forest has been applied to the data, an anomaly score is estimated for each engine. This anomaly score is pointless if it cannot be completely understood by domain experts.

For complex models, such as ensemble methods, deep networks or Isolation Forest, we cannot use the original model as its own best explanation because it is not easily understandable. Instead, we must use a simpler explanation model, which we define as any interpretable approximation of the original model. We

would like to have an average explanation of the model as well as explanation of single prediction and this is called as local explainability (Guidotti et al., 2018) which is also known as "post-hoc" explainability. In, Doshi-Velez and Kim (2017) they assert that a useful local explanation should answer the following questions : What were the main factors in the decision ? Would changing a certain factor have changed the decision ? Why did two similar-looking cases get different decisions, or vice versa ? More precisely, we would like to understand how variables contributed to the score of a single engine as well as how variables contributed in average. In addition, understand whether the contribution of a variable have a positive impact or a negative impact on the score, or in other words if a variable helps to make an observation more normal or more abnormal. In this aim, Shapley values will be used.

8.5.2 Model Interpretability with Shapley Values

Shapley values have attracted a great deal of attention in recent years in the field of interpretability, which has been originally discussed in game theory (Shapley, 1953) and recently applied to statistics (Štrumbelj and Kononenko, 2014 ; Owen and Prieur, 2017 ; Iooss and Prieur, 2017 ; Lundberg and Lee, 2017). Moreover, in the context of anomaly detection, few results have already been reported (Antwarg et al., 2019 ; Giurgiu and Schumann, 2019 ; Takeishi, 2019 ; Takeishi and Kawahara, 2020). These results have confirmed the usefullness of the Shapley value for anomaly interpretation. As described in their works, we will adopt general techniques in defining and computing the Shapley values derived from supervised learning.

Shapley values measure features importance for models in the presence of interaction between variables. This method requires retraining the model on all feature subsets $S \subseteq F$, where F is the set of all features. As described in Lundberg and Lee (2017), it assigns an importance value to each feature that represents the effect on the model prediction of including that feature. To compute this effect, a model $f_{S\cup j}$is trained with that feature present, and another model f_S is trained with the feature excluded. In our case, $f(\mathbf{x})$ is the anomaly score given by the Isolation Forest. Then, predictions from the two models are compared on the current input $f_{S\cup j}(\mathbf{x}_{S\cup j}) - f_S(\mathbf{x}_S)$, where x_S represents the values of the input features in the set S. Since the effect of withholding a feature depends on other features in the model, the preceding differences are computed for all possible subsets $S \subseteq F\backslash j$. The Shapley values are then computed and used as feature attributions. They are a weighted average of all possible differences :

$$\phi_j = \sum_{S\subseteq F\backslash j} \frac{|S|!(|F| - |S| - 1)!}{|F|!}\big[f_{S\cup j}(x_{S\cup j}) - f_S(x_S)\big], \tag{8.5}$$

At first glance, the above equation seems ridiculously complicated, but it can be easily explained in one sentence : the contribution of a feature j is the mean difference between a model trained on a subset of variables S with j and a model trained on the same subset S without j, and this is done for all the possible subsets of variables. All possible sets of feature values have to be evaluated with and without the j-th feature to calculate the exact Shapley value. For more than a few features, the exact solution to this problem becomes problematic as the number of possible coalitions exponentially increases as more features are added. In Štrumbelj and Kononenko (2014) an approximation with Monte-Carlo sampling is proposed

$$\hat{\phi}_j = \frac{1}{M}\sum_{m=1}^{M}\big(\hat{f}(\mathbf{x}^m_{+j}) - \hat{f}(\mathbf{x}^m_{-j})\big), \tag{8.6}$$

where $\hat{f}(\mathbf{x}^m_{+j})$ is the prediction for $\mathbf{x}$, but with a random number of feature values replaced by feature values from a random data point $\mathbf{z}$, except for the respective value of feature j. The vector $\mathbf{x}^m_{-j}$ is almost identical to $\mathbf{x}^m_{+j}$, but the value $\mathbf{x}^m_j$ is also taken from the sampled $\mathbf{z}$. Each of these M new instances is an "artificial object" assembled from two instances. If $\hat{\phi}_j$ is positive the value of the feature j increase the anomaly score, and it decreases if $\hat{\phi}_j$ is negative.

8.6 Results on the Data

8.6.1 Average Shapley Values

The Table 8.1 gives the average Shapley values by variable on all observations. It provides a nice interpretation of the effect of each variable on the anomaly score. A variable j with a $\hat{\phi}_j$ value close to 0 will not affect the output of the model and thus the variable is not important to detect anomalies. A high absolute $\hat{\phi}_j$ value points out that the variable plays an important role in model estimates. The sign of the contribution gives additional information on the effect of the variable. A positive contribution denotes that the variable helped

to increase the estimated anomaly score w.r.t. the average anomaly score whereas, a negative contribution decreases it. Therefore, if in average a variable has a negative $\hat{\phi}_j$, it means that it does not globally contribute to make an engine significantly different from others.

Table 8.1 : Average Shapley value by variables. Positive $\hat{\phi}_j$ values indicate that the variable tends to increase the anomaly score while, negative one indicates the opposite.

variable	ϕ
WF36_FNIN1_4	1.7839
WF36_T49C_4	1.4149
P3_XN25R_3	1.3275
T49C_FNIN1_6	1.0992
XN12R_XN25R_5	0.8475
WF36_T49C_3	0.7794
WF36_FNIN1_3	0.7782
P3_XN25R_6	0.7207
W2AR_XN12R_3	0.5513
WF36_T49C_6	0.4814
W2AR_XN12R_4	0.4696
XN12R_XN25R_6	0.4378
XN12R_XN25R_4	0.3720
WF36_T49C_5	0.2933
P3_XN25R_5	0.2284
W2AR_XN12R_6	0.1843
WF36_FNIN1_6	0.1367
CNOZ	0.1318
T3_XN25R_4	0.0514
CELL	0.0487
T49C_FNIN1_4	0.0212
WF36_FNIN1_5	-0.0029
W2AR_XN12R_5	-0.0201
XN12R_XN25R_3	-0.0446
P3_XN25R_4	-0.0780
W2AR_FNIN1_6	-0.1006
T49C_FNIN1_5	-0.1676
W2AR_FNIN1_4	-0.2927
COWL	-0.4981
W2AR_P18QSC_5	-0.5035
BMSN	-0.5225
W2AR_P18QSC_3	-0.6202
W2AR_P18QSC_4	-0.6486
W2AR_FNIN1_3	-0.7632
W2AR_FNIN1_5	-0.7816
W2AR_P18QSC_6	-0.7929
T3_XN25R_5	-0.7946
T3_XN25R_6	-0.8460
T49C_FNIN1_3	-1.0517
T3_XN25R_3	-1.1104

8.6.2 Single Prediction Explanation

Averaged explanation are useful and give insights but they are not sufficient. When an engine has a high estimated anomaly score, domain experts would like to understand which variables are responsible for this score. As an example, the engine 41 is observed. Shapley values are applied to explain how variables contributed to the score. On Figure 8.3, the average score of anomaly for engines is 0.40 and the engine 41 has an anomaly score of 0.47, which is significantly larger. This difference is decomposed variable by variable. The x-axis, ϕ, gives the weight of the contribution. On the y-axis, the engine values for each variable are displayed, ordered by decreasing importance. A value of 0 on the x-axis indicates that the variable does not play any role in the estimation of the score. Note that, most important variables for this engine correspond to the highest deviations $x^j_{41,p}$ obtained by RPCA. However, due to the possible high-order interaction between variables, a complex model was needed to assess a good estimation of the anomaly score.

