



HAL
open science

Modèles bayésiens pour l'identification de représentations antiparcimonieuses et l'analyse en composantes principales bayésienne non paramétrique

Clément Elvira

► **To cite this version:**

Clément Elvira. Modèles bayésiens pour l'identification de représentations antiparcimonieuses et l'analyse en composantes principales bayésienne non paramétrique. Traitement du signal et de l'image [eess.SP]. Ecole Centrale de Lille, 2017. Français. NNT : 2017ECLI0016 . tel-01730077v2

HAL Id: tel-01730077

<https://theses.hal.science/tel-01730077v2>

Submitted on 3 May 2018

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.

Numéro d'ordre : 330

Centrale Lille

Thèse

Présentée en vue
d'obtenir le grade de

Docteur

En

Spécialité : **Automatique, Génie informatique, Traitement du signal et
des images**

Par

CLÉMENT ELVIRA

Doctorat délivré par Centrale Lille

Titre de la thèse :

**Modèles bayésiens pour l'identification de représentations
antiparcimonieuses et l'analyse en composantes
principales bayésienne non paramétrique**

Soutenue le 10 novembre 2017 devant le jury d'examen :

Président

Florence FORBES Directrice de Recherche, INRIA, Grenoble

Rapporteurs

Rémy BOYER Maître de Conférences, HDR, Centrale Supélec

Olivier MICHEL Professeur des Universités, Grenoble INP

Examineurs

Florence FORBES Directrice de Recherche, INRIA, Grenoble

Gersende FORT Directrice de Recherche, CNRS, Toulouse

Saïd MOUSSAOUI Professeur des Universités, École Centrale de Nantes

Directeurs de thèse

Pierre CHAINAIS Professeur des Universités, Centrale Lille

Nicolas DOBIGEON Professeur des Universités, INP Toulouse

Thèse préparée dans le Laboratoire

Centre de Recherche en Informatique Signal et Automatique de Lille

Université de Lille, Centrale Lille, CNRS, UMR 9189 - CRISAL

École Doctorale SPI 072

Table des matières

INTRODUCTION	7
Modélisation et inférence bayésienne	7
Problèmes inverses et codage antiparcimonieux	9
Estimation de sous-espaces	10
Mise en page du manuscrit	11
Liste des publications	12
Plan de la thèse	13
I LOI DÉMOCRATIQUE ET CODE ANTIPARCIMONIEUX	15
1 CODAGE ANTIPARCIMONIEUX	17
1.1 Intérêt du codage antiparcimonieux	17
1.1.1 Synthèse de filtres à réponse impulsionnelle finie	18
1.1.2 Problèmes à effort minimal	18
1.1.3 Réduction d'erreur de quantification	19
1.1.4 Conversion analogique numérique	20
1.1.5 Réduction du facteur de crête	21
1.1.6 Recherche approximative de plus proches voisins	22
1.2 Acquisition compressée et parcimonie	23
1.2.1 Motivations	23
1.2.2 Liens avec la parcimonie	24
1.2.3 Reconstruction d'un signal à partir de mesures bruitées	25
1.2.4 Algorithmes pour la reconstruction	28
1.3 Théorie du codage antiparcimonieux	31
1.3.1 Pénalisation de l'amplitude maximale du code	31
1.3.2 Existence de solutions à amplitude maximale bornée	33
1.3.3 Conception de matrices de codage	36
1.4 Algorithmes de recherche de codes antiparcimonieux	37
1.4.1 Solution du problème à effort minimal	37
1.4.2 Recherche d'une représentation de Kashin	38
1.4.3 Algorithme PFA	38
1.4.4 Algorithme FITRA	40

1.5 Discussion	40
2 MODÈLE BAYÉSIEN POUR LE CODAGE ANTIPARCIMONIEUX	43
2.1 La loi démocratique	44
2.1.1 Propriétés de la loi démocratique	44
2.1.2 Méthodes d'échantillonnage	50
2.1.3 Simulations numériques	54
2.2 Modèle bayésien	55
2.2.1 Modèle bayésien hiérarchique	55
2.2.2 Algorithmes MCMC : échantillonneurs de Gibbs	57
2.2.3 Estimateurs	61
2.3 Performances à paramètres de nuisance fixés	63
2.3.1 Protocole expérimental	63
2.3.2 Performances numériques	64
2.3.3 Mélange des chaînes de Markov	65
2.3.4 Calibration du paramètre μ	65
2.4 Réduction du facteur de crête (PAPR)	67
2.4.1 Modélisation du problème	67
2.4.2 Pertinence du modèle bayésien	68
2.4.3 Comparaison avec des algorithmes supervisés	70
2.4.4 Performances en fonction du rapport des dimensions	72
2.5 Conclusion	73
2.6 Perspectives	74
2.6.1 Améliorer le modèle et consolider les expériences numériques	74
2.6.2 Concevoir un estimateur plus efficace	74
2.6.3 Changer la géométrie du problème	75
2.6.4 Étudier la loi démocratique renormalisée	75
II MODÈLE BAYÉSIEN NON PARAMÉTRIQUE POUR L'ESTIMATION DE SOUS-ESPACES	77
3 MÉTHODES BAYÉSIENNES NON PARAMÉTRIQUES POUR LES MODÈLES À VARIABLES LATENTES	79
3.1 Classification non supervisée et processus de Dirichlet à mélange	80
3.1.1 Modèles de mélange bayésiens	80
3.1.2 Modèle de mélange fini bayésien et loi sur les partitions	81
3.1.3 Construction du processus de Dirichlet à mélange	83
3.1.4 Manipulation du processus de Dirichlet à mélange	87
3.1.5 Inférence et classification non supervisée	92
3.2 Modèles à variables latentes et processus Beta Bernoulli	96
3.2.1 Modèles à variables latentes	96
3.2.2 Processus Beta-Bernoulli	97

3.2.3	<i>Simulation et inférence</i>	99
3.3	Discussion	104
4	ESTIMATION BAYÉSIENNE NON PARAMÉTRIQUE DE SOUS-ESPACES	107
4.1	Analyse en composantes principales et formulations probabilistes	108
4.1.1	<i>Vers une formulation probabiliste de la recherche de sous-espaces</i>	108
4.1.2	<i>Estimer la dimension du sous-espace</i>	109
4.1.3	<i>Positionnement du problème et objectif du chapitre</i>	111
4.2	Modèle BNP-PCA pour l'apprentissage de sous-espaces	112
4.2.1	<i>Modèle</i>	112
4.2.2	<i>Vraisemblance et choix des lois a priori</i>	112
4.3	Algorithme MCMC	114
4.3.1	<i>Loi a posteriori marginalisée</i>	114
4.3.2	<i>Algorithme MCMC</i>	115
4.4	Définition et analyse théorique des estimateurs	119
4.4.1	<i>Comportement des estimateurs MAP de la dimension</i>	120
4.4.2	<i>Un test statistique pour déterminer la dimension du sous-espace</i>	121
4.4.3	<i>Estimation des autres paramètres</i>	125
4.5	Validation expérimentale	126
4.5.1	<i>Un cas simple</i>	126
4.5.2	<i>Comportement des estimateurs de la dimension</i>	127
4.5.3	<i>Cas d'un signal constitué uniquement de bruit</i>	129
4.5.4	<i>Détection de signaux en dessous du niveau de bruit</i>	130
4.5.5	<i>Conclusion partielle</i>	131
4.6	Applications	132
4.6.1	<i>BNP-PCA et classification non supervisée</i>	132
4.6.2	<i>Sous-espace d'un cube hyperspectral</i>	134
4.7	Conclusion	137
4.8	Perspectives	138
4.8.1	<i>Aspect numérique et temps de calculs</i>	138
4.8.2	<i>Généraliser les résultats concernant les estimateurs</i>	139
4.8.3	<i>Améliorer le modèle</i>	139
4.8.4	<i>Approfondir l'application à l'imagerie hyperspectrale</i>	140
4.8.5	<i>Modéliser des sous-espaces fonctionnels</i>	141
	CONCLUSION	143
	Une formulation bayésienne du codage antiparcimonieux	143
	Une ACP bayésienne non paramétrique	145

ANNEXES 149

A LOI DÉMOCRATIQUE ET CODE ANTIPARCIMONIEUX 151

- A.1 Algorithme PFA 151
- A.2 Moments 153
 - A.2.1 Moments de la loi démocratique 153
 - A.2.2 Moments positifs de la loi démocratique 153
- A.3 Lois marginales 153
 - A.3.1 Loi des sous-ensembles 154
 - A.3.2 Convergence des marginales 156
- A.4 Lois conditionnelles 156
 - A.4.1 Divers résultats associés aux cônes 156
 - A.4.2 Loi conditionnelle d'une composante 158
- A.5 Lois conditionnelles *a posteriori* d'une composante 159
 - A.5.1 Partie uniforme 159
 - A.5.2 Parties exponentielles 160
- A.6 Densité de la loi démocratique renormalisée 161

B ESTIMATION BAYÉSIENNE NON PARAMÉTRIQUE DE SOUS-ESPACES 163

- B.1 Échantillonnage sur la variété de Stiefel 163
 - B.1.1 Loi uniforme 163
 - B.1.2 Distribution matricielle Bingham 163
- B.2 Loi Inverse Gamma translée 166
- B.3 Loi non informative de Jeffreys pour la loi *a posteriori* marginalisée 167
- B.4 Calcul de la loi *a posteriori* marginalisée 169
- B.5 Preuves des résultats d'inconsistance 170
 - B.5.1 Résultat général 170
 - B.5.2 Inconsistance sévère 173
- B.6 Loi des produits scalaires 174

Introduction

CETTE THÈSE de doctorat en traitement du signal et de l'image a débuté au sein du Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS) qui est devenu le 1^{er} janvier 2015 le Centre de Recherche en Informatique, Signal et Automatique de Lille (CRISAL). Elle a été réalisée en partenariat avec l'Institut de Recherche en Informatique de Toulouse (IRIT). Elle a été financée par le projet ANR-BNPSI (*Bayesian Nonparametric methods for Signal and Image Processing*) n° ANR-13-BS-03-0006-01.

La prochaine partie introduit les grands principes de la méthodologie bayésienne qui sera utilisée dans ce travail comme outil d'inférence. Les deux parties suivantes discutent les problèmes abordés que sont le codage antiparcimonieux et l'estimation de sous-espaces.

MODÉLISATION ET INFÉRENCE BAYÉSIENNE

UN DES OBJECTIFS de la statistique est de caractériser un phénomène aléatoire à partir de l'observation d'un nombre fini de réalisations, appelées observations, à des fins de prédiction ou de prise de décision. Le phénomène est décrit par une distribution de probabilité appelée *vraisemblance* dont les paramètres, notés θ , sont inconnus.

Dans cette thèse, les observations sont des vecteurs réels¹ notées \mathbf{y} et indexées par n . La densité de la distribution de probabilité est notée f , et

$$\mathbf{y} \sim f(\mathbf{y}|\theta). \quad (\text{I.1})$$

Un modèle statistique est défini² comme l'observation d'une variable aléatoire \mathbf{y} distribuée selon $f(\mathbf{y}|\theta)$, où le paramètre θ est inconnu. Si θ appartient à un espace de dimension finie, le modèle est dit *paramétrique*. À l'inverse, si θ appartient à un espace de dimension infinie, le modèle est dit *non paramétrique*. Nous insistons sur deux points. D'abord, le concept de modèle non paramétrique ne signifie pas « modèle sans paramètre ». Ensuite, les modèles ne sont généralement pas à l'origine des observations. Ce sont des *modèles d'analyse*, à ne pas confondre avec les *modèles de synthèse*.

L'incertitude autour du paramètre θ varie selon les problèmes. Par exemple, les commentaires d'un expert métier peuvent aider à réduire l'espace des paramètres à explorer. Une modélisation de cette information *a priori* est de considérer le paramètre θ aléatoire, et d'intégrer l'information dans sa distribution appelée loi *a priori* que l'on notera π . Un modèle

1. de \mathbb{R}^M dans la partie I et de \mathbb{R}^D dans la partie II.

2. Robert (2006), Définition 1.7.

bayésien est défini³ comme l'observation d'une variable aléatoire \mathbf{y} distribuée selon $f(\mathbf{y}|\theta)$, et d'une distribution *a priori* π sur le paramètre inconnu θ . On fait à nouveau la distinction entre modèle paramétrique et non paramétrique. L'inférence se fait désormais à partir de la loi *a posteriori* obtenue avec la formule de Bayes

$$p(\theta|\mathbf{y}) = \frac{f(\mathbf{y}|\theta)\pi(\theta)}{\int p(\mathbf{y},\theta)d\theta} \propto f(\mathbf{y}|\theta)\pi(\theta). \quad (\text{I.2})$$

La quantité $\int p(\mathbf{y},\theta)d\theta$ est appelée *évidence*. On la note $p(\mathbf{y}|\mathcal{M})$ où \mathcal{M} est le modèle bayésien mais elle est la plupart du temps abrégée $p(\mathbf{y})$.

L'évidence sert à comparer la qualité de plusieurs modèles bayésiens.

Notons que la loi *a posteriori* résume toute l'information disponible sur le paramètre à travers un compromis entre la loi *a priori* et la vraisemblance. On peut voir la loi *a priori* comme un terme de régularisation. Par exemple, nous introduirons dans le chapitre 2 une loi *a priori* qui pénalise les vecteurs à forte amplitude de la forme $\pi(\theta) \propto \exp(-\lambda\|\theta\|_\infty)$.

L'inférence du paramètre θ se fait via la définition d'estimateurs bayésiens. La conception de tels estimateurs dépend de la tâche à accomplir et s'appuie sur la loi *a posteriori* $p(\theta|\mathbf{y})$. Il existe deux estimateurs très couramment utilisés.

L'estimateur du *maximum a posteriori* (MAP) est défini comme le mode de la distribution *a posteriori* $p(\theta|\mathbf{y})$

$$\hat{\theta}_{\text{MAP}} = \arg \max_{\theta} p(\theta|\mathbf{y}) = \arg \max_{\theta} f(\mathbf{y}|\theta)\pi(\theta). \quad (\text{I.3})$$

Cet estimateur cherche le meilleur compromis entre la plausibilité des observations (la vraisemblance) et l'information apportée par l'expert (*a priori*). Nous verrons dans le chapitre 2 que cet estimateur peut aussi être interprété comme la solution de la formulation variationnelle d'un problème contraint.

L'estimateur du *minimum de l'erreur quadratique moyenne* (MMSE) est défini comme l'estimateur minimisant l'erreur

$$\text{MSE} = \mathbb{E}_{\theta} \left[\|\theta - \hat{\theta}\|^2 \right], \quad (\text{I.4})$$

et de simples calculs montrent qu'il correspond à la moyenne *a posteriori*

$$\hat{\theta}_{\text{MMSE}} = \mathbb{E}[\theta|\mathbf{y}]. \quad (\text{I.5})$$

Dans la plupart des cas, les estimateurs bayésiens sont trop compliqués pour être calculés explicitement. Une solution consiste à utiliser les méthodes de *Monte Carlo par chaîne de Markov* (MCMC) pour générer une suite d'échantillons $\{\theta^{(t)}\}_{t=1,\dots,T}$ asymptotiquement distribués suivant $p(\theta|\mathbf{y})$. Il existe une multitude d'algorithmes MCMC pour générer ces échantillons, tels que les algorithmes de *Gibbs* et de *Metropolis-Hastings*. Nous utiliserons également dans ce travail une méthode d'échantillonnage couplant une méthode d'optimisation proximale avec un échantillonneur pour générer des échantillons suivant une loi dont la densité est non différentiable (Pereyra, 2016a). Les estimateurs bayésiens sont ensuite numériquement approchés avec cette suite d'échantillons.

3. Robert (2006), Définition 1.8.

Nous terminons cette courte introduction de la méthodologie bayésienne par une discussion concernant les méthodes non paramétriques. Pour rappel, les modèles non paramétriques sont décrits par une infinité (dénombrable) de paramètres. L'existence des tels modèles remonte aux années 70 (Ferguson, 1973). Ces outils sont très utilisés par la communauté *apprentissage automatique* (Orbanz et Teh, 2010) mais encore relativement peu par celle du *traitement du signal et des images*. Le nombre de paramètres est désormais infini. Toutefois, seul un nombre fini est nécessaire en pratique pour décrire les observations et la modélisation bayésienne pénalise naturellement cette quantité. On obtient des modèles qui s'enrichissent lorsque le nombre d'observations augmente. Nous verrons au chapitre 3 à travers les processus de Dirichlet pour les modèles de mélange et le processus Beta-Bernoulli pour les modèles à variables latentes que la construction de modèles non paramétriques repose sur un socle théorique plus important. Malgré une construction assez mathématique, *la bonne nouvelle* est que nous aboutirons à des algorithmes simples à utiliser.

PROBLÈMES INVERSES ET CODAGE ANTIPARCIMONIEUX

LES PROBLÈMES INVERSES sont une classe de problèmes pour lesquels on cherche à retrouver les causes non directement mesurables d'un phénomène à partir d'observés. Mais la plupart du temps, ces observations ne suffisent pas pour déterminer mathématiquement la solution en raison de l'absence d'unicité par exemple. On dit alors que le problème est *mal posé*. On doit le régulariser en ajoutant des contraintes sur la solution afin de la rendre unique ou, au moins, restreindre l'espace des solutions admissibles. En théorie du codage, on recherche par exemple des codes robustes aux déformations qui apparaissent lors de la transmission. Dans ce but, on utilise des représentations $\mathbf{h}_1, \dots, \mathbf{h}_N$ redondantes pour coder les signaux \mathbf{y} . Si le codage est linéaire, on écrit

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}, \quad (1.6)$$

où \mathbf{x} est le code, \mathbf{H} la matrice d'encodage concaténant les vecteurs \mathbf{h}_n et \mathbf{e} représente un terme d'erreur additif. En raison de la redondance, plusieurs codes sont associés à la même observation. Il est alors courant d'ajouter une contrainte de parcimonie sur le code : on cherche le code \mathbf{x} ayant le plus de coefficients égaux à 0.

À l'inverse, un code est dit antiparcimonieux lorsque l'énergie du signal \mathbf{y} est répartie le plus uniformément possible sur toutes les composantes x_n du code. Pour des raisons de robustesse, on veut idéalement que toutes les composantes valent exactement $x_n = \pm \|\mathbf{y}\|_2 / \sqrt{N}$. Mais de tels codes n'existent pas toujours selon les propriétés de la matrice d'encodage \mathbf{H} . Nous verrons au chapitre 1 comment on peut construire des matrices d'encodage favorisant l'étalement. Souvent étudiée sous d'autres appellations, l'antiparcimonie n'est pas aussi populaire que la parcimonie. Le chapitre 1 détaille quelques applications en traitement du signal.

La façon standard d'obtenir un code antiparcimonieux est de résoudre le problème contraint

$$\mathbf{x} = \arg \min_{\mathbf{z} \in \mathbb{R}^N} \|\mathbf{z}\|_\infty \quad \text{s.c.} \quad \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2 \leq \varepsilon, \quad (\mathbf{P}_\infty^\varepsilon)$$

où ε est une erreur de reconstruction tolérée. Ici, le *problème inverse linéaire* ($\mathbf{P}_\infty^\varepsilon$) est régularisé en ajoutant une contrainte sur la norme ℓ_∞ .

Nous verrons au chapitre 1 pourquoi il est intéressant de minimiser cette norme plutôt qu'une autre quantité.

Nous soulignons que conceptuellement, le codage antiparcimonieux n'est pas un problème inverse. On ne cherche pas à retrouver un code ayant servi à la synthèse ; on cherche seulement un code qui explique bien l'observation avec de bonnes propriétés de robustesse.

À ce jour, aucune formulation bayésienne du codage antiparcimonieux n'a été proposée. Pourtant, les avantages d'une telle formulation sont multiples : liberté dans la modélisation via le choix de loi *a priori*, aucun réglage de paramètres, conception d'algorithmes génériques grâce aux méthodes MCMC, définition d'estimateurs sans équivalents pour les méthodes d'optimisation, *etc.*

Nous proposons au chapitre 2 le modèle bayésien BAC-1 pour le codage antiparcimonieux. Ce modèle s'appuie sur une nouvelle loi de probabilité que nous étudierons en détails puis que nous choisirons comme *a priori* sur les codes pour favoriser l'antiparcimonie. Nous verrons trois algorithmes MCMC dont deux s'appuyant sur des méthodes de Monte Carlo proximales pour approcher des estimateurs bayésiens. Ces algorithmes ne nécessitent aucun réglage préalable de paramètres, contrairement à la plupart des modèles d'optimisation. La pertinence du modèle sera évaluée sur des exemples numériques et une application en traitement du signal.

ESTIMATION DE SOUS-ESPACES

LA MULTIPLICATION des sources d'information amène une augmentation de la taille des données, avec des conséquences pratiques sur le stockage et le traitement. Pourtant, ces données vivent dans certaines situations dans un sous-espace de plus petite dimension qu'il faut identifier. Dans les autres cas, il est par exemple plus intéressant de perdre de l'information en réduisant la dimension pour gagner en vitesse d'exécution des algorithmes et en capacité de stockage.

L'estimation de sous-espaces peut être formulée sous l'angle de la *factorisation de matrices*, selon

$$\mathbf{Y} = \mathbf{P}\mathbf{W} + \mathbf{E} \quad (\text{I.7})$$

où \mathbf{P} est une famille génératrice du sous-espace, \mathbf{W} les coefficients sur la famille et \mathbf{E} le terme d'erreur. En un sens, il s'agit aussi d'un problème inverse linéaire, que l'*analyse en composantes principales* (ACP) régularise en cherchant la base orthonormale maximisant la variance dans chaque direction. Le choix de la dimension du sous-espace modélise le compromis à réaliser entre fidélité aux données et compression de l'information.

Pour l'estimer, il faut pénaliser le nombre de paramètres en utilisant par exemple des critères d'information ou des lois *a priori*.

Nous discutons au chapitre 3 la conception de modèles bayésiens non paramétriques grâce à deux exemples. Nous étudions d'abord les *processus de Dirichlet à mélange* pour la classification non supervisée. Nous présentons aussi le processus *Beta-Bernoulli* en faisant le lien entre la classification non supervisée et les modèles à variables latentes avec contrainte de parcimonie. Nous aboutirons *au processus du buffet indien* qui s'interprète comme une loi sur les matrices binaires de taille potentiellement infinie.

Dans le chapitre 4, nous proposons d'explorer le fort potentiel des méthodes bayésiennes non paramétriques pour estimer le sous-espace d'un ensemble de données. En effet, les méthodes non paramétriques fournissent une solution élégante et numériquement plus simple à implémenter pour ne pas fixer à l'avance la dimension du sous-espace. En interprétant l'ACP comme un problème de factorisation de matrices, nous utilisons le processus du buffet indien comme *a priori* sur l'utilisant des composantes de la famille génératrice du sous-espace. Nous couplons cette *a priori* à une loi uniforme sur les matrices orthonormales pour modéliser un sous-espace. Nous décrivons un algorithme MCMC pour approcher un estimateur de la base du sous-espace et de sa dimension. Mais nous verrons que les estimateurs de la dimension du sous-espace sont à manipuler avec prudence si l'objectif est de retrouver une dimension ayant un sens physique. On attend d'un estimateur qu'il soit consistant, c'est-à-dire qu'il se concentre autour de la bonne valeur lorsqu'un modèle de synthèse existe. Nous donnerons deux résultats théoriques d'inconsistance concernant l'estimateur MAP de la dimension et réaliserons une étude numérique de la consistance de l'estimateur MAP marginalisé. Nous proposerons aussi un estimateur alternatif exploitant des tests statistiques sur la distribution *a posteriori* des paramètres. Nous validerons le modèle et étudierons les estimateurs sur des exemples synthétiques. Nous appliquerons pour finir la méthode à un problème de classification non supervisée et d'imagerie hyperspectrale.

MISE EN PAGE DU MANUSCRIT

LE LECTEUR aura déjà remarqué qu'une large marge est laissée libre sur le côté droit de chaque page du manuscrit. Nous l'utiliserons pour insérer des commentaires, des notes historiques ainsi que des figures et des tableaux afin de compléter le propos. Cette charte graphique est inspirée du travail de Tufte (2001) et réalisée à l'aide de la classe latex TUFTE-LATEX. Nous soulignons que la présence du logo GitHub cliquable dans la marge indique la disponibilité en ligne des codes.



LISTE DES PUBLICATIONS

LA VALORISATION des résultats en codage antiparcimonieux s'est faite en trois étapes. Le modèle BAC-1 a d'abord été présenté à SSP'16 (Elvira et al., 2016). La loi démocratique et les performances numériques des algorithmes ont été étudié en détails dans IEEE TSP (Elvira et al., 2017a). Nous avons récemment présenté une version améliorée de l'échantillonneur proximal au Grets 2017 (Elvira et al., 2017d).

La valorisation des résultats pour l'estimation non paramétrique de sous-espaces s'est faite en deux temps. Le modèle BNP-PCA et l'échantillonneur de Gibbs ont d'abord été présentés à ICASSP'17 (Elvira et al., 2017c). Une étude théorique des estimateurs suivie d'expériences numériques sont soumis dans une revue (Elvira et al., 2017b).

Revue

[A1] « Bayesian Antispase Coding », dans *IEEE Transactions on Signal Processing*, 65 n° 7 (avril) : 1660-1672.

[A2] « Bayesian nonparametric principal component analysis », soumis en juillet 2017.

Conférences internationales

[C1] « Democratic prior for antispase coding », dans *Proc. IEEE Statistical Signal Processing Workshop (SSP)*, 1-4. Juin 2016.

[C2] « Bayesian nonparametric subspace estimation », dans *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, mars 2017.

Conférence nationale

[N1] « Une formulation bayésienne du codage antiparcimonieux », dans *Actes du Colloque GRETSI*, septembre 2017.

PLAN DE LA THÈSE

CE MANUSCRIT est constitué de deux parties de deux chapitres chacune.

La partie I étudie la formulation bayésienne du codage antiparcimonieux.

Le chapitre 1 présente le problème du codage antiparcimonieux en commençant par quelques exemples issus de la littérature. Nous discutons les résultats théoriques concernant les codes antiparcimonieux, la conception de matrices d'encodage et les algorithmes d'optimisation pour obtenir ces codes. Un parallèle est réalisé avec les résultats et méthodes en acquisition compressée.

Le chapitre 2 présente la formulation bayésienne du codage antiparcimonieux. Cette étude nécessite d'introduire et d'étudier une nouvelle loi de probabilité, nommée *loi démocratique*. Nous présentons trois algorithmes non supervisés utilisant des approches de Monte Carlo par chaînes de Markov (MCMC) pour approcher des estimateurs. Après avoir étudié les performances sur des exemples synthétiques, les résultats sont comparés à ceux d'algorithmes supervisés issus de l'état de l'art. Le chapitre se termine par une discussion des perspectives numériques, méthodologiques et applicatives du travail.

La partie II étudie un modèle bayésien non paramétrique pour l'estimation de sous-espaces.

Le chapitre 3 présente les fondamentaux de la modélisation bayésienne non paramétrique à travers deux problèmes classiques en apprentissage automatique : la classification non supervisée et les modèles à variables latentes. Les processus de Dirichlet et Beta-Bernoulli sont introduits. La conception d'algorithmes d'inférence est discutée pour les processus du restaurant chinois et du buffet indien.

Le chapitre 4 décrit le modèle BNP-PCA proposé pour l'estimation de sous-espaces sans fixer à l'avance sa dimension en utilisant le processus du buffet indien. Ce modèle est motivé par une revue des principales méthodes statistiques d'estimation de sous-espaces et de leur dimension. Un algorithme MCMC est décrit pour approcher des estimateurs. Deux théorèmes d'inconsistance des estimateurs du maximum *a posteriori* sont discutés, et un nouvel estimateur utilisant des tests statistiques sur les distributions *a posteriori* est proposé. La méthode est testée et étudiée empiriquement sur des exemples synthétiques. La méthode est ensuite appliquée à un problème de classification supervisée et à l'estimation de sous-espaces en imagerie hyperspectrale. Le chapitre se termine par une discussion des perspectives méthodologiques et applicatives du travail.

La conclusion rappelle les perspectives théoriques, méthodologiques et numériques présentées à la fin des chapitres 2 et 4.

Première partie

LOI DÉMOCRATIQUE

ET CODE ANTIPARCIMONIEUX

CE CHAPITRE discute la notion de codage antiparcimonieux. Un code antiparcimonieux est une représentation d'un signal à l'aide d'une matrice de codage, où l'énergie est répartie le plus uniformément possible sur toutes les composantes (Lyubarskii et Vershynin, 2010). L'intérêt de ce codage réside dans sa robustesse aux déformations. Nous verrons dans la partie 1.1 quelques applications tirées de la littérature en traitement du signal (Cadzow, 1971; Ilic et Strohmer, 2009; Jégou et al., 2012).

Toutes les matrices d'encodage ne garantissent pas l'existence de telles représentations, et les garanties diffèrent selon les propriétés de ces matrices. Des problématiques similaires sont rencontrées en *acquisition compressée*, où la matrice de codage est remplacée par une matrice d'acquisition. Mais il existe deux différences fondamentales entre les deux problèmes. Le but du codage antiparcimonieux est de répartir l'information uniformément sur tous les coefficients, contrairement à l'acquisition compressée où l'on souhaite la concentrer sur quelques coefficients non nuls. L'acquisition compressée repose sur l'hypothèse que le signal à retrouver est intrinsèquement parcimonieux tandis que le caractère antiparcimonieux d'un code n'a pas d'origine physique.

Malgré ces différences, la recherche de codes antiparcimonieux est posée comme un problème inverse : on cherche à minimiser une erreur de reconstruction en pénalisant les écarts d'amplitude entre les coefficients. Avant d'introduire le problème du codage antiparcimonieux, nous commencerons par discuter dans la partie 1.2 du problème d'acquisition compressée, déjà bien compris et étudié. Cette étude nous permettra de réaliser dans la partie 1.3, un parallèle entre les résultats théoriques des deux problèmes, puis algorithmique dans la partie 1.4.

1.1 INTÉRÊT DU CODAGE ANTIPARCIMONIEUX

CETTE PARTIE illustre l'intérêt des codes *antiparcimonieux* à travers quatre applications. Pour l'instant, on désigne par code *antiparcimonieux* une représentation \mathbf{x} d'une mesure \mathbf{y} où l'énergie est répartie le plus uniformément possible sur toutes les composantes. On appelle « répartir uniformément l'énergie » le fait qu'idéalement, chaque composante x_n de \mathbf{x} soit la forme $|x_n| = \pm \|\mathbf{y}\|_2 / \sqrt{N}$. Les concepts clés autour des codes *antiparcimonieux* seront définis dans la partie 1.3 du même chapitre.

Un code *antiparcimonieux* est obtenu en résolvant par exemple un problème d'optimisation sous contraintes. Puisque l'on désire que les coefficients du code \mathbf{x} soient similaires voire égaux en valeur absolues, il suffit de pénaliser l'amplitude des coefficients, à savoir $x_{\max} - x_{\min}$. Ici,

Pour orthographier les mots *antiparcimonie* et *antiparcimonieux*, nous avons choisi de suivre les recommandations du Dictionnaire de l'Académie française qui déconseille l'utilisation d'un trait d'union entre le préfixe « anti » et le nom ou l'adjectif qui le suit (Lacroux, 2007). Cependant, il est toléré d'utiliser un trait d'union pour les mots non encore lexicalisés, ce qui est le cas ici. C'est pour cette raison que l'on trouve différentes orthographes dans la littérature française (Vaïter et al., 2013; Mathon et al., 2013).

En résumé, on cherche une représentation \mathbf{x} d'un vecteur \mathbf{y} sur une matrice \mathbf{H} avec idéalement $x_n = \pm \alpha$, où α est un réel positif.

x_{\max} et x_{\min} désignent respectivement les valeurs maximales et minimales de la valeur absolue des coefficients, *i.e.*, $|x|$. Bien que la notion de codage *antiparcomionieux* soit récente (Lyubarskii et Vershynin, 2010), ce type de problème est étudié depuis le milieu du xx^e siècle.

1.1.1 Synthèse de filtres à réponse impulsionnelle finie

Historiquement, la recherche d'une solution à amplitude minimale en traitement du signal est apparue pour la conception de *filtre à réponse impulsionnelle finie* (RIF) (Parks et McClellan, 1972; McClellan et Parks, 2005). Un RIF est un filtre linéaire dont la réponse impulsionnelle est de durée finie. Pour un filtre numérique, cela signifie que sa réponse n'est caractérisée que par un nombre fini de valeurs du signal d'entrée. Les avantages sont multiples : stabilité, complexité, *etc.* La synthèse de filtres RIF était un sujet de recherche très actifs dans les années 1970.

La conception d'un filtre RIF peut toujours se ramener à l'étude d'un filtre passe-bas (Oppenheim et al., 1999). La méthode proposée par Parks et McClellan (1972) consiste à minimiser un critère d'erreur entre la courbe réelle et le filtre idéal. La figure 1.1 illustre le problème d'adéquation entre le filtre idéal et une fonction polynomiale. Soit \mathbf{D} un filtre numérique passe-bas idéal qui laisse passer les fréquences contenues dans une bande $\mathcal{F}_p \triangleq [0, F_p]$ et coupe les fréquences contenues dans une bande $\mathcal{F}_c \triangleq [F_c, +\infty[$. La méthode de Parks et McClellan propose de minimiser l'amplitude maximale entre ce filtre idéal et une approximation polynomiale, facile à implémenter. L'amplitude maximale est pénalisée à travers la norme ℓ_∞ . Notons \mathbf{W} un vecteur de poids et \mathbf{H} le filtre recherché. La fonction de coût à minimiser devient

$$\mathbf{H} = \arg \min_{\mathbf{F}} \left\| \mathbf{W} | \mathbf{D} - \mathbf{F} \right\|_{\infty}. \quad (1.1)$$

et l'interpolation s'appuie sur l'algorithme de Remez¹ (Remes, 1936). En somme, l'objectif est d'identifier le filtre dont l'erreur d'approximation du filtre idéal est antiparcomionieuse. La matrice \mathbf{W} permet d'ajuster l'importance de l'erreur sur chaque composante.

1.1.2 Problèmes à effort minimal

De nombreux systèmes sont cinématiquement redondants. Par exemple, plusieurs moteurs peuvent être impliqués dans le mouvement d'un bras robotisé. Pour des raisons techniques, il est parfois plus intéressant de minimiser l'amplitude maximale d'une grandeur, telles que l'accélération, la déformation, *etc.*, plutôt que l'énergie utilisée pour engendrer le mouvement. On parle dans ce cas de *solution à effort minimal*.

Considérons l'équation d'état

$$\mathbf{y}_{n+1} = \mathbf{A}\mathbf{y}_n + \mathbf{B}\mathbf{x}_{n+1}, \quad (1.2)$$

où \mathbf{y}_n élément de \mathbb{R}^M est le vecteur d'état au temps n , \mathbf{x}_n élément de \mathbb{R}^P est le vecteur d'action, P le nombre de moteurs et \mathbf{A} , \mathbf{B} deux matrices

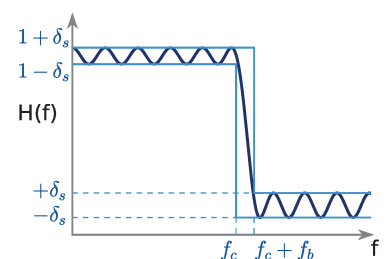


FIGURE 1.1 : Caractéristique d'un filtre passe bas.

1. L'orthographe du nom diffère entre la publication *originale* où l'auteur a traduit son nom russe par Eugène Remes, et l'algorithme traditionnellement nommé Remez par la communauté traitement du signal.

régissant la dynamique du système. L'objectif est d'aller en T actions d'un état initial \mathbf{y}_0 vers un état désiré \mathbf{y}_d . On recherche la suite d'action (\mathbf{x}_n) à réaliser ayant l'amplitude la plus faible. L'équation (1.2) se réécrit sous la forme matricielle suivante

$$\mathbf{y}_d - \mathbf{A}\mathbf{y}_0 = \mathbf{H}\mathbf{x}, \quad (1.3)$$

où \mathbf{u} est un vecteur de dimension $N = TP > P$ où l'on a concaténé toutes les actions \mathbf{x}_n , et \mathbf{H} est une matrice résumant la physique du système.

Une solution à *effort minimal* au problème (1.3) peut s'obtenir en résolvant un système de la forme (Cadzow, 1971 ; Deo et Walker, 1997 ; Zhang et al., 2002)

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_{\infty} \quad \text{s.c.} \quad \mathbf{y}_d - \mathbf{A}\mathbf{y}_0 = \mathbf{H}\mathbf{x}. \quad (1.4)$$

La résolution (1.4) peut s'interpréter comme la recherche d'un code antiparcimonieux.

1.1.3 Réduction d'erreur de quantification

En traitement du signal, la quantification est le procédé qui permet d'approcher un signal par des coefficients choisis dans un ensemble discret (généralement dénombrable) de valeurs possibles, appelé dictionnaire. L'erreur de quantification est la différence entre le signal réel et sa version approchée. Le choix du dictionnaire influe sur l'erreur de quantification, et dépend de l'application. Ce dictionnaire peut être uniforme (à pas constant), à zone morte (à pas variable), *etc* (Cvetkovic, 2003).

On montre facilement que l'on peut améliorer la robustesse d'une représentation en utilisant un dictionnaire redondant (Lyubarskii et Vershynin, 2010). Pour illustrer ce résultat, comparons l'erreur de quantification réalisée avec deux dictionnaires différents. Le premier dictionnaire est uniforme, tandis que le second autorise un codage antiparcimonieux. Nous utiliserons pour cela la notion de *trame ajustée*.

Définition 1.1. *Un dictionnaire $(\mathbf{u}_n)_{n \in [1..N]}$ est appelé trame s'il existe deux constantes réelles positives A, B telles que pour tout vecteur \mathbf{y} de \mathbb{R}^M*

$$A \|\mathbf{y}\|_2^2 \leq \sum_{n=1}^N \langle \mathbf{y}, \mathbf{u}_n \rangle^2 \leq B \|\mathbf{y}\|_2^2. \quad (1.5)$$

Si de plus $A = B = 1$, alors la trame est dite *ajustée* ou de Parseval.

Notons que si la trame (\mathbf{u}_n) est ajustée, alors pour tout vecteur \mathbf{y} de \mathbb{R}^M

$$\mathbf{y} = \sum_{n=1}^N a_n \mathbf{u}_n \quad \text{où } a_n = \langle \mathbf{y}, \mathbf{u}_n \rangle, \quad (1.6)$$

$$\|\mathbf{y}\|_2^2 = \sum_{n=1}^N \langle \mathbf{y}, \mathbf{u}_n \rangle^2. \quad (1.7)$$

Les deux dernières équations montrent en quoi les trames ajustées généralisent la notion de base d'un espace vectoriel à une famille non libre.

Pour obtenir cette équation, il faut faire émerger \mathbf{y}_T en fonction de \mathbf{y}_0 par récursivité à partir de l'équation (1.2).

Pour \mathbf{y} élément de \mathbb{R}^M tel que $\|\mathbf{y}\|_2 \leq 1$, toute trame ajustée $(\mathbf{u}_n)_{n \in [1 \dots N]}$, on note $(a_n^{\mathbf{u}})_n$ les coefficients $a_n^{\mathbf{u}} \triangleq \langle \mathbf{y}, \mathbf{u}_n \rangle$. On note $(\hat{a}_n^{\mathbf{u}})_n$ les coefficients quantifiés avec L niveaux, et $\hat{\mathbf{y}}^{\mathbf{u}}$ le vecteur \mathbf{y} après quantification $\hat{\mathbf{y}}^{\mathbf{u}} = \sum_{n=1}^N \hat{a}_n^{\mathbf{u}} \mathbf{u}_n$. Nous allons montrer que l'on peut espérer diminuer l'erreur de quantification avec un dictionnaire permettant un codage antiparcimonieux. Ce raisonnement est illustré à la figure 1.2.

Soit (\mathbf{v}_n) une trame ajustée vérifiant les équations (1.6) et (1.7). Comme $\|\mathbf{y}\|_2 \leq 1$, les coefficients $a_n^{\mathbf{v}}$ appartiennent à $[-1, 1]$ (cf. Éq (1.7)) et une quantification à L niveaux amène une erreur de quantification d'au plus $2/(L-1)$ par coefficient. D'où la borne sur l'erreur de quantification globale

$$\|\mathbf{y} - \hat{\mathbf{y}}^{\mathbf{v}}\|_2 = \left(\sum_{n=1}^N |a_n^{\mathbf{v}} - \hat{a}_n^{\mathbf{v}}|^2 \right)^{1/2} \leq \sqrt{N} \frac{2}{L-1}. \quad (1.8)$$

Répetons la même opération avec une trame ajustée (\mathbf{h}_n) permettant l'utilisation de codes antiparcimonieux. Bien que l'on souhaite idéalement $|a_n^{\mathbf{h}}| = 1/\sqrt{N}$, on peut se contenter ici de coefficients de la forme $|a_n^{\mathbf{h}}| \leq K/\sqrt{N}$. Nous soulignons que les coefficients sont désormais bornés par un terme proportionnel à $1/\sqrt{N}$. La constante K doit évidemment être le plus proche possible de 1. Cette fois, les coefficients $a_n^{\mathbf{h}}$ sont par construction dans l'intervalle $[-K/\sqrt{N}, K/\sqrt{N}]$. La borne sur l'erreur de quantification devient

$$\|\mathbf{y} - \hat{\mathbf{y}}^{\mathbf{h}}\|_2 \leq \frac{2K}{L-1}. \quad (1.9)$$

Si l'on arrive à rendre K petit, c'est-à-dire ici $K < \sqrt{N}$, alors on réduit la borne sur l'erreur de quantification. On retiendra que le codage antiparcimonieux permet ici de passer d'une borne sur l'erreur de quantification proportionnelle à $\sqrt{N}/(L-1)$ dépendant du nombre de coefficients N à une borne de la forme $K/(L-1)$, où K est une propriété du code.

1.1.4 Conversion analogique numérique

La partie précédente a traité le cas de la quantification. Mais une notion de *démocratie* proche de celle d'antiparcimonie est aussi étudiée pour la conversion analogique numérique. Les convertisseurs Sigma Delta $(\Sigma - \Delta)$ sont une classe de convertisseurs analogiques-numériques (A/N) populaire en raison de leur écriture mathématiques sous forme de produit de convolution.

Soit \mathbf{x} un signal à bande limitée à valeurs dans $[-1, 1]$. Un convertisseur $\Sigma - \Delta$, est associé à une suite de filtre $(\mathbf{h}_n)_n$ et produit une suite de N bits \mathbf{x} à valeur dans $\{\pm 1\}^N$ telle que le signal \mathbf{x} peut être approché par la quantité $\sum_{\ell} x_{\ell} \mathbf{h}_{N-\ell}$. Deux fonctions caractérisent un filtre A/N

- la fonction d'encodage E^N , qui transforme un signal échantillonné \mathbf{y} en une suite de N bits, $\mathbf{x}, E^N : [-1, 1]^N \longrightarrow \{\pm 1\}^N$,
- la fonction de décodage D^N , qui transforme une suite de N bits \mathbf{x} en un signal $\hat{\mathbf{y}}, D^N : \{\pm 1\}^N \longrightarrow [-1, 1]^N$.

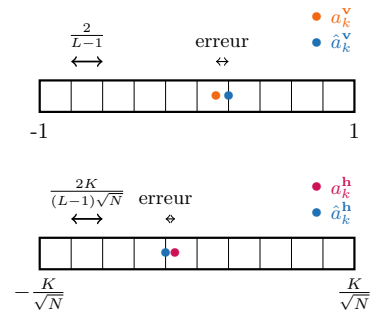


FIGURE 1.2 : Illustration de l'erreur de quantification sur un dictionnaire uniforme \mathbf{v} (haut) et un dictionnaire autorisant un codage antiparcimonieux \mathbf{h} (bas). Dans les deux cas, la quantification est réalisée avec L bandes. La distance entre deux cases représente l'erreur de quantification maximale sur un coefficient.

Optimalité Un convertisseur A/N produit une approximation d'un signal continu y . La première quantité, appelée *précision*, s'intéresse à l'erreur d'approximation. La précision $\mathcal{A}(E, D)$ est définie par

$$\mathcal{A}(E^N, D^N) = \sup_{y \in \mathcal{F}([-1,1])} \|y - D^N \circ E^N(y)\|_2, \quad (1.10)$$

où $\mathcal{F}([-1,1])$ est l'ensemble des fonctions continues à bande limitée de $[-1,1]$. Un convertisseur A/N est dit optimal s'il existe une constante $C_{\mathcal{A}}$ indépendante de N telle que

$$\forall N \in \mathbb{N}^*, \quad \mathcal{A}(E^N, D^N) \leq \frac{C_{\mathcal{A}}}{2^{N+1}}. \quad (1.11)$$

En d'autres mots, un convertisseur A/N est optimal si sa précision est meilleure que celle d'une simple binarisation du signal.

Démocratie La seconde quantité majore l'erreur survenant lorsqu'un bit du signal numérique est accidentellement inversé. Formellement, soit φ^ℓ l'opération qui inverse le bit ℓ . L'*erreur de décodage* \mathcal{E} est définie par

$$\mathcal{E}(E^N, D^N) = \sup_{\mathbf{x} \in \{\pm 1\}^N, \ell \in \mathbb{N}^*} \|D^N(\mathbf{x}) - D^N \circ \varphi^\ell(\mathbf{x})\|_2. \quad (1.12)$$

Un convertisseur A/N est dit démocratique si chaque bit encode une information dont l'énergie est de l'ordre de $1/\sqrt{N}$. Formellement, un tel convertisseur doit vérifier

$$\forall N \in \mathbb{N}^*, \quad \mathcal{E}(E^N, D^N) \leq \frac{C_{\mathcal{E}}}{N}, \quad (1.13)$$

où $C_{\mathcal{E}}$ est une constante indépendante du nombre de bit N . On retrouve un concept proche de l'antiparcimonie.

On souhaiterait idéalement construire un convertisseur $\Sigma - \Delta$ démocratique et optimal. Cependant, le théorème suivant montre qu'un tel convertisseur n'existe pas.

Théorème 1.1. (*Calderbank et Daubechies, 2002, Théorème 2.4*) Si un convertisseur $\Sigma - \Delta$ est optimal alors il n'est pas démocratique.

Ce résultat montre que la robustesse du codage antiparcimonieux se fait au prix de la qualité de l'approximation. Bien que négatif, ce résultat théorique montre que le codage antiparcimonieux a été étudié de près dans le cadre de la conversion A/N, où son intérêt réside dans sa robustesse aux déformations lors de la transmission.

1.1.5 Réduction du facteur de crête

Le *multiplexage par répartition orthogonale de fréquence* (OFDM pour *orthogonal frequency-division multiplexing* en anglais) est un procédé de codage des signaux numériques sur des sous-porteuses faisant intervenir la transformée de Fourier rapide. Le signal reconstruit après réception peut, en raison du codage, être à large crête. Cet effet a des conséquences

directes sur l'efficacité des convertisseurs analogique numérique. Ce facteur de crête est mesuré via le *Peak to Average Power Ratio (PAPR)*, définie pour tout entier L

$$\text{PAPR}_L(\mathbf{x}) = \frac{\|\mathbf{x}\|_\infty^L}{\|\mathbf{x}\|_2^2}, \quad (1.14)$$

Deux PAPR sont en réalité définis, pour les domaines continus et discrets.

où \mathbf{x} est le signal à envoyer. Comme la norme ℓ_∞ apparaît dans la quantité d'intérêt, de récents travaux se sont intéressés aux méthodes antiparcimonieuses (Ilic et Strohmer, 2009; Farrell et Jung, 2009; Studer et al., 2014). L'idée est d'exploiter les canaux non utilisés pour introduire de la redondance et augmenter la robustesse du signal transmis. En somme, l'utilisation de codes antiparcimonieux en OFDM permet de transmettre des codes dont le PAPR est plus faible, ce qui résout certains problèmes de dimensionnement de canaux.

1.1.6 Recherche approximative de plus proches voisins

La dernière application que nous allons présenter concerne l'apprentissage automatique (*machine learning* en anglais). La recherche des k plus proches voisins (k -PPV) consiste à trouver les k éléments les plus proches d'un vecteur requête dans une base. La notion de voisinage est définie au sens d'une métrique, le plus souvent la distance euclidienne.

Le signal est ici composé du vecteur requête et des éléments de la base de donnée.

Un des premiers défis est la mise en place d'algorithmes rapides qui permettent de traiter de grandes² bases de données. En raison de la taille de la base de données, il est parfois intéressant de faire des concessions sur l'efficacité des algorithmes pour gagner en vitesse d'exécution, d'où le terme de recherche *approximative*. Les solutions proposées s'appuient sur des méthodes de réduction de la dimension, de segmentation des données, etc (Jégou et al., 2011, et références à l'intérieur).

2. Grand signifie ici 10^6 à 10^9 éléments (Balu et al., 2014).

D'autres problématiques ont cependant été récemment soulevées, liées au respect de la vie privée. Il peut arriver que les propriétaires de la base ou de la requête ne soient pas prêts à partager leurs données. La plupart des solutions existantes sont en effet basées sur des méthodes de chiffrement et favorisent la sécurité au détriment de l'efficacité numérique. Une solution alternative faisant appel aux représentations antiparcimonieuses a récemment été proposée (Jégou et al., 2012; Mathon et al., 2013). Cette solution consiste à augmenter d'une quantité arbitraire la dimension des données. Les nouvelles représentations ne sont donc pas uniques, et peuvent être partagées sans compromettre l'anonymat. Parmi l'ensemble des solutions admissibles on choisit la plus antiparcimonieuse. Celle-ci a l'avantage d'être prébinarisée, *i.e.*, de la forme $\pm\alpha$ avec α un réel positif. Ce plongement de Hamming a l'avantage de rendre les distances entre éléments calculables uniquement à l'aide d'opérations XOR, permettant un gain de vitesses et d'utilisation de la mémoire comparées à une distance euclidienne. Cette méthode a donné des résultats encourageants mais le point limitant pour l'instant est le temps de calcul des représentations.

En biomédical, la requête peut contenir des données sensibles sur le patient.

1.2 ACQUISITION COMPRESSÉE ET PARCIMONIE

NOUS AVONS MOTIVÉ dans la partie précédente le problème du codage antiparcimonieux à travers quelques exemples issus de la littérature en traitement du signal. Avant d'étudier ce problème, nous allons discuter brièvement de quelques résultats en acquisition compressée, qui est un domaine de recherche relativement bien compris aujourd'hui. Cette étude nous permettra de réaliser dans les parties suivantes un parallèle théorique et algorithmique entre les deux problèmes.

En acquisition compressée, l'objectif n'est plus d'étaler l'énergie de façon uniforme mais de la concentrer sur quelques coefficients non nuls, ce qui fait le lien avec la notion de *parcimonie*. Le signal est reconstruit en résolvant un problème inverse. Nous allons passer en revue quelques résultats importants liés à ce domaine de recherche avant de revenir au problème du codage antiparcimonieux.

1.2.1 Motivations

On sait depuis les travaux fondateurs concernant l'échantillonnage qu'il est possible de reconstruire exactement un signal continu à bande limitée avec un nombre dénombrable d'échantillons (Nyquist, 1928 ; Shannon, 1949) en choisissant la bonne fréquence d'échantillonnage. Il existe une formule d'interpolation qui est exacte si la fréquence d'échantillonnage est au moins deux fois supérieure à la fréquence maximale du signal. Ces résultats ont motivé l'étude et l'utilisation de signaux numériques plutôt qu'analogiques. Cependant, le stockage, le transfert et le traitement de ces données restent limités par les moyens informatiques. Il est devenu nécessaire de compresser cette information.

La compression de l'information peut s'interpréter comme un changement de représentation. On espère que dans cette nouvelle représentation, le signal pourra s'exprimer ou à défaut s'approximer avec moins de coefficients, et sera donc moins coûteux à stocker et à traiter.

Définition 1.2. On dit qu'un signal x de \mathbb{R}^N , est parcimonieux d'ordre k s'il contient au plus k coefficients non nuls, i.e., $\|x\|_0 \leq k$. On notera Σ_k^N le sous espace de \mathbb{R}^N formé par les signaux parcimonieux d'ordre k .

Par exemple, un signal numérique à bande limitée est parcimonieux dans l'espace de Fourier. Le cadre de l'*acquisition compressée* étend les résultats sur l'échantillonnage aux signaux qui sont parcimonieux dans une autre représentation que Fourier. Cette nouvelle représentation n'est pas forcément connue à l'avance. Dans ce cas, l'*acquisition compressée* peut être vu comme la réalisation conjointe de l'échantillonnage et de la compression. En exploitant cette structure parcimonieuse, il est à nouveau possible pour certaines classes de signaux de reconstruire exactement une observation. On verra que l'*acquisition compressée* peut parfois nécessiter moins d'échantillons que ce qu'impose la *condition de Nyquist* (Candès et al., 2006a, 2006b ; Baraniuk, 2007). Dans ce chapitre, nous ferons l'hypothèse que le signal x est parcimonieux dans la base canonique de \mathbb{R}^N .

Ce procédé d'approximation est à la base des formats de compression multimédias tels que JPEG, JPEG2000, MPEG, MP3, etc.

Si le signal est parcimonieux dans une autre base Φ , on considère plutôt le problème

$$y = H\Phi^T x.$$

Dans le cadre de l'*acquisition compressée*, la mesure \mathbf{y} , élément de \mathbb{R}^M est réalisée via une multitude de projections linéaires du signal \mathbf{x} , élément de \mathbb{R}^N . Ces projections sont non adaptatives mais doivent conserver la structure du signal. On réalise M mesures via une matrice d'acquisition réelle \mathbf{H} connue de taille $M \times N$ avec idéalement $M \ll N$. La partie 1.2.3.c discutera la conception d'opérateurs d'acquisition efficace. Cette mesure peut être perturbée par un bruit \mathbf{e} , élément de \mathbb{R}^M . Cette opération de mesure s'écrit mathématiquement

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}, \quad (1.15)$$

et la position des zéros dans le signal \mathbf{x} n'est pas toujours connue à l'avance.

L'objectif est de reconstruire le signal \mathbf{x} à partir de la mesure \mathbf{y} . C'est un *problème inverse*. Trois problématiques émergent naturellement. La première concerne l'étude des conditions sur l'opérateur d'acquisition \mathbf{H} pour garantir une bonne reconstruction. La seconde concerne la conception de ces opérateurs d'acquisition. La dernière est la mise en place d'algorithmes de reconstruction efficaces. Les prochaines parties apportent des éléments de réponse à ces trois questions.

1.2.2 Liens avec la parcimonie

Dans ce problème, on recherche une solution parcimonieuse à un problème de reconstruction. On choisit ici de le formuler comme un problème de minimisation sous contraintes

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z} \in \mathbb{R}^N} \|\mathbf{z}\|_0 \quad \text{s.c.} \quad \mathbf{z} \in \mathcal{B}^\varepsilon(\mathbf{H}, \mathbf{y}), \quad (P_0^\varepsilon)$$

où \mathbf{H} est la matrice d'acquisition, \mathbf{y} est le vecteur mesure, \mathcal{B}^ε est l'ensemble des solutions admissibles et $\hat{\mathbf{x}}$ désigne l'estimateur du signal. Par exemple, $\mathcal{B}^\varepsilon(\mathbf{H}, \mathbf{y}) \triangleq \{\mathbf{z} \mid \|\mathbf{H}\mathbf{z} - \mathbf{y}\|_2 \leq \varepsilon\}$. Ces notations seront consistantes dans l'ensemble de cette partie.

Le choix de la quasi norme ℓ_0 dans (P_0^ε) est naturel puisqu'elle mesure directement le niveau parcimonie d'un vecteur. Cette fonction objectif n'est cependant pas convexe, ce qui rend la recherche de solutions difficile. Une première solution communément acceptée est de considérer plutôt la relaxation convexe de la quasi norme ℓ_0 , à savoir la norme ℓ_1 . Le problème (P_0^ε) devient

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z} \in \mathbb{R}^N} \|\mathbf{z}\|_1 \quad \text{s.c.} \quad \mathbf{z} \in \mathcal{B}^\varepsilon(\mathbf{H}, \mathbf{y}). \quad (P_1^\varepsilon)$$

Cette simplification du problème s'interprète géométriquement. En effet, la norme ℓ_1 est la plus petite norme³ telle que le problème (P_0^ε) soit convexe. La figure 1.4 ci-dessous montre les solutions du problème (P_0^ε) en dimension 2 pour différentes relaxations ℓ_p . On constate que les solutions de la relaxation ℓ_p pour $p \leq 1$ coïncident avec un sous-ensemble des solutions du problème (P_0^ε) . On espère que ce résultat se généralise aux dimensions supérieures.

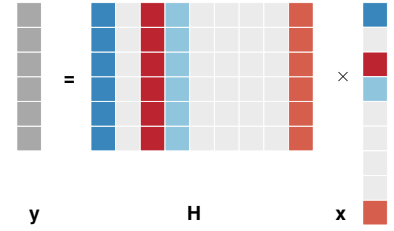


FIGURE 1.3 : Le signal \mathbf{x} est parcimonieux, seuls quelques coefficients sont non nuls. Les coefficients en gris clair sont nuls. On souhaite reconstruire \mathbf{x} à partir de la mesure \mathbf{y} . La couleur rouge ou bleu représentent le signe et l'intensité des composantes non nulles.

La recherche d'une solution approchée de (P_0^ε) est dans le cas général un problème NP-difficile (Muthukrishnan, 2005).

3. Au sens de la valeur de p dans ℓ_p .

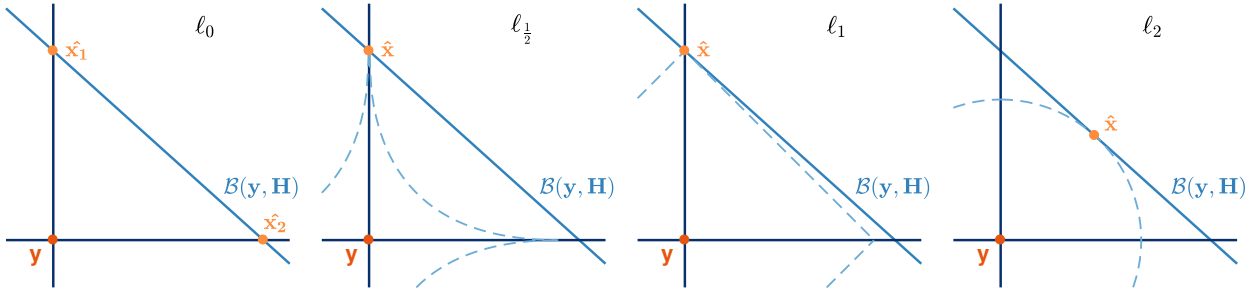


FIGURE 1.4 : Meilleure approximation d'un point de \mathbb{R}^2 par un élément du sous-espace $\mathcal{B}^0(\mathbf{y}, \mathbf{H})$ minimisant une (quasi)norme ℓ_p , pour $p \in \{0, \frac{1}{2}, 1, 2\}$.

Malgré des similitudes dans la formulation du problème (\mathbf{P}_1^ℓ) et de celui résolu par la méthode *Least Absolute Shrinkage and Selection Operator (Lasso)* (Tibshirani, 1996), il s'agit de deux problèmes conceptuellement différents. Notons que la régularisation ℓ_2 est appelée régularisation de Tikhonov (Tikhonov, 1963).

1.2.3 Reconstruction d'un signal à partir de mesures bruitées

1.2.3.a Mesure non bruitée

On s'intéresse dans cette partie au cas idéal où le signal \mathbf{x} est réellement parcimonieux. On cherche des conditions sur la matrice d'acquisition pour garantir la reconstruction du vrai signal, et en particulier l'unicité de la solution. Or, une manière d'assurer l'unicité de la solution d'un système linéaire est d'étudier l'injectivité du morphisme associé au système, *i.e.*, induit par la matrice d'acquisition \mathbf{H} . Intuitivement, si l'on connaît la position des coefficients égaux à zéro, il suffirait de résoudre un problème des moindres carrés. Pour cela, il suffit que toutes sous matrices extraites de \mathbf{H} soient non dégénérées.

Mais la position des coefficients nuls est le plus souvent inconnue et une condition plus forte est nécessaire. On verra essentiellement au théorème 1.2 qu'une condition pour reconstruire un signal parcimonieux d'ordre k est que le noyau de \mathbf{H} ne contienne pas l'espace Σ_{2k}^N . En effet, si \mathbf{x}_1 et \mathbf{x}_2 sont deux solutions parcimonieuses d'ordre k du problème, alors $\mathbf{x}_1 - \mathbf{x}_2$ est à la fois élément du noyau de \mathbf{H} et de Σ_{2k}^N . Si l'intersection est vide, *i.e.*, $\ker(\mathbf{H}) \cap \Sigma_{2k}^N \neq \{\mathbf{0}\}$, alors \mathbf{x}_1 et \mathbf{x}_2 sont égaux et l'on peut garantir l'unicité de la solution. Cette intuition est traduite en utilisant la notion de *spark* (Donoho et Elad, 2003).

Définition 1.3. Soit \mathbf{H} une matrice d'acquisition de taille $M \times N$. On appelle *spark* de \mathbf{H} le plus petit nombre de colonnes de \mathbf{H} linéairement dépendantes.

De plus, si le *spark* d'une matrice est $M + 1$, alors elle est dite *full spark*.

Le théorème ci-dessous donne une condition sur le *spark* de la matrice \mathbf{H} pour garantir l'unicité de la reconstruction.

Théorème 1.2. (Donoho et Elad (2003), corollaire 4) Soit \mathbf{H} une matrice d'acquisition. Pour tout vecteur de mesure \mathbf{y} de \mathbb{R}^M il existe un unique signal \mathbf{x} de \mathbb{R}^N parcimonieux d'ordre k , si et seulement si $\text{spark}(\mathbf{H}) > 2k$.

Par exemple, le *spark* de la matrice $\mathbf{H} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{pmatrix}$ est 3. \mathbf{H} est d'ailleurs *full spark*.

Reconstruction exacte d'un signal parcimonieux.

Le spark est le plus petit nombre de colonnes linéairement dépendantes de \mathbf{H} . Il est implicitement lié à la dimension du noyau de \mathbf{H} . Le théorème 1.2 signifie que caractériser le noyau est suffisant pour identifier un signal parcimonieux dans le cas non bruité.

1.2.3.b Stabilité en cas de bruit

Caractériser le *spark* d'une matrice est insuffisant lorsque la mesure est perturbée, i.e, $\mathbf{e} \neq \mathbf{0}$. Cette perturbation résulte par exemple d'une erreur de quantification ou d'un bruit de capteur. Pour y remédier, une seconde caractérisation des opérateurs d'acquisition est proposée (Candès et Tao, 2005).

Définition 1.4. Une trame \mathbf{H} (cf. définition 1.1) de dimensions $M \times N$ satisfait la condition propriété d'isométrie restreinte (RIP) d'ordre k s'il existe un entier δ_k de $(0, 1)$ tel que pour tout signal parcimonieux d'ordre k \mathbf{x} de Σ_k^N

$$(1 - \delta_k) \|\mathbf{x}\|_2^2 \leq \|\mathbf{H}\mathbf{x}\|_2^2 \leq (1 + \delta_k) \|\mathbf{x}\|_2^2. \quad (1.16)$$

L'équation (1.16) signifie que le morphisme induit par une matrice vérifiant la condition RIP doit approximativement conserver les distances entre les vecteurs parcimonieux. Cette condition définit implicitement une classe de *trame*, moins contraignante que les *trames de Parseval*. Ces *trames* se comportent approximativement comme des isométries mais où la contrainte d'orthogonalité a été relâchée.

Dans le cas où la perturbation est nulle, cette condition assure aussi que deux signaux parcimonieux d'ordre k ne donnent pas la même mesure.

À ce stade, il est également intéressant de remarquer qu'il existe une condition nécessaire sur le nombre d'échantillons pour qu'une matrice d'acquisition vérifie la condition RIP

Théorème 1.3. (Davenport (2010), Théorème 3.5) Soit \mathbf{H} une matrice d'acquisition de taille $M \times N$ qui satisfait la condition RIP d'ordre $2k$ de constante $\delta_{2k} \in (0, \frac{1}{2}]$. Alors

$$M \geq \frac{1}{2} \log(\sqrt{24} + 1) k \ln \left(\frac{N}{k} \right). \quad (1.17)$$

D'autres minoration existent notamment pour $\delta_{2k} > \frac{1}{2}$, dont le comportement varie selon le problème. Nous ne les aborderons pas dans ce travail. Dans tous les cas, le théorème 1.3 ou ses variantes donne une borne sur la réduction de la dimension envisageable pour une matrice vérifiant la condition RIP. Ces résultats ne disent cependant rien quant à la manière de construire des matrices vérifiant cette condition. Ce point sera abordé à la partie 1.2.3.c.

La condition RIP est étudiée en *acquisition compressée* car c'est une condition suffisante pour obtenir de bonnes garanties de reconstruction dans le cas où les mesures sont perturbées. Ce résultat est résumé par le théorème suivant

L'acronyme RIP provient de « restricted isometry property » en anglais.

Si $\mathbf{x}_1, \mathbf{x}_2$ sont deux vecteurs de Σ_k et \mathbf{H} vérifie la condition RIP d'ordre $2k$ alors $\mathbf{x}_1 - \mathbf{x}_2 \in \Sigma_{2k}$ et $\|\mathbf{y}_1 - \mathbf{y}_2\|_2 = \|\mathbf{H}(\mathbf{x}_1 - \mathbf{x}_2)\|_2 > 0$.

Théorème 1.4. (Candès (2008), Théorème 1.2) Soit \mathbf{H} une matrice d'acquisition vérifiant la condition RIP d'ordre $2k$ avec $\delta_{2k} < \sqrt{2} - 1$. Soient \mathbf{x} un signal de \mathbb{R}^N et \mathbf{y} le vecteur de mesure tel que $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}$, où \mathbf{e} est une perturbation telle que $\|\mathbf{e}\|_2 \leq \varepsilon$ avec $\varepsilon \geq 0$. Alors la solution $\hat{\mathbf{x}}$ du problème (P_1^ε) vérifie

$$\|\hat{\mathbf{x}} - \mathbf{x}\|_2 \leq \frac{C_0}{\sqrt{k}} \|\mathbf{x} - \mathbf{x}_k\|_1 + C_1 \varepsilon, \quad (1.18)$$

$$C_0 = 2 \frac{1 + (\sqrt{2} - 1)\delta_{2k}}{1 - (\sqrt{2} + 1)\delta_{2k}}$$

$$C_1 = \frac{4\sqrt{1 + \delta_{2k}}}{(1 - (\sqrt{2} + 1)\delta_{2k})}$$

où \mathbf{x}_k est l'approximation parcimonieuse d'ordre k du vrai signal \mathbf{x} où l'on a conservé que les k composantes les plus grandes en valeur absolue.

Ce résultat est aussi connu sous le nom d'instance optimale $\ell_2 - \ell_1$ (Cohen et al., 2008). Il s'agit d'une garantie de reconstruction déterministe, valable pour n'importe quelle matrice d'acquisition \mathbf{H} et tout signal \mathbf{x} . Ainsi, même dans le cas bruité, la relaxation convexe (P_1^ε) du problème (P_0) amène à une erreur de reconstruction finie, bornée par deux termes. Le premier terme est une erreur d'approximation et le second croît linéairement avec l'écart type du bruit. Notons qu'en cas de perturbation nulle et d'un signal parcimonieux d'ordre k , on retrouve la garantie de reconstruction exacte donnée au théorème 1.2.

Pour finir, il y a possibilité de raffiner la borne si l'on rajoute des hypothèses au problème. Il existe aussi des résultats plus fins lorsque l'on sait caractériser la perturbation aléatoire \mathbf{e} . De plus, il existe d'autres façons d'aborder le problème en étudiant les matrices d'acquisition vérifiant une condition dite de *propriété du noyau* (NSP), que nous n'aborderons pas dans cette thèse. On réfère le lecteur au chapitre 5 de Davenport et al. (2012) pour une revue des principaux résultats.

L'acronyme NSP provient de « Null Space Property » en anglais

1.2.3.c Conception de matrices d'acquisition

La conception de matrices d'acquisition pertinentes pour l'acquisition compressée est difficile. On a essentiellement identifié deux propriétés désirables pour une matrice d'acquisition. On doit au choix calculer le *spark* de ces matrices, ou les constantes intervenant dans la condition RIP, cf. théorèmes 1.2 et 1.4.

Le rang d'une matrice est simple à calculer, mais son *spark* nécessite une recherche combinatoire, qui est NP-difficile. Des solutions alternatives ont été proposées. Par exemple, les matrices d'acquisition \mathbf{H} dont les entrées sont des lois normales i.i.d. sont de *spark* $\text{rang}(\mathbf{H}) + 1$ avec probabilité 1. D'autres résultats permettent par exemple de borner le *spark* d'une matrice par une mesure de la cohérence entre les colonnes (Mallat et Zhang, 1993).

Il est possible de construire de manière déterministe des matrices qui satisfont la condition RIP d'ordre k , mais ces procédures requièrent un grand nombre de mesures, loin des $\mathcal{O}(k \ln(\frac{N}{k}))$ annoncées au théorème 1.3.

Par exemple $M = \mathcal{O}(k^2 \ln(N))$ (DeVore, 2007) ou $M = \mathcal{O}(kN^\alpha)$ (Indyk, 2008).

Cette difficulté peut être surmontée par le recours à l'aléatoire. Certaines matrices dont les entrées sont aléatoires vérifient en effet la condition RIP avec haute probabilité et pour un nombre de mesures plus raisonnable. Il s'agit de matrices dont les entrées sont des tirages de lois de Bernoulli, gaussiennes, *etc.* On reporte le lecteur à la partie 1.3.3 du même chapitre pour des exemples de telles matrices.

Par exemple, si les entrées d'une matrice \mathbf{H} $M \times N$ sont sous-gaussiennes avec $m = \mathcal{O}\left(\frac{k}{\delta_{2k}^2} \ln\left(\frac{N}{k}\right)\right)$, alors \mathbf{H} vérifie la condition RIP d'ordre $2k$ avec probabilité supérieure à $2 - \exp(-\delta_{2k}^2 M)$.

1.2.4 Algorithmes pour la reconstruction

Nous avons vu dans la partie précédente qu'une solution parcimonieuse s'obtient en résolvant au choix le problème (P_0^ϵ) ou sa relaxation convexe (P_1^ϵ) .

Résolution de (P_0^ϵ) . Malgré la complexité combinatoire, des algorithmes efficaces ont été développés pour résoudre directement le problème (P_0^ϵ) . On distingue les méthodes *combinatoires* des méthodes *gloutonnes*. L'objectif des méthodes combinatoires est de rechercher dans un ensemble le ou les meilleurs sous-ensembles au sens d'une fonction objectif. Malgré des similitudes dans le modèle à traiter, ces méthodes ont encore été peu exploitées (Cormode et Muthukrishnan, 2006). Les méthodes *gloutonnes* sont des méthodes itératives qui résolvent une série de problèmes approchés. Ici, le support et le signal sont successivement estimés. Bien que les résultats empiriques sont similaires à ceux obtenus avec les algorithmes d'optimisation convexes appliqués au problème (P_1^ϵ) , peu de garanties théoriques existent. Par exemple, l'algorithme de résolution itérative par seuillage *Iterative Thresholding (IT)* (Blumensath et Davies, 2009) identifie des solutions dont on connaît le niveau k de parcimonie. Cette méthode repose sur la série de problèmes approchés

$$\hat{\mathbf{x}} = \min_z \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2^2 \quad \text{s.c.} \quad \|\mathbf{z}\|_0 \leq k, \quad (1.19)$$

plus simples à résoudre. Les solutions de ces problèmes sont explicites et correspondent à une étape de descente de gradient suivie d'une opération de seuillage dur.

L'algorithme de *Poursuite Adaptive Matching Pursuit (MP)* proposé par Mallat et Zhang (1993) résout (P_0^ϵ) lorsque le niveau de parcimonie k n'est pas connu. À chaque itération, MP cherche la colonne de la matrice d'acquisition la plus corrélée avec le résidu courant. Le coefficient de $\hat{\mathbf{x}}$ correspondant est activé et sa valeur mise à jour par projection. L'étape de sélection des coefficients fait intervenir une étape de seuillage dur, ou *hard thresholding*. De nombreuses variantes de cet algorithme existent, telles que *Orthogonal Matching Pursuit* (Pati et al., 1993) *Stagewise OMP (StOMP)* (Donoho et al., 2012), *compressive sampling matching pursuit (CoSaMP)* (Needell et Tropp, 2009).

Résolution de (P_1^ϵ) . Le problème relaxé (P_1^ϵ) bénéficie de toute l'artillerie d'algorithmes provenant de l'optimisation convexe. Nous allons détailler la méthode dite de *Forward Backward (FB)*. On considère par exemple la

formulation variationnelle de (P_1^ε) pour β un réel positif

$$\begin{aligned} \hat{x} &= \arg \min_z J_\beta^1(z) \\ \text{où } J_\beta^1(z) &\triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{H}z\|_2^2 + \beta \|z\|_1, \end{aligned} \quad (V_1^\beta)$$

dont la solution, est équivalente à celle du problème (P_1^ε) pour la bonne valeur de β . La résolution de (V_1^β) est équivalente à celle de

$$\hat{x} = \arg \min_{\mathbf{x}, \mathbf{u}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \beta \|\mathbf{u}\|_1 \quad \text{s. c. } \mathbf{x} = \mathbf{u}. \quad (1.20)$$

Bien que la réécriture soit triviale, les multiplicateurs de Lagrange

$$\hat{x} = \arg \min_{\mathbf{x}, \mathbf{u}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \beta \|\mathbf{u}\|_1 + \frac{1}{2\delta} \|\mathbf{x} - \mathbf{u}\|_2^2 \quad (1.21)$$

montrent que la variable auxiliaire introduite amène à un problème séparable en \mathbf{x} et en \mathbf{u} . L'attache aux données est portée par \mathbf{x} et la pénalisation par \mathbf{u} . Le problème peut être attaqué par minimisation alternée : minimiser \mathbf{x} à \mathbf{u} fixé puis \mathbf{u} à \mathbf{x} fixé. On obtient l'algorithme itératif

$$\begin{aligned} \mathbf{x}^{(t+1)} &= \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \frac{1}{2\delta} \|\mathbf{x} - \mathbf{u}^{(t)}\|_2^2 = \mathbf{u}^{(t)} - \delta \nabla \|\mathbf{y} - \mathbf{H} \cdot\|_2^2(\mathbf{u}^{(t)}) \\ \mathbf{u}^{(t+1)} &= \arg \min_{\mathbf{u}} \beta \|\mathbf{u}\|_1 + \frac{1}{2\delta} \|\mathbf{x}^{(t+1)} - \mathbf{u}\|_2^2. \end{aligned}$$

On reconnaît à la dernière équation l'opérateur proximal de la norme ℓ_1 .

Définition 1.5. Soit $f : \mathbb{R}^N \rightarrow \mathbb{R}$ une fonction continue et δ un réel positif. On définit l'opérateur proximal, noté prox_f^δ , par

$$\forall \mathbf{x} \in \mathbb{R}^N, \quad \text{prox}_f^\delta(\mathbf{x}) \triangleq \arg \min_{\mathbf{u} \in \mathbb{R}^N} f(\mathbf{u}) + \frac{1}{2\delta} \|\mathbf{x} - \mathbf{u}\|_2^2. \quad (1.22)$$

L'opérateur proximal est un outil utile pour traiter les fonctions de coût continues mais non différentiables. L'étape de minimisation alternée se réécrit sans faire intervenir la variable auxiliaire

$$\mathbf{x}^{(t+1)} = \text{prox}_{\|\cdot\|_1}^\delta \left(\mathbf{x}^{(t)} - \delta \nabla \|\mathbf{y} - \mathbf{H} \cdot\|_2^2(\mathbf{x}^{(t)}) \right). \quad (1.23)$$

C'est l'algorithme FB (Starck et al., 2010), que l'on peut voir comme une instance d'un algorithme proximal (Parikh et Boyd, 2014). À chaque itération, on fait d'abord un pas en direction du gradient du terme d'attache aux données (mouvement *forward*) suivi d'un pas dans la direction de l'opérateur proximal associé à la norme ℓ_1 (mouvement *backward*), et donné par

$$\forall \mathbf{x}, n \quad \left[\text{prox}_{\beta \|\cdot\|_1}(\mathbf{x}) \right]_n = \text{signes}(\mathbf{x}) \times \max(|x_n| - \beta, 0). \quad (1.24)$$

Pour une version accélérée de ISTA, nous renvoyons le lecteur au travail de Beck et Teboulle (2009). L'opérateur proximal agit comme une opération de seuillage : si un coefficient est trop faible, il est ramené à 0. Toutefois, ce n'est pas un seuillage au sens propre du terme car toutes les valeurs sont modifiées : on parle de seuillage doux. La figure 1.5 compare les opérations de seuillage dur et doux dans \mathbb{R}^2 . On remarque que la fonction seuillage doux est continue, ce qui n'est pas le cas du seuillage dur vu précédemment.

Trouver cette valeur du paramètre β qui rend les problèmes équivalents n'est cependant pas facile. Ce point est par exemple discuté dans Eldar (2009).

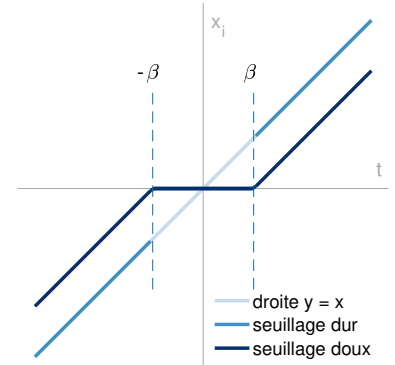


FIGURE 1.5 : Comparaison des opérateurs de seuillage dur et doux. Un seuil à β est appliqué pour le seuillage dur tandis que le seuillage doux correspond à l'opérateur

Algorithme 1.1 : Algorithme de seuillage doux itératif.

Entrées : Matrice d'acquisition \mathbf{H} , mesure \mathbf{y} , paramètre de régularisation β .

```

1  $\mathbf{r} = \mathbf{y}$  // vecteur résidu
2  $\hat{\mathbf{x}} = \mathbf{0}$  // estimateur
3 pour  $t \leftarrow 0$  à  $N_{it}$  faire
4   /* Mouvement en direction du gradient (forward) */
    $\hat{\mathbf{x}}^{(t+1/2)} = \hat{\mathbf{x}}^{(t+1/2)} - \mathbf{H}^T (\mathbf{H}\hat{\mathbf{x}}^{(t)} - \mathbf{y})$ ;
   /* Mouvement en direction du prox (backward) */
5    $\hat{\mathbf{x}}^{(t+1)} = \text{prox}_{\beta \|\cdot\|_1} (\hat{\mathbf{x}}^{(t+1/2)})$  // Opération de seuillage
   doux
6 fin
```

Output : Représentation parcimonieuse $\hat{\mathbf{x}}$.

Nous concluons en notant que les trois algorithmes présentés, à savoir **MP**, **IT**, et *Iterated Soft Thresholding Algorithm (ISTA)* font apparaître une étape de seuillage. Pour **MP** et **IT**, il s'agit d'un seuillage dur : les $N - k$ plus petites valeurs des coefficients sont ramenées à zéro. Pour **ISTA**, un coefficient n'est ramené à 0 que si sa valeur est trop faible. L'objet mathématique utilisé pour représenter l'opération de seuillage doux est l'opérateur proximal. Cet opérateur jouera également un rôle important dans la recherche de codes antiparcimonieux.

Inférence bayésienne. Nous avons vu en introduction que la méthodologie bayésienne apporte de la souplesse dans la modélisation des problèmes et fournit des algorithmes systématiques pour approcher des estimateurs avec de bonnes garanties et fournit des outils pour la prise de décision tels que des intervalles de confiance. Rappelons de plus que le problème variationnel (\mathbf{V}_1^β) nécessite de régler un paramètre β . Or, le choix de ce paramètre est difficile en raison du manque d'interprétabilité. La formalisme bayésien permet d'intégrer ce paramètre au modèle à travers le choix d'une loi *a priori*. Le paramètre devient une variable aléatoire et n'est plus fixé à l'avance. Il en résulte des algorithmes ne nécessitant aucun réglage de paramètre. Ce formalisme a naturellement été exploité pour l'acquisition compressée (Ji et al., 2008; Babacan et al., 2010). Ici, la solution du problème (\mathbf{V}_1^β) correspond à l'estimateur du *maximum a posteriori (MAP)* d'un modèle bayésien, où le terme d'attache aux données correspond à la vraisemblance et le terme de pénalité à la loi *a priori*. Le terme de pénalisation en norme ℓ_1 dans (\mathbf{V}_1^β) se traduit par une loi *a priori* de type Laplace, tandis qu'une pénalisation ℓ_0 correspond à une loi impropre.

Mais il n'est plus nécessaire de se limiter aux pénalisations ℓ_0 ou ℓ_1 pour favoriser la parcimonie. Par exemple, un modèle se basant sur des mélanges de gaussiennes a été proposé par Olshausen et al. (2000). Certaines variances sont choisies très faibles, afin de favoriser des composantes proches de 0 tandis que d'autres variances sont plus fortes pour modéliser les composantes significatives. Le cas asymptotique a aussi été considéré, où les gaussiennes modélisant les composantes non significatives sont remplacées par des lois de Bernoulli (Champagnat et al., 1996).

1.3 THÉORIE DU CODAGE ANTIPARCIMONIEUX

ON S'INTÉRESSE dans cette dernière partie à l'étude des représentations antiparcimonieuses. Le plan de l'étude est similaire à celui de la partie 1.2. On précise d'abord le problème à résoudre, puis les garanties en terme de reconstruction et d'étalement des coefficients du code. On s'intéressera ensuite à la conception de matrices de codage, équivalent des opérateurs de mesure pour l'acquisition compressée et l'on terminera par les algorithmes de recherche de représentations antiparcimonieuses.

