Approches bayésiennes non paramétriques et apprentissage de dictionnaire pour les problèmes inverses en traitement d'image

Hong-Phuong Dang

Numéro d'ordre : 311

CENTRALE LILLE

THÈSE

Présentée en vue d'obtenir le grade de

DOCTEUR

En

Spécialité
Automatique, Génie Informatique, Traitement du Signal et Image

Par

Hong-Phuong DANG

DOCTORAT DELIVRÉ PAR CENTRALE LILLE

Approches bayésiennes non paramétriques et apprentissage de dictionnaire pour les problèmes inverses en traitement d'image

Soutenue le 1er décembre 2016 devant le jury d'examen :

Présidente	
Agnès Desolneux	Directrice de recherche, CNRS, Paris
Rapporteurs	
Florence Forbes	Directrice de Recherche, INRIA, Grenoble
Cédric Févotte	Directeur de Recherche, CNRS, Toulouse
Examinateurs	
Jérôme Idier	Directeur de recherche, CNRS, Nantes
Stéphane Canu	Professeur des Universités, INSA, Rouen
Directeur de thèse	
Pierre Chainais	Maître de Conférence, HDR, Centrale Lille

Thèse préparée dans le Laboratoire

Centre de Recherche en Informatique Signal et Automatique de Lille
Université de Lille, Centrale Lille, CNRS, UMR 9189 - CRIStAL

Ecole Doctorale SPI 072
(Lille I, Lille III, Artois, ULCO, UVHC, Centrale Lille)

Numéro d'ordre : 311

Thèse de Doctorat

préparée au

Centre de Recherche en Informatique Signal et Automatique de Lille

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

Approches bayésiennes non paramétriques et apprentissage de dictionnaire pour les problèmes inverses en traitement d'image

soutenue le 1er décembre 2016 par

HONG-PHUONG DANG

pour obtenir le grade de **Docteur de Centrale Lille**

Spécialité
Automatique, Génie Informatique, Traitement du Signal et Image

Jury

Rapporteurs	
Florence Forbes	Directrice de Recherche, INRIA, Grenoble
Cédric Févotte	Directeur de Recherche, CNRS, Toulouse
Examinateurs	
Agnès Desolneux	Directrice de recherche, CNRS, Paris
Jérôme Idier	Directeur de recherche, CNRS, Nantes
Stéphane Canu	Professeur des Universités, INSA, Rouen
Directeur de thèse	
Pierre Chainais	Maître de Conférence, HDR, Centrale Lille

"Malgré les gens qui s'attachent à leurs goûts et n'aiment pas ce que vous faites, restez fidèles à ce que vous êtes. Il n'y a pas de limite à notre ambition ..... tout est possible à qui rêve, ose, travaille et n'abandonne jamais."

Xavier Dolan

Remerciements

Au bout de trois ans, j'arrive à la fin de ma thèse. J'en retire une grande expérience professionnelle et humaine. Je tiens à adresser mes remerciements à celles et ceux qui m'ont permis de réaliser ce travail et qui m'ont accompagnées tout au long de ce chemin.

Je tiens à remercier de tout cœur mes parents, mon frère qui m'ont toujours soutenus dans mes choix et qui ont tout mis en œuvre pour que je puisse mener à bien mes études. Merci pour tout ce que vous m'avez transmis, pour votre soutien, vos encouragements et votre confiance en moi. Merci à ma marraine pour ta bienveillante attention, pour tes bons repas chaque fois que je viens à Lyon.

Merci à mes anciens profs pour m'avoir donné le goût des sciences. Grâce à vous que j'ai eu le courage de choisir ce chemin. Je tiens à remercier bien particulier à Stéphane Canu et Gilles Gasso. Je remercie profondément aussi à Monsieur Emmanuel Duflos pour m'avoir présenté cette thèse.

Je tiens à remercier les membres du jury, qui ont accepté m'accompagner lors de la dernière phase de ce travail. Merci aux rapporteurs Florence Forbes et Cédric Févotte qui ont consacré du temps à la lecture de mes travaux. Merci pour les suggestions et les remarques intéressantes qu'ils m'ont indiquées. Je remercie également Agnès Desolneux, Jérôme Idier, Stéphane Canu qui ont bien voulu examiner cette thèse.

Durant mes trois années, j'ai eu la chance de rencontrer et travailler avec plusieurs chercheurs qui m'ont partagé leurs connaissances, leur enthousiasme pour la science.

Merci à François Caron, Nicolas Dobigeon, Jean-François Giovannelli, Audrey Giremus, François Septier à tous les chercheurs du projet ANR-BNPSI pour vos conseils pertinents.

Je remercie en particulier à François Caron pour m'avoir accueilli à Oxford. Nos discussions autour du buffet Indien et des réseaux bipartites m'ont éclairci énormément sur les aspects bayésiens non paramétriques.

Et surtout, *je suis tombé* sur une équipe Σ de rêve, quel bonheur de travailler avec vous.

Je n'aurai jamais assez de mots pour dire ce que je dois à Pierre, mon directeur, la première personne qui m'a guidé sur ce chemin, pour toute son assistance, sa rigueur, ses interpellations, ses commentaires critiques toujours pertinents, son disponibilité (parfois trop, parfois pas assez) tout au long de cette période. Merci d'avoir accepté conduire ce travail de bout en bout jusqu'au jour d'aujourd'hui, d'avoir souffert les affres de la rédaction avec moi.

Un grand merci également à Patrick, the big boss, pour tes conseils STENIQ, pour les dicussions sur le cinéma et les bandes dessinées et aussi pour les soirées dans un lieu chaud avec une boisson fraîche , je tiens également à remercier Ingrid, la co-organisatrice de ces soirées.

Merci Rémi pour les éclaircissements sur MCMC, pour les histoires de la France et de Lille. On y arrive mon petit demi-frère scientifique Julien oye, merci pour le beaufort, les gauffres, les pintes et les bêtises. Et merci aussi au petit nouveau Guillaume pour les gâteaux de riz et les cannelés, ils sont trop bons. Merci à tous les membres d'équipe Σ : Jonh, François, Christelle, Jérémy, Vincent, Wadih, ... Merci pour tous les bons moments passés au repas de midi, en pause café avec vous qui m'ont donné envie de continuer en thèse.

Merci à Sophie qui m'a fait découvrir le cirque, mes années à Lille seraient moins de goût sans toi. Merci à Romain, Quentin et Maxime qui sont venus pour me soutenir. Merci à Quentin et Maxime pour me remettre à niveau C et boucher des fuites de mémoires.

Finalement, ayant gardé le meilleur pour la fin, merci à Clément qui est resté, en train de rester à côté de moi. Merci de m'avoir sorti de mes doutes, mes hésistations. Merci pour tout le temps que tu as passé à relire mon manuscrit, pour ton soutien moral, pour m'avoir supporté pendant la rédaction avec tant de patience (je sais à quel point j'étais insupportable). Merci pour tout !

Approches bayésiennes non paramétriques et apprentissage de dictionnaire pour les problèmes inverses en traitement d'image

L'apprentissage de dictionnaire pour la représentation parcimonieuse est bien connu dans le cadre de la résolution de problèmes inverses. Les méthodes d'optimisation et les approches paramétriques ont été particulièrement explorées. Ces méthodes rencontrent certaines limitations, notamment liées au choix de paramètres. En général, la taille de dictionnaire doit être fixée à l'avance et une connaissance des niveaux de bruit et éventuellement de parcimonie sont aussi nécessaires. Les contributions méthodologiques de cette thèse concernent l'apprentissage conjoint du dictionnaire et de ses paramètres, notamment pour les problèmes inverses en traitement d'image. Nous étudions et proposons la méthode IBP-DL (Indien Buffet Process for Dictionary Learning) en utilisant une approche bayésienne non paramétrique. Une introduction sur les approches bayésiennes non paramétriques est présentée. Le processus de Dirichlet et son dérivé, le processus du restaurant chinois, ainsi que le processus Bêta et son dérivé, le processus du buffet indien, sont décrits. Le modèle proposé pour l'apprentissage de dictionnaire s'appuie sur un a priori de type Buffet Indien qui permet d'apprendre un dictionnaire de taille adaptative. Nous détaillons la méthode de Monte-Carlo proposée pour l'inférence. Le niveau de bruit et celui de la parcimonie sont aussi échantillonnés, de sorte qu'aucun réglage de paramètres n'est nécessaire en pratique. Des expériences numériques illustrent les performances de l'approche pour les problèmes du débruitage, de l'inpainting et de l'acquisition compressée. Les résultats sont comparés avec l'état de l'art. Le code source en Matlab et en C est mis à disposition.

Mots-clés : représentations parcimonieuses, apprentissage de dictionnaire, prolèmes inverses, bayésien non paramétrique, processus du Buffet Indien, Monte-Carlo par chaînes de Markov

Bayesian nonparametric approaches and dictionary learning for inverse problems in image processing

Dictionary learning for sparse representation has been widely advocated for solving inverse problems. Optimization methods and parametric approaches towards dictionary learning have been particularly explored. These methods meet some limitations, particularly related to the choice of parameters. In general, the dictionary size is fixed in advance, and sparsity or noise level may also be needed. In this thesis, we show how to perform jointly dictionary and parameter learning, with an emphasis on image processing. We propose and study the Indian Buffet Process for Dictionary Learning (IBP-DL) method, using a bayesian nonparametric approach. A primer on bayesian nonparametrics is first presented. Dirichlet and Beta processes and their respective derivatives, the Chinese restaurant and Indian Buffet processes are described. The proposed model for dictionary learning relies on an Indian Buffet prior, which permits to learn an adaptive size dictionary. The Monte-Carlo method for inference is detailed. Noise and sparsity levels are also inferred, so that in practice no parameter tuning is required. Numerical experiments illustrate the performances of the approach in different settings : image denoising, inpainting and compressed sensing. Results are compared with state-of-the art methods is made. Matlab and C sources are available for sake of reproducibility.

Keywords : sparse representations, dictionary learning, inverse problems, Bayesian non-parametric, Indian Buffet Process, Markov chain Monte Carlo

Sommaire

Table des figures

Liste des tableaux

Liste des algorithmes

Nomenclature

Symbol	Description
N, i	Nombre, indice d'observations
L, ℓ	Dimension, indice de la dimension d'une observation
K, k	Nombre, indice d'atomes
$\mathbf{C}$	Matrice $\mathbf{C}$
$\mathbf{c}_i = \mathbf{C}(:, i)$	$i^{\text{ème}}$ vecteur colonne de $\mathbf{C}$
$\mathbf{c}_{j,:} = \mathbf{C}(j, :)$	$j^{\text{ème}}$ vecteur ligne de $\mathbf{C}$
c_{ji}	scalaire c_{ji} de la $j^{\text{ème}}$ ligne et $i^{\text{ème}}$ colonne de $\mathbf{C}$
$\mathbf{Y}$	Matrice d'observations
$\mathbf{W}$	Matrice de coefficients
$\mathbf{D}$	Matrice de dictionnaire
$\mathbb{I}_A$	Matrice identité de taille A
$\mathcal{H}$,	Ensemble de matrices d'opérateur d'observation
$\mathbf{H}_i$	Matrice d'opérateur de $i^{\text{ème}}$ observation
$\boldsymbol{\Sigma}$, $\boldsymbol{\mu}$	Matrice de covariance et vecteur d'espérence
Σ, μ	Variance et espérance
σ, σ^2, μ	Ecart-type, variance et espérance
$\mathcal{P}, \mathcal{N}, \mathcal{U}$	Distributions : Poissons, Gaussian, Uniforme
$\mathcal{B}, \mathcal{G}, \mathcal{IG}$	Distributions : Bêta, Gamma, Inverse Gamma

Acronymes	Description
ACGS	Accelerated Collapsed Gibbs Sampling
BeP	Bernoulli Process
BP	Beta Process
BNP	Bayésien Non Paramétrique (Bayesian nonparametrics)
CRP	Chinese Restaurant Process
CS	Compressed Sensisng
DL	Dictionary Learning
DP	Dirichlet Process
IBP	Indian Buffet Process
IBP-DL	Indian Buffet Process for Dictionary Learning
i.i.d.	indépendantes et identiquement distribuées
lof	left-ordered form
MCMC	Markov-Chain Monte Carlo
MH	Metropolis-Hastings
PYP	Pitman-Yor process
RPM	Random Probability Measure

CHAPITRE 1

Introduction

Ma 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 1er janvier 2015 le Centre de Recherche en Informatique, Signal et Automatique de Lille (CRIStAL). Elle concerne l'étude de **problèmes inverses** en traitement d'image utilisant **les approches bayésiennes non paramétriques** pour **l'apprentissage de dictionnaire**. Mes travaux font également partie du projet ANR-BNPSI (*Bayesian Nonparametric methods for Signal and Image Processing*) n°ANR-13-BS-03-0006-01.

1.1 Problème inverse

"Un problème inverse est une situation dans laquelle on tente de déterminer les causes d'un phénomène à partir de l'observation expérimentale de ses effets." En traitement d'image, un problème inverse est typiquement celui de la reconstruction d'une image plausible à partir d'une image observée $\mathbf{Y}$. L'observation est le résultat d'une fonction $f(\mathbf{H}, \mathbf{X}, \mathbf{B})$ éventuellement non linéaire. En particulier, dans le cas linéaire on a :

$$\mathbf{Y} = \mathbf{H}(\mathbf{X} + \mathbf{B}). \tag{1.1}$$

$\mathbf{X}$ est l'image initiale qui est perturbée par un opérateur $\mathbf{H}$ et un bruit $\mathbf{B}$. Cela donne l'image observable $\mathbf{Y}$. On cherche à retrouver $\mathbf{X}$ à partir de $\mathbf{Y}$. L'opérateur d'observation $\mathbf{H}$ peut être :

- la matrice identité : il s'agit alors d'un problème de débruitage,
- un masque : le problème s'appelle l'inpainting où certains pixels de l'image sont désactivés et le but est de retrouver l'image entière,
- un opérateur de flou : on dit alors que l'on veut déconvoluer l'image, *i.e.* reconstruire l'image nette d'une scène originale à partir d'une version floue,
- une matrice de projection aléatoire dans le cas de l'acquisition compressée (*compressive sensing*).

Ces problèmes de reconstruction ou de restauration d'image sont souvent des problèmes mal-posés du fait d'une perte d'information lors de l'observation. Un problème est dit mal-posé au sens d'Hadamard s'il ne respecte pas l'une des conditions suivantes :

- la solution existe,
- la solution est unique,
- la solution dépend de façon continue des données (*i.e.* une petite variation dans les données n'introduit pas de grande variation dans la solution).

Dans le cas linéaire de l'équation (1.1), cette propriété peut découler de la matrice $\mathbf{H}$ lorsqu'elle est non-inversible ou mal-conditionnée. Il existe alors plusieurs solutions à l'équation (1.1). Les seules observations expérimentales $\mathbf{Y}$ ne suffisent pas à déterminer parfaitement $\mathbf{X}$.

Une approche naturelle pour résoudre (1.1) est la méthode des *moindres carrés* qui permet de s'affranchir de la difficulté posée par la matrice non-inversible. Cette méthode donne un estimateur $\hat{\mathbf{X}}_{MC}$ minimisant la norme ℓ_2 du résidu du modèle :

$$\hat{\mathbf{X}}_{MC} = \underset{\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_2. \tag{1.2}$$

Cela consiste à résoudre l'équation suivante :

$$\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{X} = \mathbf{H}^{\mathrm{T}}\mathbf{Y}. \tag{1.3}$$

La solution des moindres carrés est unique sous condition que $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ soit inversible. Elle est donnée par

$$\hat{\mathbf{X}}_{MC} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{Y}. \tag{1.4}$$

Dans le cas où $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ n'est pas inversible, la solution n'est plus unique. Pour se ramener à une solution unique, parmi toutes les solutions vérifiant $\mathbf{Y} = \mathbf{H}\mathbf{X}$, on peut choisir la solution de norme euclidienne minimale. On peut référencer ici aussi la régression pseudo-orthogonale connue sous le nom de régression ridge [1].