Domain experts have access to the contribution of the variables and they can control the validity of the results. Figure 8.4 shows the densities of variables with the highest (WF36_T49C_5) and the lowest (T3_XN25R_5) ϕ detected for engine 41. As expected, the value $x^j_{41,5}$ for j =WF36_T49C, is far from 0 and isolated in a low density area, which means that it has a really different behavior over the four stabilized points. On the other hand, for j =T3_XN25R, $x^j_{41,5}$ is close to 0 and the engine has a consistent behavior.

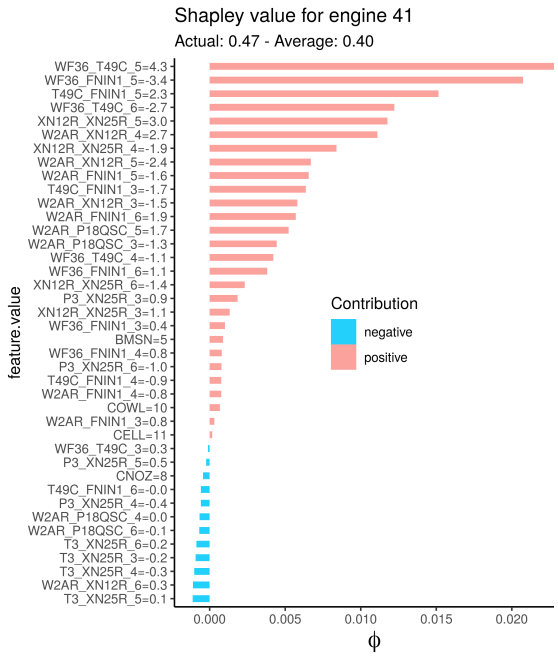


Figure 8.3 : Shapley plot of the engine 41. 0.40 is the average anomaly score and 0.47 is the predicted score of the engine 41. Feature value with positive $\hat{\phi}$ increase the score from 0.40 to 0.47, and negative value of $\hat{\phi}$ decrease the anomaly score of the engine.

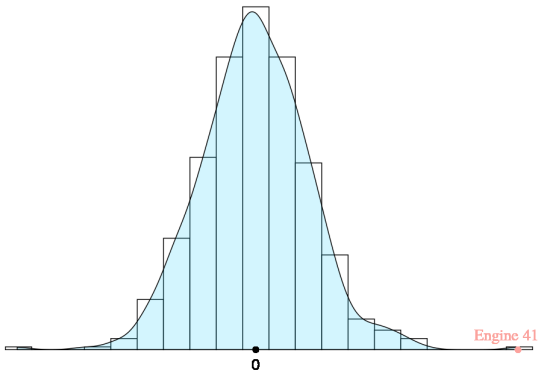


Figure 8.4 : The density of the variable WF36_T49C_5. Engine 41 is isolated in a low density area which explains why this variable has a high contribution to increase the anomaly score.

A diagram that explains the process of engines tests validation using the statistical methodology is presented in Figure 8.6. Isolation Forest helps domain experts to identify few engines and Shapley values help them to focus on some specific measures. Then a complete inspection of the engine can be done before validating the production test.

8.7 Anomaly Categorization Using Self-Organizing Map

8.7.1 Definition of the Method

A SOM is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional) discretized representation of the input space of the training samples, called a map. Each unit of the map corresponds to a prototype vector in the original high-dimensional space, and new data points are projected on the map by finding the closest prototype vector w.r.t. euclidean distance (Kohonen, 1982 ; Olteanu and Villa-Vialaneix, 2015). Self-organizing maps have been used for aircraft engine fleet monitoring in Cottrell et al. (2009) ; Côme et al. (2010b,a) ; Forest et al. (2018) and to classify transient flight phases Faure et al. (2017). No specific study has been yet conducted on using SOM to validate and categorize anomalies and especially on production tests data.

SOM has both intrinsic distances between clusters and nice two-dimensional visualization, which make it

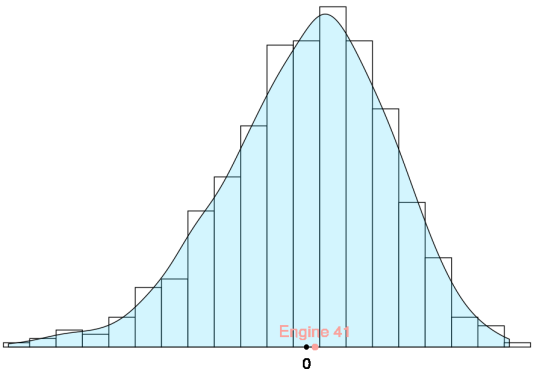


Figure 8.5 : The density of the variable T3_XN25R_5. Engine 41 lies in a high density area, which explains why this variable contributes to decrease the anomaly score.

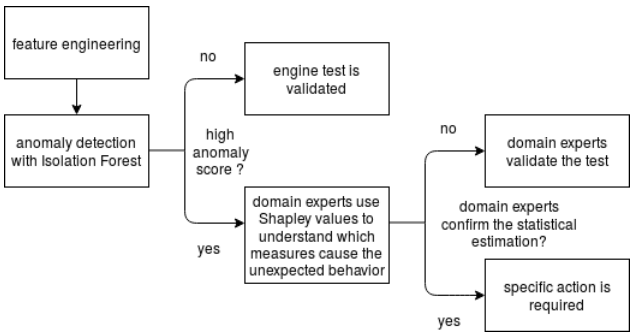


Figure 8.6 : Diagram of the process of production tests validation using Isolation Forest and Shapley values. The statistical methodology helps to highlight specific engines and measures where further analyses are needed.

a good candidate. Nonetheless, clusters are still in high-dimensions and methods such as Shapley value are not tractable in this situation. Therefore, in the next section, before modelling a SOM, specific clustering algorithm for variable selection will be used.

8.7.2 Choose a Subset of Variables with the Help of Group-Sparse Weighted K-means

Group-sparse weighted K-means generalizes the sparse weighted K-means algorithm for numerical variables in Witten and Tibshirani (2010), by using the group regularization framework. Suppose that the numerical matrix of data $\mathbf{X}$ is described by p features that are divided into L priorly known distinct groups, such that $\mathbf{X} = \left[\mathbf{X}^1|...|\mathbf{X}^L\right]$, with $\mathbf{X}^\ell \in \mathbf{R}^{n\times p_\ell}$, p_ℓ being the size of group ℓ, and $p_1 + ... + p_L = p$.