1.3.1 Pénalisation de l'amplitude maximale du code

Soit \mathbf{y} un vecteur de \mathbb{R}^M représentant une mesure (signal, image, etc). On veut représenter cette mesure sur une famille de vecteurs concaténés dans une matrice d'encodage que l'on notera \mathbf{H} . On souhaite que toutes les composantes de cette nouvelle représentation \mathbf{x} soit de la forme $|x_n| = \pm \|\mathbf{y}\|_2 / \sqrt{N}$ à des fins de robustesse, cf. partie 1.1 pour quelques exemples d'applications.

La figure 1.6 illustre graphiquement la notion de représentation antiparcimonieuse en affichant trois codes différents d'une même observation. En haut, le code est parcimonieux (cf. 1.2), au milieu on représente le code ayant l'énergie est faible (régularisation de Tikhonov (Tikhonov, 1963)) et en bas le code est antiparcimonieux. On observe bien que la composante maximale en valeur absolue d'un code antiparcimonieux est atteinte à plusieurs reprises, et sa représentation tend à être une droite horizontale. Nous allons voir comment obtenir de tels codes.

1.3.1.a Position du problème

Contrairement au cadre de l'acquisition compressée, on ne cherche pas un code \mathbf{x} qui serait à l'origine de la mesure. On est seulement à la recherche d'une nouvelle représentation qui a de bonnes propriétés d'étalement. L'unicité de la représentation \mathbf{x} n'est plus fondamentale. Au contraire, on a vu dans la partie 1.1.6 que l'absence d'unicité du code est intéressante pour garantir le respect de la vie privée, cf. partie 1.1.6. De plus, on cherchait à reconstruire en acquisition compressée un signal en dimension N à partir d'un nombre plus faible d'observations M , i.e., $M \ll N$. Il n'y a pas de contrainte sur la dimension pour le codage antiparcimonieux. Idéalement, on souhaiterait même que la dimension N de la représentation antiparcimonieuse soit égale à la dimension M du vecteur à reconstruire. Mais nous verrons dans cette partie que de telles représentations existent lorsque $M \ll N$.

Le codage antiparcimonieux est formulé comme un problème inverse linéaire

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}, \tag{1.25}$$

où \mathbf{H} est la matrice d'encodage, de taille $M \times N$ et \mathbf{e} une éventuelle perturbation. La perturbation \mathbf{e} est à interpréter comme une erreur résiduelle.

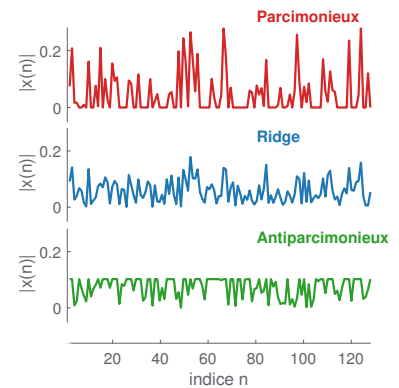


FIGURE 1.6 : Trois codages d'une même observation dans la même représentation. Haut : code parcimonieux, milieu : code obtenu avec une régularisation de Tikhonov et bas : code antiparcimonieux.

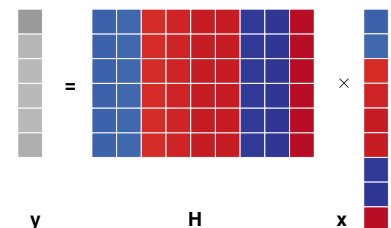


FIGURE 1.7 : On cherche une représentation antiparcimonieuse \mathbf{x} de la mesure \mathbf{y} . Toutes les composantes de la matrice d'encodage sont utilisées, et idéalement tous les coefficients sont égaux au signe près. Ici, les cases bleues et rouges représentent respectivement les valeurs positives et négatives.

Puisque que l'on cherche des représentations avec un minimum de variation d'amplitude, il est naturel de vouloir minimiser ces variations d'amplitudes via un problème de minimisation sous contraintes de la forme

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z} \in \mathbb{R}^N} \left(\max(|z|) - \min(|z|) \right) \quad \text{s.c.} \quad \mathbf{z} \in \mathcal{S}^\varepsilon(\mathbf{y}, \mathbf{H}), \quad (1.26)$$

où $\mathcal{S}^\varepsilon(\mathbf{H}, \mathbf{y}) \triangleq \{ \mathbf{z} \mid \|\mathbf{H}\mathbf{z} - \mathbf{y}\|_2 \leq \varepsilon \}$ avec ε un réel positif ou nul. Il s'agit d'un problème non convexe, donc difficile à résoudre. On préfère considérer un problème approché mais plus simple, où l'on minimise seulement la valeur du plus grand coefficient

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z} \in \mathbb{R}^N} \|\mathbf{z}\|_\infty \quad \text{s.c.} \quad \|\mathbf{H}\mathbf{z} - \mathbf{y}\|_2 \leq \varepsilon \quad (\mathbf{P}_\infty^\varepsilon)$$

où ε est toujours un réel positif ou nul. On espère que minimiser la composante la plus grande en valeur absolue va forcer les autres composantes à s'élever, et donc favoriser l'antiparcimonie.

1.3.1.b Propriétés théoriques des solutions

À nouveau, on distingue les cas où l'on cherche une reconstruction exacte, à savoir le problème (\mathbf{P}_∞^0) que l'on notera (\mathbf{P}_∞) , et les cas où l'on s'autorise une erreur de reconstruction ε en résolvant $(\mathbf{P}_\infty^\varepsilon)$. La figure 1.8 illustre le problème de minimisation de l'amplitude maximale sous contrainte de reconstruction exacte pour les normes ℓ_1 , ℓ_2 et ℓ_∞ . Notons qu'en dimension 2, les boules ℓ_1 et ℓ_∞ sont équivalentes à une rotation près. Cette propriété ne se généralise pas aux dimensions supérieures à 2.

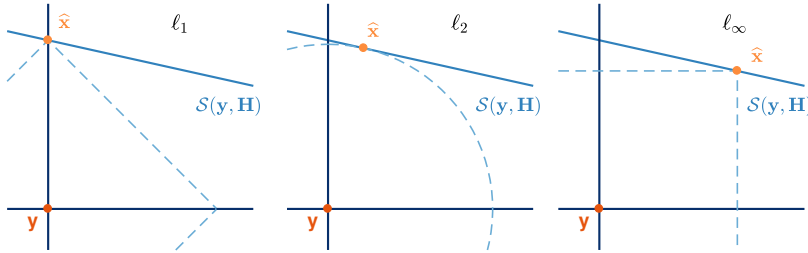


FIGURE 1.8 : Meilleure approximation d'un point de \mathbb{R}^2 par un élément du sous-espace $\mathcal{S}(\mathbf{y}, \mathbf{H})$ minimisant les (quasi)norme ℓ_p , pour $p \in \{1, 2, \infty\}$.

Suivant l'ensemble des solutions admissibles de $(\mathbf{P}_\infty^\varepsilon)$, deux définitions des codes antiparcimonieux sont données dans la littérature. Dans le cas d'une reconstruction exacte, i.e., $\mathcal{S}^\varepsilon(\mathbf{y}, \mathbf{H}) = \mathcal{S}^0(\mathbf{y}, \mathbf{H})$ les solutions sont appelées *représentations de Kashin* (Lyubarskii et Vershynin, 2010).

Définition 1.6. Soient \mathbf{y} un vecteur de \mathbb{R}^M , \mathbf{H} une matrice réelle de taille $M \times N$ et \mathbf{x} un vecteur de \mathbb{R}^N . On dit que \mathbf{x} est une représentation de Kashin de niveau $K > 0$ si

$$\mathbf{y} = \mathbf{H}\mathbf{x} \quad \text{et} \quad \|\mathbf{x}\|_\infty \leq \frac{K}{\sqrt{N}} \|\mathbf{y}\|_2. \quad (1.27)$$

On note λ le niveau de redondance, et $\lambda \triangleq \frac{N}{M}$.

Si l'on s'autorise maintenant une reconstruction avec une erreur ε non nulle, l'ensemble des solutions du problème (P_∞^ε) est encore plus large. Parmi ces solutions, on s'intéresse à celle dont la forme est la plus proche de la solution idéale, et en particulier celle dont l'amplitude maximale est la plus proche de $\|\mathbf{y}\|_2 / \sqrt{N}$. Cette solution appelée représentation *démocratique* (Studer et al., 2012; Studer et al., 2014).

Définition 1.7. Soient \mathbf{y} un vecteur de \mathbb{R}^M , $\mathbf{H} \in \mathbb{R}^{M \times N}$ une matrice de codage et \mathbf{x} un vecteur de \mathbb{R}^N . Soit $\varepsilon \geq 0$. Si pour tout vecteur \mathbf{y} de \mathbb{R}^M le problème (P_∞^ε) admet une solution $\hat{\mathbf{x}}$ vérifiant

$$\frac{K_l}{\sqrt{N}} (\|\mathbf{y}\|_2 - \varepsilon) \leq \|\hat{\mathbf{x}}\|_\infty \leq \frac{K_u}{\sqrt{N}} (\|\mathbf{y}\|_2 + \varepsilon), \quad (1.28)$$

où $0 < K_l < K_u < +\infty$, alors les représentations $\hat{\mathbf{x}}$ sont dites démocratiques. Les constantes K_l et K_u sont appelées les bornes de Kashin.

Les constantes K_l et K_u dépendent uniquement de la matrice de codage. Notons que l'on s'intéresse ici à des matrices de codage dont les bornes de Kashin sont les plus proches possibles de 1. Cette nouvelle définition n'est pas en contradiction avec la première, car dans le cas où $\varepsilon = 0$, la constante K_u correspond à la constante K dans la définition des représentations de Kashin, cf. définition 1.6.

Rappelons également que cette définition est motivée par le codage antiparcimonieux : en forçant la composante la plus grande à prendre valeur dans un intervalle proche de $\|\mathbf{y}\|_2 / \sqrt{N}$, on espère que les valeurs des autres composantes se rapprocheront de $\|\mathbf{y}\|_2 / \sqrt{N}$ en valeur absolue.

1.3.2 Existence de solutions à amplitude maximale bornée

1.3.2.a Principe d'incertitude matriciel

On a vu dans la partie 1.1 que la recherche de représentations à amplitude minimale remonte aux années 60 et concerne les systèmes à efforts minimaux (Neustadt, 1962). Ce problème a ensuite été formalisé sous forme de recherche d'une solution de (P_∞) (Cadzow, 1971). En particulier, si l'on note r le rang de la matrice d'encodage \mathbf{H} de taille $M \times N$ avec $r \leq M$, il existe une solution $\hat{\mathbf{x}}^*$ de (P_∞) dont $N - r + 1$ composantes sont égales à $\pm \|\hat{\mathbf{x}}^*\|_\infty$. La preuve, ainsi que la méthode de résolution s'appuie sur le dual Lagrangien du problème (P_∞) (voir algorithme 1.2). Ce résultat a été redécouvert plus récemment par Fuchs (2011) et complété par Studer et al. (2014).

En discutant le rang de la matrice de codage \mathbf{H} , nous parlons indirectement de contraintes sur son noyau.

Malgré la dualité entre la parcimonie et l'antiparcimonie, on retrouve une condition sur la trame \mathbf{H} pour garantir l'existence de représentation de Kashin qui évoque la condition RIP (cf. définition 1.4). Il s'agit d'un principe d'incertitude sur les supports valable pour une large gamme de matrices d'encodage. Cette condition a été introduite sous le nom de *principe d'incertitude matricielle* (PIM) par Lyubarskii et Vershynin (2010).

La condition RIP est nommée *Uniform uncertainty principle* (UUP) par Lyubarskii et Vershynin (2010).

Définition 1.8. Une matrice \mathbf{H} de $\mathbb{R}^{M \times N}$ satisfait la condition PIM de paramètres η, γ de $]0, 1[$ si pour tout vecteur \mathbf{x} de \mathbb{R}^N

$$\|\mathbf{x}\|_0 \leq \eta N \Rightarrow \|\mathbf{H}\mathbf{x}\|_2 \leq \gamma \|\mathbf{x}\|_2. \quad (1.29)$$

Notons que la condition **PIM** a initialement été introduite comme une généralisation du *principe d'incertitude* en traitement du signal numérique, dans le sens où elle donne un compromis entre l'étalement d'un signal \mathbf{x} dans deux représentations différentes : la base canonique de \mathbb{R}^N , et celle définie par \mathbf{H} .

La différence fondamentale avec la condition **RIP** est que l'on ne s'intéresse plus à contrôler la borne inférieure de la quantité $\|\mathbf{H}\mathbf{x}\|_2$. En effet, dans le cadre de l'*acquisition compressée*, l'inégalité $(1 - \varepsilon) \|\mathbf{x}\|_2 \leq \|\mathbf{H}\mathbf{x}\|_2$ pour un signal parcimonieux \mathbf{x} sert à assurer l'unicité de la solution, cf. partie 1.2. Ici, comme on cherche seulement une nouvelle représentation de l'observation \mathbf{y} , l'unicité n'est plus essentielle tant que la représentation \mathbf{x} a les bonnes propriétés d'étalement.

1.3.2.b Existence d'une représentation de Kashin

Nous allons voir avec le théorème suivant en quoi la condition **PIM** est suffisante pour obtenir une garantie sur le niveau K d'étalement de l'énergie. Cette garantie est fonction des quantités η et γ intervenant dans la condition **PIM**.

Théorème 1.5. (Lyubarskii et Vershynin (2010), Théorème 3.5) Soit \mathbf{H} une trame ajustée de $\mathbb{R}^{M \times N}$ respectant la condition **PIM** de paramètres η, γ . Alors tout vecteur \mathbf{y} de \mathbb{R}^M admet une représentation de Kashin de niveau $K = (1 - \eta)^{-1} \gamma^{-1/2}$.

Ce résultat affirme que toute matrice d'encodage \mathbf{H} qui est une trame ajustée permet l'obtention de représentation de *Kashin*, dont le niveau dépend uniquement des propriétés de \mathbf{H} . Remarquons ici que plus les paramètres η et γ seront respectivement proches de 0 et 1 et plus la *matrice de codage* \mathbf{H} sera intéressante en pratique.

La preuve du théorème 1.5 est algorithmique, ce qui permet d'anticiper la partie 1.4. La preuve repose sur une opération de troncature appliquée à un vecteur autorisant la solution exacte dans le problème (P_∞^ε). La nouvelle représentation n'est plus exacte, donc on répète la même opération avec les résidus successifs, jusqu'à atteindre une solution satisfaisante.

Idée de la preuve Introduisons la fonction de troncature au seuil L

$$\forall z \in \mathbb{R} \quad t_L(z) = \begin{cases} \frac{z}{|z|} \min(|z|, L) & \text{si } z \neq 0 \\ 0 & \text{si } z = 0 \end{cases}, \quad (1.30)$$

et l'opérateur de troncature T_L de niveau L

$$\forall \mathbf{y} \in \mathbb{R}^M, \mathbf{y} = \sum_{n=1}^N z_n \mathbf{h}_n \quad T_L(\mathbf{y}) \triangleq \sum_{n=1}^N t_L(z_n) \mathbf{h}_n. \quad (1.31)$$

Dans l'idéal il faudrait tronquer au niveau $L(\mathbf{y}) = \|\mathbf{y}\|_2 / \sqrt{N}$ mais en raison de la condition **PIM**, on ne peut qu'espérer tronquer à $L^*(\mathbf{y}) = \|\mathbf{y}\|_2 / \sqrt{\eta N}$. Pour abrégier la preuve, on admet le lemme suivant qui encadre l'erreur d'approximation par l'opérateur de troncature T_{L^*}

$$\forall \mathbf{y} \in \mathbb{R}^N \quad \|\mathbf{y} - T_{L^*}(\mathbf{y})\|_2 \leq \eta \|\mathbf{y}\|_2. \quad (1.32)$$

L'opérateur de troncature est construit pour donner un compromis, contrôlé par L , entre la reconstruction et l'étalement.

La condition **PIM** implique pour \mathbf{x} tel que $\text{supp}(\mathbf{x}) \leq \eta N$ que $\eta N \|\mathbf{x}\|_\infty \geq \|\mathbf{x}\|_2 \geq \gamma^{-1} \|\mathbf{x}\|_2$. D'où le niveau de troncature, et la majoration.

Cette inégalité n'est vraie que pour le niveau de troncature $L(\mathbf{y}) = \frac{\|\mathbf{y}\|_2}{\sqrt{\eta N}}$.

C'est dans la preuve du résultat (1.32) qu'intervient l'hypothèse de trame ajustée et la condition PIM. On définit la suite $(\mathbf{y}^{(k)})$ des résidus successifs

$$\mathbf{y}^{(k)} = \begin{cases} \mathbf{y}, & \text{si } k = 0 \\ \mathbf{y}^{(k-1)} - \mathbb{T}_{L^*(\mathbf{y})}(\mathbf{y}^{(k-1)}), & \text{si } k > 0 \end{cases}. \quad (1.33)$$

Le résultat équation (1.32) associé à une récurrence implique que pour tout $k \geq 0$, $\|\mathbf{y}^{(k)}\|_2 \leq \gamma^k \|\mathbf{y}\|_2$ et $\mathbf{y} = \sum_{k=0}^{\infty} \mathbb{T}_{L^*(\mathbf{y})}(\mathbf{y}^{(k)})$. Notons respectivement \mathbf{x} et $\mathbf{x}^{(k)}$ les codes associés aux vecteurs \mathbf{y} et $\mathbf{y}^{(k)}$. Par définition de l'opérateur \mathbb{T} , $\|\mathbf{x}^{(k)}\|_{\infty} \leq \|\mathbf{y}^{(k)}\|_2 / \sqrt{\eta N}$. En associant les deux derniers résultats, on obtient directement

$$\|\mathbf{x}\|_{\infty} \leq \sum_{k=0}^{+\infty} \gamma^k \frac{\|\mathbf{y}\|_2}{\sqrt{\eta N}} = \frac{1}{1 - \gamma} \frac{\|\mathbf{y}\|_2}{\sqrt{\eta N}}. \quad (1.34)$$

La preuve du théorème 1.5 a montré que l'hypothèse de trame ajustée et la condition PIM n'interviennent que dans la preuve d'une égalité destinée à fournir une manière algorithmique d'obtenir les représentations de Kashin. L'intuition derrière ce résultat est qu'après la troncature, ce sont les résidus qui deviennent parcimonieux. On peut alors appliquer les résultats vus dans la partie 1.3.

1.3.2.c Caractérisation des solutions approchées

Un résultat plus récent montre qu'il suffit que toute sous matrice carrée de \mathbf{H} soit de rang plein pour que la solution du problème $(\mathbf{P}_{\infty}^{\varepsilon})$ existe (Studer et al., 2014, Lemme 1). Ce résultat est plus général dans le sens où la condition sur la trame est moins forte et qu'il s'applique aux problèmes $(\mathbf{P}_{\infty}^{\varepsilon})$ pour n'importe quelle valeur $\varepsilon \geq 0$. Cette condition sur les sous matrices de \mathbf{H} se formalise avec la notion de *spark*, déjà introduite dans la définition 1.3. Pour l'*acquisition compressée* (cf. partie 1.2), les *trames full spark* sont intéressantes car elles définissent implicitement des morphismes injectifs dans l'espace des observations. Ici, encore une fois, ce sont les résidus des solutions successives qui seront parcimonieux.

La condition de *full spark* est suffisante pour affirmer le résultat suivant.

Théorème 1.6. (Studer et al. (2014), Lemme 2) *Soit \mathbf{H} une matrice full spark. Alors toute solution du problème $(\mathbf{P}_{\infty}^{\varepsilon})$ a au moins $N - M + 1$ valeurs extrêmes. De plus, si $\hat{\mathbf{x}}$ est une solution de $(\mathbf{P}_{\infty}^{\varepsilon})$, alors*

$$\|\hat{\mathbf{x}}\|_{\infty} \leq \frac{1}{\sqrt{N - M + 1}} \|\hat{\mathbf{x}}\|_2 \quad (1.35)$$

Notons que le théorème 1.6 caractérise toutes les solutions du problème $(\mathbf{P}_{\infty}^{\varepsilon})$. Il ne donne en revanche pas de garantie concernant l'existence de ces solutions.

Le rôle de la condition *full spark* dans le théorème 1.6 se comprend facilement avec un raisonnement par l'absurde. Supposons à cet effet qu'il existe des solutions de $(\mathbf{P}_{\infty}^{\varepsilon})$ admettant $K < N - M + 1$ valeurs extrêmes

et notons \hat{x}_0 une telle solution. Nous allons construire à partir de \hat{x}_0 une nouvelle solution \hat{x}' de (P_∞^ε) telle que $\|\hat{x}'\|_\infty < \|\hat{x}_0\|_\infty$. Notons \mathbf{H}_K la sous matrice de \mathbf{H} de taille $M \times (N - K)$ obtenue en extrayant les $N - K$ colonnes de \mathbf{H} correspondant aux indices des valeurs non extrêmes de \hat{x}_0 . Comme $N - K \geq M$, \mathbf{H}_K est forcément de rang M en raison de la condition *full spark*. Dans ce cas, il existe un vecteur z^K de \mathbb{R}^{N-K} vérifiant $\mathbf{H}_K z^K = \mathbf{H} \hat{x}_0$, et l'on peut construire un vecteur $\hat{x}_\alpha = \alpha \hat{x}_0 + (1 - \alpha)z$, où z est un vecteur parcimonieux d'ordre K construit à partir de z^K et α un réel de $]0, 1[$. Pour toute valeur de α , \hat{x}_α est solution de (P_∞^ε) . En jouant avec la valeur de α , on arrive à construire un vecteur \hat{x}_α tel que $\|\hat{x}_\alpha\|_\infty < \|\hat{x}_0\|_\infty$, d'où la contradiction. La figure 1.9 illustre ce raisonnement en dimension 2.

1.3.3 Conception de matrices de codage

Il est difficile de construire de manière déterministe des matrices d'encodage qui satisfont les hypothèses des théorèmes de garanties énoncés dans la partie 1.3.2. C'était déjà le cas pour l'*acquisition compressée*. Encore une fois, c'est l'aléatoire qui vient résoudre le problème en garantissant avec forte probabilité que certaines matrices aléatoires ont la bonne structure. On a identifié deux conditions sur les *trames* qui interviennent dans les résultats théoriques : la condition *PIM* (c.f. définition 1.8) et la condition de *full spark* (c.f. définition 1.3).

Montrer qu'une matrice donnée vérifie la condition *full spark* est un problème combinatoire difficile (Tillmann et Pfetsch, 2014), pour les mêmes raisons que dans la partie 1.2.3.c. On retiendra deux classes de *trames* qui vérifient cette condition (Alexeev et al., 2012). Il s'agit d'une classe de matrices déterministes et d'une classe de matrices aléatoires. Toute matrice obtenue en sous-échantillonnant les lignes d'une matrice de coefficients de Fourier vérifie la condition *full spark*. De même, toute matrice aléatoire dont les entrées sont i.i.d. vérifie la condition *full spark* avec probabilité 1.

On s'appuie dans cette thèse sur un ensemble de résultats sur les matrices aléatoires pour justifier et construire des matrices respectant la condition *PIM* (Lyubarskii et Vershynin, 2010).

Théorème 1.7. *Soit μ un réel compris entre 0 et 1 tel que $N = (1 + \mu)M$. Alors il existe une constante $c > 0$, indépendante de N, M et μ , telle qu'une matrice $M \times N$ issue d'un tirage uniforme sur la variété des matrices orthogonales vérifie la condition *PIM* avec probabilité supérieure à $1 - 2 \exp(-c\mu^2 M)$, et de paramètres*

$$\eta = 1 - \frac{\mu}{4} \quad \text{et} \quad \gamma = \frac{c\mu^2}{-\ln(\mu)}.$$

Remarquons que les valeurs les plus intéressantes sont pour $\mu \ll 1$, mais la probabilité devient plus faible.

Théorème 1.8. *Soit $\mu \in]0, 1[$ tel que $N = (1 + \mu)M$. Soient $\alpha > 0$ et Φ une matrice de taille $N \times N$ de coefficients de Fourier telle que pour*

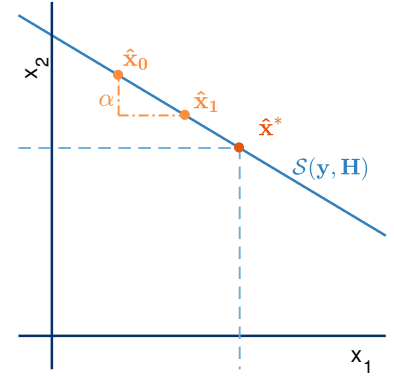


FIGURE 1.9 : Illustration de la contradiction qui apparaît lorsque l'on exhibe en dimension 2 une solution avec $K = 1 < N - N + 2 = 2$ valeurs extrêmes. \hat{x}_0 est une telle solution, dont l'axe contenant la valeur extrême est e_2 . On montre alors que l'on peut construire une solution \hat{x}_1 en longeant l'axe e_1 telle que $\|\hat{x}_1\|_\infty < \|\hat{x}_0\|_\infty$. Ici, \hat{x}^* est la solution optimale.

Les *trames* aléatoires ou obtenues en supprimant les lignes d'une matrice de coefficients de Fourier vérifie la condition *full spark*.

Les matrices rectangulaires orthogonales respectent la condition *PIM* avec forte probabilité.

Les matrices rectangulaires obtenues en sous échantillonnant les lignes de matrices de coefficients de Fourier respectent la condition *PIM* avec forte probabilité.

tout couple d'indices i, j , $|\Phi_{i,j}| \leq \alpha N^{-1/2}$. Soit \mathbf{H} une matrice de taille $M \times N$ obtenue en sélectionnant aléatoirement M lignes de Φ . Alors pour tout $p \in]0, 1[$ il existe une constante $c(p, \alpha)$ telle que la matrice \mathbf{H} satisfait la condition PIM avec probabilité supérieure à $1 - p$, et de paramètres

$$\eta = 1 - \frac{\mu}{4} \quad \text{et} \quad \gamma = \frac{c\mu^2}{\ln^4(N)}.$$

Théorème 1.9. Soit $\mu \geq 1$ tel que $N = (1 + \mu)M$ et Φ une matrice de taille $M \times N$ dont les entrées sont des tirages de lois sous-gaussiennes i.i.d. de paramètre β . Alors il existe deux constantes c et C telles que la matrice $\mathbf{H} = N^{-1/2}\Phi$ satisfait la condition PIM avec probabilité supérieure à $1 - (1 + \mu)^{-M}$, et de paramètres

$$\eta = C\beta\sqrt{\frac{\ln(\lambda)}{\lambda}} \quad \text{et} \quad \gamma = \frac{c}{\lambda},$$

où λ est le niveau de redondance $\lambda = 1 + \mu = \frac{N}{M}$, défini dans la définition 1.6.

Remarquons que ces trois approches pour construire des *trames* respectant la condition PIM ont en commun le fait que lorsque l'on souhaite augmenter la probabilité que la *trame* respecte la condition, alors les valeurs des paramètres η et γ diminuent, ce qui rend la matrice moins intéressante du point de vue de la tâche. Le théorème 1.8 énumère des résultats en bas régime, i.e., $N \in [M, 2M]$ tandis que les théorèmes 1.7 et 1.9 sont vrais pour $N \geq 2M$. Notons finalement que le théorème 1.8 est numériquement intéressant puisqu'il concerne des matrices de coefficients de Fourier, dont le calcul est en $\mathcal{O}(N \ln(N))$.

Un résultat existe aussi pour $\mu > 1$.

On peut en réalité remplacer la matrice de coefficients de Fourier par toute matrice orthogonale.

Les matrices dont les entrées sont des sous-gaussiennes i.i.d. respectent la condition PIM avec forte probabilité.

1.4 ALGORITHMES DE RECHERCHE DE CODES ANTIPARCIMONIEUX

NOUS PASSONS en revue dans cette partie les algorithmes pour le codage antiparcimonieux. Comme pour l'acquisition compressée, les premiers algorithmes résolvent historiquement le problème contraint (\mathbf{P}_∞) puis nous étudierons la formulation variationnelle. Cette fois, c'est l'opération de troncature qui va se retrouver dans l'ensemble des méthodes.

1.4.1 Solution du problème à effort minimal

Cadzow (1971) a accompagné son travail d'un algorithme pour chercher des solutions aux problèmes à effort minimal. La méthode proposée repose sur la résolution du problème dual (Luenberger, 1969) et est détaillée par l'algorithme 1.2. Mais cette méthode nécessite de déterminer le nombre de sous-ensembles des lignes qui forment une famille libre, cf. ligne 1. C'est un problème combinatoire NP difficile. Toutefois, cette méthode est justifiée dans le cadre des problèmes à effort minimaux où M représente par exemple la dimension de l'espace (généralement 3) et N le nombre d'actionneurs intervenant dans le mouvement d'un bras.

Algorithme 1.2 :**Entrées :** Mesure $\mathbf{y} \in \mathbb{R}^M$, Matrice \mathbf{H}

- 1 Déterminer le nombre L , de combinaisons de $M - 1$ lignes de \mathbf{H} qui forment une famille libre ; // Remarquer que $L \leq \binom{M}{M-1}$
 - 2 Stocker les indices des combinaisons dans L vecteurs binaires $\mathbf{i}_1, \dots, \mathbf{i}_L$ de $\{0, 1\}^M$;
 - 3 $\forall \ell \in [1, L], u_\ell \triangleq \frac{|\mathbf{y}^T \mathbf{i}_\ell|}{\|\mathbf{H}^T \mathbf{i}_\ell\|_1}$;
 - 4 $q = \arg \max_{\ell} u_\ell$;
 - 5 $\hat{\boldsymbol{\alpha}} = \frac{\text{signe}(\mathbf{y}^T \mathbf{h}_q)}{\|\mathbf{H}^T \mathbf{h}_q\|_1} \mathbf{h}_q$;
 - 6 $\mathcal{I} = \{\ell, [\mathbf{H}\hat{\boldsymbol{\alpha}}]_\ell \neq 0\}$;
 - 7 **pour chaque** $\ell \in \mathcal{I}$ **faire**
 - 8 | $\hat{\mathbf{x}}_\ell = u_q \text{signe}([\mathbf{H}\hat{\boldsymbol{\alpha}}]_\ell)$;
 - 9 **fin**
 - 10 $\mathcal{I}^c = \{1 \dots N\} \setminus \mathcal{I}$;
 - 11 $\hat{\mathbf{x}}_{\mathcal{I}^c} = (\mathbf{H}_{\mathcal{I}^c}^T \mathbf{H}_{\mathcal{I}^c})^{-1} \mathbf{H}_{\mathcal{I}^c} \mathbf{y}_{\mathcal{I}^c}$;
- Output :** $\hat{\mathbf{x}} \in \mathbb{R}^N$ solution de (\mathbf{P}_∞) .

1.4.2 Recherche d'une représentation de Kashin

Une autre approche, moins coûteuse, s'appuie sur la preuve constructive du théorème 1.5. On rappelle qu'il s'agit d'un processus itératif qui, partant de la décomposition \mathbf{x} d'un vecteur de mesure \mathbf{y} sur une *trame* \mathbf{H} la tronque à un niveau bien choisi puis réitère le même processus avec les résidus. La méthode est détaillée par l'algorithme 1.3.

Cette approche permet d'obtenir une représentation de Kashin après un nombre infini d'itérations. Pour obtenir une représentation exacte après un nombre fini d'itérations, il suffit de ne pas tronquer le résidu de la dernière itération (*i.e.* sauter la ligne 7 de l'algorithme 1.3).

Cette nouvelle approche, présentée algorithme 1.3, résout le problème de complexité de l'algorithme 1.2 pour un cas particulier de matrice d'encodage \mathbf{H} mais nécessite la connaissance avec certitude des constantes caractérisant \mathbf{H} . On a cependant vu dans la partie 1.3.3 que ces constantes ne sont pas faciles à obtenir.

Ces constantes s'obtiennent aussi par recherche exhaustive.

Il est apparu nécessaire de développer d'autres approches, s'appuyant sur les nouvelles avancées en matière de traitement des problèmes d'optimisation depuis la publication du travail de Cadzow en 1971. Les deux méthodes que nous allons étudier relâchent la condition de reconstruction exacte et considèrent plutôt la formulation variationnelle $(\mathbf{V}_\infty^\beta)$.

1.4.3 Algorithme PFA

Plus récemment, une méthode de résolution du problème (\mathbf{P}_∞) a été proposée par Fuchs (2011). Le problème (\mathbf{P}_∞) est d'abord transformé en

Algorithme 1.3 : Algorithme de calcul de représentation de *Kashin*.

Entrées : Mesure $\mathbf{y} \in \mathbb{R}^M$, trame ajustée \mathbf{H} respectant la condition PIM de paramètre η, γ , nombre d'itération N_{it} .

```

1  $L(\mathbf{y}) = \frac{\|\mathbf{y}\|_2}{\sqrt{\eta N}}$ ;
2  $\mathbf{x}^{(0)} = [0 \dots 0]^T$ ;
3  $\mathbf{y}^{(0)} = \mathbf{y}$ ;
4 pour  $r \leftarrow 1$  à  $N_{it}$  faire
5   pour  $n \leftarrow 1$  à  $N$  faire
6     // Calcul de la décomposition des résidus
7      $z_n = \langle \mathbf{y}^{(0)}, \mathbf{h}_n \rangle$ ; // Car  $\mathbf{H}$  est une trame ajustée
8     Tronquer  $z_n = t_{L(\mathbf{y})}(x_n^{(r)})$ ; // c.f. équation (1.30)
9   fin
10   $\mathbf{y}^{(r)} = \mathbf{y}^{(r-1)} - \mathbf{H}\mathbf{z}$ ; // Mise à jour des résidus
11   $\mathbf{x}^{(r)} = \mathbf{x}^{(r-1)} + \sqrt{N}\mathbf{z}$ ; // Mise à jour de la sortie
12 fin

```

Output : $\mathbf{x} \in \mathbb{R}^N$, une représentation de *Kashin* de niveau

$K = (1 - \gamma)^{-1}\eta^{-1/2}$ telle que

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2 \leq \gamma^r \|\mathbf{y}\|_2 \text{ et } \|\mathbf{x}\|_\infty \leq \frac{(1 - \gamma)^{-1}}{\sqrt{\eta N}} \|\mathbf{y}\|_2.$$

un problème d'optimisation convexe

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z} \in \mathbb{R}^N} J_\beta^\infty(\mathbf{z}) \quad (\mathbf{V}_\infty^\beta)$$

$$J_\beta^\infty(\mathbf{z}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2 + \beta \|\mathbf{z}\|_\infty,$$

où β est réel positif ou nul. Notons qu'il s'agit d'une reformulation variationnelle du problème (\mathbf{P}_∞) comme cela a déjà été fait pour (\mathbf{V}_1^β) .

Comme J_β^∞ vue comme une fonction de \mathbf{z} n'est pas différentiable en raison de la norme ℓ^∞ , Fuchs (2011) propose de s'intéresser à sa sous-différentielle. La sous-différentielle d'une fonction f en $\mathbf{x} \in \mathbb{R}^N$, notée $\partial f(\mathbf{x})$, est l'ensemble des vecteurs \mathbf{v} de \mathbb{R}^N vérifiant pour tout \mathbf{x}' de \mathbb{R}^N $f(\mathbf{x}') - f(\mathbf{x}) \geq \mathbf{v}^T(\mathbf{x}' - \mathbf{x})$. Lorsque la fonction f est différentiable, la sous-différentielle est réduite au gradient. Pour $f = \|\cdot\|_\infty$, Fuchs (2011) donne

$$\partial f(\mathbf{0}) = \{\mathbf{v} \in \mathbb{R}^N, \|\mathbf{v}\| \leq 1\}$$

$$\partial f(\mathbf{x}) = \left\{ \mathbf{v} \in \mathbb{R}^N, \|\mathbf{v}\|_1 = 1 \text{ et } \begin{array}{ll} v_n x_n \geq 0 & \text{si } |x_n| = \|\mathbf{x}\|_\infty \\ v_n = 0 & \text{si } |x_n| < \|\mathbf{x}\|_\infty \end{array} \right\}. \quad (1.36)$$

Comme J_β^∞ est convexe, une condition nécessaire et suffisante pour que \mathbf{x} soit solution de $(\mathbf{V}_\infty^\beta)$ est que la sous-différentielle de J_β^∞ en \mathbf{x} contienne le vecteur nul (Bauschke et Combettes, 2011). En développant ce résultat, $\mathbf{x} \in \mathbb{R}^N$ est solution de $(\mathbf{V}_\infty^\beta)$ si et seulement si (Fuchs, 2011, Lemme 1)

$$\exists \mathbf{v} \in \partial \|\cdot\|_\infty(\mathbf{x}), \quad \mathbf{H}^T(\mathbf{H}\mathbf{x} - \mathbf{y}) + \beta \mathbf{v} = \mathbf{0}. \quad (1.37)$$

L'équation (1.37) est utilisée pour construire un algorithme itératif mettant successivement à jour β et \mathbf{x} , et s'arrête lorsque β est suffisamment petit. Comme l'algorithme est peu intuitif, nous ne la détaillerons pas davantage. Pour être complet, nous le donnons en Annexe A.1.

1.4.4 Algorithme FITRA

La formulation variationnelle (V_∞^β) peut se résoudre en utilisant une approche de *splitting*, similairement à l'algorithme ISTA proposé pour résoudre (V_1^β). Pour calculer l'opérateur proximal de la norme ℓ_∞ donné par

$$\forall \mathbf{x} \in \mathbb{R}^N \quad \text{prox}_{\|\cdot\|_\infty}^\beta(\mathbf{x}) = \arg \min_{\mathbf{z} \in \mathbb{R}^N} \|\mathbf{z}\|_\infty + \frac{1}{2\beta} \|\mathbf{x} - \mathbf{z}\|_2^2, \quad (1.38)$$

Studer et Larsson (2013) et Studer et al. (2014) proposent une approximation basées sur des méthodes de troncature. Nous verrons dans la partie 2.1.1.f du chapitre 2 qu'une meilleure approximation est obtenue en utilisant une propriété de conjugaison de l'opérateur proximal. On obtient l'algorithme ITRA, pendant antiparcimonieux de ISTA. La version accélérée, appelée *Fast Iterated Tresholding Recursive Algorithm* (FITRA) et proposée par Studer et Larsson (2013), est détaillée par l'algorithme 1.4. On reconnaît aux lignes 3 et 4 les deux étapes classique d'un algorithme *forward-backward* comme pour ISTA.

Algorithme 1.4 : Algorithme FITRA.

Entrées : Mesure \mathbf{y} , matrice \mathbf{H} et paramètre de régularisation β .

- 1 $L = 2\lambda_{\max}(\mathbf{H}^T \mathbf{H});$ // λ_{\max} désigne la plus grande valeur propre
- 2 **pour** $t \leftarrow 0$ à N_{it} **faire**
 - // Partie différentiable \Rightarrow descente de gradient (forward)
 - 3 $\mathbf{z}^{(t+\frac{1}{2})} = \mathbf{z}^{(t)} - \frac{2}{L} \mathbf{H}^T (\mathbf{H} \mathbf{z}^{(t)} - \mathbf{y});$
 - // Partie non différentiable \Rightarrow opérateur proximal (backward)
 - 4 $\mathbf{z}^{(t+1)} = \text{prox}_{\beta \|\cdot\|_\infty}(\mathbf{z}^{(t+\frac{1}{2})});$
 - 5 Mise à jour de \mathbf{y} afin d'accélérer la convergence, voir Studer et Larsson (2013);
- 6 **fin**

Output : $\hat{\mathbf{x}} = \mathbf{e}^{(N_{it})}$, solution approchée du problème (P_∞).

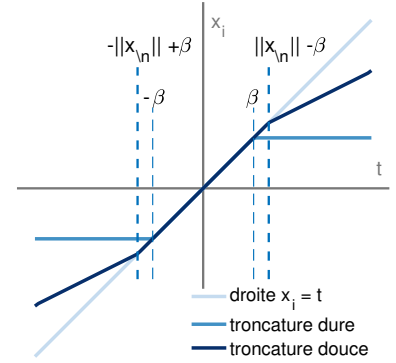


FIGURE 1.10 : Comparaison des opérateurs de troncature dure et douce pour une composante x_n . On tronque à β pour la troncature dure tandis que la troncature douce correspond à l'opérateur $\text{prox}_{\beta \|\cdot\|_\infty}$. Notons qu'ici, $\|\mathbf{x}_{\setminus n}\|_\infty = 2.5\beta$.

1.5 DISCUSSION

NOUS AVONS ÉTUDIÉ dans ce chapitre les représentations *parcimonieuses* et *antiparcimonieuses*. De telles représentations sont obtenues en résolvant un problème inverse régularisé. Nous avons d'abord présenté dans la partie 1.1 l'intérêt de ce codage à travers quelques exemples tirés de la littérature. L'intérêt des codes antiparcimonieux réside dans leur

robustesse aux déformations. Cette robustesse réduit l'erreur de transmission en télécommunication ou l'erreur de reconstruction lors de la binarisation pour le calcul de distance bit à bit.

Dans le cadre de l'*acquisition compressée*, on cherche à reconstruire un signal à partir d'une mesure en plus faible dimension. Ce qui rend la reconstruction possible est une hypothèse structurelle sur le signal, à savoir la parcimonie ici. Cette hypothèse de parcimonie s'est traduite naturellement par une pénalisation de type ℓ_0 , ou sa relaxation convexe ℓ_1 . On a vu dans la partie 1.2 que sous des hypothèses quantifiables sur la matrice d'acquisition de l'observation, on peut reconstruire exactement le signal, ou quantifier l'erreur de reconstruction en cas de données bruitées. On a évoqué la difficulté à concevoir de telles matrices d'acquisition, et on a passé en revue les principaux algorithmes de résolution. Ces algorithmes travaillent soit sur le problème contraint initial, soit sur une relaxation convexe du problème. On a également vu que l'outil commun à l'ensemble de ces méthodes est un opérateur de seuillage.

Dans la partie 1.3, nous avons formalisé et étudié la recherche de codes antiparcimonieux. Il s'agit de codes dont l'énergie est étalée le plus uniformément possible sur tous les coefficients. La recherche de tel codes est formulée à travers la résolution d'un problème inverse sous contrainte de maximiser l'étalement. Comme ce problème est difficile à résoudre, on a préféré travailler sur une version relâchée où l'on pénalise seulement la plus grande composante à travers la norme ℓ_∞ . On espère qu'en forçant la plus grande composante à être petite, on favorisera les plus petites à s'éloigner de 0. On a étudié les garanties d'existences et le comportement des solutions.

Nous avons passé en revue les différents algorithmes permettant d'obtenir des codes antiparcimonieux dans la partie 1.4. À nouveau, une méta-opération est apparue dans chacune des méthodes, à savoir un opérateur de troncature. Cet opérateur limite la valeur des composantes trop grandes, et fait finalement sens dans la mesure où l'on souhaite pénaliser la plus grande composante. Toutefois, nous n'avons pas présenté de formulation probabiliste du problème. Dans le cadre de l'*acquisition compressée*, cette formalisation était apparue naturellement en interprétant la solution de la version lagrangienne du problème comme un estimateur MAP d'un modèle bayésien. Le prochain chapitre répond à cette question en proposant un raisonnement similaire : le pendant antiparcimonieux d'une loi *a priori* de Laplace de la forme $\pi(\mathbf{x}) \propto \exp(-\lambda \|\mathbf{x}\|_1)$, est une loi de la forme $\pi(\mathbf{x}) \propto \exp(-\lambda \|\mathbf{x}\|_\infty)$. Toutefois, la loi *a priori* sur les codes antiparcimonieux qui apparaît n'a pas encore été caractérisée dans la littérature.

Modèle bayésien pour le codage 2 antiparcimonieux

L'INTÉRÊT DU CODAGE antiparcimonieux a été discuté au chapitre 1. On a fait la distinction entre les représentations de *Kashin* qui majorent l'amplitude maximale du signal, et les représentations *démocratiques* qui encadrent l'amplitude maximale. Ces représentations sont par exemple utilisées pour la réduction de facteur de crête, ou *Peak to Average Power Ratio* (PAPR) en anglais (Ilic et Strohmer, 2009).

Dans ce chapitre, on désigne par codage antiparcimonieux les représentations démocratiques. On rappelle qu'un code antiparcimonieux s'obtient en résolvant le problème de minimisation sous contraintes simplifié

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z} \in \mathbb{R}^N} \|\mathbf{z}\|_\infty \quad \text{s.c.} \quad \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2^2 \leq \varepsilon, \quad (\mathbf{P}_\infty^\varepsilon)$$

où \mathbf{y} est un vecteur mesure de \mathbb{R}^M , \mathbf{H} est une matrice d'encodage réelle de dimensions $M \times N$ et ε un réel positif ou nul. Il est aussi intéressant pour des raisons algorithmiques de considérer la formulation variationnelle de $(\mathbf{P}_\infty^\varepsilon)$

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z} \in \mathbb{R}^N} J_\beta^\infty(\mathbf{z}) \quad (\mathbf{V}_\infty^\beta)$$
$$J_\beta^\infty(\mathbf{z}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2^2 + \beta \|\mathbf{z}\|_\infty. \quad (2.1)$$

Le lien entre les problèmes contraint $(\mathbf{P}_\infty^\varepsilon)$ et variationnel $(\mathbf{V}_\infty^\beta)$ se fait à travers la valeur de β . Ce choix n'est pas trivial. Rappelons que la recherche de codes antiparcimonieux *n'est pas un problème inverse*. L'objectif est plutôt d'identifier un code faisant sens par rapport à une représentation donnée par \mathbf{H} et ayant de bonnes propriétés de robustesse.

Contrairement à l'*acquisition compressée*, la recherche de codes antiparcimonieux ne dispose pas à ce jour de formulation probabiliste. Pourtant, les avantages d'une telle formulation sont multiples. D'abord, le paramètre β peut être intégré au modèle bayésien afin de concevoir des algorithmes ne nécessitant aucun réglage de paramètres. Outre une meilleure compréhension du problème, le paradigme bayésien laisse plus de souplesse dans la conception d'estimateurs et fournit des outils pour les approcher, telles que les méthodes de *Monte Carlo par chaîne de Markov* (MCMC).

Nous proposons dans ce chapitre une formulation bayésienne du codage antiparcimonieux, appelée *Bayesian Antispase Coding 1* (BAC-1). Le modèle est élaboré en interprétant la formulation variationnelle $(\mathbf{V}_\infty^\beta)$

comme la somme d'une log-vraisemblance gaussienne et de la log-densité d'une loi *a priori*. La densité de cette loi *a priori*, appelée loi *démocratique*, fait intervenir la norme ℓ_∞ , et n'a pas été étudiée à notre connaissance.

La partie 2.1 est consacrée à la construction et à l'étude de la loi *démocratique*. La partie 2.2 présente le modèle bayésien proposé pour le codage antiparcimonieux. On étudie les performances numériques de l'algorithme dans la partie 2.3. On vérifiera que les estimateurs bayésiens coïncident bien avec les solutions d'algorithmes déterministes. On utilise dans la partie 2.3 le modèle BAC-1 pour la réduction de PAPR. La partie 2.5 conclut ce chapitre et nous discuterons des perspectives de ce travail dans la partie 2.6.

Ces travaux sont publiés dans Elvira et al. (2016, 2017a, 2017d). Les codes MATLAB/ C++ des algorithmes présentés dans ce chapitre sont disponibles en ligne.



2.1 LA LOI DÉMOCRATIQUE

NOUS CARACTÉRISONS dans cette partie une nouvelle loi *a priori* favorisant les codes antiparcimonieux. En s'appuyant sur l'interprétation de la formulation variationnelle¹ (\mathbf{V}_∞^β) de ce problème comme l'estimateur MAP (cf. (1.3)) d'un modèle bayésien, la fonction de coût J_β^∞ (cf. (2.1)) est la somme d'un terme de vraisemblance gaussienne et du logarithme négatif de la densité d'une loi *a priori* (Gribonval, 2011 ; Pereyra, 2016b). Cette loi est de la forme $p(\mathbf{x}) \propto \exp(-\lambda \|\mathbf{x}\|_\infty)$ mais ne correspond à aucune densité usuelle. Cette nouvelle loi est nommée *loi démocratique* en référence aux représentations démocratiques. La partie 2.1.1 est consacrée à l'étude des principales propriétés de la loi, tandis que la partie 2.1.2 étudie l'échantillonnage.

2.1.1 Propriétés de la loi démocratique

2.1.1.a Définition de la loi démocratique

La définition 2.1 introduit la loi *démocratique*.

Définition 2.1. Soit \mathbf{x} un vecteur aléatoire à valeur dans \mathbb{R}^N . On dit que \mathbf{x} est distribué suivant une loi démocratique de paramètre λ , notée $\mathcal{D}_N(\lambda)$, si sa densité de probabilité s'écrit

$$p(\mathbf{x}; \lambda) = \frac{1}{C_N(\lambda)} \exp(-\lambda \|\mathbf{x}\|_\infty) \quad (2.2)$$

avec $C_N(\lambda) \triangleq N! \left(\frac{2}{\lambda}\right)^N$.

La densité de la loi *démocratique* en dimension 2 pour $\lambda = 3$ est illustrée à la figure 2.1. On remarquera qu'en dimension 2, la densité ressemble à une pyramide à base carrée dont la hauteur correspond aux

1. On rappelle

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z} \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2 + \beta \|\mathbf{z}\|_\infty. \quad (\mathbf{V}_\infty^\beta)$$

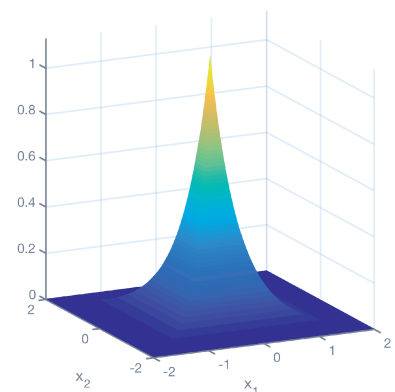


FIGURE 2.1 : Densité de probabilité de la loi démocratique $\mathcal{D}_2(3)$.

lignes de niveau de la densité. Cette intuition se généralise en dimension supérieure, et nous nous en servirons plus tard pour construire un échantillonneur.

En dimension 2, la loi *démocratique* $\mathcal{D}_2(\lambda)$ peut être obtenue en appliquant une rotation d'angle $\pi/4$ à un vecteur aléatoire dont chaque composante est distribuée suivant une loi de Laplace de paramètre $\sqrt{2}\lambda$. Cette propriété ne se généralise cependant pas aux dimensions supérieures.

La loi *démocratique* appartient également à la famille exponentielle. La factorisation associée est

$$p(\mathbf{x}; \lambda) = a(\mathbf{x}) b(\lambda) \exp(\eta(\lambda)T(\mathbf{x})) \tag{2.3}$$

où $a(\mathbf{x}) = 1, b(\lambda) = 1/C_N(\lambda), \eta(\lambda) = -\lambda$ et $T(\mathbf{x}) = \|\mathbf{x}\|_\infty$ définit une statistique suffisante.

Le calcul de la constante de normalisation $C_N(\lambda)$ dans (2.2) et plus généralement l'étude de la loi démocratique s'appuient sur les propriétés de symétrie de la norme ℓ_∞ . On introduit pour exploiter ces symétries les doubles cônes $(C_n)_{n \in \llbracket 1, N \rrbracket}$ de \mathbb{R}^N définis par

$$\forall n \in \llbracket 1, N \rrbracket, \quad C_n \triangleq \left\{ \mathbf{x}, \forall j |x_j| \leq |x_n| \right\}. \tag{2.4}$$

Les cônes $(C_n)_{n \in \llbracket 1, N \rrbracket}$ forment une partition² de \mathbb{R}^N qui sera exploitée dans les calculs d'intégrales. Si un vecteur \mathbf{x} de \mathbb{R}^N appartient au cône C_n , alors on peut réécrire $\|\mathbf{x}\|_\infty = |x_n|$. Dans ce cas, la n^e composante est appelée *composante dominante*, tandis que les $N - 1$ autres composantes sont dites *non dominantes*. Notons que la notion de *composante dominante* renvoie au concept de valeur extrême introduit dans la partie 1.3.

Nous terminons cette section par le calcul de la constante de normalisation $C_N(\lambda)$ dans (2.2), en guise d'illustration de l'utilisation des cônes (C_n) . C'est l'unique preuve qui sera détaillée dans ce chapitre. Le lecteur intéressé trouvera les preuves des résultats de ce chapitre en annexe A.

Calcul. Il est facile de remarquer que la fonction $f : \mathbf{x} \rightarrow \exp(-\lambda \|\mathbf{x}\|_\infty)$ est intégrable sur \mathbb{R}^N par équivalence des normes en dimension finie. Le calcul de la valeur de l'intégrale se simplifie en utilisant les multiples symétries de la norme ℓ_∞ . En particulier, la **parité de la fonction** et les **symétries par rapports aux cônes** C_n amènent

$$\begin{aligned} \int_{\mathbb{R}^N} \exp(-\lambda \|\mathbf{x}\|_\infty) d\mathbf{x} &= 2^N N \int_{C_1^+} \exp(-\lambda \|\mathbf{x}\|_\infty) d\mathbf{x} \\ &= 2^N N \int_{\mathbb{R}_+} x_1^{N-1} \exp(-\lambda x_1) dx_1 \\ &= \frac{2^N}{\lambda^N} N \Gamma(N). \end{aligned}$$

Ce résultat provient de la réécriture du maximum comme

$$\max(x, y) = \frac{x + y + |x - y|}{2}.$$

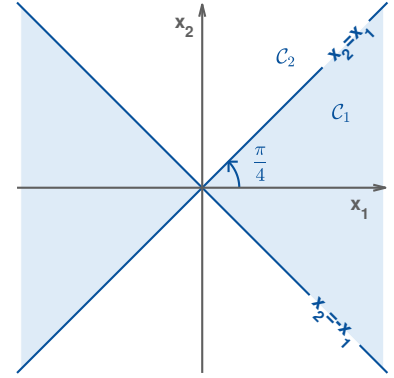


FIGURE 2.2 : Doubles cônes C_1 (bleu clair) et C_2 (non colorié) de \mathbb{R}^2 .

2. Ce n'est pas exactement une partition de \mathbb{R}^N car deux cônes peuvent partager une frontière. Cependant, ces frontières sont des droites de mesure nulle.

Où $C_1^+ \triangleq \{\mathbf{x} \in C_1, x_1 \geq 0\}$.

On fait appel au changement de variable $t_1 = \lambda x_1$.

□

2.1.1.b Fonction caractéristique

La fonction caractéristique caractérise de façon unique une variable aléatoire réelle. Pour les variables à densité telle que la loi démocratique, on l'obtient en calculant la transformée de Fourier de la densité. Elle est notamment utile pour calculer les moments à partir des zéros de ses dérivées successives, ou pour étudier l'infinie divisibilité de la distribution.

Le cas $N = 1$ est trivial puisque la loi *démocratique* et la loi de Laplace coïncident. Le cas de la dimension 2 est traité dans la propriété suivante.

Propriété 2.1. La fonction caractéristique ϕ de la distribution $\mathcal{D}_2(\lambda)$ est donnée par

$$\forall t \in \mathbb{R}^2, \quad \phi_{\mathcal{D}_2(\lambda)}(t) = \frac{\lambda^4}{(\lambda^2 + (t_1 + t_2)^2)(\lambda^2 + (t_1 - t_2)^2)}. \quad (2.5)$$

Dans le cas général, le calcul de la transformée de Fourier de la densité (2.2) nécessite à nouveau de séparer les domaines d'intégration. Quelque soit la dimension N , les calculs mènent à une somme finie de termes, mais nous n'avons pas identifié de forme factorisée.

2.1.1.c Moments

La propriété suivante donne les deux premiers moments de la loi démocratique ainsi que l'espérance du module d'une composante

Propriété 2.2. Soit $x = [x_1, \dots, x_N]^T$ un vecteur aléatoire réel distribué suivant la loi démocratique $\mathcal{D}_N(\lambda)$. Les moyennes ainsi que les covariances de chaque composante sont données par

$$\mathbb{E}[x_n] = 0 \quad \forall n \in \{1, \dots, N\} \quad (2.6)$$

$$\mathbb{E}[|x_n|] = \frac{N+1}{2\lambda} \quad \forall n \in \{1, \dots, N\} \quad (2.7)$$

$$\text{var}[x_n] = \frac{(N+1)(N+2)}{3\lambda^2} \quad \forall n \in \{1, \dots, N\} \quad (2.8)$$

$$\text{cov}[x_i, x_j] = 0 \quad \forall i \neq j. \quad (2.9)$$

Les calculs sont disponibles en annexe A.2.1. Notons que la preuve nécessite d'anticiper le corollaire 2.1 concernant les lois marginales d'un vecteur démocratique.

La figure 2.3 illustre le comportement de l'espérance du module et de la variance d'une composante en fonction des paramètres λ et N . On s'intéresse au module puisqu'à terme, on souhaite utiliser la loi démocratique pour favoriser un comportement de la forme $x_n \simeq \pm \|\mathbf{y}\|_2 / \sqrt{N}$ où \mathbf{y} est une observation.

2.1.1.d Lois marginales

On s'intéresse dans cette partie aux lois marginales d'un vecteur distribué suivant la loi démocratique. La propriété suivante donne la loi marginale de n'importe quel sous-ensemble de composante

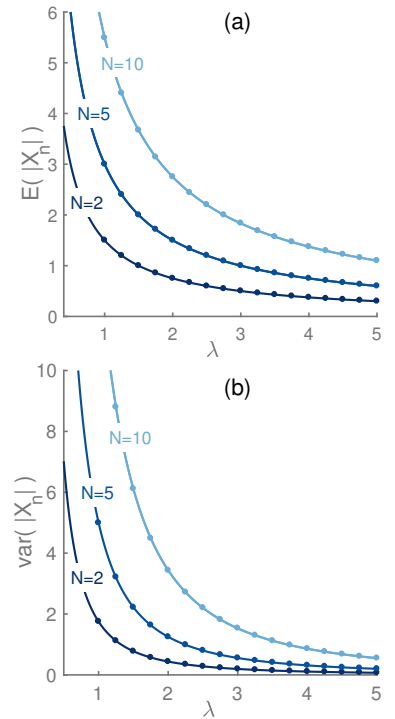


FIGURE 2.3 : (a) Espérance théoriques (courbes bleues) et empiriques (points) du module d'une composante x_n d'un vecteur x distribué suivant la loi $\mathcal{D}_N(\lambda)$ en fonction de la valeur du paramètre λ , pour différentes valeurs de N . (b) Même chose pour la variance d'une composante. Chaque estimateur est calculé sur 10^4 échantillons.

Propriété 2.3. Soit $\mathbf{x} = [x_1, \dots, x_N]^T$ un vecteur aléatoire réel distribué suivant la loi démocratique $\mathcal{D}_N(\lambda)$. Pour tout entier positif $J < N$, on note \mathcal{K}_J un sous-ensemble à J éléments de $\{1, \dots, N\}$, et $\mathbf{x}_{\setminus \mathcal{K}_J}$ le sous-vecteur de \mathbf{x} privé des J éléments indexés par \mathcal{K}_J . La loi du sous-vecteur $\mathbf{x}_{\setminus \mathcal{K}_J} \in \mathbb{R}^{N-J}$ est donnée par

$$p(\mathbf{x}_{\setminus \mathcal{K}_J}) = \frac{2^J}{C_N(\lambda)} \sum_{j=0}^J \binom{J}{j} \frac{(J-j)!}{\lambda^{J-j}} \|\mathbf{x}_{\setminus \mathcal{K}_J}\|_{\infty}^j \times \exp(-\lambda \|\mathbf{x}_{\setminus \mathcal{K}_J}\|_{\infty}). \quad (2.10)$$

La preuve complète est disponible en annexe A.3.1.

La loi marginale d'une composante s'obtient comme un corollaire de la propriété 2.3.

Corollaire 2.1. Soit $\mathbf{x} = [x_1, \dots, x_N]^T$ un vecteur aléatoire réel distribué suivant la loi démocratique $\mathcal{D}_N(\lambda)$. Les lois marginales des composantes x_n ($n = 1 \dots N$) de \mathbf{x} sont distribuées selon le mélange à N composantes suivant

$$x_n \sim \frac{1}{N} \sum_{j=1}^N d\mathcal{G}(j, \lambda), \quad (2.11)$$

où $d\mathcal{G}(a, b)$ (pour double-Gamma) est la loi Gamma symétrisée, définie ici comme la généralisation par symétrie sur \mathbb{R} de la loi Gamma $\mathcal{G}(a, b)$.

Les marginales de la loi démocratique sont donc identiquement distribuées suivant un mélange de loi Gamma symétrisées.

La figure 2.4 montre la loi démocratique $\mathcal{D}_2(\lambda)$. On observe que la marginale a tendance à être plus plate autour de 0 que la distribution projetée. Ce phénomène est formalisé à travers le théorème suivant.

Théorème 2.1. Soit $\mathbf{x} = [x_1, \dots, x_N]^T$ un vecteur aléatoire réel distribué suivant la loi démocratique $\mathcal{D}_N(\lambda)$. Alors la variable aléatoire renormalisée $\frac{\lambda}{N} x_n$ converge en distribution vers la loi uniforme sur $[-1, 1]$, notée $\mathcal{U}([-1, 1])$.

La preuve du théorème est disponible en annexe A.3.2 et s'appuie sur la convergence de la fonction caractéristique.

La figure 2.5 illustre la convergence des lois marginales renormalisées vers la loi uniforme sur $[-1, 1]$. On peut comprendre sans faire la preuve l'origine de ce résultat. Il suffit de factoriser la densité (2.11) suivant

$$p(x_n) = \frac{\lambda}{2N} \left(\sum_{j=0}^{N-1} \frac{|\lambda x_n|^j}{j!} \right) \exp(-\lambda |x_n|). \quad (2.12)$$

Or, le terme **entre parenthèses** correspond au développement limité de l'exponentielle évaluée en $\lambda |x_n|$. Lorsque N augmente, ce développement limité va s'annuler avec le terme $\exp(-\lambda |x_n|)$ sur l'intervalle où l'approximation est bonne. En dehors de cet intervalle, c'est l'exponentielle qui domine et le produit vaut 0. On reconnaît l'indicatrice d'un segment, qui s'interprète comme une loi uniforme en probabilité.

Il existe d'autres réécritures de (2.11) utilisant la fonction Gamma tronquée.

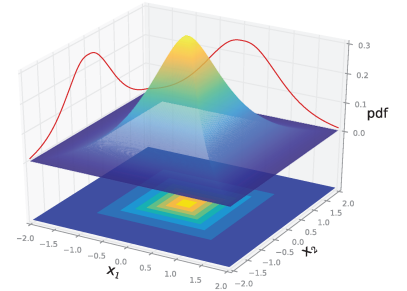


FIGURE 2.4 : La figure 3D représente la densité de la loi $\mathcal{D}_2(\lambda)$. Les courbes rouges 2D sont les marginales x_1 et x_2 , respectivement.

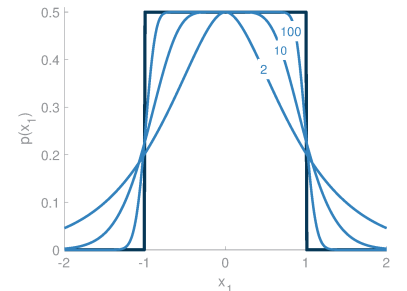


FIGURE 2.5 : La courbe bleue foncée correspond à la densité de la loi uniforme sur $[-1, 1]$. Les courbes bleues claires indiquent les densités marginales renormalisées de la loi démocratique $\mathcal{D}_N(2)$, où la valeur de N est indiquée sur la courbe.

Ce résultat confirme l'intérêt de la loi *démocratique* pour favoriser des représentations à faible amplitude. L'amplitude maximale favorisée est en sus proportionnelle au paramètre λ . Mais ce résultat souligne aussi une faiblesse de la modélisation : les amplitudes proches de zéros sont autant favorisées que les amplitudes proches de la composante dominante par l'action de la loi uniforme.

2.1.1.e Lois conditionnelles

Les lois conditionnelles sont l'ingrédient principal d'un échantillonneur de Gibbs. Pour les composantes d'un vecteur distribué selon la loi démocratique, on procède en deux temps. On donne d'abord la loi d'une composante conditionnellement aux autres et à un cône \mathcal{C}_n , défini en (2.4). Nous sommerons ensuite sur tous les cônes.

La propriété suivante concerne l'équiprobabilité des cônes $(\mathcal{C}_n)_n$. C'est une conséquence directe des propriétés de symétries de la loi.

Propriété 2.4. Soit $\mathbf{x} = [x_1, \dots, x_N]^T$ un vecteur aléatoire réel distribué suivant la loi démocratique $\mathcal{D}_N(\lambda)$. La probabilité qu'un vecteur \mathbf{x} de \mathbb{R}^N appartienne au cône \mathcal{C}_n est donné par

$$\mathbb{P}[\mathbf{x} \in \mathcal{C}_n] = \frac{1}{N}. \quad (2.13)$$

Le lemme suivant regroupe les lois conditionnelles des *composantes dominantes* et *non dominantes*. Notons que c'est le conditionnement au cône qui détermine si la composante est dominante ou non.

Lemme 2.1. Soit $\mathbf{x} = [x_1, \dots, x_N]^T$ un vecteur aléatoire réel distribué suivant la loi démocratique $\mathcal{D}_N(\lambda)$. Alors

$$x_n | \mathbf{x} \in \mathcal{C}_n \sim d\mathcal{G}(N, \lambda) \quad (2.14)$$

$$\mathbf{x}_{\setminus n} | x_n, \mathbf{x} \in \mathcal{C}_n \sim \prod_{j \neq n} \mathcal{U}(-|x_n|, |x_n|) \quad (2.15)$$

$$\mathbf{x}_{\setminus n} | \mathbf{x} \in \mathcal{C}_n \sim \mathcal{D}_{N-1}(\lambda) \quad (2.16)$$

$$\mathbb{P}[\mathbf{x} \in \mathcal{C}_n | \mathbf{x}_{\setminus n}] = \frac{1}{1 + \lambda \|\mathbf{x}_{\setminus n}\|_\infty} \quad (2.17)$$

$$\mathbb{P}(\mathbf{x}_{\setminus n} | \mathbf{x} \notin \mathcal{C}_n) = \frac{\lambda}{N-1} \frac{\|\mathbf{x}_{\setminus n}\|_\infty}{C_{N-1}(\lambda)} \exp(-\lambda \|\mathbf{x}_{\setminus n}\|_\infty). \quad (2.18)$$

La preuve du lemme est disponible en annexe A.4.1. Nous insistons sur deux résultats. Selon (2.14), la loi de la composante dominante est une loi Gamma symétrisée. Les deux modes de cette loi Gamma symétrisée sont $\pm(N-1)/\lambda$. Dans le contexte du codage antiparcomionieux, on veut que cette composante dominante soit précisément $\pm \|\mathbf{y}\|_2 / \sqrt{N}$. C'est en ajustant la valeur du paramètre λ que l'on va obtenir ce comportement. En revanche, les *composantes non dominantes* sont selon (2.15) *indépendantes et identiquement distribuées* suivant une loi uniforme³. Utilisé dans

3. Dont l'amplitude est déterminée par la connaissance de la composante dominante, car pour tout n , $|x_n| \leq \|\mathbf{x}\|_\infty$.

un échantillonneur de Gibbs, cela signifie que la composante non dominante est *a priori* uniformément répartie sur $[-|x_n|, |x_n|]$. C'est le terme de vraisemblance qui va pousser le coefficient à se rapprocher de la direction assurant la meilleure reconstruction. Notons que l'on aurait préféré que les valeurs autour de 0 soient peu probables.

L'équation (2.16) est un résultat de consistance : la loi d'un vecteur démocratique privé de sa composante dominante est encore une loi démocratique. Notons la différence avec la loi marginale donnée au théorème 2.1 : on marginalise ici la composante dominante.

Les deux derniers résultats (2.17) et (2.18) n'ont pas d'interprétation physique intéressante dans ce contexte. Ces résultats seront principalement exploités pour les preuves en annexe A.

Le dernier résultat de cette section donne la loi conditionnelle d'une composante $x_n | \mathbf{x}_{\setminus n}$ d'un vecteur démocratique

Propriété 2.5. Soit $\mathbf{x} = [x_1, \dots, x_N]^T$ un vecteur aléatoire réel distribué suivant la loi démocratique $\mathcal{D}_N(\lambda)$. La loi conditionnelle d'une composante sachant les autres s'écrit

$$p(x_n | \mathbf{x}_{\setminus n}) = (1 - c_n) \frac{1}{2 \|\mathbf{x}_{\setminus n}\|_\infty} \mathbb{1}_{\mathcal{I}_n}(x_n) + c_n \frac{\lambda}{2} e^{-\lambda(|x_n| - \|\mathbf{x}_{\setminus n}\|_\infty)} \mathbb{1}_{\mathbb{R} \setminus \mathcal{I}_n}(x_n) \quad (2.19)$$

où $c_n \triangleq \mathbb{P}[\mathbf{x} \in \mathcal{C}_n | \mathbf{x}_{\setminus n}]$ est donnée équation (2.17), $\mathbb{1}_{\mathcal{A}}(\cdot)$ est la fonction indicatrice⁴ de l'ensemble \mathcal{A} et \mathcal{I}_n est définie par

$$\mathcal{I}_n \triangleq \left(-\|\mathbf{x}_{\setminus n}\|_\infty, \|\mathbf{x}_{\setminus n}\|_\infty \right). \quad (2.20)$$

La preuve est donnée en annexe A.4.2. La densité de la loi conditionnelle (2.19) est un mélange à trois composantes, formée de deux lois exponentielles translatées et d'une loi uniforme sur un intervalle centré en 0. Les poids associés à cette loi conditionnelle sont donnés au lemme 2.1. La figure 2.6 montre les densités conditionnelles de $x_n | \mathbf{x}_{\setminus n}$ pour différentes valeurs de $\|\mathbf{x}_{\setminus n}\|_\infty$. On notera que ces lois ne dépendent pas de la dimension N du vecteur.

En plus de son utilité pour l'inférence bayésienne, la propriété 2.5 sera exploitée dans la partie 2.1.2 pour construire un échantillonneur de la loi démocratique.

2.1.1.f Opérateur proximal de la neg-log-densité

On verra dans la partie 2.1.2 que l'on peut s'appuyer sur les méthodes proximales pour générer des échantillons suivant la loi démocratique. On s'intéresse ici à l'opérateur proximal de la neg-log-densité de la loi démocratique, donnée par

$$\text{prox}_{\delta\lambda \|\cdot\|_\infty}(\mathbf{x}) = \arg \min_{\mathbf{u} \in \mathbb{R}^N} \delta\lambda \|\mathbf{u}\|_\infty + \frac{1}{2} \|\mathbf{x} - \mathbf{u}\|_2^2. \quad (2.21)$$

4. au sens des distributions, soit

$$\mathbb{1}_{\mathcal{A}}(x) = \begin{cases} 1 & \text{si } x \in \mathcal{A} \\ 0 & \text{sinon.} \end{cases}$$

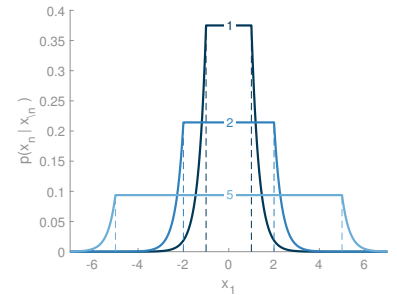


FIGURE 2.6 : Densité conditionnelle de $x_n | \mathbf{x}_{\setminus n}$ de la loi démocratique $\mathcal{D}_N(2)$, pour 3 valeurs de $\|\mathbf{x}_{\setminus n}\|_\infty$, indiquées par des pointillés sur les courbes.

L'opérateur proximal est parfois noté $\text{prox}_{\lambda \|\cdot\|_\infty}^\delta$ dans la littérature. Nous suivons ici la notation utilisée par Moreau (1962).

Notons qu'ici, λ est le paramètre de la loi *démocratique* tandis que $\delta > 0$ est le paramètre de régularisation associé à l'opérateur proximal.

L'opérateur proximal de la neg-log-vraisemblance correspond à l'opérateur proximal utilisé au chapitre 1 dans l'algorithme FITRA (cf. Éq. (1.38)). Ce résultat provient à nouveau de la correspondance entre le problème variationnel (\mathbf{V}_∞^β) et l'estimateur MAP du modèle bayésien que nous proposons dans ce chapitre.

Au chapitre 1 l'opérateur (2.21) était approché avec des méthodes d'optimisation convexe. Nous proposons de le calculer en s'appuyant sur les propriétés de l'opérateur proximal et la dualité des normes ℓ_1 et ℓ_∞ . En effet, par dualité (Moreau, 1962)

$$\text{prox}_{\lambda\|\cdot\|_\infty}(\mathbf{x}) = \mathbf{x} - \lambda \Pi_{\{\mathbf{u}, \|\mathbf{u}\|_1 \leq 1\}}(\mathbf{x}), \quad (2.22)$$

où Π est le projecteur sur la boule ℓ_1 . Ce projecteur n'a pas d'écriture explicite, mais il existe nombreuses techniques d'approximation à faible coût numérique. Une revue de ces méthodes a été réalisée par Condat (2015).

Depuis la première publication de nos résultats, d'autres travaux utilisent la propriété de dualité pour approcher numériquement (2.21), voir par exemple Luu et al. (2017).

2.1.2 Méthodes d'échantillonnage

Cette section compare trois méthodes d'échantillonnage de la loi *démocratique* : une méthode exacte, un échantillonneur de Gibbs et une méthode proximale. L'échantillonneur exact est bien sûr le plus élégant et le plus efficace numériquement (cf. la prochaine partie). Cependant, on ne peut plus l'utiliser pour l'échantillonnage d'une loi *a posteriori* dans le cadre d'un modèle bayésien. Inversement, l'échantillonneur de Gibbs et *proximal Metropolis-adjusted Langevin algorithm* (P-MALA) peuvent facilement être adaptés pour échantillonner suivant une loi *a posteriori*. Les codes MATLAB des trois algorithmes sont disponibles [en ligne](#).



2.1.2.a Échantillonneur exact

On a explicité au lemme 2.1 les lois des composantes dominantes et non dominantes. Ces deux lois sont conditionnées au cône qui contient la composante dominante. Ces résultats suggèrent la procédure suivante, décrite par l'algorithme 2.1. On choisit d'abord le cône C_n qui contiendra l'échantillon. On a vu à la propriété 2.4 que les cônes sont équiprobables. Connaissant ce cône, on tire la valeur de la composante dominante selon une loi Gamma, cf. Éq. (2.14), ainsi que son signe, uniformément. Sachant la valeur de x_n , on tire indépendamment et uniformément les valeurs des autres composantes sur $[-|x_n|, |x_n|]$, cf. Éq. (2.15). Mathématiquement, il faut intégrer la densité de la loi démocratique par rapport aux cônes $(C_n)_n$

selon

$$\begin{aligned} p(\mathbf{x}) &= \sum_{n=1}^N p(\mathbf{x}_{\setminus n} | x_n, \mathbf{x} \in \mathcal{C}_n) p(x_n | \mathbf{x} \in \mathcal{C}_n) P[\mathbf{x} \in \mathcal{C}_n] \\ &= \sum_{n=1}^N \left[\prod_{j \neq n} p(x_j | x_n, \mathbf{x} \in \mathcal{C}_n) \right] p(x_n | \mathbf{x} \in \mathcal{C}_n) P[\mathbf{x} \in \mathcal{C}_n] \quad (2.23) \end{aligned}$$

On utilise la loi des probabilités composées.

où $P[\mathbf{x} \in \mathcal{C}_n]$, $p(x_n | \mathbf{x} \in \mathcal{C}_n)$ et $p(x_j | x_n, \mathbf{x} \in \mathcal{C}_n)$ sont respectivement données aux équations (2.13) à (2.15).

Concernant la complexité numérique, cet échantillonneur nécessite une réalisation d'une loi Gamma et $N - 1$ réalisations d'une loi uniforme.

Algorithme 2.1 : Échantillonneur exact de la loi *démocratique*.

Entrées : Paramètre $\lambda > 0$, dimension N

```
// Générer le cône contenant l'échantillon
1 Tirer  $n_{\text{dom}}$  uniformément parmi  $\{1, \dots, N\}$ ;
// Générer la valeur de la composante dominante
2 Tirer  $x_{n_{\text{dom}}}$  selon l'équation (2.14);
// Générer les valeurs des composantes non dominantes
3 pour  $j \leftarrow 1$  à  $N$  ( $j \neq n_{\text{dom}}$ ) faire
4   | Tirer  $x_j$  selon l'équation (2.15);
5 fin
```

Output : $\mathbf{x} = [x_1, \dots, x_N]^T \sim \mathcal{D}_N(\lambda)$

2.1.2.b Échantillonneur de Gibbs

Comme évoquée précédemment, la densité de la variable aléatoire $x_n | \mathbf{x}_{\setminus n}$ donnée à la propriété 2.5 peut être exploitée pour concevoir un échantillonneur de Gibbs. Cet échantillonneur, décrit par l'algorithme 2.2, consiste à générer successivement chaque composante conditionnellement aux autres, dans un ordre aléatoire.

Concernant la complexité numérique, une itération de l'échantillonneur de Gibbs nécessite

- N réalisations d'une Bernoulli,
- N réalisations d'une loi uniforme ou d'une loi Exponentielle, les proportions étant aléatoires,
- *facultatif* : une réalisation d'une loi uniforme sur les permutations de $\llbracket 1, N \rrbracket$ pour améliorer les propriétés de mélange. En pratique on tire N fois d'une loi uniforme sur les entiers (Knuth, 1997, Algorithm P).

2.1.2.c P-ULA, P-MALA

Une seconde alternative pour l'échantillonnage de la loi *démocratique* s'appuie sur les méthodes de Monte Carlo proximales, ou P-MALA (Peyra, 2016a). P-MALA combine des outils d'analyse convexe aux méthodes

Algorithme 2.2 : Échantillonneur de Gibbs pour la loi *démocratique*.

Entrées : Paramètre λ , dimension N , période de chauffe T_{bi} .

```

1 Initialiser  $\mathbf{x}^{(0)} \sim \mathcal{N}(\mathbf{0}, \lambda \mathbb{I}_N)$ ;
2 pour  $t \leftarrow 1$  à  $T_{\text{bi}} + 1$  faire
3   pour  $n \leftarrow 1$  à  $N$  faire
4      $\mathbf{x}_{\setminus n}^{(t)} = [x_1^{(t)}, \dots, x_{n-1}^{(t)}, x_{n+1}^{(t-1)}, \dots, x_N^{(t-1)}]^T$ ;
5     Générer  $x_n^{(t)}$  suivant l'équation (2.19);
6      $\mathbf{x}^{(t)} = [x_1^{(t)}, \dots, x_{n-1}^{(t)}, x_n^{(t)}, x_{n+1}^{(t-1)}, \dots, x_N^{(t-1)}]^T$ ;
7   fin
8   Set  $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)}$ ;
9 fin

```

Output : $\mathbf{x}^{(t)} \sim \mathcal{D}_N(\lambda)$.

MCMC afin de générer des échantillons d'une loi de probabilité multidimensionnelle, log-concave mais éventuellement non différentiable. C'est le cas de la densité de la loi *démocratique* en raison de la norme ℓ_∞ . Notons que les méthodes d'échantillonnage classiques, telle que *Metropolis-adjusted Langevin algorithm* (MALA) (Robert et Casella, 2004, Partie 7.8.5) ou *Hamiltonian Monte Carlo* (HMC) (Neal, 2012) ne peuvent être appliquées ici car elles nécessitent que la densité soit \mathcal{C}^1 .

À l'instar de MALA, P-MALA repose sur une diffusion de Langevin. Un processus de diffusion de Langevin $\{\mathbf{Z}(t), 0 \leq t \leq T\}$ est défini comme la solution d'une équation différentielle stochastique de la forme

$$\begin{cases} \forall t, & d\mathbf{Z}(t) = \frac{1}{2} \nabla \log \pi(\mathbf{Z}(t)) dt + d\mathbf{W}(t), \\ \mathbf{Z}(0) & = \mathbf{z}_0 \end{cases}, \quad (2.24)$$

où π est une distribution cible non nulle différentiable presque partout, \mathbf{W} est un mouvement Brownien de \mathbb{R}^N et $\mathbf{z}_0 \in \mathbb{R}^N$ est la condition initiale. La solution \mathbf{Z} de l'équation différentielle stochastique converge en distribution vers la distribution cible π . Il est alors possible, en utilisant un schéma de discrétisation bien choisi, de générer des échantillons $(\mathbf{Z}[n])_n$ distribués suivant π . Lorsque la distribution π est non-différentiable, P-MALA propose de remplacer π par π_δ , sa δ -approximation de Moreau, définie comme (Moreau, 1962)

$$\forall \mathbf{x} \in \mathbb{R}^N \quad \pi_\delta(\mathbf{x}) = \sup_{\mathbf{u} \in \mathbb{R}^N} \frac{1}{\kappa} \pi(\mathbf{u}) \exp\left(-\frac{1}{2\delta} \|\mathbf{u} - \mathbf{x}\|_2^2\right), \quad (2.25)$$

où κ est la constante de normalisation. La distribution π_δ se comprend comme la distribution π vue au travers d'une fenêtre gaussienne de largeur caractéristique δ . La figure 2.7 montre la δ -approximation de Moreau de la densité de la loi démocratique $\pi = \mathcal{D}_N(1)$ pour $N = 1, 2$ et plusieurs valeurs de δ . On observe que π_δ a les mêmes modes que π , et se rapproche de π lorsque δ diminue.