Bien qu'il soit possible d'obtenir une solution unique, une pseudo inversion faite au sens des moindres carrés ne conduit pas à une bonne solution du problème inverse à cause de son instabilité, aussi appelée non robustesse au bruit. En effet, l'équation (1.4) n'exploite que des informations apportées par les données $\mathbf{Y}$ qui ne sont en général pas suffisantes pour avoir une solution unique ou robuste au bruit. Dans ce cas, il est nécessaire d'ajouter des contraintes ou des *a priori* qui permettent de réduire l'espace des possibilités afin de définir une notion de meilleure représentation et aboutir à une solution unique. Le choix d'un *a priori* est dit pertinent s'il permet de retrouver les représentations qui ont les propriétés souhaitées. Par exemple, on peut désirer la parcimonie.

1.2 Représentation parcimonieuse

Ces vingt dernières années, les représentations parcimonieuses ont levé de nombreux verrous en traitement du signal et des images [2–4]. *La parcimonie* consiste par exemple à utiliser *un petit nombre d'atomes* choisis dans un dictionnaire pour reconstruire ou restaurer un signal, une image. On cherche à décomposer le signal en éléments basiques appelés **atomes**. Chaque atome représente une partie élémentaire du signal complet. Soit $\mathbf{D}$ le dictionnaire contenant K atomes. Nous notons $\mathbf{d}_k = \mathbf{D}(:, k)$, l'atome numéro k du dictionnaire. L'image $\mathbf{X}$ à retrouver peut être recherchée sous la forme suivante :

$$\mathbf{X} = \sum_{k=1}^{K} \mathbf{d}_k \mathbf{W}(k, :) \tag{1.5}$$

où $\mathbf{W}(k,:)$ est les coefficients de la représentation $\mathbf{W}$ associés à l'atome k. $\mathbf{X}$ est donc la somme de tous les atomes pondérés pas les coefficients associés. Il est à noter qu'en traitement d'image, les atomes sont souvent sous la forme d'imagettes élémentaires appelées *patches* [4]. On appelle imagette (*patch*) un morceau souvent de taille 8 extrait à partir d'une image.

$\mathbf{W}$ est souvent appelée la matrice de coefficients ou le jeu de coefficients. Lorque la matrice $\mathbf{W}$ a un grand nombre de coefficients nuls, on parle de représentation parcimonieuse. On trouve aussi les termes représentation éparse, représentation creuse ou *sparse representation* en anglais. Les coefficients nuls indiquent les atomes non utilisés. Les coefficients non-nuls indiquent la présence des atomes dans le signal. Il peut exister une infinité de jeux de coefficients $\mathbf{W}$ permettant d'effectuer des combinaisons d'atomes différentes pour décomposer le même signal. Il est nécessaire de définir un critère pour choisir un unique jeu de coefficients $\mathbf{W}$ parmi toutes les décompositions possibles. Dans cette thèse, on s'intéresse au critère de parcimonie. En effet, la parcimonie est une approche pour résoudre les problèmes inverses où l'on est à la limite de l'identifiabilité. Le chapitre 2 revient en détails sur ces représentations parcimonieuses.

Il faut alors distinguer deux approches. Une première possibilité consiste à choisir à l'avance le dictionnaire parmi un ensemble de dictionnaires préexistants en utilisant des fonctions mathématiques (la transformée en cosinus discrète, la transformée en ondelette, *curvelets*, ...) pour ensuite identifier les coefficients associés en imposant que ceux-ci soient les moins nombreux possibles. Différentes méthodes (LASSO [5], Basis Matching Pursuit [6],...) permettent alors d'estimer ces coefficients en résolvant un problème d'optimisation, typiquement avec pénalité ℓ_0 ou ℓ_1 (relaxation d'une pénalité ℓ_0) sur les coefficients. Le choix *a priori* du dictionnaire est alors crucial et influence beaucoup la qualité des résultats obtenus.

Une deuxième possibilité consiste à apprendre un dictionnaire à partir d'un ensemble de données de référence. Une étape d'apprentissage est nécessaire, en plus de l'estimation des coefficients de la décomposition. Cette thèse adoptera cette approche.

1.3 Apprentissage de dictionnaire

On peut aborder l'apprentissage de dictionnaire sous l'angle de la "factorisation de matrice", où l'on décompose $\mathbf{X} = \mathbf{DW}$. Le modèle (1.1) devient :

$$\mathbf{Y} = \mathbf{H}(\mathbf{DW} + \mathbf{B}). \tag{1.6}$$

La récupération de $\mathbf{X}$ est équivalent dans un certain sens à la recherche d'un couple optimal $(\mathbf{D}, \mathbf{W})$ dans un sens qui reste à préciser. $\mathbf{D}$ est un dictionnaire redondant où le nombre d'atomes du dictionnaire est supérieur à la dimension des données.

L'apprentissage de dictionnaire par les méthodes d'optimisation a été beaucoup étudié [7–10]. Dans ces méthodes d'optimisation, la parcimonie est typiquement favorisée par une pénalité ℓ_0 ou ℓ_1 sur l'ensemble des coefficients de codage (d'autres

formulations sont possibles). Par exemple dans le cas du débruitage où $\mathbf{H}$ est la matrice identité :

$$(\mathbf{D}, \mathbf{W}) = \underset{(\mathbf{D},\mathbf{W})}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{W}\|_2^2 + \lambda\|\mathbf{W}\|_p \text{ où } \lambda \geqslant 0. \tag{1.7}$$

Une optimisation alternée entre $\mathbf{D}$ et $\mathbf{W}$ est proposée pour résoudre (1.7) [4]. Les méthodes d'optimisation visent à minimiser l'erreur de reconstruction tout en favorisant la parcimonie, comme dans l'équation (1.7). Il faut donc trouver un bon compromis entre l'erreur de reconstruction et la parcimonie. Ce compromis est contrôlé par λ appelé paramètre de régularisation. Il est à noter que le choix du paramètre de régularisation λ **en fonction de l'erreur de reconstruction** est important. Plus λ est grand, plus on impose la parcimonie, moins on utilise d'atomes et plus l'erreur de reconstruction est grande. Prenons un exemple sur l'équation (1.7). Dans le cas extrême, quand $\lambda \to \infty$, $\|\mathbf{W}\|_1 \to 0$ pour compenser. Quand aucun atome n'est utilisé, l'erreur est maximale. Inversement, quand $\lambda = 0$, il n'y a plus de contrainte de parcimonie. La décomposition utilise autant d'atomes que nécessaire pour minimiser l'erreur de reconstruction. Cependant rappelons qu'on effectue la décomposition sur l'image observée qui est l'image bruitée par exemple. L'objectif est de débruiter cette image. L'erreur de reconstruction joue un rôle analogue au niveau du bruit. On cherche une approximation de l'image bruitée en acceptant un certain niveau d'erreur lié au niveau de bruit. Quand $\lambda = 0$, on approche la reconstruction exacte du signal de l'image bruitée $\mathbf{Y}$ qui n'est pas satisfaisante ici. Dans les méthodes d'optimisation, un *a priori* sur l'erreur de reconstruction est nécessaire afin de choisir de façon optimale en avant le paramètre de régularisation λ. Bien que souvent robustes, les méthodes d'optimisation souffrent de certaines limitations. Elles fixent souvent à l'avance l'erreur de reconstruction, la taille du dictionnaire ou le niveau de parcimonie. Dans la littérature, un dictionnaire de taille $K = 256$ ou 512 atomes est appris le plus souvent [4, 7]. Quelques méthodes proposent un dictionnaire de taille adaptative en visant un compromis entre l'erreur de reconstruction et la parcimonie de la représentation par exemple [11–14]. Cependant, l'estimation de la taille du dictionnaire est souvent basée sur des heuristiques.

Une deuxième famille d'approches explorée dans une moindre mesure regroupe les méthodes d'apprentissage de dictionnaires via des modèles probabilistes. Dans le cadre bayésien, le problème s'écrit typiquement sous la forme d'une loi *a posteriori* :

$$\mathrm{p}(\mathbf{D}, \mathbf{W}, \sigma_\varepsilon \mid \mathbf{Y}) \propto \mathrm{p}(\mathbf{Y} \mid \mathbf{D}, \mathbf{W}, \sigma_\varepsilon)\mathrm{p}(\mathbf{D}, \mathbf{W}, \sigma_\varepsilon) \tag{1.8}$$

La vraisemblance $\mathrm{p}(\mathbf{Y} \mid \mathbf{D}, \mathbf{W}, \sigma_\varepsilon)$ est construite conformément au modèle (1.6) où le bruit est souvent gaussien. La loi *a priori* $\mathrm{p}(\mathbf{D}, \mathbf{W}, \sigma_\varepsilon)$ permet de régulariser le problème. En utilisant par exemple l'échantillonnage de Gibbs pour l'inférence, le problème peut être résolu en échantillonnant alternativement les variables selon :

$$\mathrm{p}(\mathbf{D} \mid \mathbf{Y}, \mathbf{W}, \sigma_\varepsilon) \propto \mathrm{p}(\mathbf{Y} \mid \mathbf{D}, \mathbf{W}, \sigma_\varepsilon)\mathrm{p}(\mathbf{D}) \tag{1.9}$$

$$\mathrm{p}(\mathbf{W} \mid \mathbf{Y}, \mathbf{D}, \sigma_\varepsilon) \propto \mathrm{p}(\mathbf{Y} \mid \mathbf{D}, \mathbf{W}, \sigma_\varepsilon)\mathrm{p}(\mathbf{W}) \tag{1.10}$$

$$\mathrm{p}(\sigma_\varepsilon \mid \mathbf{Y}, \mathbf{W}, \sigma_\varepsilon) \propto \mathrm{p}(\mathbf{Y} \mid \mathbf{D}, \mathbf{W}, \sigma_\varepsilon)\mathrm{p}(\sigma_\varepsilon) \tag{1.11}$$

Un avantage des approches bayésiennes est que le niveau de bruit peut aussi être échantillonné. Il n'est plus nécessaire de le fixer à l'avance.

Les modèles bayésiens paramétriques possèdent *a priori* un nombre de degrés de liberté fixé à l'avance. Dans le cas de l'apprentissage de dictionnaire, le nombre d'atomes sera fixé à l'avance. Cependant, si le nombre d'atomes est trop faible, par exemple un seul atome, le dictionnaire ne sera pas assez riche pour reconstruire le signal. Si le nombre d'atomes est très important, par exemple infini, le problème du surapprentissage apparaît. Une sélection de modèle est en général nécessaire si l'on souhaite trouver un compromis optimal.

Dans les approches bayésiennes non paramétriques, le nombre de degrés de liberté du modèle est lui-même considéré comme aléatoire et finalement contrôlé par les données lors de l'inférence. De la même façon, elles permettent de développer des méthodes d'apprentissage de dictionnaire sans fixer à l'avance la taille du dictionnaire. Non seulement on échantillonne le dictionnaire, mais aussi son nombre d'atomes lors de l'inférence. Aucun paramètre n'est fixé à l'avance. Ces propriétés ouvrent des perspectives prometteuses et encore peu explorées en traitement du signal et des images. C'est pourquoi l'apprentissage de dictionnaire s'appuyant sur les approches bayésiennes non paramétriques est exploré dans cette thèse.

En particulier, le chapitre 3 rappelle différentes méthodes d'apprentissage de dictionnaire.

1.4 Approches bayésiennes non paramétriques

Pour rappel, les méthodes paramétriques font l'hypothèse que le modèle peut être caractérisé par un vecteur de paramètres de dimension finie. Par exemple, dans un modèle de mélange paramétrique, le nombre de composantes est fixé avant d'inférer les paramètres du modèle (poids et paramètres des composantes élémentaires). Le choix de la dimension des paramètres n'est pas toujours trivial et constitue une limitation des modèles paramétriques.

Les modèles non paramétriques permettent d'éviter les choix souvent restrictifs des modèles paramétriques en définissant une distribution a priori sur des espaces fonctionnels (de dimension infinie). Les modèles non paramétriques peuvent ainsi être simplement définis comme des modèles paramétriques avec un nombre infini de paramètres. Pour résumer, *non paramétriques* ne veut pas dire qu'on ignore l'existence des paramètres des distributions. Cela indique surtout que la dimension des paramètres inconnus est aussi une variable aléatoire. On va estimer ces paramètres ainsi que leur dimension en utilisant des lois définies sur des espaces de distributions.

Les modèles bayésiens non paramétriques souvent mentionnés sont les processus gaussiens et les processus de Dirichlet. En particulier, le processus de Dirichlet à mélange (*Dirichlet Process Mixture, DPM*) [15, 16] est une distribution sur les distributions de probabilité. Le DPM dépend de deux paramètres et ses réalisations sont des mélanges infinis, par exemple le processus de Dirichlet à mélange de gaussiennes ou Poisson. Il permet de prendre en compte dès le départ le caractère aléatoire du nombre de composantes d'un mélange. Le nombre de composantes n'a pas besoin

d'être défini à l'avance, mais il sera estimé à partir des données. Le processus du restaurant chinois est un dérivé du processus de Dirichlet. Le chapitre 4 présente en détails ces processus.

Dans le cas où chaque objet peut s'associer à plusieurs classes, on parle des modèles à variables latentes (*latent feature models*). Les processus Bêta-Bernoulli ou de Buffet Indien sont deux exemples de loi *a priori* qui permet de modéliser des objets possédant un nombre inconnu de caractéristiques (*feature*). Le nombre de caractéristiques est potentiellement infinie mais est régularisé par la loi *a priori* . Ces processus sont exposés dans le chapitre 5. En particulier, le chapitre 6 détaille l'échantillonnage du processus du Buffet Indien.

Les méthodes bayésiennes non paramétriques ont connu un intérêt croissant dans la communauté de l'apprentissage (*machine learning*) [17–21]. Cependant, elles sont encore très peu utilisé dans la communauté de traitement de signal et d'image [22, 23]. A noter au passage, une session spéciale sur *Bayesian non-parametrics for signal and image processing* a été organisée par François Caron, Pierre Chainais et Audrey Giremus en 2015 à EUSIPCO (European Signal Processing Conference).

1.5 Modèle IBP-DL

Les modèles non paramétriques possèdent de très bonnes propriétés d'adaptativité qui permettent de s'affranchir d'une étape de sélection de modèle, la complexité du modèle augmentant dynamiquement avec la richesse des données d'apprentissage. En utilisant les approches bayésiennes non paramétriques dans l'apprentissage de dictionnaire, nous pouvons considérer que le dictionnaire est potentiellement de taille infinie mais une loi a priori est introduite pour favoriser la parcimonie de la représentation.
Dans le cadre de cette thèse, on s'appuie sur le processus du Buffet Indien qui nous permet à la fois de contrôler le nombre d'atomes et de favoriser la parcimonie. Un modèle appelé IBP-DL pour *Indian Buffet Process for Dictionary Learning* ou le processus de Buffet Indien pour l'apprentissage de dictionnaire, sera présenté dans le chapitre 7.