In presence of group data, we would like to discriminate groups of variables $\mathbf{X}^\ell$ by using a specific L_1-group penalty, which has been already used in the regression framework (Yuan and Lin, 2006). This allows us to select variables by group, forcing the model to select or discriminate the entire group. As described in Chavent et al. (2020), the between-class variance of each variable is multiplied by a weight and a parameter λ penalizes the weights. The latter discriminates the groups of variables with the lowest between-class variance. There is a clustering solution (groups weights and clusters) for each fixed λ. The regularization path (clustering solution given lambda) is computed at a grid of values for the penalty factor λ, covering the entire range, from a model with all the groups included to a model with only one group. The optimization procedure is quite straightforward. The algorithm is optimized in an iterative fashion : first the K-means algorithm is performed on the weighted space of features, then the partition is held fixed and the weights are updated. This iterative procedure is continued until a (local) minimum is reached.

In this context, groups are clearly formed by the variables over the stabilized points. For example, T3_XN25R_3, T3_XN25R_4, T3_XN25R_5 and T3_XN25R_6 belong to the same group. Hence, there are 9 groups of variables, and we would like to know which are the most discriminative for clustering. Moreover, Table 8.1 shows that test bench components were not significant to detect unusual behavior. Shapley values attribute them in average a negative contribution, which implies that they are not useful to model unexpected behavior. Thus, these variables are not considered in the analysis. Furthermore, the number of clusters is set to five and was found with the Silhouette method (Rousseeuw, 1987).

In Figure 8.7 we provide the path of groups' weights against the λ sequence. Weights of groups, on the

y-axis, are represented for each λ. The most important groups are 7, 4 and 5 which are respectively the groups P3_XN25R, T49C_FNIN1 and WF36_FNIN1.

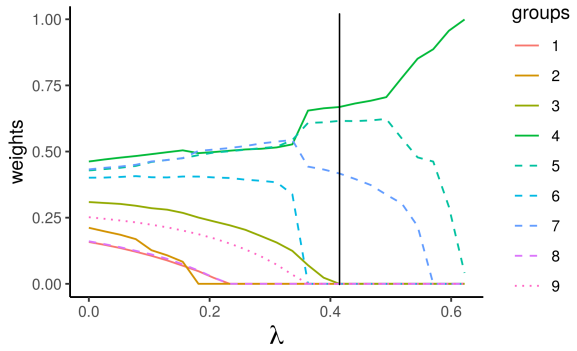


Figure 8.7 : Group weights for each value of λ. The vertical line represents the selected clustering solution where three groups of variables have non-zero weights.

In Figure 8.8 we see a big gap in terms of explained variance before $\lambda = 0.6$ If one give a closer look, the analysis points out that a subset of variables (3 groups) will give similar clustering, in terms of explained variance, to the one with all the variables included (see Figure 8.7 and Figure 8.8). We choose the clustering obtained for the value of λ represented by the vertical line. This value of λ allows selecting 3 groups with high explained variance. Higher value of λ lead to have a more similar solution to the one after the gap in Figure 8.8 in terms of weights, which seems to be a bad clustering solution. Therefore, the chosen value of λ seems to be a good trade-off between interpretability and performance. The weights obtained by groups are $\mathbf{w}_{\text{T49C_FNIN1}} = 0.67$, $\mathbf{w}_{\text{WF36_FNIN1}} = 0.62$ and $\mathbf{w}_{\text{P3_XN25R}} = 0.42$. This subspace of 12 variables will be used to represent the data with the help of the SOM algorithm.

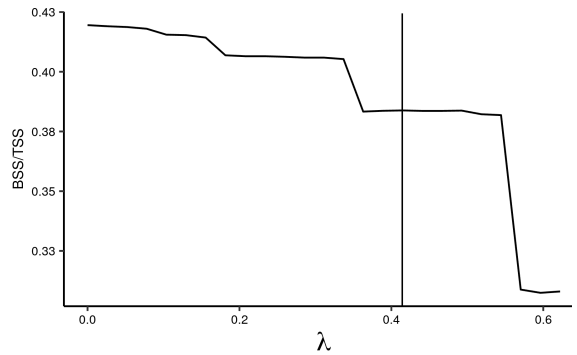


Figure 8.8 : The ratio of between-sum of squares over the total sum of squares (BSS/TSS : explained variance) for each value of λ. The vertical line represents the selected clustering solution which has an explained variance that is close to the one with the full set of variables.

8.7.3 Data Representation with SOM

The SOM allows us to have access to several different visualizations. First, we plot the distances between prototypes (Figure 8.9), which gives a representation of the grid where the colors represent the mean distance to the neighbor prototypes. The color scale goes from blue to purple, where purple indicates a large distance. In Figure 8.10 the repartition of the engines on the map is provided. Finally the Figure 8.11 is the SOM map where colors indicate the mean anomaly score of engines estimated with the Isolation Forest by clusters. The color scale goes from yellow to red, where red indicates a higher anomaly score. It is interesting to note that the engines with high anomaly score are distributed on the border of the map. The representation is very similar to the one given by the smooth distances between prototypes on Figure 8.9 which shows that the two methods agree. In addition, the map allows a categorization of the anomalies. In Figure 8.11, Super-clusters are identified thanks to the mean anomaly score of the prototypes and their proximity on the SOM map. Two map edges are thus identified as super-clusters. The comparison of these clusters of anomalies with the rest of the population will allow us to understand the discriminative variables.

The ANOVA method is applied to test significant difference between clusters (Figure 8.12). ANOVA provides a statistical test of whether two or more population means are equal. The results show that, for cluster



Figure 8.9 : Smooth distances between prototypes. The background colors indicate the distances between neighboring prototypes where pink corresponds to larges distances.

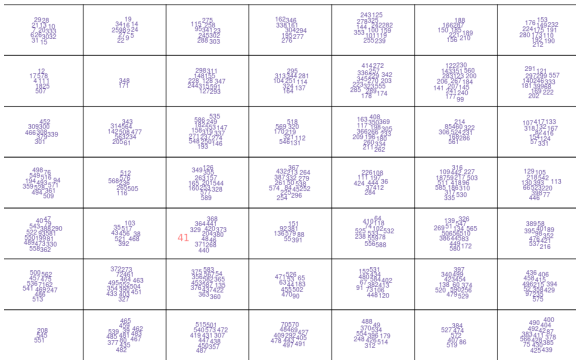


Figure 8.10 : Repartition of the engines on the map.

2, the group of variables P3_XN25R seems to be important. On the other hand, for cluster 1, the group of variables WF36_FNIN1 may explain their unusual behavior. The ANOVA shows that the set of explaining variables has been reduced to only one group. For the sake of readability, we show the ANOVA test for only two variables at the stabilized point 3, but for the other stabilized points results are similar since they are all linked by construction. Moreover, the other groups of variables are not significantly different over the three clusters.

8.8 Conclusion

In this work, statistical methods in the context of rare event detection in production tests data demonstrated high degree of efficiency and interpretability in either local (one specific engine) or global level (groups of engines). We propose a multi-scale model, giving a hierarchy in the information, allowing the experts to better understand flaws on a particular engine but also allowing them to detect more general problems. We apply two different methods : i) Isolation Forest to estimate anomaliy score and Shapley values to interpretate them ; ii) SOM on a subset of variables obtained by group-sparse weighted K-means. Both methods provide similar results : the engines detected as anomalies with the Isolation Forest coincide to the engines that have the largest distances estimated with SOM.