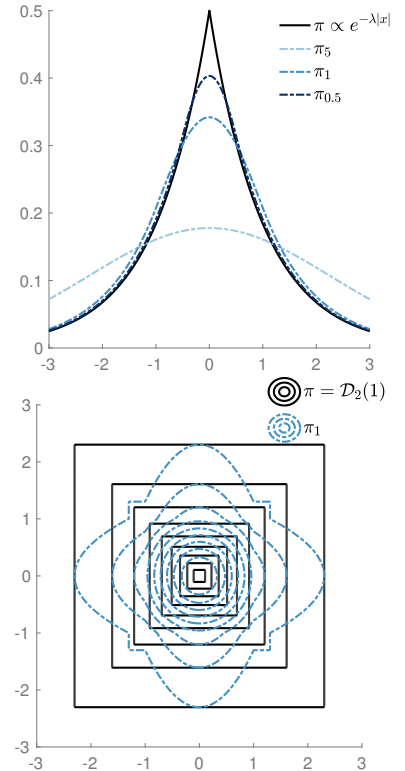


FIGURE 2.7 : Haut : approximations de Moreau de $\pi = \mathcal{D}_1(1)$ pour plusieurs valeurs de δ . Bas : Lignes de niveaux de $\pi = \mathcal{D}_2(1)$ (noir) et de son approximation de Moreau pour $\delta = 1$.

En ce sens, π_δ ressemble à π d'un point de vue géométrique mais gagne en régularité. De plus, si π est de la forme $\pi \propto \exp(g(x))$, π_δ s'exprime facilement à l'aide de l'opérateur proximal, défini en (2.21) :

$$\forall \mathbf{x} \in \mathbb{R}^N, \quad \pi_\delta(\mathbf{x}) \propto \exp\left(g\left(\text{prox}_{\delta g}(\mathbf{x})\right) - \left\|\text{prox}_{\delta g}(\mathbf{x}) - \mathbf{x}\right\|_2^2\right). \quad (2.26)$$

La distribution π_δ est pour ces raisons le bon candidat pour approcher π dans l'équation différentielle stochastique (2.24).

L'implémentation de P-MALA utilise la discrétisation d'Euler de l'équation (2.24) pour générer une chaîne de Markov $(\mathbf{x}^{(t)})_{t \in \mathbb{N}}$. Le biais potentiellement induit par la discrétisation est corrigé par une étape de Metropolis-Hasting. En pratique, à chaque itération t , on tire d'abord un candidat \mathbf{x}^* suivant

$$\mathbf{x}^* | \mathbf{x}^{(t-1)} \sim \mathcal{N}\left(\text{prox}_{\delta \|\cdot\|_\infty / 2}(\mathbf{x}^{(t-1)}), \delta \mathbf{I}_N\right). \quad (2.27)$$

Ce candidat est ensuite accepté comme état $\mathbf{x}^{(t)}$ avec probabilité

$$\alpha = \min\left(1, \frac{p(\mathbf{x}^* | \lambda)}{p(\mathbf{x}^{(t-1)} | \lambda)} \frac{q(\mathbf{x}^{(t-1)} | \mathbf{x}^*; \delta)}{q(\mathbf{x}^* | \mathbf{x}^{(t-1)}; \delta)}\right) \quad (2.28)$$

où $q(\mathbf{x}^* | \mathbf{x}^{(t-1)}, \delta)$ est la loi normale définie en (2.27). Comme recommandé dans Pereyra, 2016a, le paramètre algorithmique δ est préalablement ajusté afin d'atteindre un taux d'acceptation entre 0.4 et 0.6. La méthode est décrite par l'algorithme 2.3.

Algorithme 2.3 : Échantillonneur P-MALA pour la loi *démocratique*.

Entrées : Paramètres λ, δ , dimension N , temps de chauffe T_{bi} , nombre d'échantillons T_{MC} , initialisation $\mathbf{x}^{(0)}$

```

1 pour  $t \leftarrow 1$  à  $T_{\text{bi}} + T_{\text{MC}}$  faire
2   Tirer  $\mathbf{x}^* | \mathbf{x}^{(t-1)} \sim \mathcal{N}\left(\text{prox}_{\delta/2 \|\cdot\|_\infty}(\mathbf{x}^{(t-1)}), \delta \mathbf{I}_N\right)$ ;
3   Calculer  $\alpha$  selon (2.28);
4   Tirer  $w \sim \mathcal{U}(0, 1)$ ;
5   si  $w < \alpha$  alors
6     |  $\mathbf{x}^{(t)} = \mathbf{x}^*$ ;
7   sinon
8     |  $\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)}$ ;
9   fin
10 fin
    
```

Output : $\{\mathbf{x}^{(t)}, t = T_{\text{bi}} + 1, \dots, T_{\text{MC}}\} \sim \mathcal{D}_N(\lambda)$

Notons que puisque l'on corrige l'approximation (2.25) par une étape d'acceptation-rejet, les échantillons $(\mathbf{x}^{(t)})$ sont asymptotiquement distribués suivant la loi démocratique $\mathcal{D}_N(\lambda)$. Il n'y a pas de biais. Cette étape d'acceptation-rejet peut aussi être omise, l'algorithme est alors appelé

proximal unadjusted Langevin algorithm (P-ULA). On y gagne en vitesse d'exécution mais les échantillons ne sont plus asymptotiquement distribués suivant $\mathcal{D}_N(\lambda)$. Comme aucune mesure quantitative de l'erreur n'a été réalisée, nous n'utiliserons pas P-ULA ici.

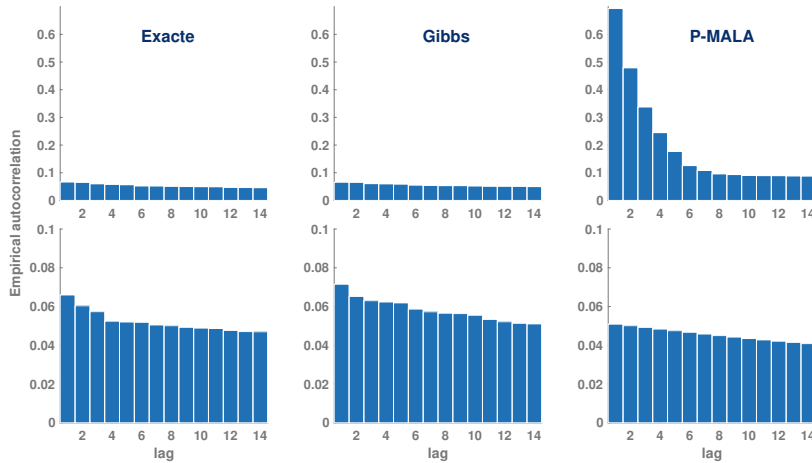
Concernant la complexité numérique, une itération de P-MALA nécessite

- le calcul de l'opérateur proximal, empiriquement estimé en $\mathcal{O}(N)$, mais pouvant être en $\mathcal{O}(N^2)$ dans le pire des cas (Condat, 2015),
- une réalisation d'une loi normale multivariée de matrice de covariance diagonale, soit N réalisations d'une lois normale centrée réduite,
- une réalisation d'une loi uniforme pour l'acceptation.

2.1.3 Simulations numériques

On utilise deux métriques pour comparer les trois échantillonneurs proposés dans les parties 2.1.2.a à 2.1.2.c. La première métrique est la fonction d'autocorrélation empirique, qui mesure la capacité de l'algorithme à mélanger la chaîne composée des échantillons successifs. La deuxième métrique est le temps d'exécution pour une implémentation MATLAB.

La figure 2.8 compare les 14 premiers temps de la fonction d'autocorrélation empirique des distributions $\mathcal{D}_2(3)$ et $\mathcal{D}_{50}(3)$, calculée sur 500 échantillons. Dans les deux configurations, les échantillonneurs exact et de Gibbs ont des performances comparables. En revanche, les performances de P-MALA, mauvaises comparées aux deux autres échantillonneurs en dimension 2, augmentent significativement avec la dimension.



Un estimateur de la fonction d'autocorrélation en dimension 1 est donné par

$$R(j) = \frac{1}{\|x\|_2^2} \sum_n (x_n - \bar{x})(x_{n-j} - \bar{x})$$

où \bar{x} est la moyenne empirique.

FIGURE 2.8 : 14 premiers temps de l'autocorrélation empirique des chaînes respectivement générées par l'échantillonneur exact (colonne de gauche), de Gibbs (colonne du milieu) et P-MALA (colonne de droite), pour $\lambda = 3$, et $N = 2$ (haut) et $N = 50$ (bas).

Le tableau 2.1 ci-dessous montre les temps d'exécution des deux échantillonneurs, pour générer 1000 échantillons de la distribution $\mathcal{D}_N(3)$ pour différentes valeurs de N . Tous les temps sont renormalisés par le temps nécessaire pour générer autant d'échantillons avec une loi normale multivariée. On retiendra que la durée de l'échantillonneur exact est du même ordre de grandeur que celui de la loi normale, que la durée de l'échantillonneur de Gibbs augmente fortement avec la dimension, tandis que les

performances de P-MALA se rapprochent de celle de l'échantillonneur exact avec la dimension.

N	Temps d'exécution renormalisé		
	exact	Gibbs	P-MALA
2	2.1×10^0	9.3×10^2	5.0×10^0
5	3.9×10^0	3.9×10^3	6.5×10^2
10	2.2×10^0	3.5×10^3	3.9×10^2
20	1.8×10^0	4.6×10^3	2.6×10^2
50	1.6×10^0	5.7×10^3	1.3×10^2
100	1.5×10^0	6.0×10^3	6.8×10^1

Sans surprise, l'échantillonneur exact s'avère plus intéressant. Ce résultat était attendu pour des raisons algorithmiques et logicielles. D'abord, l'échantillonneur exact ne nécessite que N réalisations d'une variable aléatoire contre $3N$ pour l'échantillonneur de Gibbs, voir tableau 2.2 pour un résumé des complexités numériques. Ensuite, l'échantillonneur exact n'utilise pas de boucle for MATLAB tandis que l'échantillonneur de Gibbs en utilise deux.

Bien que moins performants, l'échantillonneur de Gibbs et P-MALA ne sont pas inintéressants. Nous verrons dans la prochaine partie que ces deux algorithmes pourront être facilement modifiés pour échantillonner suivant la loi *a posteriori* d'un modèle bayésien résultant d'une loi *a priori* démocratique et ainsi approcher des estimateurs.

2.2 MODÈLE BAYÉSIEN

CETTE SECTION décrit le modèle bayésien BAC-1 proposé pour le codage antiparcimonieux. La partie 2.2.1 décrit le modèle, les méthodes d'inférences et les estimateurs sont respectivement décrits dans les parties 2.2.2 et 2.2.3.

2.2.1 Modèle bayésien hiérarchique

2.2.1.a Fonction de vraisemblance gaussienne

Soit \mathbf{y} un vecteur d'observation de \mathbb{R}^M , et \mathbf{H} une matrice de codage de dimensions $M \times N$ connue. On souhaite reconstruire ce vecteur d'observation à travers le modèle d'analyse linéaire $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}$ où $\mathbf{e} = [e_1, \dots, e_N]^T$ est le vecteur des erreurs résiduelles, supposé gaussien centré et de matrice de covariance $\sigma^2 \mathbb{I}_M$. Ce modèle d'erreur conduit à la fonction de vraisemblance

$$f(\mathbf{y}|\mathbf{x}, \sigma^2) = \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{M}{2}} \exp \left[-\frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \right]. \quad (2.29)$$

TABLE 2.1 : Temps requis pour la génération de 1000 échantillons de la loi démocratique $\mathcal{D}_N(3)$ pour différentes valeurs de N . Les temps sont renormalisés par rapport à la durée nécessaire pour générer 1000 échantillons d'une loi normale en dimension N .

TABLE 2.2 : Comparaisons des complexités numériques des échantillonneurs exacts, de Gibbs et P-MALA pour générer un échantillon suivant la loi démocratique. Les notations $\mathcal{G}, \mathcal{U}, \mathcal{N}$ et \mathcal{E} désignent respectivement les lois Gamma, Uniforme, Normale et Exponentielle.

Algorithme	Complexité
Exact	$1 \mathcal{G} + (N-1) \mathcal{U}$
Gibbs	$N \mathcal{B} + N (\mathcal{U} \text{ ou } \mathcal{E}) + N \mathcal{U}$
P-MALA	$\text{prox} \sim \mathcal{O}(N) + N \mathcal{N} + \mathcal{U}$

2.2.1.b Loi démocratique comme a priori pour le code

L'objectif de ce travail est de fournir une formulation bayésienne du codage antiparcimonieux. Pour interpréter le problème variationnel (\mathbf{V}_∞^β) comme l'estimateur MAP d'un modèle bayésien, on choisit la loi démocratique introduite dans la partie 2.1 comme loi *a priori* sur le vecteur de représentation \mathbf{x} . On espère ainsi favoriser l'antiparcimonie, *i.e.*, l'utilisation la plus uniforme possible de toutes les composantes

$$\mathbf{x} \mid \lambda \sim \mathcal{D}_N(\lambda). \quad (2.30)$$

On rappelle la définition de (\mathbf{V}_∞^β)

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z} \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2 + \beta \|\mathbf{z}\|_\infty$$

2.2.1.c Hyperparamètres du modèle BAC-1

La fonction de vraisemblance donnée en (2.29) et la loi *a priori* démocratique sur les coefficients \mathbf{x} sont paramétrées par les deux réels positifs σ^2 et λ . Dans la suite, on préférera considérer la reparamétrisation $\lambda = \frac{N+1}{2}\mu$. On a alors d'après l'équation (2.7), $\mathbb{E}[x_n] = \mu^{-1}$, qui est indépendant de la dimension. Notons $\phi_1 \triangleq \{\sigma^2, \mu\}$ l'ensemble formé par ces deux paramètres.

Les deux hyperparamètres σ^2 et μ sont liés au paramètre β dans le problème variationnel (\mathbf{V}_∞^β) à travers le produit

$$\beta = \sigma^2(N+1)\mu. \quad (2.31)$$

Ils n'ont pas d'interprétation physique mais n'interviennent que dans la modélisation du problème. On parle de *paramètres de nuisance*.

Nous avons étudié au chapitre 1 deux algorithmes pour résoudre (\mathbf{V}_∞^β), à savoir FITRA et PFA. Ces deux algorithmes nécessitent la connaissance du paramètre β , contrôlant le compromis entre la reconstruction et l'antiparcimonie. En ce sens, ce sont des algorithmes supervisés.

Un des avantages de la modélisation bayésienne est la possibilité de ne pas régler à l'avance la valeur de ce paramètre β , ou de manière équivalente, les paramètres σ^2 et μ pour le modèle BAC-1. Pour ne pas avoir à régler de paramètres, on choisit également des lois *a priori* sur σ^2 et μ .

On choisit deux lois conjuguées Gamma et Inverse Gamma comme loi *a priori* sur ϕ_1

$$\pi(\mu; a_\mu, b_\mu) \sim \mathcal{G}(a_\mu, a_\mu) \quad (2.32)$$

$$\pi(\sigma^2; a_{\sigma^2}, b_{\sigma^2}) \sim \mathcal{IG}(a_{\sigma^2}, b_{\sigma^2}), \quad (2.33)$$

où les hyperparamètres $a_\mu, b_\mu, a_{\sigma^2}$ et b_{σ^2} sont choisis de sorte à rendre la loi vague, *i.e.*, $a_\mu = b_\mu = a_{\sigma^2} = b_{\sigma^2} = 10^{-6}$. On parle d'*a priori* vague car prendre 0 comme valeur reviendrait à choisir un *a priori* de Jeffreys. Ici, on préfère choisir des valeurs proches de 0 mais non nulles et continuer à manipuler des lois de probabilités.

On aboutit à la loi *a posteriori* jointe

$$\begin{aligned}
 p(\mathbf{x}, \phi_1 | \mathbf{y}) &= (2\pi\sigma^2)^{-M/2} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2\right) \times \frac{\lambda^N}{2^N N!} \exp(-\lambda \|\mathbf{x}\|_\infty) \\
 &\times \frac{b_\mu^{a_\mu}}{\Gamma(a_\mu)} \mu^{a_\mu} \exp(-b_\mu \mu) \\
 &\times \frac{b_{\sigma^2}^{a_{\sigma^2}}}{\Gamma(a_{\sigma^2})} \left(\frac{1}{\sigma^2}\right)^{a_{\sigma^2}} \exp\left(-\frac{b_{\sigma^2}}{\sigma^2}\right).
 \end{aligned} \tag{2.34}$$

2.2.2 Algorithmes MCMC : échantillonneurs de Gibbs

La loi jointe *a posteriori* (2.34) est utilisée pour calculer des estimateurs bayésiens des codes $\hat{\mathbf{x}}$. Comme ces estimateurs n'ont pas d'expression explicite, on utilise les méthodes MCMC pour les approcher. Nous définissons ces estimateurs dans la prochaine partie (cf. 2.2.3). On rappelle que ces méthodes génèrent une collection d'échantillons $\{\mathbf{x}^{(t)}, \sigma^{2(t)}, \mu^{(t)}\}$ asymptotiquement distribués suivant la jointe *a posteriori* (2.34).

Cette partie présente trois algorithmes MCMC pour générer des échantillons suivant (2.34). Il s'agit de trois variantes d'un échantillonneur de Gibbs où l'on échantillonne tour à tour et dans un ordre aléatoire les paramètres $\mathbf{x}, \sigma^2, \mu$. Ces trois algorithmes se distinguent par la méthode utilisée pour échantillonner les coefficients \mathbf{x} . L'algorithme 2.4 présente l'algorithme de Gibbs générique, dont les étapes seront détaillées ci-après.

Algorithme 2.4 : Échantillonneur de Gibbs pour le modèle BAC-1

Entrées : Vecteur d'observation \mathbf{y} , matrice d'encodage \mathbf{H} , hyperparamètres $(a_\mu, b_\mu), (a_{\sigma^2}, b_{\sigma^2})$, nombre d'itérations de chauffe T_{bi} , nombre d'itérations T_{MC} ,

```

1 pour  $t \leftarrow 1$  à  $T_{\text{MC}}$  faire
2   échantillonner les hyperparamètres
3    $\sigma^{2(t)}$  et  $\mu^{(t)}$  selon (2.35) et (2.36) ;
4   échantillonner  $\mathbf{x}^{(t)}$  selon
5     BAC Gibbs (BAC Gibbs), voir partie 2.2.2.a ;
   ou
6     BAC P-MALA (BAC P-MALA), voir partie 2.2.2.b ;
   ou
7     BAC P-MYULA (BAC P-MYULA), voir partie 2.2.2.c ;
8 fin
    
```

Output : Collection d'échantillons $\{\mu^{(t)}, \sigma^{2(t)}, \mathbf{x}^{(t)}, t = T_{\text{bi}} + 1 \dots T_{\text{MC}}\}$ asymptotiquement distribués suivant (2.34).

Par conjugaison, les lois *a posteriori* des paramètres de nuisance σ^2 et μ sont

$$\sigma^2 | \mathbf{y}, \mathbf{H}, \mathbf{x} \sim \mathcal{IG}\left(a_{\sigma^2} + \frac{M}{2}, b_{\sigma^2} + \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2\right) \tag{2.35}$$

$$\mu | \mathbf{y}, \mathbf{H}, \mathbf{x} \sim \mathcal{G}\left(a_\mu + N, b_\mu + N \|\mathbf{x}\|_\infty\right). \tag{2.36}$$

Les prochaines sous-parties décrivent les méthodes d'échantillonnage des codes \mathbf{x} . Le premier algorithme, décrit partie 2.2.2.a procède composante par composante et est dénommé **BAC Gibbs**. Le second algorithme, **BAC P-MALA**, décrit dans la partie 2.2.2.b est une variante de P-MALA déjà utilisée pour générer des échantillons suivant la loi démocratique dans la partie 2.1.2.c. La dernière méthode, **BAC P-MYULA**, décrite dans la partie 2.2.2.c est une alternative à P-MALA exploitant la forme de la loi *a posteriori* pour construire une loi de proposition plus efficace.

2.2.2.a Échantillonnage composantes par composantes : BAC Gibbs

BAC Gibbs capitalise sur les lois conditionnelles de la loi démocratique établies dans la propriété 2.5 pour simuler la loi *a posteriori* $p(\mathbf{x}|\mathbf{y}, \phi_1)$ composante par composante. En particulier, la loi $p(x_n|\mathbf{y}, \mathbf{x}_{\setminus n}, \mu)$ d'une composante sachant les autres est un mélange de lois normales tronquées

$$x_n|\mathbf{x}_{\setminus n}, \mu, \sigma^2, \mathbf{y} \sim \sum_{\ell=1}^3 a_{n\ell} \mathcal{N}_{\mathcal{I}_{n\ell}}(m_{n\ell}, s_n^2) \quad (2.37)$$

où $\mathcal{N}_{\mathcal{I}}(\cdot, \cdot)$ est la loi normale tronquée à l'intervalle \mathcal{I} , les intervalles $(\mathcal{I}_i)_{i \in \{1,2,3\}}$ sont définis par

$$\begin{aligned} \mathcal{I}_{n1} &= (-\infty, -\|\mathbf{x}_{\setminus n}\|_{\infty}] \\ \mathcal{I}_{n2} &= (-\|\mathbf{x}_{\setminus n}\|_{\infty}, \|\mathbf{x}_{\setminus n}\|_{\infty}) \\ \mathcal{I}_{n3} &= [\|\mathbf{x}_{\setminus n}\|_{\infty}, +\infty). \end{aligned} \quad (2.38)$$

Les moyennes $(m_{\ell n})_{\ell \in \{1,2,3\}}$, les variances $(s_n)_{\ell \in \{1,2,3\}}$ et les poids du mélange $(a_{\ell n})_{\ell \in \{1,2,3\}}$ sont donnés en annexe A.5. On retiendra que chaque moyenne $m_{\ell,n}$ s'interprète comme un gradient pondéré par la valeur de la composante dominante. La forme de la loi *a posteriori* d'une composante est illustrée à la figure 2.9. On y remarque qu'en raison du mélange de lois normales tronquées, la loi *a posteriori* est discontinue.

La loi *a posteriori* conditionnelle d'une composante est à nouveau un mélange à trois composantes. Pour rappel, la propriété 2.5 affirme que la conditionnelle d'une composante est un mélange d'une loi uniforme et de deux lois exponentielles translatées. C'est la combinaison de la vraisemblance gaussienne et des lois exponentielles et uniforme qui produit les lois normales tronquées. L'échantillonnage des lois normales tronquées est réalisé en suivant la méthode développée dans Chopin (2010), et disponible en ligne.

Du point de vue de la complexité numérique, chaque itération nécessite

- $3N$ calculs de gradients, en $\mathcal{O}(MN^2)$,
- N réalisations d'une loi multinomiale,
- N réalisations d'une loi normale tronquée,
- *facultatif* : une réalisation d'une loi uniforme sur les permutations de $\llbracket 1, N \rrbracket$ (Algorithme P, cf. partie 2.1.2.b).

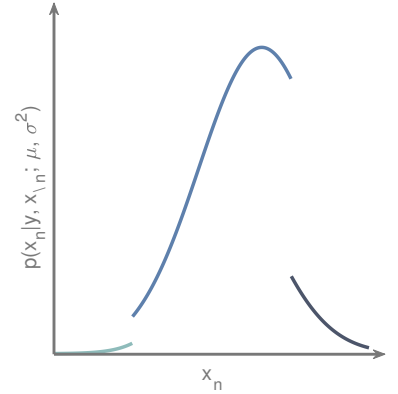


FIGURE 2.9 : Forme de la loi *a posteriori* $x_n|\mathbf{y}, \mathbf{x}_{\setminus n}, \sigma^2, \lambda$ d'une composante du vecteur de coefficient. Les trois couleurs sont respectivement associés aux intervalles \mathcal{I}_{n1} , \mathcal{I}_{n2} et \mathcal{I}_{n3} .

2.2.2.b Algorithme BAC P-MALA

On a vu que la loi a posteriori $p(\mathbf{x}|\mathbf{y}, \sigma^2, \mu)$ n'est pas différentiable mais seulement continue en raison de la norme ℓ_∞ . L'algorithme P-MALA utilisé dans la partie 2.1.2.c pour générer des échantillons suivant la loi *démocratique* peut être adapté pour échantillonner suivant $p(\mathbf{x}|\mathbf{y}, \sigma^2, \mu)$.

Pour rappel, l'algorithme P-MALA s'appuie sur un processus de diffusion de Langevin dont la loi cible est l'approximation de Moreau d'une densité de probabilité. Cette approximation de Moreau s'exprime simplement à l'aide de l'opérateur proximal de la neg-log-densité, et est ensuite corrigée par une étape d'acceptation-rejet. Ces arguments en font un outil adapté aux densités non différentiables.

Notons g la neg-log-distribution de la loi *a posteriori* $p(\mathbf{x}|\mathbf{y}, \sigma^2, \mu)$

$$\forall \mathbf{x} \in \mathbb{R}^N, \quad g(\mathbf{x}, \mu, \sigma^2) \triangleq \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \frac{N+1}{2} \mu \|\mathbf{x}\|_\infty. \quad (2.39)$$

Remarquons que $g(\mathbf{x}, \mu, \sigma^2) = J_{(N+1)\mu\sigma^2}^\infty(\mathbf{x})$, où $J_\beta^\infty(\cdot)$ a été introduit équation (2.1).

L'opérateur proximal de paramètre $\delta > 0$ de la fonction g et évalué en $\mathbf{x} \in \mathbb{R}^N$ est

$$\text{prox}_{\delta g(\cdot)}(\mathbf{x}) = \arg \max_{\mathbf{u} \in \mathbb{R}^N} \delta g(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{u}\|_2^2. \quad (2.40)$$

L'opérateur proximal (2.40) n'a pas d'expression explicite en raison de la présence de l'opérateur \mathbf{H} . Ces opérateurs sont généralement approchés par des méthodes d'optimisation convexes (Komodakis et Pesquet, 2015). Lorsque la fonction g pour laquelle on cherche à évaluer l'opérateur proximal peut se décomposer suivant

$$g = g_0 + g_1, \quad (2.41)$$

où g_0 et g_1 sont respectivement continue et différentiable, i.e $g_0 \in \mathcal{C}^0$ et $g_1 \in \mathcal{C}^1$, l'opérateur proximal peut être évalué efficacement en exploitant la différentiabilité de g_1 via l'approximation du premier ordre

$$\text{prox}_{\delta g}(\mathbf{x}) \simeq \text{prox}_{\delta g_0}(\mathbf{x} + 2\delta \nabla g_1(\mathbf{x})). \quad (2.42)$$

L'opérateur proximal $\text{prox}_{\delta g_0}$ à calculer correspond désormais à l'opérateur proximal de la norme ℓ_∞ déjà calculé dans la partie 2.1.1.f. Notons que cette approximation du premier ordre est la même que celle utilisée dans un algorithme de type *forward-backward*, dont deux implémentations ont été présentées au chapitre 1, à savoir *ISTA* et *FITRA* cf. algorithmes 1.1 et 1.4 pages 30 et 40.

En pratique, à chaque nouvelle itération t de BAC P-MALA, la génération d'un nouvel échantillon se fait en deux étapes. D'abord, un nouveau candidat \mathbf{x}^* est proposé suivant

$$\mathbf{x}^* | \mathbf{x}^{(t-1)} \sim \mathcal{N}\left(\text{prox}_{\frac{\delta}{2} g}(\mathbf{x}^{(t-1)}), \delta \mathbb{I}_N\right). \quad (2.43)$$

Ici, $g_0(\mathbf{x}) \triangleq \frac{N+1}{2} \mu \|\mathbf{x}\|_\infty$ et $g_1(\mathbf{x}) \triangleq \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2$

Ce nouveau candidat \mathbf{x}^* est ensuite accepté avec probabilité α , où

$$\alpha = \min \left(1, \frac{f(\mathbf{x}^* | \mu, \sigma^2, \mathbf{y})}{f(\mathbf{x}^{(t-1)} | \mu, \sigma^2, \mathbf{y})} \frac{q(\mathbf{x}^{(t-1)} | \mathbf{x}^*)}{q(\mathbf{x}^* | \mathbf{x}^{(t-1)})} \right), \quad (2.44)$$

où q est la densité de la loi normale multivariée de matrice de covariance $\delta \mathbb{I}_N$. Ici, δ est toujours dynamiquement réglé afin d'obtenir un taux d'acceptation compris entre 0.4 et 0.6 (Pereyra, 2016a).

BAC P-MALA est comparable à une version stochastique de FITRA. Le calcul de (2.42) correspond à une descente de gradient pour améliorer la reconstruction suivie d'un second mouvement suivant l'opérateur proximal de la neg-log-prior pour diminuer la valeur de la norme ℓ_∞ . L'étape (2.43) est un pas d'une marche aléatoire autour de l'opérateur proximal.

Notons que l'approximation équation (2.42) n'induit pas de biais sur la convergence vers la loi du processus de diffusion de Langevin⁵ car cette approximation est corrigée par l'étape d'acceptation-rejet.

Du point de vue de la complexité numérique, chaque itération nécessite

- le calcul du gradient de g_1 définie Éq. (2.42), en $\mathcal{O}(MN^2)$,
- le calcul de l'opérateur proximal Éq. (2.40), empiriquement évalué à $\mathcal{O}(N)$ par Condat (2015),
- N réalisations d'une loi normale,
- une réalisation d'une Bernoulli pour l'acceptation-rejet.

5. On rappelle qu'un processus de diffusion de Langevin est une équation différentielle stochastique, où, dans notre cas, la loi *a posteriori* (2.34) est une solution, cf. partie 2.1.2.c du même chapitre.

2.2.2.c Algorithme BAC P-MYULA

Depuis la publication de nos travaux, une version améliorée de l'algorithme P-MALA a été développée (Durmus et al., 2016). Nous la présentons dans cette partie.

On rappelle que dans P-MALA, l'opérateur proximal de la neg-log-vraisemblance avait été approché à l'aide d'une approximation du premier ordre (cf. équation (2.42)). En l'état, cette approximation n'a pas d'influence sur la distribution cible grâce à la correction due à l'étape d'acceptation-rejet. Toutefois, cette approximation affecte la capacité de mélange de la chaîne. Comme l'erreur d'approximation n'a pas été quantifiée, nous avons exclu l'utilisation de la version non corrigée.

Nous allons voir comment Durmus et al. (2016) proposent une amélioration de l'algorithme en exploitant la décomposition de la neg-log-densité en une partie différentiable et une partie continue. L'algorithme générique proposé est appelée *Moreau-Yosida Unadjusted Langevin Algorithm* (P-MYULA).

On rappelle que la neg-log-posterior du vecteur de représentation \mathbf{x} a été décomposée dans la partie 2.2.2.b selon

$$\forall \mathbf{x} \in \mathbb{R}^N, \quad g(\mathbf{x}, \mu, \sigma^2) = \underbrace{\frac{N+1}{2} \mu \|\mathbf{x}\|_\infty}_{g_0} + \underbrace{\frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2}_{g_1}. \quad (2.45)$$

Plutôt que d'approcher l'opérateur proximal de g comme à l'équation (2.42), P-MYULA propose de directement tenir compte du fait que g_1 est différentiable dans la loi cible. Dans P-MALA, l'approximation π_1^δ de la loi cible dans la diffusion de Langevin est

$$\pi_1^\delta(\mathbf{x}) \propto \sup_{\mathbf{u} \in \mathbb{R}^N} \exp \left(g(\mathbf{u}) - \frac{1}{2\delta} \|\mathbf{u} - \mathbf{x}\|_2^2 \right). \quad (2.46)$$

P-MYULA cible plutôt la distribution

$$\pi_2^\delta(\mathbf{x}) \propto \sup_{\mathbf{u} \in \mathbb{R}^N} \exp \left(g_1(\mathbf{x}) + g_0(\mathbf{u}) - \frac{1}{2\delta} \|\mathbf{u} - \mathbf{x}\|_2^2 \right). \quad (2.47)$$

En pratique, on effectue d'abord un pas de type descente de gradient

$$\bar{\mathbf{x}}^{(t-1)} = \left(1 - \frac{\gamma}{\nu} \right) \mathbf{x}^{(t-1)} - \gamma \nabla f \left(\mathbf{x}^{(t-1)} \right) + \frac{\gamma}{\nu} \text{prox}_h^\nu \left(\mathbf{x}^{(t-1)} \right)$$

suivi d'un pas de marche aléatoire

$$\mathbf{x}^* | \mathbf{x}^{(t-1)} \sim \mathcal{N} \left(\bar{\mathbf{x}}^{(t-1)}, 2\gamma \right). \quad (2.48)$$

En choisissant $\nu = L_{g_1}^{-1}$ où L_{g_1} est la constante de Lipschitz de g_1 et $\gamma = \nu/4$, l'erreur relative entre la distribution cible et la vraie loi *a posteriori* est estimée à 1% par Durmus et al. (2016). Cette erreur pourrait aussi être corrigée par une étape de Metropolis-Hastings, mais nous choisissons de ne pas le faire.

Notons que corriger le biais induit par la discrétisation augmente significativement la variance d'estimation (Durmus et al., 2016). À la vue des bonnes performances numériques de P-MYULA, nous ne nous attarderons pas sur ce point.

2.2.3 Estimateurs

On rappelle que les échantillons $\{\mathbf{x}^{(t)}, \sigma^{2(t)}, \mu^{(t)}\}$ sont utilisés pour approcher des estimateurs bayésiens. Cette section présente deux estimateurs bayésiens des codes antiparcimonieux \mathbf{x} .

On sait qu'après une période de chauffe T_{bi} (pour *burn-in* en anglais), l'ensemble des échantillons

$$\mathcal{X} \triangleq \left\{ \mathbf{x}^{(t)}, t = T_{\text{bi}} + 1, \dots, T_{\text{MC}} \right\} \quad (2.49)$$

est asymptotiquement distribué suivant la loi *a posteriori* (2.34).

2.2.3.a Estimateur MAP à paramètres fixés

Lorsque les paramètres de nuisance σ^2 et μ sont fixés, on construit un estimateur MAP défini par

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x} \in \mathbb{R}^N} p(\mathbf{x} | \mathbf{y}, \sigma^2, \mu) \simeq \arg \max_{\mathbf{x} \in \mathcal{X}} p(\mathbf{x} | \mathbf{y}, \sigma^2, \mu). \quad (2.50)$$

On rappelle que le choix de la loi *a priori* démocratique sur les codes \mathbf{x} fait que cet estimateur MAP coïncide avec la solution du problème variationnel (V_∞^β) pour $\beta = (N + 1)\mu\sigma^2$. Nous étudierons expérimentalement ce point dans la partie 2.3.4.

2.2.3.b Estimateurs MMSE

L'estimateur du *minimum de l'erreur quadratique moyenne* (MMSE) minimise l'erreur quadratique moyenne $\mathbb{E}[(\mathbf{x} - \hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}})]$, où \mathbf{x} désigne la valeur du paramètre aléatoire. On peut montrer que cet estimateur se calcule et s'approxime suivant

$$\hat{\mathbf{x}} = \mathbb{E}[\mathbf{x}|\mathbf{y}] \simeq \frac{1}{T_{\text{MC}} - T_{\text{bi}}} \sum_{t=T_{\text{bi}}+1}^{T_{\text{MC}}} \mathbf{x}^{(t)}. \quad (2.51)$$

Bien que classique dans le paradigme bayésien, l'estimateur MMSE n'est peut-être pas pertinent pour la recherche de codes antiparcimonieux. En effet, le MMSE est par construction l'estimateur le plus performant pour reconstruire des codes liés à un modèle de synthèse. Or, on ne cherche ici à reconstruire un paramètre, mais à trouver un code \mathbf{x} qui reconstruit bien l'observation tout en ayant de bonnes propriétés. En sus, on a vu que la contrainte de minimisation de la norme ℓ_∞ dans le problème (P_∞^ε) n'est pas suffisante pour garantir l'unicité de la solution. Il est fort probable que cette absence d'unicité se transmette au problème variationnel (V_∞^β).

Nous définissons aussi un estimateur MMSE des paramètres de nuisance

$$\hat{\sigma}_{\text{MMSE}}^2 = \frac{1}{T_{\text{MC}} - T_{\text{bi}}} \sum_{t=T_{\text{bi}}+1}^{T_{\text{MC}}} \sigma^2(t) \quad (2.52)$$

$$\hat{\mu}_{\text{MMSE}} = \frac{1}{T_{\text{MC}} - T_{\text{bi}}} \sum_{t=T_{\text{bi}}+1}^{T_{\text{MC}}} \mu^{(t)}. \quad (2.53)$$

2.2.3.c Estimateur MAPm

Lorsque les paramètres σ^2 et μ sont conjointement inférés avec le code \mathbf{x} , il est toujours possible de définir un estimateur MAP joint

$$(\hat{\mathbf{x}}, \hat{\sigma}^2, \hat{\mu}) = \arg \max_{\mathbf{x}, \sigma^2, \mu} p(\mathbf{x}, \sigma^2, \mu | \mathbf{y}).$$

Cependant, les paramètres σ^2 et μ n'ont qu'un intérêt algorithmique. Pour cette raison, on propose un estimateur du *maximum a posteriori marginalisé* (MAPm) qui tient compte du comportement globale des paramètres. Celui-ci s'obtient en intégrant la loi jointe *a posteriori* vis-à-vis des deux paramètres de nuisance que sont σ^2 et μ

$$\begin{aligned} \hat{\mathbf{x}}_{\text{MAPm}} &= \arg \max_{\hat{\mathbf{x}} \in \mathbb{R}^N} f(\mathbf{x}|\mathbf{y}) \\ &\simeq \arg \max_{\hat{\mathbf{x}} \in \mathcal{X}} f(\mathbf{x}|\mathbf{y}), \end{aligned} \quad (2.54)$$

où

$$\begin{aligned} f(\mathbf{x}|\mathbf{y}) &= \int_{\mathbb{R}_+^2} f(\mathbf{u}, \sigma^2, \mu) d\sigma^2 d\mu \\ &= (a_{\sigma^2} + \frac{M}{2}) \ln \left(b_{\sigma^2} + \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{u}\|_2^2 \right) + (a_\mu + N) \ln \left(b_\mu + \lambda \|\mathbf{u}\|_\infty \right). \end{aligned} \quad (2.55)$$

cf. Éq. (1.5) et discussion autour.

On rappelle la définition du problème (P_∞^ε)

$$\hat{\mathbf{x}} = \arg \min_z \|\mathbf{z}\|_\infty \quad \text{s.c.} \quad \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2^2 \leq \varepsilon,$$

avec ε un réel positif.

où \mathcal{X} a été défini en (2.49).

2.3 PERFORMANCES À PARAMÈTRES DE NUISANCE FIXÉS

CETTE PARTIE étudie les performances des algorithmes à paramètres σ^2 et μ fixés. Le protocole expérimental et les indicateurs de performances sont définis dans la partie 2.3.1. Nous comparons dans les parties 2.3.2 et 2.3.3 les performances numériques des algorithmes BAC Gibbs, BAC P-MALA et BAC P-MYULA présentés dans la partie 2.2.2. Nous étudions dans la partie 2.3.4 l'influence de la valeur du paramètre μ en comparant les estimateurs MAP et MMSE avec les solutions de FITRA, un algorithme déterministe.

2.3.1 Protocole expérimental

À l'instar de la partie 2.1.3, on étudie dans cette section les performances et les capacités de mélange des algorithmes. On s'appuie sur le modèle génératif pour élaborer un modèle de synthèse.

Soit \mathbf{H} une matrice d'encodage donnée. À σ^2 et λ donnés, on génère d'abord un vecteur de coefficients \mathbf{x}_S (pour synthèse) de \mathbb{R}^N distribué suivant la loi *démocratique* $\mathcal{D}_N(\lambda)$ où λ est arbitrairement fixé à 3. On construit l'observation \mathbf{y} , élément de \mathbb{R}^M suivant

$$\mathbf{y} = \mathbf{H}\mathbf{x}_S + \mathbf{e}, \quad (2.56)$$

où \mathbf{e} est un tirage d'une loi normale multivariée centré de matrice de covariance $\sigma^2 \mathbb{I}_M$. On choisit $\sigma^2 = 0.1$ pour la valeur de la variance de la perturbation, de sorte à obtenir à un $\text{SNR}_{\mathbf{y}}$ de 40 dB pour le premier scénario et 60 dB pour le second. Les matrices d'encodage \mathbf{H} sont générées en sous échantillonnant les lignes d'une matrice DCT quelconque, puisque l'on a vu dans la partie 1.3.3 que de telles matrices garantissent avec forte probabilité l'existence de codes antiparcimonieux.

Puisque l'on veut comparer seulement l'échantillonnage des coefficients \mathbf{x} des trois algorithmes BAC Gibbs, BAC P-MALA et BAC P-MYULA, les paramètres de nuisance σ^2 et μ sont fixés et non échantillonnés. Les codes $\hat{\mathbf{x}}$ sont construits à partir des estimateurs MAP et MMSE (cf. parties 2.2.3.a et 2.2.3.b).

Les trois premières métriques sont un rapport signal sur bruit (abrégé SNR pour *signal-to-noise ratio* en anglais) SNR de reconstruction des coefficients ($\text{SNR}_{\mathbf{x}}$) \mathbf{x}_S , un SNR de reconstruction de l'observation ($\text{SNR}_{\mathbf{y}}$), et une mesure de l'étalement des coefficients $\hat{\mathbf{x}}$. Ces métriques sont définies pour tout vecteur \mathbf{u} de \mathbb{R}^N

$$\text{SNR}_{\mathbf{x}}(\mathbf{u}) = 20 \log_{10} \frac{\|\mathbf{x}_S\|_2}{\|\mathbf{x}_S - \mathbf{u}\|_2}, \quad (2.57)$$

$$\text{SNR}_{\mathbf{y}}(\mathbf{u}) = 20 \log_{10} \frac{\|\mathbf{y}\|_2}{\|\mathbf{y} - \mathbf{H}\mathbf{u}\|_2}, \quad (2.58)$$

$$\text{PAPR}(\mathbf{u}) = \frac{N \|\mathbf{u}\|_{\infty}^2}{\|\mathbf{u}\|_2^2}. \quad (2.59)$$

Le PAPR est réel, élément du segment $[1, N]$. Nous avons déjà évoqué cette quantité au chapitre 1 (cf. partie 1.1.5). Les deux autres métriques

sont la vitesse de décroissance des temps de la fonction d'autocorrélation empirique et le temps d'exécution de l'implémentation mixte MATLAB/C++.

Deux configurations sont testées : $(M, N) = (30, 50)$ et $(M, N) = (30, 100)$. Les estimateurs sont calculés sur 10^4 itérations après un temps de chauffe de 10^3 itérations et moyennés sur 20 simulations.

2.3.2 Performances numériques

Le tableau 2.3 affiche les différents indicateurs de performances.

Méthode	Estimateur	SNR _y (dB)	SNR _x (dB)	PAPR	Temps itération (s)
BAC Gibbs	MMSE	84.4 / 150.9	5.4 / -6.3	7.7 / 4.3	$2.8 \times 10^{-5} / 5.9 \times 10^{-5}$
	MAP	66.3 / 82.1	-3.2 / -12.4	4.8 / 3.6	
P-MALA	MMSE	131.2 / 161.2	-7.8 / -2.2	3.2 / 4.1	$1.1 \times 10^{-5} / 1.4 \times 10^{-5}$
	MAP	63.9 / 83.9	-0.8 / -2.1	2.8 / 3.1	
P-MYULA	MMSE	100.6 / 148.8	0.2 / -0.8	4.1 / 3.8	$1.0 \times 10^{-5} / 1.4 \times 10^{-5}$
	MAP	62.5 / 80.5	-1.1 / -2.3	3.8 / 2.9	

TABLE 2.3 : Performances en termes de SNR_x, SNR_y, PAPR et temps de calculs par itérations pour les algorithmes BAC Gibbs, BAC P-MALA et BAC P-MYULA, pour $N = 50$ et $N = 100$.

Meilleurs SNR_y et PAPR avec la dimension. Les performances de reconstruction de l'observation sont dans l'ensemble supérieures à 50 dB. Pour les trois algorithmes et tous les estimateurs, les indicateurs de SNR_y et PAPR augmentent avec la dimension. Ces observations s'expliquent par les raisons suivantes. À dimension de l'observation M fixée, le nombre de solutions du problème (P_∞^ϵ) augmente lorsque la taille N du code augmente, donc les algorithmes bayésiens identifient plus facilement des solutions performantes. Le théorème 1.6 vu au chapitre 1 montre que le PAPR des solutions de (P_∞^ϵ) diminue lorsque N augmente. Nous étudierons plus en détail ce point dans partie 2.4.4.

Mauvaise reconstruction des codes. Malgré une bonne reconstruction des observations, aucun estimateur n'est parvenu à reconstruire les codes x_S ayant servi à la synthèse puisque les SNR_x sont tous inférieurs à 5 dB voire négatifs. On écarte l'éventualité que toutes les chaînes se soient retrouvées coincées dans des maxima locaux. Ces observations confirment la remarque de la partie 2.2.3.b : de par la multiplicité des solutions du problème (P_∞^ϵ), on ne peut pas garantir la reconstruction d'un code x_S . On rappelle toutefois que l'objectif de ce chapitre est bien *d'identifier un code antiparcomionieux* et non de retrouver un code ayant servi à la synthèse.

Temps de calculs comparables. Les temps de calculs moyens par itération sont comparables, voire identiques pour BAC P-MALA et BAC P-MYULA. Ces résultats étaient prévisibles puisque ces deux méthodes font appels aux mêmes fonctions aléatoires et les trois algorithmes sont dominés par

le calcul d'un gradient en $\mathcal{O}(MN^2)$. En terme d'aléatoire, on rappelle que BAC P-MALA et BAC P-MYULA nécessitent N réalisations d'une loi normale et une réalisation d'un Bernoulli tandis que BAC Gibbs nécessite N réalisations d'une loi normale tronquée et N réalisations d'une loi multinomiale.

2.3.3 Mélange des chaînes de Markov

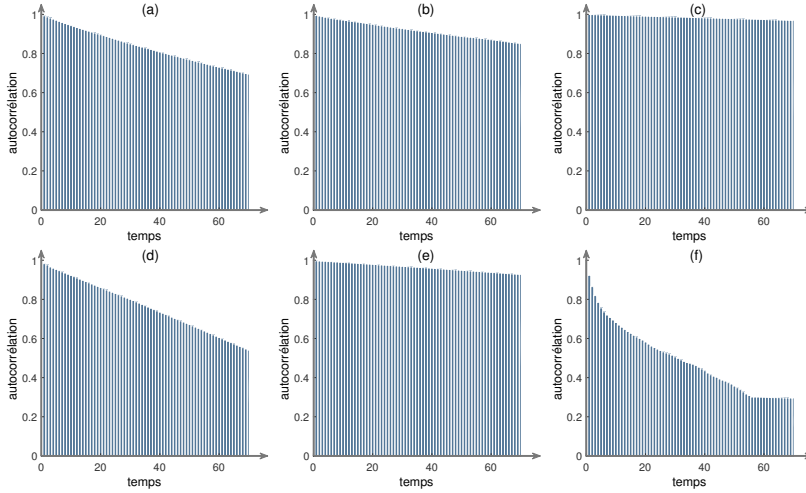


FIGURE 2.10 : 100 premiers temps de la fonction d'autocorrélation empirique pour haut : $N = 50$ et bas : $N = 100$, approchés avec les algorithmes (a & d) BAC Gibbs, (b & e) P-MALA et (c & f) P-MYULA.

La figure 2.10 affiche les 100 premiers temps de l'autocorrélation empirique pour la loi $p(\mathbf{x}|\mathbf{y})$ pour $N = 50$ et $N = 100$ approchés en utilisant les algorithmes (a & d) BAC Gibbs, (b & e) P-MALA et (c & f) P-MYULA. En dimension $N = 50$, l'algorithme BAC Gibbs admet la décroissance la plus rapide de la fonction d'autocorrélation tandis que P-MYULA est la méthode la plus efficace en dimension $N = 100$. Dans les deux cas, P-MALA est la méthode affichant la décroissance la plus lente. Notons que ce défaut était prévisible en raison de l'étape d'acceptation-rejet à chaque itération.

Pour améliorer la capacité de mélange des algorithmes, nous augmentons le nombre d'échantillons générés entre chaque itération : c'est le paramètre de *thinning* en anglais. On utilise pour P-MALA et P-MYULA un *thinning* de 10 échantillons.

2.3.4 Calibration du paramètre μ

Par construction du modèle bayésien, l'estimateur MAP coïncide avec la solution du problème variationnel (V_∞^β) pour $\beta = (N + 1)\mu\sigma^2$ lorsque les paramètres σ^2 et μ sont fixés. Comme seule la valeur du produit $\beta = \sigma^2(N + 1)\mu$ importe, la valeur de l'un des deux paramètres μ ou σ^2 est libre, et l'on peut s'attendre à des différences dues à l'efficacité numérique. Nous allons chercher la valeur optimale de μ à β fixé.

On rappelle la définition de (V_∞^β)

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z} \in \mathbb{R}^N} J_\beta^\infty(\mathbf{z})$$

$$J_\beta^\infty(\mathbf{z}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2 + \beta \|\mathbf{z}\|_\infty.$$

On fixe la valeur du produit $\beta \triangleq (N + 1)\mu\sigma^2$, et l'on fait varier la valeur de μ . On s'appuie toujours sur le protocole expérimental décrit dans la partie 2.3.1, mais nous n'allons décrire les résultats qu'en termes de $\text{SNR}_{\mathbf{y}}$ et de PAPR pour l'algorithme BAC P-MYULA.

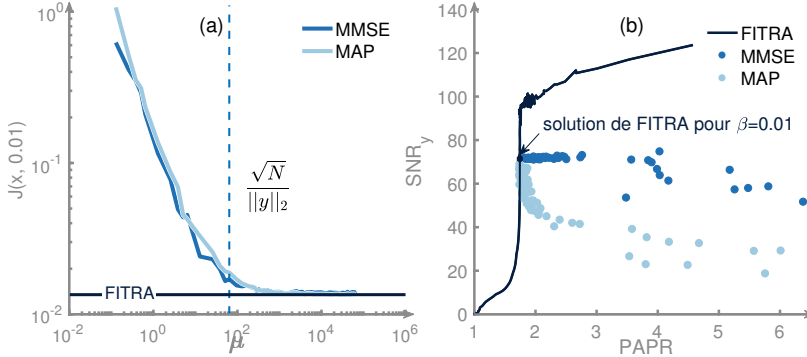


FIGURE 2.11 : Performances des estimateurs MAP et MMSE vis-à-vis de la fonction de coût J_{β}^{∞} , à β fixé en fonction de μ .

La figure 2.11-a affiche la valeur de $J_{0.01}^{\infty}(\hat{\mathbf{x}})$ ⁶ en fonction du paramètre μ sur une réalisation de \mathbf{y} pour les estimateurs MAP et MMSE, *i.e.*, pour $\beta = 0.01$, $M = 30$ et $N = 50$. Les prochains commentaires sont qualitativement indépendants du choix de β , N et de la réalisation. La droite horizontale est la valeur de $J_{0.01}^{\infty}(\hat{\mathbf{x}})$ obtenue avec FITRA, qui joue ici le rôle de solution de référence. On rappelle que FITRA est un algorithme de type *forward-backward*, qui cherche un minimum du problème variationnel ($\mathbf{V}_{\infty}^{\beta}$) et étudié en 1.4.4.

Les performances augmentent avec μ . Comme l'estimateur MAP ne dépend que de β , la courbe de $J_{0.01}^{\infty}$ en fonction de μ devrait être une droite confondue avec la droite associée à FITRA. On constate pourtant que les estimateurs MAP et MMSE sont d'autant plus proches de FITRA que le paramètre μ est grand, ou de manière équivalente lorsque σ^2 est petit. On note l'existence d'une valeur seuil, environ égale à $\sqrt{N}/\|\mathbf{y}\|_2$, à partir de laquelle on peut numériquement considérer que les estimateurs sont les mêmes. *Ce résultat positif* montre qu'en réglant correctement les paramètres du modèle, il est possible d'atteindre les mêmes performances numériques qu'un algorithme déterministe. Néanmoins, ces résultats sont surprenants dans le sens où *suresimer la valeur de μ , même d'un facteur 100 ne semble pas avoir de conséquence sur les performances.*

La figure 2.11-b représente les estimateurs MAP (points bleus clairs) et MMSE (points bleus foncés) dans le plan (PAPR, $\text{SNR}_{\mathbf{y}}$) lorsque μ varie. La courbe noire correspond aux solutions proposées par FITRA pour toutes les valeurs du paramètre de régularisation β . On constate que les chemins des deux estimateurs obtenus en faisant varier μ convergent vers le point correspondant à la solution proposée par FITRA pour $\beta = 0.01$.

Vers des méthodes SVA. Coupler les paramètres de nuisance σ^2 et μ à travers une variable auxiliaire β et faire tendre la variance de l'erreur résiduelle σ^2 vers 0 ressemble aux méthodes dites de *small variance asymptotics* (SVA). Initialement inspirée par l'interprétation de l'algorithme des

6. Voir (2.1) pour la définition de J_{β}^{∞} et la note précédente.

k -moyennes comme l'asymptotique d'un modèle probabiliste (Kulis et Jordan, 2012), les méthodes SVA sont des algorithmes déterministes rapides obtenus à partir d'un modèle bayésien pour chercher rapidement les modes d'une loi *a posteriori* (Jiang et al., 2012a).

Chemin empruntés par les estimateurs successifs. Par construction de l'expérience, tous les points correspondant à l'estimateur MAP sur la figure 2.11-b devraient être superposés et correspondre à un point de la courbe FITRA. Les variations de performances de l'estimateur MAP résultent de l'efficacité numérique de l'algorithme. En revanche, on ne peut pas certifier que l'estimateur MMSE définie par

$$\hat{\mathbf{x}}_{\text{MMSE}} = \mathbb{E}[\mathbf{x}|\mathbf{y}] = \left(\frac{1}{\pi\beta}\right)^{\frac{M}{2}} \frac{(N\mu)^{N-\frac{M}{2}}}{2^N N!} \int \mathbf{x} \exp\left(-\frac{\mu N}{\beta} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 - \mu N \|\mathbf{x}\|_\infty\right) d\mathbf{x},$$

ne dépend que de la valeur de β et surtout soit confondu avec l'estimateur MAP. Notons que plusieurs travaux, consacrés à l'étude des différences entre les estimateurs MAP et MAPm amène à penser le contraire. Par exemple, Gribonval (2011) a montré que dans le cas d'une tâche de débruitage avec bruit Gaussien, l'estimateur MMSE correspond à l'estimateur MAP d'un modèle différent. Plus récemment, Pereyra (2016b) interprète les deux estimateurs MAP et MMSE en terme de géométrie différentielle. Ces résultats sont cependant inapplicables ici car les théorèmes nécessitent que la loi *a posteriori* du modèle soit \mathcal{C}^1 , ce qui n'est pas le cas ici. Pour conclure, on ne sait pas expliquer le comportement de l'estimateur MMSE.

2.4 RÉDUCTION DU FACTEUR DE CRÊTE (PAPR)

LE FACTEUR DE CRÊTE noté PAPR en anglais, est une mesure de l'étalement d'un signal. C'est une grandeur fondamentale dans de nombreuses applications en télécommunication, telles que le dimensionnement d'amplificateurs, cf. partie 1.1.5. Nous étudions numériquement dans cette partie la capacité du modèle BAC-1 proposé dans la partie 2.2 à identifier des codes à faible PAPR, sans avoir à régler de paramètres.

Les parties 2.4.1 et 2.4.2 font le lien entre la réduction de PAPR et le codage antiparcimonieux. La partie 2.4.3 compare numériquement nos estimateurs bayésiens avec FITRA et PFA, deux algorithmes issus de la littérature. La partie 2.4.4 étudie le comportement des estimateurs lorsque le rapport N/M des dimensions augmente. Ces résultats sont présentés dans Elvira et al. (2016, 2017a).

2.4.1 Modélisation du problème

L'objectif de la réduction du facteur de crête est de trouver un code dont le PAPR est le plus proche possible de 1. Si la contrainte sur le PAPR est forte, *i.e.*, si l'on souhaite absolument trouver la meilleure représentation dont le PAPR est 1, alors les coefficients s'écrivent $x_n = s_n \alpha$, où $s_n = \pm 1$ et $\alpha > 0$. Une telle contrainte est par exemple considérée par

Jégou et al. (2012) pour la recherche approximative de plus proches voisins. Le meilleur modèle bayésien consiste à choisir une loi *a priori* sur les signes s_n telle qu'une Bernoulli et une loi sur α telle que la loi Gamma. Cependant, approcher les estimateurs associés à un tel modèle est un problème combinatoire difficile.

Si l'on relâche la contrainte sur le PAPR, alors le problème peut s'écrire comme une minimisation de PAPR sous contrainte de reconstruction de l'observation. Mais ce nouveau problème est non convexe en raison du quotient des normes dans la définition du PAPR, donc à nouveau difficile. Nous considérons un problème plus simple, à savoir minimiser seulement la norme ℓ_∞ sous contrainte de reconstruction : c'est le problème (P_∞^ε) . Nous avons fait le lien dans ce chapitre entre la version variationnel (V_∞^β) de (P_∞^ε) et le modèle bayésien décrit dans la partie 2.2.

Bien que le problème soit posé comme un problème inverse, l'objectif n'est pas de reconstruire un vecteur de coefficient ayant servi à la synthèse, mais de trouver un code faisant sens par rapport à un canal de transmission caractérisé par la matrice d'encodage \mathbf{H} . On évalue la qualité des codes $\hat{\mathbf{x}}$ de \mathbb{R}^N à l'aide de deux métriques, déjà utilisées dans la partie 2.3 : un $\text{SNR}_{\mathbf{y}}$ exprimé en dB et le PAPR, dont on rappelle les définitions

$$\text{SNR}_{\mathbf{y}}(\hat{\mathbf{x}}) = 10 \log_{10} \frac{\|\mathbf{y}\|_2}{\|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\|_2} \quad (2.60)$$

$$\text{PAPR}(\hat{\mathbf{x}}) = \frac{N \|\hat{\mathbf{x}}\|_\infty^2}{\|\hat{\mathbf{x}}\|_2^2}. \quad (2.61)$$

La solution idéale est le code dont le $\text{SNR}_{\mathbf{y}}$ est infini et le PAPR vaut 1. Intuitivement, il y a compromis à faire entre les métriques. Ce compromis est au cœur du problème variationnel (V_∞^β) , mais également dans la définition 1.7 d'un vecteur *démocratique* où la norme ℓ_∞ de la solution, et indirectement son PAPR, diminue lorsque l'on autorise une erreur de reconstruction plus grande.

Nous présenterons pour cette raison les solutions dans un plan (PAPR, $\text{SNR}_{\mathbf{y}}$), où, à l'instar d'une courbe ROC, l'objectif est d'être le plus en haut à gauche possible. Dans ce plan, illustré à la figure 2.12, les solutions des estimateurs bayésiens apparaîtront sous forme de points. On représentera également la courbe des solutions du problème variationnel (V_∞^β) vue comme une fonction du paramètre β . Cette courbe est approchée à partie des solutions que fournit l'algorithme FITRA présenté algorithme 1.4. Notons que sur la figure, $\text{SNR}_{\mathbf{y}} = 0$ et $\text{PAPR} = 1$ pour de grandes valeurs de β . En effet, lorsque le paramètre β est très grand, le terme en norme ℓ_∞ domine dans la fonction J_β^∞ intervenant dans le problème (V_∞^β) . La solution est dans ce cas le vecteur nul.

2.4.2 Pertinence du modèle bayésien

Studer et al. (2014) donnent une garantie pour justifier l'utilisation de la norme ℓ_∞ comme substitut pertinent à la minimisation de PAPR, cf.

On rappelle la définition de (P_∞^ε)

$$\hat{\mathbf{x}} = \arg \min_z \|\mathbf{z}\|_\infty \quad \text{s.c.} \quad \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2^2 \leq \varepsilon,$$

avec ε un réel positif.

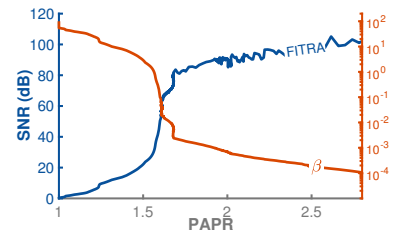


FIGURE 2.12 : Plan (PAPR, $\text{SNR}_{\mathbf{y}}$). La courbe bleue représente l'ensemble des solutions du problème (V_∞^β) pour différentes valeurs du paramètre β (courbe orange). Ici, les solutions de (V_∞^β) sont approchées avec l'algorithme FITRA.

théorème 1.6 du chapitre 1. En effet, toute solution du problème (P_∞^ε) vérifie, lorsque la matrice d'encodage est bien conditionnée,

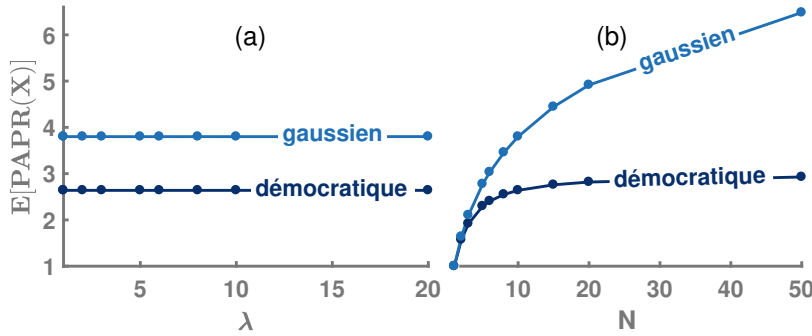
$$\|x\|_\infty \leq \frac{1}{\sqrt{N-M+1}} \|x\|_2.$$

La borne supérieure ne dépend pas de l'erreur de reconstruction ε autorisée. Ce n'est donc probablement pas une borne optimale. Toutefois, ce résultat permet d'affirmer que toute solution \hat{x} de (P_∞^ε) vérifie

$$\text{PAPR}(\hat{x}) \leq \frac{N}{N-M+1}. \quad (2.62)$$

C'est une borne supérieure sur le PAPR d'une solution de (P_∞^ε) . Cette borne supérieure tend vers 1 lorsque $M \ll N$, comme illustrée par la figure 2.13. Ce résultat montre que le problème simplifié (P_∞^ε) reste pertinent dans le cadre de la réduction de PAPR.

On justifie maintenant la pertinence de la loi démocratique comme *a priori* sur le code en comparant un PAPR intégré. Un *a priori* classique en modélisation bayésien est la loi normale, ce qui reviendrait à remplacer la norme ℓ_∞ par la norme ℓ_2 dans (V_∞^β) .



La figure 2.14 compare l'évolution des quantités $\mathbb{E}[\text{PAPR}(X)]$ sous la loi démocratique et sous une loi normale multivariée de matrice de covariance $\lambda \mathbb{I}_N$. Pour rendre les comparaisons cohérentes, la variance de la gaussienne est ajustée afin de correspondre à la variance de chaque composante de la loi démocratique, cf. Éq. (2.8).

PAPR intégré indépendant de λ . La figure 2.14-a montre que l'espérance *a priori* du PAPR ne dépend pas du paramètre d'échelle de la loi démocratique ni de la variance de la gaussienne. En effet,

$$\mathbb{E}_{f_\lambda} [\text{PAPR}(X)] = \int_{\mathbb{R}^N} \frac{N \|z\|_\infty^2}{\|z\|_2^2} f_1(z) dz, \quad (2.63)$$

où f_λ désigne la densité de la loi normale multivariée de matrice de covariance $\lambda^2 \mathbb{I}_N$ ou la loi démocratique $\mathcal{D}_N(\lambda)$. Le terme de droite dans (2.63) après changement de variable ne dépend plus de λ .

PAPR plus petit pour $\mathcal{D}_N(\lambda)$. À variance égale, on observe figure 2.14-b que la loi démocratique propose en moyenne des échantillons avec un PAPR plus faible lorsque N augmente. Dans tous les cas, on vérifie bien

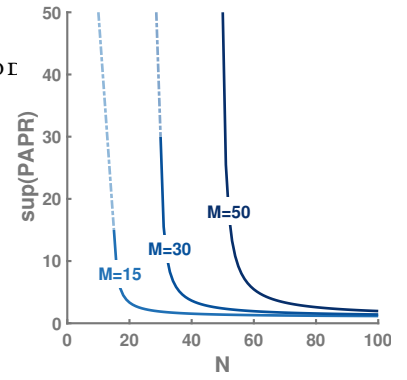


FIGURE 2.13 : Comportement du PAPR des estimateurs en fonction de la dimension N des coefficients, pour différentes valeurs de la dimension M des observations. La ligne pointillée atténuée représente une extrapolation linéaire du PAPR pour $N < M$.

FIGURE 2.14 : Évolution du PAPR sous les lois démocratique (courbe bleue foncée) et normale (courbe bleue claire), (a) en fonction du paramètre λ lorsque $N = 10$ et (b) de la dimension N lorsque $\lambda = 3$. La variance des composantes de la gaussienne est ajustée pour correspondre à la variance de chaque composante de la loi démocratique, donnée équation (2.8).

via le changement de variable $z = \lambda x$, d'où $dz = \lambda^{-N} x$.

que la loi démocratique est un meilleur *a priori* pour favoriser les représentations à faible PAPR. Notons que l'on aurait pu également comparer le comportement du PAPR sous des lois de la forme $p(\mathbf{x}) \propto \exp(-\lambda \|\mathbf{x}\|_p)$ avec $2 < p < +\infty$. On s'attend dans ce cas à passer continûment de l'une à l'autre des courbes en augmentant p .

2.4.3 Comparaison avec des algorithmes supervisés

Ce premier scénario considère des données en petites dimensions : $M = 40$ et $N = 60$. Les observations \mathbf{y} , sont générées suivant des lois normales⁷, comme dans Studer et al. (2014). Les matrices d'encodages \mathbf{H} sont générées en sous-échantillonnant les lignes d'une matrice DCT quelconque, puisque l'on a vu dans la partie 1.3.3 que de telles matrices garantissent avec forte probabilité l'existence de représentations antipar-cimonieuses. Les performances sont moyennées sur 50 simulations. Pour chaque simulation, un vecteur d'observations \mathbf{y} et une matrice d'encodage \mathbf{H} sont générés. Les estimateurs bayésiens sont calculés sur 1×10^4 itérations après une période de chauffe de 2×10^3 itérations. Nos algorithmes sont comparés à deux méthodes issues de la littérature, évoquées au chapitre 1 : FITRA et PFA. FITRA et PFA sont deux méthodes supervisées qui nécessitent de fixer la valeur de β dans J_β^∞ ⁸. On utilise deux valeurs de β . La première est obtenue en prenant $\beta_{\text{MMSE}} \triangleq (N + 1)\mu_{\text{MMSE}}\sigma_{\text{MMSE}}^2$, où μ_{MMSE} et σ_{MMSE}^2 sont les estimateurs MMSE des deux paramètres de nuisance, cf. partie 2.2.3.b. On ajuste la seconde valeur de β , notée β_{snr} de sorte à obtenir le même $\text{SNR}_{\mathbf{y}}$ que le meilleur estimateur bayésien. Ces solutions sont désignées par l'ajout des suffixes « snr » et « mmse ». Pour illustrer l'effet de régularisation de la norme ℓ_∞ , nos estimateurs sont également comparés à la solution des moindres carrés, ainsi qu'aux estimateurs MAPm et MMSE d'un modèle bayésien où l'on choisit une loi *a priori* normale sur les coefficients \mathbf{x} . Notons que ce dernier modèle revient à résoudre un problème des moindres carrés régularisé par une norme ℓ_2 .

7. La raison pour laquelle le modèle de synthèse est différent du modèle génératif a déjà été évoquée précédemment.

8. On rappelle, cf. Éq. (2.1) p. 43

$$J_\beta^\infty(\mathbf{u}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{u}\|_2 + \beta \|\mathbf{u}\|_\infty$$

	$\text{SNR}_{\mathbf{y}}$ (db)	PAPR	temps (s)
P-MALA mapm	29.3	2.8	3.2×10^1
P-MALA mmse	19.3	3.9	<i>idem</i>
BAC Gibbs mapm	8.8	3.0	1.4×10^2
BAC Gibbs mmse	4.3	2.9	<i>idem</i>
FITRA-mmse	34.4	1.7	2.6×10^{-2}
FITRA-snr	29.3	1.9	<i>idem</i>
PFA-mmse	29.6	1.6	5.3×10^{-2}
PFA-snr	29.3	1.8	<i>idem</i>
LS	∞	6.6	6.1×10^{-2}
Gaussien mapm	∞	5.9	1.9×10^1
Gaussien mmse	73.1	6.8	<i>idem</i>

TABLE 2.4 : Scénario 1 : Résultats exprimés en termes de $\text{SNR}_{\mathbf{y}}$ et PAPR pour tous les algorithmes considérés. Nous considérons comme infini les $\text{SNR}_{\mathbf{y}}$ supérieurs à 100 dB.

Le tableau 2.4 résume l'ensemble des résultats. *Performances numériques.* Les estimateurs MAPm et MMSE approchés

avec P-MALA atteignent une erreur de reconstruction de $SNR_y = 29.3\text{dB}$ et $SNR_y = 19.3\text{dB}$, respectivement. En revanche, les estimateurs approchés avec BAC Gibbs ont à la fois un SNR_y plus faible et un PAPR plus élevé. À SNR_y équivalent, les algorithmes supervisés FITRA et PFA atteignent toujours de meilleures performances comparés aux estimateurs bayésiens, et le plus performant est toujours FITRA.

Effet de la régularisation ℓ_∞ . Les solutions estimées par BAC-1, FITRA et PFA ont toutes un PAPR plus faible, *i.e.*, un étalement plus faible, comparées aux estimateurs des moindres carrés où obtenus avec une pénalisation ℓ_2 . Cette observation valide empiriquement l'effet de la norme ℓ_∞ comme pénalisation pertinente pour la réduction de PAPR.

Temps de calcul. À la différence des simulations dans la partie 2.3, tous les algorithmes présentés dans cette partie sont implémentés en MATLAB, y compris BAC Gibbs et BAC P-MALA. On observe que la différence d'implémentation a un effet non négligeable sur les résultats : le coût d'une itération est plus élevé d'un facteur 100 pour BAC Gibbs. En conséquence, les estimateurs ont été approchés avec un nombre plus faible d'itérations et sont moins pertinents. Ici, BAC Gibbs est plus lent que BAC P-MALA d'un facteur 10, alors que nous avons conclu dans la partie 2.3 que les deux algorithmes étaient équivalents en terme de complexité. Ici, la différence provient du nombre de boucle for.

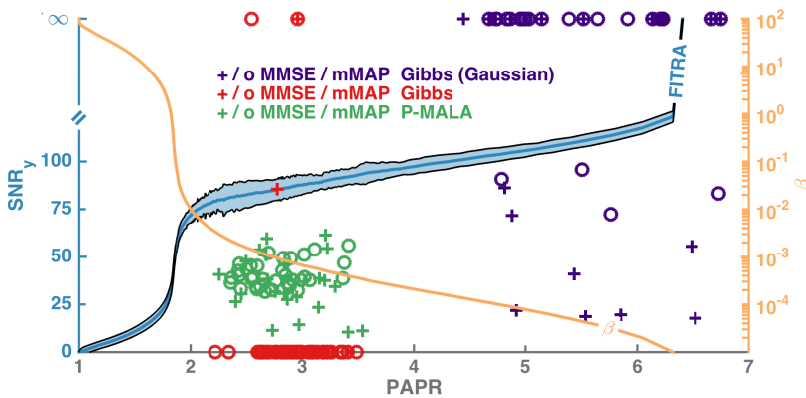


FIGURE 2.15 : Scénario 1 : Estimateurs bayésiens affichés dans le plan (PAPR, SNR_y). La courbe bleue représente les solutions du problème (V_∞^β) pour toutes les valeurs du paramètre β approchées avec l'algorithme FITRA. Les croix / cercles représentent respectivement les estimateurs MMSE / MAPm pour le modèle avec un *a priori* gaussien sur le code, BAC Gibbs et BAC P-MALA.

La figure 2.15 montre les estimateurs bayésiens dans le plan (PAPR, SNR_y), superposés avec la courbe des solutions de FITRA obtenue en faisant varier continûment le paramètre β de l'algorithme.

Existence d'une transition de phase. On remarque que les solutions du problème (V_∞^β) approchées avec l'algorithme FITRA pour toutes les valeurs de β mettent en évidence une région appelée *zone critique*. Cette zone, illustrée à la figure 2.16, est située autour de la transition de phase, lorsque les solutions, à faibles PAPR et SNR_y passent brusquement à des solutions à PAPR et SNR_y élevés.

Positions des estimateurs bayésiens. Les estimateurs approchés avec BAC Gibbs sont presque tous situés dans une zone à très faible SNR_y mais à PAPR comparable à ceux de P-MALA. Il s'agit donc d'estimateurs peu pertinents. En revanche, les estimateurs approchés avec P-MALA sont presque tous situés au niveau de la zone critique. On en conclut que

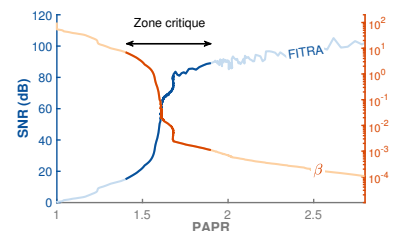


FIGURE 2.16 : Solutions de l'algorithme FITRA (courbe bleue) dans le plan (PAPR, SNR_y) obtenu en faisant varier la valeur du paramètre β (courbe orange). La zone foncée correspond à une de transition de phase, appelée *zone critique* ici.

le modèle BAC-1 est capable d'identifier de façon non supervisée des solutions pertinentes au problème de réduction de PAPR. Nous avons de plus observé qu'à nombre d'itérations équivalents, l'algorithme BAC P-MALA est plus efficace que BAC Gibbs.

2.4.4 Performances en fonction du rapport des dimensions

Nous proposons de vérifier expérimentalement l'amélioration des performances des estimateurs bayésiens lorsque le ration N/M augmente. On s'attend à ce que le PAPR diminue à SNR_y constant ou que le SNR_y augmente à PAPR constant, cf. Éq. (2.62). L'intuition derrière ce résultat est que lorsque le rapport augmente, le nombre de combinaison de coefficients la forme $\pm \|\mathbf{y}\|_2 / \sqrt{N}$ augmente exponentiellement avec N . Inversement, si le PAPR reste constant, on s'attend à ce que la reconstruction soit meilleure. Ce comportement a d'ailleurs été empiriquement constaté par Studer et al. (2012).

Protocole. Dans ce scénario, la dimension M est fixée à $M = 128$, et la dimension des coefficients va varier de $N = 128$ à $N = 256$, soit un rapport N/M variant de 1 à 2. Les observations \mathbf{y} et les matrices de codage \mathbf{H} sont générées comme dans l'expérience précédente, cf. partie 2.4.3. On présente uniquement dans cette partie les estimateurs MMSE et MAPm approchés avec l'implémentation MATLAB de BAC P-MALA. Ces estimateurs seront à nouveau comparés à deux versions de FITRA : FITRA-SNR et FITRA-MMSE. Nos estimateurs bayésiens sont approchés avec 5×10^4 itérations après une période de chauffe de 5×10^3 itérations.

On rappelle que FITRA-mmse s'obtient en prenant $\beta_{\text{MMSE}} \triangleq (N+1)\mu_{\text{MMSE}}\sigma_{\text{MMSE}}^2$ comme paramètre de régularisation, et FITRA-snr s'obtient en ajustant la valeur de β de sorte à obtenir le même SNR_y que le meilleur estimateur bayésien.

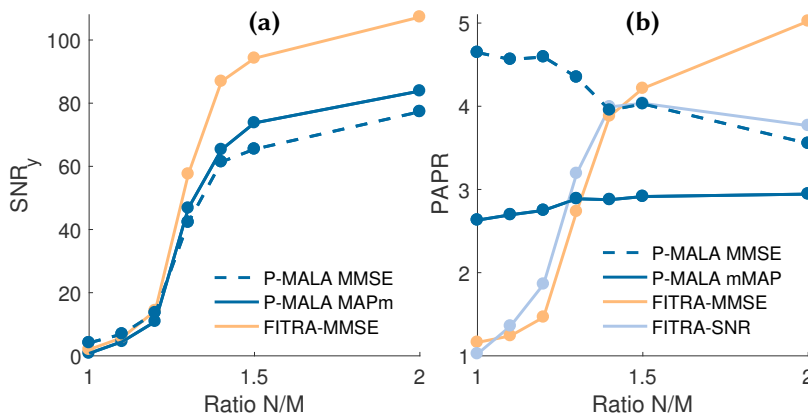


FIGURE 2.17 : Scénario 2 : évolutions du (a) SNR_y et (b) PAPR des estimateurs en fonction du rapport N/M .

La figure 2.17 montre l'évolution des deux critères, à savoir : (a) le SNR_y et (b) le PAPR des estimateurs en fonction du ratio.

PAPR constant mais augmentation du SNR_y . À l'instar de l'expérience précédente, FITRA-MMSE atteint un meilleur SNR_y que les estimateurs bayésiens. On remarque cependant à la figure 2.17-b que les deux estimateurs bayésiens atteignent un PAPR plus faible, *i.e.*, un étalement plus faible, lorsque le rapport N/M dépasse 1.4. on observe également pour les deux estimateurs bayésiens MMSE et MAPm approchés avec P-MALA que le SNR_y augmente avec le ration N/M tandis que le PAPR reste constant, voire décroît légèrement pour l'estimateur MMSE. En conclu-

sion, nos estimateurs bayésiens sont de plus en plus performants au fur à mesure que le rapport N/M augmente.

2.5 CONCLUSION

NOUS AVONS proposé dans ce chapitre une modélisation bayésienne du codage antiparcimonieux. Nous avons vu au chapitre 1 qu'un code antiparcimonieux $\hat{\mathbf{x}}$ d'un signal \mathbf{y} sur une matrice de codage \mathbf{H} s'obtient en résolvant le problème contraint, où ϵ est un réel positif

$$\hat{\mathbf{x}} = \arg \min_z \|\mathbf{z}\|_\infty \quad \text{s.c.} \quad \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2^2 \leq \epsilon. \quad (\mathbf{P}_\infty^\epsilon)$$

Nous avons construit le modèle bayésien en interprétant la formulation variationnelle de $(\mathbf{P}_\infty^\epsilon)$, à savoir

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{z} \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2^2 + \beta \|\mathbf{z}\|_\infty, \quad (\mathbf{V}_\infty^\beta)$$

comme l'estimateur MAP d'un modèle bayésien dont la vraisemblance est Gaussienne et la loi *a priori* sur le code \mathbf{x} de la forme $p(\mathbf{x}) \propto \exp(-\lambda \|\mathbf{x}\|_\infty)$ où λ est un réel positif. Les trois prochains paragraphes résument les contributions de ce chapitre.

Étude de la loi démocratique. La loi *a priori* sur le code que nous venons d'évoquer n'avait à notre connaissance pas encore été étudiée. La partie 2.1 a été consacrée à l'étude des principales propriétés de cette loi que nous avons baptisée *loi démocratique* en références aux représentations démocratiques, cf. chapitre 1 : constante de normalisation, moments, lois marginales et conditionnelles. Nous avons également proposés trois échantillonneurs de la loi démocratique : un échantillonneur exact, un échantillonneur de Gibbs et P-MALA, un algorithme utilisant des méthodes proximales.

Modèle bayésien pour le codage antiparcimonieux. Nous avons construit dans la partie 2.2 BAC-1, un modèle bayésien pour le codage antiparcimonieux en choisissant la loi démocratique comme *a priori* sur le code. Les paramètres de nuisance, à savoir la variance de l'erreur résiduelle et le paramètre d'intensité de la loi démocratique sont intégrés au modèle bayésien à travers des loi *a priori* conjuguées. Nous avons proposé trois estimateurs du code : un estimateur MAP et deux estimateurs intégrés MMSE et MAPm. Nous avons proposé trois échantillonneurs de Gibbs pour approcher ces estimateurs qui diffèrent quant à l'échantillonnage du code : BAC Gibbs procède composante par composante, BAC P-MALA s'appuie sur P-MALA mais nécessite une approximation du premier ordre de l'opérateur proximal de la neg-log-posterior et BAC P-MYULA exploite la régularité de la neg-log-posterior pour éviter cette approximation.

Simulation numérique et application à la réduction de PAPR. Les deux parties suivantes ont été consacrées aux simulations numériques. La partie 2.3 a étudié les performances des trois algorithmes lorsque les

paramètres de régularisations sont fixés. Nous avons montré que les estimateurs bayésiens sont capables d'identifier les mêmes solutions que FITRA, un algorithme déterministe minimisant (V_∞^β). Dans la partie 2.4, nous avons appliqué le modèle BAC-1 pour la réduction de facteur de crête, ou PAPR. Nous avons montré que les estimateurs sont capables d'identifier des solutions pertinentes de façon non supervisée.

Ces travaux sont publiés dans Elvira et al. (2016, 2017a, 2017d). Les codes MATLAB/ mex des algorithmes présentés dans ce chapitre sont disponibles [en ligne](#).



2.6 PERSPECTIVES

2.6.1 Améliorer le modèle et consolider les expériences numériques

Nous pensons qu'il est possible de significativement améliorer les résultats numériques de la partie 2.4. D'abord, P-MYULA a montré de très bonnes performances à la fois en terme de reconstruction et en temps de calculs.

Le modèle BAC-1 présenté dans la partie 2.2.1 modélise indépendamment les paramètres de nuisance σ^2 et μ . Or, c'est le produit $\beta = \sigma^2(N + 1)\mu$ qui contrôle l'estimateur MAP de x , ou de manière équivalente, la solution du problème (V_∞^β). Nous proposons de considérer un nouveau modèle, *Bayesian Antisparsity Coding 2* (BAC-2), où l'on paramètre directement le produit $\beta = \sigma^2\lambda$, tout en capitalisant sur les résultats de la partie 2.3.4 pour calibrer la valeur de σ^2 et μ . En choisissant à nouveau une loi *a priori* conjuguée Inverse Gamma sur β , la loi *a posteriori* est encore par conjugaison une loi Inverse Gamma et l'échantillonneur de Gibbs décrit algorithme 2.4 est encore valide en corrigeant la ligne 3.