L'objectif de nos travaux est précisément d'explorer le fort potentiel des méthodes bayésiennes non paramétriques pour l'apprentissage de dictionnaire, encore très peu utilisé par la communauté traitement du signal et des images. Cette thèse apporte des contributions méthodologiques, notamment pour les problèmes inverses en traitement d'image. Une approche bayésienne non paramétrique IBP-DL (Indian Buffet Process for Dictionary Learning) est proposée grâce à la loi *a priori* décrite par le Processus de Buffet Indien. Cette méthode donne un dictionnaire efficace avec un nombre d'atomes adapté. En outre, les niveaux du bruit et de la parcimonie sont également déduits de sorte qu'aucun réglage de paramètre n'est nécessaire. Le modèle IBP-DL est ensuite implémenté dans les problèmes inverses linéaires en traitement d'image. Le chapitre 8 illustre les résultats obtenus pour évaluer la pertinence de la méthode IBP-DL.

1.6 Liste des publications

Dans [24] et [25], un modèle IBP-DL pour le problème de débruitage est présenté. Une description plus détaillé de ce modèle est publiée dans [26]. Dans [27], le modèle IBP-DL est élaboré pour résoudre des problèmes de l'inpainting et compressive sensing au-delà du débruitage de base. Un échantillonneur de Gibbs marginalisé accéléré est aussi dérivé pour l'inpainting dans cette thèse.

1. Hong-Phuong Dang, Pierre Chainais. **Indian Buffet Process Dictionary Learning for image inpainting**. *IEEE Workshop on Statistical Signal Processing (SSP)*, 2016 [27].
2. Hong-Phuong Dang, Pierre Chainais. **Towards dictionaries of optimal size : a Bayesian non parametric approach**. *Journal of Signal Processing Systems*, 1-12, 2016 [26].
3. Hong-Phuong Dang, Pierre Chainais. **A Bayesian non parametric approach to learn dictionaries with adapted numbers of atoms**. Intel best paper award. *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6, 2015 [25].
4. Hong-Phuong Dang, Pierre Chainais. **Approche bayésienne non paramétrique dans l'apprentissage du dictionnaire pour adapter le nombre d'atomes**. *Conférence Nationale GRETSI*, 2015 [24].
5. Hong-Phuong Dang, Pierre Chainais. **Indian Buffet Process Dictionary Learning : algorithms and applications to image processing**. *International Journal of Approximate Reasoning* (en révision) [28].

1.7 Organisation du manuscrit

Ce manuscrit est organisé de la façon suivante :
Le chapitre 2 introduit les outils de base de la décomposition parcimonieuse. Quelques méthodes d'optimisation seront présentées. Les bases des méthodes d'échantillonnage aléatoire, appelées méthodes de Monte Carlo sont ensuite abordées.
Le chapitre 3 dresse un état de l'art des méthodes d'apprentissage de dictionnaire, et précise l'intérêt des approches bayésiennes, en particulier non paramétriques.
Le chapitre 4 donne une revue du processus fondateur de l'approche bayésienne non paramétrique, le processus de Dirichlet, et son utilisation pour les modèles de mélange. La liaison entre le processus de Dirichlet et les autres distributions non paramétriques comme la représentation de Backwell-Mac Queen ou le processus de restaurant chinois est aussi présentée.
Le chapitre 5 s'intéresse au processus de Buffet Indien (IBP), une distribution non paramétrique pour les modèles à caractéristiques latentes (*latent features*) qui est intéressante pour l'apprentissage de dictionnaire. Plusieurs façons de construire le processus de buffet indien seront décrites. Les modèles plus généraux de processus du buffet indien qui permettent de capturer le comportement données-atomes en loi de puissance et de reproduire des propriétés statistiques fines des données sont aussi évoqués.
Le chapitre 6 fournit différents algorithmes d'inférence de l'IBP dans le contexte d'un modèle linéaire gaussien.

Le chapitre 7 est consacré au modèle IBP-DL pour l'apprentissage de dictionnaire de taille adaptative en utilisant le processus de Buffet Indien. Différentes versions de l'échantillonnage de Gibbs pour l'inférence de ce modèle sont aussi étudiées.
Le chapitre 8 illustre la pertinence de l'approche IBP-DL sur des problèmes inverses en traitement d'image : débruitage, inpainting, compressive sensing. Les résultats obtenus sont comparés avec ceux d'autres méthodes de l'état de l'art.
Le chapitre 9 termine ce travail par un exposé de perspectives théoriques, algorithmiques et applicatives et une conclusion.

CHAPITRE 2

Décomposition parcimonieuse

Les représentations parcimonieuses permettent de résoudre les problèmes inverses où l'on est à la limite de l'identifiabilité. Elles connaissent de nombreux succès en traitement du signal et des images depuis ces vingt dernières années. Ce chapitre définit d'abord la notion de représentation parcimonieuse et les problèmes qui en résultent. On s'intéressera ensuite aux différents algorithmes qui permettent la représentation d'un signal par un dictionnaire éventuellement redondant en utilisant une approche par optimisation ou une méthode bayésienne.

2.1 Représentations parcimonieuses

L'objectif de tout changement de représentation est d'exprimer un signal sous une autre forme afin de pouvoir le traiter et l'analyser de façon plus simple, plus efficace. Représenter un signal de manière parcimonieuse consiste à le décomposer en utilisant un dictionnaire éventuellement redondant, où seul *un petit nombre d'atomes* est effectivement utilisé dans le dictionnaire. Un dictionnaire est dit redondant si sa taille est supérieure à la dimension de l'espace dans lequel vivent les données. Cette représentation parcimonieuse pour le signal comporte un grand nombre de valeurs nulles. Ce chapitre est consacré à la recherche d'une décomposition parcimonieuse d'un signal dans un dictionnaire connu. On étudie dans un premier temps un signal non bruité $\mathbf{y}$ que l'on souhaite décrire de la manière suivante :

$$\mathbf{y} = \mathbf{D}\mathbf{w} = \sum_{k=1}^{K} \mathbf{d}_k w_k \tag{2.1}$$

où $\mathbf{y}$ est un vecteur colonne de dimension L. $\mathbf{D}$ est la matrice de dictionnaire de taille $L \times K$ où K est le nombre d'atomes. Le vecteur colonne $\mathbf{w}$ de taille K encode la représentation parcimonieuse de l'observation $\mathbf{y}$. Le cœfficient w_k représente l'encodage du signal associé à l'atome numéro k.

La recherche d'une représentation parcimonieuse peut se traduire sous la forme d'un problème de minimisation sous contraintes :

$$\min_{\mathbf{w}} \|\mathbf{w}\|_0 \text{ sous contrainte } \mathbf{y} = \mathbf{D}\mathbf{w}. \tag{2.2}$$

La norme ℓ_0 sur $\mathbf{w}$ compte le nombre de coefficients non nuls dans $\mathbf{w}$. En minimisant le nombre de coefficients non-nuls, on cherche une représentation optimale $\mathbf{w}$ contenant un nombre minimal de termes non nuls, ce qui correspond au critère de parcimonie. Toutefois, la norme ℓ_0 est non convexe donc la solution obtenue peut correspondre à un minimum local. La minimisation exacte selon la norme ℓ_0 est

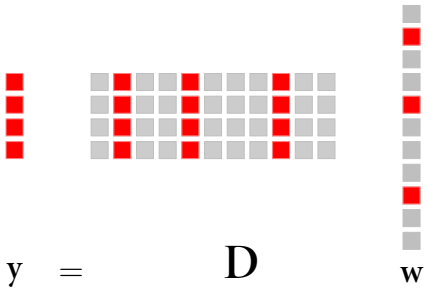


Figure 2.1 – **y** est éparse sur **D** : seuls quelques coefficients sont actifs (non nuls) dans **w**.

connue comme un problème d'optimisation NP-complet [29], *i.e.*, non calculable en temps polynomial sur un ordinateur. On doit effectuer une recherche exhaustive parmi toutes les combinaisons d'atomes différentes pour trouver la solution. C'est un problème qui passe difficilement à l'échelle. On s'intéressera dans la section suivante à différents algorithmes de recherche de décompositions parcimonieuses qui contournent la difficulté en réexprimant le problème (2.2), par exemple en relâchant la norme ℓ_0.

2.2 Algorithmes de poursuite adaptative pénalité ℓ_0

Les algorithmes de poursuite ou algorithmes gloutons *(greedy)* sont des algorithmes itératifs qui cherchent à résoudre le problème (2.2) grâce à une heuristique. Le principe de ces algorithmes est de chercher un par un les atomes actifs de façon à diminuer l'erreur entre le signal original et le signal parcimonieux. L'algorithme s'arrête quand un critère d'arrêt défini *a priori* est atteint. Cette optimisation locale ne peut pas assurer l'optimisation globale dans le cas général mais elle offre de bonnes propriétés de décroissance du résidu tout en garantissant la parcimonie.

2.2.1 Matching Pursuit

L'algorithme de Poursuite Adaptative (*Matching Pursuit, MP*), voir **Algo. 1**, est proposé par Mallat et Zhang en 1993 [30]. Le principe de la méthode est le suivant. D'abord, l'atome le plus corrélé avec le signal à l'itération courante est sélectionné, il s'agit de l'atome dont la contribution au résidu courant est la plus grande. Cette contribution est ensuite soustraite pour générer un nouveau résidu. Il est à noter qu'un atome k peut être sélectionné plusieurs fois durant la procédure. L'algorithme 1 présente l'algorithme de Matching Pursuit. Le critère d'arrêt est typiquement défini soit par le nombre maximal T d'itérations soit par le résidu $\|r^{(T)}\|_2$. En sortie d'algorithme, $\mathbf{y}$ s'exprime sous la forme : $\mathbf{y} = \sum_{t=1}^{T} w^{(t)}\mathbf{d}_k^{(t)} + \mathbf{r}^{(T)}$.

2.2.2 Orthogonal Matching Pursuit

L'agorithme de Poursuite Adaptative Orthogonale ou *Orthogonal Matching Pursuit (OMP)* [31], voir **Algo. 2**, est une extension de MP. Le principe de base de OMP reste le même que celui de MP. La seule différence est la mise à jour des cœfficients,

Initialisation : t $\leftarrow 0$; $\mathbf{r}^{(t)} \leftarrow \mathbf{y}$; $\mathbf{D}$
répéter
t$\leftarrow$ t+1 ;
Recherche de l'atome optimal :

$$k^{(t)} \leftarrow \underset{k}{\text{argmax}} \mid \langle \mathbf{r}^{(t-1)}, \mathbf{d}_k \rangle \mid$$

Mise à jour du coefficient : $w^{(t)} \leftarrow \mid \langle \mathbf{r}^{(t-1)}, \mathbf{d}_k^{(t)} \rangle \mid$
Mise à jour du résidu : $\mathbf{r}^{(t)} \leftarrow \mathbf{r}^{(t-1)} - \mathbf{d}_k^{(t)} w^{(t)}$
jusqu'au critère d'arrêt.

Algorithme 1 : Algorithme Matching Pursuit

Initialisation : t $\leftarrow 0$; $k^{(0)} \leftarrow 0$ $\mathbf{r}^{(0)} \leftarrow \mathbf{y}$; $\widetilde{\mathbf{D}}^{(0)} \leftarrow \varnothing$; $\mathbf{D}$
répéter
t$\leftarrow$ t+1 ;
Recherche de l'atome optimal :

$$\begin{cases} k^{(t)} \leftarrow \underset{k}{\text{argmax}} \mid \langle \mathbf{r}^{(t-1)}, \mathbf{d}_k \in \mathbf{D} \rangle \mid \\ \mathbf{d}_k \notin \widetilde{\mathbf{D}}^{(t-1)} \end{cases}$$

Ajout du nouvel atome au dictionnaire actif : $\widetilde{\mathbf{D}}^{(t)} \leftarrow [\widetilde{\mathbf{D}}^{(t-1)} \ \mathbf{d}_k^{(t)}]$
Mise à jour des coefficients : $\mathbf{w}^{(t)} \leftarrow \underset{\mathbf{w}}{\text{argmin}} \|\mathbf{y} - \widetilde{\mathbf{D}}^{(t)}\mathbf{w}\|_2^2$
Mise à jour du résidu : $r^{(t)} \leftarrow r^{(t-1)} - \widetilde{\mathbf{D}}^{(t)}\mathbf{w}^{(t)}$
jusqu'au critère d'arrêt.

Algorithme 2 : Algorithme Orthogonal Matching Pursuit

et donc du résidu. À chaque itération t, OMP recalcule tous les cœfficients en projetant le signal sur l'espace engendré par tous les atomes sélectionnés, contrairement à MP où seul le nouveau coefficient est mis à jour par projection sur le nouvel atome. La mise à jour de l'ensemble des coefficients est réalisée en utilisant la méthode des moindres carrés.

Le problème des moindres carrés $\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}} \|\mathbf{y} - \widetilde{\mathbf{D}}\mathbf{w}\|_2^2$ peut être résolu de la façon suivante : $\hat{\mathbf{w}} = \widetilde{\mathbf{D}}^{\dagger}\mathbf{y}$ avec $\widetilde{\mathbf{D}}^{\dagger} \leftarrow (\widetilde{\mathbf{D}}^{\mathrm{T}}\widetilde{\mathbf{D}})^{-1}\widetilde{\mathbf{D}}^{\mathrm{T}}$, aussi appelée la pseudo inverse de $\widetilde{\mathbf{D}}$. Il faut noter que contrairement à MP, la mise à jour les coefficients dans OMP est faite par la projection orthogonale du signal sur un sous espace (sous-dictionnaire $\widetilde{\mathbf{D}}$) de dimension de plus en plus grande. La projection interdit à un atome de $\mathbf{D}$ déjà sélectionné d'être choisi à nouveau pour l'ajouter dans $\widetilde{\mathbf{D}}$. OMP effectue de fait une réduction de dimension. Certains algorithmes basés sur OMP suggèrent une acceptation de plusieurs atomes à la fois afin d'accélérer le processus, par exemple Stagewise OMP (StOMP [32]), Regularized Orthogonal Matching Pursuit (ROMP [33]), voire le rejet d'atomes déjà sélectionnés comme Compressive Sampling Matching Pursuit (CoSAMP [34]).

2.3 Basis pursuit - Basis pursuit denoising pénalité ℓ_1

2.3.1 Basis pursuit

La norme ℓ_0 n'est pas convexe et pose des problèmes de minimum local dans un problème NP-complet. L'algorithme de poursuite de base ou *Basis pursuit* [6] approxime la norme ℓ_0 du problème (2.2) par une relaxation convexe de la norme ℓ_1, c'est-à-dire la somme des valeurs absolues des coefficients :

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1 \text{ sous contrainte } \mathbf{y} = \mathbf{D}\mathbf{w}. \tag{2.3}$$

Sous l'hypothèse que la solution est strictement parcimonieuse, l'équation (2.3) donne la solution de l'équation (2.2). La convexification du problème permet d'élargir le champs des méthodes de résolution : optimisation convexe, dualité lagrangienne... avec garantie d'une solution unique correspondant à un minimum global. Son avantage est d'être calculable par des techniques de programmation linéaire même s'il s'agit d'une solution approchée.

2.3.2 Basis pursuit denoising

La décomposition du signal dans l'équation (2.1) ne tient pas en compte de l'existence du bruit. L'équation (2.3) permet de résoudre le problème posé dans le modèle (2.1) sous hypothèses de parcimonie stricte. On modélise maintenant la perturbation d'un signal par un bruit

$$\mathbf{y} = \mathbf{D}\mathbf{w} + \boldsymbol{\varepsilon}. \tag{2.4}$$

On doit par conséquent reformuler le problème ainsi que la contrainte. Le bruit est généralement supposé gaussien centré i.i.d.. Ceci équivaut à choisir une erreur de reconstruction quadratique. L'idée est de rechercher une approximation parcimonieuse d'un signal à une petite erreur ϵ près. Ce problème s'appelle Basis Pursuit DeNoising (BPDN) [6]. BPDN est une version de BP (voir partie 2.3.1) qui tolère du bruit. On cherche maintenant à résoudre le problème suivant :

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1 \text{ sous contrainte } \|\mathbf{y} - \mathbf{D}\mathbf{w}\|_2^2 \leq \epsilon. \tag{2.5}$$

Quand $\epsilon = 0$, ce problème revient à Basis Pursuit.