Moreover, an other contribution of this work is the use of SOM to validate anomalies detection methods. On the contrary of Isolation Forest, SOM provides visualizations and categorizations of the anomalies which gives additional information to better understand and verify the estimated anomalies.

Explainability in unsupervised learning is a new field that needs to be explored, and this work is a step forward in this direction. Some further investigations are needed in both theoretical and applied domains. In future works, we plan to explore those points and also we will develop a method to transform the anomaly score in a binary score which will help to domain experts in their decisions.

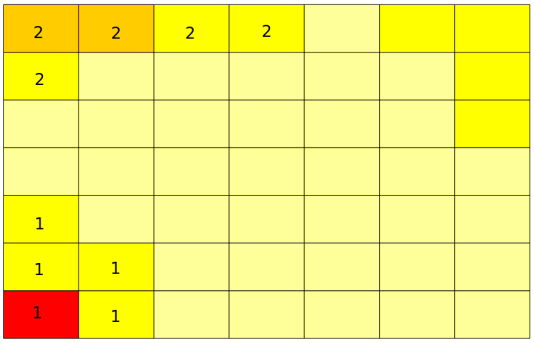


Figure 8.11 : SOM map of engines where the colors represent the anomaly score estimated with Isolation Forest averaged by clusters. Color scale goes from yellow to red, where red corresponds to a higher score.

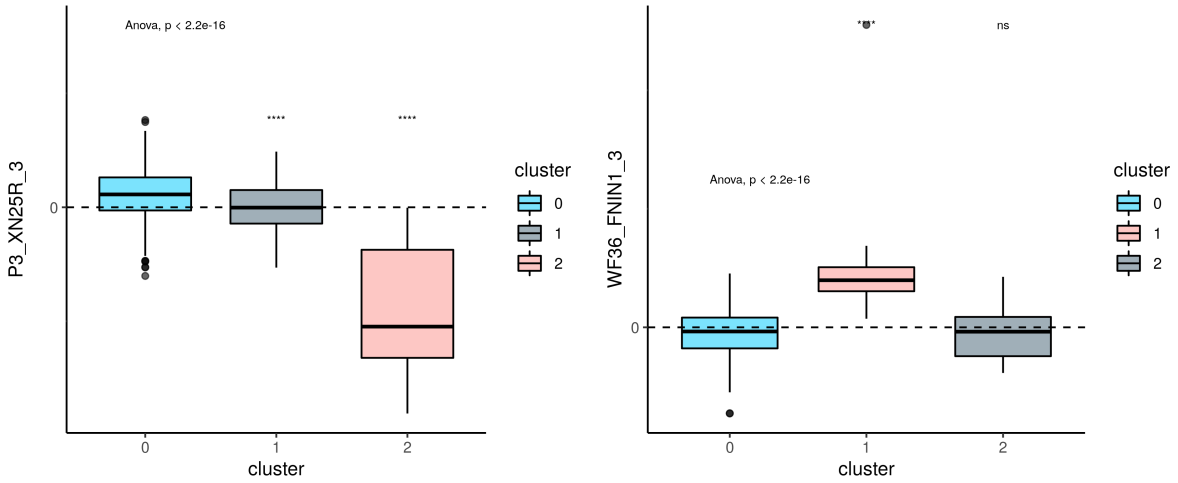


Figure 8.12 : Boxplot of ANOVA test comparing two different area of the map Figure 8.9. For the two variables considered, only one cluster (the pink one) is truly different from the overall population of engines, implying that the two clusters of anomalies can be explained by different subsets of variables.

9

Conclusion

Dans ce manuscrit, nous avons abordé les questions d'importance et de sélection de variables en apprentissage supervisé et non supervisé et notamment en clustering. Un résumé des contributions se trouve dans l'introduction Section 1.4 et les conclusions, limitations et perspectives de chacune d'entre elles sont présentées dans les chapitres correspondants.

Ce travail de thèse met l'accent sur un point majeur des statistiques et du *machine learning* ou apprentissage automatique : l'importance de prendre en compte les corrélations dans les données. Nous avons illustré l'impact que celles-ci peuvent avoir en clustering (en particulier dans le Chapitre 4), discuté leurs effets dans le cadre de la prédiction (en particulier dans le Chapitre 6) et étudié leurs incidences sur la sélection et l'importance de variables (en particulier dans le Chapitre 4 et dans le Chapitre 6).

L'analyse en composantes principales a pour but de réduire la dimension d'un ensemble de données comprenant un grand nombre de variables interdépendantes, tout en conservant autant que possible la *dispersion* présente dans l'ensemble de données. Cette *dispersion*, dans lesquelles interviennent les corrélations, n'est pas nécessairement synonyme d'information. En effet, multiplier les descripteurs d'un même phénomène ne correspond pas à une augmentation de l'information que l'on a de ce phénomène. Cela dépend de l'application et dans notre cas les corrélations ont été considérées comme de la redondance et c'est suivant ce point de vue que nous avons mené nos travaux.

Dans cet esprit, les corrélations ont inéluctablement une influence sur les résultats des méthodes d'apprentissage dans un grand nombre de domaines des statistiques comme par exemple la détection d'anomalies. Si deux variables sont identiques, alors une valeur extrême sur l'une comptera double dans le calcul du score d'anomalies pour la plupart des algorithmes qui en font la détection. Il serait pertinent de modifier ces algorithmes en présence de corrélations.

De manière générale, les méthodologies d'apprentissage automatique sont théoriquement et empiriquement plus faciles à étudier et à utiliser lorsque les données sont décorrélées. Une suite intéressante de ces travaux de thèse consisterait à mettre au point d'autres méthodes, dans la lignée du Chapitre 6, en transformant les données en données synthétiques décorrelées construites à partir du regroupement de variables similaires. Un des objectifs serait d'associer cette transformation de l'espace des données à l'objectif de l'algorithme (prédiction, clustering d'observations, détection d'anomalies).

Plusieurs pistes sont à étudier mais une semble particulièrement prometteuse dans le domaine de l'apprentissage profond ou *deep learning*, où il est possible de simultanément apprendre une représentation décorrélée des données et de résoudre le problème en question. Par exemple, des algorithmes basés sur les *autoencoders*, notamment leur version *variationnel* (Kingma and Welling, 2019), permettent d'obtenir ce que l'on appelle une *disentengled representation* ou *représentation démêlée* qui capture l'information des variables en les regroupant en facteurs décorrélés.

Développer de nouvelles méthodes de représentation des données nécessite une définition précise des objectifs à satisfaire et des métriques permettant de les évaluer même si dans le cadre non supervisé il est extrêmement difficile de valider les résultats. En fait, la manière de sélectionner un modèle et d'en déduire l'importance des variables dépend de l'objectif à atteindre. Pour reprendre Georg Cantor, *to ask the right question is harder than to answer it.*

Bibliographie

M. Al Hasan, V. Chaoji, S. Salem, and M. J. Zaki. Robust partitional clustering by outlier and density insensitive seeding. *Pattern Recognition Letters*, 30(11) :994–1002, 2009.

Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural computation*, 9 (7) :1545–1588, 1997.