La figure 2.18 montre les premiers résultats numériques du modèle BAC-2. La figure 2.18-a affiche 200 itérations de l'algorithme MCMC basé sur P-MYULA. On observe que ces itérations sont désormais proches de la courbe FITRA et de la transition de phase, et l'on peut considérer que l'estimateur MAPm est sur la courbe. La figure 2.18-b montre la distribution *a posteriori* marginale $\beta|y$ qui est désormais interprétable.

Pour finir, les résultats de la partie 2.4 ont été obtenus à partir d'une implémentation MATLAB. Nous avons constaté que cette implémentation est moins efficace qu'une implémentation C++, jusqu'à un facteur 100 pour l'échantillonneur composante par composante. Maintenant que tous les algorithmes ont été ré-implémentés, il serait intéressant de relancer les simulations en augmentant le nombre d'itérations et les dimensions du problème.

2.6.2 Concevoir un estimateur plus efficace

Nous avons étudié trois estimateurs des codes antiparcomieux : deux estimateurs MAP et un estimateur MMSE. L'estimateur MAP est justifié

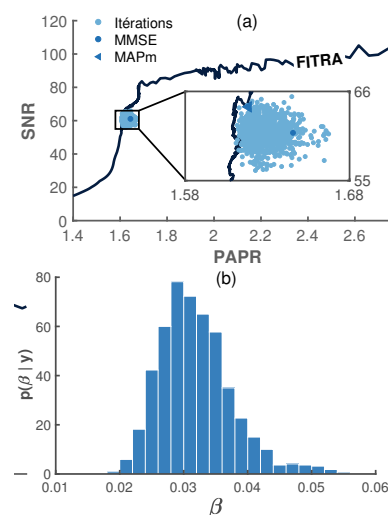


FIGURE 2.18 : (a) 200 dernières itérations de l'algorithme MCMC lorsque le paramètre β est également échantillonné. La courbe noire représente l'ensemble des solutions de FITRA obtenue en faisant varier β . (b) Loi *a posteriori* marginale $\beta|y$.

car il coïncide avec la solution du problème variationnel (\mathbf{V}_∞^β) et l'estimateur MMSE s'est révélé numériquement pertinent. Toutefois, nous rappelons que le MMSE est conçu pour minimiser l'erreur quadratique intégrée sur le modèle. Or, le codage antiparcimonieux a pour objectif d'identifier des codes pertinents vis-à-vis d'un canal de transmission et non pas de retrouver les paramètres d'un modèle de synthèse.

Nous pensons qu'il est possible d'exploiter la distribution empirique $p(\mathbf{x}|\mathbf{y})$ pour concevoir un estimateur plus pertinent et plus efficace. Nous avons vu que toute solution du problème contraint ($\mathbf{P}_\infty^\varepsilon$) possédait au minimum $N - M + 1$ valeurs extrêmes, cf. partie 1.3.2.a. Cependant, les indices et le signe de ces valeurs extrêmes sont inconnus. Nous pensons pouvoir estimer ces quantités à partir de la distribution $p(\mathbf{x}|\mathbf{y})$.

2.6.3 Changer la géométrie du problème

La loi démocratique a été construite pour faire correspondre l'estimateur MAP du code antiparcimonieux avec la solution du problème variationnel (\mathbf{V}_∞^β). Nous avons aussi remarqué que le comportement de cette loi n'est pas optimal pour le codage antiparcimonieux. Par exemple, on a vu au lemme 2.1 que les composantes non-dominantes sont distribuées uniformément sur le compact $[-|x_d|, |x_d|]$, où x_d est la composante dominante. Or, on aimerait plutôt favoriser les zones proches en valeur absolue de $|x_d|$ afin de réduire l'amplitude du signal.

La raison est que la densité de la loi démocratique ne dépend que de la valeur de la composante dominante, à travers le terme $\|\mathbf{x}\|_\infty$. Ce comportement peut être corrigé en ajoutant un paramètre de forme. Nous introduisons la loi démocratique à deux paramètres, dont la densité s'écrit pour tout vecteur \mathbf{x} de \mathbb{R}^N

$$f(\mathbf{x}) = \frac{\lambda^N \gamma^N}{2^N \Gamma(N\gamma + 1)} \prod_{n=1}^N |x_n|^\gamma e^{-\lambda \|\mathbf{x}\|_\infty}, \quad (2.64)$$

où γ et λ sont deux réels positifs. On peut montrer que la loi conditionnelle d'une composante dominante est de la forme $p(x|x_d) \propto |x_d|^\gamma$ où x_d est la composante dominante et γ le paramètre de forme de la loi.

La figure 2.19-a montre la densité de probabilité de la loi démocratique à deux paramètres en dimension 2 et vue du dessus, pour $(\gamma, \lambda) = (2, 1)$. On remarque que la loi a tendance à favoriser les zones autour des diagonales, donc les codes à faible PAPR.

La figure 2.19-b montre la loi conditionnelle des composantes pour plusieurs valeurs du paramètre de forme γ . On constate que plus γ est grand et plus la loi a tendance à favoriser des valeurs proches de la composante dominante, ce qui est le comportement recherché.

2.6.4 Étudier la loi démocratique renormalisée

La densité de la loi démocratique à deux paramètres ressemble à celle d'une loi Gamma multivariée. Or, nous verrons dans le prochain chapitre

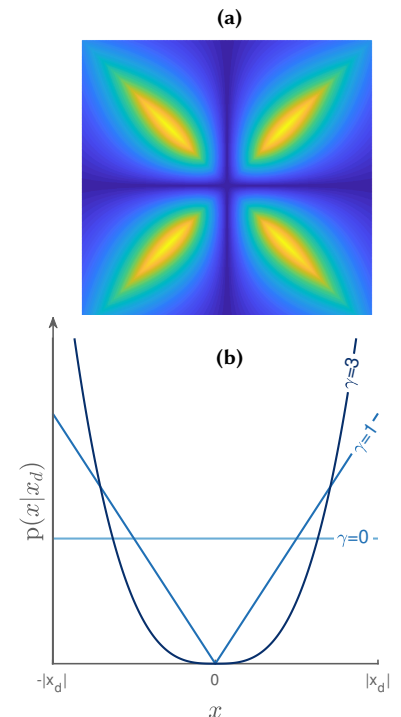


FIGURE 2.19 : (a) Densité de probabilité de la loi démocratique généralisée en dimension 2 pour $(\gamma, \lambda) = (2, 1)$. La couleur jaune correspond aux zones à forte probabilité. (b) Loi d'une composante non-dominante conditionnée à la valeur de la composante dominante pour la démocratique à deux paramètres pour plusieurs valeurs de γ .

que les lois Gamma multivariées sont utilisées pour construire la loi de Dirichlet, une famille de lois sur l'ensemble des vecteurs à coefficients positifs dont la somme vaut 1, aussi appelé *simplexe*. En suivant le même schéma, on peut construire une telle loi à partir de la loi démocratique.

Considérons la loi démocratique à deux paramètres tronquée à \mathbb{R}_+^N . La densité de cette nouvelle loi est la même que celle présentée en (2.64) à une constante multiplicative 2^N près. La loi d'un vecteur renormalisé est donnée par le théorème suivant

Théorème 2.2. Soit $\mathbf{X} = [X_1, \dots, X_N]$ un vecteur aléatoire distribué selon la loi démocratique à deux paramètres tronquée à \mathbb{R}_+^N de paramètre λ, γ . Alors le vecteur aléatoire $\mathbf{Y} = (\sum_{n=1}^N X_n)^{-1} \mathbf{X}$ est distribué selon

$$\forall \mathbf{y} \in \mathbb{R}_+^N, \quad p(\mathbf{y}) = \frac{\gamma^{N-1}}{N \|\mathbf{y}\|_\infty^{N\gamma}} \prod_{\ell=1}^N y_\ell^{\gamma-1} \mathbb{1}_{\Delta_N}(\mathbf{y}). \quad (2.65)$$

où Δ_N est l'ensemble des vecteurs de \mathbb{R}^N , positifs et de somme égale à 1.

La preuve est disponible en annexe A.6.

La figure 2.20 montre la densité de la loi renormalisée sur Δ_3 . Le maximum est atteint en $[1/3, 1/3, 1/3]$, tandis que les minima sont les vecteurs de la forme $[0, \dots, 1, \dots, 0]$. Comme pour la loi démocratique, la densité est décrite en dimension 3 par un cône à tête triangulaire.

Les propriétés de cette nouvelle famille de lois sont intéressantes pour la classification non supervisée. Par exemple, il est parfois intéressant d'obtenir des groupes de données de taille similaire, voir par exemple Miller et al. (2015). La loi que nous venons de présenter est un bon candidat comme *a priori* sur les proportions des classes dans un modèle bayésien pour favoriser des proportions comparables.

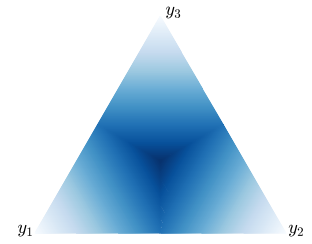


FIGURE 2.20 : Densité vue du dessus de la loi \mathcal{D}_N^+ renormalisée en dimension $N = 3$. La couleur blanche représente les zones à faibles probabilités.

Deuxième partie

MODÈLE BAYÉSIEN NON PARAMÉTRIQUE
POUR L'ESTIMATION DE SOUS-ESPACES

Méthodes bayésiennes non paramétriques

pour les modèles à variables latentes

LES MODÈLES à variables latentes, (*latent feature models* en anglais) sont une classe de modèles pour décrire des observations ou données par des paramètres souvent non observés mais éventuellement partagés. Par exemple, la figure 3.1 montre un jeu de données constitués de six plats. Chaque plat est obtenu en mélangeant et en travaillant plusieurs ingrédients et seul le résultat final est observé. Ici, les ingrédients sont les variables latentes.

Lorsque l'on impose que chaque observation ne soit décrite que par un unique paramètre, on parle plutôt de *classification non supervisée* (*clustering* en anglais). On cherche à regrouper les observations suivant une mesure de similarité. Chaque variable latente est associée à un groupe, appelé *classe*. Dans l'exemple de la figure 3.1, on s'attend dans ce cas à ce que les classes séparent les plats par catégories : en *soupe*, en *sauce*, en *grillade*, etc.

Une question importante lors de la phase de modélisation est le choix du nombre de degrés de liberté, ou de manière équivalente, le nombre de paramètres du modèle. Ici, cela correspond au nombre d'ingrédients à découvrir ou de catégorie de plats. Une solution à ce problème consiste à intégrer la complexité au sein d'un modèle bayésien hiérarchique. Le nombre de degrés de liberté devient une variable aléatoire où l'incertitude est modélisée par une loi *a priori*, telle que la loi de Poisson. Cependant, ces modèles sont souvent difficiles à manipuler et requièrent des méthodes d'échantillonnage sophistiquées du fait du nombre de paramètres variables.

À l'inverse, les *méthodes bayésiennes non paramétriques* considèrent qu'il existe une infinité dénombrable de paramètres, mais que seuls un nombre fini sont utilisés par un nombre fini de réalisations. Par construction, de nouveaux paramètres sont découverts au fur et à mesure que les données sont observées. En d'autres mots, le modèle s'affine avec l'arrivée de nouvelles observations. Dans notre exemple, cela signifie que le nombre d'ingrédients augmente au fur et à mesure que l'on goûte à de nouveaux plats. Toutefois, on souhaite décrire nos plats avec moins d'ingrédients que de plats observés, sinon la classification est inutile. En effet, si l'on autorise le nombre de paramètres à croître trop vite avec le nombre d'observations, un risque de *sur-apprentissage* apparaît. Nous verrons que les modèles bayésiens non paramétriques induisent naturellement un effet régularisant pour éviter le sur-apprentissage.



FIGURE 3.1 : Jeu de données formées de plats

1. Bò bún
2. Magret de canard aux figues
3. Feijoada
4. Paella
5. Bún riêu
6. Carbonnade flamande

La partie 3.1 est consacrée à la classification non supervisée. Par analogie avec un modèle paramétrique, nous construisons un modèle non paramétrique dont l'ingrédient principal sera le *processus de Dirichlet*. La partie 3.2 est consacrée aux modèles à variables latentes, où nous introduirons le processus *Beta-Bernoulli*. La dernière partie dresse un bilan du chapitre.

3.1 CLASSIFICATION NON SUPERVISÉE ET PROCESSUS DE DIRICHLET À MÉLANGE

LE PREMIER modèle non paramétrique étudié dans ce chapitre est le processus de Dirichlet à mélange, que l'on introduit à partir du problème de classification non supervisée (ou *clustering* en anglais). Nous aboutirons à un nouveau processus à valeurs sur l'ensemble des partitions : le processus du restaurant chinois.

Étant donné un ensemble de N observations $\mathbf{y}_1, \dots, \mathbf{y}_N$ de \mathbb{R}^D , noté $\mathbf{y}_{1:N}$, la classification non supervisée vise à séparer ces points en K classes (ou *clusters* en anglais). Le nombre de classes K n'est pas connu à l'avance, sauf mention explicite du contraire.

La figure 3.2 illustre ce problème en dimension $D = 2$ pour $N = 1000$ points. Une réponse courante à la question « Combien y a-t-il de classe sur cet exemple ? » est 3. De nombreuses méthodes et heuristiques ont été proposées dans le cadre de la sélection de modèle. Nous aurons l'occasion d'y revenir dans la fin de cette partie.

Si l'on accepte cette réponse, il reste à séparer les indices $1 \dots N$ des points en 3 sous-ensembles disjoints qui constituent une *partition* de $\llbracket 1, N \rrbracket$. La partie 3.1.1 construit un modèle bayésien paramétrique adapté à ce problème. Son rôle est double : formaliser le problème et introduire les notations et la démarche qui sera généralisée dans la partie suivante. Si l'on souhaite que le modèle évalue lui-même la diversité au sens statistique des données, le modèle bayésien paramétrique ne suffit plus. La partie 3.1.3 discute le processus de Dirichlet à mélange, un modèle bayésien non-paramétrique où le nombre de classes de la partition est libre.

3.1.1 Modèles de mélange bayésiens

La formulation probabiliste la plus courante du problème de classification non supervisée utilise les *modèles de mélange*. Chaque classe C_k est décrite par une densité de probabilité p_k éventuellement paramétrée par $\theta_k \in \Theta$, où Θ est l'espace des paramètres. On supposera que les densités p_k appartiennent à la même famille de lois.

Dans l'exemple figure 3.2, les classes ont l'air elliptiques. Une loi normale multivariée est donc adaptée pour décrire une classe. Dans ce cas, le paramètre θ_k désigne le vecteur moyenne $\boldsymbol{\mu}_k$ et la matrice de covariance $\boldsymbol{\Sigma}_k$, tels que $\theta_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Notons ℓ_1, \dots, ℓ_N les variables associant une observation à une classe, i.e., $\ell_n = k$ si \mathbf{y}_n appartient à la classe C_k . Les partitions des données

La structure des deux parties est similaire : on introduit d'abord le processus stochastique associé à la formulation bayésienne, puis on déduit de cette partie technique un processus plus simple à étudier et qui sera utilisé en pratique.

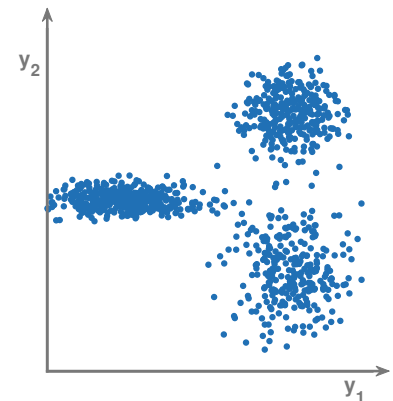


FIGURE 3.2 : 1000 points dans \mathbb{R}^2 générés aléatoirement et que l'on souhaite séparer en classes. Nous réutiliserons cet ensemble de points dans cette partie.

s'obtiennent à partir des variables ℓ_1, \dots, ℓ_N . Si \mathbf{y}_n appartient à la classe \mathcal{C}_k , on sait que

$$\mathbf{y}_n | \ell_n = k \sim \mathbf{p}_k(\cdot | \theta_k). \quad (3.1)$$

Marginalement, la loi d'une observation est donnée par

$$\mathbf{y}_n \sim \sum_{k=1}^K \mathbb{P}[\ell_n = k] \mathbf{p}_k(\cdot | \theta_k). \quad (3.2)$$

où K est fini pour les modèles paramétriques et infini sinon. On pose $a_k \triangleq \mathbb{P}[\ell_n = k]$, la proportion de la classe \mathcal{C}_k . La densité marginale (3.2) est appelée *modèle de mélange*. C'est une combinaison convexe de densités de probabilité. Par construction, la suite $\mathbf{a} \triangleq (a_k)_k$ vérifie $\sum_{k=1}^K a_k = 1$. Ces conditions de positivité et de somme à 1 définissent une variété appelée *simplexe* notée Δ

$$\Delta \triangleq \left\{ (a_k)_{k \in \mathbb{N}} \mid \forall k \in \mathbb{N}^*, a_k \geq 0 \text{ et } \sum_{k \in \mathbb{N}^*} a_k = 1 \right\}. \quad (3.3)$$

Restreint à un nombre fini K d'éléments, le simplexe est réduit à un ensemble de K -uplet noté Δ_K . Notons que Δ_K est une partie d'un hyperplan de \mathbb{R}^K . La figure 3.3 représente le simplexe Δ_3 .

Étant donnée la connaissance d'une suite de \mathbf{a} et d'une suite de paramètres $\boldsymbol{\theta} \triangleq (\theta_k)_k$, notons φ la mesure discrète définie sur Θ par

$$\forall \theta \in \Theta, \quad \varphi(\theta) = \sum_{k \in \mathbb{N}} a_k \delta_{\theta_k}(\theta). \quad (3.4)$$

La mesure φ est appelée *mesure de mélange*. On dit que la mesure de mélange est *finie* si les suites \mathbf{a} et $\boldsymbol{\theta}$ ne comportent qu'un nombre fini de valeurs non nulles. La mesure de mélange permet de réécrire plus simplement le modèle équation (3.2)

$$\mathbf{p}(\mathbf{y}_n) = \int_{\Theta} \mathbf{p}(\mathbf{y}_n | \theta) \varphi(d\theta). \quad (3.5)$$

En bref, la mesure de mélange φ résume toute l'information sur les poids \mathbf{a} et les paramètres $\boldsymbol{\theta}$.

3.1.2 Modèle de mélange fini bayésien et loi sur les partitions

Dans le cadre de la modélisation bayésienne, les paramètres θ_k et a_k sont également aléatoires. On rappelle que cet aléa modélise l'incertitude autour des paramètres. La mesure de mélange φ devient une *mesure aléatoire*. Une façon de générer des mesures aléatoires est de générer des suites de variables aléatoires θ_k et a_k .

Le choix de la distribution sur $\Delta \times \Theta^{\mathbb{N}}$ pour modéliser l'incertitude est libre. Toutefois, l'inférence et la modélisation sont souvent plus simples lorsque l'on utilise des variables *indépendantes et identiquement distribuées*. Dans cette partie, nous considérons un modèle de mélange fini à K classes. L'espace des paramètres se réduit à $\Delta_K \times \Theta^K$.

Il s'agit de la formule des probabilités totales.

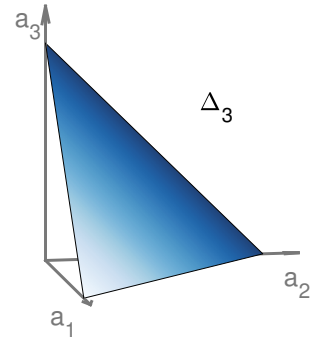


FIGURE 3.3 : Représentation du simplexe Δ_3 dans \mathbb{R}^3 . La couleur n'a ici pas de signification.

Il existe des cas où utiliser des variables dépendantes fait sens. Par exemple, si θ_k modélise le centre de la classe, alors on peut avoir envie d'ajouter au modèle de la répulsion entre les centres des classes. Ce n'est pas possible avec des variables *i.i.d.* Nous présenterons au prochain chapitre un modèle utilisant des variables dépendantes.

Soit G_0 une mesure de probabilité sur Θ . On pose

$$\forall k, \theta_k \stackrel{\text{i.i.d.}}{\sim} G_0. \quad (3.6)$$

La loi G_0 joue le rôle de loi *a priori* sur les paramètres. Par exemple, si $\theta_k = (\boldsymbol{\mu}_k, \Sigma_k)$, on peut choisir une loi *a priori* conjuguée normale-Wishart, notée \mathcal{NW} .

Une loi à valeur sur le simplexe et conjuguée avec la mesure de mélange (3.4) est la loi de Dirichlet dont la densité pour tout vecteur $\mathbf{a} = [a_1, \dots, a_K]$ de Δ_K est donnée par

$$p(\mathbf{a}) = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K a_k^{\alpha_k - 1}. \quad (3.7)$$

La loi de Dirichlet généralise la loi Beta à tout K -uplet, et contient la loi uniforme sur le simplexe. Par conjugaison, la loi *a posteriori* des poids est à nouveau une loi de Dirichlet. Sa densité en dimension 3 est illustrée à la figure 3.4.

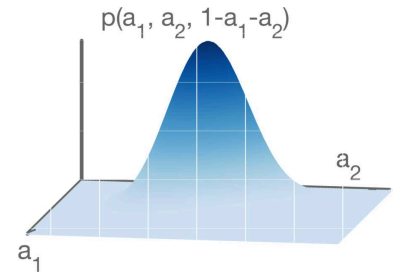


FIGURE 3.4 : Loi de Dirichlet sur Δ_3 de paramètre $(3, 3, 3)$.

Nous généraliserons dans la partie 3.1.3 la loi de Dirichlet sur Δ_K à une loi sur le simplexe Δ , le processus de Dirichlet. Pour comprendre la construction du processus, il est utile de rappeler la construction de la loi de Dirichlet que nous avons déjà évoquée dans la partie 2.6.4. L'idée est de générer un vecteur de nombres réels positifs puis de le renormaliser par la somme. Une loi qui se prête bien à ce contexte est la loi Gamma puisqu'elle est directement à valeurs dans $[0, +\infty[$.

Soient $\alpha_1, \dots, \alpha_K$ et β des réels strictement positifs. Considérons K variables aléatoires V_1, \dots, V_K distribuées suivant une loi Gamma, telles que pour tout entier k , $V_k \sim \Gamma(\alpha_k, \beta)$. Notons $V_0 \triangleq \sum_{k=1}^K V_k$. Alors le vecteur aléatoire $[V_1/V_0, \dots, V_K/V_0]$ suit une loi de *loi de Dirichlet* de paramètres $\alpha_1, \dots, \alpha_K$. Notons que la valeur de β dans la loi Gamma n'a pas d'influence sur la loi ainsi définie.

On peut se servir des valeurs de \mathbf{a} élément de Δ_K pour induire une partition ℓ_1, \dots, ℓ_N sur les observations. Le choix le plus courant comme loi *a priori* sur la partition est la loi multinomiale \mathcal{M}_K à K éléments,

$$\forall n, \ell_n | \mathbf{a} \sim \mathcal{M}_K(\cdot | \mathbf{a}). \quad (3.8)$$

Ce choix est à nouveau motivé par la conjugaison de la loi multinomiale avec la loi de Dirichlet. La loi *a posteriori* d'un label est dans ce cas à nouveau une loi multinomiale.

Nous n'évoquons que brièvement le cas de l'inférence. Si l'objectif est d'estimer une partition via les labels ℓ_k , alors il faut construire des estimateurs à partir de la loi *a posteriori* $P[\ell_1, \dots, \ell_N | \mathbf{y}_{1:N}]$. Même si l'on peut évaluer numériquement la loi *a posteriori* pour chaque N -uplets de labels, le nombre de partitions à tester, appelé *nombre de Stirling de seconde espèce*, croît exponentiellement avec le nombre d'observations¹. Pour pallier ce problème, on approche ces estimateurs à l'aide d'algorithmes **MCMC**.

1. À k fixé, le nombre de Stirling de seconde espèce, noté $\left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\}$, possède l'équivalent

$$\left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\} \underset{n \rightarrow +\infty}{\sim} \frac{k^n}{k!}.$$

3.1.3 Construction du processus de Dirichlet à mélange

Cette partie généralise le modèle de mélange bayésien paramétrique introduit dans la partie précédente. L'objet final est le *processus de Dirichlet à mélange* (pour *Dirichlet process mixture, DPM*). La construction nécessite de renormaliser un processus Gamma et demande un investissement mathématique plus important que dans le cas paramétrique. Nous verrons dans la prochaine partie comment manipuler le processus pour obtenir des algorithmes plus simples pour faire de l'inférence.

Nous introduisons dans la partie 3.1.3.a les objets permettant de définir un processus Gamma, que l'on définit dans la partie 3.1.3.b. Le DPM est défini dans la partie 3.1.3.c.

Le lecteur plus intéressé par les aspects pratiques peut sauter les deux prochaines sous-parties et se rendre directement à la définition du processus de Dirichlet à mélange en 3.1.3.c.

3.1.3.a Mesures complètement aléatoires

On a vu dans la partie 3.1.2 comment générer des mesures aléatoires φ de la forme $\varphi = \sum_{k=1}^K a_k \delta_{\theta_k}$ à partir d'un nombre fini de variables aléatoires i.i.d. On souhaite désormais ne pas préciser à l'avance le nombre de classe, i.e., on veut que K soit également aléatoire.

L'approche naïve consistant à générer une infinité d'échantillons suivant une loi Gamma et renormaliser par la somme ne fonctionne pas puisqu'une telle somme diverge presque sûrement. La bonne approche pour généraliser la loi de Dirichlet passe par le concept de *mesures complètement aléatoires* (CRM, pour *complete random measure* en anglais).

Définition 3.1. Soit φ une mesure aléatoire définie sur un espace probabilisé (Θ, \mathcal{A}) . La mesure φ est dite complètement aléatoire, abrégée CRM, si pour tout entier n et tous sous-ensembles $A_1 \dots A_n$ deux à deux disjoints de Θ les variables aléatoires $\varphi(A_1) \dots \varphi(A_n)$ sont indépendantes.

En outre, si φ est une CRM, $A_1 \dots A_n$ une partition de Θ , alors $\varphi(\cup_{\ell} A_{\ell}) = \sum_{\ell} \varphi(A_{\ell})$ et les variables aléatoires $\varphi(A_{\ell})$ sont indépendantes. Sous certaines conditions sur \mathcal{A} , il est même possible pour toute valeur de n de trouver une partition de θ telles que les $\varphi(A_{\ell})$ sont identiquement distribuées. Ces mesures aléatoires génèrent donc des lois infiniment divisibles.

Gardons en tête la construction de la loi de Dirichlet : on veut générer des paramètres θ_k i.i.d. et des poids positifs i.i.d. selon une loi Gamma. Le bon objet pour construire une telle suite est un processus de Poisson homogène.

Un processus de Poisson sur l'espace probabilisé $(\Theta, \mathcal{A}, G_0)$ est une collection dénombrable de points de Θ . Dans notre cas, ces points sont les paramètres des classes et G_0 joue le rôle de loi *a priori* sur ces paramètres. On associe à cette collection de points une mesure de comptage, notée N . Si pour tout $A \in \mathcal{A}$ la loi marginale $N(A)$ est une loi de Poisson de paramètre $G_0(A)$, alors N est une mesure aléatoire de Poisson.

Bien que similaires, un processus ponctuel de Poisson et une mesure aléatoire de Poisson sont deux objets bien distincts. Le premier est une collection de points de Θ tandis que le second est une mesure sur (Θ, \mathcal{A}) .

Cet exemple montre que l'on peut se servir d'une *mesures complètement aléatoires (CRM)* pour définir un processus stochastique. La réciproque est fautive : tout processus stochastique ne définit pas une CRM.

Nous allons nous servir de la mesure aléatoire de Poisson pour générer des mesures aléatoires plus sophistiquées. Puisque l'on souhaite à la fois générer des paramètres et des masses, considérons dans ce cas une mesure aléatoire de Poisson N sur l'espace augmenté $\Theta \times \mathbb{R}^+$. On rappelle que \mathbb{R}^+ correspond à l'espace où vivent les masses non normalisées. Notons θ_k les paramètres et v_k les masses non renormalisées. Soit G_0 une mesure de probabilité sur Θ , et ν une mesure sur \mathbb{R}^+ appelée *mesure de diffusion*. On définit la *mesure de Lévy* comme la loi jointe sur $\Theta \times \mathbb{R}^+$, donnée par $\mu \triangleq G_0 \times \nu$. Comme le processus de Poisson sous-jacent est à accroissements indépendants, la mesure de Lévy suffit à caractériser le processus Gamma. (Khintchine, 1934; Lévy, 1937).

Si N est la mesure aléatoire associée à une réalisation d'un processus stochastique de mesure μ , alors N est de la forme

$$N = \sum_{k \in \mathbb{N}^*} \delta_{\theta_k, v_k}. \quad (3.9)$$

On définit à partir de N une nouvelle mesure aléatoire φ_N

$$\varphi_N \triangleq \sum_{k \in \mathbb{N}^*} v_k \delta_{\theta_k}. \quad (3.10)$$

Par construction, φ_N est une mesure aléatoire sur Θ , et est également une CRM. La figure 3.5 montre un tirage φ d'une mesure complètement aléatoire sous forme de bâtons. La position d'un bâton indique la valeur du paramètre associé, et sa taille est proportionnelle à la valeur du poids. On voit que les bâtons se concentrent autour de la région chargée par la mesure de base G_0 , et les poids associés sont distribués suivant ν . En l'état, la réalisation φ ne peut pas encore servir de mesure de mélange car les poids v_k ne somment pas à 1.

3.1.3.b Processus Gamma

Pour définir le processus stochastique qui va généraliser le vecteur de lois Gamma, nous allons choisir une mesure intensité telle que pour $A \subset \Theta$ et $I \subset \mathbb{R}^+$, $\varphi_V(A \times I)$ est une loi Gamma.

Définition 3.2. Soit α un réel positif, $G = \{(\theta_1, v_1), \dots\}$ un processus ponctuel de Poisson sur $\Theta \times [0, +\infty[$ de mesure d'intensité

$$\mu_G(d\theta, dv) = G_0(d\theta) \alpha v^{-1} e^{-\beta v} dv. \quad (3.11)$$

On appelle réalisation d'un processus Gamma, notée $\varphi_G \sim \Gamma P(\alpha, G_0)$, la mesure complètement aléatoire

$$\varphi_G \triangleq \sum_{k=1}^{+\infty} v_k \delta_{\theta_k}. \quad (3.12)$$

Pour finir, la mesure aléatoire de Poisson joue un rôle plus central dans l'analyse de CRM. Il existe un théorème de représentation qui stipule qu'une CRM sur Θ peut s'écrire comme la somme d'une mesure continue déterministe et d'une mesure aléatoire de Poisson (Kingman, 1967).

La mesure de Lévy est parfois appelée *compensateur* (compositor en anglais).

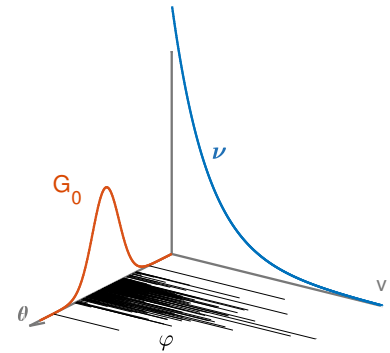


FIGURE 3.5 : Tirage d'une mesure aléatoire φ sur $\Theta \times \mathbb{R}^+$, de mesure de Lévy $G_0 \times \nu$. Les courbes orange et bleue représentent respectivement les mesures G_0 et ν . Les bâtons noirs représentent φ . La position des bâtons représente les paramètres, tandis que la longueur des bâtons est la masse associée.

Notons que la mesure d'intensité équation (3.11) ressemble à la densité d'une loi Gamma dégénérée. On peut montrer que pour tout $A \subset \Theta$ et $B \subset \mathbb{R}^+$, la quantité $\varphi_G(A \times B)$ est une loi Gamma de paramètres $\alpha G_0(A)$ et β , d'où le nom du processus. Pour finir, l'espérance du nombre de points d'une réalisation, donnée par

$$\mathbb{E} [N(\Theta \times [0, +\infty])] = \mu_G(\Theta \times [0, +\infty]) = G_0(\Theta) \int_0^{+\infty} \alpha v^{-1} e^{-\beta v} \mathrm{d}v = +\infty, \quad (3.13)$$

est infinie. En d'autres termes, la mesure aléatoire construite à partir d'une réalisation d'un processus Gamma contient presque sûrement une infinité dénombrable de points, ou paramètres ici.

Pour démontrer ce résultat, il faut comparer les transformées de Laplace de $\varphi(A \times B)$ et d'une loi Gamma (Applebaum, 2004).

3.1.3.c Le processus de Dirichlet à mélange

Nous allons construire le processus de Dirichlet en tant que processus Gamma renormalisé. Considérons un processus Gamma $\varphi_G \sim \Gamma P(\alpha, G_0)$. Construisons une nouvelle mesure φ_D sur Θ à partir de φ_G , définie par ses marginales : pour tout entier K , toute partition A_1, \dots, A_K de Θ , on définit le vecteur aléatoire $[\varphi_D(A_1), \dots, \varphi_D(A_K)]$ par

$$\forall k, \quad \varphi_D(A_k) \triangleq \frac{\varphi_G(A_k)}{\sum_{k=1}^K \varphi_G(A_k)}. \quad (3.14)$$

Comme pour tout k , $\varphi_G(A_k)$ est distribué suivant une loi Gamma de paramètres $(\alpha G_0(A_k), \beta)$, le vecteur aléatoire $[\varphi_D(A_1), \dots, \varphi_D(A_K)]$ suit une loi de Dirichlet de paramètres $\varphi_G(A_1), \dots, \varphi_G(A_K)$. Le processus ainsi défini est le processus de Dirichlet.

Théorème 3.1. (Ferguson, 1973). Soit α un réel positif. Soit (Θ, \mathcal{A}) un espace mesurable et G_0 une mesure de probabilité sur (Θ, \mathcal{A}) . On dit qu'une mesure aléatoire φ_D est distribuée selon un processus de Dirichlet de distribution de base G_0 et de facteur d'échelle α si pour toute partition mesurable (A_1, \dots, A_K) de Θ en K éléments, le vecteur aléatoire $[\varphi_D(A_1), \dots, \varphi_D(A_K)]$ vérifie

$$\varphi_D(A_1), \dots, \varphi_D(A_K) \sim \text{Dir}(\alpha G_0(A_1), \dots, \alpha G_0(A_K)). \quad (3.15)$$

Ce processus est noté $\varphi_D \sim DP(\alpha, G_0)$.

Les vecteurs aléatoires formés par les mesures des partitions sont appelés lois fini-dimensionnelles. L'existence du processus renormalisé n'est pas trivial pour des raisons de consistance vis-à-vis de toutes les partitions de Θ . Ici, c'est le théorème de *consistance de Kolmogorov* qui assure l'existence (Feller, 1968, Partie 4.6). Ce théorème donne des conditions suffisantes d'existence de processus définis uniquement par leurs lois fini-dimensionnelles.

En bref, un processus de Dirichlet est un processus stochastique dont les marginales sont des lois de Dirichlet. Comme la loi de Dirichlet, le paramètre de forme β du processus Gamma n'a aucune influence. Notons qu'en tant que processus Gamma renormalisé, les variables aléatoires

$\varphi_D(A_1), \dots, \varphi_D(A_K)$ ne sont pas indépendantes. Le processus de Dirichlet n'est donc pas une CRM.

Le processus dépend de deux paramètres. Le premier est un paramètre d'échelle, noté α et contrôle la dispersion des poids π_k . Le second est la mesure de base G_0 qui joue le rôle de loi *a priori* sur les paramètres θ . La figure 3.6 montre deux réalisations du processus de Dirichlet pour deux valeurs de α . Nous verrons dans la prochaine partie comment générer ces réalisations. On constate que lorsque α est petit, la mesure aléatoire concentre sa masse sur un petit nombre de poids. À l'inverse, lorsque α grandit, la mesure aléatoire a tendance à s'étaler.

Nous allons passer en revue deux propriétés du processus de Dirichlet. D'abord, les moments s'expriment facilement en fonction des moments de la loi de Dirichlet : soient A un élément de \mathcal{A} , A^c son complémentaire dans Θ et $\varphi_D \sim DP(\alpha, G_0)$ un tirage d'un processus de Dirichlet. On obtient, en considérant le vecteur aléatoire $[\varphi_D(A), \varphi_D(A^c)]$

$$\mathbb{E}[\varphi_D(A)] = G_0(A) \quad (3.16)$$

$$\text{Var}[\varphi_D(A)] = \frac{G_0(A)(1 - G_0(A))}{1 + \alpha}. \quad (3.17)$$

Sans parler de modèle de mélange, nous allons voir comment l'observation de paramètres tirés suivant une réalisation φ_D d'un processus de Dirichlet permet de caractériser la réalisation. En d'autres mots, nous allons étudier le processus *a posteriori*. Dans ce but, considérons le modèle suivant, pour N un entier positif

$$\begin{aligned} \varphi_D &\sim DP(\alpha, G_0) \\ \forall n \in \llbracket 1, N \rrbracket, \quad \theta_n &\stackrel{\text{i.i.d.}}{\sim} \varphi_D. \end{aligned} \quad (\mathcal{M}_D)$$

Ici, on a d'abord tiré une mesure aléatoire suivant un processus de Dirichlet, puis N paramètres suivant cette mesure. Suivant ce modèle, la mesure aléatoire φ_D n'est pas directement observée ; seuls les paramètres $\theta_1, \dots, \theta_N$ qui coïncident presque sûrement avec des atomes de φ_D le sont. Nous allons voir comment l'observation de ces N paramètres permet d'améliorer la connaissance sur φ_D . On s'intéresse dans ce but à la loi conditionnelle $\varphi_D | \theta_1, \dots, \theta_N$.

Soit (A_1, \dots, A_K) une partition de Θ . Comme, (A_1, \dots, A_K) est une partition, pour tout entier positif n , le vecteur aléatoire $(\delta_{\theta_n}(A_1), \dots, \delta_{\theta_n}(A_K))$ est composé d'un seul 1 et de $K - 1$ zéros. La distribution associée est donc une loi multinomiale de paramètres $(G_0(A_1), \dots, G_0(A_K))$. Or, la loi multinomiale est conjuguée avec la loi de Dirichlet. D'où, en exploitant l'indépendance entre les θ_n et la conjugaison entre la loi multinomiale et la loi de Dirichlet

$$\begin{aligned} \varphi_D(A_1), \dots, \varphi_D(A_K) &\left| \left\{ \delta_{\theta_n}(A_1), \dots, \delta_{\theta_n}(A_K), \forall n = 1 \dots N \right\} \right. \\ &\sim \mathcal{D} \left(\alpha G_0(A_1) + \sum_{n=1}^N \delta_{\theta_n}(A_1), \dots, \alpha G_0(A_K) + \sum_{n=1}^N \delta_{\theta_n}(A_K) \right). \end{aligned} \quad (3.18)$$

Le résultat (3.18) est valable pour toutes les partitions de Θ . On a donc

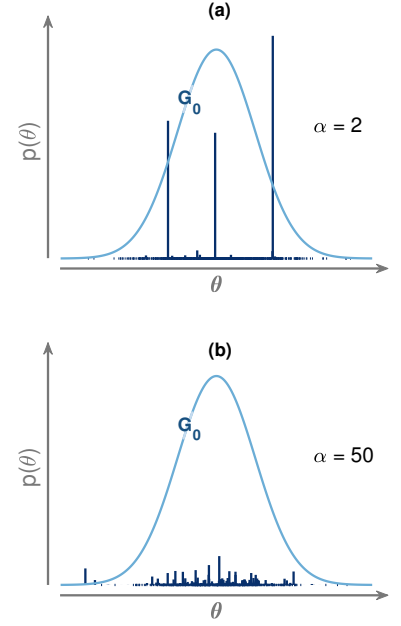


FIGURE 3.6 : Deux réalisations d'un processus de Dirichlet avec $\Theta = \mathbb{R}$, G_0 est la loi normale et (a) $\alpha = 2$ et (b) $\alpha = 50$. La position des barres représente la valeur du paramètre et la hauteur le poids π_k associé.

construit un nouveau processus de Dirichlet

$$\varphi_D | \theta_1, \dots, \theta_N \sim DP \left(\alpha, G_0 + \sum_{n=1}^N \delta_{\theta_n} \right). \quad (3.19)$$

Comme on a présenté le processus de Dirichlet en utilisant une mesure de probabilité comme mesure de base, on présente plutôt ce résultat selon

$$\varphi_D | \theta_1, \dots, \theta_N \sim DP \left(\alpha + N, \frac{\alpha}{N + \alpha} G_0 + \frac{1}{N + \alpha} \sum_{n=1}^N \delta_{\theta_n} \right). \quad (3.20)$$

Ainsi, la loi *a posteriori* d'un processus de Dirichlet est encore un processus de Dirichlet. Cette propriété est fondamentale pour l'inférence. On y retrouve le compromis bayésien classique entre le terme *a priori* G_0 et les observations θ_n .

Cette loi présente également un effet d'agrégation : si un paramètre θ a déjà été tiré, alors il a plus de chance d'être tiré à nouveau. Cet effet d'agrégation est une propriété recherchée dans le cadre de la classification non supervisée.

3.1.4 Manipulation du processus de Dirichlet à mélange

Nous avons construit dans la partie précédente le processus de Dirichlet. Il est possible de manipuler directement la loi *a posteriori* d'une CRM renormalisée. Mais nous allons voir dans cette partie qu'il est possible de déduire des algorithmes simples pour l'inférence.

3.1.4.a Procédure *Stick-breaking*

Le processus de Dirichlet a été construit en renormalisant des processus Gamma. Toutefois, cette construction ne fournit pas de moyens de générer des réalisations du processus de Dirichlet, à savoir les mesures aléatoires φ_D . Nous allons exposer dans cette partie une procédure pour générer de telles mesures aléatoires.

On a vu que qu'une réalisation d'un processus de Dirichlet contient presque sûrement une infinité dénombrable de paramètres θ_k auxquels sont associés des poids π_k . Sethuraman (1994) propose une construction de la suite des poids $(\pi_k)_k$, élément du simplexe Δ qui se comprend facilement à travers une métaphore consistant à subdiviser itérativement un bâton. Cette procédure est appelée *stick breaking* en anglais. L'intuition derrière ce procédé est que les marginales d'une loi de Dirichlet sont des lois Beta.

Considérons un bâton de longueur 1. On prélève un premier morceau de longueur V_1 suivant une loi Beta² de paramètres $(1, \alpha)$ à valeurs dans $[0, 1]$, et l'on retire une longueur $a_1 = V_1$. Le bâton est maintenant de taille $1 - a_1$. On tire une nouvelle fraction $V_2 \sim B(1, \alpha)$. On casse un deuxième morceau de la longueur restante $a_2 = V_2(1 - V_1)$. On répète ce procédé une infinité dénombrable de fois et l'on obtient une suite α élément du simplexe Δ . Si l'on associe à chaque poids un paramètre

Il existe un procédé similaire pour simuler une loi de Dirichlet que nous n'avons pas évoquée.

2. La loi Beta est une famille de lois de probabilité définie sur $[0, 1]$ paramétrée par deux réels positifs a et b . Sa densité évaluée en x , élément de $[0, 1]$ est

$$p(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1} (1-x)^{b-1}.$$

La loi Beta coïncide avec la loi de Dirichlet à deux éléments de paramètres (a, b) .

θ tiré selon G_0 , alors on génère une mesure de probabilité aléatoire φ .
Formellement

$$\forall k \in \mathbb{N}^*, \quad \begin{cases} V_k & \stackrel{\text{i.i.d.}}{\sim} \mathcal{B}(1, \alpha) \\ a_k & = V_k \prod_{l=1}^{k-1} (1 - V_l), \\ \theta_k & \sim G_0 \end{cases} \quad (\text{GEM})$$

et $\varphi = \sum_{k=1}^{\infty} a_k \delta_{\theta_k}$.

Alors φ est une réalisation d'un processus de Dirichlet (Sethuraman, 1994). Notons qu'ainsi défini, on a bien $\sum_{k=1}^{+\infty} a_k = 1$. La distribution sur la suite des poids \mathbf{a} est parfois notée GEM^3 (Pitman, 2006).

3. Les lettres **GEM** sont pour Griffiths, Engen and McCloskey.

Il est numériquement impossible de générer une infinité dénombrable de coefficients. En pratique, on considère une version tronquée du **GEM**, en choisissant le nombre de coefficients K_{\max} suffisamment grand pour la somme des K_{\max} coefficients a_k générés soit proche de 1. Les deux réalisations montrées à la figure 3.6 ont été générées en utilisant la procédure *stick breaking* avec 1000 coefficients. Dans ce cas, nous avons obtenu $|1 - \sum_{k=1}^{1000} a_k| \simeq 2 \times 10^{-16}$.

3.1.4.b Loi marginale des paramètres et urne de Blackwell-MacQueen

La procédure *stick breaking*, ou **GEM**, fournit un premier moyen de générer des réalisations d'un processus de Dirichlet et faire de l'inférence dans le cadre d'un modèle de mélange. Bien qu'efficace numériquement, le **GEM** requiert de tronquer la mesure aléatoire φ_D . En un sens, cela revient à procéder à une approximation paramétrique. Nous allons présenter dans cette partie une seconde procédure à base d'un modèle d'urne qui permet de générer des paramètres sans expliciter la mesure aléatoire φ_D et donc sans procéder à une approximation paramétrique.

Pour des raisons pratiques d'inférence, on souhaite obtenir la loi marginale des paramètres θ_n . Conditionnés à la réalisation de la mesure aléatoire φ_D , les paramètres θ_n sont **i.i.d.**. Cependant, on ne peut rien affirmer concernant la loi marginale. Pour comprendre le comportement marginal des paramètres, il faut introduire le concept de *variables échangeables*.

Définition 3.3. Une suite de variables aléatoires $(x_n)_n$ est dite *échangeable* si la distribution de la suite est invariante par permutation des indices, i.e., si pour tout entier N et toute permutation σ de $\llbracket 1, N \rrbracket$

$$x_1, \dots, x_N \stackrel{\text{Loi}}{=} x_{\sigma(1)}, \dots, x_{\sigma(N)}. \quad (3.21)$$

L'échangeabilité est une version affaiblie du concept de variables **i.i.d.**. L'ordre n'a pas d'importance mais les variables ne sont pas indépendantes.

Un théorème de représentation des variables échangeables a été prouvé par de Finetti dans le cadre de variables binaires (de Finetti, 1931). Il stipule qu’une suite de variables aléatoires binaires $(x_n)_n$ est échangeable si et seulement si il existe une variable cachée w telle que les variables aléatoires $x_n|w$ sont *i.i.d.*. Ce théorème a été depuis généralisé.

Théorème 3.2. *Une suite de variable aléatoire $(x_n)_n$ est échangeable si et seulement si il existe une mesure aléatoire φ telle que pour tout entier N , tous boréliens A_1, \dots, A_N , la loi jointe s’écrit*

$$p(x_1 \in A_1, \dots, x_N \in A_N) = \int \left(\prod_{n=1}^N \varphi(A_n) \right) \mathbb{P}(d\varphi). \quad (3.22)$$

La mesure φ , non explicite, est appelée la mesure de de Finetti.

Ce théorème relie les notions d’échangeabilité et d’indépendance. Conditionnée à la bonne mesure aléatoire, une suite de variables aléatoires échangeables est indépendante.

La suite de paramètres $(\theta_n)_n$ définie au modèle (\mathcal{M}_D) est *i.i.d.* conditionnée au tirage du processus φ_D . La loi marginale des paramètres est donc échangeable. La loi $\theta_{N+1}|\theta_1 \dots \theta_N$, appelée *loi prédictive*, s’obtient comme suit. Soit A un borélien de Θ . Alors

$$\begin{aligned} p(\theta_{N+1} \in A|\theta_1, \dots, \theta_N) &= \mathbb{E}_{\theta_{N+1}} \left[\mathbb{1}_A|\theta_1, \dots, \theta_N \right] \\ &= \mathbb{E}_{\varphi_D} \left[\mathbb{E}_{\theta_{N+1}} \left[\mathbb{1}_A|\varphi_D, \theta_1, \dots, \theta_N \right]|\theta_1, \dots, \theta_N \right] \\ &= \mathbb{E}_{\varphi_D} \left[p(\theta_{N+1} \in A|\varphi_D)|\theta_1, \dots, \theta_N \right] \\ &= \frac{\alpha}{\alpha + N} G_0(A) + \sum_{n=1}^N \frac{1}{N + \alpha} \delta_{\theta_n}(A). \end{aligned} \quad (3.23)$$

On peut se servir de (3.23) pour générer des paramètres $\theta_1, \dots, \theta_N$ échangeables dont la mesure de de Finetti est le processus de Dirichlet. Cette représentation, que nous allons décrire à travers la métaphore d’une urne remplie de boules de couleurs, est appelée *urne de Blackwell-MacQueen* (Blackwell et MacQueen, 1973).

Dans cette métaphore, Θ est l’espace des couleurs, éventuellement continu, et G_0 une mesure de probabilité sur les couleurs. On considère une urne initialement vide, que l’on va itérativement remplir. À la première itération, on tire une couleur $\theta_1 \sim G_0$. À l’itération $N + 1$ on tire une nouvelle boule suivant (3.23). En pratique, si l’on prend $A = \{\theta_\ell\}$ pour $\ell \leq N$ dans (3.23), la probabilité qu’une couleur déjà existante soit tirée est

$$p(\theta_{N+1} \in \{\theta_\ell\}|\theta_1 \dots \theta_N) = \frac{\alpha}{N + \alpha} G_0(\{\theta_\ell\}) + \sum_{n=1}^N \frac{1}{\alpha + N} \delta_{\theta_n}(\{\theta_\ell\}). \quad (3.24)$$

Notons que comme $\{\theta_\ell\}$ est un singleton, $G_0(\{\theta_\ell\})$ est nulle si G_0 est continue.

De Finetti a de plus montré que lorsque les variables sont binaires, la mesure invariante est la loi Beta. Notons l’importance de ce résultat : en modélisation bayésienne, on représente l’incertitude sur le paramètre par de l’aléatoire. Le théorème nous assure qu’ici, cette distribution sur le paramètre existe !

Dans (3.22), l’intégrale porte sur la mesure aléatoire non explicite.

On rappelle

$$\begin{aligned} \varphi_D &\sim DP(\alpha, G_0) \\ \theta_n|\varphi_D &\stackrel{i.i.d.}{\sim} \varphi_D. \end{aligned} \quad (\mathcal{M}_D)$$

par définition.

formule dite *Tower formula*.

par indépendance, cf. Def. \mathcal{M}_D .

En combinant (3.16) et (3.20).

L’urne de Blackwell-MacQueen est conçue comme une généralisation de l’urne de Pólya (Johnson et Kotz, 1977). Le fonctionnement de l’urne de Pólya est similaire mais il n’y a qu’un nombre fini de couleur.

La loi (3.24) est exactement la loi conditionnelle d'une nouvelle couleur conditionnellement aux autres. Comme la loi sur les couleurs est échangeable, on a en réalité obtenue la loi de n'importe quel tirage de couleurs conditionnellement aux autres. Si l'on revient au problème des modèles de mélange non paramétriques, on dispose donc de tous les ingrédients pour mettre en place un échantillonneur de Gibbs sur les paramètres de chaque classe. Toutefois, cet algorithme échantillonne les paramètres un par un, ce qui aboutit à un mélange très lent⁴ de la chaîne. Nous verrons dans la prochaine partie un nouvel algorithme travaillant directement sur les partitions.

On retrouve dans (3.24) l'effet d'agrégation : si une couleur θ_ℓ a été tirée n_ℓ fois, alors la probabilité qu'elle soit à nouveau tirée est $n_\ell / (\alpha + N)$. Il y a donc une probabilité non nulle pour que les $\theta_1, \dots, \theta_N$ prennent des valeurs identiques. La valeurs des paramètres $\theta_1, \dots, \theta_N$ définit ainsi implicitement une *partition* de $\llbracket 1, N \rrbracket$, dont la taille est le nombre de valeurs distinctes de $\theta_1, \dots, \theta_N$. C'est cette partition qui nous intéresse dans le cadre de la classification non supervisée. Cette partition est aléatoire, et nous nous intéresserons à ses propriétés dans la prochaine partie.

4. Pour le comprendre, considérons le cas suivant. Imaginons que ℓ paramètres formant une classe sont dans un état θ . Pour obtenir la même partition mais pour une valeur θ' , il faut que chacun des ℓ paramètres choisissent un à un la valeur θ' , ce qui peut prendre plusieurs itérations.

3.1.4.c Loi sur les partitions et processus du restaurant chinois

Cette partie introduit le *processus du restaurant chinois* (CRP), pour *chinese restaurant process* en anglais, à partir de l'urne de Blackwell-MacQueen. Le CRP est étroitement lié à la loi des permutations d'un gène, nommée *loi d'Ewens* (Ewens, 2004). L'objectif est de comprendre la distribution sur les partitions induite par le processus de Dirichlet, et de construire un nouvel algorithme travaillant directement sur ces partitions.

Soient $\theta_1, \dots, \theta_N$ N paramètres générés suivant un modèle d'urne de Blackwell-MacQueen de paramètre α . On s'intéresse aux sous ensembles formés par les paramètres de même valeur. En regroupant les indices des paramètres de même valeurs, on obtient une partition π_N de $\llbracket 1, N \rrbracket$. Notons \mathcal{P}_N l'ensemble des partitions de $\llbracket 1, N \rrbracket$. D'après l'équation (3.24), la partition π_{N+1} s'obtient en insérant l'entier $N + 1$ à l'un des éléments de la partition proportionnellement à sa cardinalité, ou en créant un nouveau sous-ensemble formé uniquement de $N + 1$ avec probabilité proportionnelle à α .

La suite $(\pi_n)_n$ ainsi définie est une chaîne de Markov inhomogène d'espace d'état $\bigcup_{n \geq 1} \mathcal{P}_n$, où chaque π_n est à valeur dans \mathcal{P}_n . On remarque immédiatement que $\pi_1 = \{1\}$. En notant b les éléments ou bloc d'une partition, le noyau de transition de la chaîne est donné pour tout couple $(\pi, \pi') \in \mathcal{P}_n \times \mathcal{P}_{n+1}$, par

$$\mathbb{P}[\pi_{n+1} = \pi' | \pi_n = \pi] = \begin{cases} \frac{\#b}{n+\alpha} & \text{si } \pi' \text{ s'obtient en rajoutant } n+1 \text{ dans le bloc } b \text{ pour tout } b \in \pi \\ \frac{\alpha}{n+\alpha} & \text{si } \pi' \text{ s'obtient en rajoutant le singleton } \{n+1\} \text{ à la partition } \pi, \\ 0 & \text{sinon,} \end{cases} \quad (3.25)$$

où $\#b$ est la taille du bloc b . Une réalisation du CRP à N clients est notée $\pi_N \sim CRP(\alpha, N)$.

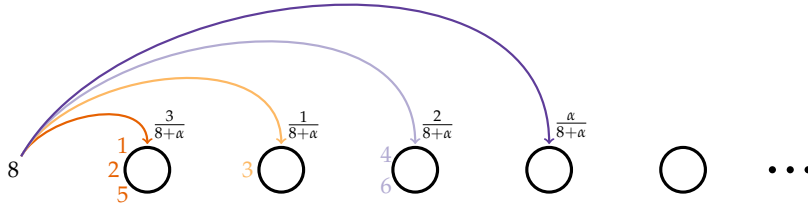


FIGURE 3.7 : Génération d’une réalisation du CRP pour $N = 8$. Ici, 7 clients sont déjà répartis sur 3 tables. Le 8^e client peut s’asseoir à l’une des trois tables déjà occupées ou en dresser une nouvelle.

Le processus stochastique défini équation (3.25) est appelé le CRP en raison de la métaphore culinaire associée.

La figure 3.7 illustre cette métaphore. L’espace d’état est un restaurant, les ensembles de la partition des tables et les indices des clients. À chaque itération, le nouveau client s’assoit soit à une table existante proportionnellement au nombre de client assis soit à une nouvelle table proportionnellement à α . Nous garderons dans cette partie ce vocabulaire pour étudier les propriétés du processus.

On voit déjà à ce stade l’effet du paramètre α . Si $\alpha = 0$, aucune nouvelle table n’est ajoutée. À l’inverse, si $\alpha \rightarrow +\infty$, π_n est réduit à la partition triviale $\{\{1\}, \{2\}, \dots\}$.

Dans le cadre pratique de la classification non supervisée, les observations, *i.e.* les clients ici, n’arrivent pas séquentiellement. On préfère alors considérer la loi marginale sur la partition $P[\pi_n]$.

Théorème 3.3. Pour tout entier n et toute partition π de \mathcal{P}_n , on a

$$P[\pi_n = \pi] = \frac{\alpha^{|\pi|}}{\alpha(\alpha + 1) \dots (\alpha + n - 1)} \prod_{b \in \pi} (\#b - 1)! \quad , \quad (3.26)$$

où $|\pi|$ est le nombre d’éléments de la partition.

La loi (3.26) est aussi appelée *fonction de probabilité de la partition échangeable (EPDF)*. La loi (3.26) est invariante vis-à-vis des permutations σ sur l’arrivée des clients, d’où l’adjectif « échangeable ». On peut comprendre la forme du résultat (3.26) sans complètement développer la preuve : le dénominateur est le produit des constantes de normalisation successives en $(\alpha + n - 1)^{-1}$. Le terme en $\alpha^{|\pi|}$ provient des $|\pi|$ cas où l’on a créé une nouvelle table tandis que le terme en $(\#b - 1)!$ est le produit des tailles successives des tables où les clients vont se rajouter.

D’un point de vue algorithmique, l’échangeabilité permet la mise en place d’un algorithme de Gibbs, comme dans le cas de l’urne de Blackwell-MacQueen. La différence⁵ est que l’on travaille ici directement sur la partition et non plus sur la partition induite par les valeurs des paramètres. Un tel algorithme sera présenté dans la partie 3.1.5.

La preuve se fait par récurrence, et pourra être trouvée au chapitre 14 de Chafaï et Malrieu (2015).

5. Reprenons l’exemple de la note 4 p. 90. Désormais, pour faire le même mouvement, il suffit de mettre à jour une seule fois la valeur du paramètre.

Notons que pour $n \geq 3$, l'EPPF n'est jamais la loi uniforme sur les partitions. En effet, il est facile⁶ de trouver deux partitions π et π' de \mathcal{P}_n telles que $P[\pi] \neq P[\pi']$.

Il est possible de déterminer la loi du nombre de tables, *i.e.*, la loi de la taille de la partition, ainsi que les moments.

Théorème 3.4. Soit $(\pi_n)_n$ un tirage d'un processus du restaurant chinois de paramètre α . Notons K_N le nombre de tables après l'arrivée de N clients. Alors

$$P[K_N = K] = s(N, K) \frac{\alpha^K}{\alpha(\alpha + 1) \dots (\alpha + N - 1)}, \quad (3.27)$$

$$\mathbb{E}[K_N] = \sum_{n=1}^N \frac{\alpha}{\alpha + n} \underset{N \rightarrow +\infty}{\sim} \alpha \ln(N), \quad (3.28)$$

$$\text{Var}[K_N] = \sum_{n=1}^N \frac{\alpha n}{(\alpha + n)^2} \underset{N \rightarrow +\infty}{\sim} \alpha \ln(N), \quad (3.29)$$

où $s(N, K)$ est le nombre de Stirling de première espèce, qui compte le nombre de permutations de N objets composés d'exactly K cycles dis-joints.

La figure 3.8 illustre la loi du nombre de tables (*cf.* (3.27)) appelée *équation d'Antoniak* (Antoniak, 1974). Les équations (3.28) et (3.29) montrent que les deux premiers moments de la loi du nombre de tables se comportent asymptotiquement comme ceux d'une loi de Poisson. L'espérance du nombre de table est proportionnel à $\ln(N)$. Si l'on choisi le restaurant chinois comme loi *a priori* sur les partitions pour une tâche de classification non supervisée, on obtient un modèle qui pénalise le nombre de classe tout en favorisant la diversité lorsque le nombre d'observations augmente.

Ces observations concluent la construction du modèle de mélange bayésien non paramétrique que nous allons utiliser dans la prochaine partie pour une formulation bayésienne non paramétrique de la classification non supervisée. Nous retiendrons qu'en partant d'un objet mathématique complexe, nous avons abouti au CRP, *une loi relativement simple et interprétable* sur l'ensemble des partitions.

3.1.5 Inférence et classification non supervisée

Nous avons étudié l'objet généralisant les modèles de mélanges paramétriques. Ce travail a nécessité un investissement mathématique mais nous allons voir dans cet exemple comment utiliser le CRP dans une formulation non paramétrique du problème de classification non supervisée.

3.1.5.a Modèle et estimateurs

Reprenons le modèle présenté dans l'exemple introductif partie 3.1.2. Chaque classe est décrite par une loi normale multivariée, de vecteur moyenne μ et de matrice de covariance Σ . Le paramètre θ fait toujours référence au couple (μ, Σ) . On choisi une loi *a priori* Normale-Wishart

6. Par exemple si $n = 3$ et $\alpha \neq \sqrt{2}$, on peut considérer $\pi = \{\{1, 2, 3\}\}$ et $\pi' = \{\{1\}, \{2\}, \{3\}\}$, on a $P[\pi]/P[\pi'] = 2/\alpha^2$.

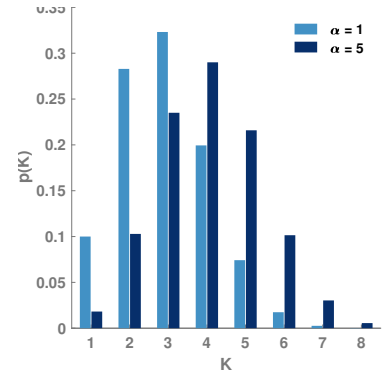


FIGURE 3.8 : Distribution théorique du nombre de tables d'un processus du restaurant chinois après l'arrivée de 10 clients, pour $\alpha = 1$ (bleu clair) et $\alpha = 5$ (bleu foncé). Ici, $\mathbb{E}[K_{10} | \alpha = 1] = 2.9$ et $\mathbb{E}[K_{10} | \alpha = 5] = 5.8$.

comme loi *a priori* sur les paramètres θ . Cette fois-ci, on spécifie directement la loi *a priori* sur les partitions de $[1, \dots, N]$: on choisit le processus du restaurant chinois. Le modèle s'écrit

$$\begin{aligned} \pi &\sim CRP(\alpha, N) \\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k | \pi &\stackrel{\text{i.i.d.}}{\sim} \mathcal{NW}, \quad k = 1 \dots |\pi| \\ \mathbf{y}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi &\sim \mathcal{N}(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \end{aligned} \quad (\mathcal{M})$$

Le modèle \mathcal{M} est bien non paramétrique puisque le nombre de classes, noté K , n'est pas fixé à l'avance et dépend de la partition π . On a vu au théorème 3.4 que K croît proportionnellement au logarithme du nombre d'observations.

On ne s'intéresse ici qu'à un estimateur de la partition des données, construit à partir de la loi marginale *a posteriori*

$$P[\pi | \mathbf{y}_1, \dots, \mathbf{y}_N] = \int_{\Theta^{|\pi|}} p(\pi, \theta | \mathbf{y}_1, \dots, \mathbf{y}_N) d\theta. \quad (3.30)$$

Nous proposons par exemple l'estimateur du *maximum a posteriori marginalisé* (MAPm), donné par

$$\hat{\pi}_{\text{MAP}} = \arg \max_{\pi} P[\pi | \mathbf{y}_1, \dots, \mathbf{y}_N]. \quad (3.31)$$

Bien que la loi marginale *a posteriori* $P[\pi | \mathbf{y}_1, \dots, \mathbf{y}_N]$ possède dans cet exemple une expression explicite, déterminer son maximum revient à tester toutes les partitions⁷ possibles. Cette méthode est trop coûteuse pour être envisagée.

Nous utilisons plutôt les méthodes MCMC pour explorer l'espace des solutions en simulant suivant (3.30) pour approcher numériquement les estimateurs (Maceachern, 1994). On dispose grâce à la partie 3.1.3 de tous les outils pour mettre en place un échantillonneur de Gibbs.

Ici, l'algorithme de Gibbs consiste à ré-allouer successivement et dans un ordre aléatoire les éléments de la partition. Soit n l'indice de l'observation à ré-allouer, π la partition courante et $\pi_{\setminus n}$ la partition π où l'on a retiré l'élément n . Par souci de simplification, notons \mathcal{C} une composante de la partition et $\mathbf{y}_{\mathcal{C}}$ la portion des observations dont les indices sont les éléments de \mathcal{C} . Il y a $|\pi_{\setminus n}| + 1$ possibilités : ajouter n à l'une des $|\pi_{\setminus n}|$ classes existantes ou créer une nouvelle classe, le singleton $\{n\}$. La ré-allocation se fait selon la loi avec probabilité

$$p(\pi' | \mathbf{y}) \propto \begin{cases} |\mathcal{C}| \frac{f(\mathbf{y}_{\mathcal{C} \cup \{n\}})}{f(\mathbf{y}_{\mathcal{C}})} & \text{si } \pi' = (\pi_{\setminus n} \setminus \mathcal{C}) \cup (\mathcal{C} \cup \{n\}) \\ \alpha f(\mathbf{y}_n) & \text{si } \pi' = \pi_{\setminus n} \cup \{n\}. \end{cases} \quad (3.32)$$

L'équation (3.32) s'appuie sur l'EPPF du restaurant chinois. Rappelons que c'est la propriété d'échangeabilité de l'EPPF permet de traiter une observation \mathbf{y} indépendamment de son indice.

La figure 3.9 montre l'approximation avec 5000 itérations de l'estimateur MAPm donné équation (3.31) du problème de classification non

7. Le nombre de partitions d'un ensemble à N éléments observables est le *nombre de Bell*.

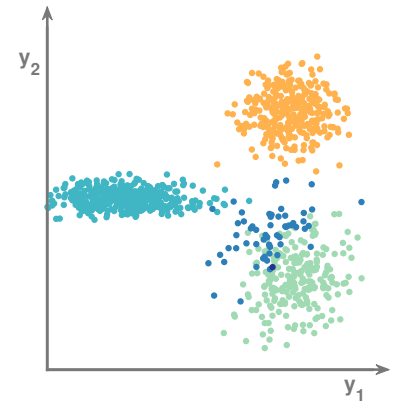


FIGURE 3.9 : Estimateur MAPm de la partition des données selon le modèle \mathcal{M} , approché sur 5000 itérations. Chaque couleur représente une classe.

supervisée appliquées aux données de la figure 3.2. Les données sont statistiquement bien expliquées au sens du modèle \mathcal{M} par cinq classes. On constate que les tailles de chaque classe sont différentes, allant de 500 points à un seul. Cette observation illustre la tendance de ce modèle à maintenir quelques tables peu occupées, *i.e.*, des classes avec peu de points ici. De plus, on observe que les points associés à des classes peu représentées sont tous situés dans une région d’ambiguïté située entre des classes plus importantes (points en bleu foncé).

3.1.5.b Consistance

De nombreux travaux sont toujours consacrés à l’étude du comportement asymptotique de modèles bayésiens. Par exemple, le théorème de Bernstein-von Mises (Cam et Yang, 2000) donne des conditions suffisantes sur un modèle bayésien pour assurer un théorème centrale limite de la loi *a posteriori* des paramètres. Si θ représente les paramètres et θ_0 la valeur ayant servi à la synthèse que l’on cherche à retrouver, alors, sous certaines conditions non explicitées ici, la loi *a posteriori* $\sqrt{N}(\theta - \hat{\theta}) | \mathbf{y}_{1:N}$ est asymptotiquement proche d’une loi normale. Cependant, Freedman (1965) a prouvé que ce résultat n’est presque sûrement plus valide lorsque le paramètre θ vit dans un espace de probabilité infini dénombrable, comme c’est le cas pour les modèles bayésiens non paramétriques.

Ce résultat négatif a mené à formuler différemment les résultats asymptotiques, notamment à travers la notion de consistance.

Définition 3.4. *Considérons un modèle bayésien $(P_\theta)_\theta$ paramétré par θ à valeur dans Θ , de dimension potentiellement infinie. Soit $\mathbf{y}_{1:N}$ N observations. La loi *a posteriori* $\theta | \mathbf{y}_{1:N}$ notée μ_N est consistante en θ_0 élément de Θ si pour tout voisinage \mathcal{V} de θ_0 , la quantité $\mu_N(\mathcal{V})$ converge P_{θ_0} -presque sûrement vers 1.*

*Par extension, l’estimateur MAP de θ est dit consistant si la loi *a posteriori* $p(\theta | \mathbf{y}_{1:N})$ l’est.*

Notons que l’on ne suppose pas dans la définition 3.4 l’existence d’une vraie valeur du paramètre. Ce point de vue est dit *subjectiviste*, et s’oppose à celui dit *classique* développé dans le paragraphe précédent.

La consistance des processus de Dirichlet à mélange a déjà été étudiée, voire par exemple Ghosal (2009) pour une revue des résultats. Ces résultats donnent des conditions suffisantes sur les lois *a priori* du modèle pour obtenir la consistance. Par exemple, Ghosal et al. (1999) montre la consistance lorsque le noyau est Gaussien et la loi sur les hyperparamètres est conjuguée. Bien que ces résultats tendent à conforter l’utilisation des méthodes non paramétriques, la consistance ne s’étend pas forcément aux lois marginales *a posteriori*. Puisque le nombre de classe est également une variable aléatoire, il est naturel de se demander s’il est possible de construire des estimateurs efficaces du nombre de classes.

Par exemple, McCullagh et Yang (2008) alertait déjà sur la difficulté d’un processus de Dirichlet à mélange de séparer et identifier deux classes.

Dans le modèle \mathcal{M} , la loi sur les partitions induit une loi sur le nombre de classes, que l'on note K_N où N indique le nombre d'observations. La loi *a posteriori* marginale $P[K_N | \mathbf{y}_1, \dots, \mathbf{y}_N]$ est approchée avec le même échantillonneur de Gibbs décrit équation (3.32), et l'on définit l'estimateur du *maximum a posteriori marginalisé* (MAPm)

$$\hat{K}_{\text{MAPm}} = \arg \max_{K \in \mathbb{N}} P[K_N = K | \mathbf{y}_1, \dots, \mathbf{y}_N]. \quad (3.33)$$

Alors, Miller et Harrison (2014) ont montré le résultat négatif suivant

Théorème 3.5. (Miller et Harrison, 2014, Théorème 6) Soit $\mathbf{y}_1, \dots, \mathbf{y}_N$ N observations aléatoires. Alors pour tout entier k , le résultat suivant est vrai

$$\limsup_{N \rightarrow +\infty} P[K_N = k] < 1. \quad (3.34)$$

La preuve repose sur la propension du restaurant chinois à proposer et à maintenir de petites tables, déjà évoquée dans la partie 3.1.4.c. Pour comprendre (3.34), supposons que le modèle de synthèse à l'origine des observations \mathbf{y}_n est effectivement un mélange paramétrique à K_0 classes. On aimerait qu'asymptotiquement, la loi *a posteriori* marginale du nombre de classes se concentre autour de K_0 , c'est-à-dire $\lim_{N \rightarrow +\infty} P[K_N = K_0] = 1$. Le théorème 3.5 affirme que ce ne sera jamais le cas. En ce sens, le théorème est en faveur d'une inconsistance de l'estimateur MAPm du nombre de classes. Bien que non détaillé dans ici, ce résultat est valable pour une plus large gamme de modèles.

La figure 3.10 illustre ce résultat sur un cas simple. On génère N observations \mathbf{y} suivant une loi normale en dimension 2. Pour chaque valeur de N , on approxime la loi marginale *a posteriori* $P[K | \mathbf{y}]$ du modèle \mathcal{M} . Ici, le vrai nombre de classe est donc 1. Pourtant, on constate que plus N augmente, et plus la probabilité $P[K = 1 | \mathbf{y}_{1:N}]$ diminue. Les grandes valeurs de K sont d'ailleurs de plus en plus explorées lorsque N augmente.

Ce dernier résultat peut sembler remettre en question l'intérêt de la méthodologie proposée. Toutefois, ce résultat négatif est à pondérer. D'abord, le résultat d'inconsistance n'est pour l'instant pas vrai pour tous les estimateurs, et peu de recherches ont été menées dans la conception d'estimateurs plus efficaces. Nous aurons l'occasion de revenir sur ce point au prochain chapitre.

Ensuite, il existe une classe de problèmes où l'inconsistance n'est pas un problème. Dans ce cas, les modèles bayésiens non-paramétriques apportent un côté adaptatif : les modèles se raffinent lorsque le nombre d'observations augmente. Historiquement, le processus de Dirichlet à mélange a été introduit pour l'estimation de densité non paramétrique. Dans l'exemple introductif du chapitre, on voulait classer des plats. Comme le nombre d'ingrédients et de combinaisons sont infinis, on s'attend à ce que le nombre de classes augmente avec le nombre de plats.

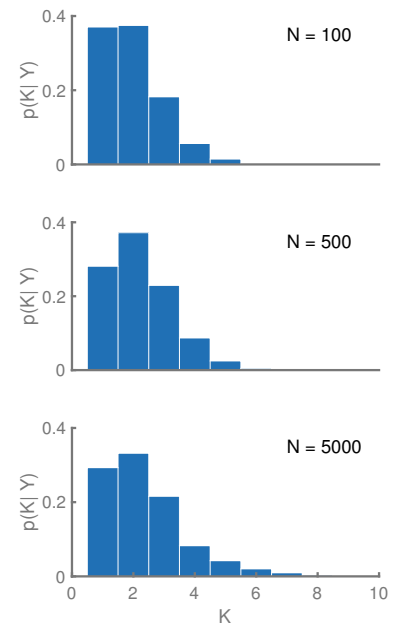


FIGURE 3.10 : Loi *a posteriori* $K | \mathbf{y}_{1:N}$ du nombre de classes pour différentes valeurs de N , et $\alpha = 1$.

3.2 MODÈLES À VARIABLES LATENTES ET PROCESSUS BETA BERNOULLI

LA CLASSIFICATION non supervisée, étudiée dans la partie précédente, vise à partitionner les données et chaque observation est associée à une et une seule classe. Dans le cas général des modèles à variables latentes un point peut être associé à plusieurs classes, *i.e.* plusieurs paramètres.

Dans la partie 3.1 on cherchait à obtenir une partition, *i.e.* une collection d'ensembles disjoints

$$\pi = \left\{ \{1, 3, 5\}, \{2, 4\}, \{6\} \right\}. \quad (3.35)$$

Ici, on désire obtenir une collection d'ensemble de la forme

$$\pi = \left\{ \{1, 3, 4, 5\}, \{1, 2, 4\}, \{5, 6\} \right\}, \quad (3.36)$$

où les indices qui se répètent sont **coloriés**. Cette collection indique quels paramètres sont associés à quelles observations. C'est en quelque sorte une partition où l'on a relâché la contrainte d'intersection vide. On dit que π est une *famille de sous-ensembles*. Une représentation possible de π est une matrice binaire, où les lignes représentent les observations et les colonnes les classes. L'entrée n, k vaut 1 si l'observation n est associée au groupe k , et 0 sinon. La figure 3.11 ci-contre donne un exemple d'une telle matrice.

Dans cette partie, nous allons voir comment générer des collections où l'on ne spécifie pas à l'avance le nombre de sous-ensembles.

3.2.1 Modèles à variables latentes

Considérons un ensemble d'observations $\mathbf{y}_{1:N}$ de \mathbb{R}^D . Les modèles à variables latentes font l'hypothèse qu'il existe un ensemble de caractéristiques non observées, ou *latent features*, qui permettent d'expliquer ces observations. Le nombre de caractéristiques est potentiellement infini, mais seul un sous ensemble fini des caractéristiques est nécessaire pour décrire une observation, et une caractéristique peut être partagée par plusieurs observations. Nous allons voir comment utiliser les collections de sous-ensembles pour représenter un modèle à variables latentes.

Formellement, un modèle à variables latentes suppose que les observations $\mathbf{y}_{1:N}$ sont décrites par une loi de probabilité p , paramétrée par deux quantités. La première est l'ensemble des caractéristiques $\theta_1, \dots, \theta_K$ élément de Θ , où Θ est l'espace des caractéristiques. Ces caractéristiques sont communes à l'ensemble des observations, et le nombre de caractéristiques K est potentiellement infini. La seconde quantité contient l'attribution des caractéristiques, sous forme d'un vecteur binaire de taille K , noté \mathbf{z}_n . L'élément $z_{n,k}$ vaut 1 si la caractéristique k est nécessaire pour décrire \mathbf{y}_n , et 0 sinon. Ainsi,

$$\forall n, \quad \mathbf{y}_n | \mathbf{z}_n, \theta_1, \dots, \theta_K \sim p(\cdot | \mathbf{z}_n, \theta_1, \dots, \theta_K). \quad (3.37)$$

Les vecteurs $\mathbf{z}_{1:N}$ induisent une collection de sous-ensembles, comme à l'équation (3.36), qui peut se représenter sous forme matricielle. Cette

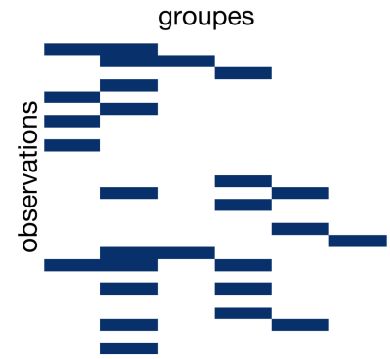


FIGURE 3.11 : Représentation d'une collection sous forme matricielle. Les cases bleues foncées représentent des 1 et les cases blanches des 0.

matrice, que l'on note \mathbf{Z} , contient N lignes mais un nombre K de colonnes potentiellement infini. D'après ce que l'on a dit précédemment, la somme de chaque ligne de la matrice \mathbf{Z} est finie presque sûrement. Notons que cette contrainte est triviale pour la classification non supervisée : on exigeait qu'une observation soit décrite par exactement un paramètre.

La suite de ce chapitre décrit une formulation bayésienne non paramétrique des modèles à variables latentes. Comme pour les modèles de mélange, nous commençons par étudier un processus stochastique servant d'*a priori* joint sur les paramètres et les attributions : c'est le processus Beta. Comme ce processus est difficile à manipuler en pratique, nous en déduisons le *processus du buffet indien (IBP)*, *i.e.* la loi sur les collections. Comme dans la partie 3.1.2, on pourrait commencer par formuler la version paramétrique du problème. On fixerait K le nombre de caractéristiques, puis on tirerait des variables binaires suivant des lois de Bernoulli, et K caractéristiques *i.i.d.* suivant une mesure de base G_0 . L'IBP, premier modèle bayésien non paramétrique pour ce problème a d'ailleurs été formulé en tant que modèle limite d'un tel modèle paramétrique (Ghahramani et Griffiths, 2006). Toutefois, tous les ingrédients pour traiter rigoureusement ce problème de façon non paramétrique ont été introduits dans la partie 3.1.4.c.

À nouveau, le lecteur intéressé par les aspects pratiques peut directement se rendre à la partie 3.2.3.c.

3.2.2 Processus Beta-Bernoulli

On construit dans cette partie le processus Beta-Bernoulli qui servira de loi *a priori* non paramétrique pour les modèles à variables latentes. La représentation que nous allons présenter s'appuie sur l'association d'un processus Beta et d'un processus de Bernoulli et est due à Thibaux et Jordan (2007). C'est le pendant pour les modèles à variables latentes du processus de Dirichlet à mélange.

Par analogie avec la classification non supervisée, il faut générer deux suites. La première est notée θ et contient les caractéristiques latentes. La seconde suite, notée \mathbf{z}_n est binaire et indique quelles caractéristiques sont nécessaires pour décrire l'observation \mathbf{y}_n . Le bon objet pour générer une quantité dénombrable de caractéristiques est le processus ponctuel de Poisson. Les caractéristiques sont à nouveau distribuées suivant une mesure de base G_0 , et associé à un label 1 si la caractéristique est associée à l'observation, 0 sinon. Ce processus est appelé *processus de Bernoulli*. Cependant, on souhaite que les caractéristiques obtenues avec deux réalisations d'un processus de Bernoulli soient d'intersection non vide. Ce ne sera presque sûrement pas le cas si la mesure de base G_0 est continue. La solution va consister à utiliser une mesure aléatoire discrète comme mesure de base.

Soit G_0 une mesure finie sur Θ , l'espace des caractéristiques. Notons α la masse de G_0 , $\alpha \triangleq G_0(\Theta) < +\infty$. Notons que G_0/α joue à nouveau le rôle de loi *a priori* sur l'espace des paramètres. Une réalisation d'un processus Beta-Bernoulli est une mesure aléatoire sur $\Theta \times \{0, 1\}$. La

construction se fait en deux étapes : on spécifie d'abord le processus qui génère une collection de paramètres associés à un 0 ou un 1 avec des lois de Bernoulli puis le processus qui génère les paramètres de ces Bernoulli.

Définition 3.5. Soit G_0 une mesure finie sur Θ . Soit $Be = \{(\theta_1, z_1), \dots\}$ un processus ponctuel de Poisson sur $\Theta \times \{0, 1\}$ de mesure de base

$$\mu_{Be}(d\theta, dw) = G_0(d\theta)\delta_1(dw). \quad (3.38)$$

On appelle réalisation d'un processus de Bernoulli, notée $\varphi_{Be} \sim BeP(G_0)$, la mesure complètement aléatoire

$$\varphi_{Be} \triangleq \sum_{k=1}^{+\infty} z_k \delta_{\theta_k}, \quad (3.39)$$

où z_k vaut 0 ou 1.

Une réalisation d'un processus de Bernoulli est donc bien une collection de variables binaires et de paramètres. Comme la définition s'appuie à nouveau sur un processus de Poisson, la collection est bien dénombrable. Cependant, si la mesure de base G_0 sur les paramètres est continue, ce qui est le cas dans beaucoup de modèles, les paramètres ne seront presque sûrement pas partagés d'une réalisation à l'autre. En conséquence, on ne générera que des collections triviales, c'est-à-dire des collections composées d'un unique élément.