2.4 Programmation quadratique : LASSO

BPDN (*Basis Pursuit Denoising*) est bien connu sous le nom de LASSO [5] (*Least Absolute Shrinkage and Selection Operator*), après avoir été introduit par Tibshirani en 1996. Il permet de minimiser l'erreur quadratique, avec une borne sur la somme des valeurs absolues des coefficients. Le problème (2.5) peut aussi être posé sous la forme suivante :

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{D}\mathbf{w}\|_2^2 \text{ sous contrainte } \|\mathbf{w}\|_1 \leq \delta \tag{2.6}$$

ou encore à l'aide du multiplicateur de Lagrange qui permet d'indiquer l'influence de la contrainte dans le coût de la solution. On rappelle ici le principe de cette approche :

Définition 1. *Soit le problème d'optimisation* $\min_{x\in\mathbb{R}^d} f(x)$ *sous contraintes* $h(x) = 0$ *et* $g(x) \leqslant 0$*, et soient les vecteurs* $\mu \in \mathbb{R}^m$ *et* $\lambda \in \mathbb{R}^n_+$*. La fonction* $\mathcal{L} : \mathbb{R}^{d+n+m} \to \mathbb{R}$ *:*

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \sum_{i=1}^{m} \mu_i h_i(x) + \sum_{i=1}^{n} \lambda_i g_i(x)$$

est appelée Lagrangien ou fonction lagrangienne du problème.

Définition 2. *Soit le problème d'optimisation* $\min_{x\in\mathbb{R}^d} f(x)$ *sous contraintes* $h(x) = 0$ *et* $g(x) \leqslant 0$*, et soient les vecteurs* $\mu \in \mathbb{R}^m$ *et* $\lambda \in \mathbb{R}^n_+$*. La fonction* $\phi : \mathbb{R}^{m+n} \to \mathbb{R}$ *définie par :*

$$\phi(\mu, \lambda) = \min_{x\in\mathbb{R}^d} \mathcal{L}(x, \mu, \lambda)$$

est la fonction duale du problème. Les paramètres λ *et* μ *sont appelés variables duales. Dans ce contexte, les variables* x *sont appelées variables primales.*

Par exemple, l'équation (2.6) peut s'écrire sous la forme :

$$\begin{aligned} &\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{D}\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_1 \\ &\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{D}\mathbf{w}\|_2^2 + \lambda \sum_{k=1}^{K} |w_k| \end{aligned} \tag{2.7}$$

avec $\lambda > 0$. Il existe des valeurs de λ dans eq. (2.7), ϵ dans eq. (2.5), δ dans eq. (2.6) qui sont compatibles avec la même solution.

Dans le problème LASSO, λ contrôle la parcimonie de $\mathbf{w}$. Soit $\{w_{lasso}[k]\}_{k=1:K}$ les solutions du problème LASSO. Plus λ augmente plus les $w_{lasso}[k]$ tendent vers 0 et pour un λ suffisamment grand, certains sont exactement égaux à zéro. Le choix du paramètre de régularisation λ en fonction du niveau de bruit $\boldsymbol{\varepsilon}$ est important. Ce choix influence le résultat final. Comme la norme ℓ_1 n'est pas différentiable en 0, la fonction objectif du LASSO n'est pas différentiable. Différents algorithmes ont été développés afin de trouver les solutions en se basant sur la convexité du problème d'optimisation.

L'algorithme *Iterative Soft Thresholding* [35] est un algorithme de seuillage itératif reposant sur le seuillage doux qui permet de résoudre le problème (2.7). Notons au passage que l'opérateur de seuillage doux est un opérateur de proximité [36] associé à la norme ℓ_1. L'opérateur de seuillage dur est associé à la norme ℓ_0.

Certains algorithmes reposent sur une approximation à λ fixé. On parle des algorithmes de descente coordonnée par coordonnée qui approchent la solution en ne modifiant qu'une seule composante de $\mathbf{w}_{lasso}$ à la fois. Par exemple, on a l'algorithme *Shooting* [37] proposé par Fu (1998), ou l'algorithme de seuillage itératif [38], ou encore Pathwise Coordinate Optimization [39] proposé par Friedman, Hastie, Höfling, et Tibshirani. La descente de coordonnées circulaire [40] a été aussi proposée dans la littérature. On peut également citer les méthodes de points intérieurs [6].

2.5 Algorithme LARS

D'autres méthodes se regroupent dans la famille des algorithmes d'homotopie qui fournissent les solutions du LASSO pour chaque valeur de son paramètre. Elles considèrent λ en tant que paramètre d'homotopie. Parmi ces algorithmes, on retrouve notamment le Least-Angle Regression (LARS) [41] que l'on présente ici. Si l'on construit le chemin de solutions de $\mathbf{w}$ en fonction du paramètre λ, on s'aperçoit que la solution LASSO est linéaire par morceaux. Ainsi, on peut construire une suite de $T+1$ valeurs croissantes de $\lambda_0 = 0 < \lambda_1 < \ldots \lambda_T$ telles que $\hat{\mathbf{w}}_{\lambda_0}$ soit l'estimateur des moindres carrées, $\hat{\mathbf{w}}_{\lambda_1}$ ait exactement un coefficient nul, $\hat{\mathbf{w}}_{\lambda_2}$ deux coefficients nuls, etc. jusqu'à $\hat{\mathbf{w}}_{\lambda_T}$ ne contienne que les coefficients nuls. La détermination de cette suite de valeurs de λ et des $\mathbf{w}$ correspondant se fait par l'algorithme Lars.

L'algorithme Least Angle Regression (LARS) (Algorithme de LARS pour la méthode LASSO) [41] est un algorithme d'homotopie itératif forward qui permet de fournir rapidement l'ensemble de toutes les solutions LASSO

$$\{\mathbf{w}_{lasso}^{\lambda_t} \mid \lambda_t \geqslant 0\}_{t=1:T}. \tag{2.8}$$

Définition 3. *Soit $\Delta(\lambda_t)$, l'ensemble des indices des atomes sur le niveau de λ_t*

$$\Delta(\lambda_t) = \left\{k \in \{1;\ldots,K\} \text{ t.q. } \mid \boldsymbol{d}_k^T(\boldsymbol{y} - \boldsymbol{D}\boldsymbol{w}_{lasso}^{\lambda_t}) \mid = \lambda_t\right\}. \tag{2.9}$$

Le principe de l'algorithme LARS s'appuie sur le maximum de corrélation entre les atomes $\mathbf{d}_k$ et le résidu $\mathbf{y} - \mathbf{D}\mathbf{w}_{lasso}^{\lambda_t}$. On décrit ci-dessous l'idée de l'algorithme LARS pour résoudre le problème LASSO.

- on commence avec $\lambda_0 = \infty$. Dans ce cas le vecteur $\hat{\mathbf{w}} = 0^K$. Par conséquent, le résidu vaut $\mathbf{y}$. On identifie l'atome $\mathbf{d}_k$ ayant la plus grande corrélation avec $\mathbf{y}$. On ajoute k à la liste des atomes retenus.
- l'étape 1 : On décroît λ_1 vers 0 jusqu'à ce qu'il existe un $\mathbf{d}_k$ tel que k rejoigne $\Delta(\lambda_1)$.
- l'étape $t < T$: On continue de décroître λ_t vers 0. L'ensemble $\Delta(\lambda_t)$ peut être changé par rapport à $\Delta(\lambda_{t-1})$ car :
  - un atome k qui n'appartenait pas avant à $\Delta(\lambda_{t-1})$ peut rejoindre maintenant $\Delta(\lambda_t)$, par conséquent, $w_{lasso}^{\lambda_t}[k] \neq 0$.
  - un atome k qui appartenait à $\Delta(\lambda_{t-1})$ mais n'appartient pas à $\Delta(\lambda_t)$, par conséquent, $w_{lasso}^{\lambda_t}[k] = 0$.
- l'étape T : L'algorithme s'arrête quand $\lambda_T = 0$. Le problème LASSO pour tout λ est alors résolu.

L'algorithme LARS explore l'ensemble des solutions de manière itérative, en explorant successivement les atomes les plus corrélés avec le résidu courant. Cependant, en cas de présence de groupe d'atomes corrélés, l'algorithme va avoir tendance à sélectionner un atome par groupe, lésant l'exploration des solutions.

2.6 Décomposition parcimonieuse et méthodes de Monte Carlo

On peut aussi poser le problème dans le cadre probabiliste. Dans cette partie, les méthodes de Monte Carlo par chaînes de Markov [42] (MCMC) sont présentées pour résoudre le problème.

Initialisation : t $\leftarrow 0$; $\theta^{(0)} \sim p_0(\theta)$;
pour $t = 1 : T$ **faire**
  t$\leftarrow$ t+1 ;
  $\theta^{(t)} \sim \mathcal{K}(\theta \mid \theta^{(t-1)})$;
fin pour

Algorithme 3 : Algorithme MCMC générique.

2.6.1 Principe des algorithmes MCMC

Les méthodes MCMC sont des méthodes qui permettent d'échantillonner selon une distribution de probabilité $\mathrm{p}(.)$ donnée. Ces méthodes se basent sur la construction d'une chaîne de Markov $\{\theta^{(t)}\}_{t=1:T}$ qui est asymptotiquement distribuée selon $\mathrm{p}(\theta)$. On introduit tout d'abord un échantillon initial $\theta^{(0)}$ distribué suivant $\mathrm{p}_0(\theta)$. L'échantillonnage sera ensuite effectué en se basant sur un noyau Markovien appelé aussi noyau de transition [42] $\mathcal{K}(\theta \mid \theta')$. Ce noyau $\mathcal{K}(\theta \mid \theta')$ doit respecter certaines propriétés [42]. En particulier, le noyau est construit de telle façon que la distribution invariante de la chaîne soit la distribution cible $\mathrm{p}(\theta)$

$$\int_{\Theta} \mathcal{K}(\theta \mid \theta')\mathrm{p}(\theta')\mathrm{d}\theta' = \mathrm{p}(\theta), \forall\theta \in \Theta. \tag{2.10}$$

L'algorithme 3 donne un algorithme MCMC générique.

La densité empirique formée par les échantillons aléatoires $\theta^{(t)}$ converge vers la distribution cible p au fur et à mesure que t augmente. Ces échantillons sont ensuite utilisés pour approximer des estimateurs. En pratique, on exclue d'abord une première étape dite de chauffe (*burn-in*) avant de considérer que l'algorithme a atteint un régime stationnaire où l'on a bien $\theta \sim \mathrm{p}$.

2.6.2 Inférence bayésienne pour une décomposition parcimonieuse

Dans le cadre bayésien, le problème de décomposition parcimonieuse s'écrit sous la forme d'une loi *a posteriori* :

$$\mathrm{p}(\mathbf{w} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon) \propto \mathrm{p}(\mathbf{y} \mid \mathbf{D}, \mathbf{w}, \sigma_\varepsilon)\mathrm{p}(\mathbf{w}). \tag{2.11}$$

La vraisemblance $\mathrm{p}(\mathbf{y} \mid \mathbf{D}, \mathbf{w}, \sigma_\varepsilon)$ est construite conformément au modèle (2.4). Le bruit ε qui perturbe le signal est généralement gaussien i.i.d. de variance σ_ε^2. La vraisemblance est alors une loi Normale d'espérance $\mathbf{Dw}$ et de variance σ_ε^2 :

$$\mathrm{p}(\mathbf{y} \mid \mathbf{D}, \mathbf{w}, \sigma_\varepsilon) = \frac{1}{(2\pi\sigma_\varepsilon^2)^{P/2}} \exp\left\{-\frac{1}{2\sigma_\varepsilon^2}\mathrm{tr}\left[(\mathbf{y}-\mathbf{Dw})^{\mathrm{T}}(\mathbf{y}-\mathbf{Dw})\right]\right\}. \tag{2.12}$$

Pour rappel, la dimension de $\mathbf{y}$ est L. La distribution $\mathrm{p}(\mathbf{w})$ s'appelle la loi *a priori* .

Pour résoudre le problème en utilisant les méthodes MCMC, on cherche à échantillonner $\mathbf{w}$ selon $\mathrm{p}(\mathbf{w} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon)$. Ces échantillons sont ensuite utilisés pour approximer des estimateurs bayésiens tels que le maximum *a posteriori* (MAP) ou l'espérance *a posteriori* du minimum de l'erreur quadratique moyenne (minimum mean-square error MMSE).

En particulier, on définit l'estimateur MAP comme étant le maximum *a posteriori* du paramètre, donné par :

$$\max_{\mathbf{w}} \mathrm{p}(\mathbf{w} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon) \tag{2.13}$$

$$\textit{ou bien} \quad \max_{\mathbf{w}} \mathrm{p}(\mathbf{y} \mid \mathbf{D}, \mathbf{w}, \sigma_\varepsilon)\mathrm{p}(\mathbf{w}) \tag{2.14}$$

ou encore

$$\max_{\mathbf{w}} \log \mathrm{p}(\mathbf{y} \mid \mathbf{D}, \mathbf{w}, \sigma_\varepsilon) + \log \mathrm{p}(\mathbf{w}). \tag{2.15}$$

qui est équivalent au problème :

$$\min_{\mathbf{w}} - \log \mathrm{p}(\mathbf{y} \mid \mathbf{D}, \mathbf{w}, \sigma_\varepsilon) - \log \mathrm{p}(\mathbf{w}). \tag{2.16}$$

En comparant (2.16) avec l'expression des méthodes d'optimisation lorsque $\mathrm{p}(\mathbf{w}) \propto \exp(-\lambda\|\mathbf{w}\|_p)$, on l'opérateur MAP est ici donné par :

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{D}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_p, \tag{2.17}$$

on remarque que la loi *a priori* joue le rôle de la régularisation qui permet de favoriser la parcimonie. Dans la suite, différents types de loi *a priori* pour les coefficients parcimonieux seront présentés.

Nous rappelons maintenant les deux algorithmes les plus couramment utilisés en MCMC, que l'on retrouvera dans les autres chapitres :

— l'échantillonneur de Gibbs, basé sur des échantillonnages conditionnels,
— l'algorithme Metropolis-Hastings, basé sur une procédure d'acceptation-rejet.

2.6.3 Échantillonneur de Gibbs

L'échantillonneur de Gibbs permet de générer un ensemble de variables aléatoires suivant une loi cible sans utiliser directement son expression supposée difficile à manipuler, mais en utilisant *les densités conditionnelles* de chacune des variables aléatoires [42]. Dans notre cas, on cherche à échantillonner $\mathbf{w}$ qui contient K coefficients. Il est difficile de générer d'un bloc $\mathbf{w}$ suivant la loi cible $\mathrm{p}(\mathbf{w} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon)$. L'algorithme 4 présente l'échantillonnage de Gibbs de $\mathrm{p}(\mathbf{w} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon)$ en générant alternativement chaque élément w_k selon la distribution conditionnelle $\mathrm{p}(w_k \mid \mathbf{y}, \mathbf{D}, \mathbf{w}_{\neq k}, \sigma_\varepsilon)$.

Pour un nombre d'itérations suffisamment grand, chaque vecteur $\mathbf{w}$ généré peut être considéré comme étant une réalisation de la loi *a posteriori* $\mathrm{p}(\mathbf{w} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon)$. Ces échantillons sont ensuite utilisés pour construire des estimateurs.