J. L. Andrews and P. D. McNicholas. Variable selection for clustering and classification. *Journal of Classification*, 31(2) :136–153, 2014.

L. Antwarg, B. Shapira, and L. Rokach. Explaining anomalies detected by autoencoders using shap. *arXiv preprint arXiv :1903.02407*, 2019.

K. J. Archer and R. V. Kimes. Empirical characterization of random forest variable importance measures. *Computational statistics & data analysis*, 52(4) :2249–2260, 2008.

E. Arias-Castro and X. Pu. A simple approach to sparse clustering. *Computational Statistics & Data Analysis*, 105 :217–228, 2017.

D. Arthur and S. Vassilvitskii. k-means++ : The Advantages of Careful Seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, 2007.

L. Auret and C. Aldrich. Empirical comparison of tree ensemble variable importance measures. *Chemometrics and Intelligent Laboratory Systems*, 105(2) :157–170, 2011.

P. Awasthi, A. Blum, and O. Sheffet. Center-based clustering under perturbation stability. *Information Processing Letters*, 2012.

M.-F. Balcan and Y. Liang. Clustering under perturbation resilience. *SIAM Journal on Computing*, 2016.

M.-F. Balcan, N. Haghtalab, and C. White. k-center Clustering under Perturbation Resilience. *ACM Transactions on Algorithms*, 2020.

J. D. Banfield and A. E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821, 1993.

R. F. Barber and E. J. Candès. Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5) :2055–2085, 2015.

S. Ben-David. Clustering-what both theoreticians and practitioners are doing wrong. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

S. Ben-David and L. Reyzin. Data stability in clustering : A closer look. *Theoretical Computer Science*, 2014.

S. Ben-David and U. Von Luxburg. Relating clustering stability to properties of cluster boundaries. *21st Annual Conference on Learning Theory, COLT 2008*, 2008.

S. Ben-David, U. Von Luxburg, and D. Pál. A sober look at clustering stability. In *International Conference on Computational Learning Theory*, 2006.

S. Ben-David, D. Pál, and H. U. Simon. Stability of k-means clustering. In *International conference on computational learning theory*, 2007.

A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 2002.

C. Bénard, S. Da Veiga, and E. Scornet. Mda for random forests : inconsistency, and a practical solution via the sobol-mda. *arXiv :2102.13347*, 2021.

G. Biau and E. Scornet. A random forest guided tour. *Test*, 25(2) :197–227, 2016.

C. Biernacki, G. Celeux, and G. Govaert. Exact and monte carlo calculations of integrated likelihoods for the latent class model. *Journal of Statistical Planning and Inference*, 140(11) :2991–3002, 2010.

Y. Bilu and N. Linial. Are stable instances easy ? *Combinatorics Probability and Computing*, 2012.

H. D. Bondell and B. J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1) :115–123, 2008.

A.-L. Boulesteix, A. Bender, J. Lorenzo Bermejo, and C. Strobl. Random forest gini importance favours snps with large minor allele frequency : impact, sources and recommendations. *Briefings in Bioinformatics*, 13 (3) :292–304, 2012.

C. Bouveyron and C. Brunet. Simultaneous model-based clustering and visualization in the fisher discriminative subspace. *Statistics and Computing*, 22(1) :301–324, 2012.

C. Bouveyron and C. Brunet-Saumard. Discriminative variable selection for clustering with the sparse fisher-em algorithm. *Computational Statistics*, 29(3) :489–513, 2014a.

C. Bouveyron and C. Brunet-Saumard. Model-based clustering of high-dimensional data : A review. *Computational Statistics & Data Analysis*, 71 :52–78, 2014b.

C. Bouveyron, G. Celeux, T. B. Murphy, and A. E. Raftery. *Model-based clustering and classification for data science : with applications in R*, volume 50. Cambridge University Press, 2019.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. doi : 10.1017/CBO9780511804441.

S. Boyd, N. Parikh, and E. Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

L. Breiman. Random forests. *Machine learning*, 45(1) :5–32, 2001.

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof : identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

Š. Brodinová, P. Filzmoser, T. Ortner, C. Breiteneder, and M. Rohm. Robust and sparse k-means clustering for high-dimensional data. *Advances in Data Analysis and Classification*, 13(4) :905–932, 2019.

S. Bubeck, M. Meila, and U. V. Luxburg. How the initialization affects the stability of the k-means algorithm. *ESAIM - Probability and Statistics*, 2012.

P. Bühlmann, P. Rütimann, S. van de Geer, and C.-H. Zhang. Correlated variables in regression : clustering and sparse estimation. *Journal of Statistical Planning and Inference*, 143(11) :1835–1858, 2013.

M. E. Burczynski, R. L. Peterson, N. C. Twine, K. A. Zuberek, B. J. Brodeur, L. Casciotti, V. Maganti, P. S. Reddy, A. Strahs, F. Immermann, et al. Molecular classification of crohn's disease and ulcerative colitis patients using transcriptional profiles in peripheral blood mononuclear cells. *The journal of molecular diagnostics*, 8(1) :51–61, 2006.

T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 1974.

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis ? *Journal of the ACM (JACM)*, 58(3) :1–37, 2011.

G. Celeux and G. Govaert. Gaussian parsimonious clustering models. *Pattern recognition*, 28(5) :781–793, 1995.

G. Celeux, M.-L. Martin-Magniette, C. Maugis-Rabusseau, and A. E. Raftery. Comparing model selection and regularization approaches to variable selection in model-based clustering. *Journal de la Societe francaise de statistique*, 155(2) :57–71, 2014.

G. Celeux, C. Maugis-Rabusseau, and M. Sedki. Variable selection in model-based clustering and discriminant analysis with a regularization approach. *Advances in Data Analysis and Classification*, 13(1) :259–278, 2019.

S. Chakraborty and S. Das. A strongly consistent sparse k-means clustering with direct l_1 penalization on variable weights. *arXiv preprint arXiv :1903.10039*, 2019.

S. Chakraborty, D. Paul, S. Das, and J. Xu. Entropy weighted power k-means clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 691–701. PMLR, 2020.

M. Chavent and G. Chavent. Group-sparse block pca and explained variance. *arXiv preprint arXiv :1705.00461*, 2017.

M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. Handling correlations in clustering : ought the variables to be standardised not only by their variance but also by their correlations ? a.

M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. A new variable importance measure based on variable clustering for random forests. b.

M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. Review of sparse clustering methods. c.

M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. Detect, correct and verify the presence of bias in reception test data. d.

M. Chavent, Y. Lechevallier, and O. Briant. Divclus-t : A monothetic divisive hierarchical clustering method. *Computational Statistics & Data Analysis*, 52(2) :687–701, 2007.

M. Chavent, V. Kuentz, B. Liquet, and L. Saracco. Clustofvar : An r package for the clustering of variables. *arXiv preprint arXiv :1112.0295*, 2011.

M. Chavent, V. Kuentz-Simonet, and J. Saracco. Orthogonal rotation in pcamix. *Advances in Data Analysis and Classification*, 6(2) :131–146, 2012.

M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. Sparse k-means for mixed data via group-sparse clustering. In M. Verleysen, editor, *28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN) : October 2-4, 2020*, pages 235–240, Online event, 2020. European Symposium on Artificial Neural Networks (ESANN), i6doc.com.