La solution à ce problème consiste à construire une mesure discrète φ_B à partir de la mesure potentiellement continue G_0 . En effet, si $\varphi_B = \sum_{\ell} w_{\ell} \delta_{\theta_{\ell}}$ est la mesure de base d'un processus de Bernoulli, alors une réalisation φ_{Be} est de la forme $\varphi_{Be} = \sum_{\ell} z_{\ell} \delta_{\theta_{\ell}}$ avec $z_{\ell} \sim Be(w_{\ell})$. Il suffit d'être capable de générer une infinité dénombrable de couples $(w_{\ell}, \theta_{\ell})$, tels que w_{ℓ} est dans $[0, 1]$ et θ_{ℓ} dans Θ . Dans le cas paramétrique, on avait choisi la loi Beta pour sa propriété de conjugaison avec la loi de Bernoulli. Dans le cas non paramétrique, c'est le processus Beta qui est conjugué avec le processus de Bernoulli.

Définition 3.6. Soit $B = \{(w_1, \theta_1), \dots\}$ un processus ponctuel de Poisson sur $\Theta \times [0, 1]$ de mesure

$$\mu_B(d\theta, dw) = G_0(d\theta)\gamma w^{-1}(1-w)^{\gamma-1}dw. \quad (3.40)$$

On appelle réalisation d'un processus Beta, notée $\varphi_B \sim BP(\gamma, G_0)$, la mesure complètement aléatoire

$$\varphi_B \triangleq \sum_{k=1}^{+\infty} w_k \delta_{\theta_k}. \quad (3.41)$$

Vue comme une fonction de w , la mesure μ_B ressemble à une loi Beta dégénérée et ne charge que l'intervalle $[0, 1]$. À nouveau

$$\mu_B(\Theta \times [0, 1]) = \int_{\Theta \times [0, 1]} \mu_B(d\theta, dw) = +\infty, \quad (3.42)$$

donc le processus de Poisson sous-jacent génère bien presque sûrement une infinité dénombrable de points. Comme pour le processus Gamma, on peut montrer que les marginales d'un processus Beta sont des lois Beta en comparant les transformées de Laplace.

On utilise volontairement une notation différente pour les poids dans l'équation (3.41) comparée à la partie précédente. Cette distinction est faite pour rappeler que les deux suites de poids se comportent différemment ; les poids a d'un tirage d'un processus de Dirichlet vérifie $\sum_{k=1}^{+\infty} a_k = 1$, tandis que la somme des poids w_k est finie avec probabilité 1 puisque $\mathbb{E}[w] = \alpha < +\infty$.

L'assemblage des deux processus, que l'on appelle *processus Beta-Bernoulli*, est décrit par le modèle suivant

$$\begin{aligned} \varphi_B | \gamma, G_0 &\sim BP(\gamma, G_0) \\ \varphi_{B,1}, \dots, \varphi_{B,N}, \theta | \varphi_B &\stackrel{\text{i.i.d.}}{\sim} \mathcal{BP}(\varphi_B). \end{aligned} \quad (\mathcal{M}_{B-B\epsilon})$$

Dans le modèle $\mathcal{M}_{B-B\epsilon}$, les paramètres θ_ℓ sont partagés par les réalisations du processus de Bernoulli $\varphi_{B,n}$.

La figure 3.12 montre 50 réalisations du modèle $\mathcal{M}_{B-B\epsilon}$. Chaque ligne correspond à une réalisation $\varphi_{B,n}$, et chaque point bleu indique la position d'un paramètre associé à un 1. De fait, chaque colonne représente la collection implicitement définie par les 1 des variables $z_{k,n}$. On affiche en dessous du graphique la réalisation du processus Beta. On voit que les caractéristiques les plus activées correspondent logiquement aux points de φ_B où la masse est la plus grande. Nous verrons dans la prochaine partie comment générer des réalisations d'un processus Beta.

Le nombre de paramètre décrivant une observation, *i.e.* le nombre de succès de la suite $(z_{k,n})_k$ est aléatoire. Cependant, comme

$$\mathbb{E}[\mathbf{z}] = \sum_{k=1}^{+\infty} w_k = \int_{\Theta \times \{0,1\}} w \mu_B(d\theta, dw) < +\infty, \quad (3.43)$$

le lemme de Borel Cantelli nous assure que, presque sûrement, seul un nombre fini de z_k vaut 1, *i.e.*, seul un nombre fini de paramètres est activé à chaque réalisation $\varphi_{B,n}$ et le cahier des charges est respecté.

Comme pour le processus de Dirichlet, le processus Beta n'est pas directement observé. Cependant, on peut exploiter les réalisations du processus de Bernoulli pour le caractériser : il s'agit de la loi *a posteriori*. Soient $\varphi_{B,1}, \dots, \varphi_{B,N}$ N réalisations du modèle $\mathcal{M}_{B-B\epsilon}$. Alors, la loi *a posteriori* de la mesure φ_B est encore un processus Beta

$$\varphi_B | \varphi_{B,1}, \dots, \varphi_{B,N}, \gamma, G_0 \sim BP\left(\gamma, G_0 + \sum_{n=1}^N \varphi_{B,n}\right). \quad (3.44)$$

On préfère écrire (3.44) sous une forme normalisée

$$\varphi_B | \varphi_{B,1:N}, \gamma, G_0 \sim BP\left(\gamma + N, \frac{\gamma}{\gamma + N} G_0 + \sum_{n=1}^N \frac{1}{\gamma + N} \varphi_{B,n}\right). \quad (3.45)$$

Ainsi, la loi *a posteriori* d'un processus Beta est encore un processus Beta dont la mesure de base a été mise à jour. Cette nouvelle mesure de base comporte une partie discrète, qui est un compromis entre l'information *a priori* G_0 et l'information empirique fournie par les $\varphi_{B,n}$. Notons que plus le nombre d'observations N est grand, et plus l'influence de la mesure de base G_0 diminue.

3.2.3 Simulation et inférence

Nous manipulons dans cette partie le processus Beta-Bernoulli afin de le rendre utilisable en pratique. Le plan est relativement similaire à celui

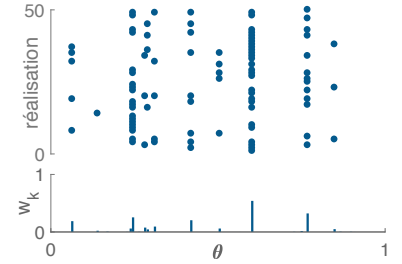


FIGURE 3.12 : La figure du haut illustre 50 réalisations d'un processus de Bernoulli. Chaque ligne est une réalisation et chaque point représente un 1. La figure du bas est la mesure de base du processus de Bernoulli, *i.e.* le tirage du processus Beta déjà présenté à la figure 3.13.

de la partie 3.1.4 : on présente d'abord dans la partie 3.2.3.a un moyen de générer des réalisations suivant le modèle $\mathcal{M}_{B-\mathcal{B}}$. Nous présenterons un modèle d'urne dans la partie 3.2.3.b, qui aboutit à un premier échantillonneur de Gibbs. Nous déterminerons dans la dernière partie la loi sur les collections sous-jacente au modèle, ce qui aboutira à un échantillonneur plus efficace.

3.2.3.a Simulation de processus Beta-Bernoulli

Générer un processus de Bernoulli lorsque la mesure de base G_0 est continue revient à échantillonner un processus de Poisson. En revanche, échantillonner un processus Beta est plus compliqué car la mesure de Lévy est difficile à évaluer lorsque G_0 est continue. Comme pour le processus de Dirichlet, il existe une procédure de *stick breaking* (Teh et al., 2007). Toutefois, la propriété de conjugaison des processus donnée équation (3.45) fournit une procédure efficace mais plus formelle pour générer des réalisations (Thibaux et Jordan, 2007).

Soit G_0 une mesure continue sur Θ , γ un réel positif et $\varphi_B \sim BP(\gamma, G_0)$. L'équation (3.45) montre que la mesure de base du processus Beta mis à jour est la somme d'une mesure continue et d'une mesure discrète. Si $N = 1$, et si $X_1 \sim \mathcal{B}P(G_0)$, alors la partie discrète et la partie continue de la loi *a posteriori* de φ_B sont traitées indépendamment. On peut écrire

$$\varphi_B \stackrel{d}{=} D_1 + C_1, \quad (3.46)$$

où $D_1 \sim BP(\gamma + 1, \frac{1}{\gamma+1}\varphi_{B,1})$ et $C_1 \sim BP(\gamma + 1, \frac{1}{\gamma+1}G_0)$. Les notations D et C font respectivement références aux qualités discrètes et continues des nouvelles mesures de base.

Comme la mesure aléatoire X_1 est discrète, elle s'écrit $X_1 = \sum_k \delta_{\theta_k}$ où les θ_k sont i.i.d. suivant G_0 . Une réalisation de D_1 s'écrit alors $D_1 = \sum_k w_k \delta_{\theta_k}$ où les w_k sont des lois Beta de paramètres $(1, \gamma)$.

En répétant le même raisonnement sur C_1 , on obtient par récurrence

$$\forall n \in \mathbb{N}^*, \quad B \stackrel{d}{=} \sum_{i=1}^n D_i + C_n \quad (3.47)$$

$$\forall n \in \mathbb{N}^*, X_n \sim \mathcal{B}P\left(\frac{\gamma}{\gamma + n - 1} G_0\right) \quad (3.48)$$

$$\forall n \in \mathbb{N}^*, D_n \sim \mathcal{B}P\left(\gamma + n, \frac{1}{\gamma + n} X_n\right) \quad (3.49)$$

$$\forall n \in \mathbb{N}^*, C_n \sim \mathcal{B}P\left(\gamma + n, \frac{\gamma}{\gamma + n} G_0\right). \quad (3.50)$$

On voit équation (3.50) que la masse de la mesure de base tend vers 0. On a donc asymptotiquement $\lim_{n \rightarrow +\infty} D_n \stackrel{d}{=} \varphi_B$. On peut donc se servir de la suite de mesure B_n pour approcher une réalisation d'un processus Beta.

La figure 3.13 montre sur le même graphique un tirage d'un processus Beta et la densité cumulée associée approchée en prenant $n = 1000$

Pour être consistant, les notations φ_B , φ_D et φ_C devraient être employées.

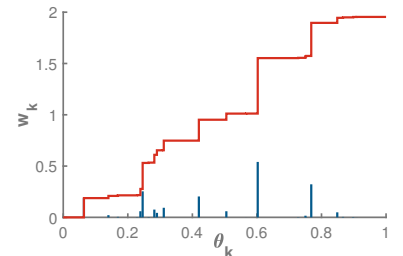


FIGURE 3.13 : Tirage d'un processus Beta de paramètre 5 et de mesure de base la loi uniforme sur $[0, 1]$. La courbe rouge représente la mesure cumulée empirique $\int_0^t \varphi_B(d t)$.

dans (3.48). On vérifie bien que la quantité $\varphi_B([0, 1])$, *i.e.*, la somme des poids, n'est pas égale à 1. Les réalisations $\varphi_{\mathcal{B}_e}$ ne sont pas des mesures de probabilités.

3.2.3.b Un modèle d'urne

Nous avons mis en évidence pour les processus de Dirichlet à mélange l'existence d'un modèle d'urne. Ce modèle d'urne a permis la mise en place d'algorithmes d'échantillonnage simples pour faire de l'inférence. C'est encore le cas pour le processus Beta-Bernoulli.

Pour le processus de Dirichlet à mélange, nous avons marginalisé la loi sur les paramètres par rapport au processus de Dirichlet. Ici, nous allons marginaliser vis-à-vis du processus Beta. Les mêmes calculs mènent à

$$P[\varphi_{\mathcal{B}_e, N+1} | \varphi_{\mathcal{B}_e, 1:N}] = \frac{\gamma}{\gamma + N} G_0 + \sum_{n=1}^N \frac{1}{\gamma + N} \varphi_{\mathcal{B}_e, n}. \quad (3.51)$$

D'où

$$\varphi_{\mathcal{B}_e, N+1} | \varphi_{\mathcal{B}_e, 1:N} \sim \text{BeP}\left(\frac{\gamma}{\gamma + N} G_0 + \sum_{n=1}^N \frac{1}{\gamma + N} \varphi_{\mathcal{B}_e, n}\right). \quad (3.52)$$

Une autre forme plus lisible est toutefois préférée. Considérons des singletons de Θ , notés $\{\theta_k\}$. D'après l'équation (3.52), la masse de cet ensemble est proportionnelle à $m_{N,k} \triangleq \sum_{n=1}^N \varphi_{\mathcal{B}_e, n}(\{\theta_k\})$, ce qui revient à compter le nombre de mesures $\varphi_{\mathcal{B}_e, n}$ chargeant θ_k . Pour cette raison, la loi du processus équation (3.52) est plutôt notée

$$m_{N,k} \triangleq \sum_{n=1}^N \varphi_{\mathcal{B}_e, n}(\{\theta_k\}) \quad (3.53)$$

$$\varphi_{\mathcal{B}_e, N+1} | \varphi_{\mathcal{B}_e, 1:N} \sim \text{BeP}\left(\frac{\gamma}{\gamma + N} G_0 + \sum_{k=1}^{+\infty} \frac{m_{N,k}}{\gamma + N} \delta_{\theta_k}\right). \quad (3.54)$$

La loi *a posteriori* marginale est encore un processus de Bernoulli dont la mesure de base est composée d'une partie continue et d'une partie discrète. L'échantillonnage est réalisé en séparant les contributions discrète et continue : on additionne le résultat d'une réalisation d'un processus de Poisson de mesure de base $\frac{\gamma}{\gamma + N} G_0$ au tirage des Bernoulli pour chaque valeurs distinctes des paramètres θ_k et proportionnellement à $\frac{m_{N,k}}{\gamma + N}$.

Le schéma algorithmique que nous venons de décrire est exactement celui d'un échantillonneur de Gibbs des paramètres θ_k . À chaque itération de l'algorithme et pour chaque observation \mathbf{y}_n , on teste via les Bernoulli si \mathbf{y}_n est bien décrit par le paramètre θ_k , puis l'on tire suivant une loi de Poisson des nouveaux paramètres. Cependant, cet algorithme souffre des mêmes défauts que celui issu de l'urne de Blackwell-MacQueen pour les processus de Dirichlet à mélange : si l'on souhaite mettre à jour la valeur d'un paramètre commun à un ensemble d'observations, il faut d'abord qu'une observation quitte l'ensemble, tire la nouvelle valeur puis que toutes les autres la rejoignent. Comme dans la partie 3.1, une façon de résoudre ce problème consiste à séparer l'échantillonnage des collections et des paramètres. C'est ce que nous allons faire dans la prochaine partie.

3.2.3.c Le processus du buffet indien

Pour concevoir un échantillonneur plus efficace, nous avons évoqué le besoin de déterminer la loi sur les collections d'ensemble induite par le processus Beta-Bernoulli. C'est le *processus du buffet indien* (*indian buffet process*, IBP). Pour rappel, le CRP avait été obtenu comme loi sur les partitions induite par un processus de Dirichlet à mélange.

Chaque tirage $\varphi_{\mathcal{B},n}$ est une collection de caractéristiques θ_k . Comme pour les modèles de mélange, l'équation (3.52) montre que certaines caractéristiques sont communes à plusieurs tirages. Pour le voir, considérons N tirages, et les suites binaires $\mathbf{z}_n \triangleq (\varphi_{\mathcal{B},n}(\{\theta_k\}))_k$. Si l'on regroupe les vecteurs \mathbf{z}_n dans une matrice binaire \mathbf{Z} , à N lignes mais à une infinité dénombrable de colonnes, on obtient une matrice représentant une collection de sous-ensembles comme présentée en début de partie, cf. figure 3.11. L'IBP est la loi de cette matrice aléatoire.

L'IBP a d'abord été introduit par Ghahramani et Griffiths (2006) pour $\gamma = 1$ comme limite d'un modèle Beta-Bernoulli paramétrique. Dans la version que nous allons présenter, l'IBP est contrôlé par deux paramètres réels positifs, $\alpha \triangleq G_0(\Theta)$ et γ , et noté $IBP(\alpha, \gamma)$. L'échantillonnage est élégamment décrit par la métaphore culinaire suivante.

Un client entre dans un restaurant et se sert d'un nombre $K_1 \sim \mathcal{P}(\alpha)$ plats, où \mathcal{P} désigne la loi de Poisson. Lorsque n clients se sont servis de K plats, notons m_{nk} le nombre fois où le plat $n \leq K$ a été choisi par les k clients. Le client $n + 1$ entre et se sert avec probabilité $\frac{m_{nk}}{\gamma+n}$ des plats déjà choisis par les clients précédents, puis se sert d'un nombre $K_{n+1} \sim \mathcal{P}(\frac{\gamma}{\gamma+n}\alpha)$ nouveaux plats. La formule des probabilités composées associée à une récurrence donne

$$\begin{aligned} \mathbb{P}[\mathbf{Z}] &= \prod_{n=1}^N \mathbb{P}[\mathbf{z}_n | \mathbf{z}_1, \dots, \mathbf{z}_{n-1}] \\ &= \frac{(\alpha\gamma)^K}{\prod_{n=1}^N K_n!} e^{\alpha\gamma \sum_{n=1}^N \frac{1}{\gamma+n-1}} \prod_{k=1}^K \frac{\Gamma(m_{Nk})\Gamma(N - m_{Nk} + \gamma)}{\Gamma(N + \gamma)}, \end{aligned} \quad (3.55)$$

où K est la somme des plats ajoutés par les N clients, $K = \sum_{n=1}^N K_n$. Notons le rôle des deux paramètres α et γ . Plus α est grand, plus le nombre de plats croît, d'où son nom paramètre de *masse*. γ contrôle l'utilisation d'un plat; plus γ est grand et moins un plat aura tendance à être partagé, d'où le nom paramètre de *concentration*.

La figure 3.14 montre une réalisation de l'IBP pour $(\alpha, \gamma) = (7, 1)$.

Cette description est cependant incomplète car les lignes d'une matrice $\mathbf{Z} \sim IBP(\alpha, \gamma)$ ne sont pas échangeables. En effet, on voit équation (3.55) que la loi des variables K_n , dépend de n , et n'est donc pas invariante par permutation. En bref, comme la répartition des 1 évolue au fil des lignes, ces lignes ne sont pas échangeables.

La solution consiste à considérer la loi sur des classes d'équivalences de matrices. Comme l'ordre des colonnes importe peu, on dira que deux

La définition de m_{nk} rappelle volontairement celle en (3.53).

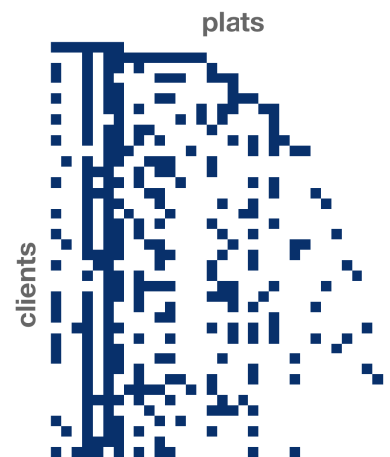


FIGURE 3.14 : une réalisation du processus du buffet indien à deux paramètres pour $N = 40$ et $(\alpha, \gamma) = (7, 1)$. Si le client n s'est servi du plat k , alors la case (n, k) est coloriée en bleu, et blanche sinon.

matrices binaires sont équivalentes si l'on peut passer de l'une à l'autre en permutant les colonnes. La figure 3.15 montre deux représentants de la même classe d'équivalence.

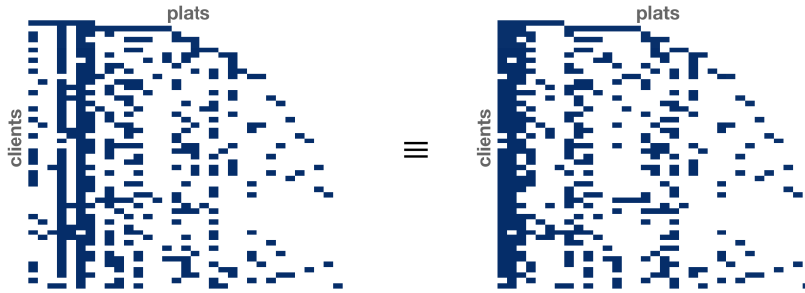


FIGURE 3.15 : Gauche : une réalisation du processus du buffet indien à deux paramètres pour $N = 50$ et $(\alpha, \gamma) = (7, 1)$. Droite : classe d'équivalence de la matrice à gauche.

On ne s'intéresse qu'à la loi du nombre de 1 par ligne, *i.e.*, du nombre de cases bleues. La position des 1 n'importe plus. Sans détailler rigoureusement les calculs, il est facile de se convaincre que cette loi est la loi de Poisson $\mathcal{P}(\alpha)$. Pour la première ligne, c'est vrai par construction, $K_1 \sim \mathcal{P}(\alpha)$. La seconde ligne possède $\sum_{i=1}^{K_1} U_i + K_2$ cases bleues, où $U_i \stackrel{\text{i.i.d.}}{\sim} \text{Be}(\frac{1}{1+\gamma})$ et $K_2 \sim \mathcal{P}(\frac{\gamma}{1+\gamma}\alpha)$. Par additivité de la loi de Poisson, la somme est une loi de Poisson de paramètre α . Les mêmes arguments s'appliquent aux lignes suivantes.

Notons $[\mathbf{Z}]$ la classe d'équivalence de \mathbf{Z} . La loi de $[\mathbf{Z}]$ s'obtient à partir de la loi de \mathbf{Z} en tenant compte de toutes les permutations possibles sur les colonnes. Il y a en tout 2^{N-1} colonnes différentes, que l'on indexe par h . La loi $[\mathbf{Z}]$ devient

$$\mathbb{P}([\mathbf{Z}]) = \frac{(\alpha\gamma)^K}{\prod_{h=1}^{2^{N-1}} K_h!} e^{\alpha\gamma \sum_{n=1}^N \frac{1}{\gamma+n-1}} \prod_{k=1}^K \frac{\Gamma(m_{Nk})\Gamma(N - m_{Nk} + \gamma)}{\Gamma(N + \gamma)}, \quad (3.56)$$

où K_h désigne le nombre de répétitions de la colonne indexée par h .

Par analogie avec le processus du restaurant chinois, la loi équation (3.56) est appelée *Exchangeable Feature Probability Function* (EFPF) (Broderick et al., 2013).

Comme pour les processus de Dirichlet à mélange, on est parti d'un modèle statistique complexe pour aboutir à une loi sur les collections d'ensembles, ou de manière équivalente, sur les matrices binaires de taille potentiellement infinie. Cette formulation permet également de comprendre le fonctionnement de l'IBP à travers quelques propriétés.

Considérons une matrice binaire $\mathbf{Z} \sim \text{IBP}(\alpha, \gamma)$. La loi du nombre total de plats, notée K , s'obtient en additionnant les plats ajoutés par les clients successifs. Par additivité de la loi de Poisson, $K \sim \mathcal{P}(\alpha\gamma \sum_{n=1}^N \frac{1}{\gamma+n-1})$. Asymptotiquement,

$$\mathbb{E}[K] = \alpha\gamma \sum_{n=1}^N \frac{1}{\gamma+n-1} \underset{N \rightarrow +\infty}{\sim} \alpha\gamma \ln N. \quad (3.57)$$

Comme pour le CRP, le nombre de plats, *i.e.*, le nombre de paramètres, est en espérance sous linéaire. En pratique, cela signifie que le modèle

utilisera en moyenne $\mathcal{O}(\ln(N))$ paramètres pour décrire N observations. L'IBP induit donc naturellement un effet régularisant sur le nombre de paramètres.

On rappelle que le nombre de 1 des lignes de \mathbf{Z} suit une loi de Poisson de paramètre α grâce à la propriété d'échangeabilité. Par additivité de la loi de Poisson, le nombre de 1 dans \mathbf{Z} suit encore une loi de Poisson de paramètre $N\alpha$. Notons que le ratio des espérances du nombre de 1 et de la taille de la matrice, *i.e.*, la quantité

$$\frac{\alpha N}{N\mathbb{E}[K]} \underset{N \rightarrow +\infty}{\sim} \frac{1}{\gamma \ln(N)}, \quad (3.58)$$

tend vers 0. En pratique, cela signifie que le modèle a tendance à favoriser des matrices \mathbf{Z} creuses. Ce résultat rend l'IBP intéressant si l'on souhaite construire des modèles favorisant la parcimonie.

La partie consacrée au processus Beta-Bernoulli et de l'IBP s'achèvent ici. Nous décrirons un cas pratique de l'utilisation de l'IBP dans un modèle à variables latentes au prochain chapitre.

3.3 DISCUSSION

NOUS AVONS DISCUTÉ dans ce chapitre deux modèles bayésiens non paramétriques pour traiter les problèmes de classification non supervisées et les modèles à variables latentes. Le côté non paramétrique se traduit dans les deux cas par l'utilisation de processus de Poisson pour générer des mesures aléatoires. Ces modélisations sont assez formelles et complexes à utiliser dans le cadre de l'inférence. Pour cette raison, nous avons déterminé un processus sous-jacent à valeur directement sur les partitions ou les collections d'ensembles. Ce sont les processus du restaurant chinois et du buffet indien. Ces deux processus se comprennent et s'expliquent facilement à l'aide d'une métaphore, et sont simples à manipuler. Le CRP nous a permis de construire un échantillonneur de Gibbs pour faire de l'inférence. Comme nous le verrons au prochain chapitre, il est possible de faire de même avec l'IBP pour les modèles à variables latentes.

En définitive, retenons que la définition formelle des mesures aléatoires est importante pour comprendre l'origine des algorithmes et définir des modèles plus sophistiqués. Mais d'un point de vue pratique, il est toujours possible de se ramener à des algorithmes simples qui ne nécessitent pas de manipuler des objets mathématiques complexes.

D'un point de vue modélisation, ces modèles non paramétriques ont l'avantage de ne pas fixer à l'avance l'ordre du modèle. La modélisation bayésienne apporte naturellement un effet régularisant. Cet effet se comprend très facilement : les modèles ont besoin de $\mathcal{O}(\ln(N))$ paramètres pour décrire N observations. Cela signifie également que les modèles se raffinent au fur et à mesure que le nombre d'observations augmente. Même si le formalisme bayésien non-paramétrique n'a pas été pensé dans ce but, il est naturel de vouloir se servir de ces modèles pour estimer le

nombre de paramètres. Cependant, nous avons montré pour la classification non supervisée que les estimateurs bayésiens de l'ordre du modèle sont à manipuler avec prudence. Toutefois, peu de résultats existent concernant la consistance de ces estimateurs d'une manière générale. Nous aurons l'occasion de discuter ce point plus en détails au prochain chapitre.

Ce chapitre est loin d'avoir couvert de façon exhaustive les deux sujets. Nous évoquions précédemment la conception de modèles plus sophistiqués. Ces modèles visent par exemple à récupérer des comportements en loi de puissance de certaines quantités, comme la taille des classes (Pitman et Yor, 1997) ou les fréquences d'utilisation de chaque plats (Caron, 2012). Ces modèles sont construits de la même façon que ceux de ce chapitre, en ajoutant par exemples des paramètres supplémentaires dans les mesures de Lévy (cf. équations (3.11) et (3.40)). À nouveau, ces modèles s'interprètent et s'utilisent facilement à partir de versions modifiées du CRP ou de l'IBP.

Si l'on poursuit le raisonnement bayésien, on peut aussi désirer paramétrer la mesure de base G_0 , ce qui aboutit à des modèles hiérarchiques (Teh et al., 2006 ; Thibaux et Jordan, 2007). Ces modèles sont utiles si l'on souhaite réaliser conjointement une tâche sur plusieurs jeux de données. Par exemple, on peut avoir envie de faire de la classification non supervisée sur plusieurs jeux de données tout en partageant des paramètres.

Les estimateurs Bayésiens pour la classification non supervisée ont été abordés dans ce chapitre avec des méthodes MCMC. D'autres approches ont été développées pour l'inférence, que nous n'avons pas évoquées. Ces approches s'appuient sur les modèles *stick breaking*, des approximations variationnelles, *etc.*

Pour finir, il existe d'autres problèmes où l'on fait appel à des modèles non paramétriques : processus Gaussien pour la régression (Rasmussen et Williams, 2005), processus de Mondrian pour la segmentation (Roy et Teh, 2009), *etc.*

Estimation bayésienne non paramétrique

4

de sous-espaces

AVEC L'ACCROISSEMENT de la quantité d'information disponible, la majorité des chaînes de traitement d'apprentissage automatique débute par une phase de réduction de la dimension. Mais cette étape provoque généralement une perte d'information. Il est donc crucial de bien estimer le sous-espace vis-à-vis de la tâche à accomplir.

La plupart des méthodes probabilistes d'estimation de sous-espaces fixent à l'avance la dimension du sous-espace. Des modèles hiérarchiques dans lesquels la dimension est également aléatoire ont été étudiés, mais nécessitent de recourir à des méthodes d'échantillonnage difficiles à mettre en place. Nous proposons dans ce chapitre le modèle d'*analyse en composantes principales bayésienne non paramétrique* (BNP-PCA). Il s'agit d'une ACP bayésienne qui utilise un modèle non paramétrique pour laisser libre la dimension du sous-espace. Le modèle BNP-PCA couple une loi *a priori* sur l'ensemble des matrices orthonormales avec un *processus du buffet indien* (IBP). Les estimateurs bayésiens sont ensuite approchés par un algorithme MCMC.

Puisque la dimension du sous-espace est également une variable aléatoire, il est naturel de vouloir l'estimer. Nous nous intéresserons à la consistance théorique et empirique des estimateurs de ce paramètre.

Le modèle BNP-PCA est ensuite validé sur un jeu de données synthétiques. L'étude numérique se conclut par deux applications sur des données réelles. Dans la première application, le modèle BNP-PCA est couplé avec un modèle de mélange bayésien afin de séparer deux classes sans réaliser à l'avance de réduction de la dimension ni régler de paramètres. Dans la deuxième application, on estime le sous-espace contenant les pixels d'un cube hyperspectral. L'objectif est double : les pixels vivent naturellement dans un sous-espace dont la dimension est étroitement liée au nombre d'objets présents sur la scène, souvent inconnu. Estimer le bon sous-espace permet de réduire la quantité de bruit capturé et améliore les performances des algorithmes.

Ce travail a été présenté dans Elvira et al. (2017c) et soumis dans Elvira et al. (2017b).

La partie 4.1 dresse un bref panorama des méthodes statistiques d'estimation de sous-espaces et de leur dimension. Le modèle BNP-PCA et l'algorithme MCMC sont décrits en 4.2 et 4.3. Nous discutons des estimateurs dans la partie 4.4. Les parties 4.5 et 4.6 présentent les résultats expérimentaux. La partie 4.7 conclut le chapitre puis la partie 4.8 discute des perspectives de ce travail.

4.1 ANALYSE EN COMPOSANTES PRINCIPALES ET FORMULATIONS PROBABILISTES

LE POINT DE DÉPART de ces travaux est l'interprétation de l'ACP comme le maximum de vraisemblance d'un modèle à variables latentes. Nous verrons ensuite comment estimer la dimension de ce sous-espace.

4.1.1 Vers une formulation probabiliste de la recherche de sous-espaces

L'ACP est une méthode de référence d'analyse de données consistant à trouver une nouvelle représentation des données à partir de variables décorréelées. Géométriquement l'ACP consiste à trouver une base orthonormée \mathbf{P} de \mathbb{R}^D , dont les éléments sont appelés *composantes principales*. La matrice \mathbf{P} est choisie telle que pour tout entier $K \leq D$, le sous-espace de \mathbb{R}^K induit par les K premières composantes principales minimise l'erreur de reconstruction quadratique d'un jeu de données $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$ de \mathbb{R}^D . Formellement, une interprétation de l'ACP est de chercher une base orthonormée \mathbf{P} de \mathbb{R}^D telle que chaque sous-matrice $\mathbf{P}_{:,1:K}$ formée par les K premières colonnes de \mathbf{P} vérifie (Jolliffe, 1986)

$$\forall K \in \{1, \dots, D\}, \quad \mathbf{P}_{:,1:K} = \underset{\mathbf{U} \in \mathcal{S}_D^K}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{U}\mathbf{U}^T\mathbf{Y}\|_2^2, \quad (4.1)$$

où \mathcal{S}_D^K est la variété des matrices orthonormées de dimensions $D \times K$, aussi appelée *variété de Stiefel* et $\|\cdot\|_2$ est la norme spectrale. L'ACP permet la réduction de la dimension en ne conservant que les K premières colonnes de \mathbf{P} .

Toutefois, l'équation (4.1) ne fournit pas d'outils clés en main pour mesurer la pertinence de chaque composante principale au vue d'une distribution statistique sur les données. Ce constat a amené Tipping et Bishop (1999b) à interpréter l'ACP comme l'estimateur du maximum de vraisemblance du modèle à variables latentes

$$\forall n \in \{1, \dots, N\}, \quad \mathbf{y}_n = \mathbf{W}\mathbf{x}_n + \boldsymbol{\epsilon}_n, \quad (4.2)$$

où \mathbf{W} est la matrice $D \times D$ de facteurs latents, \mathbf{x}_n le vecteur de \mathbb{R}^D contenant le poids de chaque facteur et $\boldsymbol{\epsilon}_n$ est un bruit blanc gaussien. Si l'on suppose que les coefficients \mathbf{x}_n sont gaussiens centrés de matrice de covariance \mathbb{I}_D , alors la loi marginale $p(\mathbf{Y}|\mathbf{W})$ est identifiable par conjugaison

$$p(\mathbf{Y}|\mathbf{W}) = \int p(\mathbf{Y}|\mathbf{W}, \mathbf{x}_1, \dots, \mathbf{x}_N) d\mathbf{x}_1 \dots d\mathbf{x}_N = \mathcal{N}\left(\mathbf{0}_D, (\mathbf{W}^T\mathbf{W} + \sigma^2\mathbb{I}_D)\right). \quad (4.3)$$

Tipping et Bishop (1999b) ont montré que l'estimateur du maximum de vraisemblance marginal associé à (4.3) sont les vecteurs propres de la matrice de covariance empirique $\mathbf{Y}^T\mathbf{Y}$ corrigée par la matrice de covariance du bruit σ^2 , bien qu'aucune contrainte d'orthogonalité ne soit spécifiée. Pour cette raison, ce modèle a été dénommé *probabilistic PCA (PPCA)*.

Les avantages du modèle PPCA sont multiples. Par exemple, PPCA permet la recherche de sous-espaces lorsque des données sont manquantes (Tip-

On obtient ce résultat en cherchant les facteurs latents \mathbf{W} maximisant la vraisemblance gaussienne donnée en (4.3).

ping et Bishop, 1999a, 1999b) en exploitant des algorithmes d'*espérance-maximisation* (EM), ou de capturer la non linéarité des données en utilisant des mélanges de PPCA.

Ces contributions forment le point de départ de l'utilisation des méthodes probabilistes pour l'apprentissage de sous-espaces. Plusieurs travaux ont cherché à incorporer de l'information *a priori* dans un modèle de type PPCA dans le but de régulariser une ACP lorsque peu de données sont disponibles (Bishop, 1999a; Dobigeon et Tourneret, 2010). D'autres travaux ont approfondi la formulation probabiliste d'une ACP non linéaire, y compris avec des méthodes bayésiennes (Bolton et al., 2003; Lawrence, 2005; Lian, 2009). On peut également évoquer la conception de modèles robustes à la présence de données corrompues et de données aberrantes (Archambeau et al., 2008; Schmitt et Vakili, 2016).

4.1.2 Estimer la dimension du sous-espace

Dans la plupart des cas, estimer la dimension du sous-espace revient à trouver un compromis entre perte d'information et gain de performance. Nous verrons dans la partie 4.6.1 qu'il existe des applications où estimer la dimension du sous-espace a un sens physique, comme en imagerie hyperspectrale. Estimer cette dimension peut s'interpréter comme résoudre un problème de choix de modèle. Dans la plupart des modèles présentés dans la partie précédente, l'erreur de reconstruction est un critère pour évaluer la qualité d'un modèle. Mais augmenter la dimension du sous-espace permettra toujours de diminuer l'erreur. C'est un problème de *sur-apprentissage*. Par exemple, augmenter le nombre de facteurs latents dans (4.3) fera toujours augmenter la vraisemblance $p(\mathbf{Y}|\mathbf{W})$ du modèle. Par conséquent, si l'on veut estimer la dimension, il faut pénaliser le nombre de paramètres. Cette pénalisation peut être introduite en ajoutant un terme *ad hoc* au critère à minimiser ou à travers le choix de la loi *a priori* dans un modèle bayésien.

Sous certaines hypothèses, le dimension du sous-espace d'un cube hyperspectral est lié au nombre d'objets macroscopiques présents dans une scène.

4.1.2.a Pénalité sur K

Plusieurs critères ont été introduits pour forcer une pénalisation du nombre de paramètres.

Akaike (1974) propose de choisir le modèle (le sous-espace ici) qui maximise la quantité $\ln p(\mathbf{Y}|\mathbf{W}_{\text{ML}}) - K$, où \mathbf{W}_{ML} désigne l'estimateur du maximum de vraisemblance des facteurs latents et K le nombre de paramètres. C'est le *critère d'information d'Akaike* (AIC).

L'évidence d'un modèle bayésien, *i.e.*, $\int p(\mathbf{Y}, \mathbf{W})d\mathbf{W}$ ici, est un outil courant pour la sélection de modèle, intervenant dans le calcul des facteurs de Bayes. Mais l'évidence est généralement difficile à calculer. Le *critère d'information de Bayes* (BIC) proposé par Schwarz (1978) s'obtient en approchant l'évidence avec entre autres une approximation de Laplace. La quantité finale à maximiser est $\ln p(\mathbf{Y}|\mathbf{W}_{\text{MAP}}) - 0.5K \ln N$, où \mathbf{W}_{MAP} désigne l'estimateur MAP des facteurs latents, K le nombre de paramètres et N le nombre d'observations. Le BIC est réputé plus efficace car il tient compte du nombre d'observations. Notons que l'on retrouve la pénalisa-

La méthode de Laplace permet de calculer des intégrales de la forme $\int_{-\infty}^{+\infty} e^{f(t)} dt$ en procédant à un développement de Taylor de $f(t)$ au second ordre autour de t_0 , un maximum de f . Cela revient à approcher f par une Gaussienne.

tion en $\ln(N)$ du nombre d'observations induit par le CRP et l'IBP vus au chapitre 3.

Dans une formulation bayésienne, la dimension K du sous-espace est aussi considérée comme une variable aléatoire. Mais la loi *a posteriori* discrète $K|\mathbf{Y}$ n'est généralement pas simple à manipuler. Notons que dès le départ, Bishop (1999a, 1999b) a proposé un estimateur de la taille du sous-espace induit par la version bayésienne de PPCA qui évite de manipuler directement cette loi discrète. Dans ce modèle hiérarchique, chaque colonne de \mathbf{w}_k suit *a priori* une loi normale centrée de variance $\alpha_k \mathbb{I}_D$. On espère que certaines valeurs de α_k vont tendre vers 0. Dans ce cas, les colonnes α_k associées vont tendre vers le vecteur nul. Si l'on désactive les colonnes de variance proche de 0, on obtient effectivement un estimateur de la taille du sous-espace.

Deux algorithmes sont discutés : une méthode MCMC et un EM associé à des approximations variationnelles. Bishop observe empiriquement qu'un effet de *pruning* apparaît : certaines variances α_k prennent automatiquement la valeur 0. Toutefois, la méthode est peu robuste aux données aberrantes, et un biais apparaît lorsque le nombre de données augmente.

Un modèle hiérarchique similaire a été précédemment proposé par MacKay (1995) dans une formulation bayésienne des réseaux de neurones. Les \mathbf{w}_k sont les vecteurs de poids de chaque étage d'un réseau de neurones et les α_k représentent l'incertitude sur ces poids.

4.1.2.b Modèle bayésien hiérarchique paramétrique avec *a priori* sur K

Pour manipuler la variable aléatoire discrète K , plusieurs approches considèrent un *a priori* hiérarchique de la forme $p(\mathbf{W}|K)P[K]$. Nous évoquons deux méthodes associant ce modèle hiérarchique à une *décomposition en valeurs singulières* (SVD) des facteurs latents \mathbf{W} .

Minka (2000) procède à une approximation de Laplace de la vraisemblance, permettant d'approcher analytiquement la vraisemblance marginale $p(\mathbf{Y}|K)$. Il s'agit d'une version adaptée pour ce modèle du critère BIC vu précédemment.

Smídl et Quinn (2007) poursuivent le raisonnement bayésien en choisissant une loi *a priori* uniforme sur le paramètre K et définissent un estimateur du *maximum a posteriori* (MAP). Une approximation variationnelle de la loi *a posteriori* est cette fois-ci utilisée afin de simplifier le problème.

Les approximations variationnelles et de Laplace permettent de calculer analytiquement des estimateurs approchés. Mais les calculs sont plus difficiles à réaliser lorsque le modèle d'analyse (donné ici en (4.2)) est plus complexe.

À l'inverse, les méthodes MCMC permettent d'approcher des estimateurs bayésiens exacts. Ici, la difficulté réside dans la construction d'une chaîne de Markov dont les états, *i.e.*, les modèles, n'ont pas tous la même dimension. Un outil pour construire une telle chaîne est une méthode de *Monte Carlo par chaîne de Markov à sauts réversibles* (RJMCMC), proposée par Green (1995).

Considérons L modèles différents $\mathcal{M}_1, \dots, \mathcal{M}_L$, où L est potentiellement infini. Les modèles, indexés par ℓ , sont paramétrés par d_ℓ paramètres θ_ℓ à valeurs dans \mathbb{R}^{d_ℓ} . Pour la réduction de dimension, le modèle \mathcal{M}_ℓ caractérise par exemple un sous-espace de dimension ℓ . Une chaîne de

Markov à saut réversible est une chaîne de Markov où l'espace d'état est $\bigcup_{\ell=1}^L \mathbb{R}^{d_\ell} \times \{1, \dots, L\}$. Un état de la chaîne est un couple $(\boldsymbol{\theta}, \ell)$ et la distribution stationnaire la loi jointe sur $\boldsymbol{\theta}$ et ℓ . Green (1995) propose d'utiliser un algorithme de Metropolis Hastings pour modéliser les sauts réversibles. Ces sauts sont réalisés entre des espaces de tailles éventuellement différentes. Or, un saut réversible nécessite la définition d'une bijection entre les espaces d'arrivée et de départ, ce qui est en l'état impossible. Pour résoudre ce problème, Green a proposé d'ajouter des variables auxiliaires, sans interprétation physique, afin d'égaliser la taille des espaces. La probabilité d'accepter le saut fait alors intervenir le Jacobien de la bijection.

Des modèles d'estimation de sous-espaces utilisant les méthodes RJMCMC ont été utilisés par exemple par Zhang et al. (2004) en apprentissage automatique, et par Dobigeon et al. (2008) pour le démélange hyperspectral (dont nous discuterons une application dans la partie 4.6.2) pour réaliser conjointement le démélange et l'estimation du nombre de spectres purs. Malgré de bons résultats empiriques, ces méthodes sont difficiles à mettre en place, l'expression du Jacobien est souvent difficile à obtenir et numériquement coûteuse.

4.1.3 Positionnement du problème et objectif du chapitre

Dans cette partie, nous avons étudié l'estimation de sous-espace formulée comme un problème à variables latentes, où la variable non observée est la base du sous-espace. Pour estimer la dimension du sous-espace, les algorithmes issus des approches paramétriques ont recours à des approximations de la loi *a posteriori* ou à des méthodes d'échantillonnage de type RJMCMC.

L'objectif de ce travail est d'explorer le potentiel des méthodes bayésiennes non paramétriques pour l'estimation de sous-espace. Nous avons vu dans la partie précédente que l'ACP peut être formulée de façon probabiliste en utilisant un modèle à variables latentes. Or, nous avons étudié au chapitre 3 une loi *a priori* non paramétrique pour les modèles à variables latentes : le *processus du buffet indien* (IBP).

Les avantages du recours à un modèle non paramétrique sont multiples. L'IBP pénalise naturellement le nombre de paramètres, *i.e.*, la dimension K du sous-espace ici. Les modèles non paramétriques incluent directement une infinité dénombrable de paramètres, mais seul un nombre fini sont activés à chaque itération. L'échantillonnage de K est fait implicitement via le nombre de composantes actives, et ne nécessite plus le calcul de Jacobien comme pour les méthodes RJMCMC. Nous verrons également que la méthode proposée est suffisamment flexible pour être facilement adaptée à d'autres modèles.

De manière surprenante, peu de modèles s'appuyant sur des méthodes non paramétriques ont été proposées pour l'estimation de sous-espace. Par exemple, Zare et Gader (2008) se sont inspirés des processus de Dirichlet à mélange pour proposer un algorithme glouton dans le cadre

du démixage hyperspectral, où le nombre de spectres purs n'est pas fixé à l'avance. Notons que Wang et Zhu (2015) ont récemment proposé d'étendre les mélanges de PPCA, initialement proposé par Tipping et Bishop (1999a), en s'appuyant aussi sur les processus de Dirichlet à mélange.

4.2 MODÈLE BNP-PCA POUR L'APPRENTISSAGE DE SOUS-ESPACES

4.2.1 Modèle

Notons $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$ la matrice de dimensions $D \times N$ dont chaque colonne représente une observation $\mathbf{y}_n = [y_{1,n} \dots y_{D,n}]^T$. On supposera sans perte de généralité que les observations sont centrées, i.e., $\bar{\mathbf{y}} \triangleq \frac{1}{N} \sum_{n=1}^N \mathbf{y}_n$ est le vecteur nul.

On suppose que les observations \mathbf{Y} vivent dans un sous-espace de dimension $K \leq D$ inconnue. On cherche dans ce travail à inférer conjointement le sous-espace et sa dimension. Dans ce but, considérons le modèle à variables latentes suivant

$$\forall n \in \{1, \dots, N\}, \quad \mathbf{y}_n = \mathbf{P}(\mathbf{z}_n \odot \mathbf{x}_n) + e_n \quad (4.4)$$

où $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_D]$ est une base orthonormée de \mathbb{R}^D , i.e., $\mathbf{P}^T \mathbf{P} = \mathbb{I}_D$, $\mathbf{z}_n = [z_{1,n}, \dots, z_{D,n}]^T$ est un vecteur binaire ($\mathbf{z} \in \{0, 1\}^D$), $\mathbf{x}_n = [x_{1,n}, \dots, x_{D,n}]^T$ est un vecteur de coefficients réels. On rassemble respectivement les vecteurs \mathbf{z}_n et \mathbf{x}_n dans les matrices \mathbf{Z} et \mathbf{X} de dimensions $D \times N$. Le symbole \odot désigne le produit terme-à-terme, ou produit de Hadamard. Le dernier terme e_n de l'équation (4.4) est un bruit additif, modélisant à la fois l'imprécision du modèle et le bruit de mesure. On le supposera gaussien, de matrice de covariance $\sigma^2 \mathbb{I}_D$. Notons qu'un modèle paramétrique similaire a déjà été étudié par Besson et al. (2011) pour régulariser une SVD lorsqu'il n'y a pas suffisamment d'observations.

La recherche de sous-espaces est posée comme un modèle à variables latentes. Mais contrairement à PPCA, on force l'orthogonalité des facteurs latents \mathbf{P} dans (4.4). Les vecteurs orthogonaux \mathbf{p}_k font référence aux composantes principales dans la terminologie de l'ACP. Notons que c'est la parcimonie des variables binaires $z_{k,n}$ pour $k \in \{1, \dots, D\}$ qui réduira le nombre de composantes orthogonales nécessaires pour reconstruire l'observation \mathbf{y}_n . On espère que certaines lignes de la matrice \mathbf{Z} ne comporteront que des 0, signifiant que les directions \mathbf{p}_k ne sont pas pertinentes.

4.2.2 Vraisemblance et choix des lois *a priori*

La fonction de vraisemblance s'obtient de manière classique en utilisant l'additivité du bruit blanc. Considérons N observations indépendantes rassemblées dans la matrice \mathbf{Y} . Alors, la vraisemblance est donnée par

$$f(\mathbf{Y} | \mathbf{P}, \mathbf{Z}, \mathbf{X}, \sigma^2) = (2\pi\sigma^2)^{-DN/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{n=1}^N \|\mathbf{y}_n - \mathbf{P}(\mathbf{z}_n \odot \mathbf{x}_n)\|_2^2\right). \quad (4.5)$$

Les paramètres du modèle associés à la fonction de vraisemblance sont la base orthonormée \mathbf{P} , la matrice binaire \mathbf{Z} , les coefficients \mathbf{X} et la variance du bruit σ^2 . Notons $\theta \triangleq (\mathbf{P}, \mathbf{Z}, \sigma^2)$ l'ensemble des paramètres. La matrice \mathbf{X} en est exclue car ce paramètre sera plus tard marginalisé. Cette partie détaille les lois *a priori* que nous proposons sur θ et \mathbf{X} .

Base orthonormée \mathbf{P} . Par définition du modèle, \mathbf{P} est une matrice orthonormée, élément du groupe orthogonal $\mathcal{O}_D \triangleq \left\{ \mathbf{U} \in \mathbb{R}^{D \times D}, \mathbf{U}^T \mathbf{U} = \mathbb{I}_D \right\}$. Comme aucune autre information n'est disponible *a priori*, on choisit une loi uniforme sur le groupe orthogonal. La densité de cette loi uniforme par rapport à la mesure de Lebesgue est

$$p_{\mathbf{U}}(\mathbf{P}) = \frac{1}{\text{vol}(\mathcal{O}_D)} \mathbb{1}_{\mathcal{O}_D}(\mathbf{P}), \quad (4.6)$$

où (Muirhead, 1982, p. 70)

$$\text{vol}(\mathcal{O}_D) = \frac{2^K \pi^{\frac{D^2}{2}}}{\pi^{\frac{1}{4}K(K-1)} \prod_{i=1}^D \Gamma\left(\frac{D-i+1}{2}\right)}. \quad (4.7)$$

Matrice binaire \mathbf{Z} . La matrice \mathbf{Z} résume l'information d'utilisation des composantes de la base orthonormée \mathbf{P} par les observations. Comme les observations \mathbf{Y} sont supposées vivre dans un sous-espace de \mathbb{R}^D , on s'attend à ce que plusieurs lignes de \mathbf{Z} ne contiennent que des zéros. Une ligne de zéros correspond en effet à une direction ne portant pas d'information. Comme on ne connaît pas à l'avance le nombre de lignes de zéros, on choisit une loi *a priori* de type **IBP** décrite au chapitre 3. Notons que le nombre de lignes contenant au moins un 1, noté K , est *a priori* potentiellement infini, mais sera ici forcément inférieur à D car la matrice \mathbf{P} est orthonormée. On a aussi vu au chapitre précédent que l'**IBP** pénalise naturellement le nombre de paramètres, *i.e.*, la dimension K ici. Cette propriété est intéressante en vue d'une réduction de dimension.

Coefficients \mathbf{X} . On choisit des lois *a priori* normales centrées pour les coefficients, à l'instar de **PPCA**. Notons que ce choix peut se justifier en exploitant le théorème de la limite centrale. En effet, Les coefficients $x_{k,n}$ résultent de la somme d'un grand nombre de variables aléatoires par la projection orthogonale. Nous exploiterons comme **PPCA** la conjugaison de la loi normale avec la fonction de vraisemblance donnée en (4.5) pour marginaliser ce paramètre. Afin de mesurer directement la pertinence d'une direction \mathbf{p}_k , on règle la variance *a priori* des coefficients suivant une loi dite de Zellner (Punskaya et al., 2002)

$$\forall k \in \mathbb{N}, \quad \mathbf{x}_k | \delta_k^2, \sigma^2 \sim \mathcal{N}(0, \delta_k^2 \sigma^2 \mathbb{I}_N) \quad (4.8)$$

Selon cette formulation, l'hyperparamètre δ_k^2 correspond au rapport entre les valeurs propres d'une **ACP** et la variance du bruit. Ainsi, lorsqu'un paramètre d'échelle δ_k^2 est plus grand que 1, cela signifie que la composante associée porte une information dont la puissance est plus importante que celle du bruit.

Variance du bruit σ^2 . On choisit de manière classique une loi non informative de Jeffreys pour la variance du bruit

$$\pi(\sigma^2) \propto \frac{1}{\sigma^2} \mathbb{1}_{\mathbb{R}_+}(\sigma^2). \quad (4.9)$$

Hyperparamètres. Le modèle décrit ci-dessus fait intervenir deux hyperparamètres : le paramètre α de l'IBP contrôlant le nombre de composantes et le vecteur des paramètres d'échelle $\delta = \{\delta_1^2, \dots, \delta_K^2\}$. Notons $\phi \triangleq \{\delta, \alpha\}$ l'ensemble des hyperparamètres. On modélise également l'incertitude autour de ces hyperparamètres par deux lois *a priori*.

On choisit pour α une loi non informative de Jeffreys (voir annexe B.3) sans information supplémentaire sur la taille du sous-espace

$$\pi(\alpha) \propto \frac{1}{\alpha} \mathbb{1}_{\mathbb{R}_+}(\alpha). \quad (4.10)$$

Nous marginaliserons dans la prochaine partie la loi *a posteriori* par rapport aux coefficients \mathbf{X} . La loi conjuguée pour le paramètre d'échelle sera la loi *loi Inverse Gamma translatée (sIG)*, donnée par

$$\pi(\delta_k^2 | a_\delta, b_\delta) = \frac{b_\delta^{a_\delta}}{\gamma(a_\delta, b_\delta)} \left(\frac{1}{1 + \delta_k^2} \right)^{a_\delta+1} \exp\left(-\frac{b_\delta}{1 + \delta_k^2} \right) \mathbb{1}_{\mathbb{R}_+}(\delta_k^2), \quad (4.11)$$

où $\gamma(a, b)$ désigne la fonction Gamma incomplète¹ et a_δ, b_δ sont deux hyperparamètres positifs contrôlant la forme de la loi sIG. On fixe $(a_\delta, b_\delta) = (1, 0.1)$ dans le but d'obtenir une loi vague. Notons que la version non informative de la loi sIG reviendrait à prendre $(a_\delta, b_\delta) = (0, 0)$, ce qui est interdit ici car la loi *a posteriori* serait impropre. En effet, la mesure sur l'espace des paramètres doit être finie pour un modèle non paramétrique². La loi sIG a été récemment introduite dans le cadre du débruitage de signaux audios dans Godsill (2010) et la caractérisation de la distribution de masse d'un ensemble d'étoile dans Zaninetti (2013). L'annexe B.2 rappelle quelques propriétés de la loi sIG.

1. $\gamma(a, b) = \int_0^b t^{a-1} e^{-t} dt$.

2. cf. partie 3.2.2. Si la mesure G_0 sur l'espace des paramètres n'est pas finie, alors l'espérance dans (3.43) n'est plus finie. Cela signifie que chaque observation utilisera presque sûrement une infinité de paramètres.

4.3 ALGORITHME MCMC

CETTE PARTIE décrit l'échantillonneur de Gibbs proposé pour générer des échantillons suivant la loi *a posteriori* marginalisée $p(\theta, \phi | \mathbf{Y})$. Nous discuterons des estimateurs bayésiens dans la prochaine partie.

4.3.1 Loi *a posteriori* marginalisée

Marginaliser la loi *a posteriori* par rapport à un paramètre permet de réduire la dimension de l'espace des paramètres à explorer. Les algorithmes MCMC sont généralement plus performants (D. A. van Dyk et Park, 2008). Puisque l'on cherche à estimer une base d'un sous-espace seulement, la matrice de coefficients \mathbf{X} peut être vue comme un paramètre de nuisance. Nous profitons des propriétés de conjugaison de la loi normale pour

marginaliser la loi *a posteriori* du modèle vis-à-vis de ce paramètre. La loi *a posteriori* marginalisée s'obtient en calculant

$$f(\boldsymbol{\theta}, \boldsymbol{\phi} | \mathbf{Y}) = \int_{\mathbb{R}^{DN}} f(\mathbf{Y} | \boldsymbol{\theta}, \mathbf{X}) f(\boldsymbol{\theta}, \mathbf{X} | \boldsymbol{\phi}) f(\boldsymbol{\phi}) d\mathbf{X}. \quad (4.12)$$

On obtient (voir annexe B.4 pour les détails)

$$\begin{aligned} f(\boldsymbol{\theta}, \boldsymbol{\phi} | \mathbf{Y}) &= \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{ND}{2}} \exp\left(-\frac{\text{tr}(\mathbf{Y}^T \mathbf{Y})}{2\sigma^2}\right) \\ &\times \prod_{k=1}^D \exp\left[\frac{1}{2\sigma^2} \frac{\delta_k^2}{1 + \delta_k^2} \sum_n z_{k,n} \langle \mathbf{p}_k, \mathbf{y}_n \rangle^2\right] \\ &\times \prod_{k=1}^D \left(\frac{1}{1 + \delta_k^2} \right)^{a_\delta + \frac{1}{2} \sum_n z_{k,n}} \exp\left(-\frac{b_\delta}{1 + \delta_k^2}\right) \\ &\times \frac{\alpha^K}{\prod_k K_n!} e^{-\alpha \sum_n \frac{1}{i}} \prod_k \frac{(N - m_k)! (m_k - 1)!}{N!} \\ &\times \left(\frac{b_\delta^{a_\delta}}{\gamma(a_\delta, b_\delta)} \right)^K (\sigma^2)^{-1} \alpha^{-1} \mathbb{1}_{\mathcal{O}_D}(\mathbf{P}), \end{aligned} \quad (4.13)$$

où K est le nombre de lignes actives dans \mathbf{Z} . À ce niveau, il est encore possible de marginaliser vis-à-vis d'un sous ensemble des paramètres d'échelle $\boldsymbol{\delta}$, mais l'échantillonneur de Gibbs devient plus difficile à mettre en place. Nous exploiterons cette propriété plus tard.

On remarque aussi que la quantité $\exp\left(-\frac{\text{tr}(\mathbf{Y}^T \mathbf{Y})}{2\sigma^2} + \frac{1}{2\sigma^2} \frac{\delta_k^2}{1 + \delta_k^2} \sum_n z_{k,n} \langle \mathbf{p}_k, \mathbf{y}_n \rangle^2\right)$ n'est rien d'autre qu'une loi normale multivariée centrée pour la variable \mathbf{y}_n de matrice de covariance

$$\Sigma_n^{-1} = \frac{1}{\sigma^2} \left(\mathbb{I}_D - \sum_{k=1}^D \frac{\delta_k^2}{1 + \delta_k^2} z_{k,n} \mathbf{p}_k \mathbf{p}_k^T \right). \quad (4.14)$$

La loi de Kailath Bishop (2006) donne directement la forme de la matrice de covariance

$$\Sigma = \sigma^2 \left(\mathbb{I}_D + \sum_{k=1}^D \delta_k^2 z_{k,n} \mathbf{p}_k \mathbf{p}_k^T \right). \quad (4.15)$$

Comme \mathbf{P} vérifie $\mathbf{P}^T \mathbf{P} = \mathbb{I}_D$, les valeurs singulières de $\sum_{k=1}^D z_{k,n} \mathbf{p}_k \mathbf{p}_k^T$ ne sont que des 1 et des 0. On obtient la constante de normalisation pour une observation \mathbf{y}_n

$$(2\pi\sigma^2)^{D/2} \prod_{k=1}^D (1 + \delta_k^2)^{1/2}.$$

ce qui est cohérent avec équation (4.13).

4.3.2 Algorithme MCMC

Nous décrivons dans cette partie l'échantillonneur de Gibbs générant des échantillons asymptotiquement distribués suivant (4.13). On rappelle qu'un échantillonneur de Gibbs nécessite d'explicitier les loi *a posteriori* conditionnelles de chaque paramètres. L'échantillonnage de $p(\mathbf{Z} | \mathbf{Y}, \mathbf{P}, \delta^2, \sigma^2)$ résumée par l'algorithme 4.2 est de loin l'étape la plus technique. Par conjugaison, l'échantillonnage des autres paramètres et hyperparamètres est relativement classique. L'algorithme 4.1 résume toute la procédure.

Algorithme 4.1 : Échantillonneur de Gibbs du modèle BNP-PCA

Entrées : $\mathbf{Y}, n_{\text{mc}}$

```

1 pour  $t \leftarrow 1$  à  $n_{\text{mc}}$  faire
  // Activation de nouvelles directions
2 Générer  $\mathbf{Z}^{(t)}$  et  $\mathbf{P}^{(t-\frac{1}{2})}$  comme décrit à l'algorithme 4.2;
  // Mise à jour des directions actives.
3 pour  $k \leftarrow 1$  à  $K$  faire
4   Calculer  $\mathbf{N}_{K \setminus k}$ , une base de  $\mathbf{P}_{\setminus k}^{\perp (t-\frac{1}{2})}$ ;
5   Générer  $\mathbf{v}_k$  selon (4.22);
6    $\mathbf{p}_k^{(t)} = \mathbf{N}_{K \setminus k} \mathbf{v}_k$ ;
7   Générer  $\delta_k^2(t)$  selon (4.25);
8 fin
  // Mise à jour des hyperparamètres.
9 Générer  $\sigma^2(t)$  selon (4.23);
10 Générer  $\alpha^{(t)}$  selon (4.24);
11 fin

```

Output : Une collection d'échantillons

$$\left\{ \mathbf{P}^{(t)}, \mathbf{Z}^{(t)}, \delta_k^2(t), \sigma^2(t), \alpha^{(t)} \right\}_{t=1}^{n_{\text{mc}}}$$
 asymptotiquement distribués selon (4.13).

Matrice binaire \mathbf{Z} . Notons $m_k(n) \triangleq \sum_{\ell \neq n} z_{k,\ell}$ le nombre d'observations, \mathbf{y}_n exclue, qui utilisent la direction \mathbf{p}_k . Les indices des directions pour lesquelles $m_k(n) = 0$ sont appelés *singletons*. Notons \mathcal{J}_n l'ensemble des singletons et \mathcal{I}_n l'ensemble des non-singletons, *i.e.*, l'ensemble des indices tels que $m_k(n) > 0$. Notons que $\mathcal{I}_n \cup \mathcal{J}_n = \{1, \dots, K\}$.

Les coefficients $z_{k,n}$ dont les indices ne correspondent pas à des singletons sont mis à jour à travers une étape d'échantillonnage de Gibbs où les paramètres d'échelles δ_k^2 peuvent être marginalisés. Les loi conditionnelles résultantes sont des lois de Bernoulli de paramètres

$$\frac{\mathbb{P}(z_{k,n} = 1 | \mathbf{Y}, \mathbf{P}, \sigma^2)}{\mathbb{P}(z_{k,n} = 0 | \mathbf{Y}, \mathbf{P}, \sigma^2)} = \frac{m_k(n)}{N - 1 - m_k(n)} \exp\left(\frac{1}{2\sigma^2} \langle \mathbf{p}_k, \mathbf{y}_n \rangle^2\right) \times \frac{\gamma\left(a + 1, b + \frac{1}{2\sigma^2} \langle \mathbf{p}_k, \mathbf{y}_n \rangle^2\right)}{\gamma(a, b)} \frac{b^a}{\left(b + \frac{1}{2\sigma^2} \langle \mathbf{p}_k, \mathbf{y}_n \rangle^2\right)^{a+1}} \quad (4.16)$$

où

$$a = a_\delta + \sum_{i=1, i \neq n}^N z_{k,i} \quad (4.17)$$

$$b = b_\delta + \frac{1}{2\sigma^2} \sum_{i=1, i \neq n}^N z_{k,i} \langle \mathbf{p}_k, \mathbf{y}_i \rangle^2. \quad (4.18)$$

Pour les singletons, la mise à jour doit être réalisée conjointement avec l'activation et la suppression de nouvelles directions pour que la chaîne de Markov soit valide (Knowles et Ghahramani, 2011). Knowles et Ghahramani (2011) proposent de passer par une étape de Metropolis Hastings. Notons $\kappa = \text{card}(\mathcal{J}_n)$ le nombre de singletons et $\mathbf{P}_{\mathcal{J}_n} \triangleq [\tilde{\mathbf{p}}_1, \dots, \tilde{\mathbf{p}}_\kappa]$ la

sous-matrices de \mathbf{P} obtenue en retenant les colonnes indexées par \mathcal{J}_n . De même, on note $\mathbf{P}_{\mathcal{I}_n}$ la sous-matrice de \mathbf{P} obtenue en retenant les colonnes de \mathbf{P} indexées par \mathcal{I}_n . L'étape de Metropolis Hastings propose de passer de l'état $\mathcal{E} = \{\kappa, \mathbf{P}_{\mathcal{J}_n}\}$ au nouvel état $\mathcal{E}^* = \{\kappa^*, \mathbf{P}_{\mathcal{J}_n}^*\}$. Considérons la loi de proposition hiérarchique

$$q(\kappa^*, \mathbf{P}_{\mathcal{J}_n}^* | \kappa, \mathbf{P}_{\mathcal{J}}, \mathbf{P}_{\mathcal{I}}) = q(\kappa^* | \mathbf{P}_{\mathcal{I}}) q(\mathbf{P}_{\mathcal{J}_n}^* | \kappa^*, \mathbf{P}_{\mathcal{I}}). \quad (4.19)$$

On tire d'abord un nouveau nombre de singletons κ^* , et, sachant ce nombre de singletons, on simule de nouvelles directions. Remarquons que la loi de proposition équation (4.19) est conditionnée à $\mathbf{P}_{\mathcal{I}_n}$. Ce conditionnement est possible puisque l'on cherche à échantillonner la distribution $f(\mathbf{Z}, \mathbf{P}_{\mathcal{J}} | \mathbf{Y}, \mathbf{P}_{\mathcal{I}}, \sigma^2)$. On choisit une loi de Poisson $\mathcal{P}(\alpha)$ renormalisée comme loi de proposition du nouveau nombre de singletons. En effet, la loi de Poisson $\mathcal{P}(\alpha/N)$ est proche de la loi du nombre de singletons sur une ligne. L'idée de la renormalisation est de rarement proposer trop de nouvelles directions, et de favoriser la désactivation de directions via la masse en 0. On choisit d'ajouter une masse $\text{card}(\mathcal{I}_n)/D$ ponctuelle en zéro, suivant

$$q(\kappa^* | \mathbf{P}_{\mathcal{I}}) = \frac{\text{card}(\mathcal{I}_n)}{D} \delta_0(\kappa) + \left(1 - \frac{\text{card}(\mathcal{I}_n)}{D}\right) \mathcal{P}(\alpha). \quad (4.20)$$

Notons que c'est à travers la gestion des singletons que l'on active / désactive de nouvelles directions, *i.e.*, que l'on explore des sous-espaces de dimensions différentes. Lorsque le nombre de singletons augmente, on explore des sous-espaces de plus grande dimension. À l'inverse, lorsque le nombre de singletons diminue, on réduit la dimension. Une fois que κ^* est choisi, une nouvelle matrice \mathbf{Z}^* est construite en concaténant les lignes indexées par \mathcal{I}_n et κ^* lignes de 0 avec un 1 à la position n .

Une distribution de von Mises-Fisher matricielle $\text{vMF}(\mathbf{C})$ (cf. annexe B.1) est choisie pour proposer κ^* nouvelles directions actives $\mathbf{P}_{\mathcal{J}}$. Le paramètre \mathbf{C} est construit à partir de la projection des observations dans l'orthogonal des directions existantes. En particulier, les colonnes de \mathbf{C} sont les κ^* vecteurs propres associés aux plus grandes valeurs propres de la projection de $\mathbf{Y}\mathbf{Y}^T$ sur l'orthogonal de $\mathbf{P}_{\mathcal{I}_n}$. Les colonnes de \mathbf{C} sont ensuite multipliées par les valeurs propres correspondantes. Ainsi, les directions proposées se situent autour des vecteurs obtenues avec une ACP classique.

La figure 4.1 illustre le fonctionnement de la loi de proposition des nouvelles directions sur un exemple simple où $\kappa^* = 1$ et l'orthogonal du sous-espace courant est de dimension 2.

Le mouvement $\mathcal{E} \rightarrow \mathcal{E}^*$ est finalement accepté avec probabilité

$$u_{\mathcal{E} \rightarrow \mathcal{E}^*} = \frac{f(\mathbf{Y} | \mathbf{P}_{\mathcal{I}_n}, \mathbf{P}_{\mathcal{J}_n}^*, \mathbf{Z}^*, \sigma^2) p(\mathcal{E}^*) q(\mathcal{E} | \mathcal{E}^*, \mathbf{P}_{\mathcal{I}_n})}{f(\mathbf{Y} | \mathbf{P}_{\mathcal{I}_n}, \mathbf{P}_{\mathcal{J}_n}, \mathbf{Z}, \sigma^2) p(\mathcal{E}) q(\mathcal{E}^* | \mathcal{E}, \mathbf{P}_{\mathcal{I}_n})}. \quad (4.21)$$

La méthode est résumée par l'algorithme 4.2. Pour finir, nous soulignons que l'échantillonnage de \mathbf{Z} traite simultanément une partie de \mathbf{P} .

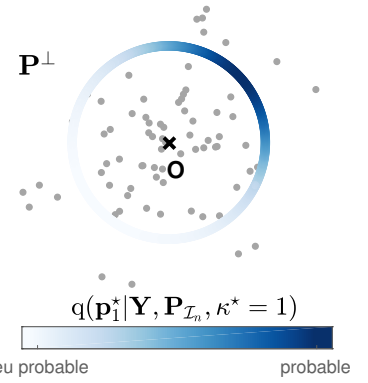


FIGURE 4.1 : Illustration sur un exemple synthétique simple du fonctionnement de la loi de proposition des nouvelles directions. Ici, la dimension de l'orthogonal du sous-espace à l'itération courante est 2. L'espace représenté est l'orthogonal du sous-espace courant dont une base est $\mathbf{P}_{\mathcal{I}_n}$. Les points gris sont les projections des observations \mathbf{Y} . Le cercle de couleur symbolise la densité de la loi de proposition pour $\kappa^* = 1$ direction.

Algorithme 4.2 : Échantillonnage détaillé de la matrice binaire \mathbf{Z}

Entrées : $\mathbf{Y}, \mathbf{Z}^{(t-1)}, \mathbf{P}^{(t-1)}, \sigma^{2(t-1)}, \delta_k^{2(t-1)}$

```

1  $\mathbf{P}^{(t-\frac{1}{2})} = \mathbf{P}^{(t-1)}$ ;
2 pour  $n \leftarrow 1$  à  $N$  faire
   // Identification des singletons
3   pour  $k \leftarrow 1$  à  $K$  faire
4     Calculer  $m_k(n) = \sum_{\ell \neq n} z_{k,\ell}^{(t-1)}$ ;
5   fin
6    $\mathcal{I}_n \triangleq \{k, m_k(n) > 0\}$ ;
7    $\mathcal{J}_n \triangleq \{k, m_k(n) = 0\}$ ;
   // Échantillonnage des directions actives
8   pour chaque  $k$  in  $\mathcal{I}_n$  faire
9     Générer  $z_{k,n}^{(t)}$  selon (4.16);
10  fin
   // Proposition d'un nouveau nombre de singleton
11   $\kappa \triangleq \text{card}(\mathcal{J}_n)$ ;
12   $\mathbf{P}_{\mathcal{I}_n} \triangleq [\mathbf{p}_k, k \in \mathcal{I}_n]$ ;
13  Générer  $\kappa^*$  selon (4.20);
   // Génération itérative de nouvelles directions
14   $\mathbf{P}_{\mathcal{J}_n}^* = []$ ;
15  pour  $k \leftarrow 1$  à  $\kappa^*$  faire
16    Soit  $\mathbf{N}$  une base orthonormée de  $[\mathbf{P}_{\mathcal{I}_n}, \mathbf{P}_{\mathcal{J}_n}^*]$ ;
17    Soit  $\mathbf{v}$  le premier vecteur propre de  $\mathbf{N}^T \mathbf{Y} \mathbf{Y}^T \mathbf{N}$  et  $\lambda$  la valeur
      propre associée;
18    Générer  $\mathbf{p}_k^* \sim \text{vMF}(\mathbf{v}, \lambda)$ ;
19     $\mathbf{P}_{\mathcal{J}_n}^* = [\mathbf{P}_{\mathcal{J}_n}^*, \mathbf{p}_k^*]$ ;
20  fin
   // Étape d'acceptation
21  Calculer  $u_{\mathcal{E} \rightarrow \mathcal{E}^*}$  selon (4.21);
22  Générer  $u \sim \mathcal{U}([0, 1])$ ;
23  si  $u \leq u_{\mathcal{E} \rightarrow \mathcal{E}^*}$  alors
24     $\mathcal{E} = \mathcal{E}^*$  et  $\mathbf{P}^{(t-\frac{1}{2})}$ ;
25     $K = K - \kappa + \kappa^*$ ;
26  fin
27 fin

```

Output : $\mathbf{Z}^{(t)}, \mathbf{P}^{(t-\frac{1}{2})}$.

Base orthonormée \mathbf{P} . L'échantillonnage de la base orthonormée \mathbf{P} est réalisé en deux temps. On échantillonne d'abord les K colonnes ne correspondant pas à des lignes de 0 dans \mathbf{Z} , appelées *colonnes actives*. Les $D - K$ autres colonnes sont dites *inactives*. En effet, comme la loi *a priori* sur \mathbf{P} est une loi uniforme sur la variété de Stiefel, la loi marginale sur les K premières composantes est également une loi uniforme sur la variété de Stiefel $\mathcal{S}_K^D \triangleq \{ \text{matrices } \mathbf{U} \text{ de dimensions } D \times K, \mathbf{U}^T \mathbf{U} = \mathbb{I}_K \}$. De plus, la loi conditionnelle d'une composante est aussi une loi uniforme sur \mathcal{S}_1^D (Hoff, 2009).

Notons $\mathcal{A} \subset \{1, \dots, D\}$ l'ensemble des K indices correspondant aux numéros des colonnes actives de \mathbf{P} . Décomposons \mathbf{P} en deux matrices rectangulaires, *i.e.*, $\mathbf{P} = [\mathbf{P}_{\mathcal{A}}, \mathbf{P}_{\bar{\mathcal{A}}}]$ où $\mathbf{P}_{\mathcal{A}}$ et $\mathbf{P}_{\bar{\mathcal{A}}}$ désignent respectivement les matrices $D \times K$ et $D \times (D - K)$ des directions actives et inactives. Notons également $\mathbf{P}_{\mathcal{A} \setminus k}$ la sous matrice de $\mathbf{P}_{\mathcal{A}}$ obtenue en supprimant la colonne k , et $\mathbf{N}_{\mathcal{A} \setminus k}$ une matrice orthonormée dont les $(D - K + 1)$ colonnes forment une base orthonormée du noyau de $\mathbf{P}_{\mathcal{A} \setminus k}$. Comme $\mathbf{p}_k \in \mathbf{P}_{\mathcal{A} \setminus k}^\perp$, la direction \mathbf{p}_k peut s'écrire comme le produit $\mathbf{p}_k = \mathbf{N}_{\mathcal{A} \setminus k} \mathbf{v}_k$ où \mathbf{v}_k est un vecteur unitaire de dimension $D - K + 1$. Alors

$$f(\mathbf{v}_k | \mathbf{Y}, \mathbf{P}_{\mathcal{A} \setminus k}, \mathbf{Z}, \delta_k^2, \sigma^2) \propto \exp \left(\frac{1}{2\sigma^2} \frac{\delta_k^2}{1 + \delta_k^2} \mathbf{v}_k^T \mathbf{N}_{\mathcal{A} \setminus k}^T \left(\sum_{n=1}^N z_{k,n} \mathbf{y}_n \mathbf{y}_n^T \right) \mathbf{N}_{\mathcal{A} \setminus k} \mathbf{v}_k \right). \quad (4.22)$$

On reconnaît heureusement une distribution de Bingham sur la sphère de dimension $(D - K + 1)$. On reporte le lecteur à l'annexe B.1 pour une discussion sur l'échantillonnage de cette distribution.

Autres paramètres et hyperparamètres. Les lois conditionnelles des paramètres hyperparamètres σ^2 , α et δ s'obtiennent classiquement par conjugaison

$$\sigma^2 | \mathbf{Y}, \mathbf{Z}, \mathbf{P}, \delta \sim \mathcal{IG} \left(\frac{ND}{2}, \frac{1}{2} \text{tr}(\mathbf{Y}\mathbf{Y}^T) - \sum_{k,n} \frac{1}{2} \frac{\delta_k^2}{1 + \delta_k^2} z_{k,n} \langle \mathbf{y}_n, \mathbf{p}_k \rangle^2 \right) \quad (4.23)$$

$$\alpha | \mathbf{Y}, \mathbf{Z} \sim \mathcal{G} \left(K, \sum_{n=1}^N \frac{1}{n} \right) \quad (4.24)$$

$$f(\delta_k^2 | \mathbf{P}, \mathbf{Z}, \sigma^2) \propto \left(\frac{1}{1 + \delta_k^2} \right)^{a_\delta + \frac{1}{2} \sum_n z_{k,n} + 1} \exp \left[- \frac{1}{1 + \delta_k^2} \left(b_\delta + \frac{1}{2\sigma^2} \sum_{k,n} z_{k,n} \langle \mathbf{p}_k, \mathbf{y}_n \rangle^2 \right) \right]. \quad (4.25)$$

4.4 DÉFINITION ET ANALYSE THÉORIQUE DES ESTIMATEURS

CETTE PARTIE discute le comportement théorique des estimateurs de la dimension du sous-espace contenant les données, qui est un des objectifs du modèle BNP-PCA. Nous présentons dans la partie 4.4.1 deux résultats négatifs concernant l'inconsistance de l'estimateur MAP de la dimension K du sous-espace. Nous proposons dans la partie 4.4.2 un nouvel estimateur de K à partir de tests statistiques. La partie 4.4.3 décrit les estimateurs des paramètres et hyperparamètres du modèle.

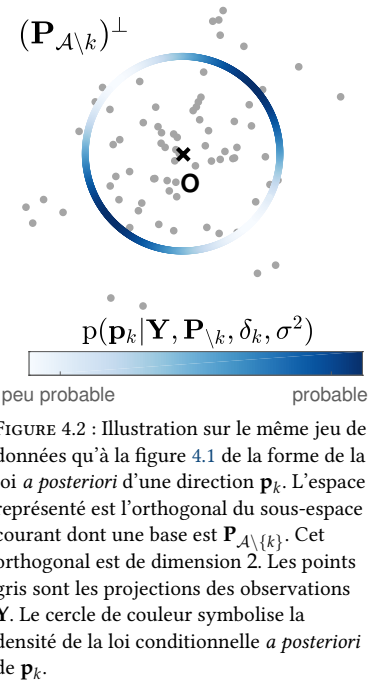


FIGURE 4.2 : Illustration sur le même jeu de données qu'à la figure 4.1 de la forme de la loi *a posteriori* d'une direction \mathbf{p}_k . L'espace représenté est l'orthogonal du sous-espace courant dont une base est $\mathbf{P}_{\mathcal{A} \setminus \{k\}}$. Cet orthogonal est de dimension 2. Les points gris sont les projections des observations \mathbf{Y} . Le cercle de couleur symbolise la densité de la loi conditionnelle *a posteriori* de \mathbf{p}_k .

Le lecteur plus intéressé par les illustrations expérimentales peut se diriger vers les parties 4.5 et 4.6.

4.4.1 Comportement des estimateurs MAP de la dimension

On s'intéresse dans cette partie à l'étude théorique du comportement asymptotique d'un estimateur de type *maximum a posteriori* (MAP) de la dimension du sous espace, défini par

$$\widehat{K}_{\text{MAP}}^\alpha = \operatorname{argmax}_{k \in \{0, \dots, D\}} \mathbb{P}[K = k \mid \mathbf{Y}_N, \alpha]. \quad (4.26)$$

Rappelons que la distribution *a posteriori* est estimée à partir des itérations de l'algorithme MCMC décrit dans la partie 4.3. Nous soulignons un point important ici : le paramètre α est fixé dans (4.26).

Nous avons défini de façon similaire au chapitre 3 un estimateur MAP du nombre de composantes d'un modèle de mélange (cf. partie 3.1.5.b) exploitant le *processus du restaurant chinois* (CRP). Or, nous avons vu que cet estimateur MAP est inconsistant au sens de la définition 3.4 rappelée dans la marge (Miller et Harrison, 2014).

Peu de résultats existent pour l'IBP. Récemment, Chen et al. (2016) se sont intéressés à la consistance de la loi *a posteriori* des paramètres de modèles phylogénétiques. Toutefois, aucun résultat concernant l'estimateur défini en (4.26) n'a été publié à notre connaissance.

Le théorème suivant montre que l'estimateur MAP de la dimension du sous-espace K défini en (4.26) n'est pas consistant. Le résultat est formulé comme dans Miller et Harrison (2014).

Théorème 4.1. Soient \mathbf{Y}_N une matrice concaténant N observations de \mathbb{R}^D et K_N la variable aléatoire associée à la taille du sous-espace inféré par le modèle BNP-PCA (cf. partie 4.2) Alors,

$$\forall k < D \quad \limsup_{N \rightarrow +\infty} \mathbb{P}[K_N = k \mid \mathbf{Y}_N, \alpha] < 1 \quad (4.27)$$

$$\limsup_{N \rightarrow +\infty} \mathbb{P}[K_N = D \mid \mathbf{Y}_N, \alpha] > 0 \quad (4.28)$$

La preuve est disponible en annexe B.5.

Le théorème 4.1 signifie que la loi *a posteriori* du nombre de composante ne se concentrera pas asymptotiquement autour d'une valeur. La raison est une fuite de la masse de la loi *a posteriori* vers l'infini, limitée ici par la contrainte $K \leq D$. C'est le même phénomène que pour le CRP et les modèles de mélange. C'est un résultat négatif. Mais nous soulignons que cela n'implique pas que l'estimateur $\widehat{K}_{\text{MAP}}^\alpha$ défini en (4.26) est biaisé. Mais un corollaire du théorème est que cet estimateur est inconsistant. Ce résultat motivera la définition d'un nouvel estimateur dans la partie 4.4.2. Notons que le premier résultat équation (4.27) peut être étendu à une plus large gamme de modèles à variables latentes dont l'ingrédient principal est l'IBP, tels que Dang et Chainais (2017) en apprentissage de dictionnaire ou Chen et al. (2016) en biostatistique. Le résultat équation (4.28) est propre à ce modèle : il résulte uniquement de la contrainte d'orthogonalité qui impose $K \leq D$ (cf. preuve en annexe B.5).