2.6.4 Algorithme Metropolis-Hastings

L'algorithme Metropolis-Hastings (MH) appartient aussi à la famille des méthodes MCMC. La première version de l'algorithme a été introduite en 1953 [43] en considérant le cas particulier de la distribution de Boltzmann. Cette première version a été beaucoup utilisée pour traiter des problèmes en physique statistique. En 1970, l'algorithme a été généralisé par Hastings [44] au cas de n'importe quelle distribution.

Initialisation : t $\leftarrow 0$; $\mathbf{w}^{(0)} \sim p_0(\mathbf{w})$;
répéter
  t$\leftarrow$ t+1 ;
  pour $k = 1 : K$ **faire**
    $w_k^{(t)} \sim \mathrm{p}(w_k \mid \mathbf{D}, w_1^{(t)}, \ldots, w_{k-1}^{(t)}, w_{k+1}^{(t-1)}, \ldots, w_K^{(t-1)}, \sigma_\varepsilon)$;
  fin pour
jusqu'au critère d'arrêt.

Algorithme 4 : Échantillonnage de Gibbs de $\mathrm{p}(\mathbf{w} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon)$.

Initialisation : t $\leftarrow 0$; $\mathbf{w}^{(0)} \sim \mathrm{p}_0(\mathbf{w})$
répéter
  t$\leftarrow$ t+1 ;
  $\widetilde{\mathbf{w}} \sim \mathrm{q}(\widetilde{\mathbf{w}} \mid \mathbf{w}^{(t-1)})$;
  $a = \min\left(1, \dfrac{\mathrm{p}(\widetilde{\mathbf{w}} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon)}{\mathrm{p}(\mathbf{w}^{(t-1)} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon)} \dfrac{\mathrm{q}(\mathbf{w}^{(t-1)} \mid \widetilde{\mathbf{w}})}{\mathrm{q}(\widetilde{\mathbf{w}} \mid \mathbf{w}^{(t-1)})}\right)$
  si $a > \mathcal{U}_{[0,1]}$ **alors**
    $\mathbf{w}^{(t)} \rightarrow \widetilde{\mathbf{w}}$;
  sinon
    $\mathbf{w}^{(t)} \rightarrow \mathbf{w}^{(t-1)}$;
  fin si
jusqu'au critère d'arrêt.

Algorithme 5 : L'algorithme Metropolis-Hastings avec une loi de proposition indépendante des échantillons précédents.

La loi cible peut être connue à une constante multiplicative près, ce qui est souvent le cas lorsque l'on calcule la densité *a posteriori* à l'aide du théorème de Bayes. Si l'on ne sait pas simuler suivant cette loi, l'algorithme MH propose d'introduire à partir de la loi cible un *noyau de transition arbitraire* que l'on appelle *loi instrumentale* ou aussi *loi de proposition* à partir de laquelle on sait échantillonner. Le support de la loi de proposition doit inclure le support de la loi cible, c'est-à-dire que la loi de proposition ne doit pas être nulle sur le support de la loi cible.

L'algorithme 5 synthétise les étapes d'échantillonnage de $\mathbf{w}$ par Metropolis-Hastings. Si $\mathbf{w}$ suit une loi cible $\mathrm{p}(\mathbf{w} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon)$, on introduit une loi instrumentale conditionnelle $\mathrm{q}(\widetilde{\mathbf{w}} \mid \mathbf{w})$.

A l'instant (t), on simule un échantillon aléatoire $\widetilde{\mathbf{w}}$ selon la loi de proposition. On l'accepte ou le rejette à l'aide d'une procédure d'acceptation-rejet avec une probabilité de mouvement a. La loi de proposition peut être symétrique *i.e.* $\mathrm{q}(\widetilde{\mathbf{w}} \mid \mathbf{w}) = \mathrm{q}(\mathbf{w} \mid \widetilde{\mathbf{w}})$. Dans ce cas, l'expression de la probabilité de mouvement devient le rapport des lois *a posteriori* :

$$a = \min\left(1, \frac{\mathrm{p}(\widetilde{\mathbf{w}} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon)}{\mathrm{p}(\mathbf{w}^{(t-1)} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon)}\right). \tag{2.18}$$

On peut également choisir une loi de proposition indépendante des échantillons précédents où $\mathrm{q}(\widetilde{\mathbf{w}} \mid \mathbf{w}) = \mathrm{q}(\widetilde{\mathbf{w}})$. L'expression de la probabilité de mouvement est

dans ce cas :

$$a = \min\left(1, \frac{\mathrm{p}(\widetilde{\mathbf{w}} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon)}{\mathrm{p}(\mathbf{w}^{(t-1)} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon)} \frac{\mathrm{q}(\mathbf{w}^{(t-1)})}{\mathrm{q}(\widetilde{\mathbf{w}})}\right). \tag{2.19}$$

Remarque : *La chaîne peut prendre plusieurs fois la même valeur, les échantillons successifs ne sont pas i.i.d..*

2.6.5 Intérêt de l'inférence bayésienne

Dans les méthodes d'optimisation, par exemple basis pursuit denoising (*cf.* 2.4), le choix du niveau de bruit (erreur de reconstruction) est important afin de choisir un paramètre de régularisation λ compatible. Dans le cadre bayésien, nous pouvons intégrer la variance du bruit aux variables du modèle. Non seulement $\mathbf{w}$ peut être échantillonné, mais aussi le niveau de bruit en considérant une loi *a priori* . Un bruit gaussien centré i.i.d. est souvent considéré pour le modèle (2.4). Échantillonner le niveau de bruit peut se comprendre comme échantillonner la variance σ_ε^2. La variance du bruit σ_ε^2 est maintenant considérée comme une variable aléatoire de loi *a priori* $\mathrm{p}(\sigma_\varepsilon^2)$. Le choix des lois *a priori* sera présenté dans la prochaine partie 2.6.6. La loi *a posteriori* s'écrit maintenant

$$\mathrm{p}(\mathbf{w}, \sigma_\varepsilon \mid \mathbf{y}, \mathbf{D}) \propto \mathrm{p}(\mathbf{y} \mid \mathbf{D}, \mathbf{w}, \sigma_\varepsilon)\mathrm{p}(\mathbf{w}, \sigma_\varepsilon^2). \tag{2.20}$$

En utilisant l'échantillonnage de Gibbs pour l'inférence par exemple, le problème peut être résolu en échantillonnant alternativement selon

$$\mathrm{p}(\mathbf{w} \mid \mathbf{y}, \mathbf{D}, \sigma_\varepsilon) \propto \mathrm{p}(\mathbf{y} \mid \mathbf{D}, \mathbf{w}, \sigma_\varepsilon)\mathrm{p}(\mathbf{w}) \tag{2.21}$$

$$\mathrm{p}(\sigma_\varepsilon^2 \mid \mathbf{y}, \mathbf{D}, \mathbf{w}) \propto \mathrm{p}(\mathbf{y} \mid \mathbf{D}, \mathbf{w}, \sigma_\varepsilon)\mathrm{p}(\sigma_\varepsilon^2). \tag{2.22}$$

2.6.6 Choix de la loi *a priori*

La loi *a priori* est le pendant Bayésien de la régularisation. Le choix peut d'abord être motivé par une connaissance préalable du phénomène étudié. Ce choix peut également résulter d'intérêts calculatoires, comme dans le cas des lois conjuguées. Dans le cas où aucune connaissance préalable du système n'est disponible, on peut vouloir choisir des lois les plus *non informatives* possibles. C'est par exemple le cas des lois *a priori* de Jeffreys [45] qui sont construites en minimisant un critère d'information, dit de Fischer. En particulier, le choix de lois non informatives peut amener à considérer des lois *a priori* impropres. Cependant il faut rester vigilant quant-au fait que la loi *a posteriori* doit être propre.

L'inférence bayésienne s'appuie le plus souvent sur la loi *a posteriori* . Pour faciliter le calcul de $\mathrm{p}(\sigma_\varepsilon^2 \mid \mathbf{y}, \mathbf{D}, \mathbf{w})$, la loi *a priori* $\mathrm{p}(\sigma_\varepsilon^2)$ est choisie de sorte que la loi *a posteriori* et la loi *a priori* aient la même forme. On parle de lois conjuguées. Cette stratégie sera souvent utilisée dans la suite, d'où l'importance de la famille exponentielle. On suppose que le bruit est gaussien, centré. La loi conjuguée de sa variance σ_ε^2 est une loi Inverse Gamma. La table 2.1 rappelle quelques autres exemples de lois conjuguées.

Vraisemblance $\mathrm{p}(x \mid \theta)$	Loi *a priori* $\mathrm{p}(\theta)$	Loi *a posteriori* $\mathrm{p}(\theta \mid x)$
$\mathcal{N}(\theta, \sigma^2)$	$\mathcal{N}(\mu; \upsilon^2)$	$\mathcal{N}(x/\sigma^2 + \mu/\upsilon^2; (1/\sigma^2 + 1/\upsilon^2)^{-1})$
Gamma(n, θ)	Gamma$(\alpha; \beta)$	Gamma$(\alpha + n, \beta + x)$
Binomial (n, θ)	Beta$(\alpha; \beta)$	Beta$(\alpha + n, \beta + x)$
Poisson(θ)	Gamma$(\alpha; \beta)$	Gamma$(\alpha + x, \beta + 1)$

TABLE 2.1 – Exemples de lois conjuguées.

2.6.7 Les lois *a priori* pour des coefficients parcimonieux

Cette partie présente des exemples de choix de la loi *a priori* sur la représentation $\mathbf{w}$ pour favoriser la parcimonie.

2.6.7.1 Loi *a priori* impropre

Dans l'équation (2.2), pour favoriser la parcimonie, le problème posé utilise la norme ℓ_0 sur les coefficients. Quel type de loi *a priori* faut-il choisir sur les coefficients pour correspondre à la norme ℓ_0 ? Dans la partie 2.6.2, on a aussi vu que, l'opposé du logarithme de la loi *a priori* est équivalent à la partie régularisation dans les méthodes d'optimisation (voir eq. (2.16) et eq. (2.17)). On peut imaginer chercher une probabilité $\mathrm{p}(\mathbf{w})$ vérifiant :

$$-\log \mathrm{p}(\mathbf{w}) = \lambda \|\mathbf{w}\|_0. \tag{2.23}$$

Rappelons que $\mathbf{w}$ est un vecteur de taille K. La potentielle loi $\mathrm{p}(\mathbf{w}) \propto \exp(-\lambda\|\mathbf{w}\|_0)$ est bien une mesure positive bornée à 1, mais son intégrale n'est pas définie :

$$\int_{\mathbb{R}^K} \mathrm{p}(\mathbf{w}) d\mathbf{w} \geqslant \int_{\mathbb{R}^K} \exp(-\lambda K) \mathrm{d}\mathbf{w} = \infty \tag{2.24}$$

Ce n'est donc pas une mesure de probabilité. C'est ce type de mesure que l'on définit comme loi *a priori* impropre. L'utilisation de la loi impropre peut empêcher l'inférence par des méthodes MCMC. Par contre, comme présenté dans 2.6.2, l'inférence bayésienne se base sur la loi *a posteriori* , qui n'a généralement besoin d'être connu qu'à une constante multiplicative près. Il suffit ainsi que la loi *a posteriori* soit propre pour pouvoir utiliser les méthodes MCMC. Dans notre cas, la vraisemblance $\mathrm{p}(\mathbf{y} \mid \mathbf{D}, \mathbf{w}, \sigma_\varepsilon)$ est construite conformément au modèle (2.4) où le bruit est généralement supposé gaussien i.i.d. La vraisemblance est donc une loi normale d'espérance $\mathbf{Dw}$ et de variance σ_ε^2. On peut alors montrer que

$$\int_{\mathbb{R}^K} \mathrm{p}(\mathbf{y} \mid \mathbf{D}, \mathbf{w}, \sigma_\varepsilon) \mathrm{p}(\mathbf{w}) d\mathbf{w} < +\infty. \tag{2.25}$$

Par conséquent la loi *a posteriori* $\mathrm{p}(\mathbf{w} \mid \mathbf{y}, \mathbf{D}, \mathbf{w}, \sigma_\varepsilon) \propto \mathrm{p}(\mathbf{y} \mid \mathbf{D}, \mathbf{w}, \sigma_\varepsilon)\mathrm{p}(\mathbf{w})$ est propre.

2.6.7.2 Loi de Laplace

La norme ℓ_0 de $\mathbf{w}$ est souvent remplacée par la norme ℓ_1 dans les méthodes d'optimisation. Du point de vue probabiliste, l'utilisation de la norme ℓ_1 correspond à la

log-vraisemblance de coefficients donc à une loi *a priori* Laplacienne. Le vecteur $\mathbf{w}$ contenant K éléments est supposé laplacien i.i.d. : $\mathbf{w} \sim \mathcal{L}aplace(0, \frac{1}{\lambda})$.

$$\mathrm{p}(\mathbf{w}) = \prod_{k=1}^{K} \mathrm{p}(w_k) \tag{2.26}$$

avec $\mathrm{p}(w_k) = \frac{\lambda}{2} \exp(-\lambda \mid w_k \mid)$. Le neg-logarithme de $\mathrm{p}(\mathbf{w})$ obtenu est donc :

$$-\log \mathrm{p}(\mathbf{w}) \propto \lambda \sum_{k=1}^{K} |w_k| \propto \lambda \|\mathbf{w}\|_1. \tag{2.27}$$

2.6.7.3 Mélange de gaussiennes

Les modèles probabilistes permettent de proposer de nombreuses approches. Un exemple est une loi *a priori* de type *mélange de gaussiennes* qui a été proposé par Olshausen et Millman [46] pour modéliser $\mathbf{w}$.

$$\mathrm{p}(\mathbf{w} \mid \pi) = \prod_{k=1}^{K} \pi_k \mathcal{N}_k(0, \sigma_{1,k}^2) + (1 - \pi_k)\mathcal{N}_k(0, \sigma_{2,k}^2) \tag{2.28}$$

où l'indicateur binaire $\pi_k \in \{0, 1\}$ indique le choix entre les deux distributions gaussiennes : $\mathcal{N}_k(0, \sigma_{1,k}^2)$ et $\mathcal{N}_k(0, \sigma_{2,k}^2)$ avec $\sigma_{1,k}^2 \gg \sigma_{2,k}^2$. La première gaussienne a une grande variance afin de modéliser les coefficients de poids forts tandis que les coefficients de poids faibles sont modélisés par la deuxième gaussienne de petite variance. Les poids π_k favorisent d'autant plus la parcimonie qu'ils sont proches de 0.

2.6.7.4 Modèle Bernoulli-gaussien

En faisant tendre $\sigma_{2,k}^2$ vers 0, on obtient le modèle Bernoulli-gaussien qui régularise un *a priori* de type gaussien sur les coefficients en ajoutant une masse ponctuelle en zéro par l'intermédiaire d'une loi Bernoulli. Soit $\mathbf{w} = (\mathbf{z}, \mathbf{s})$, une variable aléatoire sur $\{0, 1\} \times \mathbb{R}$. $\mathbf{z}$ est une variable Bernoulli i.i.d de paramètre π.

$$P(\mathbf{z} = 1) = 1 - P(\mathbf{z} = 0) = \pi. \tag{2.29}$$

$\mathbf{s}$ est une variable gaussienne i.i.d. centrée et de variance $\sigma_{\mathbf{s}}^2$.

$$\mathbf{s} \sim \mathcal{N}(0, \sigma_{\mathbf{s}}^2 \mathbb{I}_K). \tag{2.30}$$

$\mathbf{z}$ et $\mathbf{s}$ sont des vecteurs de taille K et $\mathbf{w} = \mathbf{z} \odot \mathbf{s}$ et $\odot$ est le produit terme à terme. Quand $z_k = 1$, le coefficient w_k utilisé pour coder le signal sera s_k, sinon w_k sera 0 quand $z_k = 0$, d'où :

$$\mathrm{p}(\mathbf{w} \mid \pi, \sigma_{\mathbf{s}}^2) = \frac{1}{(2\pi\sigma_{\mathbf{s}}^2)^{K/2}} \exp\left\{ -\frac{1}{2\sigma_{\mathbf{s}}^2} \mathrm{tr}\left[\mathbf{s}^{\mathrm{T}}\mathbf{s}\right] \right\} \prod_{k=1}^{K} (\pi_k)^{z_k} (1 - \pi_k)^{z_k}. \tag{2.31}$$

Ce modèle servira de point de départ au modèle IBP-DL présenté dans le chapitre 7.