M. Chavent, R. Genuer, and J. Saracco. Combining clustering of variables and feature selection using random forests. *Communications in Statistics-Simulation and Computation*, 50(2) :426–445, 2021a.

M. Chavent, J. Lacaille, A. Mourer, and M. Olteanu. Handling correlations in random forests : which impacts on variable importance and model interpretability ? In M. Verleysen, editor, *29th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN) : October 6-8, 2021*, pages 569–574, Online event, 2021b. European Symposium on Artificial Neural Networks (ESANN), i6doc.com.

E. Côme, M. Cottrell, M. Verleysen, and J. Lacaille. Aircraft engine health monitoring using self-organizing maps. In *Industrial Conference on Data Mining*, pages 405–417. Springer, 2010a.

E. Côme, M. Cottrell, M. Verleysen, and J. Lacaille. Self organizing star (sos) for health monitoring. 2010b.

M. Cottrell, P. Gaubert, C. Eloy, D. François, G. Hallaux, J. Lacaille, and M. Verleysen. Fault prediction in aircraft engines using self-organizing maps. In *International workshop on self-organizing maps*, pages 37–44. Springer, 2009.

J. A. Cuesta-Albertos, A. Gordaliza, and C. Matrán. Trimmed k-means : An attempt to robustify quantizers. *The Annals of Statistics*, 25(2) :553–576, 1997.

D. L. Davies and D. W. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1979.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society : Series B (Methodological)*, 39(1) :1–22, 1977.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7 :1–30, 2006.

B. Desgraupes. ClusterCrit : Clustering Indices. *CRAN Package*, 2013.

A. W. Diallo, N. Niang, and M. Ouattara. Sparse subspace k-means. 2021.

R. Díaz-Uriarte and S. A. De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1) :1–13, 2006.

T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees : Bagging, boosting, and randomization. *Machine learning*, 40(2) :139–157, 2000.

C. Ding and X. He. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29, 2004.

C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos. Locally adaptive metrics for clustering high dimensional data. *Data Mining and Knowledge Discovery*, 14(1) :63–97, 2007.

F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv :1702.08608*, 2017.

S. Dudoit and J. Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome biology*, 2002.

J. C. Dunn. Well-Separated Clusters and Optimal Fuzzy Partitions. *Journal of Cybernetics*, 1974.

R. Duroux and E. Scornet. Impact of subsampling and tree depth on random forests. *ESAIM : Probability and Statistics*, 22 :96–128, 2018.

B. Efron. Second thoughts on the bootstrap. *Statistical science*, pages 135–140, 2003.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2) : 407–499, 2004.

M. Falasconi, A. Gutierrez, M. Pardo, G. Sberveglieri, and S. Marco. A stability based validity method for fuzzy clustering. *Pattern Recognition*, 2010.

Y. Fang and J. Wang. Selection of the number of clusters via the bootstrap method. *Computational Statistics and Data Analysis*, 56(3) :468–477, 2012.

C. Faure. *Détection de ruptures et identification des causes ou des symptômes dans le fonctionnement des turboréacteurs durant les vols et les essais*. PhD thesis, Université Panthéon-Sorbonne-Paris I, 2018.

C. Faure, M. Olteanu, J.-M. Bardet, and J. Lacaille. Using self-organizing maps for clustering anc labelling aircraft engine data phases. In *2017 12th international workshop on self-organizing maps and learning vector quantization, clustering and data visualization (wsom)*, pages 1–8. IEEE, 2017.

M. Fop and T. B. Murphy. Variable selection methods for model-based clustering. *Statistics Surveys*, 12 : 18–65, 2018.

F. Forest. *Unsupervised Learning of Data Representations and Cluster Structures : Applications to Large-scale Health Monitoring of Turbofan Aircraft Engines*. PhD thesis, Université Sorbonne Paris Nord, 3 2021.

F. Forest, J. Lacaille, M. Lebbah, and H. Azzag. A generic and scalable pipeline for large-scale analytics of continuous aircraft engine data. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1918–1924. IEEE, 2018.

E. W. Forgy. Cluster analysis of multivariate data : efficiency versus interpretability of classifications. *biometrics*, 21 :768–769, 1965.

A. Foss, M. Markatou, B. Ray, and A. Heching. A semiparametric method for clustering mixed data. *Machine Learning*, 105(3) :419–458, 2016.

J. Fridlyand and S. Dudoit. Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method. Technical report, 2001.

J. Friedman, T. Hastie, R. Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3) :432–441, 2008.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1) :1, 2010.

J. H. Friedman and J. J. Meulman. Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 66(4) :815–849, 2004.

R. Genuer and J.-M. Poggi. Arbres cart et forêts aléatoires, importance et sélection de variables. 2017.

R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern recognition letters*, 31(14) :2225–2236, 2010.

R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. Vsurf : an r package for variable selection using random forests. *The R Journal*, 7(2) :19–33, 2015.

I. Giurgiu and A. Schumann. Additive explanations for anomalies detected from multivariate temporal data. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2245–2248, 2019.

R. Gnanadesikan, J. R. Kettenring, and S. L. Tsao. Weighting and selection of variables for cluster analysis. *Journal of classification*, 12(1) :113–136, 1995.

N. Goix. *Apprentissage automatique et extrêmes pour la détection d'anomalies*. PhD thesis, Paris, ENST, 2016.

W. Gong, R. Zhao, and S. Grünewald. Structured sparse k-means clustering via laplacian smoothing. *Pattern Recognition Letters*, 112 :63–69, 2018.

J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

B. Gregorutti, B. Michel, and P. Saint-Pierre. Correlation and variable importance in random forests. *Statistics and Computing*, 27(3) :659–678, 2017.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5) :1–42, 2018.

J. Hämäläinen, S. Jauhiainen, and T. Kärkkäinen. Comparison of internal clustering validation indices for prototype-based clustering. *Algorithms*, 2017.

G. Hamerly and C. Elkan. Learning the K in K-means. *NIPS*, 2004. ISSN 10495258.

D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45(2) :171–186, 2001.

J. A. Hartigan and M. A. Wong. Algorithm as 136 : A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1) :100–108, 1979.

T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity : the lasso and generalizations*. Chapman and Hall/CRC, 2019.

C. Hennig. Cluster-wise assessment of cluster stability. *Computational Statistics & Data Analysis*, 52(1) : 258–271, 2007.

C. Hennig. Dissolution point and isolation robustness : robustness criteria for general cluster analysis methods. *Journal of multivariate analysis*, 99(6) :1154–1176, 2008.

C. Hennig and M. Imports. Package 'fpc'. *Flexible Procedures for Clustering*, 2015.

S. Hess and W. Duivesteijn. K Is the Magic Number — Inferring the Number of Clusters Through Nonparametric Concentration Inequalities. In *EMCL-PKDD*, 2019.

C. Higuera, K. J. Gardiner, and K. J. Cios. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PloS one*, 10(6) :e0129126, 2015.