Définition 3.4 (rappel). Considérons un modèle bayésien $(\mathbb{P}_\theta)_\theta$ paramétré par θ à valeur dans Θ , de dimension potentiellement infinie. Soit $\mathbf{y}_{1:N}$ N observations. La loi *a posteriori* $\mathbb{p}(\theta \mid \mathbf{y}_{1:N})$ notée μ_N est consistante en θ_0 élément de Θ si pour tout voisinage U de θ_0 , la quantité $\mu_N(U)$ converge \mathbb{P}_{θ_0} -presque sûrement vers 1. Par extension, l'estimateur MAP de θ est dit consistant si la loi *a posteriori* $\mathbb{p}(\theta \mid \mathbf{y}_{1:N})$ l'est.

Un résultat d'inconsistance plus fort existe lorsque l'on rajoute des hypothèses sur la distribution des observations \mathbf{Y}_N .

Théorème 4.2. Soient \mathbf{Y}_N la matrice concaténant D observations de \mathbb{R}^D indépendantes et identiquement distribuées, distribuées selon une loi normale centrée de covariance $\sigma_y^2 \mathbb{I}_D$ et K_N la variable aléatoire associée à la taille du sous-espace inféré par le modèle BNP-PCA. Alors

$$\mathbb{P}\left[K_N = 0 \mid \mathbf{Y}_N, \alpha, \sigma_y^2\right] \xrightarrow[N \rightarrow +\infty]{\text{P.s.}} 0. \quad (4.29)$$

La preuve est disponible en annexe B.5.2.

Le théorème 4.2 peut être interprété de deux façons. Comme les observations \mathbf{y} sont distribuées suivant des lois normales $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_D)$, on peut les assimiler à des réalisations du bruit. Dans ce cas, la bonne valeur de la dimension du sous-espace est 0 et le théorème 4.2 affirme que l'estimateur marginal conditionnel

$$\hat{K}_{\text{MAP}}^{\alpha, \sigma_y^2} = \operatorname{argmax}_{k \in \{0, \dots, D\}} \mathbb{P}\left[K = k \mid \mathbf{Y}_N, \alpha, \sigma^2 = \sigma_y^2\right] \quad (4.30)$$

est asymptotiquement inconsistant puisque l'on aimerait retrouver la valeur 0. Nous rappelons qu'à nouveau, on parle d'un estimateur conditionné au paramètre α et à la variance σ_y^2 des observations fixée.

À l'inverse, si les observations \mathbf{Y}_N sont interprétées comme un signal vivant dans \mathbb{R}^D , alors l'équation (4.28) et le théorème 4.2 rassurent quant à la capacité du modèle à identifier asymptotiquement le signal. Mais l'objectif de ce chapitre est de proposer un modèle pour l'identification de sous-espaces. Or cette seconde interprétation met en avant un signal occupant tout l'espace. Pour cette raison, nous écartons volontairement cette seconde interprétation.

La prochaine partie propose un nouvel estimateur de la dimension du sous-espace travaillant sur la distribution *a posteriori* des directions \mathbf{P} . Pour finir, nous soulignons que ces deux résultats concernent des estimateurs MAP conditionnés à la valeur du paramètre de concentration α fixé. En pratique, ce paramètre est intégré au modèle bayésien décrit dans la partie 4.2 et l'échantillonneur de Gibbs présenté dans la partie 4.3.2 permet d'approcher un estimateur du MAPm par rapport à α de la forme

$$\hat{K}_{\text{MAPm}} = \operatorname{argmax}_{k \in \{0, \dots, D\}} \mathbb{P}\left[K = k \mid \mathbf{Y}_N\right]. \quad (4.31)$$

Mais il n'existe pour l'instant aucun résultat théorique concernant cet estimateur. Nous verrons dans la partie 4.5.2 que l'estimateur \hat{K}_{MAPm} semble empiriquement consistant.

4.4.2 Un test statistique pour déterminer la dimension du sous-espace

On a vu dans la partie 4.4.1 que l'estimateur MAP $\hat{K}_{\text{MAP}}^\alpha$ basé sur la distribution *a posteriori* $\mathbb{p}(K \mid \mathbf{Y}, \alpha)$ n'est pas forcément pertinent. Bien qu'en pratique nous approchons l'estimateur \hat{K}_{MAPm} , nous proposons

dans cette partie un nouvel estimateur exploitant des tests statistiques sur la distribution *a posteriori* $p(\mathbf{P}|\mathbf{Y})$. En particulier, nous testerons quelles distribution *a posteriori* des composantes sont uniformes sur la variété de Stiefel à l'aide d'un test de Kolmogorov-Smirnov. Nous verrons en effet qu'une distribution uniforme correspond asymptotiquement à une composante non significative.

Notons $\mathcal{A} \subset \{1, \dots, D\}$ l'ensemble des indices des colonnes de \mathbf{P} correspondant à des directions actives et $\bar{\mathcal{A}}$ son complémentaire dans $\{1, \dots, D\}$. Rappelons qu'une direction \mathbf{p}_k est active si la ligne k de la matrice binaire \mathbf{Z} contient au moins un 1. Notons $\mathbf{P}_{\mathcal{A}}$ et $\mathbf{P}_{\bar{\mathcal{A}}}$ les sous-matrices de \mathbf{P} contenant respectivement les colonnes de \mathbf{P} indexées par \mathcal{A} et $\bar{\mathcal{A}}$. $\mathbf{P}_{\mathcal{A}}$ représente une base du sous-espace contenant le signal, et $\mathbf{P}_{\bar{\mathcal{A}}}$ une base de l'orthogonal. On a vu équation (4.22) que la loi *a posteriori* conditionnelle de chaque direction non active $\mathbf{p}_\ell, \forall \ell \in \bar{\mathcal{A}}$ peut s'écrire $\mathbf{p}_\ell = \mathbf{N}_{\mathcal{A}}\mathbf{w}_\ell$ où \mathbf{w}_ℓ est un vecteur unitaire de dimension $D - L$ et $\mathbf{N}_{\mathcal{A}}$ une base de l'orthogonal de $\mathbf{P}_{\mathcal{A}}$. Alors, la distribution de chaque \mathbf{w}_ℓ s'écrit

$$f(\mathbf{w}_\ell | \mathbf{Y}_N, \mathbf{P}_{\mathcal{A}}, \mathbf{Z}, \delta_k^2, \sigma^2) \propto \exp\left(\frac{\delta_\ell^2}{1 + \delta_\ell^2} \mathbf{w}_\ell^T \boldsymbol{\Lambda}_{\ell, N} \mathbf{w}_\ell\right) \quad (4.32)$$

avec $\bar{L} \triangleq \text{card}\{\bar{\mathcal{A}}\} = D - L$ et

$$\boldsymbol{\Lambda}_{\ell, N} = \sum_{n=1}^N \mathbf{N}_{\mathcal{A}}^T \mathbf{y}_n \mathbf{y}_n^T \mathbf{N}_{\mathcal{A}}. \quad (4.33)$$

La matrice $\boldsymbol{\Lambda}_{\ell, N}$ agit comme la matrice de covariance empirique de la projection des observations dans l'orthogonal de l'espace courant. La loi *a posteriori* jointe de $\mathbf{w}_{\bar{\mathcal{A}}}$, la matrice concaténant les $D - L$ vecteurs unitaires \mathbf{w}_ℓ est donnée par

$$f(\mathbf{w}_{\bar{\mathcal{A}}} | \mathbf{Y}_N, \mathbf{P}_{\mathcal{A}}, \mathbf{Z}, \delta_k^2, \sigma^2) \propto \text{etr}\left(\Delta \mathbf{w}_{\bar{\mathcal{A}}}^T \boldsymbol{\Lambda}_{\ell, N} \mathbf{w}_{\bar{\mathcal{A}}}\right), \quad (4.34)$$

où Δ est une matrice diagonale de taille \bar{L} , dont les entrées sont les quotients $\delta_\ell^2 / (1 + \delta_\ell^2)$. Si le sous-espace $\mathbf{P}_{\mathcal{A}}$ a correctement identifié le signal, alors l'orthogonal de $\mathbf{P}_{\mathcal{A}}$ ne contient que le bruit des observations. Puisque le bruit est isotrope dans le modèle équation (4.4), la matrice $\boldsymbol{\Lambda}_{\ell, N}$ tend vers

$$\lim_{N \rightarrow +\infty} \frac{1}{N} \boldsymbol{\Lambda}_{\ell, N} = \gamma_\ell \sigma^2 \mathbb{I}_{\bar{L}}. \quad (4.35)$$

En d'autres mots, la loi *a posteriori* des directions non significatives $\mathbf{w}_{\bar{\mathcal{A}}}$ est asymptotiquement la loi uniforme sur $\mathcal{O}_{\bar{L}}$.

Nous exploitons ce résultat pour construire une série de tests statistiques pour estimer la dimension du sous-espace. L'idée est de chercher la plus petite dimension L du sous-espace telle que la distribution des matrices orthonormées projetées $\mathbf{w}_{\bar{\mathcal{A}}}$ est la loi uniforme sur le groupe orthogonal $\mathcal{O}_{\bar{L}}$. Comme les matrices orthonormées $\mathbf{w}_{\bar{\mathcal{A}}}$ vivent possiblement en grande dimension et que $K \ll D$, les tests statistiques sur la distribution $\mathbf{w}_{\bar{\mathcal{A}}}$ risquent d'être peu efficaces.

Nous proposons une solution pratique au problème des tests en grande dimension. Considérons la suite des \bar{L} produits scalaires, définis par

$$\omega_\ell \triangleq |\mathbf{w}_\ell^T \mathbf{u}_\ell|, \quad \ell \in \bar{\mathcal{A}}, \quad (4.36)$$

où les vecteurs $\{\mathbf{u}_\ell\}_{\ell \in \bar{\mathcal{A}}}$ représentent $\bar{L} \triangleq D - L$ vecteurs unitaires quelconques de dimension \bar{L} . Puisque les \mathbf{u}_ℓ peuvent être n'importe quel vecteur unitaire, nous suggérons de les tirer aléatoirement sur la sphère de dimension \bar{L} . Le théorème suivant donne la distribution des produits scalaires ω_ℓ sous l'hypothèse que les vecteurs \mathbf{w}_ℓ sont uniformément distribués sur la sphère.

Théorème 4.3. Soient L un entier de $\{0, \dots, D\}$, $\mathbf{w} = [\mathbf{w}_1, \dots, \mathbf{w}_{D-L}]^T$ une matrice aléatoire uniformément distribuée sur le groupe orthogonal \mathcal{O}_{D-L} et $\mathbf{u}_1, \dots, \mathbf{u}_{D-L}$ ($D - L$) vecteurs unitaires de dimension $(D - L)$. Soit $\boldsymbol{\omega} = [\omega_1, \dots, \omega_{D-L}]^T$ le vecteur aléatoire tel que pour tout $\ell \in [1, D - L]$ $\omega_\ell \triangleq |\mathbf{w}_\ell^T \mathbf{u}_\ell|$. Alors, les composantes du vecteur aléatoire $\boldsymbol{\omega}$ sont identiquement distribuées et la fonction de répartition (cdf, pour cumulated distribution function en anglais) d'une composante ω_ℓ est donnée par

$$\begin{aligned} \mathbb{P}(\omega_\ell \leq \lambda) &= \frac{\text{vol}(\mathcal{O}_{L-2})}{\text{vol}(\mathcal{O}_{L-1})} 2 \int_0^\lambda (1 - z^2)^{(L-3)/2} dz \\ &= 2\lambda_l \frac{\text{vol}(\mathcal{O}_{L-2})}{\text{vol}(\mathcal{O}_{L-1})} {}_2F_1\left(\frac{1}{2}, -\frac{L-3}{2}; \frac{3}{2}; \lambda^2\right). \end{aligned} \quad (4.37)$$

La preuve est disponible en annexe B.6. Notons que l'équation (4.37) donne également la densité de probabilité des produits scalaires. La fonction ${}_2F_1$ est une fonction hypergéométrique d'arguments matriciels (Koev et Edelman, 2006). À titre d'illustration, la figure 4.3 compare la forme des distributions empiriques et théoriques données équation (4.37) pour quatre valeurs de $D - L$.

Nous proposons d'utiliser le théorème 4.3 pour estimer la dimension du sous-espace à partir d'une série de tests de Kolmogorov-Smirnov.

La distribution *a posteriori* $p(\mathbf{P}|\mathbf{Y})$ est approchée par les échantillons

$\{\mathbf{P}^{(t)}\}_{t=1}^{n_{\text{mc}}}$ générés avec l'algorithme MCMC décrit partie 4.3.2.

Considérons un ensemble d'indices \mathcal{A} associés aux colonnes de \mathbf{P} , et $\bar{\mathcal{A}}$ son complémentaire du $\{1, \dots, D\}$. Nous utilisons un test de Kolmogorov-Smirnov pour tester si les produits scalaires associés aux colonnes de \mathbf{P} indexées par $\bar{\mathcal{A}}$ sont distribués suivant (4.37). Si tel est le cas, alors les directions indexées par $\bar{\mathcal{A}}$ sont distribuées uniformément, donc non significatives. Le nouvel estimateur est alors défini comme la taille du plus petit sous-ensemble d'indices \mathcal{A} tel que l'ensemble $\bar{\mathcal{A}}$ est associé à des composantes non significatives.

En pratique, on commence par tester la valeur 0, *i.e.*, un signal composé uniquement de bruit.

Nous choisissons de commencer par les petites dimensions de sous-espace car on cherche $K \ll D$. Ensuite, l'ensemble \mathcal{A} des indices des composantes significatives est composé des indices associés aux L plus grands facteurs d'échelles δ_k^2 . L'hypothèse nulle du test est définie comme

$$\mathcal{H}_0^{(L)} : \omega_\ell \stackrel{\text{cdf}}{\sim} (4.37), \quad \forall \ell \in \bar{\mathcal{A}}.$$

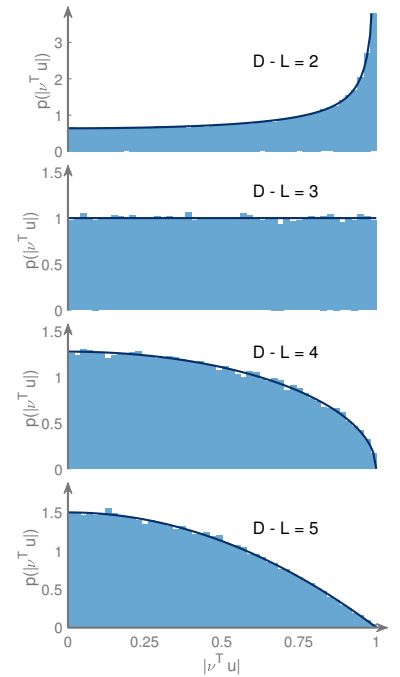


FIGURE 4.3 : Distributions empiriques (histogramme bleu clair, calculé avec 100000 échantillons) et théoriques (courbes bleues foncées) des densités de probabilités associées à l'équation (4.37) pour 4 valeurs différentes de la dimension $D - L$.

Notons que si l'hypothèse nulle n'est pas rejetée pour la valeur L_0 , alors elle ne devrait pas l'être non plus pour $L > L_0$. Inversement, si l'hypothèse nulle est rejetée pour la valeur $L = L_0$, alors elle devrait l'être également pour tout $L < L_0$. Nous choisissons de réaliser le test avec un niveau de confiance de 5%.

Le théorème 4.3 assure que les composantes du vecteur aléatoire $\{\omega_\ell\}_{\ell \in \bar{A}}$ sont identiquement distribuées. Nous en profitons pour réaliser un test statistique par candidat à la dimension en combinant tous les échantillons plutôt que $D - L$ tests³.

Le nouvel estimateur de la dimension du sous-espace, noté \hat{K}_{KS} , est défini par

$$\hat{K}_{KS} = \min \left\{ L \in \{0, \dots, D\} \mid \mathcal{H}_0^{(D-L)} \text{ n'est pas rejetée} \right\}, \quad (4.38)$$

i.e., comme la plus petite valeur de la dimension du sous-espace telle que l'hypothèse nulle n'est pas rejetée. Pour finir, si l'hypothèse nulle $\mathcal{H}_0^{(\ell)}$ est rejetée pour toutes les valeurs de $\ell \in \{1, \dots, D - 1\}$, alors $\hat{K}_{KS} = D$ par convention. Cela signifie que le signal occupe tout l'espace ou que l'algorithme n'a pas été capable de séparer le signal du bruit avec le nombre d'observations considérées et le nombre d'itérations réalisées. Le calcul de l'estimateur \hat{K}_{KS} est résumé à l'algorithme 4.3.

Algorithme 4.3 : Estimer le nombre de directions pertinentes

Entrées : Une collection d'échantillons $\left\{ \mathbf{p}_1^{(t)}, \dots, \mathbf{p}_D^{(t)}, \mathbf{Z}^{(t)} \right\}_{t=T_{bi}+1}^{T_{MC}}$ générée suivant Alg. 4.1.

- 1 À chaque itérations, réordonner les échantillons $\mathbf{p}_k^{(t)}$ par rapport à leur fréquence d'activation donnée par le nombre de 1 dans une ligne de $\mathbf{Z}^{(t)}$;
- 2 **pour** $L \leftarrow 0$ à $D - 1$ **faire**
- 3 Générer $\mathbf{u}_1, \dots, \mathbf{u}_D \stackrel{\text{i.i.d.}}{\sim} \mathcal{S}_{D-L}^1$;
- 4 **pour** $t \leftarrow n_{\text{burn}} + 1$ à $n_{\text{burn}} + n_{\text{iter}}$ **faire**
- 5 Soit \mathbf{N}_L une base de l'orthogonal de $\mathbf{p}_1^{(t)}, \dots, \mathbf{p}_L^{(t)}$;
- 6 Calculer $\omega_{L+1}^{L(t)} \triangleq |\mathbf{p}_{L+1}^{(t)T} \mathbf{N}_L^T \mathbf{u}_{L+1}|, \dots, \omega_D^{L(t)} \triangleq |\mathbf{p}_D^{(t)T} \mathbf{N}_L^T \mathbf{u}_D|$;
- 7 **fin**
- 8 Concaténer les vecteurs $\omega_{L+1}^{L(t)}, \dots, \omega_D^{L(t)}$;
- 9 **si** \mathcal{H}_L *n'est pas rejeté* **alors**
- 10 $\hat{K}_{KS} = L$;
- 11 quitter la boucle;
- 12 **fin**
- 13 **fin**

Output : \hat{K}_{KS} , un estimateur du nombre de composante.

3. Cette concaténation est possible car les composantes sont identiquement distribuées.

Remarque Nous avons discuté dans les deux dernières parties d'estimateurs capables d'identifier la dimension d'un sous-espace ayant servi lors de la synthèse. Or, c'est précisément le rôle de l'estimateur du MMSE (cf. Éq. (I.5)). Néanmoins, cet estimateur ne prend pas en compte le fait que la

valeur estimée doit être entière. Il aurait été intéressant d'étudier numériquement le comportement de cet estimateur, en utilisant par exemple un arrondi à l'entier le plus proche. Comme nous avons proposé un estimateur alternatif \widehat{K}_{KS} , nous n'étudierons pas cet estimateur MMSE.

4.4.3 Estimation des autres paramètres

Cette partie discute l'estimation de la variance du bruit σ^2 , de la base du sous-espace \mathbf{P} des facteurs d'échelle δ_k^2 .

Variance du bruit. La variance du bruit est estimée avec un estimateur MMSE approché selon

$$\widehat{\sigma}_{\text{MMSE}}^2 \simeq \frac{1}{T_{\text{MC}} - T_{\text{bi}}} \sum_{t=T_{\text{bi}}+1}^{T_{\text{MC}}} \sigma^2(t). \quad (4.39)$$

Base du sous-espace. L'estimation de de la base orthonormée \mathbf{P} du sous-espace et des paramètres d'échelles est plus difficile pour deux raisons. D'abord, le nombre de directions associées au signal varie au cours des itérations t . Lorsque $K^{(t)}$ directions sont actives, les $D - K^{(t)}$ directions restantes sont tirées uniformément sur l'orthogonal des directions actives. Nous proposons de calculer un estimateur $\widehat{\mathbf{P}}$ conditionné à un estimateur \widehat{K} de la dimension du sous-espace, cf. la partie précédente pour la définition de ces estimateurs. En pratique, cela revient à n'utiliser *in fine* que les itérations pour lesquelles $K^{(t)} = \widehat{K}$.

La variété de Stiefel n'est pas stable par moyenne arithmétique. Besson et al. (2011, 2012) ont proposé de remplacer la distance euclidienne qui intervient dans la définition de l'estimateur MMSE par une distance sur la variété.

Soient $\mathbf{P}_1, \mathbf{P}_2$ deux matrices de \mathcal{S}_K^D . Une distance naturelle entre les deux matrices est la somme des carrés des angles entre les composantes, que l'on obtient à partir des valeurs singulières de $\mathbf{P}_1^T \mathbf{P}_2$. L'estimateur minimisant la distance angulaire entre un sous-espace estimé et le sous-espace ayant servi à la synthèse est appelé estimateur du *minimum de la distance quadratique* (MMSD). En pratique, on définit conditionnellement à \widehat{K} la matrice

$$\Phi = \sum_{t=T_{\text{bi}}+1}^{T_{\text{MC}}} \sum_{k=1}^{\widehat{K}} \mathbf{p}_k^{(t)} \mathbf{p}_k^{T(t)}. \quad (4.40)$$

L'estimateur $\widehat{\mathbf{P}}_{\widehat{K}}$ s'obtient en prenant les \widehat{K} premiers vecteurs propres de Φ associés aux plus grandes valeurs propres.

Facteurs d'échelle. Conditionnellement à \widehat{K} , on utilise un estimateur de type MMSE pour les paramètres d'échelle δ_k^2 approchés selon

$$\forall k \leq \widehat{K}, \quad \widehat{\delta}_k^2 \simeq \frac{1}{T_{\text{MC}}} \sum_{t=T_{\text{bi}}+1}^{T_{\text{MC}}} \delta_k^2(t). \quad (4.41)$$

4.5 VALIDATION EXPÉRIMENTALE

LE MODÈLE BNP-PCA décrit dans la partie 4.2 est étudié numériquement sur 4 cas d'étude. En particulier, nous verrons que

1. BNP-PCA identifie correctement les paramètres d'échelle et la base du sous-espace sur un cas très simple où le bruit est faible devant le signal,
2. BNP-PCA reconnaît un signal composé uniquement de bruit contrairement à l'estimateur MAP étudié en 4.4.1,
3. l'estimateur MAPm semble consistant contrairement à l'estimateur MAP étudié en 4.4.1,
4. BNP-PCA n'identifie pas un signal en dessous du niveau de bruit.

Le modèle de synthèse est le même pour les quatre expériences et diffère du modèle d'analyse. Pour chaque réalisation, on génère une matrice orthonormée \mathbf{H} uniformément sur le groupe orthogonal \mathcal{O}_D ainsi que n vecteurs de coefficients \mathbf{u}_n réels, de dimension K_S (pour synthèse), selon une loi normale centrée multivariée de matrice de covariance diagonale renormalisée par la variance du bruit σ^2 . Chaque observation est ensuite générée selon

$$\mathbf{y}_n = \mathbf{H}\mathbf{u}_n + \mathbf{e}_n, \quad (4.42)$$

où \mathbf{e}_n est un bruit blanc gaussien additif de matrice de covariance $\sigma^2\mathbb{I}_D$. Puisque les variances des coefficients sont relatives au niveau de bruit dans le modèle d'analyse, on ne considéra qu'une seule valeur de σ^2 , à savoir $\sigma^2 = 0.1$. Tous les résultats sont moyennés sur 20 simulations, et les estimateurs sont calculés sur $T_{MC} = 5000$ itérations après $T_{bi} = 100$ itérations. La dimension D , la taille du sous-espace K_S et le nombre d'observations n seront spécifiés pour chaque expérience.

4.5.1 Un cas simple

Nous illustrons les performances de l'algorithme BNP-PCA sur deux exemples simples où le bruit est faible devant le signal. Les deux scénarios sont ($D = 16, K = 4, N = 100$) et ($D = 36, K = 6, N = 500$). Les facteurs d'échelle sont choisis tels que $\delta_k^2 = 50/k$, où k est le numéro de la composante.

Les figures 4.4-a et 4.4-b montrent la distribution *a posteriori* $p(K|\mathbf{Y})$ pour les deux scénarios. Dans les deux cas, on observe que le maximum des deux histogrammes, *i.e.*, l'estimateur \hat{K}_{MAPm} (cf. Éq. (4.31)), correspond à la dimension du sous-espace K (\sqrt{D} ici) dans les deux cas. Ces résultats n'étaient pas nécessairement attendus depuis que l'on avait vu aux théorèmes 4.1 et 4.2 que les estimateurs MAP conditionnels étaient inconsistants. Nous discuterons plus en détails ce point dans la prochaine expérience. Notons que dans cet exemple très simple, nous ne discuterons pas des résultats de l'estimateur \hat{K}_{KS} .

Les Figures 4.4-c et 4.4-e montrent la dispersion des 8 premiers facteurs d'échelles pour les deux scénarios. Dans les deux cas, les facteurs d'échelles sont correctement identifiés. On observe que les facteurs d'échelles correspondant à des composantes non significatives (les 4 derniers Fig. 4.4-c et les 2 derniers Fig. 4.4-e) semblent identiquement distribués comme prévu, voir Éq. (4.37).

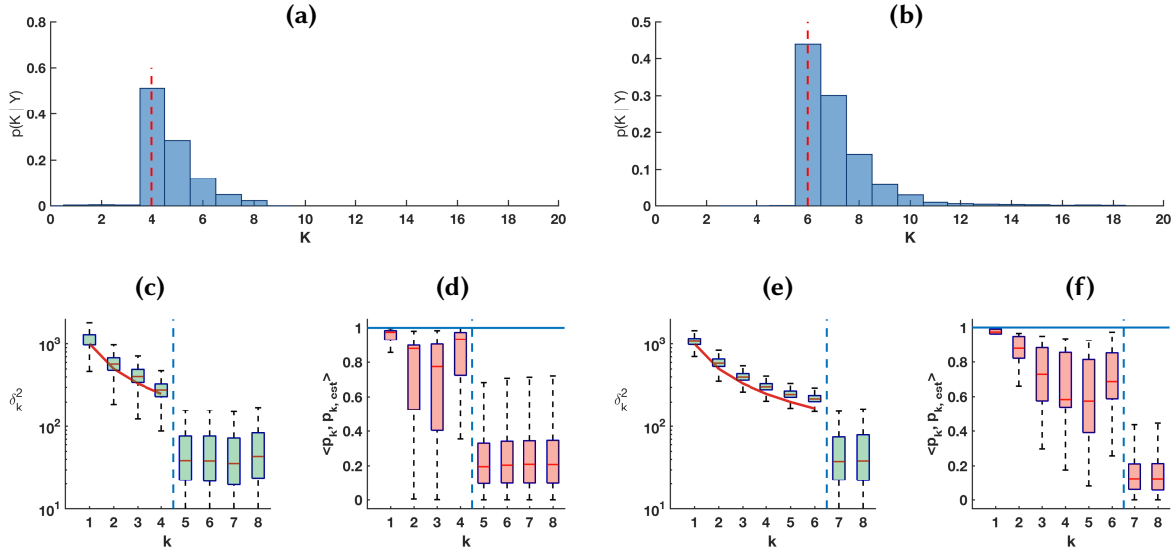


FIGURE 4.4 : Haut : distribution *a posteriori* $K|Y$ pour (a) $D = 16, n = 100$, et (b) $D = 36, n = 500$. Bas : Dispersion des (c) & (e) facteurs d'échelles δ_k^2 et (d) & (f) produits scalaires $\hat{\mathbf{P}}^T \mathbf{P}$ pour $D = 16, n = 100$ et $D = 36, n = 500$. Les lignes rouges indiquent les vraies valeurs de $\delta_1^2, \dots, \delta_K^2$.

Les Figures 4.4-e et 4.4-f montrent la dispersion des 8 premiers alignements entre les directions \mathbf{p}_k estimées et les directions \mathbf{h}_k ayant servi à la synthèse pour les deux scénarios. Ces alignements sont calculés à partir du produit scalaire euclidien $|\langle \mathbf{p}_k, \mathbf{h}_k \rangle|$, qui correspond ici à un cosinus. On observe que les alignements correspondant à des composantes actives sont en moyenne supérieurs à 0.8. En revanche, les alignements correspondant à des composantes non significatives (les 4 dernières Fig. 4.4-c et les 2 dernières Fig. 4.4-e) sont faibles et eux-aussi identiquement distribués autour d'une valeur que l'on peut prédire théoriquement et correspondant à l'espérance de la variable aléatoire ω en (4.37).

Ces observations ont motivé la construction de l'estimateur \hat{K}_{KS} décrit dans la partie 4.4.2. Lorsque qu'une direction est significative, son alignement sera statistiquement élevé (*cf.* les K_S premiers alignements), et ne ressembleront à aucune des distributions du théorème 4.3. En revanche, nous espérons que les tests statistiques seront capables d'identifier la distribution des produits scalaires associés aux composantes non significatives.

Pour conclure, ces premières expériences ont montré que l'algorithme BNP-PCA est capable d'identifier le sous-espace contenant un signal, ainsi que les facteurs d'échelles associés à chaque direction sur un cas simple. Nous avons aussi observé que l'estimateur \hat{K}_{MAPm} identifie la bonne valeur du sous-espace alors que les théorèmes 4.1 et 4.2 donnaient des résultats négatifs pour des estimateurs similaires. Nous clarifierons ce point dans la prochaine partie.

4.5.2 Comportement des estimateurs de la dimension

Cette expérience étudie le comportement de l'estimateur MAPm de la dimension du sous-espace lorsque le nombre d'observations grandit. Rappelons que les théorèmes 4.1 et 4.2 affirment que les estimateurs MAP conditionnels \hat{K}_{MAP}^α et $\hat{K}_{MAP}^{\alpha, \sigma_y^2}$ sont inconsistants. Cependant, aucun

résultat positif ou négatif n'a été formulé concernant l'estimateur \hat{K}_{MAPm} . Nous verrons ici que cet estimateur semble empiriquement consistant pour ce modèle de synthèse.

On fait varier N de 100 à 5000 pour $D = 16, 25$ et 36 , et les facteurs d'échelle pour la synthèse sont choisis tels que $\delta^2 = 50/k$, où $1 \leq k \leq K_S$ est le numéro de composante. Dans les trois cas, on choisit $K_S = \sqrt{D}$.

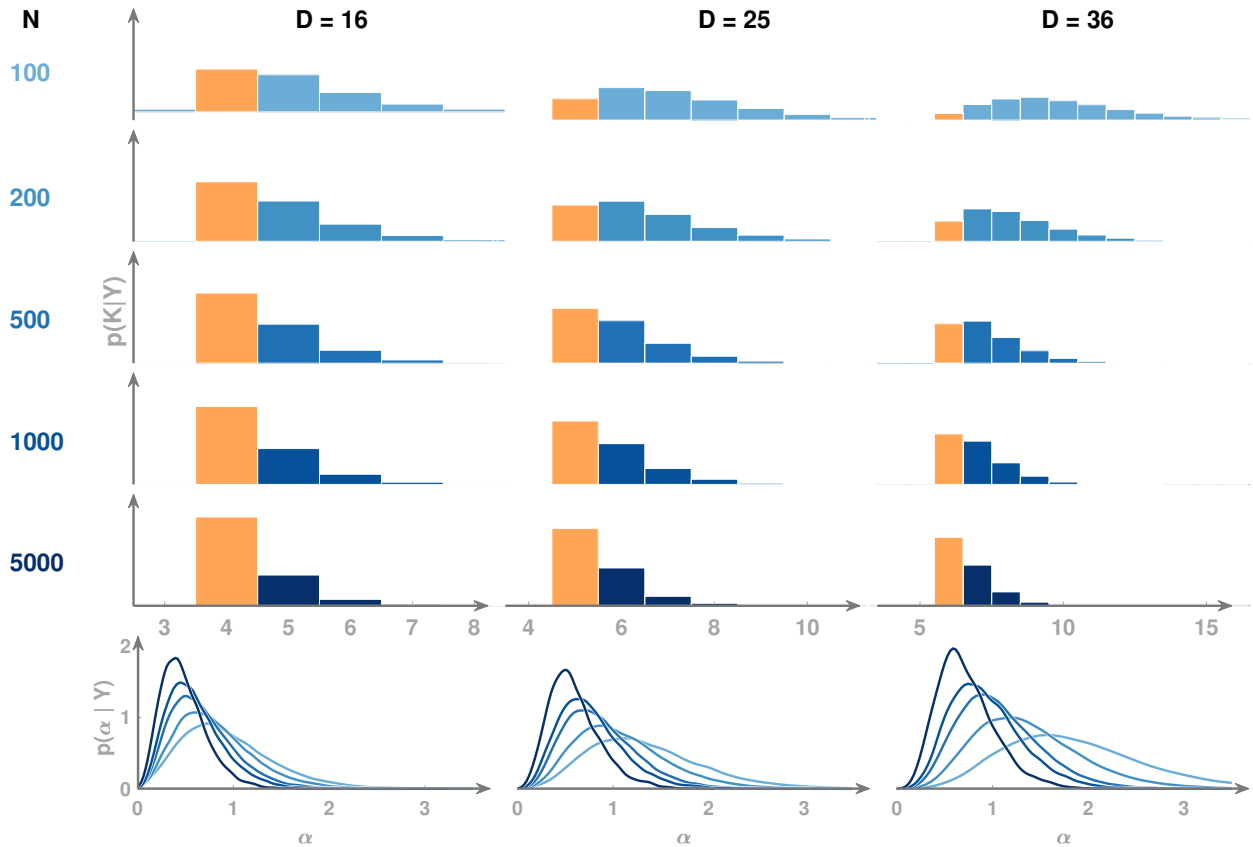


FIGURE 4.5 : Distributions marginales *a posteriori* empiriques du nombre de composantes actives $K|Y$, pour différentes configurations. De gauche à droite : $D = 16, 25$ et 36 . Le nombre d'observations varie de 100 à 5000. La barre orange sur les histogrammes indique la vraie dimension du sous-espace. Les trois figures en bas sont les distributions marginales empiriques $\alpha | Y$.

La figure 4.5 montre la distribution *a posteriori* empirique $p(K|Y)$, la dimension du sous-espace pour différentes valeurs de D et de N . Lorsque $D = 16$, l'estimateur **MAPm** retrouve la bonne valeur de la dimension, *i.e.*, $\hat{K} = 4$ quel que soit le nombre d'observations. Pour $D = 25$ et 36 , l'estimateur **MAPm** retrouve la bonne valeur de K seulement lorsque le nombre d'observations est grand : à partir de $N = 500$ pour $D = 25$ et $N = 1000$ pour $D = 36$. De manière générale, on observe que les lois *a posteriori* se concentrent autour de la bonne valeur de la dimension du sous-espace lorsque le nombre d'observations augmente. Ces observations amènent à penser que l'estimateur **MAPm** est consistant. On observe de plus que l'estimateur **MAPm** retrouve la bonne dimension pour approximativement $\ln(N) \gtrsim K$.

Ces observations ne sont pas en contradiction avec le théorème 4.1. En effet, dans le modèle BNP-PCA, le paramètre α est ici également une variable aléatoire. L'estimateur **MAPm** est intégré sur l'ensemble des valeurs

de α . Ainsi, une solution au problème d'inconsistance de l'estimateur peut consister à échantillonner conjointement l'hyperparamètre α , tandis que fixer α impose *a priori* $K \simeq \alpha \ln(N)$.

La dernière ligne de la figure 4.5 illustre l'évolution de la loi *a posteriori* du paramètre α en fonction du nombre d'observations pour chaque valeur de D . On constate que lorsque N augmente, le mode de la distribution $p(\alpha|\mathbf{Y})$ se rapproche de 0. Rappelons que *a priori*, la moyenne de la dimension du sous-espace est proportionnelle à $\alpha \ln N$. Lorsque α est fixé, cet effet *a priori* agit encore *a posteriori* (cf. théorème 4.1). Mais lorsque l'on autorise α à varier, on observe empiriquement que ce paramètre compense le terme en $\ln N$ afin que le produit se stabilise autour de la bonne valeur K de la dimension. Nous ne connaissons pas ni la portée ni l'origine de ce phénomène. En particulier, nous ne savons pas si ce résultat est propre au choix d'un *a priori* de Jeffreys sur α , impropre (avec une masse infinie en 0), où s'il reste vrai si l'on choisit un *a priori* différent sur α .

Pour conclure, cette expérience a montré que l'estimateur MAPm retrouve la bonne valeur de la dimension du sous-espace contenant un signal lorsque le nombre d'observations est suffisamment grand, donc asymptotiquement. *Ce résultat positif* vient contrebalancer les théorèmes 4.1 et 4.2. Nous soulignons que ces résultats sont valables pour ce modèle de synthèse (qui est différent mais proche du modèle d'analyse), et que le nombre d'observations nécessaires afin de construire un estimateur correct dépend du modèle génératif et de la dimension. Il serait intéressant d'étudier le comportement de cette quantité. Pour l'instant, on ne peut que se risquer à conjecturer qu'il faut approximativement $\ln(N) \simeq K$ soit $N \simeq e^K$ observations pour que l'estimateur MAPm retrouve la bonne valeur de la dimension.

4.5.3 Cas d'un signal constitué uniquement de bruit

Dans cette expérience, les paramètres d'échelle du modèle de synthèse sont tous choisis égaux à 0. Puisque le signal est composé uniquement de bruit, $K_S = 0$. $N = 500$ observations en dimension $D = [9, 16, 25, 36]$ sont générées. Rappelons que l'interprétation de ce scénario est double : un signal formée uniquement de bruit ou un signal de dimension D . L'objectif de cette expérience est de vérifier que l'on retrouve bien $\hat{K} = 0$.

Comme $K_S = 0$, chaque observation \mathbf{y}_n est générée suivant une loi normale multivariée de matrice de covariance $\sigma^2 \mathbb{I}_D$. La matrice de covariance empirique vérifie asymptotiquement

$$\lim_{N \rightarrow +\infty} \frac{1}{N} \mathbf{Y} \mathbf{Y}^T = \sigma^2 \mathbb{I}. \quad (4.43)$$

De fait, quel que soit le niveau de parcimonie, la loi *a posteriori* jointe sur les directions (cf. Éq. (4.22)) \mathbf{P} vérifie

$$p(\mathbf{P}|\mathbf{Y}, \delta^2, \sigma^2) \underset{N \rightarrow +\infty}{\propto} 1, \quad (4.44)$$

ce qui correspond à la distribution uniforme sur le groupe orthogonal \mathcal{O}_D . Dans ce cas précis, on s'attend donc à ce que toutes les directions soient distribuées uniformément.

L	0	1	2	3	4	5
$D = 9$	0.05	0.05	0.05	0.05	0.05	0
$D = 16$	0.05	0	0.05	0.05	0	0
$D = 25$	0.05	0.1	0.05	0.1	0.05	0
$D = 36$	0.05	0.05	0.05	0.05	0.05	0.05

La table 4.1 affiche tous les résultats moyennés sur 20 répétitions des tests d'adéquation de Kolmogorov-Smirnov contre les lois des produits scalaires exhibées au théorème 4.3. On voit pour toutes les 4 dimensions considérées que l'estimateur \widehat{K}_{KS} défini équation (4.38) retrouve bien $\widehat{K}_{KS} = K_S = 0$ dans 95% des cas. Cette marge d'erreur coïncident ici avec les 5% d'erreur dans la définition du test statistique. Comme $1/20 = 0.05$, il aurait fallu répéter l'expérience au moins une centaine de fois pour obtenir des résultats statistiquement représentatifs. Le facteur limitant est pour l'instant le temps de calcul lié à l'implémentation MATLAB. Nous discuterons ce point en 4.8.1.

Pour conclure, le nouvel estimateur \widehat{K}_{KS} est capable d'identifier un sous-espace de dimension 0. Le résultat négatif du théorème 4.2 ne s'applique pas non plus à cet estimateur.

4.5.4 Détection de signaux en dessous du niveau de bruit

Dans cette dernière expérience, on cherche à identifier à quel point l'algorithme BNP-PCA est capable de retrouver des composantes du signal dont l'énergie est faible devant le niveau de bruit. Cela correspond ici à des facteurs d'échelles δ^2 inférieurs à 1.

On génère $N = 500$ et $N = 2000$ observations en dimension $D = 16$ suivant le modèle de synthèse pour $K_S = D$. Les facteurs d'échelles sont choisis proportionnellement à $10/k^2$, où k est le numéro de la composante, tels les quatre premiers facteurs d'échelles $\delta_1^2, \dots, \delta_4^2$ valent $[10, 2.2, 0.9, 0.5, 0.3]$. Notons que seuls deux composantes sont au dessus du niveau de bruit. Dans le cas idéal, on aimerait que toutes les composantes du signal soient retrouvées.

Les figures 4.6-b et 4.6-d montrent les lois *a posteriori* marginales empiriques de la dimension du sous-espace $K|\mathbf{Y}$. On observe dans les deux cas que l'estimateur MAPm de la dimension du sous-espace vaut $\widehat{K}_{MAPm} = 2$. La table 4.2 affiche également les résultats de l'estimateur \widehat{K}_{KS} moyennés sur les 20 réalisations. On constate que l'estimateur retrouve majoritairement des sous-espaces de dimension 2 (pour $N = 500$) et 3 ($N = 2000$).

TABLE 4.1 : Résultats des tests d'adéquation de Kolmogorov-Smirnov pour différentes valeurs de la dimension D (indiquée dans la première colonne), $n = 500$ observations et lorsque le signal est un bruit blanc gaussien de variance $\sigma^2 = 0.1$. La ligne du haut correspond aux différents candidats à la dimension du sous-espace. Le résultat du tests est 1 lorsque la distribution uniforme est rejetée, et 0 sinon. Tous les résultats sont moyennés sur 20 répétitions.

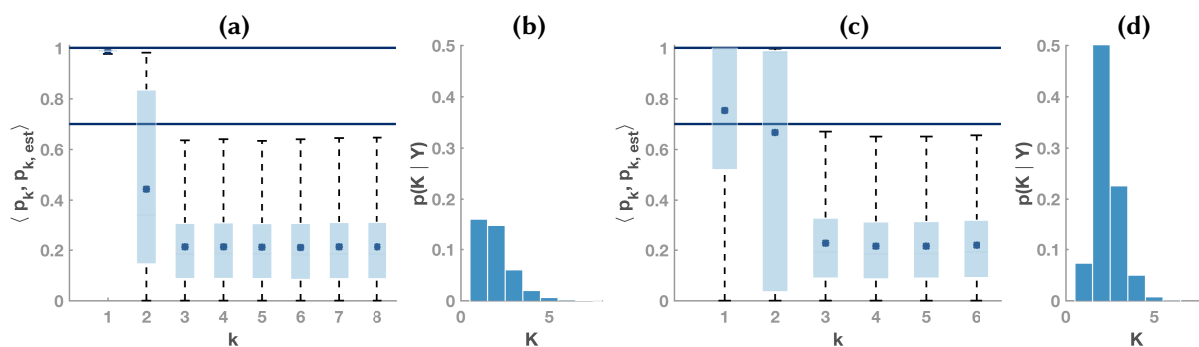


FIGURE 4.6 : (a) et (c) : répartition *a posteriori* des produits scalaires entre les directions $\mathbf{p}^{(t)}$ et les directions ayant servies à la synthèse, pour $N = 500$ et $N = 2000$. Les deux droites horizontales encadrent la région $[0.8, 1]$.

(b) et (d) : Lois *a posteriori* marginales $K|Y$ pour $N = 500$ et $N = 2000$.

Pour compléter ces observations, les figures 4.6-a et 4.6-c montrent un intervalle de confiance *a posteriori* des produits scalaires $\langle \mathbf{p}_k, \mathbf{h}_k \rangle \mid \mathbf{Y}$ pour $N = 500$ et $N = 2000$. Puisque la valeur du produit scalaire est le cosinus de l'angle entre les deux directions, une valeur proche de 1 correspond à un bon alignement. Dans les deux cas, les deux premières directions sont bien identifiées et se démarquent des suivantes. En revanche, les produits scalaires des directions 3 à 16 sont visuellement similaires. Toutefois, un écart statistique non visible à l'œil nu mais détecté par les tests statistiques fait que $\hat{K}_{KS} = 3$ pour le second scénario.

Pour conclure, cette expérience a montré que le modèle BNP-PCA n'est pas capable d'identifier des composantes en dessous du niveau de bruit. C'est une limitation de l'algorithme.

TABLE 4.2 : Répartitions de la valeur des estimateurs \hat{K}_{KS} sur 20 réalisations pour $N = 500$ et $N = 2000$.

\hat{K}_{KS}	1	2	3	4
$N = 500$	0 %	65 %	35 %	0 %
$N = 2000$	0 %	5 %	95 %	0 %

4.5.5 Conclusion partielle

L'analyse expérimentale du comportement du modèle BNP-PCA et des estimateurs s'arrête ici. À ce stade, nous pouvons déjà tirer une conclusion partielle. Avant tout, ces expériences rassurent quant à la capacité de l'algorithme à identifier un sous-espace après les deux résultats négatifs de la partie 4.4.1. La consistance empirique de l'estimateur MAPm laisse même penser que le problème venait seulement des estimateurs MAP.

On peut se demander si l'on observe le même phénomène pour les modèles de mélange utilisant le CRP. L'étude de la distribution *a posteriori* des alignements valide montre que l'algorithme est capable d'identifier un sous-espace pertinent. Cette étude valide également l'uniformité de la distribution des composantes non significatives et sur lequel repose le nouvel estimateur \hat{K}_{KS} en 4.4.2. Notons que nous n'avons pas évoqué les temps de calcul, long ici en raison de l'implémentation MATLAB non optimisée. Nous reviendrons sur ce point en 4.8.1.

La limite que nous avons relevée est l'incapacité des estimateurs à identifier des composantes en dessous du niveau de bruit alors que le modèle d'analyse l'autorise. Bien que non traité, nous pensons que le cas d'un bruit anisotrope sera aussi problématique : l'algorithme estimera un bruit moyen, et les composantes en dessous de ce niveau ne seront pas identifiées. Ce point nous amènera d'ailleurs dans l'application en 4.6.2 à blanchir les données avant d'en estimer le sous-espace.

4.6 APPLICATIONS

NOUS APPLIQUONS la méthode **BNP-PCA** à deux problèmes d'apprentissage. Dans la partie 4.6.1, le modèle est légèrement modifié afin de coupler la recherche de sous-espace à une tâche de classification non supervisée. La partie 4.6.2 illustre l'estimation de sous-espace dans le cadre du démélange hyperspectral.

4.6.1 BNP-PCA et classification non supervisée

Nous illustrons la flexibilité du modèle BNP-PCA en couplant l'estimation du sous-espace avec une tâche de classification non supervisée. En effet, la plupart des chaînes de traitement de classification non supervisée commencent par une étape de réduction de la dimension. On espère qu'intégrer l'estimateur de la partition sur tous les sous-espaces va permettre d'améliorer les performances. On rappelle qu'une partie entière du chapitre 3 est consacrée aux modèles de mélange bayésiens.

Modèle. Supposons que la distribution des coefficients \mathbf{X} est bien décrite dans un sous-espace par un mélange de deux lois normales

$$\forall n, \quad \mathbf{x}_n \sim \pi \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Delta}_0) + (1 - \pi) \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Delta}_1), \quad (4.45)$$

où chaque Gaussienne décrit une classe. Les vecteurs $\boldsymbol{\mu}_i = [\mu_{i,1}, \dots, \mu_{i,k}]^T$ et $\boldsymbol{\Delta}_i = \text{diag}(\delta_{i,1}^2, \dots, \delta_{i,k}^2)$ pour $i \in \{0, 1\}$ sont respectivement les vecteurs moyennes et matrices de covariances de chaque loi normale. On appelle $\boldsymbol{\ell} = [\ell_1, \dots, \ell_N]$ le vecteur de variables latentes induit par le modèle de mélange. Le label ℓ_n vaut 0 ou 1 selon que l'observation \mathbf{y}_n appartient à la classe \mathcal{C}_0 ou \mathcal{C}_1 .

Des lois *a priori* vagues conjuguées sont choisies pour les hyperparamètres $\boldsymbol{\mu}_i$, $\boldsymbol{\Delta}_i$ et $\boldsymbol{\ell}$. Pour tout entier n de $\llbracket 1, N \rrbracket$ et tout k

$$\mu_{i,k} \sim \mathcal{N}(\mathbf{0}, s^2) \quad (4.46)$$

$$\delta_{i,k}^2 \sim s\mathcal{IG}(a_\delta, b_\delta) \quad (4.47)$$

$$\ell_n | \pi \sim \pi \delta_1 + (1 - \pi) \delta_2 \quad (4.48)$$

$$\pi \sim \mathcal{U}([0, 1]), \quad (4.49)$$

où $s = 10$ est choisi grand devant la variance des données afin de faire office d'*a priori* vague. Notons que choisir des lois non informatives de type Jeffreys est à nouveau interdit pour conserver l'intégrabilité de la loi *a posteriori*.

Estimateur. On rappelle que l'objectif n'est pas d'estimer la dimension mais de proposer un estimateur des labels intégré sur tous les sous-espaces. On utilisera dans cette expérience, un estimateur **MAPm** des label défini par

$$\forall n, \quad \widehat{\ell}_n = \arg \max_{m \in \{0, 1\}} \mathbb{P}[\ell_n = m | \mathbf{Y}], \quad (4.50)$$

et approché avec les itérations de l'algorithme MCMC.

Pour annuler les problèmes de commutation d'étiquettes inhérents à ce type de modèle, on traite les itérations de l'algorithme de Gibbs comme recommandé dans Marin et Robert (2007) (Partie 6-4). En particulier, on suppose que les deux points les plus éloignés en distance euclidienne, notés A et B , appartiennent à deux classes différentes. On attribue au point A la classe \mathcal{C}_0 , et au point B la classe \mathcal{C}_1 . Pour chaque itération de l'algorithme de Gibbs, on change les labels tels que tous les points appartenant aux mêmes classes que les points A et B soient respectivement dans les classes \mathcal{C}_0 et \mathcal{C}_1 .

Protocole. La méthode est illustrée sur un sous-ensemble de MNIST, un jeu de données de chiffres manuscrits disponible en ligne. Ce jeu de données est construit à partir des 200 premières images des chiffres 6 et 7, pour un total de 400 images (voir figure 4.7-a et -b). Chaque observation est initialement une image 28×28 représentant le chiffre 6 ou 7. On supprime les pixels de variance nulle, ce qui correspond grossièrement aux contours de l'image (voir figure 4.7-c). Les pixels restants sont concaténés dans un vecteur de taille 572.

Notre méthode est comparée à deux algorithmes. Le premier algorithme, appelé, GMM-EM pour *Gaussian Mixture Model EM*, est un *espérance-maximisation (EM)* pour les modèles de mélange. Le second algorithme, nommé GMM-MCMC pour *Gaussian Mixture Model MCMC*, est le modèle de mélange bayésien décrit dans Marin et Robert (2007) (Chapitre 6), associé à un estimateur MAPm similaire à celui donné équation (4.50), approché sur 10000 itérations.

Pour illustrer l'effet de la réduction de la dimension, on réalise plusieurs jeux de données en projetant les observations sur les K premiers vecteurs de l'ACP, pour $K = 1 \dots 200$. On fait tourner les algorithmes GMM-EM et GMM-MCMC sur les données projetées pour chaque valeur de K . En revanche, on lance notre méthode sur le jeux de données complet.

La partition oracle des données est connue pour MNIST. On utilise cette connaissance pour mesurer les performances des algorithmes à travers un taux d'erreur de partitionnement, défini par

$$\tau_p = \frac{\# \text{ observations mal classées}}{\# \text{ observations}}. \quad (4.51)$$

La figure 4.8 résume l'ensemble des résultats de classification. On observe que GMM-EM GMM-MCMC ont un comportement similaire : les deux méthodes sont performantes lorsque les données sont projetées sur un espace de faible dimension (moins de 1% d'erreur pour GMM-EM et 1.9% pour GMM-MCMC lorsque $K = 7$). Une transition de phase apparaît lorsque la dimension du sous-espace augmente : entre les dimensions 30 et 50 pour GMM-MCMC et entre 80 et 140 pour GMM-EM. Au delà de la transition de phase, les performances se dégradent significativement. En comparaison, la méthode que nous proposons atteint 1.5% d'erreur de partitionnement sans traitement préalable des données. À titre d'indication, nous indiquons par des droites les dimensions les plus explorées par l'algorithme . On constate que la marginale concentre 70% de sa masse

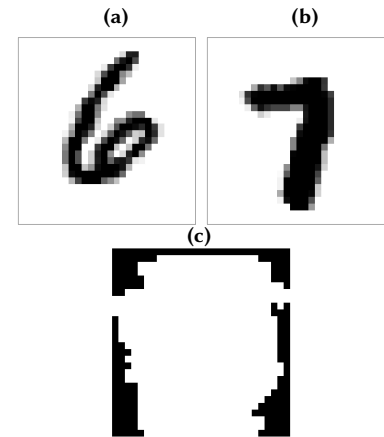


FIGURE 4.7 : (a) et (b) Une image des chiffres 6 et 7 de MNIST. (c) Pixels du jeux de données correspondant aux pixels de variance nulle, non retenus. Une image contient 784 pixels, mais seul 572 pixels sont de variance non nulle.

Notons que l'écart de performances observé entre les algorithmes GMM-EM et GMM-MCMC pourrait être réduit en renforçant l'implémentation de GMM-MCMC.

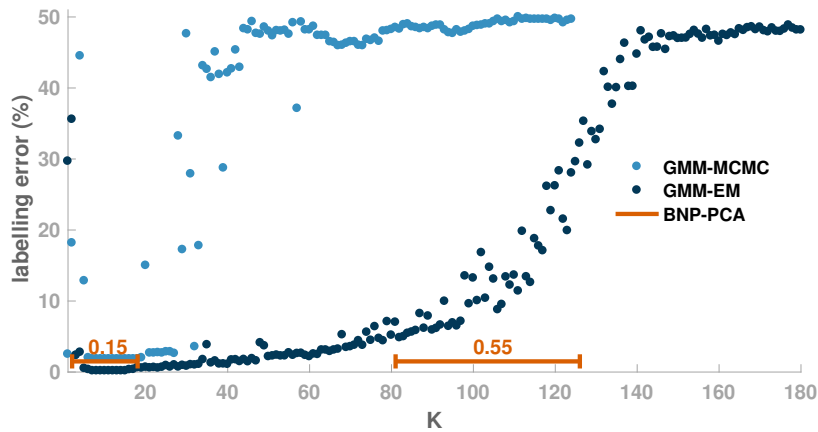


FIGURE 4.8 : Erreur de partitionnement τ_p des algorithmes GMM-EM (bleu foncé) et GMM-MCMC (bleu clair) sur 400 images extraites de MNIST, en fonction de K , la dimension de l'espace de projection des observations. Les deux intervalles oranges indiquent les dimensions pour lesquelles notre algorithme est resté 70% du temps.

entre les états $K \in [3, 18]$ et $K \in [83, 130]$. Le premier intervalle correspond aux dimensions où les deux algorithmes paramétriques atteignent les meilleures performances. Le second intervalle est en revanche plus difficile à interpréter.

Pour conclure, on a montré dans cette partie que le modèle BNP-PCA peut être légèrement modifié pour intégrer la réduction de la dimension à une autre tâche, telle que la classification non supervisée. Malgré la modification, la loi *a posteriori* reste conjuguée et l'algorithme de Gibbs change peu. Les résultats expérimentaux ont montré que la méthode BNP-PCA est capable de proposer un estimateur du partitionnement des données intégré sur l'ensemble des sous-espaces complètement non supervisés : il n'y a pas de paramètres à régler ni pour la dimension du sous-espace sur lequel projeter les données ni pour le partitionnement. Cet estimateur atteint malgré le peu d'itérations des performances comparables à des modèles de mélange de gaussiennes pour lesquels on a projeté les données sur un sous-espace pertinent.

4.6.2 Sous-espace d'un cube hyperspectral

Dans cette partie, nous utilisons la méthode BNP-PCA pour réaliser une étape de prétraitement classique en imagerie hyperspectrale. Pour ce type d'imagerie, le spectre de la lumière est échantillonné à haute fréquence sur une large bande spectrale. Par exemple, le capteur Aviris développé par la Nasa (Green et al., 1998), acquiert 224 bandes entre 400 et 2500 nm, de largeur 10 nm. L'ensemble des pixels est appelé « cube hyperspectral ». Ces notions sont illustrées à la figure 4.9.

Une hypothèse couramment admise, nommée *hypothèse de mélange linéaire* est que chaque objet macroscopique possède une signature spectrale propre et chaque pixel est une combinaison convexe de signatures spectrales. Les signatures spectrales sont appelées *spectres purs* et les proportions du mélange *abondances*. En pratique, on ajoute aussi un bruit additif lié à la mesure. Le *démélange hyperspectral* consiste à estimer ces signatures spectrales et leurs proportions au sein de chaque pixel. C'est

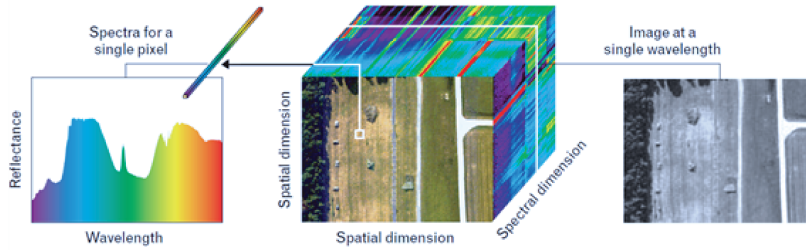


FIGURE 4.9 : Exemple de cube hyperspectral. Le cube est visible au milieu, avec au dessus sa projection sur l'espace RGB. À gauche est représenté sous forme d'une courbe la valeur d'un pixel, et à droite en niveau de gris l'ensemble des pixels pour une plage de reflectance. [Source en ligne](#)

un problème de séparation de source (Bioucas-Dias et al., 2012) ou de factorisation de matrice.

Comme les spectres purs sont très corrélés, que la dimension et le nombre de données sont grands, les algorithmes de démixage débutent par une étape de réduction de la dimension. Néanmoins, une attention particulière doit être portée à cette étape. En effet, l'hypothèse de mélange linéaire implique que les données vivent dans un sous-espace de dimension $K = R - 1$. Mais le nombre de signatures spectrales, notées R , est généralement inconnu. La plupart des méthodes estiment d'abord K , puis recherchent $R = K + 1$ spectres purs dans la scène. Estimer K revient à un problème de choix de modèle et les techniques issues de la théorie de l'information telles que le *critère d'information d'Akaike* ou la *longueur de description minimale* ont été explorées (Chang et Du, 2004). D'autres méthodes, telle que HySime (*hyperspectral signal identification by minimum error* en anglais) (Bioucas-Dias et Nascimento, 2008), cherchent des ruptures dans la décroissance des valeurs propres de la matrice de covariance empirique après blanchiment des données. Dans ce travail, nous proposons d'utiliser le modèle BNP-PCA pour estimer le sous-espace contenant les pixels dans le but d'appliquer des algorithmes de démixage, de détection, etc. L'objectif est double : réduire le temps de traitement et l'énergie du bruit capturée.

Jeux de données. La méthode est appliquée à Cuprite, un jeu de données couramment utilisé en télédétection. Cuprite est une ancienne zone minière dans le sud du Nevada aux États-Unis. Le cube hyperspectral est formé de 2500 pixels comprenant chacun 190 bandes spectrales, allant de 400 nm à 2500 nm. Pour des raisons de coût computationnel, on ne considère qu'une observations sur deux, ce qui réduit le nombre d'observations à 1250.

Le modèle BNP-PCA suppose que le bruit de capteur est blanc, gaussien et isotrope. Cette hypothèse peut ne pas être vérifiée en imagerie hyperspectrale. En effet, des corrélations au niveau du bruit peuvent apparaître entre deux pixels proches, mais également entre les bandes spectrales voisines. Nous pré-traitons les observations afin de les blanchir en s'appuyant sur l'algorithme HySime (Bioucas-Dias et Nascimento, 2008) pour estimer la matrice de covariance du bruit.

Protocole. Les estimateurs sont calculés sur 3000 itérations après une période de chauffe de 100 itérations. L'estimateur \hat{K}_{KS} est comparé à deux

méthodes d'estimation de la dimension de sous-espace. Les deux premières sont des méthodes issues de l'apprentissage automatique savoir L-PCA et OVPCA, proposées respectivement par Minka (2000) et Smidl et Quinn (2007), déjà évoquées dans la partie 4.1.2. La troisième méthode est HySime (Bioucas-Dias et Nascimento, 2008) et est spécifique à l'hyperspectral. Notons que les différentes campagnes de mesures terrain ont amené les spécialistes à s'accorder sur une dizaine de spectres purs, soit un sous-espace de dimension 9 environ.

Résultats. La table 4.3 résume les résultats d'estimation de la dimension du sous-espace. HySime est la méthode qui fournit l'estimateur le plus proche de la dimension réelle. On observe que les méthodes d'apprentissage automatique, ainsi que l'estimateur \hat{K}_{MAPm} surestiment d'un facteur 2 la dimension.

L'estimateur \hat{K}_{KS} trouve un sous-espace de dimension 13 ce qui est proche de la valeur recherchée. Notons qu'ici, les modèles d'analyse associés aux méthodes L-PCA, OVPCA et BNP-PCA diffèrent du modèle de mélange linéaire. Cette erreur de modélisation peut être à l'origine des écarts entre les performances de HySime, méthode spécialisée et les méthodes génériques d'estimation de la dimension. Une analyse comparative sur des données synthétiques permettrait de quantifier l'effet de cette erreur.

TABLE 4.3 : Résultats de l'estimation de la dimension du sous-espace décrivant le cube hyperspectral Cuprite, pour les méthodes L-PCA, OVPCA et HySime, ainsi que les estimateurs \hat{K}_{KS} et \hat{K}_{MAPm} .

Algorithme	\hat{K}
L-PCA	25
OVPCA	23
HySime	10
\hat{K}_{KS}	13
\hat{K}_{MAPm}	25

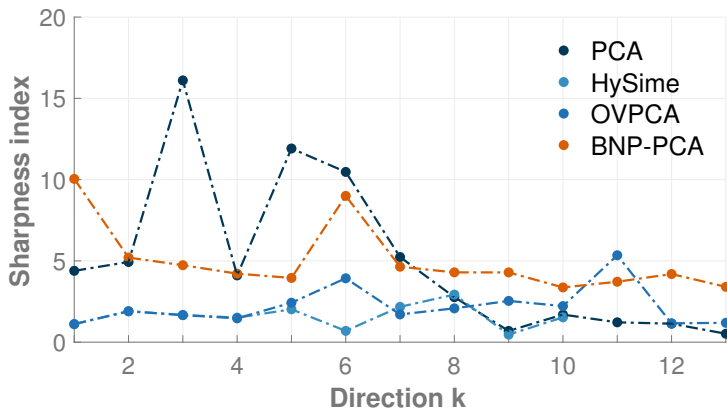


FIGURE 4.10 : Évolution du Sharpness Index des \hat{K}_{KS} premiers vecteurs propres de l'ACP (bleu foncé) ainsi que des directions significatives (orange) inférées par BNP-PCA.

Pertinence du sous-espace. Pour évaluer la pertinence du sous-espace inféré par BNP-PCA, on s'intéresse aux cartes de projection du cube hyperspectral. Ces cartes sont obtenues en projetant l'ensemble des pixels du cube hyperspectral sur les directions $\mathbf{p}_1, \dots, \mathbf{p}_{\hat{K}}$. Chaque carte est une image en niveau de gris. On évalue la pertinence d'une carte de projection en mesurant son Sharpness Index, introduit par Blanchet et Moisan (2012). Cet indice mesure la distance entre un signal et un processus Gaussien représentant le bruit. Plus l'indice est élevé, et moins le signal est susceptible de contenir du bruit.

La figure 4.10 montre les Sharpness Index des 13 premières cartes de projections obtenues avec l'estimateur de la base du sous-espace conditionné à 13, décrit dans la partie 4.4.3, ainsi que des 13 premiers vecteurs propres de l'ACP. On observe qu'à l'exception des directions 3, 5 et 7 nos estima-

teurs fournissent des *Sharpness Index* meilleurs que ceux obtenus avec les vecteurs propres de l'ACP ou les autres approches. Cela signifie que les directions inférées par BNP-PCA ont capturé davantage d'informations spatiales pertinente ce qui augurent de meilleures performances de démelange.

4.7 CONCLUSION

NOUS AVONS proposé dans ce chapitre une *analyse en composantes principales bayésienne non paramétrique* (BNP-PCA). En s'appuyant sur l'interprétation probabiliste de l'ACP proposée par Tipping et Bishop (1999b), la base du sous-espace d'intérêt est vue comme une variable latente. Le *processus du buffet indien*, une loi *a priori* non paramétrique pour les modèles à variables latentes est couplée à une loi uniforme sur la variété des matrices orthonormées, à travers le modèle d'analyse

$$\mathbf{y} = \mathbf{P}(\mathbf{z} \odot \mathbf{x}) + \mathbf{e}, \quad (4.52)$$

où \mathbf{y} est le vecteur d'observation, \mathbf{P} une matrice orthonormée, \mathbf{z} un vecteur binaire, \mathbf{x} le vecteur de coefficient et \mathbf{e} un bruit additif. La parcimonie dans l'utilisation des colonnes de la base orthonormée induite par l'IBP fait que seul un sous-ensemble des colonnes est significatif. Ces colonnes forment la base du sous-espace d'intérêt. Ainsi, la dimension de ce sous-espace n'est pas fixée à l'avance et est également une variable aléatoire. Nous avons proposé un algorithme MCMC pour échantillonner suivant la loi *a posteriori* jointe du modèle dans le but d'approcher des estimateurs.

Comme la dimension du sous-espace est une variable aléatoire, nous avons étudié la consistance d'un estimateur MAP. En particulier, nous avons montré deux résultats négatifs d'inconsistance concernant l'estimateur défini par

$$\hat{K}_{\text{MAP}}^\alpha = \arg \max_{k \in \{0, \dots, D\}} P[K = k | \mathbf{Y}, \alpha]. \quad (4.53)$$

Ces résultats affirment que lorsque le nombre d'observations tend vers l'infini, la loi *a posteriori* ne converge pas vers un Dirac en la bonne valeur. Ce résultat négatif nous a amené à considérer un estimateur alternatif, basé sur des tests statistiques que nous avons validé numériquement. Ce nouvel estimateur exploite la distribution *a posteriori* des directions non-significatives, et compare leur écart à une loi uniforme via des produits scalaires.

L'algorithme BNP-PCA a été validé à travers quatre expériences sur des jeux de données synthétiques. Nous retiendrons que

- l'algorithme BNP-PCA identifie correctement les sous-espaces sur des exemples simples,
- le nouvel estimateur proposé reconnaît un signal composé uniquement de bruit,
- l'estimateur marginalisé par rapport à α \hat{K}_{MAPm} défini par

$$\hat{K}_{\text{MAPm}} = \arg \max_{k \in \{0, \dots, D\}} P[K = k | \mathbf{Y}] \quad (4.54)$$

s'est avéré numériquement consistant sur l'exemple considéré.

- l'algorithme BNP-PCA n'est pas capable de détecter un signal plus faible que le bruit.

Pour finir, le modèle BNP-PCA a été utilisé dans deux applications sur des données réelles.

La première expérience a consisté à coupler l'estimation de sous-espace à une tâche de classification non supervisée. L'objectif est de séparer en deux classes un jeu de données sans les pré-traiter. L'algorithme final permet d'approcher un estimateur des labels intégrés sur l'ensemble des sous-espaces. Numériquement, les performances sont proches de celles obtenue en utilisant une méthode classique après avoir pré-traité les données.

La seconde expérience a consisté à estimer la dimension du sous-espace d'un cube hyperspectral, ou, de manière équivalente le nombre de composantes macroscopiques. Les estimateurs approchés avec BNP-PCA ont trouvé une valeur proche plus de la valeur oracle que les méthodes d'estimation de sous-espace issue de l'apprentissage automatique mais sans toutefois égaler les méthodes adaptées à ce type d'imagerie.

Pour finir, le modèle BNP-PCA étend la PPCA proposée par Tipping et Bishop (1999b) en rendant aléatoire la dimension du sous-espace sans passer par un modèle hiérarchique difficile à manipuler et échantillonner. L'algorithme MCMC proposé est capable d'estimer un sous-espace pertinent et sa dimension. Malgré un résultat négatif concernant un estimateur de la dimension, les estimateurs proposés se sont révélés numériquement convaincants. Les deux applications ont montré que BNP-PCA permet de traiter un problème d'apprentissage sans aucun réglage de paramètres concernant la réduction de la dimension.

4.8 PERSPECTIVES

4.8.1 Aspect numérique et temps de calculs

Nous n'avons pas discuté dans les deux parties expérimentales des performances en terme de temps de calculs. Les résultats présentés ont été obtenus avec une implémentation MATLAB relativement lente. Un facteur limitant la portabilité à un autre langage (tel que le C++) est la disponibilité des bibliothèques pour l'évaluation numérique des fonctions hypergéométriques.

En plus d'un travail sur l'implémentation numérique, le temps de calcul peut être amélioré en changeant la méthode d'inférence. Par exemples, une formulation variationnelle de l'IBP a été proposée par Doshi et al. (2009). Cette formulation repose sur une approximation paramétrique de l'IBP.

Plus récemment, les méthodes dite *small variance asymptotics* (SVA) ont été adaptées aux méthodes bayésiennes non paramétriques (Jiang et al., 2012b ; Broderick et al., 2012). L'idée est de faire tendre la variance du

bruit vers 0 et de considérer l'estimateur MAP comme le minimum d'une fonction de coût. Pour l'IBP, la fonction de coût que l'on obtient est la somme d'une erreur de reconstruction quadratique et d'un terme en $K \ln(N)$. On reconnaît le critère BIC évoqué en 4.1.2.a.

4.8.2 Généraliser les résultats concernant les estimateurs

On a vu aux théorèmes 4.1 et 4.2 que l'estimateur MAP conditionné à la valeur du paramètre α de la dimension du sous-espace est inconsistant. Ces résultats d'inconsistance ont été formulés pour le modèle BNP-PCA. Dans la continuité des résultats présentés dans Miller et Harrison (2014), on peut facilement les généraliser à une classe plus générale de modèles non paramétriques utilisant l'IBP.

Dans ce travail, le problème d'inconsistance de l'estimateur MAP a été contourné en exploitant la distribution *a posteriori* des directions non-significatives. Les techniques utilisées ici ne sont pas généralisables à d'autres modèles. Une seconde piste à considérer, plus exploratoire vise à exploiter la distributions des fréquences d'activations, données pour chaque direction k par $N^{-1} \sum_{n=1}^N z_{k,n}$. En effet, il a récemment été montré par Rousseau et Mengersen (2011) qu'en surestimant le nombre de classes et en réglant correctement les hyperparamètres d'un modèle de mélange bayésien paramétrique, la proportion des classes en trop tend vers 0 lorsque le nombre d'observations augmente. Il serait intéressant de vérifier si 1) ce résultat s'étend aux modèles bayésiens à variables latentes dans le cas paramétrique et 2) si ce l'on peut étendre ces résultats aux modèles non paramétriques.

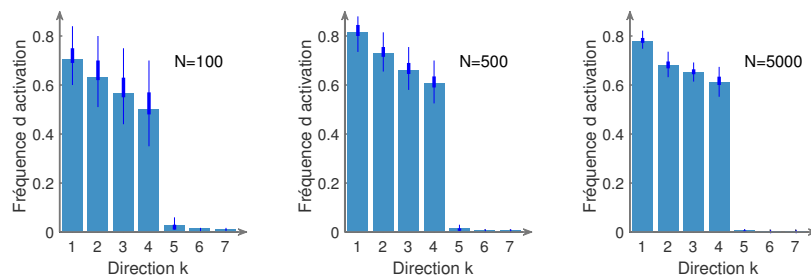


FIGURE 4.11 : Évolution des 7 plus grandes fréquences d'activations en fonction du nombre d'observations pour $D = 16$ et $K_S = 4$ lors de l'expérience présentée dans la partie 4.5.2. La courbe **bleue foncée** montre les quartiles 2 et 3 de chaque distribution.

La figure 4.11 montre l'évolution de la fréquence d'activation en fonction du nombre d'observations lors de l'expérience présentée dans la partie 4.5.2 pour $D = 16$. On observe expérimentalement que les fréquences d'activations des composantes non significatives (supérieures à 4 ici) tendent vers 0.

4.8.3 Améliorer le modèle

Un reproche qui peut être fait à l'encontre du choix des lois *a priori* pour le modèle est l'utilisation de l'IBP pour une tâche de sélection de modèle. Pour comprendre cette critique, rappelons que la mesure de Lévy du processus Bêta sous-jacent à l'IBP (cf. partie 3.2) s'écrit

$\mu_B(d\theta, dw) = \gamma w^{-1} (1-w)^{\gamma-1} dw G_0(d\theta)$. Le nombre de saut d'une mesure complètement aléatoire issue d'un tirage d'un tel processus est presque sûrement infini dénombrable puisque l'intégrale n'est pas finie. Ici, cela revient à choisir une loi *a priori* sur la dimension du sous-espace qui place une masse non nulle en l'infini.

Plusieurs généralisations de l'IBP ont été proposées, généralement dans le but de capturer des comportements en loi de puissance des fréquences d'utilisation des plats (Teh et Gorur, 2009; Caron, 2012). Teh et Gorur (2009) propose de remplacer le processus Bêta par un processus Bêta-stable, dont la mesure de Lévy s'écrit

$$\mu_{Bs}(d\theta, dw) = a \frac{\gamma(1+\gamma)}{\gamma(1-b)\gamma(\gamma+b)} w^{-b-1} (1-w)^{\gamma+b-1} dw G_0(d\theta), \quad (4.55)$$

où a et b sont deux nouveaux paramètres. Teh et Gorur (2009) contraint b à être compris entre 0 et 1 pour obtenir des comportements en loi de puissance, mais rien n'empêche de choisir b négatif. Si tel est le cas, l'intégrale de μ_{Bs} sur son domaine de définition est finie ce qui signifie que le nombre de sauts, *i.e.*, la dimension du sous-espace, est presque sûrement finie *a priori*. Il serait intéressant de voir si cette modification amènerait à des résultats plus pertinent dans l'utilisation de lois non paramétriques pour la sélection de modèle.

4.8.4 Approfondir l'application à l'imagerie hyperspectrale

Cette partie décrit deux directions à explorer dans le cadre d'une application de type démélange hyperspectral.

La première perspective est expérimentale. On a évoqué dans la partie 4.6.2 le fait que l'hypothèse de mélange linéaire en hyperspectral diffère du modèle d'analyse associée à BNP-PCA. Il serait intéressant de quantifier cette erreur de modèle par une analyse comparative sur des données synthétiques. Cela permettrait par exemple de quantifier le nombre d'observations nécessaires pour obtenir avec grande probabilité un bon estimateur avec \hat{K}_{KS} .

La seconde perspective est plutôt méthodologique. Comme pour la de classification non supervisée (*cf.* partie 4.6.1), on peut penser réaliser conjointement le démélange hyperspectral et l'estimation du sous-espace. Une façon d'aborder le problème consisterait à décomposer les coefficients \mathbf{X} dans (4.5) comme un produit de spectres purs (\mathbf{M}) et d'abondances (\mathbf{a}) et de choisir des lois *a priori* sur \mathbf{M} et \mathbf{a} . Un choix classique consiste à choisir des lois compatibles avec le modèle de mélange linéaire : loi de Dirichlet pour les abondances et lois normales pour les spectres purs (Dobigeon et al., 2008). Cependant, un tel choix ferait perdre la conjugaison du modèle et par conséquent l'intérêt de la méthode. Une solution consisterait à choisir une loi normale pour les composantes convexes mais à mettre la contrainte de positivité sur la moyenne de la loi normale.

Cette approche a été utilisée pour un problème différent par Pereyra et McLaughlin (2017).

4.8.5 Modéliser des sous-espaces fonctionnels

Les modèles non-paramétriques sont définis sur des espaces de dimension potentiellement infinie. Cependant, la contrainte d'orthogonalité des directions dans le modèle **BNP-PCA** fait que la dimension des paramètres ne dépassera jamais la dimension des observations.

L'analyse de données fonctionnelles est une discipline où les observations sont des fonctions (voir Horváth et Kokoszka (2012) pour une revue). L'espace ambiant est alors vraiment de dimension infinie. De nombreux travaux sont consacrés à la définition d'une « **ACP** fonctionnelle », ou dans une moindre mesure, à l'estimation de sous-espaces fonctionnels Suarez et Ghosal (2016). Dans la plupart des cas, ces travaux se ramènent à une approche paramétrique. Légèrement modifié, le modèle **BNP-PCA** pourrait être utilisé pour la recherche de sous-espace fonctionnel. En effet, à la manière d'une approche à noyau, on pourrait réécrire le modèle directement à partir de la loi *a posteriori* équation (4.13) en réinterprétant le produit scalaire afin d'obtenir un modèle conjugué.

Conclusion

NOUS RÉSUMONS dans ce chapitre l'ensemble des contributions et questions ouvertes qui ont été abordées au cours de ce travail. Le manuscrit est construit autour de deux axes : les deux premiers chapitres abordent le *codage antiparcimonieux* et les deux derniers chapitres l'estimation *bayésienne non paramétrique de sous-espaces*. Le point commun entre ces deux classes de problèmes est le changement de dimension : le codage antiparcimonieux vise à augmenter la robustesse d'une représentation en augmentant la dimension tandis que l'on cherche à réduire la complexité du problème lorsque l'on estime un sous-espace.

UNE FORMULATION BAYÉSIENNE DU CODAGE ANTIPARCIMONIEUX

LE CHAPITRE 1 a présenté l'intérêt des codes antiparcimonieux à travers plusieurs exemples introductifs tels que la réduction de facteur de crête, les problèmes à efforts minimaux ou la recherche approximative de plus proches voisins. Le point commun dans tous ces exemples est la recherche de représentations où l'énergie, *i.e.*, l'information, est répartie uniformément sur toutes les composantes du code.

En ce sens, le codage antiparcimonieux s'oppose aux problèmes inverses sous contraintes de parcimonie tels que ceux rencontrés en acquisition compressée, où l'on cherche à concentrer l'énergie sur un petit nombre de composantes. Pour bien comprendre les relations entre ces problèmes, nous avons d'abord discuté trois grandes problématiques en acquisition compressée : garanties, conception de matrices d'acquisition et algorithmes. Ensuite, nous avons présenté une synthèse bibliographique sur le codage antiparcimonieux. La recherche de tels codes est posée comme un problème inverse linéaire où l'on pénalise l'amplitude du code. Nous avons discuté les principaux résultats issus de la littérature :

- résultats théoriques concernant l'existence et les propriétés de ces codes,
- construction de « bonnes » matrices de codage en utilisant l'aléa,
- algorithmes de codage s'appuyant sur des opérations de troncature.

LE CHAPITRE 2 présente la formulation bayésienne du codage antiparcimonieux qui est la première contribution de ce travail. Cette nouvelle vision du problème s'obtient en interprétant la formulation variationnelle du codage antiparcimonieux en la somme d'un terme de neg-log-vraisemblance et d'un terme de neg-log-densité d'une loi *a priori*. Le

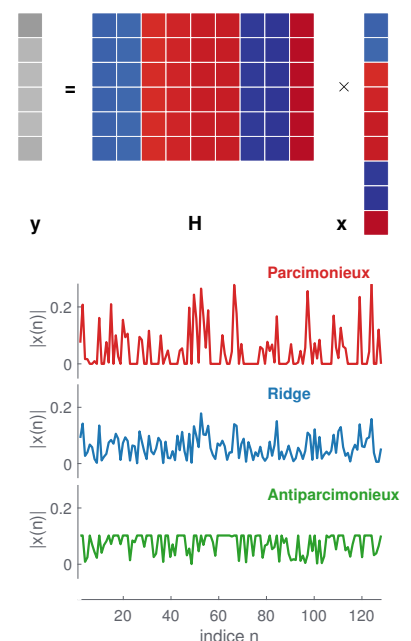


FIGURE C.12 : haut : Codage antiparcimonieux, cf. Fig. 1.7. Bas : Comparaison des pénalisations ℓ_1, ℓ_2 et ℓ_∞ , cf. Fig. 1.6.

terme coïncidant avec la loi *a priori* ne correspond à aucune densité connue. Nous avons commencé par étudier et caractériser cette nouvelle loi, baptisée *démocratique*. Il s'agit d'une loi multivariée de la famille exponentielle, de la forme $\pi(\mathbf{x}) \propto \exp(-\lambda \|\mathbf{x}\|_\infty)$. Nous avons ensuite utilisé la loi démocratique dans un modèle bayésien hiérarchique afin de favoriser les codes antiparcimonieux. Ce modèle hiérarchique inclut les hyperparamètres dans la modélisation dans le but de minimiser les réglages. Nous avons ensuite étudié trois algorithmes MCMC pour générer des échantillons suivant la loi *a posteriori* du modèle et approcher des estimateurs : un échantillonneur de Gibbs, exploitant les conjugaisons de la loi démocratique et deux méthodes exploitant les relations entre l'optimisation proximale et les méthodes de Monte-Carlo. La méthode a ensuite été appliquée à la réduction de facteur de crête. Ces expériences ont montré que les estimateurs approchés avec les méthodes MCMC parviennent à égaler les solutions proposées par les algorithmes déterministes mais en ne nécessitant aucun réglage de paramètres. De plus, l'algorithme exploitant les méthodes proximales s'est avéré plus efficace en grandes dimensions qu'un échantillonneur de Gibbs traditionnel.