Remarque : *On peut aussi ajouter π dans le modèle en ajoutant un loi a priori , par exemple une loi Bêta conjuguée. Dans ce cas, on regarde la loi jointe a posteriori de $\mathbf{w}$ et π.*

2.7 Discussion

Il existe aussi d'autres méthodes de décomposition et des méthodes hybrides basées sur les méthodes ci-dessus que l'on ne présente pas dans ce document. Ce chapitre n'est qu'un aperçu global des représentations parcimonieuses. Comme discuté dans l'introduction, le dictionnaire $\mathbf{D}$ a une très grande influence sur la qualité de la décomposition et la parcimonie du signal. En effet, si le dictionnaire ne contient pas du tout ou très peu d'atomes adaptés aux structures présentes dans le signal, la décomposition sera mauvaise quelle que soit la méthode choisie. Réciproquement, lors de l'apprentissage d'un dictionnaire, l'efficacité de la procédure d'apprentissage pourra être impactée par la méthode de décomposition parcimonieuse choisie.

Un objectif important de cette thèse est de contribuer à répondre à la question fondamentale suivante : Comment construire un dictionnaire optimal ? Cette notion d'optimalité doit être définie par rapport à un critère. Dans cette thèse, le critère est d'obtenir la meilleure reconstruction dans les problèmes inverses notamment en traitement d'image. On évalue la pertinence des dictionnaires dans le chapitre 8 en étudiant leur performances de reconstruction d'images détériorées.

CHAPITRE 3

Apprentissage de Dictionnaire

Le chapitre précédent a présenté plusieurs méthodes d'estimation parcimonieuse des cœfficients de décomposition sur un dictionnaire donné. Mais au delà du critère et de l'algorithme utilisés, le choix du dictionnaire conditionne aussi la qualité et le niveau de parcimonie d'une décomposition. Dans ce chapitre, en plus de l'estimation des cœfficients de la décomposition, nous nous intéressons à la façon de construire le dictionnaire favorisant au mieux la parcimonie. Cette étape est appelée apprentissage de dictionaire, où *Dictionnairy Learning (DL)* en anglais.

3.1 Analyse en Composantes

Les méthodes dites d'Analyse en Composantes sont un recueil de techniques qui consiste à *apprendre une base* adaptée à un jeu de données selon un critère. Ces méthodes sont en particulier utilisées pour faire de la réduction de la dimension.

3.1.1 Analyse en Composantes Principales

L'Analyse en Composantes Principales (ACP) ou en anglais *Principal Component Analysis (PCA)* [47] est un outil classique de traitement de signal qui consiste à apprendre *une base orthonormale* à partir des données. Cette méthode est généralement suivie d'une étape de réduction de la dimension de ces données. Bien que l'ACP nous donne une *base orthonormale*, sans lien avec la parcimonie, elle fournit une première notion de l'apprentissage de dictionnaire. Les coefficients sont obtenus par la projection des données sur cette base.

Soit $\mathbf{Y} \in \mathbb{R}^{L\times N}$, un nuage de N points dans un espace de dimension L. L'ACP consiste à projeter ces points sur un sous-espace à K dimensions (avec $K \leq L$) choisi de façon à minimiser l'erreur de reconstruction quadratique. Soit une matrice de données centrées $\mathbf{Y}$. Sa matrice de covariance empirique est $\mathbf{\Sigma} = \frac{1}{N}\mathbf{Y}\mathbf{Y}^{\mathrm{T}} \in \mathbb{R}^{L\times L}$. Nous voulons projeter linéairement $\mathbf{y}_i \in \mathbb{R}^L$ *i.e.*

$$\mathbf{w}_i = \mathbf{D}^{\mathrm{T}}\mathbf{y}_i \text{ sous contrainte } \mathbf{D}^{\mathrm{T}}\mathbf{D} = \mathbb{I}_L, \tag{3.1}$$

où les vecteurs de $\mathbf{D}$ sont orthogonaux 2 à 2 : $\mathbf{d}_j^{\mathrm{T}}\mathbf{d}_k = \delta_{j,k}$, $\forall j, k$. Pour reconstruire $\mathbf{y}_i$ on distingue deux cas :

- $K = L$: il s'agit d'un changement de base, et donc pas de réduction de dimension, pas de perte d'information. En particulier, $\mathbf{D}$ est inversible et $\mathbf{D}^{-1} = \mathbf{D}^{\mathrm{T}}$. Dans ce cas $\mathbf{D}\mathbf{w}_i = \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{y}_i$ et $\mathbf{y}_i = \mathbf{D}\mathbf{w}_i$.
- $K < L$, *i.e.* réduction de dimension, la reconstruction de $\mathbf{y}_i$ est faite par l'approximation. $\hat{\mathbf{y}_i} \approx \mathbf{D}\mathbf{w}_i$, ou encore, $\hat{\mathbf{y}_i} \approx \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{y}_i$.

L'ACP définit le projecteur $\mathbf{D}$ qui minimise l'erreur quadratique d'approximation :

$$\mathbf{D} = \underset{\mathbf{D}\in\mathbb{R}^{L\times K}}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{y}_i\|^2 \text{ où } K < L. \tag{3.2}$$

Ceci revient à maximiser par rapport à $\mathbf{D}$ la variance $\mathbf{D}^{\mathrm{T}}\mathbf{\Sigma}\mathbf{D}$ des points projetés. On cherche ainsi à trouver les K vecteurs qui portent le maximum de variance des données. On peut montrer qu'il s'agit des K vecteurs propres associés aux K plus grandes valeurs propres de la matrice de covariance empirique $\mathbf{\Sigma}$.

Dans l'ACP les composantes principales calculées sont estimées à partir de la matrice de covariance empirique. D'un point de vue probabiliste, quand la densité de distribution des données est gaussienne, l'ACP impose la contrainte d'indépendance aux statistiques d'ordre deux.
Cela est toutefois un inconvénient lorsque les données ne sont pas distribuées de façon Gaussienne. La section suivante présente une autre méthode qui relâche la contrainte d'orthogonalité et va plutôt chercher à trouver une famille de cœfficients indépendants.

3.1.2 Analyse en Composantes Indépendantes

L'ACP fournit une matrice orthogonale qui définit un ensemble de directions selon lesquelles les composantes sont décorrélées. Ces directions sont définies à une rotation près, et ne correspondent pas nécessairement à des composantes indépendantes aux ordres supérieurs à 2 lorsque les distributions sous-jacentes ne sont pas gaussiennes. L'ICA (Independent Component Analysis) [48] cherche précisément à identifier des directions indépendantes, souvent à partir d'observations non gaussiennes qui ont au préalables été blanchies grâce à une ACP. Il s'agit alors d'identifier une matrice orthogonale (donc de rotation) telles que les composantes des données projetées sur les colonnes de cette matrice soient indépendantes (et non gaussiennes).

Une approche classique consiste à identifier des composantes minimisant leur information mutuelle. Soit $\mathbf{y}$ une observation. On cherche une matrice orthogonale $\mathbf{D}^{\mathrm{T}}$ telle que les composantes de $\mathbf{w} = \mathbf{D}^{\mathrm{T}}\mathbf{y}$ où $\mathbf{w} = (w_k)_{k=1,K}$ soit indépendantes. Si $\mathrm{p}(w)$ est la densité de probabilité de w, on note $H(w)$ l'entropie différentielle d'une variable aléatoire :

$$H(w) = -\int \mathrm{p}(w)\log \mathrm{p}(w)\mathrm{d}w \tag{3.3}$$

On montre alors que l'information mutuelle entre ces composantes est telle que :

$$I(\mathbf{w}) = \sum_{k=1}^{K} H(w_k) - H(\mathbf{w}) = \sum_{k=1}^{K} H(w_k) - H(\mathbf{y}) \tag{3.4}$$

La quantité $I(\mathbf{w})$ s'interprète comme la divergence de Kullback-Leibler entre la densité jointe $g(\mathbf{w})$ et sa version factorisée $\prod_{k=1}^{K} g_k(w_k)$. Il y a égalité lorsque les composantes w_k sont indépendantes. Il apparaît que l'objectif de l'ICA est d'identifier la

matrice $\mathbf{D}^{\mathrm{T}}$ qui minimise $I(\mathbf{w})$, c'est-à-dire qui minimise la somme des entropies individuelles des composantes w_k. Cela revient à identifier les composantes les moins gaussiennes puisque la distribution gaussienne est aussi la distribution d'entropie maximale.

Plusieurs approches ont été proposées pour résoudre ce problème [49, 50], éventuellement en approchant l'écart à la gaussienne par la mesure de la kurtosis (moments d'ordre 4). On obtient alors une factorisation de matrice impliquant une matrice orthogonale (donc non redondante) maximisant l'indépendance entre les colonnes. Ce type d'approches est particulièrement pertinent en séparation de sources.

3.2 Apprentissage de dictionnaire en traitement d'image

L'image est un ensemble de données. En traitement d'image, chaque donnée est en général une imagette (ou *patch*) [4]. Soit $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N] \in \mathbb{R}^{L \times N}$, un ensemble de N patches extraits d'une image. Chaque $\mathbf{y}_i$ représente un patch de taille $\sqrt{L} \times \sqrt{L}$ rangé par ordre lexicographique dans un vecteur colonne de dimension $\mathbb{R}^L$. Les patches sont souvent de taille 8×8 [4]. Par exemple, avec une image de taille 256×256, on a $N = 62001$ de patches 8×8 chevauchants (*overlapping*) ou $N = 255025$ pour une image de taille 512×512. L'idée est de chercher à décomposer de manière parcimonieuse des patchs d'image sur un ensemble redondant d'atomes. Cet ensemble s'appelle dictionnaire où l'atome a même taille qu'une patch. En résumé, on cherche à apprendre un dictionnaire redondant pour représenter l'image de façon parcimonieuse. Un dictionnaire est dit redondant si sa taille est supérieure à la dimension de l'espace dans lequel vivent les données. Plusieurs travaux proposent alors d'apprendre un dictionnaire redondant où le nombre d'atomes K est supérieur à la dimension L de l'espace depuis les travaux de [2]. Le plus souvent dans la littérature, un dictionnaire de $K = 256$ ou 512 atomes est appris [4, 7, 51].

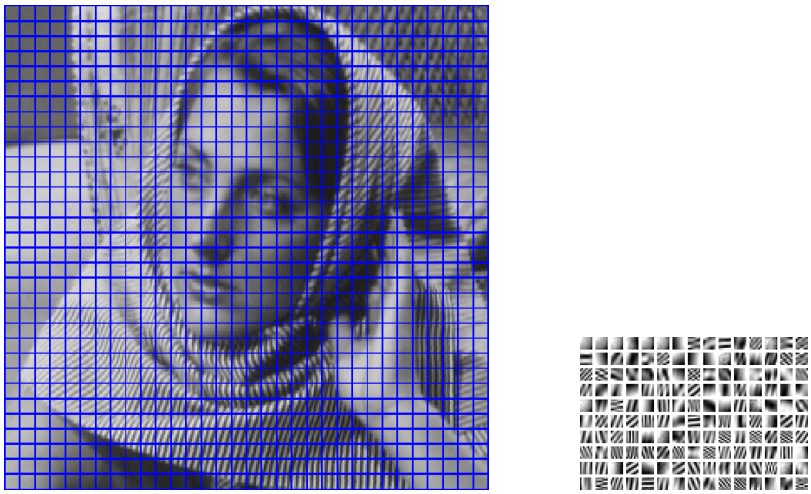
La figure 3.1 illustre l'exemple de la localisation des patches d'une image et son dictionnaire appris à partir de la méthode IBP-DL [25] qui va être présenté au chapitre 7. Nous cherchons à représenter $\mathbf{Y}$ par une combinaison linéaire $\mathbf{W}$ d'atomes, notés $\mathbf{D}$. L'apprentissage de dictionnaire revient à factoriser la matrice des observations sous la forme du produit de deux matrices :

$$\mathbf{Y} \approx \mathbf{DW} \tag{3.5}$$

où $\mathbf{D} = (\mathbf{d}_1, ..., \mathbf{d}_K) \in \mathbb{R}^{L \times K}$ que nous voulons apprendre est la matrice de dictionnaire dont les vecteurs colonnes sont les K atomes du dictionnaire. La matrice des cœfficients est $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_N] \in \mathbb{R}^{K \times N}$. Chaque vecteur colonne $\mathbf{w}_i$ encode la représentation parcimonieuse de chaque observation $\mathbf{y}_i$ sur $\mathbf{D}$. Chaque atome $\mathbf{d}_k$ du dictionnaire est normalisé, c'est-à-dire de norme ℓ_2 unitaire : $\|\mathbf{d}_k\|_2 = 1, \forall k = 1, \ldots, K$ Afin de trouver la meilleure représentation parcimonieuse, le problème de l'équation (3.5) consiste à trouver $\mathbf{D}$ et $\mathbf{W}$ en minimisant l'erreur de reconstruction et en respectant le contrainte de parcimonie. Le problème est par exemple :

$$\min_{\mathbf{D},\mathbf{W}} \|\mathbf{Y} - \mathbf{DW}\|_F^2 \text{ sous contraintes } \|\mathbf{w}_i\|_0 \leq T_0 \ \forall i \text{ et } \|\mathbf{d}_k\|_2 = 1 \ \forall k \tag{3.6}$$

où $T_0 \in \mathbb{N}^*$, $\|.\|_F$ est la norme de Frobenius, $i = 1, \ldots, N$, $k = 1, \ldots, K$.



(a)Patches extraits sur l'image (b)Dictionnaire : 150 atomes

Figure 3.1 – (a) L'image et sa segmentation en patches, (b) son dictionnaire appris à partir de la méthode IBP-DL avec des atomes ordonnés par contribution décroissante. Les deux figures sont représentées à la même échelle.

Initialisation : t$\leftarrow$ t+1 ; $\mathbf{D}^{(0)}$;
répéter
t$\leftarrow$ t+1 ;
$\mathbf{w}_i^{(t)} = \underset{\mathbf{w}_i}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{D}^{(t-1)}\mathbf{w}_i\|_2^2$ sous contrainte $\|\mathbf{w}_i\|_0 \leq T_0 \; \forall i$
$\mathbf{D}^{(t)} = \underset{\mathbf{D}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{W}^{(t)}\|_F^2$ sous contrainte $\|\mathbf{d}_k\|_2 = 1 \; \forall k$
jusqu'au La convergence est atteinte

Algorithme 6 : Principe des algorithmes d'apprentissage de dictionnaire par optimisation alternée.

L'algorithme 6 présente le principe des algorithmes d'apprentissage de dictionnaire par optimisation alternée qui se basent sur deux étapes :

1. Dans l'étape 1, en fixant le dictionnaire, les coefficients sont mis à jour en respectant les contraintes de parcimonie. Il s'agit des méthodes de décomposition parcimonieuse présentées dans le chapitre 2.
2. L'étape 2 consiste la mise à jour du dictionnaire où les coefficients sont fixés.