D. V. Hinkley. Inference about the change-point in a sequence of random variables. 1970.

D. P. Hofmeyr. Degrees of freedom and model selection for k-means clustering. *arXiv preprint arXiv :1806.02034*, 2018.

K. Honda, H. Araki, T. Matsui, and H. Ichihashi. A new approach to robust k-means clustering based on fuzzy principal component analysis. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 208–213. IEEE, 2008.

K. Honda, A. Notsu, and H. Ichihashi. Pca-guided k-means with variable weighting and its application to document clustering. In *International Conference on Modeling Decisions for Artificial Intelligence*, pages 282–292. Springer, 2009.

G. Hooker and L. Mentch. Please stop permuting features : An explanation and alternatives. *arXiv preprint arXiv :1905.03151*, 2019.

G. Hooker, L. Mentch, and S. Zhou. Unrestricted permutation forces extrapolation : variable importance requires at least one more model, or there is no free variable importance. *Statistics and Computing*, 31(6) : 1–16, 2021.

J. Z. Huang, M. K. Ng, H. Rong, and Z. Li. Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (5) :657–668, 2005.

Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3) :283–304, 1998.

L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 1985a.

L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1) :193–218, 1985b.

Z. Huo and G. Tseng. Integrative sparse k-means with overlapping group lasso in genomic applications for disease subtype discovery. *Ann. Appl. Stat.*, 11(2) :1011–1039, 06 2017. doi : 10.1214/17-AOAS1033. URL https://doi.org/10.1214/17-AOAS1033.

F. Hutter, H. Hoos, and K. Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *International conference on machine learning*, pages 754–762. PMLR, 2014.

B. Iooss and C. Prieur. Shapley effects for sensitivity analysis with dependent inputs : comparisons with sobol'indices, numerical estimation and applications. 2017.

L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pages 433–440, 2009.

L. Jing, M. K. Ng, and J. Z. Huang. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Transactions on Knowledge & Data Engineering*, (8) :1026–1041, 2007.

C. M. Judd, G. H. McClelland, and C. S. Ryan. *Data analysis : A model comparison approach*. Routledge, 2011.

A. Kalogeratos and A. Likas. Dip-means : an incremental clustering method for estimating the number of clusters. *Advances in neural information processing systems*, 25 :2393–2401, 2012.

M. K. Kerr and G. A. Churchill. Experimental design for gene expression microarrays. *Biostatistics*, 2001.

A. Kessy, A. Lewin, and K. Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72 (4) :309–314, 2018.

D. P. Kingma and M. Welling. An introduction to variational autoencoders. *arXiv preprint arXiv :1906.02691*, 2019.

T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1) : 59–69, 1982.

Y. Kondo, M. Salibian-Barrera, and R. Zamar. A robust and sparse k-means clustering algorithm. *arXiv preprint arXiv :1201.6082*, 2012.

Y. Kondo, M. Salibian-Barrera, and R. Zamar. Rskc : an r package for a robust and sparse k-means clustering algorithm. *Journal of Statistical Software*, 72(1) :1–26, 2016.

T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 2004.

M. H. Law, M. A. Figueiredo, and A. K. Jain. Simultaneous feature selection and clustering using mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 26(9) :1154–1166, 2004.

É. Lebarbier and T. Mary-Huard. Une introduction au critère bic : fondements théoriques et interprétation. *Journal de la Société française de statistique*, 147(1) :39–57, 2006.

E. Levine and E. Domany. Resampling method for unsupervised estimation of cluster validity. *Neural Computation*, 2001.

X. Li, Y. Wang, S. Basu, K. Kumbier, and B. Yu. A debiased mdi feature importance measure for random forests. *arXiv preprint arXiv :1906.10845*, 2019.

F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

S. P. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2) :129–137, 1982. ISSN 15579654. doi : 10.1109/TIT.1982.1056489.

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

A. Mabilia. *Dynamique non-linéaire d'une soufflante en rotation*. PhD thesis, Lyon, 2020.

J. 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.

M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, K. Hornik, M. Studer, et al. Package 'cluster'. *Dosegljivo na*, 2013.

M. Marbac and M. Sedki. Variable selection for model-based clustering using the integrated complete-data likelihood. *Statistics and Computing*, 27(4) :1049–1063, 2017.

M. Marbac, M. Sedki, and T. Patin. Variable selection for mixed data clustering : application in human population genomics. *Journal of Classification*, 37(1) :124–142, 2020.

P.-A. Mattei, C. Bouveyron, and P. Latouche. Globally sparse probabilistic pca. In *Artificial Intelligence and Statistics*, pages 976–984. PMLR, 2016.

C. Maugis, G. Celeux, and M.-L. Martin-Magniette. Variable selection for clustering with gaussian mixture models. *Biometrics*, 65(3) :701–709, 2009a.

C. Maugis, G. Celeux, and M.-L. Martin-Magniette. Variable selection in model-based clustering : A general variable role modeling. *Computational Statistics & Data Analysis*, 53(11) :3872–3882, 2009b.

S. Maurus and C. Plant. Skinny-dip : clustering in a sea of noise. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1055–1064, 2016.

G. J. McLachlan and K. E. Basford. *Mixture models : Inference and applications to clustering*, volume 38. M. Dekker New York, 1988.

G. J. McLachlan and T. Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

G. J. McLachlan, S. X. Lee, and S. I. Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6 :355–378, 2019.

M. Meila. Comparing clusterings by the Variation of Information. In *COLT*, 2003.

M. Meilă. The uniqueness of a good optimum for k-means. In *Proceedings of the 23rd international conference on Machine learning*, pages 625–632, 2006.

L. Mentch and S. Zhou. Getting better from worse : Augmented bagging and a cautionary tale of variable importance. *arXiv preprint arXiv :2003.03629*, 2020.

M. Meqqadmi, P.-E. Mosser, T. Brichler, and J. Lacaille. Reducing the impact of test bench component on the thrust margin measurement. In *Annual Conference of the PHM Society*, volume 9, 2017.

L. F. S. Merchante, Y. Grandvalet, and G. Govaert. An efficient approach to sparse linear discriminant analysis. *arXiv preprint arXiv :1206.6472*, 2012.

D. S. Modha and W. S. Spangler. Feature weighting in k-means clustering. *Machine learning*, 52(3) :217–237, 2003.

U. Möller and D. Radke. A cluster validity approach based on nearest-neighbor resampling. *Proceedings - International Conference on Pattern Recognition*, 2006.

U. Möller and D. Radke. Performance of data resampling methods for robust class discovery based on clustering. *Intelligent Data Analysis*, 2006.

L. C. Morey and A. Agresti. The Measurement of Classification Agreement : An Adjustment of the Rand Statistic for Chance Agreement. *Educational and Psychological Measurement*, 1984.

A. Mourer, F. Forest, M. Lebbah, H. Azzag, and J. Lacaille. Selecting the number of clusters k with a stability trade-off : an internal validation criterion. *arXiv preprint arXiv :2006.08530*, 2020a.

A. Mourer, J. Lacaille, M. Olteanu, and M. Chavent. Automatic detection of rare observations during production tests using statistical models. In *Annual Conference of the PHM Society*. PHM, 2020b.

S. Nembrini, I. R. König, and M. N. Wright. The revival of the gini importance ? *Bioinformatics*, 34(21) : 3711–3718, 2018.