En résumé, nous avons proposé une formulation bayésienne du codage antiparcimonieux qui a nécessité d'introduire et d'étudier une nouvelle loi de probabilité. La non différentiabilité de ce nouvel *a priori* nous a amené à concevoir des échantillonneurs à base de méthodes proximales, issus des récents développements en théorie de l'échantillonnage. Nous obtenons trois algorithmes MCMC pour calculer des codes antiparcimonieux sans aucun réglage de paramètre. Ce chapitre est une synthèse des résultats présentés dans Elvira et al. (2016, 2017a, 2017d).

PERSPECTIVES. Le chapitre 2 s'est conclu sur les perspectives de la première moitié de ce travail. Nous retiendrons trois grands axes.

Numérique. L'implémentation C++ des algorithmes a permis de multiplier d'un facteur 10 la vitesse des algorithmes. Il serait intéressant de relancer les expériences numériques du chapitre en augmentant le nombre d'itérations voire les dimensions du problème.

Théorique. Il existe des garanties théoriques concernant les codes antiparcimonieux qui ne sont jamais atteintes par nos estimateurs. Cela concerne en particulier le nombre de valeurs extrêmes et l'étalement du code. Les positions de ces valeurs extrêmes sont inconnues, mais nous pensons pouvoir les estimer à l'aide de la distribution *a posteriori* du code. Il s'agirait d'un nouvel estimateur bayésien sans équivalent direct avec une méthode d'optimisation.

Nous avons observé qu'à valeur maximale $\|\mathbf{x}\|_\infty$ fixée, la loi démocratique favorise autant le code 1-parcimonieux $[\|\mathbf{x}\|_\infty, 0, \dots, 0]$ que le code antiparcimonieux $[\pm\|\mathbf{x}\|_\infty]$. Cette raison nous laisse penser qu'une généralisation de la loi démocratique, de la forme $\pi(\mathbf{x}) \propto \prod_\ell |x_\ell|^{\gamma-1} \exp(-\lambda \|\mathbf{x}\|_\infty)$ constitue un *a priori* plus pertinent pour favoriser les codes antiparcimonieux. Cette nouvelle loi favoriserait les éléments proches des frontières des cônes de \mathbb{R}^D (voir figure ci-contre pour un rappel graphique de la définition d'un cône). Or, ces frontières sont les zones où le code est antiparcimonieux.

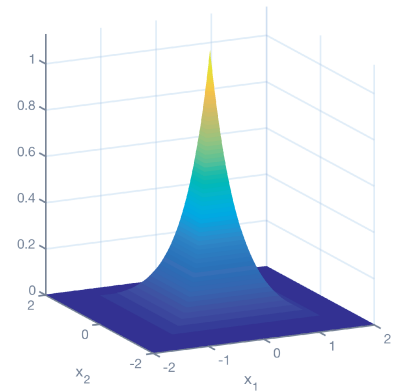


FIGURE C.13 : Densité de la loi démocratique de paramètre $\lambda = 3$ en dimension 2, cf. partie 2.1.1.a.

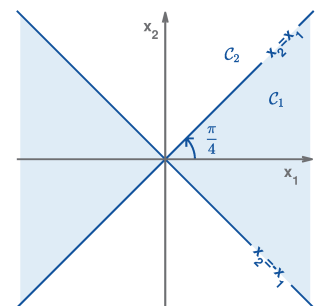
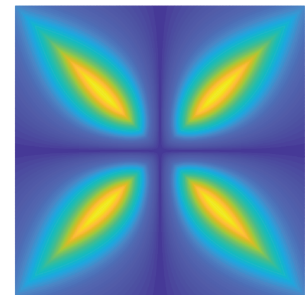


FIGURE C.14 : Haut : Densité de la loi démocratique généralisée, cf. Fig. 2.19. Bas : Doubles cônes de \mathbb{R}^2 , cf. Fig. 2.2.

Applicatif. Le dernier axe de perspective vise à étendre l'utilisation de la loi démocratique à d'autres problèmes. On est par exemple capable d'expliciter la loi d'un vecteur démocratique renormalisé, *i.e.*, la loi de $|\mathbf{x}|/\|\mathbf{x}\|_2$ lorsque \mathbf{x} est démocratique. On identifie une nouvelle loi sur les combinaisons convexes, que l'on peut manipuler analytiquement. Cette famille de loi est par exemple utilisée pour modéliser les proportions dans un modèle de mélange bayésien. La loi démocratique renormalisée favorisent les proportions similaires, ce qui n'est pas sans rappeler la propriété de *micro-clustering* recherchée dans certains problèmes de partitionnement non supervisé, comme la détection et résolution d'entités (Miller et al., 2015).

UNE ACP BAYÉSIENNE NON PARAMÉTRIQUE

LE CHAPITRE 3 a introduit deux lois *a priori* non paramétriques en discutant deux problèmes classiques en apprentissage automatique. La modélisation non paramétrique permet de ne pas fixer à l'avance le nombre de degrés de liberté d'un modèle. Les modèles s'enrichissent au fur et à mesure que le nombre d'observations augmente. Dans les deux cas, nous avons abouti à des algorithmes relativement simples à utiliser.

La première partie introduit le *processus de Dirichlet* à partir du problème de classification non supervisée (*clustering*). Le processus est présenté comme une loi jointe sur les paramètres et les proportions de chaque classe. Pour faciliter l'inférence, nous avons caractérisé la loi induite sur les partitions : c'est le *processus du restaurant chinois (CRP)*. Nous avons ensuite discuté l'utilisation de ce processus sur un exemple simple. Bien que nous n'ayons pas utilisé ce processus dans notre travail, l'intérêt de cette présentation est double : introduire la méthode bayésienne non paramétrique et discuter des résultats théoriques concernant la consistance des estimateurs que nous étendrons au chapitre 4.

La seconde partie introduit le *processus Beta-Bernoulli* à partir des modèles à variables latentes. Le processus est présenté comme une loi jointe sur les variables latentes et leur attribution. L'utilisation des variables latentes par les observations peut être représentée sous la forme d'une matrice binaire de taille potentiellement infinie. Pour faciliter l'inférence, nous avons caractérisé la loi induite sur cette matrice binaire : c'est le *processus du buffet indien (IBP)*. Ce processus est l'un des ingrédients du modèle décrit dans le chapitre suivant.

LE CHAPITRE 4 a été consacré au modèle bayésien non paramétrique proposé pour l'estimation de sous-espace.

Le chapitre a débuté sur une synthèse des principales méthodes probabilistes d'estimation de sous-espaces. Cette taxonomie a mis en avant deux problématiques : estimer la dimension de ce sous-espace et en estimer une base. Nous avons mis en évidence la nécessité de pénaliser les modèles trop complexes en termes de nombres de paramètres.

Nous avons présenté le modèle bayésien non paramétrique BNP-PCA

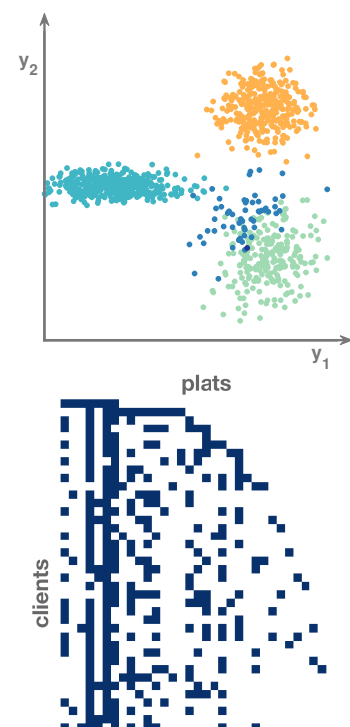


FIGURE C.15 : Haut : un exemple de l'utilisation du bayésien non paramétrique pour la classification non supervisée, cf. discussion en 3.1.5.

Bas : Une réalisation de l'IBP, cf. Fig. 3.14.

pour l'estimation de sous-espace. BNP-PCA est construit en utilisant la formulation classique de l'analyse en composantes principales comme un problème à variables latentes, où les variables cachées sont les composantes de la base du sous-espace. Nous avons associé l'IBP, une loi *a priori* non paramétrique pour les modèles à variables latentes vu au chapitre 3 à une loi uniforme sur les bases orthonormales. Chaque direction orthonormale porte un signal dont l'énergie est caractérisée par un facteur d'échelle relatif au niveau de bruit. Ce modèle permet d'inférer une base du sous-espace sans fixer à l'avance sa dimension. Le modèle est accompagné d'un algorithme MCMC pour générer des échantillons suivant loi *a posteriori* des paramètres dans le but d'approcher des estimateurs. Comme un des objectifs du modèle est d'inférer la dimension du sous-espace, nous avons étudié la consistance des estimateurs MAP de la dimension. Nous entendons par consistance ici le fait que le mode de loi *a posteriori* de la dimension du sous-espace se concentre autour de la valeur associée au modèle de synthèse. Nous avons montré deux résultats négatifs d'inconsistance concernant deux de ces estimateurs. Bien qu'en pratique nous calculons un estimateur MAP marginalisé où le paramètre α de l'IBP est aussi échantillonné, nous avons proposé un estimateur alternatif de la dimension basé sur des tests statistiques sur la loi *a posteriori* des directions orthonormales. Nous avons ensuite défini les estimateurs de la base du sous-espace et des facteurs d'échelles conditionnellement à la dimension estimée.

La fin du chapitre 4 est expérimentale. Nous avons d'abord validé la méthode sur des exemples synthétiques. Nous retiendrons que

1. l'algorithme BNP-PCA identifie correctement les sous-espaces sur des exemples simples,
2. le nouvel estimateur proposé reconnaît un signal composé uniquement de bruit,
3. l'estimateur MAPm de la dimension, calculé lorsque le paramètre α est également échantillonné, s'est avéré empiriquement consistant contrairement aux estimateurs MAP,
4. l'algorithme BNP-PCA n'est pas capable de détecter un signal plus faible que le bruit.

La dernière partie a été consacrée à deux applications sur des données réelles. Dans le premier exemple, le modèle BNP-PCA est couplé avec une tâche de partitionnement à deux classes. Le modèle BNP-PCA a été légèrement modifié mais reste conjugué. Le nouvel algorithme a été capable d'estimer la partition d'un ensemble de données sous forme d'images sans pré-traiter à l'avance les observations.

Dans le second exemple, nous avons dénombré le nombre d'éléments purs présents sur une scène en estimant la dimension du sous-espace contenant les pixels d'un cube hyperspectral. La valeur de la dimension proposée par l'estimateur que nous avons proposé est proche de celle sur laquelle s'accordent les spécialistes. À l'aide d'une mesure sur le niveau de bruit, nous avons montré que le sous-espace obtenu est plus pertinent que celui inféré par les méthodes d'apprentissage classique, ce qui augure de meilleures performances pour des tâches de télédétection.

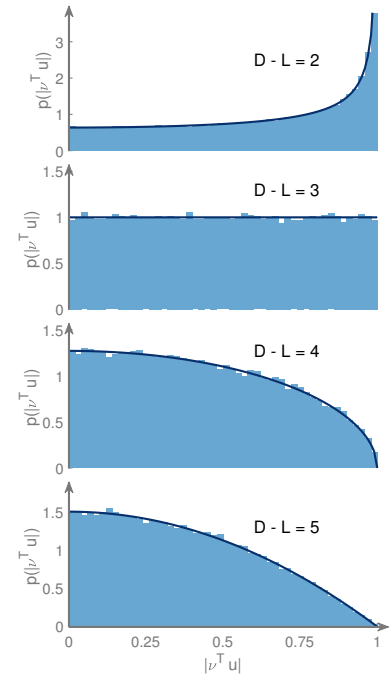


FIGURE C.16 : Distribution théorique des produits scalaires des composantes non significatives, cf. Fig. 4.3.

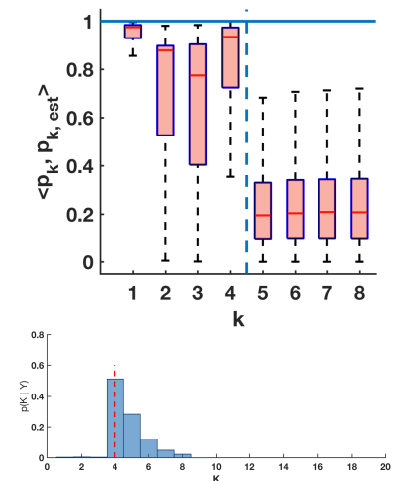


FIGURE C.17 : Résultat de l'algorithme MCMC, cf. expérience en 4.5.1.

En résumé, nous avons proposé une formulation bayésienne non paramétrique de l'ACP en couplant un IBP avec une loi uniforme sur les matrices orthonormées pour estimer un sous-espace et sa dimension. Les estimateurs proposés ont été étudiés théoriquement et numériquement et la méthode a été appliquée à deux problèmes d'apprentissage. Ce chapitre constitue une synthèse des résultats présentés dans Elvira et al. (2017c) et soumis dans Elvira et al. (2017b).

PERSPECTIVES. Le chapitre 4 s'est conclu sur les perspectives méthodologiques et applicatives de ce travail.

Méthodologiques. Une première perspective vise à élargir les résultats d'inconsistance des estimateurs à une plus large classe de modèles s'appuyant sur l'IBP. Nous pensons également pouvoir résoudre ce problème d'inconsistance inhérent à l'IBP en utilisant une généralisation du processus du buffet indien. De façon plus exploratoire, il serait intéressant de caractériser la distribution des fréquences d'utilisation des variables latentes induite par l'IBP. Nous pensons qu'il est possible de construire avec ces quantités un nouvel estimateur de la dimension plus efficace.

Applicatives. Nous envisageons deux nouvelles applications. La première vise à coupler l'estimation de sous-espace avec une tâche de démixage hyperspectral. Nous pensons que réaliser conjointement ces deux opérations peut améliorer les performances. La seconde application vise à adapter le modèle afin de pouvoir traiter des données fonctionnelles.

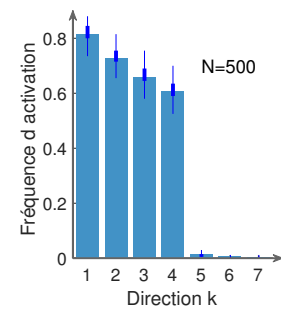


FIGURE C.18 : Nouvel estimateur de la dimension du sous-espace en utilisant la distribution des fréquences d'activation, cf. Fig. 4.11.

ANNEXES

Loi démocratique et code antiparcimonieux

A

A.1 ALGORITHME PFA

La méthode de résolution du problème (P_∞) a été proposée par Fuchs (2011) commence par transformer le problème (P_∞) en un problème d'optimisation convexe

$$\begin{aligned} \hat{\mathbf{x}} &= \arg \min_{\mathbf{z} \in \mathbb{R}^N} J_\beta^\infty(\mathbf{z}) \\ J_\beta^\infty(\mathbf{z}) &\triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{z}\|_2 + \beta \|\mathbf{z}\|_\infty, \end{aligned} \quad (\mathbf{V}_\infty^\beta)$$

où β est réel positif ou nul. Notons qu'il s'agit d'une reformulation variationnelle du problème (P_∞) comme cela a déjà été fait pour (\mathbf{V}_1^β).

Comme J_β^∞ vue comme une fonction de \mathbf{z} n'est pas différentiable en raison de la norme ℓ^∞ , Fuchs (2011) propose de s'intéresser à sa sous-différentielle. La sous-différentielle d'une fonction f en $\mathbf{x} \in \mathbb{R}^N$, notée $\partial f(\mathbf{x})$, est l'ensemble des vecteurs \mathbf{v} de \mathbb{R}^N vérifiant pour tout \mathbf{x}' de \mathbb{R}^N $f(\mathbf{x}') - f(\mathbf{x}) \geq \mathbf{v}^T(\mathbf{x}' - \mathbf{x})$. Lorsque la fonction f est différentiable, la sous-différentielle est réduite au gradient. Pour $f = \|\cdot\|_\infty$, Fuchs (2011) donne

$$\begin{aligned} \partial f(\mathbf{0}) &= \{\mathbf{v} \in \mathbb{R}^N, \|\mathbf{v}\| \leq 1\} \\ \partial f(\mathbf{x}) &= \left\{ \mathbf{v} \in \mathbb{R}^N, \|\mathbf{v}\|_1 = 1 \text{ et } \begin{array}{ll} v_n x_n \geq 0 & \text{si } |x_n| = \|\mathbf{x}\|_\infty \\ v_n = 0 & \text{si } |x_n| < \|\mathbf{x}\|_\infty \end{array} \right\}. \end{aligned} \quad (\text{A.1})$$

Comme J_β^∞ est convexe, une condition nécessaire et suffisante pour que \mathbf{x} soit solution de (\mathbf{V}_∞^β) est que la sous-différentielle de J_β^∞ en \mathbf{x} contienne le vecteur nul (Bauschke et Combettes, 2011). En développant ce résultat, $\mathbf{x} \in \mathbb{R}^N$ est solution de (\mathbf{V}_∞^β) si et seulement si (Fuchs, 2011, Lemme 1)

$$\exists \mathbf{v} \in \partial \|\cdot\|_\infty(\mathbf{x}), \quad \mathbf{H}^T(\mathbf{H}\mathbf{x} - \mathbf{y}) + \beta \mathbf{v} = \mathbf{0}. \quad (\text{A.2})$$

L'équation (A.2) est utilisée pour construire un algorithme itératif mettant successivement à jour β et \mathbf{x} . Pour décrire ces opérations, il est nécessaire d'introduire quelques notations. Soit $\mathbf{x}^{(t)}$ un vecteur de \mathbb{R}^N correspondant à une itération de l'algorithme. Pour des raisons pratiques, nous omettrons la plupart du temps l'exposant t . Notons \mathcal{I}^\uparrow le sous-ensemble de $\llbracket 1, N \rrbracket$ contenant les indices n de \mathbf{x} correspondant aux valeurs extrêmes, *i.e.*, tels que $|x_n| = \|\mathbf{x}\|_\infty$, et \mathcal{I}^\downarrow le sous-ensemble $\llbracket 1, N \rrbracket \setminus \mathcal{I}^\uparrow$. On définit respectivement les sous-vecteurs et sous-matrices de \mathbf{x} , \mathbf{v} et \mathbf{H} en extrayant les composants ou les colonnes dont les indices sont contenus

Cette annexe rassemble les preuves des résultats énoncés au chapitre 2. Les preuves sont classées par section.

dans \mathcal{I}^\uparrow et \mathcal{I}^\downarrow . Ces sous-vecteurs et sous-matrices sont notés en ajoutant l'exposant \uparrow ou \downarrow .

En séparant les sous-vecteurs induits par \mathcal{I}^\uparrow et \mathcal{I}^\downarrow dans (1.37), on obtient

$$\mathbf{H}^{\uparrow T}(\mathbf{H}^\downarrow \mathbf{x}^\downarrow + \mathbf{H}^\uparrow \text{signe}(\mathbf{v}^\uparrow) \|\mathbf{x}\|_\infty - \mathbf{y}) = -\beta \mathbf{v}^\uparrow \quad (\text{A.3})$$

$$\mathbf{H}^{\downarrow T}(\mathbf{H}^\downarrow \mathbf{x}^\downarrow + \mathbf{H}^\uparrow \text{signe}(\mathbf{v}^\uparrow) \|\mathbf{x}\|_\infty - \mathbf{y}) = \mathbf{0}, \quad (\text{A.4})$$

puis en isolant \mathbf{x}^\downarrow , \mathbf{v}^\downarrow et β dans (A.3) et (A.4)

$$\mathbf{x}^{\downarrow(t+1)} = \mathbf{X}_1^{(t+1)} + \mathbf{X}_2^{(t+1)} \|\mathbf{x}\|_\infty \quad (\text{A.5})$$

$$\mathbf{v}^{\uparrow(t+1)} = \mathbf{V}_1^{(t+1)} - \mathbf{V}_2^{(t+1)} \|\mathbf{x}\|_\infty \quad (\text{A.6})$$

$$\beta^{(t+1)} = B_1^{(t+1)} - B_2^{(t+1)} \|\mathbf{x}\|_\infty. \quad (\text{A.7})$$

À chaque itération, Fuchs (2011) propose de faire diminuer la valeur de $\|\mathbf{x}^{(t)}\|_\infty$ jusqu'à ce que l'un de ces deux événements se produise

- une composante de \mathbf{x}^\downarrow atteint $\|\mathbf{x}^{(t)}\|_\infty$ en valeur absolue,
- une composante de \mathbf{v}^\uparrow devient nulle.

On retient la nouvelle valeur de $\|\mathbf{x}\|_\infty$ et l'on met à jour \mathbf{x} , \mathbf{v} et β . Si l'on itère jusqu'à obtenir $\beta^{(t)} = 0$, alors $\mathbf{x}^{(t)}$ est une potentielle solution de (\mathbf{P}_∞) . Si en revanche on recherche une solution de $(\mathbf{V}_\infty^\beta)$ pour une valeur de β donnée, et la procédure est appelée *Path Following Algorithm* (PFA). L'algorithme A.1 résume la procédure.

Algorithme A.1 : Algorithme PFA.

Entrées : Mesure $\mathbf{y} \in \mathbb{R}^M$, matrice de codage \mathbf{H} , valeur cible β^* .

- 1 Initialiser $\mathbf{x}^{(0)}$, $\mathbf{v}^{(0)}$ et $\beta^{(0)}$ et $t = 0$;
- 2 Déterminer \mathcal{I}^\uparrow l'ensemble des indices des valeurs extrêmes de $\mathbf{x}^{(0)}$;
- 3 $\mathcal{I}^\downarrow = \llbracket 1, N \rrbracket \setminus \mathcal{I}^\uparrow$;
- 4 **tant que** $\beta^{(t)} > \beta^*$ **faire**
- 5 Déterminer $\mathbf{X}_1, \mathbf{X}_2, \mathbf{V}_1, \mathbf{V}_2, B_1$ et B_2 ;
- 6 $x_\infty = \|\mathbf{x}^{(t)}\|_\infty$;
- 7 **tant que faire**
- 8 **si** $\exists j, |x_j^\downarrow| = \|x\|_\infty$ **alors**
- 9 $\mathcal{I}^\uparrow = \mathcal{I}^\uparrow \cup \{j\}$ et $\mathcal{I}^\downarrow = \mathcal{I}^\downarrow \setminus \{j\}$;
- 10 $t = t + 1$ et quitter la boucle;
- 11 **fin**
- 12 **si** $\exists j, |v_j^\uparrow| = 0$ **alors**
- 13 $\mathcal{I}^\uparrow = \mathcal{I}^\uparrow \setminus \{j\}$ et $\mathcal{I}^\downarrow = \mathcal{I}^\downarrow \cup \{j\}$;
- 14 $t = t + 1$ et quitter la boucle;
- 15 **fin**
- 16 Diminuer x_∞ et mettre à jour $\mathbf{x}^{\downarrow(t+1)}, \mathbf{v}^{\uparrow(t+1)}, \beta^{(t+1)}$ suivant (A.5), (A.6) et (A.7);
- 17 **fin**
- 18 **fin**

Output : $\mathbf{x}^{(t)} \in \mathbb{R}^N$, solution de $J_{\beta^*}^\infty$.

En outre

$$\begin{aligned} \mathbf{X}_1^{(t+1)} &= (\mathbf{H}^{\downarrow T} \mathbf{H}^\downarrow)^{-1} \mathbf{H}^{\downarrow T} \mathbf{y} \\ \mathbf{X}_2^{(t+1)} &= -(\mathbf{H}^{\downarrow T} \mathbf{H}^\downarrow)^{-1} \mathbf{H}^{\uparrow T} \mathbf{y} \text{signe}(\mathbf{v}^{\uparrow(t)}) \\ \mathbf{V}_1^{(t+1)} &= \frac{1}{\beta^{(t)}} (\mathbf{H}^{\uparrow T} \mathbf{y} - \mathbf{X}_1^{(t+1)}) \\ \mathbf{V}_2^{(t+1)} &= \frac{1}{\beta^{(t)}} \mathbf{H}^{\uparrow T} (\mathbb{I} - \mathbf{H}^{\uparrow T} \mathbf{H}^\downarrow (\mathbf{H}^{\downarrow T} \mathbf{H}^\downarrow)^{-1}) \\ &\quad \times \mathbf{H}^\uparrow \text{signe}(\mathbf{v}^{\uparrow(t)}) \\ B_1^{(t+1)} &= -\text{signe}(\mathbf{v}^{\uparrow(t)})^T \mathbf{H}^{\uparrow T} (\mathbf{H}^\downarrow \mathbf{x}^{\downarrow(t)} - \mathbf{y}) \\ B_2^{(t+1)} &= (\mathbf{H}^\uparrow \text{signe}(\mathbf{v}^{\uparrow(t)}))^T \\ &\quad \times (\mathbb{I} - \mathbf{H}^{\uparrow T} \mathbf{H}^\downarrow (\mathbf{H}^{\downarrow T} \mathbf{H}^\downarrow)^{-1} \mathbf{H}^\downarrow) \\ &\quad \times \mathbf{H}^\uparrow \text{signe}(\mathbf{v}^{\uparrow(t)}) \end{aligned}$$

A.2 MOMENTS

CETTE SECTION détaille les calculs des deux premiers moments de la loi *démocratique*, en version simple et en valeur absolue. Dans toute cette section, $\mathbf{x} = [x_1, \dots, x_N]^T$ désigne un vecteur aléatoire de \mathbb{R}^N réel distribué suivant la loi *démocratique* $\mathcal{D}_N(\lambda)$.

A.2.1 Moments de la loi démocratique

L'espérance nulle du vecteur \mathbf{x} résulte de la parité de la fonction

$$f : \mathbf{x} \longrightarrow \mathbf{x} \exp(-\lambda \|\mathbf{x}\|_\infty).$$

On utilise pour le calcul de la variance la loi marginale, qui découle de l'équation (2.11) donnée au corollaire 2.1. La marginale à deux composantes correspond à la somme de N densités Gamma symétrisées dont le premier paramètre varie de 1 à N . La formule de König-Huygens donne

$$\begin{aligned} \text{Var}(x_n) &= \frac{1}{N} \sum_{\ell=1}^N \text{Var}(y \mid y \sim d\mathcal{G}(\ell, \lambda)) \\ &= \frac{1}{N} \sum_{\ell=1}^N \frac{\ell(\ell+1)}{\lambda^2} \\ &= \frac{(N+1)(N+2)}{3\lambda^2}. \end{aligned}$$

En terme de chronologie, la corollaire 2.1 apparaît après la propriété 2.2. La preuve de cette propriété ne s'appuie cependant sur aucun résultat intermédiaire.

On reconnaît la somme des premiers carrés et des premiers entiers.

Soient n_1, n_2 deux indices entiers distincts inférieurs à N . La fonction

$$x_{n_1}, x_{n_2} \longrightarrow x_{n_1} x_{n_2} \mathbf{p}(x_{n_1}, x_{n_2}; \lambda)$$

est anti-symétrique par rapport la droite $x_{n_1} = 0$ ou de manière équivalente, $x_{n_2} = 0$. L'intégrale est donc nulle.

La densité marginales à deux composantes, se déduit de la propriété 2.3.

A.2.2 Moments positifs de la loi démocratique

Le calcul de la moyenne est similaire et réalisé comme suit

$$\begin{aligned} \mathbb{E}[|X_n|] &= \frac{\lambda}{N} \sum_{\ell=0}^{N-1} \int_{\mathbb{R}_+} |x_n| \frac{\lambda^\ell}{\ell!} |x_n|^\ell \exp(-\lambda x_n) dx_n \\ &= \frac{\lambda}{N} \sum_{\ell=0}^{N-1} \int_{\mathbb{R}_+} \frac{1}{\ell! \lambda^2} |z_n|^\ell \exp(-z_n) dz_n \\ &= \frac{1}{N\lambda} \sum_{\ell=0}^{N-1} \frac{\Gamma(\ell+2)}{\Gamma(\ell+1)!} \\ &= \frac{1}{N\lambda} \sum_{\ell=0}^{N-1} (\ell+1) \\ &= \frac{N+1}{2\lambda}. \end{aligned}$$

On s'appuie toujours sur l'équation (2.12).

Changement de variable $z_n = \lambda x_n$.

Il s'agit de la somme des premiers entiers.

A.3 LOIS MARGINALES

DEUX PREUVES sont décrites dans cette section. La partie A.3.1 démontre la propriété 2.3 concernant la densité marginale d'un sous-vecteur

en rapport avec la loi *démocratique*. La partie A.3.2 est consacrée à la preuve de la convergence de la suite des lois marginales vers la distribution uniforme. Dans toute cette section, $\mathbf{x} = [x_1, \dots, x_N]^T$ désigne un vecteur aléatoire distribué suivant la loi *démocratique* $\mathcal{D}_N(\lambda)$.

A.3.1 Loi des sous-ensembles

On détaille dans cette section le calcul de la densité marginale d'un vecteur aléatoire après marginalisation d'un sous-ensemble de composantes, donnée à la propriété 2.3.

Soit J un entier tel que $J < N$ et \mathcal{K}_J un sous-ensemble de $\{1, \dots, N\}$ à J éléments. On désigne respectivement par $\mathbf{x}_{\mathcal{K}_J}$ et $\mathbf{x}_{\setminus \mathcal{K}_J}$ les sous-vecteurs de \mathbf{x} composés des J éléments indexés par \mathcal{K}_J et dont les J éléments indexés par \mathcal{K}_J ont été ôtés. La densité du vecteur aléatoire $\mathbf{x}_{\setminus \mathcal{K}_J}$ s'obtient selon

$$p(\mathbf{x}_{\setminus \mathcal{K}_J}) = \frac{2^J}{C_N(\lambda)} \int_{\mathbb{R}_+^J} \exp(-\lambda \|\mathbf{x}\|_\infty) d\mathbf{x}_{\mathcal{K}_J} \quad (\text{A.8})$$

pour tout entier J et tout sous-ensembles \mathcal{K}_J .

On procède par récurrence. On considère la proposition suivante, indexée par J et notée $\mathcal{P}(J)$

$\mathcal{P}(J)$: « Pour tout sous-ensemble \mathcal{K}_J de $\{1, \dots, N\}$ à J éléments, la propriété 2.3 est vraie ».

Si $J = 0$, \mathcal{K}_J est l'ensemble vide et la densité considérée n'est rien d'autre que la densité de la loi *démocratique* $\mathcal{D}_N(\lambda)$, ce qui est cohérent avec le résultat recherché. Donc $\mathcal{P}(0)$ est vraie.

Soit J un entier de $\{1, \dots, N-1\}$. Supposons que $\mathcal{P}(J)$ est vraie. Soit \mathcal{K}_J un sous-ensemble et k un entier de $\{1, \dots, N\} \setminus \mathcal{K}_J$. On considère le sous-ensemble $\mathcal{K}_{J+1} = \mathcal{K}_J \cup \{k\}$. La densité $p(\mathbf{x}_{\setminus \mathcal{K}_{J+1}})$ s'écrit

$$\begin{aligned} p(\mathbf{x}_{\setminus \mathcal{K}_{J+1}}) &= \int_{\mathbb{R}} p(\mathbf{x}_{\setminus \mathcal{K}_J}) dx_k \\ &= 2 \int_{\mathbb{R}_+} \frac{2^J}{C_N(\lambda)} \sum_{j=0}^J \binom{J}{j} \frac{(J-j)!}{\lambda^{J-j}} \|\mathbf{x}_{\setminus \mathcal{K}_J}\|_\infty^j \times \exp(-\lambda \|\mathbf{x}_{\setminus \mathcal{K}_J}\|_\infty) dx_k. \end{aligned}$$

On inverse les signes somme et intégrale, et l'on procède par disjonction de cas sur les valeurs de j .

Pour $j = 0$, \mathbb{R}_+ est partitionné selon $\mathbb{R}_+ = [0, \|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_\infty) \cup [\|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_\infty, +\infty)$, et

$$\int_{\mathbb{R}_+} \exp(-\lambda \|\mathbf{x}_{\setminus \mathcal{K}_J}\|_\infty) dx_k = \|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_\infty \exp(-\lambda \|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_\infty) + \frac{1}{\lambda} \exp(-\lambda \|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_\infty). \quad (\text{A.9})$$

La démarche est la même pour $j > 0$ mais requiert plusieurs intégrations par parties, provenant de l'intégration sur $[\|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_\infty, +\infty)$. On obtient

Cette preuve est peu satisfaisante car non constructive. Le calcul direct de l'équation (A.8) est beaucoup plus long.

Étape 1 : initialisation

Étape 2 : Récurrence

On utilise le fait que $\mathcal{P}(J)$ est vraie.

L'inversion somme intégrale est possible car la somme est finie et les fonctions intégrables.

$$\int_{\|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_{\infty}}^{+\infty} x_k^j e^{-\lambda x_k} dx_k = \|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_{\infty}^{j+1} e^{-\lambda \|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_{\infty}} + \sum_{\ell=1}^{j+1} \frac{j!}{(j+1-\ell)!} \frac{1}{\lambda^{\ell}} \|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_{\infty}^{j+1-\ell} e^{-\lambda \|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_{\infty}}. \quad (\text{A.10})$$

En raccordant le résultat pour $j = 0$, on obtient

$$\begin{aligned} p(\mathbf{x}_{\setminus \mathcal{K}_{J+1}}) &= \frac{2^{J+1}}{C_N(\lambda)} e^{-\lambda \|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_{\infty}} \\ &\quad \times \sum_{j=0}^J \binom{J}{j} \frac{(J-j)!}{\lambda^{J-j}} \left(\|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_{\infty}^{j+1} + \sum_{\ell=1}^{j+1} \frac{j!}{(j+1-\ell)!} \frac{1}{\lambda^{\ell}} \|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_{\infty}^{j+1-\ell} \right). \end{aligned}$$

À cette étape de la preuve, la densité $p(\mathbf{x}_{\setminus \mathcal{K}_{J+1}})$ est le produit d'une fonction exponentielle et d'un polynôme en $\|\mathbf{x}_{\setminus \mathcal{K}_{J+1}}\|_{\infty}$. Il reste pour prouver $\mathcal{P}(J+1)$ à rassembler les termes de même degré. On distinguera trois cas dans les trois prochains paragraphes.

Il n'y a qu'un seul terme de degré $J+1$, provenant de la somme pour $j = J$. La valeur associée est

Termes de degré $J+1$.

$$\binom{J}{J} \frac{(J-J)!}{\lambda^{J-J}} = 1 = \binom{J+1}{J+1} \frac{(J+1-J-1)!}{\lambda^{J+1-J-1}}. \quad (\text{A.11})$$

Le terme de degré 0 apparaît pour toutes les valeurs de j , lorsque $\ell = j+1$. La réunion de ces termes se réécrit

Termes de degré 0.

$$\begin{aligned} \sum_{j=0}^J \binom{J}{j} \frac{(J-j)!}{\lambda^{J-j}} \times \frac{j!}{\lambda^{j+1}} &= \frac{1}{\lambda^{J+1}} \sum_{j=0}^J \binom{J}{j} (J-j)! j! \\ &= \binom{J+1}{0} \frac{(J+1)!}{\lambda^{J+1}}. \end{aligned} \quad (\text{A.12})$$

Les termes de degré $0 < p < J+1$ proviennent de tous les termes de la somme pour tout $j > p$ tels que $\ell = j+1-p$. D'où

Termes de degré $0 < p < J+1$.

$$\begin{aligned} \binom{J}{p-1} \frac{(J-p+1)!}{\lambda^{J-p+1}} + \sum_{j=p}^J \binom{J}{j} \frac{(J-j)!}{\lambda^{J-j}} \frac{j!}{p!} \frac{1}{\lambda^{j+1-p}} \\ &= \frac{1}{\lambda^{J+1-p}} \left(\binom{J}{p-1} (J-p+1)! + \frac{1}{p!} \sum_{j=p}^J \binom{J}{j} (J-j)! j! \right) \\ &= \frac{(J+1-p)!}{\lambda^{J+1-p}} \left(\binom{J}{p-1} + \frac{J!}{p!(J-p)!} \right) \\ &= \frac{(J+1-p)!}{\lambda^{J+1-p}} \binom{J+1}{p}. \end{aligned} \quad (\text{A.13})$$

Donc $\mathcal{P}(j+1)$ est vraie, et la propriété 2.3 est prouvée par récurrence.

A.3.2 Convergence des marginales

Le théorème 2.1 concerne la convergence de la suite des marginales renormalisées vers une loi uniforme. Soit N un entier. La densité de la marginale de la loi *démocratique* est un mélange de N densités Gamma.¹ Alors

$$\begin{aligned}\phi_{\frac{\lambda}{N}X_N}(t) &= \phi_{X_N}\left(\frac{\lambda}{N}t\right) \\ &= \frac{1}{2N} \sum_{\ell=1}^N \int_{\mathbb{R}} \frac{\lambda^\ell}{\Gamma(\ell)} |x_n^{\ell-1}| \exp\left(-\lambda|x_n| + i\frac{\lambda}{N}tx_n\right) dx_n \\ &= \frac{1}{2N} \sum_{\ell=1}^N \left(1 - i\frac{t}{N}\right)^{-\ell} - \frac{1}{2N} \sum_{\ell=1}^N \left(1 + i\frac{t}{N}\right)^{-\ell} \\ &= \frac{\left(1 - i\frac{t}{N}\right)^{-N} - \left(1 + i\frac{t}{N}\right)^{-N}}{2it}.\end{aligned}$$

Il reste à analyser le comportement asymptotique de la fonction caractéristique. Soit t un réel, et N un entier suffisamment grand pour que $\frac{t}{N} < 1$. Alors, $|i\frac{t}{N}| < 1$ et $i\frac{t}{N} \neq 1$, et le logarithme complexe $\ln(1 - i\frac{t}{N})$ est bien défini. Alors

$$\ln\left(1 - i\frac{t}{N}\right) = -i\frac{t}{N} + N\mathcal{O}\left(\frac{t^2}{N^2}\right). \quad (\text{A.14})$$

D'où

$$\begin{aligned}\left(1 - i\frac{t}{N}\right)^{-N} &= \exp\left(-N \log\left(1 - i\frac{t}{N}\right)\right) \\ &= \exp\left(it + \mathcal{O}\left(\frac{t^2}{N}\right)\right).\end{aligned}$$

De même, $\left(1 + i\frac{t}{N}\right)^{-N} = \exp\left(-it + \mathcal{O}\left(\frac{t^2}{N}\right)\right)$. Ainsi, quand N tend vers l'infini, la suite de fonctions $(\phi_{\frac{\lambda}{N}X_N})$ converge simplement vers la fonction $t \rightarrow \frac{e^{it} - e^{-it}}{2it}$, qui est la fonction caractéristique de la loi uniforme sur $[-1, 1]$.

A.4 LOIS CONDITIONNELLES

A.4.1 Divers résultats associés aux cônes

Les preuves associées aux équations (2.14) à (2.18) du lemme 2.1 seront découpées par sous-section.

Preuve du résultat équation (2.14)

Le résultat équation (2.14) s'obtient en appliquant le théorème de Bayes

$$p(x_n | \mathbf{x} \in \mathcal{C}_n) = \frac{p(x_n, \mathbf{x} \in \mathcal{C}_n)}{P[\mathbf{x} \in \mathcal{C}_n]},$$

où $P[\mathbf{x} \in \mathcal{C}_n]$ est donné à la propriété 2.4. D'où, en intégrant vis-à-vis des

On démontre la convergence en distribution par une convergence de la suite des fonctions caractéristiques.
1. cf. corollaire 2.1.

Ce résultat s'obtient en séparant les intégrales sur \mathbb{R}_- et \mathbb{R}_+ et en reconnaissant la somme des termes d'une suite géométrique.

C'est une fonction à valeur dans \mathbb{C} .

Ce résultat fait appel à la formule de Taylor

$N - 1$ autres composantes

$$\begin{aligned} p(x_n \mid \mathbf{x} \in \mathcal{C}_n) &= \frac{N2^{N-1}}{C_N(\lambda)} \int_0^{|x_n|} \exp(-\lambda |x_n|) dx_{i \neq n} \\ &= \frac{1}{2} \frac{\lambda^N}{\Gamma(N)} |x_n|^{N-1} \exp(-\lambda |x_n|), \end{aligned}$$

qui est bien une loi Gamma symétrisée.

Preuve du résultat équation (2.15)

Il faut ici prouver que, conditionnellement au cône contenant le vecteur \mathbf{x} et à la valeur de $\|\mathbf{x}\|_\infty$, la loi des composantes *non dominantes* est le produit de $N - 1$ lois uniformes indépendantes et uniformément distribuées.

D'abord

$$p(\mathbf{x}_{\setminus n} \mid x_n, \mathbf{x} \in \mathcal{C}_n) = \frac{P[\mathbf{x} \in \mathcal{C}_n \mid \mathbf{x}] p(\mathbf{x})}{p(x_n \mid \mathbf{x} \in \mathcal{C}_n) P[\mathbf{x} \in \mathcal{C}_n]}$$

où $p(x_n \mid \mathbf{x} \in \mathcal{C}_n)$ et $P[\mathbf{x} \in \mathcal{C}_n]$ sont respectivement données par les équations (2.13) et (2.14). De plus, $P[\mathbf{x} \in \mathcal{C}_n \mid \mathbf{x}]$ est la fonction indicatrice² du cône \mathcal{C}_n . D'où

$$\begin{aligned} p(\mathbf{x}_{\setminus n} \mid x_n, \mathbf{x} \in \mathcal{C}_n) &= \frac{2 \Gamma(N)}{\lambda^N |x_n|^{N-1} e^{-\lambda |x_n|}} \frac{N e^{-\lambda \|\mathbf{x}\|_\infty}}{C_N(\lambda)} \mathbb{1}_{\substack{|x_j| \leq |x_n| \\ \forall j \neq n}}(\mathbf{x}) \\ &= \frac{1}{2^{N-1} |x_n|^{N-1}} \mathbb{1}_{\substack{|x_j| \leq |x_n| \\ \forall j \neq n}}(\mathbf{x}). \end{aligned} \quad (\text{A.15})$$

L'équation (A.15) correspond exactement au produit des densités de $N - 1$ lois uniformes sur $[-|x_n|, |x_n|]$ indépendantes.

Preuve du résultat équation (2.16)

Le résultat équation (2.16) provient de la combinaison du théorème de Bayes et d'une marginalisation vis-à-vis de x_n

$$\begin{aligned} p(\mathbf{x}_{\setminus n} \mid \mathbf{x} \in \mathcal{C}_n) &= \frac{2N}{C_N(\lambda)} \int_{\|\mathbf{x}_{\setminus n}\|_\infty}^{+\infty} \exp(-\lambda |x_n|) dx_n \\ &= \frac{1}{C_{N-1}(\lambda)} \exp\left(-\lambda \|\mathbf{x}_{\setminus n}\|_\infty\right), \end{aligned}$$

qui est bien la loi *démocratique* $\mathcal{D}_{N-1}(\lambda)$.

Preuve du résultat équation (2.17)

En appliquant une nouvelle fois le théorème de Bayes

$$\begin{aligned} P[\mathbf{x} \in \mathcal{C}_n \mid \mathbf{x}_{\setminus n}] &= \frac{p(\mathbf{x} \in \mathcal{C}_n, \mathbf{x}_{\setminus n})}{p(\mathbf{x}_{\setminus n})} \\ &= \frac{1}{P[\mathbf{x}_{\setminus n}]} \times \int_{\|\mathbf{x}_{\setminus n}\|_\infty}^{+\infty} p(\mathbf{x}) dx_n \\ &= \frac{1}{P[\mathbf{x}_{\setminus n}]} \times \int_{\|\mathbf{x}_{\setminus n}\|_\infty}^{+\infty} \frac{\lambda^N}{2^N N!} \exp(-\lambda \|\mathbf{x}\|_\infty) dx_n \\ &= \frac{1}{P[\mathbf{x}_{\setminus n}]} \times 2 \int_{\|\mathbf{x}_{\setminus n}\|_\infty}^{+\infty} \frac{\lambda^N}{2^N N!} \exp(-\lambda x_n) dx_n \\ &= \frac{1}{p(\mathbf{x}_{\setminus n})} \times \frac{\lambda^{N-1}}{2^{N-1} N!} \exp\left(-\lambda \|\mathbf{x}_{\setminus n}\|_\infty\right) \end{aligned}$$

Il s'agit de deux applications imbriquées du théorème de Bayes.

Et $p(\mathbf{x})$ est la densité de la loi *démocratique*.

2. au sens des mesures

Rappel : $p(\mathbf{x})$ est la densité de la loi *démocratique*.

où la densité marginale $p(\mathbf{x}_{\setminus n})$ est donnée en (2.11) et (2.12). D'où le résultat.

Preuve du résultat équation (2.18)

On utilise une dernière fois la règle de Bayes

$$\begin{aligned} p(\mathbf{x}_{\setminus n} \mid \mathbf{x} \notin \mathcal{C}_n) &= \frac{p(\mathbf{x}_{\setminus n}, \mathbf{x} \notin \mathcal{C}_n)}{P[\mathbf{x} \notin \mathcal{C}_n]} \\ &= \frac{1}{P[\mathbf{x} \notin \mathcal{C}_n]} \times \int_0^{\|\mathbf{x}_{\setminus n}\|_\infty} p(\mathbf{x}) d\mathbf{x}_n \\ &= \frac{1}{P[\mathbf{x} \notin \mathcal{C}_n]} 2 \int_0^{\|\mathbf{x}_{\setminus n}\|_\infty} \frac{\lambda^N}{2^N N!} \exp(-\lambda \|x\|_\infty) d\mathbf{x}_n \\ &= \frac{1}{P[\mathbf{x} \notin \mathcal{C}_n]} \times \frac{\lambda^N}{2^{N-1} N!} \|\mathbf{x}_{\setminus n}\|_\infty \exp(-\lambda \|\mathbf{x}_{\setminus n}\|_\infty). \end{aligned}$$

Rappel : $p(\mathbf{x})$ est la densité de la loi démocratique.

Or, $P[\mathbf{x} \notin \mathcal{C}_n] = 1 - P[\mathbf{x} \in \mathcal{C}_n] = \frac{N-1}{N}$ d'après la propriété 2.4. D'où le résultat.

A.4.2 Loi conditionnelle d'une composante

La stratégie pour déterminer la densité conditionnelle $p(x_n \mid \mathbf{x}_{\setminus n})$ de x_n consiste à conditionner par rapport au cône \mathcal{C}_n , selon

$$p(x_n \mid \mathbf{x}_{\setminus n}) = \underbrace{p(x_n \mid \mathbf{x}_{\setminus n}, \mathbf{x} \in \mathcal{C}_n)}_{(b)} P[\mathbf{x} \in \mathcal{C}_n \mid \mathbf{x}_{\setminus n}] + \underbrace{p(x_n \mid \mathbf{x}_{\setminus n}, \mathbf{x} \notin \mathcal{C}_n)}_{(c)} P[\mathbf{x} \notin \mathcal{C}_n \mid \mathbf{x}_{\setminus n}]. \quad (\text{A.16})$$

Les quatre prochaines sous-sections sont consacrées aux calculs de ces termes.

Termes (a) $P[\mathbf{x} \in \mathcal{C}_n \mid \mathbf{x}_{\setminus n}]$ est donné Éq. (2.17), et $P[\mathbf{x} \notin \mathcal{C}_n \mid \mathbf{x}_{\setminus n}] = 1 - P[\mathbf{x} \in \mathcal{C}_n \mid \mathbf{x}_{\setminus n}]$.

Terme (b) Le terme $p(x_n \mid \mathbf{x}_{\setminus n}, \mathbf{x} \in \mathcal{C}_n)$ s'obtient après deux inversions de Bayes

$$p(x_n \mid \mathbf{x}_{\setminus n}, \mathbf{x} \in \mathcal{C}_n) = \frac{P(x_{\setminus n}, x_n)}{p(x_{\setminus n} \mid \mathbf{x} \in \mathcal{C}_n) P[\mathbf{x} \in \mathcal{C}_n]} \mathbf{1}_{\mathcal{C}_n}(\mathbf{x}). \quad (\text{A.17})$$

Or $p(x_{\setminus n}, x_n)$ correspond à la densité de la loi démocratique, \mathbf{x} appartient au cône \mathcal{C}_n par construction et $P[\mathbf{x} \in \mathcal{C}_n]$ est donnée (2.13). Alors

$$\begin{aligned} p(x_{\setminus n} \mid \mathbf{x} \in \mathcal{C}_n) &= \int_{\mathbb{R}} \frac{1}{P[\mathbf{x} \in \mathcal{C}_n]} p(\mathbf{x}, \mathbf{x} \in \mathcal{C}_n) d\mathbf{x}_n \\ &= N \int_{\|\mathbf{x}_{\setminus n}\|_\infty}^{+\infty} p(\mathbf{x}) d\mathbf{x}_n \\ &= 2N \int_{\|\mathbf{x}_{\setminus n}\|_\infty}^{+\infty} \frac{\lambda^N}{2^N N!} \exp(-\lambda |x_n|) d\mathbf{x}_n \\ &= \frac{\lambda^{(N-1)}}{2^{(N-1)} (N-1)!} \exp(-\lambda \|\mathbf{x}_{\setminus n}\|_\infty). \quad (\text{A.18}) \end{aligned}$$

$p(\mathbf{x})$ désigne la densité de la loi démocratique.

On obtient en combinant les équations (2.2), (2.13) et (A.18), dans (A.17)

$$p(x_n \mid \mathbf{x}_{\setminus n}, \mathbf{x} \in \mathcal{C}_n) = \frac{\lambda}{2} \exp(-\lambda(|x_n| - \|\mathbf{x}_{\setminus n}\|_\infty)) \mathbf{1}_{|x_n| \geq \|\mathbf{x}_{\setminus n}\|_\infty}(x_n). \quad (\text{A.19})$$

Terme (c) Comme pour le résultat équation (A.17), deux lois de Bayes imbriquées donnent

$$p(x_n | x_{\setminus n}, \mathbf{x} \notin \mathcal{C}_n) = \frac{1}{p(x_{\setminus n} | \mathbf{x} \notin \mathcal{C}_n) P[\mathbf{x} \notin \mathcal{C}_n]} p(x_{\setminus n}, x_n) \mathbf{1}_{\mathcal{C}_n^c}(\mathbf{x}). \quad (\text{A.20})$$

Le terme $p(x_{\setminus n} | \mathbf{x} \notin \mathcal{C}_n)$, de la même façon que (A.18)

$$p(x_{\setminus n} | \mathbf{x} \notin \mathcal{C}_n) = \frac{\lambda \|x_{\setminus n}\|_\infty}{N-1} \frac{\lambda^{(N-1)}}{2^{(N-1)}(N-1)!} \exp\left(-\lambda \|x_{\setminus n}\|_\infty\right). \quad (\text{A.21})$$

De même que pour l'équation (A.19), insérer l'équation (A.21) dans l'équation (A.20) donne

$$p(x_n | x_{\setminus n}, \mathbf{x} \notin \mathcal{C}_n) = \frac{1}{2 \|x_{\setminus n}\|_\infty} \mathbf{1}_{|x_n| \leq \|x_{\setminus n}\|_\infty}(x_n). \quad (\text{A.22})$$

Finalement, la loi conditionnelle équation (2.19) s'obtient en combinant les équations (2.13), (A.19) et (A.22) dans l'équation (A.16).

A.5 LOIS CONDITIONNELLES A POSTERIORI D'UNE COMPOSANTE

CETTE ANNEXE présente le calcul de la loi *a posteriori* d'une composante conditionnellement aux autres, décrite dans la partie 2.2.2.a.

À l'instar de la méthode proposée partie 2.1.2.a, on marginalise la loi conditionnelle vis-à-vis des cônes \mathcal{C}_n en utilisant la formule des probabilités totales. Pour simplifier les notations, notons ω_n l'événement $\mathbf{x} \in \mathcal{C}_n$ et $\bar{\omega}_n$ son complémentaire. Alors

$$p(x_n | \mathbf{y}, \sigma^2, \mathbf{x}_{\setminus n}, \lambda) \propto \underbrace{p(\mathbf{y} | \mathbf{x}, \sigma^2) p(x_n | \mathbf{x}_{\setminus n}, \omega_n) p(\omega_n | \mathbf{x}_{\setminus n})}_{\text{Parties exponentielles}} + \underbrace{p(\mathbf{y} | \mathbf{x}, \sigma^2) p(x_n | \mathbf{x}_{\setminus n}, \bar{\omega}_n) p(\bar{\omega}_n | \mathbf{x}_{\setminus n})}_{\text{Partie uniforme}}. \quad (\text{A.23})$$

Comme pour la loi conditionnelle d'une composante d'un vecteur distribué suivant la loi *démocratique*, la loi conditionnelle *a posteriori* est un mélange à trois composantes. On traitera à part dans les deux prochaines sous-parties les membres intervenant dans l'équation (A.23). Notons préalablement Δ le bloc

$$\Delta = \left(\frac{1}{2\pi\sigma^2}\right)^{M/2} \exp\left[\frac{1}{2\sigma^2} \left(\mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \left(\sum_{\ell \neq n} x_\ell \mathbf{h}_\ell\right) + \left(\sum_{\ell \neq n} x_\ell \mathbf{h}_\ell\right)^T \left(\sum_{\ell \neq n} x_\ell \mathbf{h}_\ell\right)\right)\right]. \quad (\text{A.24})$$

Pour simplifier la compréhension du calcul, trois termes sont mis en évidence avec de la couleur : **une constante commune aux trois composantes du mélange**, **les poids de chaque terme du mélange** et **la loi**.

A.5.1 Partie uniforme

Dans ce premier morceaux, on sait que \mathbf{x} n'appartient pas au cône \mathcal{C}_n . La posterior conditionnelle fait alors intervenir le produit d'une loi uniforme sur \mathcal{I} défini équation (2.38) et de la vraisemblance Gaussienne.

$$\begin{aligned}
p(\mathbf{y}|\mathbf{x},\sigma^2) p(x_n|\mathbf{x}_{\setminus n},\bar{\omega}_n) p(\bar{\omega}_n|\mathbf{x}_{\setminus n}) &= \Delta \times \exp \left[-\frac{1}{2\sigma^2} \left(x_n^2 \|\mathbf{h}_n\|_2 - 2x_n \mathbf{h}_n^T (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) \right) \right] \\
&\times \frac{\lambda \|\mathbf{x}_{\setminus n}\|_\infty}{1 + \lambda \|\mathbf{x}_{\setminus n}\|_\infty} \times \frac{1}{2 \|\mathbf{x}_{\setminus n}\|_\infty} \mathbf{1}_{|x_n| < \|\mathbf{x}_{\setminus n}\|_\infty} (x_n) \\
&= \Delta \times \frac{1}{1 + \lambda \|\mathbf{x}_{\setminus n}\|_\infty} \times \frac{\lambda}{2} \mathbf{1}_{|x_n| < \|\mathbf{x}_{\setminus n}\|_\infty} (x_n) \\
&\times \exp \left(-\frac{\|\mathbf{h}_n\|_2}{2\sigma^2} \left\| x_n - \frac{\mathbf{h}_n^T}{\|\mathbf{h}_n\|_2} (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) \right\|_2 \right) \exp \left[+\frac{\|\mathbf{h}_n\|_2}{2\sigma^2} \left(\frac{1}{\|\mathbf{h}_n\|_2} \mathbf{h}_n^T (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) \right)^2 \right] \\
&= \frac{\Delta}{1 + \lambda \|\mathbf{x}_{\setminus n}\|_\infty} \frac{\lambda}{2} \times \exp \left[+\frac{\|\mathbf{h}_n\|_2}{2\sigma^2} \left(\frac{1}{\|\mathbf{h}_n\|_2} \mathbf{h}_n^T (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) \right)^2 \right] \\
&\times \exp \left(-\frac{\|\mathbf{h}_n\|_2}{2\sigma^2} \left\| x_n - \frac{\mathbf{h}_n^T}{\|\mathbf{h}_n\|_2} (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) \right\|_2 \right) \mathbf{1}_{|x_n| < \|\mathbf{x}_{\setminus n}\|_\infty} (x_n) \tag{A.25}
\end{aligned}$$

il reste à mettre en évidence les constantes de normalisation pour aboutir à une loi normale tronquée.

$$\begin{aligned}
p(\mathbf{y}|\mathbf{x},\sigma^2) p(x_n|\mathbf{x}_{\setminus n},\bar{\omega}_n) p(\bar{\omega}_n|\mathbf{x}_{\setminus n}) &= \frac{\Delta}{1 + \lambda \|\mathbf{x}_{\setminus n}\|_\infty} \frac{\lambda}{2} \\
&\times \exp \left[+\frac{\|\mathbf{h}_n\|_2}{2\sigma^2} \left(\frac{1}{\|\mathbf{h}_n\|_2} \mathbf{h}_n^T (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) \right)^2 \right] \times 2 \int_{x=0}^{\|\mathbf{x}_{\setminus n}\|_\infty} \mathcal{N}(\mu_{2n}, s_n^2) dx \\
&\times \mathcal{N}_{[-\|\mathbf{x}_{\setminus n}\|_\infty, \|\mathbf{x}_{\setminus n}\|_\infty]}(\mu_{2n}, s_n^2) \tag{A.26}
\end{aligned}$$

où

$$\mu_{2n} = \frac{\mathbf{h}_n^T}{\|\mathbf{h}_n\|_2} (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) \tag{A.27}$$

$$s_n^2 = \frac{\sigma^2}{\|\mathbf{h}_n\|_2}. \tag{A.28}$$

A.5.2 Parties exponentielles

Les calculs sont similaires pour la partie exponentielle. Commençons par la partie positive, *i.e.*, lorsque $\mathbf{x} \in \mathcal{C}_n^+$

$$\begin{aligned}
p(\mathbf{y}|\mathbf{x},\sigma^2) p(x_n|\mathbf{x}_{\setminus n}, \mathbf{x} \in \mathcal{C}_n^+) p(\mathbf{x} \in \mathcal{C}_n^+|\mathbf{x}_{\setminus n}) &= \Delta \times \exp \left[\frac{1}{2\sigma^2} \left(x_n^2 \|\mathbf{h}_n\|_2 - 2x_n \mathbf{h}_n^T (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) \right) \right] \\
&\times \frac{\lambda}{2} \exp \left[-\lambda x_n + \lambda \|\mathbf{x}_{\setminus n}\|_\infty \right] \times \frac{1}{1 + \lambda \|\mathbf{x}_{\setminus n}\|_\infty} \mathbf{1}_{x_n \geq \|\mathbf{x}_{\setminus n}\|_\infty} (x_n) \\
&= \frac{\Delta}{1 + \lambda \|\mathbf{x}_{\setminus n}\|_\infty} \frac{\lambda}{2} \\
&\times \exp \left[+ \frac{\|\mathbf{h}_n\|_2}{2\sigma^2} \left(\frac{1}{\|\mathbf{h}_n\|_2} \mathbf{h}_n^T (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) - \sigma^2 \lambda \right)^2 \right] \exp \left[+\lambda \|\mathbf{x}_{\setminus n}\|_\infty \right] \\
&\times \exp \left[- \frac{\|\mathbf{h}_n\|_2}{2\sigma^2} \left\| x_n - \frac{1}{\|\mathbf{h}_n\|_2} \left(\mathbf{h}_n^T (\mathbf{x} - \sum_{i \neq n} x_i \mathbf{h}_i) - \sigma^2 \lambda \right) \right\|_2 \right] \mathbf{1}_{x_n \geq \|\mathbf{x}_{\setminus n}\|_\infty} (x_n)
\end{aligned} \tag{A.29}$$

D'où, en renormalisant

$$\begin{aligned}
p(\mathbf{y}|\mathbf{x},\sigma^2) p(x_n|\mathbf{x}_{\setminus n}, \mathbf{x} \in \mathcal{C}_n^+) p(\mathbf{x} \in \mathcal{C}_n^+|\mathbf{x}_{\setminus n}) &= \frac{\Delta}{1 + \lambda \|\mathbf{x}_{\setminus n}\|_\infty} \frac{\lambda}{2} \\
&\times \exp \left[+ \frac{\|\mathbf{h}_n\|_2}{2\sigma^2} \left(\frac{1}{\|\mathbf{h}_n\|_2} \mathbf{h}_n^T (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) - \sigma^2 \lambda \right)^2 \right] \exp \left[+\lambda \|\mathbf{x}_{\setminus n}\|_\infty \right] \int_{\|\mathbf{x}_{\setminus n}\|_\infty}^{+\infty} \mathcal{N}(t; \mu_{3n}, s_n^2) dt \\
&\times \mathcal{N}_{[\|\mathbf{x}_{\setminus n}\|_\infty, +\infty)}(\mu_{3n}, s_n^2),
\end{aligned} \tag{A.30}$$

avec :

$$\mu_{3n} = \frac{1}{\|\mathbf{h}_n\|_2} \left(\mathbf{h}_n^T (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) - \sigma^2 \lambda \right). \tag{A.31}$$

De même pour la partie négative, *i.e.*, lorsque $\mathbf{x} \in \mathcal{C}_n^-$

$$\begin{aligned}
p(\mathbf{y}|\mathbf{x},\sigma^2) p(x_n|\mathbf{x}_{\setminus n}, \mathbf{x} \in \mathcal{C}_n^-) p(\mathbf{x} \in \mathcal{C}_n^-|\mathbf{x}_{\setminus n}) &= \frac{\Delta}{1 + \lambda \|\mathbf{x}_{\setminus n}\|_\infty} \frac{\lambda}{2} \\
&\times \exp \left[+ \frac{\|\mathbf{h}_n\|_2}{2\sigma^2} \left(\frac{1}{\|\mathbf{h}_n\|_2} \mathbf{h}_n^T (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) + \sigma^2 \lambda \right)^2 \right] \exp \left[+\lambda \|\mathbf{x}_{\setminus n}\|_\infty \right] \int_{-\infty}^{-\|\mathbf{x}_{\setminus n}\|_\infty} \mathcal{N}(t; \mu_{1n}, s_n^2) dt \\
&\times \mathcal{N}_{[-\infty, \|\mathbf{x}_{\setminus n}\|_\infty)}(\mu_{1n}, s_n^2),
\end{aligned} \tag{A.32}$$

avec :

$$\mu_{1n} = \frac{1}{\|\mathbf{h}_n\|_2} \left(\mathbf{h}_n^T (\mathbf{y} - \sum_{i \neq n} x_i \mathbf{h}_i) + \sigma^2 \lambda \right). \tag{A.33}$$

A.6 DENSITÉ DE LA LOI DÉMOCRATIQUE RENORMALISÉE

CETTE PREUVE est un exercice classique de changement de variable multivarié en probabilité. Nous réaliserons directement le calcul dans le cas de la loi démocratique à deux paramètres, présentée dans la partie 2.6.3. Le résultat pour la loi démocratique s'obtient en prenant $\gamma = 1$.

Soit $\mathbf{X} = [X_1, \dots, X_N]$ un vecteur aléatoire distribué suivant la loi démocratique à deux paramètres, λ et γ , en dimension N . Nous allons caractériser la distribution de la variable aléatoire $\mathbf{Y} = (\sum_{n=1}^N X_n)^{-1} \mathbf{X}$. Considérons le changement de variable Φ_N

$$\begin{aligned} \mathbb{R}_+^N &\longrightarrow [0, 1]^{N-1} \times \mathbb{R}_+ \\ x_1, \dots, x_N &\longmapsto y_1, \dots, y_{N-1}, s \end{aligned} \quad (\text{A.34})$$

La matrice jacobienne associée à ce changement de variable est

$$J_\Phi = \frac{\partial(x_1, \dots, x_N)}{\partial(y_1, \dots, y_{N-1}, s)} \begin{pmatrix} s & 0 & \dots & 0 & y_1 \\ 0 & s & \dots & 0 & y_2 \\ 0 & 0 & \ddots & 0 & \vdots \\ 0 & \dots & 0 & s & y_{N-1} \\ -s & -s & \dots & -s & 1 - \sum_{n=1}^{N-1} y_n \end{pmatrix}. \quad (\text{A.35})$$

Le déterminant de la matrice jacobienne se calcule facilement via le pivot de Gauss : il suffit d'ajouter les $N - 1$ premières lignes à la dernière pour obtenir une matrice triangulaire supérieure, d'où $|J_\Phi| = s^{N-1}$.

La densité f_Y du vecteur renormalisé s'écrit alors

$$\begin{aligned} f_Y(y_1, \dots, y_{N-1}, s) &= f_X(x_1, \dots, x_N) s^{N-1} \\ &= \frac{\lambda^{N\gamma} \gamma^N}{\Gamma(N\gamma + 1)} s^{N\gamma - N} \prod_{n=1}^N (y_n)^{\gamma-1} e^{-\lambda s \|\mathbf{y}\|_\infty} s^{N-1}. \end{aligned}$$

Remarquons qu'à ce stade, on peut déjà affirmer que les variables (y_1, \dots, y_{N-1}) et s ne sont pas indépendantes, contrairement aux mêmes quantités sous la loi Gamma.

En raison de la perte d'indépendance il faut marginaliser vis-à-vis de s si l'on souhaite obtenir la loi d'un vecteur démocratique, d'où

$$\begin{aligned} \int_{\mathbb{R}_+} f_Y(y_1, \dots, y_{N-1}, s) ds &= \frac{\lambda^{N\gamma} \gamma^N}{\Gamma(N\gamma + 1)} \frac{\Gamma(N\gamma)}{\lambda^{N\gamma} \|\mathbf{y}\|_\infty^{N\gamma}} \prod_{n=1}^N y_n^{\gamma-1} \\ &= \frac{\gamma^{N-1}}{N \|\mathbf{y}\|_\infty^{N\gamma}} \prod_{n=1}^N y_n^{\gamma-1}. \end{aligned} \quad (\text{A.36})$$

Estimation bayésienne non paramétrique de sous-espaces

B

B.1 ÉCHANTILLONNAGE SUR LA VARIÉTÉ DE STIEFEL

CETTE PARTIE est consacrée à l'échantillonnage de variables aléatoires sur la sphère. La première rappelle l'échantillonnage de la loi uniforme. La seconde partie décrit en détails l'échantillonnage de la loi *Bingham matricielle*.

B.1.1 Loi uniforme

L'échantillonnage de la loi uniforme sur la variété de Stiefel \mathcal{S}_K^D se fait simplement en suivant la procédure décrite par Hoff, 2007 que nous allons rappeler. L'algorithme ne nécessite que des loi normales et est rappelé à l'algorithme B.1.

Algorithme B.1 : Échantillonneur exact de la loi uniforme sur la variété de Stiefel.

Entrées : dimensions K, D

- 1 Générer $\mathbf{v}_1 \sim \mathcal{N}(0, \mathbb{I}_D)$;
- 2 $\mathbf{p}_1 = \mathbf{v}_1 / \|\mathbf{v}_1\|_2$;
- 3 **pour** $d \leftarrow 2$ à D **faire**
- 4 Déterminer \mathbf{N}_d , une base de l'orthogonal de $\mathbf{p}_1, \dots, \mathbf{p}_{d-1}$;
- 5 Générer $\mathbf{v}_d \sim \mathcal{N}(0, \mathbb{I}_{D+1-d})$;
- 6 $\mathbf{p}_d = \mathbf{v}_d / \|\mathbf{v}_d\|_2$;
- 7 **fin**

Output : $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_K] \sim \mathcal{U}_{\mathcal{S}_K^D}$.

B.1.2 Distribution matricielle Bingham

La loi de Bingham est une loi sur la sphère unitaire de \mathbb{R}^D notée \mathcal{S}_D . On rappelle que sa densité par rapport à la mesure de Haar sur la sphère, notée \mathcal{H} , est

$$\mathbf{p}_B(\mathbf{x}|\mathbf{A}) \propto \exp(\mathbf{x}^T \mathbf{A} \mathbf{x}) \mathcal{H}_{\mathcal{S}_D}(\mathbf{x}), \quad \mathbf{x} \in \mathcal{S}_D \quad (\text{B.1})$$

où \mathbf{A} est une matrice symétrique de taille $D \times D$. Au passage, la constante de normalisation de la loi s'exprime en fonction de la fonction hypergéométrique de paramètre matricielle (Kume et Walker, 2009). La méthode proposée s'appuie sur la méthodologie développée dans Diaconis et al.,

2013. L'algorithme consiste principalement à réécrire la densité de la distribution de Bingham contre la mesure de Lebesgue.

B.1.2.a Transformation du produit en somme

Le premier changement de variable vise à transformer le produit matriciel équation (B.1) en somme. Notons \mathbf{E} la matrice formée des vecteurs propres de \mathbf{A} et Λ la matrice diagonale contenant les valeurs propres de \mathbf{A} . Considérons le changement de variable linéaire $\phi_1 : \mathbf{x} \rightarrow \mathbf{z} = \mathbf{E}\mathbf{x}$. La densité équation (B.1) se réécrit par rapport à la mesure de Haar sur \mathcal{S}_D ,

$$p_B(\mathbf{z}|\mathbf{A}) \propto \exp\left(\sum_{d=1}^D \lambda_d z_d^2\right) \mathcal{H}_{\mathcal{S}_D}(\mathbf{z}), \quad \mathbf{z} \in \mathcal{S}_D \quad (\text{B.2})$$

B.1.2.b Formule de l'aire

La formule de l'aire ou *area formula* en anglais, permet de réécrire la densité équation (B.2) par rapport à la mesure de Lebesgue. En bref, on va plutôt chercher à échantillonner une paramétrisation de la sphère. Le bon outil pour traduire la géométrie de cette variété est le jacobien de la paramétrisation.

La dimension de la variété \mathcal{S}_D est $D - 1$ en raison de la contrainte $\|\mathbf{y}\|^2 = 1$. Soit \mathbf{y} un élément de \mathcal{S}_D . Supposons sans perte de généralité $y_1 \dots y_{D-1}$ libres, et $y_D^2 = \pm \left(1 - \sum_{i=1}^{D-1} y_i^2\right)$. Pour résoudre l'ambiguïté au niveau du signe, considérons

$$\mathcal{S}_D^+ = \{\mathbf{y} \in \mathcal{S}_D : y_D \geq 0\} \quad (\text{B.3})$$

$$\mathcal{S}_D^- = \{\mathbf{y} \in \mathcal{S}_D : y_D < 0\}. \quad (\text{B.4})$$

Par symétrie de la norme, on obtient directement

$$p(\mathbf{y}) = \frac{1}{2}p(\mathbf{y} | \mathbf{y} \in \mathcal{S}_D^+) + \frac{1}{2}p(\mathbf{y} | \mathbf{y} \in \mathcal{S}_D^-) \quad (\text{B.5})$$

Soit les paramétrisations

$$\begin{aligned} \phi_2^+ : \left\{ (y_1, \dots, y_{D-1}) : \sum_{m=1}^{D-1} y_m^2 \leq 1 \right\} &\longrightarrow \mathcal{S}_D^+ \\ (y_1, \dots, y_{D-1}) &\longrightarrow \left(y_1, \dots, y_{D-1}, \sqrt{1 - \sum_{d=1}^{D-1} y_d^2} \right), \end{aligned} \quad (\text{B.6})$$

et ϕ_2^- définie sur \mathcal{S}_D^- . Le sous jacobien du ϕ_2^+ est donné par

$$D_{\phi_2^+} = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \cdots & \frac{\partial f_1}{\partial y_{D-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_D}{\partial y_1} & \cdots & \frac{\partial f_D}{\partial y_{D-1}} \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \\ \frac{y_1}{(1-\sum y_i^2)^{3/2}} & \cdots & \frac{y_{D-1}}{(1-\sum y_i^2)^{2/2}} \end{bmatrix} \quad (\text{B.7})$$

Le jacobien $(J_{\phi_2^+})^2 = \det(D_{\phi_2^+}^T D_{\phi_2^+})$ est le déterminant de $I_{D-1} + \mathbf{v}^T \mathbf{v}$ où \mathbf{v} est donné par $\mathbf{v} = \frac{1}{(1-\sum y_i^2)^{1/2}} (y_1 \dots y_{D-1})$.

Ce déterminant vaut (Bishop, 2006, Annexe C)

$$(J_{\phi_2^+})^2 = 1 + \mathbf{v}\mathbf{v}^T = \frac{1}{|\mathbf{y}_D|^2}. \quad (\text{B.8})$$

La densité de la distribution de Bingham se réécrit par rapport à la mesure de Lebesgue

$$p(\mathbf{y} \mid E, \Lambda) \propto \exp\left(\sum_i \lambda_i y_i^2\right) \frac{1}{|\mathbf{y}_D|} \mathbb{1}_{y_D \geq 0}(\mathbf{y}) + \exp\left(\sum_i \lambda_i y_i^2\right) \frac{1}{|\mathbf{y}_D|} \mathbb{1}_{y_D < 0}(\mathbf{y}). \quad (\text{B.9})$$

B.1.2.c Échantillonneur de Gibbs

L'approche considérée dans Kume et Walker, 2006 y_d conditionnellement aux autres, à partir de équation (B.9). Toutefois, la conditionnelle $y_i \mid \mathbf{y}_{\setminus i}$ ne charge que le compacte $[0, 1 - \sum_{j \neq i} y_j]$, ce qui aboutie à de mauvaises propriétés de mélange.

Hoff, 2009 propose de considérer le modèle relâché suivant. Supposons que l'on s'intéresse à la composante y_1 . Le signe des y_i n'influe pas sur la densité, donc peuvent être tiré aléatoirement avec probabilité $\frac{1}{2}$. On considère le changement de variable

$$\phi_3 : y_1, \dots, y_{D-1} \rightarrow \theta, \mathbf{q}, \quad (\text{B.10})$$

où $\theta = y_1^2$ et $\mathbf{q} = [\frac{y_2^2}{1-y_1^2}, \dots, \frac{y_{D-1}^2}{1-y_1^2}]$. La densité équation (B.9) devient

$$p(\theta \mid \mathbf{q}) \propto \exp\left(\theta(\lambda_1 - \mathbf{q}^T \Lambda_{-1})\right) \theta^{-\frac{1}{2}} (1 - \theta)^{(D-3)/2}. \quad (\text{B.11})$$

L'équation (B.11) est une loi unidimensionnelle, définie sur $[0, 1]$. On génère des échantillons suivant cette loi en utilisant des méthodes de *rejet*.

B.1.2.d Algorithme

L'algorithme final est résumé ci-dessous.

Algorithme B.2 : Échantillonneur exact de la loi de *Bingham*.**Entrées :** Matrice paramètre \mathbf{A} , dimension D , Nombre d'itération N_{iter}

```

1 Déterminer  $\mathbf{E}, \Lambda$  telle que  $\mathbf{A} = \mathbf{E}^T \Lambda \mathbf{E}$ ;
2 Initialiser  $\mathbf{x}^{(0)} \sim \mathcal{N}(0, 1)$ ;
3 Normaliser  $\mathbf{x}^{(0)} = [\frac{x_1}{\|\mathbf{x}\|_2}, \dots, \frac{x_D}{\|\mathbf{x}\|_2}]$ ;
4  $\mathbf{y}^{(0)} = \mathbf{E}\mathbf{x}^{(0)}$ ;
5 pour  $t \leftarrow 1$  à  $N_{\text{iter}}$  faire
6   pour  $d \leftarrow 1$  à  $D$  faire
7      $\theta \triangleq y_d^{2(t)}$ ;
8      $\mathbf{q} \triangleq \frac{1}{1-\theta} \mathbf{y}_d^{(t)}$ ;
9     Tirer  $\theta$  suivant (B.11);
10    Tirer  $s_d$  uniformément dans  $\{-1, 1\}$ ;
11     $y_d^{(t+1)} = s_d \sqrt{\theta}$ ;
12    pour  $l \leftarrow 1$  à  $D, l \neq d$  faire
13       $y_l^{(t+1)} = \text{signe}(y_l^{(t)}) \sqrt{(1-\theta)q_l}$ ;
14    fin
15  fin
16   $\mathbf{x}^{(t+1)} = \mathbf{E}\mathbf{y}^{(t)}$ ;
17 fin

```

B.2 LOI INVERSE GAMMA TRANSLATÉE

LA LOI Inverse Gamma tradatée, abrégée sIG, apparaît comme loi conjuguée sur les facteurs d'échelles dans le modèle présenté partie 4.2. Cette loi a été récemment introduite dans le cadre du débruitage de signaux audio Godsill (2010) et pour caractériser la distribution de masse d'un ensemble d'étoile Zaninetti (2013).

La densité de la loi est proportionnelle à

$$p(x) \propto \left(\frac{1}{1+x}\right)^{a+1} \exp\left(-\frac{1}{1+x}b\right) \mathbb{1}_{\mathbb{R}_+}(x).$$

La constante de normalisation s'obtient après trois changements de variables successifs

$$\begin{aligned} \int_{\mathbb{R}_+} \left(\frac{1}{1+x}\right)^{a+1} \exp\left(-\frac{1}{1+x}b\right) dx &= \int_1^{+\infty} \left(\frac{1}{y}\right)^{a+1} \exp\left(-\frac{1}{y}b\right) dy && \text{On pose } y = 1+x. \\ &= \int_0^1 z^{a-1} \exp(-bz) dz && \text{On pose } z = y^{-1}. \\ &= \frac{1}{b^a} \int_0^b z^{a-1} \exp(-z') dz' && \text{On pose } z' = bz. \\ &= \frac{1}{b^a} \gamma(a, b), \end{aligned}$$

où γ désigne la fonction Gamma tronquée.

La figure B.1 affiche la densité de la loi Inverse Gamma traduite pour plusieurs valeurs de a et b .

Les deux premiers changements de variables suggèrent la procédure suivante pour échantillonner la loi Inverse Gamma traduite. On échantillonne d'abord Z suivant une loi Gamma tronquée à $[0, 1]$ de paramètres a, b puis on calcule successivement $Y = Z^{-1}$ et $X = Y - 1$. L'échantillonnage de lois Gamma tronquée est réalisée suivant la méthode développée dans Damien et Walker (2001).

Les deux premiers moments s'obtiennent en appliquant les mêmes changements de variables

$$\begin{aligned}\mathbb{E}[X] &= \int_{\mathbb{R}_+} Kx \left(\frac{1}{1+x}\right)^{a+1} \exp\left(-\frac{1}{1+x}b\right) dx \\ &= K \int_1^{+\infty} (y-1) \left(\frac{1}{y}\right)^{a+1} \exp\left(-\frac{1}{y}b\right) dy \\ &= K \int_1^{+\infty} \left(\frac{1}{y}\right)^a \exp\left(-\frac{1}{y}b\right) dy - 1 \\ &= \frac{b^a}{\gamma(a,b)} \int_0^1 z^{a-2} \exp(-bz) dz - 1 \\ &= b \frac{\gamma(a-1,b)}{\gamma(a,b)} - 1.\end{aligned}$$

Pour la variance, on s'appuie sur le théorème de Koenig

$$\begin{aligned}K \int_{\mathbb{R}_+} x^2 \left(\frac{1}{1+x}\right)^{a+1} \exp\left(-\frac{1}{1+x}b\right) dx &= K \int_1^{+\infty} (y-1)^2 \left(\frac{1}{y}\right)^{a+1} \exp\left(-\frac{1}{y}b\right) dy \\ &= K \int_1^{+\infty} \left(\frac{1}{y}\right)^{a-1} \exp\left(-\frac{1}{y}b\right) dy - 2\mathbb{E}[Y] + 1 \\ &= K \int_0^1 z^{a-3} \exp(-bz) dz - 2 \underbrace{\mathbb{E}[Y]}_{\text{pas de } -1, \text{ attention!}} + 1 \\ &= b^2 \frac{\gamma(a-2,b)}{\gamma(a,b)} - 2b \frac{\gamma(a-1,b)}{\gamma(a,b)} + 1.\end{aligned}$$

De fait,

$$\begin{aligned}\text{Var}[X] &= \left(b^2 \frac{\gamma(a-2,b)}{\gamma(a,b)} - 2b \frac{\gamma(a-1,b)}{\gamma(a,b)} + 1\right) - \left(b \frac{\gamma(a-1,b)}{\gamma(a,b)} - 1\right)^2 \\ &= \frac{b^2}{\gamma(a,b)} \left(\gamma(a-2,b) - \frac{\gamma(a-1,b)^2}{\gamma(a,b)}\right).\end{aligned}$$

B.3 LOI NON INFORMATIVE DE JEFFREYS POUR LA LOI A POSTERIORI MARGINALISÉE

CETTE PARTIE dérive la loi non informative de Jeffreys pour les facteurs d'échelles δ^2 du modèle BNP-PCA décrit dans la partie 4.2.

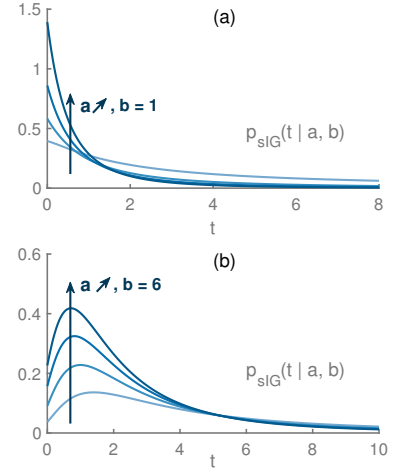


FIGURE B.1 : Densité de la loi Inverse Gamma traduite pour (a) $a = 0.25, 1, 1.5, 2$ $b = 1$, et (b) $a = 1.5, 2, 2.3, 2.5$ $b = 6$.

par linéarité.