Dans la suite, plusieurs méthodes d'apprentissage de dictionnaire seront présentées. Dans ces méthodes, le critère d'arrêt peut par exemple être un nombre maximum d'itérations fixé, une erreur de reconstruction à atteindre ou un test sur la convergence de cette erreur.

3.2.1 Algorithme de descente de gradient

Les dictionnaires redondants ont été introduit par Olshausen et Field en 1996 [2]. Dans leurs premiers travaux [2, 52] sur l'apprentissage de dictionnaire redondant, la

mise à jour des coefficients et du dictionnaire repose sur une descente de gradient. La normalisation du dictionnaire est effectuée après sa mise à jour, elle n'est pas incluse dans le critère. Chaque atome du dictionnaire est multiplié par un facteur qui permet à tous les coefficients d'être de variance unité.

3.2.2 Méthode des Directions Optimales

La méthode de directions optimales [53] ou en anglais *Method of Optimal Directions*, (MOD) permet d'apprendre itérativement le dictionnaire. Pour chaque itération, la mise à jour des coefficients parcimonieuse est typiquement réalisée avec un algorithme de poursuite par exemple MP ou OMP (c.f 2.2). La mise à jour du dictionnaire est faite globalement sur l'ensemble du dictionnaire. Cela consiste à minimiser l'erreur quadratique : $\min_{\mathbf{D}} \|\mathbf{Y} - \mathbf{DW}\|_F^2$. Pour cela, MOD propose d'évaluer le dictionnaire à l'itération t de la manière suivante :

$$\mathbf{D}^{(t)} = \mathbf{D}^{(t-1)} + \Delta \tag{3.7}$$

où Δ est une matrice qui contient les vecteurs de réglage optimaux correspondant aux *directions optimales* pour la mise à jour du dictionnaire. D'après les calculs détaillés dans [54, 55] l'expression de Δ est :

$$\Delta = \mathbf{R}\mathbf{W}^{(t-1)\mathrm{T}}(\mathbf{W}^{(t-1)}\mathbf{W}^{(t-1)\mathrm{T}})^{-1} \tag{3.8}$$

où $\mathbf{R}$ est la matrice de résidu et $\mathbb{R} = \mathbf{Y} - \mathbf{D}^{(t-1)}\mathbf{W}^{(t-1)}$.

L'équation (3.7) maintenant devient :

$$\begin{aligned}
\mathbf{D}^{(t)} &= \mathbf{D}^{(t-1)} + (\mathbf{Y} - \mathbf{D}^{(t-1)}\mathbf{W}^{(t-1)})\mathbf{W}^{(t-1)\mathrm{T}}(\mathbf{W}^{(t-1)}\mathbf{W}^{(t-1)\mathrm{T}})^{-1} \\
&= \mathbf{D}^{(t-1)} + (\mathbf{Y}\mathbf{W}^{(t-1)\mathrm{T}} - \mathbf{D}^{(t-1)}\mathbf{W}^{(t-1)}\mathbf{W}^{(t-1)\mathrm{T}})(\mathbf{W}^{(t-1)}\mathbf{W}^{(t-1)\mathrm{T}})^{-1} \\
&= \mathbf{D}^{(t-1)} + (\mathbf{Y}\mathbf{W}^{(t-1)\mathrm{T}}(\mathbf{W}^{(t-1)}\mathbf{W}^{(t-1)\mathrm{T}})^{-1} - \mathbf{D}^{(t-1)}) \\
&= \mathbf{Y}\mathbf{W}^{(t-1)\mathrm{T}}(\mathbf{W}^{(t-1)}\mathbf{W}^{(t-1)\mathrm{T}})^{-1}
\end{aligned} \tag{3.9}$$

La solution est resemble à la solution des moindres carrés où $\mathbf{W}^{(t-1)\mathrm{T}}(\mathbf{W}^{(t-1)}\mathbf{W}^{(t-1)\mathrm{T}})^{-1}$ est la pseudo-inverse de $\mathbf{W}^{(t-1)}$. Ensuite, les colonnes du dictionnaire sont normalisées. On remarque aussi que dans le cas où le dictionnaire contient un grand nombre d'atomes, c'est-à-dire K est très grand l'inversion matricielle de $\mathbf{W}\mathbf{W}^{\mathrm{T}}$ (de taille $K \times K$) devient numériquement coûteuse.

3.2.3 Algorithme K-SVD

L'algorithme K-SVD [7, 56] propose de résoudre l'équation (3.6) en mettant à jour alternativement le dictionnaire et les coefficients d'encodage associés. Le principe est de fixer d'abord le dictionnaire $\mathbf{D}$ et d'estimer la matrice des coefficients $\mathbf{W}$ par approximation parcimonieuse. Cette étape est appelée étape de codage parcimonieux (*Sparse Coding Stage*) dans laquelle on cherche à résoudre par exemple :

$$\min_{\mathbf{W}} \sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{D}\mathbf{w}_i\|_F^2 \text{ sous contrainte } \|\mathbf{w}_i\|_0 \leq T_0 \,, \forall i. \tag{3.10}$$

Les algorithmes de poursuite (c.f 2.2) ont été proposées pour aborder le problème. Par exemple la méthode à base d'Orthogonal Matching Pursuit (OMP) a été utilisé dans [56] pour une application de débruitage en utilisant K-SVD. La norme ℓ_0 est non-convexe (voir 2). La solution obtenue peut correspondre à un minimum local. Il existe des méthodes qui choisissent d'approximer la norme ℓ_0 par la norme ℓ_1.

La seconde étape s'appelle *Codebook Update* ou mise à jour du dictionnaire. Le support de $\mathbf{W}$ est conservé, le dictionnaire $\mathbf{D}$ et les coefficients non nuls de $\mathbf{W}$ sont mis à jour en utilisant la première composante de la SVD du terme d'erreur, sous contrainte que chaque atome soit normalisé.

$$\min_{\mathbf{D}} \|\mathbf{Y} - \mathbf{D}\mathbf{W}\|_F^2. \tag{3.11}$$

Dans cette étape, l'algorithme K-SVD va optimiser successivement chaque atome $\mathbf{d}_k$ du dictionnaire indépendamment des autres. Pour rappel, $\mathbf{w}_{j,:}$ est le $j^{\text{ème}}$ vecteur ligne de matrice $\mathbf{W}$ (ce n'est pas $\mathbf{w}_j$ qui est le $j^{\text{ème}}$ vecteur colonne de matrice $\mathbf{W}$). Et $\|.\|_F$ désigne la norme de Frobenius. Soit $\mathbf{E}_{\neq k}$ l'erreur de reconstruction sans compter l'utilisation de l'atome k. La fonction objectif de l'équation (3.11) peut être réécrite de la manière suivante :

$$\|\mathbf{Y} - \sum_{j=1}^{K} \mathbf{d}_j \mathbf{w}_{j,:}\|_F^2 = \|\mathbf{Y} - \sum_{\substack{j=1 \\ j\neq k}}^{K} \mathbf{d}_j \mathbf{w}_{j,:} - \mathbf{d}_k \mathbf{w}_{k,:}\|_F^2 = \|\mathbf{E}_{\neq k} - \mathbf{d}_k \mathbf{w}_{k,:}\|_F^2. \tag{3.12}$$

K-SVD propose de définir $\mathbf{r}_k$ les indices des données associées à l'atome k (c'est à dire des coefficients non nuls).

$$\mathbf{r}_k = \{i \mid 1 \leq i \leq N, \mathbf{w}_{k,:}(i) \neq 0\}. \tag{3.13}$$

Soit m_k le nombre de données associées à l'atome k. Soit $\mathbf{\Omega}_k$ une matrice de taille $N \times m_k$, pour $t \in [1, m_k]$, $\mathbf{\Omega}_k(\mathbf{r}_k(t), t)$=1 et 0 si non. On obtient le vecteur ligne contenant des coefficients non nuls associés à l'atome k : $\mathbf{w}^{\mathbf{r}}_{j,:} = \mathbf{w}_{j,:}(\mathbf{r}_k) = \mathbf{w}_{j,:}\mathbf{\Omega}_k$. La matrice qui ne contient que des données utilisant l'atome k : $\mathbf{Y}^{\mathbf{r}}_k = \mathbf{Y}(:, \mathbf{r}_k) = \mathbf{Y}\mathbf{\Omega}_k$. De la même manière, on obtient $\mathbf{E}^{\mathbf{r}}_{\neq k} = \mathbf{E}_{\neq k}(:, \mathbf{r}_k) = \mathbf{E}_{\neq k}\mathbf{\Omega}_k$. La décomposition SVD de $\mathbf{E}^{\mathbf{r}}_{\neq k} = \mathbf{U}\mathbf{\Delta}\mathbf{V}^{\mathrm{T}}$. La façon de mettre à jour $\mathbf{d}_k$ et $\mathbf{w}^{\mathbf{r}}_{j,:}$ est :

- $\mathbf{d}_k$ est la première colonne de $\mathbf{U}$, la norme ℓ_2 de $\mathbf{d}_k$ est donc égale à 1 ;
- $\mathbf{w}^{\mathbf{r}}_{j,:}$ est la première colonne de $\mathbf{V}$ multiplié par $\mathbf{\Delta}(1,1)$.

Certaines méthodes basées sur K-SVD comme EK-SVD[11], Sub clustering K-SVD[12] ou Stagewise K-SVD[13] suggèrent une augmentation ou une diminution de la taille du dictionnaire afin de déterminer automatiquement le nombre *efficace* d'atomes du dictionnaire. L'apprentissage d'un dictionnaire de taille adaptative a récemment été proposé par la méthode DLENE[14]. Deux atomes initiaux sont récursivement améliorés en visant un compromis entre l'erreur de reconstruction et la parcimonie de la représentation.

3.2.4 Approches bayésienne et méthodes de Monte-Carlo

L'apprentissage de dictionnaire par des approches bayésiennes a été étudié dans une moindre mesure. Dans le cadre Bayésien, le dictionnaire $\mathbf{D}$ est maintenant consi-

déré comme une variable aléatoire. La recherche d'un dictionnaire et d'une représentation parcimonieuse amène l'écriture d'une loi de probabilité *a posteriori* . Les méthodes de Monte Carlo par chaines de Markov (MCMC) peuvent être utilisées afin d'échantillonner suivant cette loi. Le modèle d'analyse des facteurs de processus bêta (*Beta Process Factor Analysis*, BPFA) [51] propose une approche de la famille bayésienne pour l'apprentissage de dictionnaire. Cette approche introduit une loi *a priori* de type Bêta-Bernoulli sur le support des représentations pour favoriser la parcimonie. Pour rappel, une matrice d'observation $\mathbf{Y} \in \mathbb{R}^{L \times N}$ contient N vecteurs colonnes $\mathbf{y}_i$ avec $i = 1, \dots N$. Ces observations sont perturbées par un bruit gaussien i.i.d $\boldsymbol{\varepsilon}$. Afin de retrouver les observations *nettes* (sans bruit), on veut décomposer $\mathbf{Y}$ sous la forme :

$$\mathbf{Y} = \mathbf{DW} + \boldsymbol{\varepsilon} \tag{3.14}$$

où $\boldsymbol{\varepsilon}$ de taille $L \times N$ est le bruit additif, $\mathbf{D}$ de taille $L \times K$ est le dictionnaire et $\mathbf{W}$ de taille $K \times N$ est la représentation parcimonieuse sur $\mathbf{D}$. Le modèle BFFA est décrit par :

$$\mathbf{y}_i = \mathbf{D}\mathbf{w}_i + \boldsymbol{\varepsilon}_i \tag{3.15}$$
$$\boldsymbol{\varepsilon}_i \sim \mathcal{N}(0, \sigma_{\boldsymbol{\varepsilon}}^2 \mathbb{I}_L) \tag{3.16}$$
$$\mathbf{d}_k \sim \mathcal{N}(0, L^{-1}\mathbb{I}_L), \forall k = 1, \dots, K \tag{3.17}$$
$$\mathbf{w}_i = \mathbf{z}_i \odot \mathbf{s}_i, \tag{3.18}$$
$$\mathbf{s}_i \sim \mathcal{N}(0, \sigma_s^2 \mathbb{I}_K), \tag{3.19}$$
$$\mathbf{z}_i \overset{i.i.d}{\sim} \prod_{k=1}^{K} \text{Bernoulli}(\pi_k) \tag{3.20}$$
$$\pi \overset{i.i.d}{\sim} \prod_{k=1}^{K} \text{Beta}(a/K, b(K-1)/K) \tag{3.21}$$

où π_k est le $k^{\text{ème}}$ élément de π, $\mathbb{I}$ représente la matrice identité. Il est à noter que ce modèle inclut un bruit Gaussien sphérique additif, et les a priori sur $\mathbf{d}_k$ et $\mathbf{s}_i$ sont des lois normales. Ici, la représentation parcimonieuse contient deux parties :

— le support des représentations $\mathbf{Z}$ qui est une matrice binaire,
— la valeur des coefficients $\mathbf{S}$ qui est une matrice réelle.

L'opérateur $\odot$ est le produit matriciel de Hadamard qui est le produit terme à terme des deux matrices de mêmes dimensions. Dans l'esprit de la loi *a priori* Bernoulli-Gaussienne présentée dans 2.6.7.4, la parcimonie de $\mathbf{W}$ est induite par celle de la matrice $\mathbf{Z}$. Quand $\mathbf{Z}(k,i)$ vaut 0, $\mathbf{W}(k,i)$ vaut 0, quand $\mathbf{Z}(k,i)$ vaut 1, $\mathbf{W}(k,i)$ prend la valeur de $\mathbf{S}(k,i)$. $\mathbf{Z}$ suit de plus ici une loi *a priori* Bêta-Bernoulli et non pas seulement Bernoulli. C'est-à-dire, une loi *a priori* Bêta est choisie pour les poids (les probabilités d'utilisation) des atomes. Il est à noter que les lois *a priori* de type Bêta des paramètres des lois de type Bernoulli constituent une famille conjuguée. Chaque atome $\mathbf{d}_k$ du dictionnaire $\mathbf{D}$ est de taille L. Le modèle impose la variance $\sigma_{\mathbf{D}}^2$ de $\mathbf{D}$ à $1/L$ en raison de problème l'indétermination du couple ($\mathbf{D}$,$\mathbf{W}$) à un facteur multiplicatif près.