K. K. Nicodemus. Letter to the editor : On the stability and ranking of predictors from random forest variable importance measures. *Briefings in bioinformatics*, 12(4) :369–373, 2011.

K. K. Nicodemus and J. D. Malley. Predictor correlation impacts machine learning algorithms : implications for genomic studies. *Bioinformatics*, 25(15) :1884–1890, 2009.

M. Olteanu and N. Villa-Vialaneix. On-line relational and multiple relational som. *Neurocomputing*, 147 : 15–30, 2015.

A. B. Owen and C. Prieur. On shapley value for measuring importance of dependent inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1) :986–1002, 2017.

W. Pan and X. Shen. Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 8(May) :1145–1164, 2007.

M. Y. Park, T. Hastie, and R. Tibshirani. Averaged gene expressions for regression. *Biostatistics*, 8(2) : 212–227, 2007.

D. Pelleg and A. Moore. X-means : Extending K-means with Efficient Estimation of the Number of Clusters. In *International Conference on Machine Learning (ICML)*, 2000.

T. Rabenoro. *Outils statistiques de traitement d'indicateurs pour le diagnostic et le pronostic des moteurs d'avions*. PhD thesis, Université Paris 1 Panthéon Sorbonne, 2015.

A. E. Raftery and N. Dean. Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473) :168–178, 2006.

S. Ray and R. Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*, 1999.

S. Reid and R. Tibshirani. Sparse regression and marginal testing using cluster prototypes. *Biostatistics*, 17 (2) :364–376, 2016.

S. Romano, N. X. Vinh, J. Bailey, and K. Verspoor. Adjusting for chance clustering comparison measures. *arXiv preprint arXiv :1512.01286*, 2015.

P. J. Rousseeuw. Silhouettes : A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 1987.

A. Saltelli. Making best use of model evaluations to compute sensitivity indices. *Computer physics communications*, 145(2) :280–297, 2002.

M. Sandri and P. Zuccolotto. A bias correction algorithm for the gini variable importance measure in classification trees. *Journal of Computational and Graphical Statistics*, 17(3) :611–628, 2008.

G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

E. Scornet. Trees, forests, and impurity-based variable importance. *arXiv preprint arXiv :2001.04295*, 2020.

L. Scrucca and A. E. Raftery. clustvarsel : a package implementing variable selection for gaussian model-based clustering in r. *Journal of Statistical Software*, 84, 2018.

L. Scrucca, M. Fop, T. B. Murphy, and A. E. Raftery. mclust 5 : clustering, classification and density estimation using gaussian finite mixture models. *The R journal*, 8(1) :289, 2016.

M. Sedki, G. Celeux, and C. Maugis-Rabusseau. Selvarmix : Ar package for variable selection in model-based clustering and discriminant analysis with a regularization approach. *INRIA Techical report*, 2014.

A. Sénéchal. *Réduction de vibrations de structure complexe par shunts piézoélectriques : application aux turbomachines*. PhD thesis, Paris, CNAM, 2011.

O. Shamir and N. Tishby. Cluster stability for finite samples. In *Advances in Neural Information Processing Systems*, 2007.

L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28) :307–317, 1953.

D. B. Sharma, H. D. Bondell, and H. H. Zhang. Consistent group identification and variable selection in regression with correlated predictors. *Journal of Computational and Graphical Statistics*, 22(2) :319–340, 2013.

N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2) :231–245, 2013.

M. Smolkin and D. Ghosh. Cluster stability scores for microarray data in cancer studies. *BMC bioinformatics*, 2003.

I. M. Sobol. Sensitivity analysis for non-linear mathematical models. *Mathematical modelling and computational experiment*, 1 :407–414, 1993.

J. S. Strauss, J. J. Bartko, and W. T. Carpenter. The use of clustering techniques for the classification of psychiatric patients. *British Journal of Psychiatry*, 1973.

C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn. Bias in random forest variable importance measures : Illustrations, sources and a solution. *BMC bioinformatics*, 8(1) :1–21, 2007.

C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC bioinformatics*, 9(1) :1–11, 2008.

E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3) :647–665, 2014.

T. Su and J. Dy. A deterministic method for initializing k-means clustering. In *16th IEEE international conference on tools with artificial intelligence*, pages 784–786. IEEE, 2004.

W. Sun, J. Wang, Y. Fang, et al. Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electronic Journal of Statistics*, 6 :148–167, 2012.

N. Takeishi. Shapley values of reconstruction errors of pca for explaining anomaly detection. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 793–798. IEEE, 2019.

N. Takeishi and Y. Kawahara. On anomaly interpretation via shapley values. *arXiv preprint arXiv :2004.04464*, 2020.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society : Series B (Methodological)*, 58(1) :267–288, 1996.

R. Tibshirani and G. Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 2005.

R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society Series B*, 2001a.

R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 63(2) :411–423, 2001b.

R. J. Tibshirani and B. Efron. An introduction to the bootstrap. *Monographs on statistics and applied probability*, 57 :1–436, 1993.

L. Toloşi and T. Lengauer. Classification with correlated features : unreliability of feature ranking and solutions. *Bioinformatics*, 27(14) :1986–1994, 2011.

A. Ultsch. Clustering with SOM : U*C. In *Workshop on Self Organizing Feature Maps*, 2005.

L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

A. Vijayaraghavan, A. Dutta, and A. Wang. Clustering stable instances of euclidean k-means. In *Advances in Neural Information Processing Systems*, 2017.

N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison : Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 2010.

U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 2007.

U. Von Luxburg. Clustering stability : An overview. *Foundations and Trends® in Machine Learning*, 2010.

H. Wang and C. Leng. A note on adaptive group lasso. *Computational statistics & data analysis*, 52(12) : 5277–5286, 2008.

S. Wang and J. Zhu. Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics*, 64(2) :440–448, 2008.

J. H. Ward. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301) :236–244, 1963. https://www.jstor.org/stable/2282967.

D. M. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490) :713–726, 2010. doi : 10.1198/jasa.2010.tm09415. URL https://doi.org/10.1198/jasa.2010.tm09415. PMID : 20811510.

D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3) :515–534, 2009.

D. M. Witten, A. Shojaie, and F. Zhang. The cluster elastic net for high-dimensional regression with unknown variable grouping. *Technometrics*, 56(1) :112–122, 2014.

M. Wright and A. Ziegler. ranger : A fast implementation of random forests for high dimensional data in c++ and r. *arXiv :1508.04409*, 2015.

B. Xie, W. Pan, and X. Shen. Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables. *Electronic journal of statistics*, 2 :168, 2008.

X. L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

Q. Xu, C. Ding, J. Liu, and B. Luo. Pca-guided search for k-means. *Pattern Recognition Letters*, 54 :50–55, 2015.

K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 2001.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 68(1) :49–67, 2006.

Q. Zhao, M. Xu, and P. Fränti. Extending external validity measures for determining the number of clusters. *International Conference on Intelligent Systems Design and Applications*, 2011.

H. Zhou, W. Pan, and X. Shen. Penalized model-based clustering with unconstrained covariance matrices. *Electronic journal of statistics*, 3 :1473, 2009.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society : series B (statistical methodology)*, 67(2) :301–320, 2005.