On rappelle la vraisemblance marginalisée

$$p(\mathbf{y} | \sigma^2, \mathbf{P}, \delta^2) = \left(\frac{1}{2\pi\sigma^2} \right)^{D/2} \left(\frac{1}{1+\delta^2} \right)^{K/2} \exp \left[-\frac{1}{2\sigma^2} \left(\mathbf{y}^T \mathbf{y} - \frac{\delta^2}{1+\delta^2} \mathbf{y}^T \mathbf{P} \mathbf{P}^T \mathbf{y} \right) \right]. \quad (\text{B.12})$$

Dériver la log-vraisemblance donne

$$\frac{\partial}{\partial \delta^2} \ln p(\mathbf{y} | \delta^2, \sigma^2, \mathbf{P}) = -\frac{K}{2} \frac{1}{1+\delta^2} + \frac{1}{2\sigma^2} \frac{1}{(1+\delta^2)^2} \mathbf{y}^T \mathbf{P} \mathbf{P}^T \mathbf{y}.$$

Ensuite

$$\begin{aligned} \left(\frac{\partial}{\partial \delta^2} \ln p(\mathbf{y} | \delta^2, \sigma^2, \mathbf{P}) \right)^2 &= \underbrace{\frac{1}{4(\sigma^2)^2} \frac{1}{(1+\delta^2)^4} \left(\mathbf{y}^T \mathbf{P} \mathbf{P}^T \mathbf{y} \right)^2}_{\mathcal{A}_3} \\ &\quad - \underbrace{\frac{2K}{4\sigma^2(1+\delta^2)^3} \mathbf{y}^T \mathbf{P} \mathbf{P}^T \mathbf{y}}_{\mathcal{A}_2} + \underbrace{\frac{K^2}{4(1+\delta^2)^2}}_{\mathcal{A}_1}. \end{aligned}$$

Il ne nous reste plus qu'à prendre l'espérance sous la loi $p(\mathbf{y} | \dots)$ en exploitant la linéarité de l'espérance. Par linéarité, on peut séparer les trois termes. Le terme de droite est simple puisqu'il ne dépend pas de \mathbf{y} . Séparons le calculs des deux autres termes

Calcul de \mathcal{A}_1 :

$$\int_{\mathbb{R}^D} \frac{K^2}{4(1+\delta^2)^2} p(\mathbf{y}) \, d\mathbf{y} = \frac{K^2}{4(1+\delta^2)^2}$$

Calcul de \mathcal{A}_2 : La clé de voûte du calcul est le changement de variables. Cela consiste à d'abord diagonaliser $\mathbf{P} \mathbf{P}^T = \mathbf{A}^T \mathbf{\Lambda} \mathbf{A}$, puis à poser le changement de variable linéaire $\mathbf{z} = \mathbf{A} \mathbf{y}$. Pour la première intégrale,

$$\begin{aligned} &\int_{\mathbb{R}^D} \mathbf{y}^T \mathbf{P} \mathbf{P}^T \mathbf{y} \, p(\mathbf{y}) \, d\mathbf{y} \\ &\propto \int_{\mathbb{R}^D} \mathbf{y}^T \mathbf{P} \mathbf{P}^T \mathbf{y} \exp \left[-\frac{1}{2} \mathbf{y}^T \frac{1}{\sigma^2} \left(\mathbb{I}_D - \frac{\delta^2}{1+\delta^2} \mathbf{P} \mathbf{P}^T \right) \mathbf{y} \right] \, d\mathbf{y} \\ &\propto \int_{\mathbb{R}^D} \mathbf{z}^T \mathbf{\Lambda} \mathbf{z} \exp \left[-\frac{1}{2\sigma^2} \mathbf{z}^T \left(\mathbb{I}_D - \frac{\delta^2}{1+\delta^2} \mathbf{\Lambda} \right) \mathbf{z} \right] |\mathbf{A}| \, d\mathbf{z} \\ &\propto \sigma^2 K (1+\delta^2), \end{aligned} \quad (\text{B.13})$$

Notons que le calcul est facilité par le fait que toutes les matrices sont diagonales après le changement de variable. On peut intégrer termes à termes.

Calcul de \mathcal{A}_3 : Les mêmes techniques sont employées. Attention cependant, cette fois-ci le moment d'ordre 4 de la gaussienne intervient, qui est

donnée par $3\sigma^4$ après simplification. De fait

$$\begin{aligned}
& \int_{\mathbb{R}^D} \left(\mathbf{y}^T \mathbf{P} \mathbf{P}^T \mathbf{y} \right)^2 p(\mathbf{y}) \, d\mathbf{y} \\
& \propto \int_{\mathbb{R}^D} \left(\mathbf{y}^T \mathbf{P} \mathbf{P}^T \mathbf{y} \right)^2 \exp \left\{ -\frac{1}{2} \left(\mathbf{y}^T \frac{1}{\sigma^2} \left(\mathbb{I}_D - \frac{\delta^2}{1+\delta^2} \mathbf{P} \mathbf{P}^T \right) \mathbf{y} \right) \right\} \, d\mathbf{y} \\
& \propto \int_{\mathbb{R}^D} \left(\mathbf{z}^T \Lambda \mathbf{z} \right)^2 \exp \left\{ -\frac{1}{2\sigma^2} \left(\mathbf{z}^T \underbrace{\left(\mathbb{I}_D - \frac{\delta^2}{1+\delta^2} \Lambda \right)}_{\mathbf{Q}} \mathbf{z} \right) \right\} |\mathbf{A}| \, d\mathbf{z} \\
& \propto \int_{\mathbb{R}^D} \sum_{d=1}^D \lambda_d^2 z_d^4 \exp \left(\mathbf{z}^T \mathbf{Q} \mathbf{z} \right) \, d\mathbf{z} + 2 \int_{\mathbb{R}^D} \sum_{i=1}^{D-1} \sum_{j=i+1}^D \lambda_i \lambda_j z_i^2 z_j^2 \exp \left(\mathbf{z}^T \mathbf{Q} \mathbf{z} \right) \, d\mathbf{z}.
\end{aligned}$$

Pour faciliter la suite des calculs, on peut supposer sans pertes de généralités que les K valeurs propres non nulles (et égale à 1) de Λ sont sur les derniers éléments de la diagonale. Alors, les $D - K$ premiers éléments sont des 0. On a

$$\int_{\mathbb{R}^D} \sum_{d=1}^D z_d^4 \exp \left(\mathbf{z}^T \mathbf{Q} \mathbf{z} \right) \, d\mathbf{z} = 3\sigma^4 \left(D - K + K(1 + \delta^2) \right)$$

et

$$\begin{aligned}
2 \int_{\mathbb{R}^D} \sum_{i=1}^{D-1} \sum_{j=i+1}^D z_i^2 z_j^2 \exp \left(\mathbf{z}^T \mathbf{Q} \mathbf{z} \right) \, d\mathbf{z} &= \sum_{i=1}^{D-K-1} 0 \times i \times \sigma^4 + \sum_{i=1}^{K-1} i \times \sigma^4 (1 + \delta^2)^2 \\
&= \sum_{i=1}^{K-1} i \times \sigma^4 (1 + \delta^2)^2 \\
&= \frac{1}{2} K(K-1) \sigma^4 (1 + \delta^2)^2
\end{aligned}$$

Assemblage

$$\begin{aligned}
& \int_{\mathbb{R}^K} \left(\frac{\partial}{\partial \delta^2} \ln f(\mathbf{y} \mid \delta^2, \sigma^2, \mathbf{P}) \right)^2 \, d\mathbf{y} \\
&= \mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3 \\
&= \frac{K^2}{4(1 + \delta^2)^2} - \frac{2K}{4\sigma^2(1 + \delta^2)^3} \left(\sigma^2 + K(1 + \delta^2) \right) \\
&\quad + \frac{1}{4(\sigma^2)^2} \frac{1}{(1 + \delta^2)^4} \left(3\sigma^4 K(1 + \delta^2) + \frac{1}{2} (K-1)(K) \sigma^4 (1 + \delta^2)^2 \right) \\
&= \frac{K^2}{4(1 + \delta^2)^2} - \frac{2K^2}{4(1 + \delta^2)^2} + \frac{1}{4(1 + \delta^2)^2} (K-1)(K) + \frac{3}{4(1 + \delta^2)^2} K \\
&= \frac{K}{2(1 + \delta^2)^2}.
\end{aligned}$$

La quantité que l'on vient de calculer est l'information de Fischer. On en déduit la loi non informative de Jeffreys

$$f(\delta^2) \propto \frac{1}{1 + \delta^2}. \tag{B.14}$$

B.4 CALCUL DE LA LOI A POSTERIORI MARGINALISÉE

LA DISTRIBUTION marginale s'obtient en calculant

$$f(\boldsymbol{\theta}, \phi | \mathbf{Y}) = \int_{\mathbb{R}^{Dn}} f(\mathbf{Y} | \boldsymbol{\theta}, \mathbf{X}) f(\boldsymbol{\theta}, \mathbf{X} | \phi) f(\phi) d\mathbf{X}.$$

L'idée de la preuve est de séparer l'exponentielle en deux. Les coefficients $x_{k,n}$ correspondant à des composantes inactives, *i.e.*, telles que $z_{k,n} = 0$ s'intègrent à 1 puisque seul la loi *a priori* intervient. Il ne reste qu'une constante de la forme $\prod_{k=1}^K (2\pi\delta_k^2)^{-z_k^T \mathbf{z}_k / 2}$ où \mathbf{z}_k désigne le numéro de la ligne.

Le terme restant dans l'exponentielle est

$$-\frac{1}{2\sigma^2} \sum_{n=1}^n \left(\|\mathbf{y}_n - \sum_{\substack{k \\ z_{k,n}=1}} \mathbf{p}_k \mathbf{x}_n\|_2^2 + \sum_{\substack{k \\ z_{k,n}=1}} \frac{1}{\delta_k^2} \mathbf{x}_n^T \mathbf{x}_n \right). \quad (\text{B.15})$$

La norme ℓ_2 dans (B.15) se simplifie en $\mathbf{p}_l^T \mathbf{p}_m = \delta_{l,m}$ où $\delta_{l,m}$ est le symbole de Kronecker. De plus, la loi *a posteriori* (B.15) est conjuguée à la loi normale. Le terme restant est la constante $(2\pi\delta_k^2\sigma^2 / (1 + \delta_k^2))^{z_k^T \mathbf{z}_k / 2}$ ainsi qu'un terme proportionnel à $\mathbf{y}_n^T \mathbf{p}_k \mathbf{p}_k^T \mathbf{y}_n$ qui peut être réécrit comme $(\mathbf{p}_k^T \mathbf{y}_n)^2$.

Finalement, la loi *a posteriori* marginale (4.13) s'obtient en combinant tous les termes.

B.5 PREUVES DES RÉSULTATS D'INCONSISTANCE

LES DEUX PREUVES de cette partie concernent les résultats d'inconsistance associés à l'estimateur MAP du modèle BNP-PCA. Les deux preuves sont construites et emploient des arguments similaires à ceux employés dans Miller et Harrison (2013, 2014).

B.5.1 Résultat général

On introduit préalablement quelques notations. On appelle dans cette partie $\mathcal{A}(K, n)$ l'ensemble des matrices binaires \mathbf{Z} avec K lignes et n colonnes. Pour chaque matrice binaire \mathbf{Z} , on appelle $\mathcal{B}(\mathbf{Z})$ l'ensemble des matrices \mathbf{Z}' identiques à \mathbf{Z} mais contenant une ligne supplémentaire avec une seule composante active (un seul 1). Pour alléger l'écriture, on utilisera parfois la notation $\mathbf{Z}'(j)$ où j désigne l'indice de la colonne contenant l'élément actif. Pour finir, on note $c_n(K, \alpha)$ la quantité

$$c_n(K, \alpha) \triangleq \max_{\mathbf{Z} \in \mathcal{A}(K, n)} \max_{\mathbf{Z}' \in \mathcal{B}(\mathbf{Z})} \frac{\mathbb{P}[\mathbf{Z} | \alpha]}{\mathbb{P}[\mathbf{Z}' | \alpha]}. \quad (\text{B.16})$$

B.5.1.a Deux lemmes préliminaires

Le lemme suivant caractérise le comportement de la loi *a priori* lorsque l'on ajoute des colonnes avec peu de composantes non nulles

Lemme B.1. Pour tout α, K

$$\limsup_{n \rightarrow +\infty} \frac{1}{n} c_n(K, \alpha) \leq +\infty. \quad (\text{B.17})$$

Démonstration. Soient n, K deux entiers positifs, \mathbf{Z}, \mathbf{Z}' deux matrices binaires appartenant respectivement à $\mathcal{A}(K, n)$ et $\mathcal{B}(\mathbf{Z})$.

On rappelle que la loi d'une matrice binaire a été donnée équation (3.55). On obtient, en notant K_{new}^h le nombre de colonnes de \mathbf{Z}' identiques à celle qui a été ajoutée,

$$\frac{\mathbb{P}[\mathbf{Z}|\alpha]}{\mathbb{P}[\mathbf{Z}'|\alpha]} \leq \frac{n}{\alpha} K_{\text{new}}^h \leq \frac{K}{\alpha} n,$$

d'où le résultat. \square

Le second lemme caractérise le comportement de la vraisemblance lorsque l'on ajoute des colonnes avec peu de composantes non nulles

Lemme B.2. Soient \mathbf{Z}, \mathbf{Z}' deux matrices binaires, appartenant respectivement à $\mathcal{A}(K, n)$ et $\mathcal{B}(\mathbf{Z})$. Alors,

$$\mathbb{p}(Y_{1:N} | \mathbf{Z}) \leq \kappa \mathbb{p}(Y_{1:N} | \mathbf{Z}'), \quad (\text{B.18})$$

où

$$\kappa = b_\delta \frac{\gamma(a_\delta, b_\delta)}{\gamma(a_\delta + 1, b_\delta)}. \quad (\text{B.19})$$

Démonstration. Notons Θ l'ensemble des paramètres et hyperparamètres tels que

$$\mathbb{p}(\mathbf{Y}|\mathbf{Z}) = \int_{\Theta} \mathbb{p}(\mathbf{Y}|\theta\mathbf{Z})\mathbb{p}(\theta|\mathbf{Z})d\theta.$$

Soit \mathbf{Z}' une matrice binaire appartenant à $\mathcal{B}(\mathbf{Z})$, et j l'indice de la composante active dans la nouvelle ligne. L'activation de la j^{e} composante rajoute un terme à la vraisemblance de la forme

$$\frac{1}{1 + \delta_{K+1}^2} \exp\left(\frac{\delta_{K+1}^2}{1 + \delta_{K+1}^2} \frac{\langle \mathbf{y}_j, \mathbf{p}_{K+1} \rangle^2}{\sigma^2}\right). \quad (\text{B.20})$$

Or, le terme dans l'exponentielle est toujours positif donc l'exponentielle peut être minorée par 1. On obtient ensuite, en intégrant sur δ_{K+1}^2

$$\mathbb{p}(Y_{1:N} | \mathbf{Z}') \geq \frac{b_\delta^{a_\delta}}{\gamma(a_\delta, b_\delta)} \frac{\gamma(a_\delta + 1, b_\delta)}{b_\delta^{a_\delta + 1}} \mathbb{p}(Y_{1:N} | \mathbf{Z}),$$

d'où le résultat. \square

B.5.1.b Preuve du résultat d'inconsistance

On a, pour tout entier j de $\llbracket 1, n \rrbracket$

$$\begin{aligned} & \mathbb{p}(\mathbf{Y}, K_n = K | \alpha) \\ &= \sum_{\mathbf{Z}_K \in \mathcal{A}(K, n)} \mathbb{P}[\mathbf{Z}_K] \mathbb{p}(\mathbf{Y} | \mathbf{Z}_K, \alpha) \\ &\leq \sum_{\mathbf{Z}_K \in \mathcal{A}(K, n)} n c_n(K, \alpha) \mathbb{P}[\mathbf{Z}'(j) | \alpha] \kappa \mathbb{p}(\mathbf{Y} | \mathbf{Z}'(j), \alpha). \end{aligned} \quad (\text{B.21})$$

où la dernière inégalité s'obtient en appliquant les lemmes B.1 et B.2. Comme l'inégalité est vraie pour tout entier j de $\llbracket 1, n \rrbracket$, on peut moyenner les résultats sur les n éléments, d'où

$$\begin{aligned}
& \mathbb{p}(\mathbf{Y}, K_n = K | \alpha) \\
& \leq \sum_{\mathbf{Z}_K \in \mathcal{A}(K, n)} \sum_{j=1}^n \kappa c_n(K, \alpha) \mathbb{P}[\mathbf{Z}'(j) | \alpha] \mathbb{p}(\mathbf{Y} | \mathbf{Z}'(j), \alpha) \\
& \leq \kappa c_n(K, \alpha) \sum_{\mathbf{Z}_K \in \mathcal{A}(K, n)} \sum_{\mathbf{Z}' \in \mathcal{A}(K+1, \alpha)} \mathbb{p}(\mathbf{Y} | \mathbf{Z}'(j) | \alpha) \\
& \mathbb{1}_{\mathcal{B}(\mathbf{Z})}(\mathbf{Z}') \\
& \leq \kappa c_n(K, \alpha) \sum_{\mathbf{Z}' \in \mathcal{A}(K+1, \alpha)} \text{card}\{\mathbf{Z}, \mathbf{Z}' \in \mathcal{B}(\mathbf{Z})\} \mathbb{p}(\mathbf{Y} | \mathbf{Z}'(j) | \alpha) \mathbb{1}_{\mathcal{B}(\mathbf{Z})}(\mathbf{Z}').
\end{aligned}$$

Cependant, pour chaque matrice \mathbf{Z}' de $\mathcal{A}(K+1, \alpha)$, il y a au plus une matrice \mathbf{Z} vérifiant la conditions, donc

$$\mathbb{p}(\mathbf{Y}, K_n = K | \alpha) \leq \kappa c_n(K, \alpha) \sum_{\mathbf{Z}' \in \mathcal{A}(K+1, \alpha)} \mathbb{p}(\mathbf{Y} | \mathbf{Z}'(j) | \alpha) \mathbb{1}_{\mathcal{B}(\mathbf{Z})}(\mathbf{Z}'). \tag{B.22}$$

La preuve se termine en appliquant une simple règle de Bayes. Pour $K < D$

$$\begin{aligned}
\mathbb{p}(K_n = K | \mathbf{Y}, \alpha) &= \frac{\mathbb{p}(K_n = K, \mathbf{Y} | \alpha)}{\sum_{k=0}^{\infty} \mathbb{p}(K_n = k, \mathbf{Y}, \alpha)} \\
&< \frac{\mathbb{p}(K_n = K, \mathbf{Y} | \alpha)}{\mathbb{p}(K_n = K, \mathbf{Y}, \alpha) + \mathbb{p}(K_n = K+1 | \mathbf{Y}, \alpha)} \\
&< \frac{c_n(K, \alpha) \kappa}{c_n(K, \alpha) \kappa + 1} \\
&< 1.
\end{aligned}$$

De même, pour $K = D$

$$\begin{aligned}
\mathbb{p}(K_n = D | \mathbf{Y}, \alpha) &= \frac{\mathbb{p}(K_n = D, \mathbf{Y} | \alpha)}{\sum_{k=0}^{\infty} \mathbb{p}(K_n = k | \mathbf{Y}, \alpha)} \\
&\geq \frac{\mathbb{p}(K_n = D, \mathbf{Y} | \alpha)}{\sum_{k=0}^D (c_n(k, \alpha) \kappa)^{K-k} \mathbb{p}(K_n = k | \mathbf{Y}, \alpha)} \\
&\geq \frac{1}{\sum_{k=0}^D (c_n(k, \alpha) \kappa)^{D-k}} \\
&\geq \frac{1}{1 + \sum_{k=1}^D (c_n(k, \alpha) \kappa)^k} \\
&> 0.
\end{aligned}$$

Notons pour conclure que la preuve peut se généraliser facilement à une plus large gamme de modèle. En effet, la preuve complète ne repose que sur les deux lemmes. Le lemme B.1 est un résultat propre à l'IBP. Le lemme B.2 est modèle dépendant, mais assez simple à vérifier tant que les lois *a priori* sont régulières. Toutefois, le dernier résultat équation (4.28) est propre au modèle BNP-PCA et découle des contraintes d'orthogonalité.

B.5.2 Inconsistance sévère

On suppose maintenant que pour tout n , $\mathbf{y}_n \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_D)$. Considérons la quantité g

$$g(\mathbf{Y}, \mathbf{Z}, \mathbf{P}, \delta^2) = \frac{\mathcal{K}(a_{\delta^2}, b_{\delta^2})^K}{\text{vol}(\mathcal{S}_D)} \prod_{k=1}^K \left(\frac{1}{1 + \delta_k^2} \right)^{a_{\delta^2} + \mathbf{z}_k^T \mathbf{z}_k} \exp \left[-\frac{1}{1 + \delta_k^2} \left(b_{\delta^2} + \frac{1}{2\sigma^2} \sum_{n=1}^n z_{k,n} \langle \mathbf{p}_k, \mathbf{y}_n \rangle^2 \right) \right], \quad (\text{B.23})$$

i.e, $g \propto \mathbf{p}(\mathbf{Z}, \mathbf{P}, \delta^2 | \mathbf{Y}, \sigma^2, \alpha)$.

On rappelle qu'ici, K_n est la variable aléatoire associée à la dimension du sous-espace latent. Par définition,

$$\begin{aligned} \mathbb{P}[K_n = 0 | \mathbf{Y}, \sigma^2, \alpha] &= \frac{\mathbb{P}(K_n = 0, \mathbf{Y} | \sigma^2, \alpha)}{\sum_{K=1}^{+\infty} \mathbb{P}(K_n = K, \mathbf{Y} | \sigma^2, \alpha)} \\ &\leq \frac{1}{1 + \frac{\mathbb{P}(K_n = 1, \mathbf{Y} | \sigma^2, \alpha)}{\mathbb{P}(K_n = 0, \mathbf{Y} | \sigma^2, \alpha)}}. \end{aligned} \quad (\text{B.24})$$

Considérons la quantité qui apparaît au dénominateur à l'équation (B.24). Cette quantité peut se réécrire

$$\begin{aligned} \frac{\mathbb{P}(K_n = 1, \mathbf{Y} | \sigma^2, \alpha)}{\mathbb{P}(K_n = 0, \mathbf{Y} | \sigma^2, \alpha)} &= \sum_{\mathbf{Z}, K_n=1} \int_{\mathcal{S}_D} \int_{\mathbb{R}_+} g(\mathbf{Y}, \mathbf{Z}, \delta^2, \mathbf{P}) d\sigma^2 d\mathbf{P} d\delta^2 \frac{\mathbb{P}[\mathbf{Z} | \alpha]}{\mathbb{P}[\mathbf{0} | \alpha]} \\ &= \sum_{\mathbf{Z}, K_n=1} \int_{\mathcal{S}_D} \int_{\mathbb{R}_+} g(\mathbf{Y}, \mathbf{Z}, \delta^2, \mathbf{P}) d\sigma^2 d\mathbf{P} d\delta^2 \times \alpha \frac{(n - \mathbf{z}_1^T \mathbf{z}_1)! (\mathbf{z}_1^T \mathbf{z}_1 - 1)!}{N!}. \end{aligned}$$

Or, par construction, la matrice \mathbf{Z} intervenant à la dernière ligne ne contient qu'une seule ligne. On peut donc décomposer l'équation en une somme sur le nombre de composantes actives suivant

$$\frac{\mathbb{P}(K_n = 1, \mathbf{Y} | \sigma^2, \alpha)}{\mathbb{P}(K_n = 0, \mathbf{Y} | \sigma^2, \alpha)} = \sum_{l=1}^n \sum_{\mathbf{Z}, K_n=1, \mathbf{z}_1^T \mathbf{z}_1=l} \frac{\alpha}{l} \frac{1}{\binom{n}{l}} \int_{\mathcal{S}_D} \int_{\mathbb{R}_+} g(\mathbf{Y}, \mathbf{Z}, \delta^2, \mathbf{P}) d\sigma^2 d\mathbf{P} d\delta^2.$$

On définit, pour tout entier $\ell \leq n$, la U-statistique

$$U_\ell(\mathbf{Y}) \triangleq \frac{1}{\binom{n}{\ell}} \sum_{\substack{\mathbf{Z}, K_n=1, \\ \mathbf{z}_1^T \mathbf{z}_1=\ell}} \int_{\mathcal{S}_D \cup \mathbb{R}_+} g(\mathbf{Y}, \mathbf{Z}, \delta^2, \mathbf{P}) d\sigma^2 d\mathbf{P} d\delta^2, \quad (\text{B.25})$$

où le support des permutations dans l'équation (B.25) est donné par les indices des composantes non nulles de \mathbf{Z} . La loi forte des grandes nombres pour les U-statistiques (Hoeffding, 1961) nous permet d'affirmer que pour tout ℓ dans $[[1, n]]$

$$U_\ell(\mathbf{Y}) \xrightarrow[n \rightarrow +\infty]{p.s.} \mathbb{E}_{\mathbf{Y}} \left[\int_{\mathcal{S}_D \cup \mathbb{R}_+} g(\mathbf{Y}, \mathbf{Z}, \delta^2, \mathbf{P}) d\sigma^2 d\mathbf{P} d\delta^2 \right] = 1. \quad (\text{B.26})$$

Notons que la dernière égalité provient du fait que la quantité g est une loi de probabilité. En conséquence, pour tout $L \leq n$

$$\frac{\mathbb{P}(K_n = 1, \mathbf{Y} | \sigma^2, \alpha)}{\mathbb{P}(K_n = 0, \mathbf{Y} | \sigma^2, \alpha)} \geq \sum_{\ell=1}^L \frac{\alpha}{\ell} U_\ell(\mathbf{Y}) \xrightarrow[n \rightarrow +\infty]{a.s.} \sum_{\ell=1}^L \frac{\alpha}{\ell}. \quad (\text{B.27})$$

Sachant que l'équation (B.27) est vraie pour tout L , et que la série harmonique $\sum_{\ell} \frac{1}{\ell}$ diverge, alors la quantité $\frac{\mathbb{P}(K_n=1, \mathbf{Y}|\sigma^2, \alpha)}{\mathbb{P}(K_n=0, \mathbf{Y}|\sigma^2, \alpha)}$ diverge presque sûrement lorsque n tend vers l'infini.

B.6 LOI DES PRODUITS SCALAIRES

CETTE APPENDICE décrit le calcul des distributions marginales des produits scalaires sous la loi uniforme apparaissant au théorème 4.3. L'idée générale de la preuve consiste à adapter le vecteur que l'on va projeter à l'aire élémentaire en coordonnées sphériques.

Soit \mathbf{u} un vecteur unitaire de \mathbb{R}^L . On rappelle qu'il faut voir L ici comme la dimension de l'orthogonal, i.e. $L = D - K$. Considérons une matrice aléatoire distribuée uniformément sur le groupe orthogonal \mathcal{O}_L . On a déjà vu que dans ce cas, la loi marginale d'une colonne, notée $\boldsymbol{\nu}$, est distribuée uniformément sur la variété de Stiefel \mathcal{S}_L^1 . Notons w la variable aléatoire associée au produit scalaire $w = |\langle \mathbf{u}, \boldsymbol{\nu} \rangle| = |\boldsymbol{\nu}^T \mathbf{u}|$. On obtient la densité de w via

$$p_w(w \leq \lambda) = p_{\boldsymbol{\nu}}(|\boldsymbol{\nu}^T \mathbf{u}| \leq \lambda) = \int \mathbf{1}_{|\boldsymbol{\nu}^T \mathbf{u}| \leq \lambda}(\boldsymbol{\nu}) d\boldsymbol{\nu}, \quad (\text{B.28})$$

où l'intégrale dans la dernière équation s'écrit par rapport à la mesure de Haar sur la sphère.

C'est ici que l'on va exploiter les propriétés d'invariance de la loi uniforme sur la sphère par rapport aux rotations. Soit \mathbf{R} une matrice de rotation telle que $\mathbf{e} = \mathbf{R}\mathbf{u}$ où $\mathbf{e} = [1, 0, 0, \dots]$. Par invariance de la mesure de Haar, l'équation (B.28) se réécrit en coordonnée sphérique (voir note dans la marge)

$$p(w \leq \lambda) = \frac{1}{\mathcal{S}_{L-1}} \int \mathbf{1}_{|\cos(\phi_1)| \leq \lambda}(\mathbf{v}) d\mathcal{S}^{L-1}.$$

Comme $|\cos(\phi_1)| \leq \lambda$ si ϕ_1 appartient à l'ensemble $[\arccos(\lambda), \pi - \arccos(\lambda)]$, on a, par symétrie par rapport à $\pi/2$

$$\begin{aligned} p(w \leq \lambda) &= \frac{2}{\mathcal{S}_{L-1}} \int_{\phi_1=\arccos(\lambda)}^{\pi/2} \int_{\phi_2 \dots \phi_{L-2}=0}^{\pi} \int_{\phi_{L-1}=0}^{2\pi} \\ &\quad \sin^{L-2}(\phi_1) \dots \sin(\phi_{L-2}) d\phi_1 \dots d\phi_{L-1} \\ &= 2 \frac{\mathcal{S}_{L-2}}{\mathcal{S}_{L-1}} \int_{\phi_1=\arccos(\lambda)}^{\pi/2} \sin^{L-2}(\phi_1) d\phi_1, \end{aligned}$$

en reconnaissant l'aire élément de la sphère en dimension $L - 2$. On a, en considérant le changement de variable $y = \cos(\phi_1)$,

$$p(w \leq \lambda) = \frac{\mathcal{S}_{L-2}}{\mathcal{S}_{L-1}} 2 \int_0^\lambda \sin^{L-3}(\arccos(y)) dy.$$

La relation entre les coordonnées cartésiennes et sphériques est

$$\begin{aligned} v_1 &= \cos(\phi_1) \\ v_2 &= \sin(\phi_1) \cos(\phi_2) \\ &\vdots \end{aligned}$$

$$\begin{aligned} v_{D-1} &= \sin(\phi_1) \dots \sin(\phi_{D-2}) \cos(\phi_{D-1}) \\ v_D &= \sin(\phi_1) \dots \sin(\phi_{D-1}). \end{aligned}$$

L'aire élémentaire de la D -sphère en coordonnées sphérique, paramétrée par les $D - 1$ angles, devient

$$\begin{aligned} d\mathcal{S}^D &= \sin^{D-2}(\phi_1) \sin^{D-3}(\phi_2) \dots \\ &\quad \dots \sin(\phi_{D-2}) d\phi_1 \dots d\phi_{D-1}. \end{aligned}$$

Sachant que $\sin(\arccos(y))$ s'écrit aussi $\sqrt{1-y^2}$, on obtient, après deux changements de variables

$$\begin{aligned} \int_0^\lambda \sin^{L-3}(\arccos(y)) dy &= \int_0^\lambda (1-y^2)^{L-3} dy \\ &= \lambda \int_0^1 (1-\lambda^2 y^2)^{L-3} dy \\ &= \frac{\lambda}{2} \int_0^1 (1-\lambda^2 z)^{L-3} z^{-1/2} dz. \end{aligned}$$

On obtient une expression explicite de l'intégrale en utilisant le Corollaire 1.6.3.2 page 36 dans Gupta et Nagar (1999) avec les paramètres $\alpha = \frac{1}{2}$, $\beta = -\frac{L-3}{2}$, $\gamma = \frac{3}{2}$ et $R = \lambda^2$. D'où

$$\int_0^\lambda \sin^{L-3}(\arccos(y)) dy = 2\lambda {}_2F_1\left(\frac{1}{2}, -\frac{L-3}{2}; \frac{3}{2}; \lambda^2\right),$$

ce qui est le résultat attendu.

Bibliographie

- Akaike, H. 1974. « A new look at the statistical model identification ». *IEEE Transactions on Automatic Control* 19, n° 6 (décembre) : 716–723. doi : [10.1109/TAC.1974.1100705](https://doi.org/10.1109/TAC.1974.1100705). (Cf. page 109).
- Alexeev, Boris, Jameson Cahill et Dustin G. Mixon. 2012. « Full Spark Frames ». *Journal of Fourier Analysis and Applications* 18 (6) : 1167–1194. doi : [10.1007/s00041-012-9235-4](https://doi.org/10.1007/s00041-012-9235-4). (Cf. page 36).
- Antoniak, Charles E. 1974. « Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems ». *The Annals of Statistics* 2, n° 6 (novembre) : 1152–1174. doi : [10.1214/aos/1176342871](https://doi.org/10.1214/aos/1176342871). (Cf. page 92).
- Applebaum, D. 2004. *Lévy Processes and Stochastic Calculus*. Cambridge Studies in Advanced Mathematics. Cambridge University Press. <https://books.google.fr/books?id=q7eDUjdJxIkC>. (Cf. page 85).
- Archanbeau, Cédric, Nicolas Delannay et Michel Verleysen. 2008. « Mixtures of robust probabilistic principal component analyzers ». *Neurocomputing* 71 (7-9) : 1274–1282. doi : [10.1016/j.neucom.2007.11.029](https://doi.org/10.1016/j.neucom.2007.11.029). (Cf. page 109).
- Babacan, S. D., R. Molina et A. K. Katsaggelos. 2010. « Bayesian Compressive Sensing Using Laplace Priors ». *IEEE Trans. Image Process.* 19, n° 1 (janvier) : 53–63. doi : [10.1109/TIP.2009.2032894](https://doi.org/10.1109/TIP.2009.2032894). (Cf. page 30).
- Balu, R., T. Furon et H. Jégou. 2014. « Beyond "project and sign" for cosine estimation with binary codes ». Dans *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, 68884–68888. Mai. doi : [10.1109/ICASSP.2014.6854934](https://doi.org/10.1109/ICASSP.2014.6854934). (Cf. page 22).
- Baraniuk, Richard G. 2007. « Compressive Sensing ». *IEEE Signal Processing Magazine* 24, n° 4 (juillet) : 118–121. doi : [10.1109/MSP.2007.4286571](https://doi.org/10.1109/MSP.2007.4286571). (Cf. page 23).
- Bauschke, H.H., et Patrick Louis Combettes. 2011. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. 468. Springer. doi : [10.1007/978-1-4419-9467-7](https://doi.org/10.1007/978-1-4419-9467-7). (Cf. pages 39, 151).
- Beck, Amir, et Marc Teboulle. 2009. « A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems ». *SIAM Journal on Imaging Sciences* (Philadelphia, PA, USA) 2, n° 1 (mars) : 183–202. doi : [10.1137/080716542](https://doi.org/10.1137/080716542). (Cf. page 29).
- Besson, Olivier, Nicolas Dobigeon et Jean-Yves Tournier. 2011. « Minimum mean square distance estimation of a subspace ». *IEEE Transactions on Signal Processing* 59, n° 12 (décembre) : 5709–5720. doi : [10.1109/TSP.2011.2166548](https://doi.org/10.1109/TSP.2011.2166548). (Cf. pages 112, 125).
- . 2012. « CS decomposition based Bayesian subspace estimation ». *IEEE Transactions on Signal Processing* 60, n° 8 (août) : 4210–4218. doi : [10.1109/TSP.2012.2197619](https://doi.org/10.1109/TSP.2012.2197619). (Cf. page 125).
- Bioucas-Dias, J. M., A. Plaza, Nicolas Dobigeon, M. Parente, Q. Du, P. Gader et J. Chanussot. 2012. « Hyperspectral Unmixing Overview : Geometrical, Statistical, and Sparse Regression-Based Approaches ». *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5, n° 2 (avril) : 354–379. doi : [10.1109/JSTARS.2012.2194696](https://doi.org/10.1109/JSTARS.2012.2194696). (Cf. page 135).

- Bioucas-Dias, J.M., et J.M.P. Nascimento. 2008. « Hyperspectral Subspace Identification ». *Geoscience and Remote Sensing, IEEE Transactions on* 46, n° 8 (août) : 2435–2445. doi :10 . 1109 /TGRS . 2008 . 918089. (Cf. pages 135, 136).
- Bishop, Christopher M. 1999a. « Bayesian PCA ». Dans *Advances in Neural Information Processing Systems 11*, sous la direction de M. J. Kearns, S. A. Solla et D. A. Cohn, 382–388. MIT Press. <http://papers.nips.cc/paper/1549-bayesian-pca.pdf>. (Cf. pages 109, 110).
- . 1999b. « Variational Principal Components ». Dans *Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'99*, 1 : 509–514. IEE, janvier. <http://research.microsoft.com/apps/pubs/default.aspx?id=67241>. (Cf. page 110).
- . 2006. *Pattern Recognition and Machine Learning*. Springer. <http://research.microsoft.com/en-us/um/people/cmbishop/prml/>. (Cf. pages 115, 165).
- Blackwell, David, et James B. MacQueen. 1973. « Ferguson Distributions Via Polya Urn Schemes ». *The Annals of Statistics* 1, n° 2 (mars) : 353–355. doi :10 . 1214/aos/1176342372. (Cf. page 89).
- Blanchet, G., et L. Moisan. 2012. « An explicit sharpness index related to global phase coherence ». Dans *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1065–1068. Mars. doi :10 . 1109/ICASSP.2012.6288070. (Cf. page 136).
- Blumensath, Thomas, et Mike E. Davies. 2009. « Iterative hard thresholding for compressed sensing ». *Applied and Computational Harmonic Analysis* 27 (3) : 265–274. doi :10 . 1016/j . acha . 2009 . 04 . 002. (Cf. page 28).
- Bolton, Richard J., David J. Hand et Andrew R. Webb. 2003. « Projection techniques for nonlinear principal component analysis ». *Statistics and Computing* 13 (3) : 267–276. doi :10 . 1023/A : 1024274801715. (Cf. page 109).
- Broderick, Tamara, Michael I. Jordan et Jim Pitman. 2013. « Cluster and Feature Modeling from Combinatorial Stochastic Processes ». *Statistical Science* 28, n° 3 (août) : 289–312. doi :10 . 1214 / 13 - STS434. (Cf. page 103).
- Broderick, Tamara, Brian Kulis et Michael I. Jordan. 2012. « MAD-Bayes : MAP-based asymptotic derivations from Bayes ». *arXiv preprint*. (Cf. page 138).
- Cadzow, J. 1971. « Algorithm for the minimum-effort problem ». *IEEE Trans. Autom. Contr.* 16, n° 1 (février) : 60–63. doi :10 . 1109/TAC . 1971 . 1099634. (Cf. pages 17, 19, 33, 37).
- Calderbank, A.R., et I. Daubechies. 2002. « The pros and cons of democracy ». *IEEE Transactions on Information Theory* 48, n° 6 (juin) : 1721–1725. doi :10 . 1109/TIT . 2002 . 1003852. (Cf. page 21).
- Cam, Lucien Le, et Grace Lo Yang. 2000. *Asymptotics in Statistics*. Springer New York. doi :10 . 1007 /978 - 1 - 4612 - 1166 - 2. (Cf. page 94).
- Candès, Emmanuel Jean. 2008. « The restricted isometry property and its implications for compressed sensing ». *Comptes Rendus Mathématique* 753 (9) : 483–598. doi :10 . 1016/j . crma . 2008 . 03 . 014. (Cf. page 27).
- Candès, Emmanuel Jean, Justin Keith Romberg et Terence Tao. 2006a. « Robust uncertainty principles : exact signal reconstruction from highly incomplete frequency information ». *IEEE Transactions on Information Theory* 52, n° 2 (février) : 489–509. doi :10 . 1109/TIT . 2005 . 862083. (Cf. page 23).
- . 2006b. « Stable signal recovery from incomplete and inaccurate measurements ». *Communications on Pure and Applied Mathematics* 59 (8) : 1207–1223. doi :10 . 1002/cpa . 20124. (Cf. page 23).

- Candès, Emmanuel Jean, et Terence Tao. 2005. « Decoding by linear programming ». *IEEE Transactions on Information Theory* 51, n° 12 (décembre) : 4203–4215. doi : [10.1109/TIT.2005.858979](https://doi.org/10.1109/TIT.2005.858979). (Cf. page 26).
- Caron, Francois. 2012. « Bayesian nonparametric models for bipartite graphs ». Dans *Advances in Neural Information Processing Systems 25*, sous la direction de F. Pereira, C. J. C. Burges, L. Bottou et K. Q. Weinberger, 2051–2059. Curran Associates, Inc. <http://papers.nips.cc/paper/4837-bayesian-nonparametric-models-for-bipartite-graphs.pdf>. (Cf. pages 105, 140).
- Chafaï, Djalil, et Florent Malrieu. 2015. *Recueil de modèles aléatoires*. XVI+410p. Novembre. <https://hal.archives-ouvertes.fr/hal-01243589>. (Cf. page 91).
- Champagnat, F., Y. Goussard et J. Idier. 1996. « Unsupervised deconvolution of sparse spike trains using stochastic approximation ». *IEEE Transactions on Signal Processing* 44, n° 12 (décembre) : 2988–2998. doi : [10.1109/78.553473](https://doi.org/10.1109/78.553473). (Cf. page 30).
- Chang, Chein-I, et Qian Du. 2004. « Estimation of number of spectrally distinct signal sources in hyperspectral imagery ». *IEEE Transactions on Geoscience and Remote Sensing* 42, n° 3 (mars) : 608–619. doi : [10.1109/TGRS.2003.819189](https://doi.org/10.1109/TGRS.2003.819189). (Cf. page 135).
- Chen, Mengjie, Chao Gao et Hongyu Zhao. 2016. « Posterior Contraction Rates of the Phylogenetic Indian Buffet Processes ». *Bayesian Anal.* 11, n° 2 (juin) : 477–497. (Cf. page 120).
- Chopin, Nicolas. 2010. « Fast simulation of truncated Gaussian distributions ». *Statistics and Computing* 21, n° 2 (janvier) : 275–288. doi : [10.1007/s11222-009-9168-1](https://doi.org/10.1007/s11222-009-9168-1). (Cf. page 58).
- Cohen, Albert, Wolfgang Dahmen et Ronald DeVore. 2008. « Compressed sensing and best k -term approximation ». *Journal of the American Mathematical Society* 22, n° 1 (juillet) : 211–231. doi : [10.1090/s0894-0347-08-00610-3](https://doi.org/10.1090/s0894-0347-08-00610-3). (Cf. page 27).
- Condat, Laurent. 2015. « Fast Projection onto the Simplex and the ℓ_1 Ball ». *Mathematical Programming* : 1–11. doi : [10.1007/s10107-015-0946-6](https://doi.org/10.1007/s10107-015-0946-6). (Cf. pages 50, 54, 60).
- Cormode, G., et S. Muthukrishnan. 2006. « Combinatorial Algorithms for Compressed Sensing ». Dans *2006 40th Annual Conference on Information Sciences and Systems*, 198–201. Mars. doi : [10.1109/CISS.2006.286461](https://doi.org/10.1109/CISS.2006.286461). (Cf. page 28).
- Cvetkovic, Z. 2003. « Resilience properties of redundant expansions under additive noise and quantization ». *IEEE Transactions on Information Theory* 49, n° 3 (mars) : 644–656. doi : [10.1109/TIT.2002.808107](https://doi.org/10.1109/TIT.2002.808107). (Cf. page 19).
- D. A. van Dyk et Taeyoung Park. 2008. « Partially Collapsed Gibbs Samplers : Theory and Methods ». *J. Amer. Stat. Assoc.* 103, n° 482 (juin) : 790–796. (Cf. page 114).
- Damien, Paul, et Stephen G. Walker. 2001. « Sampling truncated normal, beta, and gamma densities ». *Journal of Computational and Graphical Statistics* 10, n° 2 (décembre) : 206–215. (Cf. page 167).
- Dang, Hong-Phuong, et Pierre Chainais. 2017. « Indian Buffet Process dictionary learning : Algorithms and applications to image processing ». *International Journal of Approximate Reasoning* 83 : 1–20. doi : [10.1016/j.ijar.2016.12.010](https://doi.org/10.1016/j.ijar.2016.12.010). (Cf. page 120).
- Davenport, Mark A. 2010. « Random observations on random observations : Sparse signal acquisition and processing ». Thèse de doct., Rice University. (Cf. page 26).
- Davenport, Mark A., Marco F. Duarte, Yonina C. Eldar et Gitta Kutyniok. 2012. « Introduction to compressed sensing ». Dans *Compressed Sensing : Theory and Applications*, sous la direction d'Yonina C. Eldar et Gitta Editors Kutyniok, 1–64. Cambridge University Press. doi : [10.1017/CBO9780511794308.002](https://doi.org/10.1017/CBO9780511794308.002). (Cf. page 27).

- de Finetti, Bruno. 1931. « Funzione caratteristica di un fenomeno aleatorio ». *Atti della R. Accademia Nazionale dei Lincei, Ser. 6. Memorie, Classe di Scienze Fisiche, Matematiche e Naturali* 4 : 251–299. (Cf. page 89).
- Deo, A. S., et I. D. Walker. 1997. « Minimum effort inverse kinematics for redundant manipulators ». *IEEE Transactions on Robotics and Automation* 13, n° 5 (octobre) : 767–775. doi : [10.1109/70.631238](https://doi.org/10.1109/70.631238). (Cf. page 19).
- DeVore, Ronald A. 2007. « Deterministic constructions of compressed sensing matrices ». *Journal of Complexity* 23 (4) : 918–925. doi : [10.1016/j.jco.2007.04.002](https://doi.org/10.1016/j.jco.2007.04.002). (Cf. page 27).
- Diaconis, Persi, Susan Holmes et Mehrdad Shahshahani. 2013. « Sampling from a Manifold ». Dans *Advances in Modern Statistical Theory and Applications : A Festschrift in honor of Morris L. Eaton*, t. Volume 10, 102–125. Institute of Mathematical Statistics. doi : [10.1214/12-IMSCOLL1006](https://doi.org/10.1214/12-IMSCOLL1006). (Cf. page 163).
- Dobigeon, Nicolas, et Jean-Yves Tournet. 2010. « Bayesian Orthogonal Component Analysis for Sparse Representation ». *IEEE Transactions on Signal Processing* 58, n° 5 (mai) : 2675–2685. doi : [10.1109/TSP.2010.2041594](https://doi.org/10.1109/TSP.2010.2041594). (Cf. page 109).
- Dobigeon, Nicolas, Jean-Yves Tournet et C. I. Chang. 2008. « Semi-Supervised Linear Spectral Unmixing Using a Hierarchical Bayesian Model for Hyperspectral Imagery ». *IEEE Transactions on Signal Processing* 56, n° 7 (juillet) : 2684–2695. doi : [10.1109/TSP.2008.917851](https://doi.org/10.1109/TSP.2008.917851). (Cf. pages 111, 140).
- Donoho, David L., et Michael Elad. 2003. « Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization ». *Proceedings of the National Academy of Sciences* 100 (5) : 2197–2202. doi : [10.1073/pnas.0437847100](https://doi.org/10.1073/pnas.0437847100). (Cf. page 25).
- Donoho, David L., Y. Tsaig, I. Drori et J. L. Starck. 2012. « Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit ». *IEEE Transactions on Information Theory* 58, n° 2 (février) : 1094–1121. doi : [10.1109/TIT.2011.2173241](https://doi.org/10.1109/TIT.2011.2173241). (Cf. page 28).
- Doshi, Finale, Kurt Miller, Jurgen Van Gael et Yee Whye Teh. 2009. « Variational Inference for the Indian Buffet Process ». Dans *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009*, 137–144. <http://www.jmlr.org/proceedings/papers/v5/doshi09a.html>. (Cf. page 138).
- Durmus, Alain, Eric Moulines et Marcelo Pereyra. 2016. « Efficient Bayesian computation by proximal Markov chain Monte Carlo : when Langevin meets Moreau ». Décembre. eprint : <https://arxiv.org/pdf/1612.07471.pdf>. (Cf. pages 60, 61).
- Eldar, Yonina C. 2009. « Generalized SURE for Exponential Families : Applications to Regularization ». *IEEE Transactions on Signal Processing* 57, n° 2 (février) : 471–481. doi : [10.1109/TSP.2008.2008212](https://doi.org/10.1109/TSP.2008.2008212). (Cf. page 29).
- Elvira, Clément, Pierre Chainais et Nicolas Dobigeon. 2016. « Democratic prior for anti-sparse coding ». Dans *Proc. IEEE Statistical Signal Processing Workshop (SSP)*, 1–4. Juin. doi : [10.1109/SSP.2016.7551813](https://doi.org/10.1109/SSP.2016.7551813). (Cf. pages 12, 44, 67, 74, 144).
- . 2017a. « Bayesian Antisparse Coding ». *IEEE Transactions on Signal Processing* 65, n° 7 (avril) : 1660–1672. doi : [10.1109/TSP.2016.2645543](https://doi.org/10.1109/TSP.2016.2645543). (Cf. pages 12, 44, 67, 74, 144).
- . 2017b. « Bayesian nonparametric principal component analysis ». *soumis*. (Cf. pages 12, 107, 147).
- . 2017c. « Bayesian nonparametric subspace estimation ». Dans *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, 2247–2251. Mars. doi : [10.1109/ICASSP.2017.7952556](https://doi.org/10.1109/ICASSP.2017.7952556). (Cf. pages 12, 107, 147).

- . 2017d. « Une formulation bayésienne du codage antiparcimonieux ». Dans *Actes du Colloque GRETSI*. Juan Les Pin, France, septembre. (Cf. pages 12, 44, 74, 144).
- Ewens, Warren J. 2004. *Mathematical Population Genetics*. Springer New York. doi :10.1007/978-0-387-21822-9. (Cf. page 90).
- Farrell, Brendan, et Peter Jung. 2009. « A Kashin Approach to the Capacity of the Discrete Amplitude Constrained Gaussian Channel ». Dans *Proc. Int. Conf. Sampling Theory and Applications (SAMPTA)*. Marseille, France, mai. (Cf. page 22).
- Feller, William. 1968. *An Introduction to Probability Theory and Its Applications*. T. 2. Wiley, janvier. <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20%7B%5C&%7Dpath=ASIN/0471257087>. (Cf. page 85).
- Ferguson, Thomas S. 1973. « A Bayesian Analysis of Some Nonparametric Problems ». *The Annals of Statistics* 1, n° 2 (mars) : 209–230. doi :10.1214/aos/1176342360. (Cf. pages 9, 85).
- Freedman, David A. 1965. « On the Asymptotic Behavior of Bayes Estimates in the Discrete Case II ». *Ann. Math. Statist.* 36, n° 2 (avril) : 454–456. doi :10.1214/aoms/1177700155. (Cf. page 94).
- Fuchs, J.-J. 2011. « Spread representations ». Dans *Proc. IEEE Asilomar Conf. Signals, Systems, Computers*, 814–817. Novembre. doi :10.1109/ACSSC.2011.6190120. (Cf. pages 33, 38, 39, 151, 152).
- Ghahramani, Zoubin, et Thomas L. Griffiths. 2006. « Infinite latent feature models and the Indian buffet process ». Dans *Advances in Neural Information Processing Systems*, sous la direction d'Y. Weiss, P. B. Schölkopf et J. C. Platt, 475–482. MIT Press. <http://papers.nips.cc/paper/2882-infinite-latent-feature-models-and-the-indian-buffet-process.pdf>. (Cf. pages 97, 102).
- Ghosal, S., J. K. Ghosh et R. V. Ramamoorthi. 1999. « Posterior consistency of Dirichlet mixtures in density estimation ». *Ann. Statist.* 27, n° 1 (mars) : 143–158. doi :10.1214/aos/1018031105. (Cf. page 94).
- Ghosal, Subhashis. 2009. « The Dirichlet process, related priors and posterior asymptotics ». Dans *Bayesian Non-parametrics* : sous la direction de Nils Lid Hjort, Chris Holmes, Peter Muller et Stephen G. Walker, 35–79. Cambridge : Cambridge University Press, novembre. doi :10.1017/CBO9780511802478.003. (Cf. page 94).
- Godsill, Simon. 2010. « The Shifted Inverse-gamma Model for Noise-floor Estimation in Archived Audio Recordings ». *Signal Process.* (Amsterdam, The Netherlands, The Netherlands) 90, n° 4 (avril) : 991–999. doi :10.1016/j.sigpro.2009.04.039. (Cf. pages 114, 166).
- Green, P. J. 1995. « Reversible jump Markov Chain Monte Carlo Methods computation and Bayesian model determination ». *Biometrika* 82, n° 4 (décembre) : 711–732. (Cf. pages 110, 111).
- Green, Robert O, Michael L Eastwood, Charles M Sarture, Thomas G Chrien, Mikael Aronsson, Bruce J Chippendale, Jessica A Faust et al. 1998. « Imaging Spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) ». *Remote Sensing of Environment* 65 (3) : 227–248. doi :10.1016/S0034-4257(98)00064-9. (Cf. page 134).
- Gribonval, Rémi. 2011. « Should Penalized Least Squares Regression be Interpreted as Maximum A Posteriori Estimation? ». *IEEE Transactions on Signal Processing* 59, n° 5 (mai) : 2405–2410. doi :10.1109/TSP.2011.2107908. (Cf. pages 44, 67).
- Gupta, Amit, et Nagar. 1999. *Matrix Variate Distributions*. 1^{re} éd. Monographs and Surveys in Pure and Applied Mathematics. Chapman / Hall/CRC. (Cf. page 175).

- Hoeffding, Wassily. 1961. « The strong law of large numbers for U-statistics ». *Institute of Statistics mimeo series* 302. (Cf. page 173).
- Hoff, Peter D. 2007. « Model Averaging and Dimension Selection for the Singular Value Decomposition ». *Journal of the American Statistical Association* 102 (478) : 674–685. <http://www.jstor.org/stable/27639896>. (Cf. page 163).
- . 2009. « Simulation of the Matrix Bingham–von Mises–Fisher Distribution, With Applications to Multivariate and Relational Data ». *Journal of Computational and Graphical Statistics* 18 (2) : 438–456. (Cf. pages 119, 165).
- Horváth, Lajos, et Piotr Kokoszka. 2012. *Inference for Functional Data with Applications*. Springer New York. doi : [10.1007/978-1-4614-3655-3](https://doi.org/10.1007/978-1-4614-3655-3). (Cf. page 141).
- Ilic, J., et T. Strohmer. 2009. « PAPR reduction in OFDM using Kashin’s representation ». Dans *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Comm.* 444–448. Perugia, Italy. doi : [10.1109/SPAWC.2009.5161824](https://doi.org/10.1109/SPAWC.2009.5161824). (Cf. pages 17, 22, 43).
- Indyk, Piotr. 2008. « Explicit Constructions for Compressed Sensing of Sparse Signals ». Dans *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 30–33. SODA ’08. San Francisco, California : Society for Industrial / Applied Mathematics. <http://dl.acm.org/citation.cfm?id=1347082.1347086>. (Cf. page 27).
- Jégou, H., M. Douze et C. Schmid. 2011. « Product Quantization for Nearest Neighbor Search ». *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, n° 1 (janvier) : 117–128. doi : [10.1109/TPAMI.2010.57](https://doi.org/10.1109/TPAMI.2010.57). (Cf. page 22).
- Jégou, H., T. Furon et J. J. Fuchs. 2012. « Anti-sparse coding for approximate nearest neighbor search ». Dans *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, 2029–2032. Mars. doi : [10.1109/ICASSP.2012.6288307](https://doi.org/10.1109/ICASSP.2012.6288307). (Cf. pages 17, 22, 68).
- Ji, S., Y. Xue et L. Carin. 2008. « Bayesian Compressive Sensing ». *IEEE Transactions on Signal Processing* 56, n° 6 (juin) : 2346–2356. doi : [10.1109/TSP.2007.914345](https://doi.org/10.1109/TSP.2007.914345). (Cf. page 30).
- Jiang, Ke, Brian Kulis et Michael I. Jordan. 2012a. « Small-Variance Asymptotics for Exponential Family Dirichlet Process Mixture Models ». Dans *Advances in Neural Information Processing Systems 25*, sous la direction de F. Pereira, C. J. C. Burges, L. Bottou et K. Q. Weinberger, 3158–3166. Curran Associates, Inc. <http://papers.nips.cc/paper/4853-small-variance-asymptotics-for-exponential-family-dirichlet-process-mixture-models.pdf>. (Cf. page 67).
- . 2012b. « Small-Variance Asymptotics for Exponential Family Dirichlet Process Mixture Models ». Dans *Advances in Neural Information Processing Systems 25*, sous la direction de F. Pereira, C. J. C. Burges, L. Bottou et K. Q. Weinberger, 3158–3166. Curran Associates, Inc. <http://papers.nips.cc/paper/4853-small-variance-asymptotics-for-exponential-family-dirichlet-process-mixture-models.pdf>. (Cf. page 138).
- Johnson, N.L., et S. Kotz. 1977. *Urn models and their application : an approach to modern discrete probability theory*. WILEY SERIES in PROBABILITY and STATISTICS : APPLIED PROBABILITY and STATISTICS SECTION Series. Wiley. (Cf. page 89).
- Jolliffe, I. T. 1986. *Principal Component Analysis*. New York : Springer-Verlag. (Cf. page 108).
- Khintchine, A. 1934. « Korrelationstheorie der stationären stochastischen Prozesse ». *Mathematische Annalen* 109 : 604–615. <http://eudml.org/doc/159698>. (Cf. page 84).

- Kingman, J. F. C. 1967. « Completely random measures. » *Pacific Journal of Mathematics* 21 (1) : 59–78. <http://projecteuclid.org/euclid.pjm/1102992601>. (Cf. page 84).
- Knowles, David, et Zoubin Ghahramani. 2011. « Nonparametric Bayesian sparse factor models with application to gene expression modeling ». *Ann. Appl. Stat.* 5, n° 2B (juin) : 1534–1552. doi :[10.1214/10-AOAS435](https://doi.org/10.1214/10-AOAS435). (Cf. page 116).
- Knuth, Donald E. 1997. *The Art of Computer Programming, Volume 2 (3rd Ed.) : Seminumerical Algorithms*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc. (Cf. page 51).
- Koev, Plamen, et Alan Edelman. 2006. « The efficient evaluation of the hypergeometric function of a matrix argument ». *Mathematics of Computation* 75, n° 254 (janvier) : 833–847. doi :[10.1090/s0025-5718-06-01824-2](https://doi.org/10.1090/s0025-5718-06-01824-2). (Cf. page 123).
- Komodakis, N., et J. C. Pesquet. 2015. « Playing with Duality : An overview of recent primal-dual approaches for solving large-scale optimization problems ». *IEEE Signal Processing Magazine* 32, n° 6 (novembre) : 31–54. doi :[10.1109/MSP.2014.2377273](https://doi.org/10.1109/MSP.2014.2377273). (Cf. page 59).
- Kulis, Brian, et Michael I. Jordan. 2012. « Revisiting K-means : New Algorithms via Bayesian Nonparametrics ». Dans *Proceedings of the International Conference on Machine Learning (ICML)*, 1131–1138. ICML'12. Edinburgh, Scotland : Omnipress. <http://dl.acm.org/citation.cfm?id=3042573.3042719>. (Cf. page 67).
- Kume, Alfred, et Stephen G. Walker. 2006. « Sampling from compositional and directional distributions ». *Statistics and Computing* 16 (3) : 261–265. doi :[10.1007/s11222-006-8077-9](https://doi.org/10.1007/s11222-006-8077-9). (Cf. page 165).
- . 2009. « On the Fisher-Bingham distribution ». *Statistics and Computing* 19 (2) : 167–172. (Cf. page 163).
- Lacroux, Jean-Pierre. 2007. *Orthotypo : Orthographe et Typographie françaises*. Quintette. <http://www.orthotypographie.fr/index.html>. (Cf. page 17).
- Lawrence, Neil. 2005. « Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models ». *J. Mach. Learn. Res.* 6 (décembre) : 1783–1816. <http://dl.acm.org/citation.cfm?id=1046920.1194904>. (Cf. page 109).
- Lévy, Paul. 1937. *Théorie de l'addition des variables aléatoires*. Paris : Gauthier-Villars. (Cf. page 84).
- Lian, H. 2009. « Bayesian Nonlinear Principal Component Analysis Using Random Fields ». *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, n° 4 (avril) : 749–754. doi :[10.1109/TPAMI.2008.212](https://doi.org/10.1109/TPAMI.2008.212). (Cf. page 109).
- Luenberger, David G. 1969. *Optimization by vector space methods*. 1st. New York, NY, USA : Wiley. (Cf. page 37).
- Luu, Tung Duy, Jalal Fadili et Christophe Chesneau. 2017. « Sampling from non-smooth distribution through Langevin diffusion ». Working paper or preprint, mars. <https://hal.archives-ouvertes.fr/hal-01492056>. (Cf. page 50).
- Lyubarskii, Y., et R. Vershynin. 2010. « Uncertainty Principles and Vector Quantization ». *IEEE Transactions on Information Theory* 56, n° 7 (juillet) : 3491–3501. doi :[10.1109/TIT.2010.2048458](https://doi.org/10.1109/TIT.2010.2048458). (Cf. pages 17–19, 32–34, 36).
- Maceachern, Steven N. 1994. « Estimating normal means with a conjugate style dirichlet process prior ». *Communications in Statistics - Simulation and Computation* 23 (3) : 727–741. doi :[10.1080/03610919408813196](https://doi.org/10.1080/03610919408813196). (Cf. page 93).

- MacKay, David J. C. 1995. « Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks ». *Network : Computation in Neural Systems* 6 (3) : 469–505. doi : [10 . 1088 / 0954-898X\6\3\011](https://doi.org/10.1088/0954-898X\6\3\011). (Cf. page 110).
- Mallat, Stéphane G., et Zhifeng Zhang. 1993. « Matching pursuits with time-frequency dictionaries ». *IEEE Transactions on Signal Processing* 41, n° 12 (décembre) : 3397–3415. doi : [10 . 1109 / 78 . 258082](https://doi.org/10.1109/78.258082). (Cf. pages 27, 28).
- Marin, Jean-Michel, et Christian P. Robert. 2007. *Bayesian Core : A Practical Approach to Computational Bayesian Statistics*. New York, NY, USA : Springer. (Cf. page 133).
- Mathon, Benjamin, Teddy Furon, Laurent Amsaleg et Julien Bringer. 2013. « Recherche approximative de plus proches voisins efficace et sûre ». Dans *Actes du Colloque GRETSI*, ID238. Brest, France, septembre. [https : // hal . archives - ouvertes . fr / hal - 00823879](https://hal.archives-ouvertes.fr/hal-00823879). (Cf. pages 17, 22).
- McClellan, J. H., et T. W. Parks. 2005. « A personal history of the Parks-McClellan algorithm ». *IEEE Signal Processing Magazine* 22, n° 2 (mars) : 82–86. doi : [10 . 1109 / MSP . 2005 . 1406492](https://doi.org/10.1109/MSP.2005.1406492). (Cf. page 18).
- McCullagh, Peter, et Jie Yang. 2008. « How many clusters? » *Bayesian Anal.* 3, n° 1 (mars) : 101–120. doi : [10 . 1214 / 08 - BA304](https://doi.org/10.1214/08-BA304). (Cf. page 94).
- Miller, Jeffrey W., et Matthew T. Harrison. 2013. « A simple example of Dirichlet process mixture inconsistency for the number of components ». Dans *Advances in Neural Information Processing Systems 26*, sous la direction de C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani et K. Q. Weinberger, 199–206. Curran Associates, Inc. [http : // papers . nips . cc / paper / 4880 - a - simple - example - of - dirichlet - process - mixture - inconsistency - for - the - number - of - components . pdf](http://papers.nips.cc/paper/4880-a-simple-example-of-dirichlet-process-mixture-inconsistency-for-the-number-of-components.pdf). (Cf. page 170).
- . 2014. « Inconsistency of Pitman-Yor Process Mixtures for the Number of Components ». *J. Mach. Learn. Res.* 15, n° 1 (janvier) : 3333–3370. [http : // dl . acm . org / citation . cfm ? id = 2627435 . 2697071](http://dl.acm.org/citation.cfm?id=2627435.2697071). (Cf. pages 95, 120, 139, 170).
- Miller, Jeffrey, Brenda Betancourt, Abbas Zaidi, Hanna Wallach et Rebecca C. Steorts. 2015. « Microclustering : When the Cluster Sizes Grow Sublinearly with the Size of the Data Set » (décembre). eprint : [1512 . 00792](https://arxiv.org/abs/1512.00792). (Cf. pages 76, 145).
- Minka, Thomas P. 2000. « Automatic choice of dimensionality for PCA ». Dans *Advances in Neural Information Processing Systems*, 13 : 514. doi : [10 . 1 . 1 . 93 . 1144](https://doi.org/10.1.1.93.1144). (Cf. pages 110, 136).
- Moreau, Jean-Jacques. 1962. « Fonctions convexes duales et points proximaux dans un espace hilbertien ». *CR Acad. Sci. Paris Sér. A Math.* [http : // www . numdam . org / article / BSMF_1965 __ 93 __ 273_0 . pdf](http://www.numdam.org/article/BSMF_1965__93__273_0.pdf). (Cf. pages 49, 50, 52).
- Muirhead, Robb J. 1982. *Aspects of multivariate statistical theory*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley. (Cf. page 113).
- Muthukrishnan, S. 2005. « Data Streams : Algorithms and Applications ». *Foundations and Trends in Theoretical Computer Science* 1 (2) : 117–236. doi : [10 . 1561 / 0400000002](https://doi.org/10.1561/0400000002). (Cf. page 24).
- Neal, Radford M. 2012. « MCMC using Hamiltonian dynamics ». eprint : [arXiv : 1206 . 1901](https://arxiv.org/abs/1206.1901). (Cf. page 52).
- Needell, D., et J.A. Tropp. 2009. « CoSaMP : Iterative signal recovery from incomplete and inaccurate samples ». *Applied and Computational Harmonic Analysis* 26 (3) : 301–321. doi : [10 . 1016 / j . acha . 2008 . 07 . 002](https://doi.org/10.1016/j.acha.2008.07.002). (Cf. page 28).

- Neustadt, Lucien W. 1962. « Minimum Effort Control Systems ». *Journal of the Society for Industrial and Applied Mathematics Series A Control* 1, n° 1 (janvier) : 16–31. doi : [10.1137/0301002](https://doi.org/10.1137/0301002). (Cf. page 33).
- Nyquist, H. 1928. « Certain Topics in Telegraph Transmission Theory ». *Transactions of the American Institute of Electrical Engineers* 47, n° 2 (avril) : 617–644. doi : [10.1109/T-AIEE.1928.5055024](https://doi.org/10.1109/T-AIEE.1928.5055024). (Cf. page 23).
- Olshausen, Bruno A., Bruno A. Olshausen et K. Jarrod Millman. 2000. « Learning sparse codes with a mixture-of-Gaussians prior ». *Advances in Neural Information Processing Systems* 12 : 841–847. doi : [10.1.1.40.4639](https://doi.org/10.1.1.40.4639). (Cf. page 30).
- Oppenheim, Alan V., Ronald W. Schaffer et John R. Buck. 1999. *Discrete-time Signal Processing (2Nd Ed.)* Upper Saddle River, NJ, USA : Prentice Hall. (Cf. page 18).
- Orbanz, Peter, et Yee Whye Teh. 2010. « Bayesian Nonparametric Models ». Dans *Encyclopedia of Machine Learning*. Springer. (Cf. page 9).
- Parikh, Neal, et Stephen Boyd. 2014. « Proximal Algorithms ». *Foundations and Trends in Optimization* (Hanover, MA, USA) 1, n° 3 (janvier) : 127–239. doi : [10.1561/2400000003](https://doi.org/10.1561/2400000003). (Cf. page 29).
- Parks, T., et J. McClellan. 1972. « Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase ». *IEEE Transactions on Circuit Theory* 19, n° 2 (mars) : 189–194. doi : [10.1109/TCT.1972.1083419](https://doi.org/10.1109/TCT.1972.1083419). (Cf. page 18).
- Pati, Y. C., R. Rezaifar et P. S. Krishnaprasad. 1993. « Orthogonal matching pursuit : recursive function approximation with applications to wavelet decomposition ». Dans *Proc. IEEE Asilomar Conf. Signals, Systems, Computers*, 40–44 vol.1. Novembre. doi : [10.1109/ACSSC.1993.342465](https://doi.org/10.1109/ACSSC.1993.342465). (Cf. page 28).
- Pereyra, Marcelo. 2016a. « Proximal Markov chain Monte Carlo algorithms ». *Statistics and Computing* 26 (4) : 745–760. doi : [10.1007/s11222-015-9567-4](https://doi.org/10.1007/s11222-015-9567-4). (Cf. pages 8, 51, 53, 60).
- . 2016b. « Revisiting maximum-a-posteriori estimation in log-concave models : from differential geometry to decision theory » (décembre). <https://arxiv.org/abs/1612.06149>. (Cf. pages 44, 67).
- Pereyra, Marcelo, et Steve McLaughlin. 2017. « Fast Unsupervised Bayesian Image Segmentation With Adaptive Spatial Regularisation ». *Trans. Img. Proc.* (Piscataway, NJ, USA) 26, n° 6 (juin) : 2577–2587. doi : [10.1109/TIP.2017.2675165](https://doi.org/10.1109/TIP.2017.2675165). (Cf. page 141).
- Pitman, Jim. 2006. *Combinatorial Stochastic Processes*. Sous la direction de Jean Picard. Springer-Verlag. doi : [10.1007/b11601500](https://doi.org/10.1007/b11601500). (Cf. page 88).
- Pitman, Jim, et Marc Yor. 1997. « The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator ». *The Annals of Statistics* 25, n° 2 (avril) : 855–900. doi : [10.1214/aop/1024404422](https://doi.org/10.1214/aop/1024404422). (Cf. page 105).
- Punskaya, E., C. Andrieu, Arnaud Doucet et W.J. Fitzgerald. 2002. « Bayesian curve fitting using MCMC with applications to signal segmentation ». *IEEE Transactions on Signal Processing* 50, n° 3 (mars) : 747–758. (Cf. page 113).
- Rasmussen, Carl Edward, et Christopher K. I. Williams. 2005. *Gaussian Processes for Machine Learning*. The MIT Press. (Cf. page 105).
- Remes, Eugene. 1936. « Sur une propriété extrême des polynômes de Tchebychef ». *Communications de l'Institut des Sciences Mathématiques et Mécaniques de l'Université de Kharkoff et de la Société Mathématique de Kharkoff* 13 (1) : 93–95. (Cf. page 18).
- Robert, Christian P. 2006. *Le choix bayésien*. 1^{re} éd. Statistique et probabilités appliquées. Springer-Verlag Paris. (Cf. pages 7, 8).

- Robert, Christian P., et George Casella. 2004. *Monte Carlo Statistical Methods*. Springer New York. doi :[10.1007/978-1-4757-4145-2](https://doi.org/10.1007/978-1-4757-4145-2). (Cf. page 52).
- Rousseau, Judith, et Kerrie Mengersen. 2011. « Asymptotic behaviour of the posterior distribution in overfitted mixture models ». *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* 73, n° 5 (août) : 689–710. doi :[10.1111/j.1467-9868.2011.00781.x](https://doi.org/10.1111/j.1467-9868.2011.00781.x). (Cf. page 139).
- Roy, Daniel M., et Yee Whye Teh. 2009. « The Mondrian Process ». Dans *Advances in Neural Information Processing Systems 21*, sous la direction de D. Koller, D. Schuurmans, Y. Bengio et L. Bottou, 1377–1384. Curran Associates, Inc. <http://papers.nips.cc/paper/3622-the-mondrian-process.pdf>. (Cf. page 105).
- Schmitt, Eric, et Kaveh Vakili. 2016. « The FastHCS algorithm for robust PCA ». *Statistics and Computing* 26 (6) : 1229–1242. doi :[10.1007/s11222-015-9602-5](https://doi.org/10.1007/s11222-015-9602-5). (Cf. page 109).
- Schwarz, Gideon. 1978. « Estimating the Dimension of a Model ». *The Annals of Statistics* 6, n° 2 (mars) : 461–464. doi :[10.1214/aos/1176344136](https://doi.org/10.1214/aos/1176344136). (Cf. page 109).
- Sethuraman, Jayaram. 1994. « A constructive definition of Dirichlet priors ». *Statistica Sinica* 4 (2) : 639–650. <http://www.jstor.org/stable/24305538>. (Cf. pages 87, 88).
- Shannon, C. E. 1949. « Communication in the Presence of Noise ». *Proceedings of the IRE* 37, n° 1 (janvier) : 10–21. doi :[10.1109/JRPROC.1949.232969](https://doi.org/10.1109/JRPROC.1949.232969). (Cf. page 23).
- Smídl, Vaclav, et Anthony Quinn. 2007. « On Bayesian principal component analysis ». *Comput. Stat. Data Anal.* 51, n° 9 (mai) : 4101–4123. (Cf. pages 110, 136).
- Starck, Jean-Luc, Fionn Murtagh et Jalal Fadili. 2010. *Sparse Image and Signal Processing : Wavelets, Curvelets, Morphological Diversity*. New York, NY, USA : Cambridge University Press. (Cf. page 29).
- Studer, Christoph, Tom Goldstein, Wotao Yin et Richard G. Baraniuk. 2014. « Democratic Representations » (janvier). eprint : [1401.3420](https://arxiv.org/abs/1401.3420). (Cf. pages 22, 33, 35, 40, 68, 70).
- Studer, Christoph, et E. G. Larsson. 2013. « PAR-Aware Large-Scale Multi-User MIMO-OFDM Downlink ». 31, n° 2 (février) : 303–313. doi :[10.1109/JSAC.2013.130217](https://doi.org/10.1109/JSAC.2013.130217). (Cf. page 40).
- Studer, Christoph, Wotao Yin et Richard G. Baraniuk. 2012. « Signal representations with minimum ℓ_∞ -norm ». Dans *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, 1270–1277. Octobre. doi :[10.1109/Allerton.2012.6483364](https://doi.org/10.1109/Allerton.2012.6483364). (Cf. pages 33, 72).
- Suarez, Adam J., et Subhashis Ghosal. 2016. « Bayesian Estimation of Principal Components for Functional Data ». *Bayesian Analysis* (avril). doi :[10.1214/16-ba1003](https://doi.org/10.1214/16-ba1003). (Cf. page 141).
- Teh, Yee Whye, et Dilan Gorur. 2009. « Indian Buffet Processes with Power-law Behavior ». Dans *Advances in Neural Information Processing Systems 22*, sous la direction d'Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams et A. Culotta, 1838–1846. Curran Associates, Inc. <http://papers.nips.cc/paper/3638-indian-buffet-processes-with-power-law-behavior.pdf>. (Cf. page 140).
- Teh, Yee Whye, Dilan Grür et Zoubin Ghahramani. 2007. « Stick-breaking Construction for the Indian Buffet Process ». Dans *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, sous la direction de Marina Meila et Xiaotong Shen, 2 : 556–563. Proceedings of Machine Learning Research. San Juan, Puerto Rico : PMLR, mars. (Cf. page 100).
- Teh, Yee Whye, Michael I. Jordan, M. J. Beal et D. M. Blei. 2006. « Hierarchical Dirichlet Processes ». *Journal of the American Statistical Association* 101 (476) : 1566–1581. (Cf. page 105).

- Thibaux, Romain, et Michael I. Jordan. 2007. « Hierarchical beta processes and the Indian buffet process ». IN *PRACTICAL NONPARAMETRIC AND SEMIPARAMETRIC BAYESIAN STATISTICS 2007* : 227–242. doi :[10 . 1 . 1 . 121 . 455](https://doi.org/10.1.1.121.455). (Cf. pages 97, 100, 105).
- Tibshirani, Robert. 1996. « Regression Shrinkage and Selection via the Lasso ». *Journal of the Royal Statistical Society. Series B (Methodological)* 58 (1) : 267–288. <http://www.jstor.org/stable/2346178>. (Cf. page 25).
- Tikhonov, Andrey. 1963. « Solution of incorrectly formulated problems and the regularization method ». *Soviet Math. Dokl.* 4. (Cf. pages 25, 31).
- Tillmann, A. M., et M. E. Pfetsch. 2014. « The Computational Complexity of the Restricted Isometry Property, the Nullspace Property, and Related Concepts in Compressed Sensing ». *IEEE Transactions on Information Theory* 60, n° 2 (février) : 1248–1259. doi :[10 . 1109/TIT . 2013 . 2290112](https://doi.org/10.1109/TIT.2013.2290112). (Cf. page 36).
- Tipping, Michael E., et Christopher M. Bishop. 1999a. « Mixtures of Probabilistic Principal Component Analyzers ». *Neural Comput.* (Cambridge, MA, USA) 11, n° 2 (février) : 443–482. doi :[10 . 1162/089976699300016728](https://doi.org/10.1162/089976699300016728). (Cf. pages 108, 112).
- . 1999b. « Probabilistic Principal Component Analysis ». *Journal of the Royal Statistical Society. Series B (Methodological)* 61, n° 3 (janvier) : 611–622. (Cf. pages 108, 109, 137, 138).
- Tufte, Edward R. 2001. *The Visual Display of Quantitative Information*. Cheshire, Connecticut : Graphics Press. (Cf. page 11).
- Vaiter, Samuel, Gabriel Peyré et Jalal M. Fadili. 2013. « Robustesse au bruit des régularisations polyédrales ». Dans *Actes du Colloque GRETSI*, ID130. Brest, France, septembre. <https://hal.archives-ouvertes.fr/hal-00927075>. (Cf. page 17).
- Wang, Yining, et Jun Zhu. 2015. « DP-space : Bayesian Nonparametric Subspace Clustering with Small-variance Asymptotics ». Dans *Proceedings of the International Conference on Machine Learning (ICML)*, sous la direction de David Blei et Francis Bach, 862–870. JMLR Workshop / Conference Proceedings. <http://jmlr.org/proceedings/papers/v37/wanga15.pdf>. (Cf. page 112).
- Zaninetti, L. 2013. « A right and left truncated gamma distribution with application to the stars ». *Advanced Studies in Theoretical Physics* 7 : 1139–1147. doi :[10 . 12988/astp . 2013 . 310125](https://doi.org/10.12988/astp.2013.310125). (Cf. pages 114, 166).
- Zare, A., et P.D. Gader. 2008. « Endmember detection using the Dirichlet process ». Dans *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 1–4. Décembre. doi :[10 . 1109 / ICPR . 2008 . 4761289](https://doi.org/10.1109/ICPR.2008.4761289). (Cf. page 111).
- Zhang, Yunong, Jun Wang et Yangsheng Xu. 2002. « A dual neural network for bi-criteria kinematic control of redundant manipulators ». *IEEE Transactions on Robotics and Automation* 18, n° 6 (décembre) : 923–931. doi :[10 . 1109/TRA . 2002 . 805651](https://doi.org/10.1109/TRA.2002.805651). (Cf. page 19).
- Zhang, Zhihua, Kap Luk Chan, James T. Kwok et Dit-Yan Yeung. 2004. « Bayesian Inference on Principal Component Analysis Using Reversible Jump Markov Chain Monte Carlo ». Dans *Proceedings of the 19th National Conference on Artificial Intelligence*, 372–377. AAAI'04. San Jose, California : AAAI Press. <http://dl.acm.org/citation.cfm?id=1597148.1597210>. (Cf. page 111).

Cette bibliographie contient 166 entrées.

Modèles bayésiens pour l'identification de représentations antiparcimonieuses et l'analyse en composantes principales bayésienne non paramétrique

Cette thèse étudie deux modèles paramétriques et non paramétriques pour le changement de représentation. L'objectif des deux modèles diffère. Le premier cherche une représentation en plus grande dimension pour gagner en robustesse. Le second projette les données dans un sous-espace pertinent à des fins de modélisation. Une introduction au codage antiparcimonieux est présentée. L'objectif est de répartir uniformément l'information d'un signal sur toutes les composantes de sa représentation *en plus grande dimension* dans un but de robustesse. La recherche d'un tel code s'exprime comme un problème inverse impliquant une régularisation de type ℓ_∞ . À ce jour, seules les méthodes déterministes nécessitant le réglage de paramètres ont été explorées. Nous proposons une formulation bayésienne du problème impliquant une nouvelle loi de probabilité baptisée démocratique, qui pénalise les fortes amplitudes. Un échantillonneur de Gibbs et deux algorithmes MCMC proximaux sont présentés pour approcher des estimateurs bayésiens. La méthode non supervisée présentée est appelée BAC-1. Des expériences numériques illustrent les performances de l'approche pour la réduction de facteur de crête. À des fins de modélisation, il est nécessaire d'identifier un sous-espace pertinent *de dimension réduite*. Mais les méthodes probabilistes proposées nécessitent généralement de fixer à l'avance la dimension du sous-espace. Ce travail introduit une version bayésienne non paramétrique de l'analyse en composantes principales nommée BNP-PCA. La méthode couple une loi uniforme sur l'ensemble des bases orthonormales à un *a priori* non paramétrique de type buffet indien pour favoriser une utilisation parcimonieuse des composantes principales. Tous les paramètres inconnus sont intégrés au sein d'un modèle hiérarchique de sorte qu'aucun réglage n'est nécessaire. L'inférence est réalisée à l'aide des méthodes de MCMC. Une étude de l'estimation de la dimension du sous-espace est menée. Le comportement de la méthode BNP-PCA est étudié à travers plusieurs expériences numériques. Nous montrons la flexibilité de la méthode BNP-PCA sur deux applications.

Mots-clés : antiparcimonie, loi démocratique, méthodes proximales, méthode de Monte-Carlo par chaînes de Markov, bayésien non paramétrique, variété de Stiefel, processus du buffet indien.

This thesis proposes Bayesian parametric and nonparametric models for signal representation. The first model infers a higher dimensional representation of a signal for sake of robustness. The second model identifies a lower dimensional representation of a signal for modelisation and model selection. Enforcing the information to be spread uniformly over a higher dimensional representation exhibits relevant properties such as robustness. These so called anti-sparse representations are obtained by solving a linear inverse problem with an ℓ_∞ -norm penalty. We propose in this thesis a Bayesian formulation of anti-sparse coding involving a new probability distribution, referred to as the democratic prior. A Gibbs and two proximal samplers are proposed to approximate Bayesian estimators. The algorithm is called BAC-1. Simulations on synthetic data illustrate the performances of the two proposed samplers and the results are compared with state-of-the art methods. Principal component analysis is very popular to perform dimension reduction. The selection of the number of significant components is essential but often based on some practical heuristics depending on the application. Few works have proposed a probabilistic approach to infer the number of significant components. We propose a Bayesian nonparametric principal component analysis (BNP-PCA). The proposed model involves an Indian buffet process to promote a parsimonious use of principal components, which is assigned a prior distribution defined on the manifold of orthonormal basis. Inference is done using MCMC methods. The estimators of the latent dimension are theoretically and empirically studied. The relevance of the approach is assessed on two applications.

keywords : anti-sparse representation, democratic distribution, proximal methods, MCMC methods, Bayesian nonparametrics, dimension reduction, distribution on the Stiefel manifold, Indian buffet process.