Comme cela a été détaillé section 2.6.5, l'intérêt d'utiliser l'inférence bayésienne est entre autres de pouvoir négliger la connaissance du niveau de bruit. Le niveau de

bruit sera ainsi conjointement échantillonné en choisissant une loi *a priori* conjuguée pour sa variance. La variance des coefficients σ_S^2 , le vecteur des poids des coefficients π_k seront aussi échantillonnés. Le problème est écrit maintenant sous la forme d'une loi a posteriori :

$$p(\mathbf{D}, \mathbf{Z}, \mathbf{S}, \pi, \sigma_S, \sigma_\varepsilon \mid \mathbf{Y}) \tag{3.22}$$

L'échantillonnage de Gibbs est utilisé pour mettre à jour alternativement les paramètres et les variables du modèle :

1. La loi a posteriori de chaque atome $\mathbf{d}_k$ est donnée par :

$$p(\mathbf{d}_k|\mathbf{Y}, \mathbf{D}_{-k}, \mathbf{Z}, \mathbf{S}, \sigma_\varepsilon) \propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i; \mathbf{D}\mathbf{w}_i, \sigma_\varepsilon^2 \mathbb{I}_L)\mathcal{N}(\mathbf{d}_k; 0, L^{-1}\mathbb{I}_L) \tag{3.23}$$

$$\propto \mathcal{N}(\boldsymbol{\mu}_{\mathbf{d}_k}, \boldsymbol{\Sigma}_{\mathbf{d}_k}) \tag{3.24}$$

où

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{d}_k} &= (\sigma_D^{-2}\mathbb{I}_L + \sigma_\varepsilon^{-2} \sum_{i=1}^{N} w_{ki}^2)^{-1} \\ \boldsymbol{\mu}_{\mathbf{d}_k} &= \sigma_\varepsilon^{-2}\boldsymbol{\Sigma}_{\mathbf{d}_k} \sum_{i=1}^{N} w_{ki}(\mathbf{y}_i - \sum_{j\neq k}^{K} \mathbf{d}_j w_{ji}) \end{aligned} \tag{3.25}$$

2. Chaque élément z_{ki} du support des représentations $\mathbf{Z}$ est échantillonné selon la loi a postériori suivante :

$$p(z_{ki}|\mathbf{Y}, \mathbf{D}, \mathbf{Z}_{-ki}, \mathbf{S}, \sigma_\varepsilon, \pi_k) \propto \mathcal{N}(\mathbf{y}_i; \mathbf{D}(\mathbf{z}_i \odot \mathbf{s}_i), \sigma_\varepsilon^2)P(z_{ki}|\pi_k) \tag{3.26}$$

Les probabilités a posteriori de $z_{ki} = 0$ ou 1 sont proportionnelles à (p_0, p_1) définis par

$$\begin{cases} p_0 &= 1 - \pi_k \\ p_1 &= \pi_k \exp\left[-\frac{1}{2\sigma_\varepsilon^2}(s_{ki}^2 \mathbf{d}_k^T\mathbf{d}_k - 2s_{ki}\mathbf{d}_k^T(\mathbf{y}_i - \sum_{j\neq k}^{K} \mathbf{d}_j z_{ji} s_{ji})\right] \end{cases} \tag{3.27}$$

z_{ki} peut être tiré selon la distribution de Bernoulli suivante :

$$z_{ki} \sim \text{Bernoulli}\left(\frac{p_1}{p_0 + p_1}\right). \tag{3.28}$$

3. La loi a posteriori de chaque élément s_{ki} de $\mathbf{S}$ est donnée selon (3.29) :

$$p(s_{ki}|\mathbf{Y}, \mathbf{D}, \mathbf{Z}, \mathbf{S}_{-ki}, \sigma_\varepsilon, \sigma_S) \propto \mathcal{N}(\mathbf{y}_i; \mathbf{D}(\mathbf{s}_i \odot \mathbf{z}_i), \sigma_\varepsilon^2\mathbb{I}_L)\mathcal{N}(\mathbf{s}_i; 0, \sigma_S^2\mathbb{I}_K) \tag{3.29}$$

$$\propto \mathcal{N}(\mu_{s_{ki}}, \Sigma_{s_{ki}}) \tag{3.30}$$

où

$$\begin{aligned} z_{ki} = 1 \Rightarrow &\begin{cases} \Sigma_{s_{ki}} = (\sigma_\varepsilon^{-2}\mathbf{d}_k^T\mathbf{d}_k + \sigma_S^{-2})^{-1} \\ \mu_{s_{ki}} = \sigma_\varepsilon^{-2}\Sigma_{s_{ki}}\mathbf{d}_k^T(\mathbf{y}_i - \sum_{j\neq k}^{K} \mathbf{d}_j w_{ji}) \end{cases} \\ z_{ki} = 0 \Rightarrow &\begin{cases} \Sigma_{s_{ki}} = \sigma_S^2 \\ \mu_{s_{ki}} = 0 \end{cases} \end{aligned} \tag{3.31}$$

4. Le vecteur des poids des atomes π peut être échantillonné aussi. On peut montrer que la loi a postériori de chaque élément π_k est une loi Bêta.

$$p(\pi_k|a,b,\mathbf{z}_{k,:}) \propto \prod_{i=1}^{N} \text{Bernoulli}(z_{ki};\pi_k)\ \text{Beta}(\pi_k; a/K, b(K-1)/K) \tag{3.32}$$

$$\propto \text{Beta}\left(a/K + \sum_{i=1}^{N} z_{ki}, b(K-1)/K + N - \sum_{i=1}^{N} z_{ki}\right) \tag{3.33}$$

où $\mathbf{z}_{k,:}$ est bien le $k^{\text{ème}}$ vecteur ligne de la matrice $\mathbf{Z}$.

5. La variance des coefficients σ_S^2 a une loi *a priori* conjuguée de type Inverse Gamma (c.f 2.6.5). Pour simplifier, la loi a postériori de l'inverse de la variance s'appelant aussi la précision des coefficients $\frac{1}{\sigma_S^2}$ sera calculée. La loi *a priori* de la précision est donc de type Gamma.

$$p(\frac{1}{\sigma_S^2}|c,d,\mathbf{S}) \propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{s}_i; 0, \sigma_S^2\mathbb{I}_K)\ \text{Gamma}(\frac{1}{\sigma_S^2}; c, d) \tag{3.34}$$

$$\propto \text{Gamma}\left(c + \frac{KN}{2}, d + \frac{1}{2}\sum_{i=1}^{N} \mathbf{s}_i^T\mathbf{s}_i\right) \tag{3.35}$$

6. Idem pour la variance du bruit σ_ε, la loi *a posteriori* de la précision du bruit est :

$$p(\frac{1}{\sigma_\varepsilon^2}|e,f,\mathbf{Y},\mathbf{D},\mathbf{W}) \propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i; \mathbf{D}\mathbf{w}_i, \sigma_\varepsilon^2\mathbb{I}_L)\ \text{Gamma}(\frac{1}{\sigma_\varepsilon^2}; e, f) \tag{3.36}$$

$$\propto \text{Gamma}\left(e + \frac{LN}{2}, f + \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{D}\mathbf{w}_i\|_2^2\right) \tag{3.37}$$

Le modèle d'analyse des facteurs de processus bêta (*Beta Process Factor Analysis*, BPFA) [51] est un dans des premières dans des premiers travaux àl'interface entre les modèles bayésiens non paramétriques et le traitement de signal et des images. Cependant, malgré le titre de l'article qui annonce une approche non paramétrique, il s'agit en fait d'une approximation paramétrique de l'IBP, car elle fonctionne avec un nombre d'atomes fixé à l'avance.

3.3 Discussion

Il existe une connaissance implicite dans les méthodes présentées ci-dessus, c'est le nombre d'atomes K du dictionnaire. Il est cependant naturel de se demander s'il est possible d'estimer le nombre d'atome K, *i.e.* sans le fixer en avance ? Le plus souvent dans la littérature, un dictionnaire de $K = 256$ ou 512 atomes est appris [4, 7, 51]. Il est à noter que dans la plupart des méthodes d'optimisation, non seulement la taille du dictionnaire est fixée en avance, mais le niveau de bruit σ_ε est aussi connu.

Les approches bayésiennes permettent tout d'abord de ne pas fixer à l'avance le niveau de bruit et de le considérer comme une variable aléatoire selon une loi *a priori*. Il est ensuite échantillonné suivant la loi *a posteriori* . De plus, en utilisant les approches bayésiennes non paramétriques, la taille K du dictionnaire peut aussi être inférée.

Il existe une connexion entre le modèle BPFA [51] et les approches Bayésiennes non paramétriques. Malgré cette connexion, cela correspond à une approximation paramétrique du processus non paramétrique appelé Buffet Indien [57, 58] puisque cette approche fonctionne avec un (grand) nombre d'atomes, fixé à l'avance.

Une approche bayésienne non paramétrique considère que le dictionnaire est potentiellement de taille infinie mais on introduit un *a priori* contrôlant le nombre d'atomes et également favorisant la parcimonie de la représentation. Elle permet de développer des méthodes d'apprentissage de dictionnaire sans fixer à l'avance la taille du dictionnaire et de commencer avec un dictionnaire vide au départ. Ses propriétés ouvrent des perspectives prometteuses et encore peu explorées en traitement du signal et des images.

Le but de cette thèse est d'étudier les approches bayésiennes non paramétriques afin de proposer une méthode d'apprentissage de dictionnaire vraiment non paramétrique, dans le sens où aucun paramètre ne doit être fixé en avance. Tout sera estimé lors de l'inférence, la taille du dictionnaire incluse. Le prochain chapitre introduira les approches bayésiennes non paramétriques.

Notons au passage qu'il est possible d'ajouter des contraintes de positivité lors de l'apprentissage de dictionnaire. Par exemple, on est amené en traitement des signaux audios à travailler avec des spectrogrammes qui représentent la répartition de l'énergie d'un signal sur les fréquences au cours du temps. Dans ce cas, une approche de factorisation en matrice non-négative (NMF) est parfois proposée [59, 60].

CHAPITRE 4

Processus de Dirichlet à mélange

Le processus de Dirichlet est vu comme le processus fondateur de l'approche bayésienne non paramétrique [15, 61]. L'intérêt du processus de Dirichlet est connu dans les approches de classification [61–65].
Dans le cas de la classification non supervisée, les données sont vues comme des réalisations indépendantes d'une distribution inconnue souvent définie comme un mélange fini de gaussiennes de paramètres inconnus. Chaque mode du mélange correspond à une classe (*cluster*) différente. On cherche à estimer ces paramètres qui sont inconnus mais de dimension finie fixée par le nombre de classes choisi à l'avance. Le choix de la dimension des paramètres (lié au nombre de clusters) n'est pas toujours simple pour un modèle paramétrique.
Pour éviter ce choix souvent restrictif, les approches bayésiennes non paramétriques considèrent le nombre de clusters comme potentiellement infini en introduisant des lois *a priori* sur des espaces fonctionnels. Nous détaillerons cette approche pour le processus de Dirichlet à mélange.

4.1 Avant propos

Nous commençons par introduire le processus de Dirichlet de façon intuitive comme limite d'un modèle de mélange où le nombre de composantes tend vers l'infini. Nous procéderons à une définition plus formelle dans la section suivante. Soit $\mathbf{y}_1 \dots \mathbf{y}_N$ N observations tirées d'une densité inconnue F. On suppose à priori que F est un mélange fini de gaussiennes $\mathcal{N}(\mu_k, \sigma_k^2)$ où π_k, μ_k et σ_k^2 sont respectivement les poids, moyennes et variances inconnus de chaque composante. Notons $\mathbf{z}_n = [z_{1n} \dots z_{Kn}]$ la suite de variables latentes indiquant à quelle composante du mélange l'observation $\mathbf{y}_n$ est associée. On introduit alors le modèle hiérarchique [16]

$$\begin{aligned} \pi_1 \dots \pi_K &\sim \mathrm{p} \\ z_{1n} \dots z_{Kn} \mid \pi_1 \dots \pi_K &\sim \mathrm{Mult}(\pi_1 \dots \pi_K) \\ \mu_k, \sigma_k^2 &\sim \mathbb{G}_0 \\ \mathbf{y}_n \mid \mathbf{z}_n, \left\{\mu_k, \sigma_k^2\right\} &\sim \mathcal{N}(\mathbf{y}_n \mid \mu_k, \sigma_k^2), \end{aligned}$$

où $\mathbb{G}_0$ est la mesure de base sur les paramètres de moyenne et de variance. F s'écrit alors marginalement

$$F(\cdot) = \int \mathcal{N}(\cdot \mid \mu, \sigma^2)\mathbb{G}(\mathrm{d}\mu \mathrm{d}\sigma^2),$$

et où $\mathbb{G}$ est l'objet $\mathbb{G}(\cdot) \mid \pi_1 \dots \pi_K, \mu_1 \dots \mu_K, \sigma_1^2 \dots \sigma_K^2 = \sum_{k=1}^K \pi_k \delta_{\mu_k, \sigma_k^2}(\cdot)$. Si l'on fait maintenant tendre K vers l'infini, on peut montrer que $\mathbb{G}$ reste défini. C'est cet

objet que l'on appellera processus de Dirichlet à Mélange (DPM). Le processus de Dirichlet sera alors décrit par une loi de probabilité sur les mesures aléatoires $\mathbb{G}$.

4.2 Processus de Dirichlet (DP)

4.2.1 Loi de Dirichlet

Avant de parler du processus de Dirichlet, rappelons tout d'abord la définition et les propriétés de la distribution de Dirichlet. La distribution de Dirichlet est une généralisation de la loi Bêta. Pour rappel, les familles de lois Bêta et multinomiales sont conjuguées (*cf.* Table 2.1). On peut définir *la famille des lois de Dirichlet comme la conjuguée de la famille des lois Multinomiales.*

Définition 4. *Un vecteur aléatoire* $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_K]$ *à valeur dans le simplexe de* $\mathbb{R}^K$ *:*

$$\Delta_K = \left\{ [\pi_1, \pi_2, \ldots, \pi_K]; \pi_k > 0, k = 1, 2, \ldots, K; \sum_{k=1}^{K} \pi_k = 1 \right\} \tag{4.1}$$

suit une loi de Dirichlet de paramètres $\alpha_1, \alpha_2, \ldots, \alpha_K > 0$ *si sa densité de probabilité par rapport à la mesure de Lebesgue s'écrit :*

$$\frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}. \tag{4.2}$$

Remarque : *Si* $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ suit une loi de Dirichlet, notée

$$\boldsymbol{\pi} \sim \mathcal{D}ir(\alpha_1, \ldots, \alpha_K), \tag{4.3}$$

alors les K–1 premières composantes de π possèdent la distribution définie précédemment et $\boldsymbol{\pi}$ vérifie

$$\pi_K = 1 - \pi_1 - \cdots - \pi_{K-1}. \tag{4.4}$$

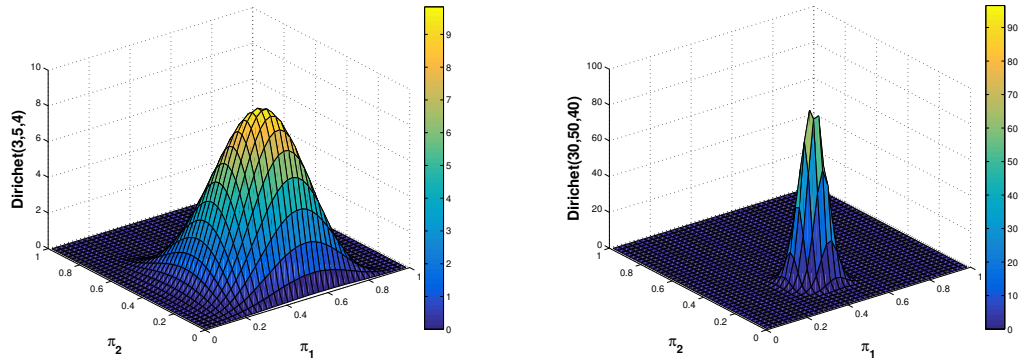
Remarque : Quand K vaut 2, on retrouve une loi Bêta de paramètres (α_1, α_2) sur l'équation (4.2). C'est pourquoi on parle d'une généralisation de la loi Bêta.

La figure 4.1 illustre un exemple de densités de la loi Dirichlet. Plus les α sont grands, au sens plus $\|\alpha\|_\infty$ est grand, plus la densité de la loi Dirichlet se concentre autour d'un "pic".

Propriété 1. *Soit* $\pi = (\pi_1, \ldots, \pi_K)$ *une variable aléatoire distribuée suivant une loi de Dirichlet de paramètre* $(\alpha_1 \ldots \alpha_K)$*. Notons* $\alpha = \sum_{i=1}^{K} \alpha_i$ *. Alors :*

$$\mathbb{E}[\pi_k] = \frac{\alpha_k}{\alpha} \tag{4.5}$$

$$\mathrm{Var}[\pi_k] = \frac{\alpha_k(\alpha - \alpha_k)}{\alpha^2(\alpha + 1)}. \tag{4.6}$$



(a) (b)

FIGURE 4.1 – Illustration des densités de Dirichlet avec K=3, dans le cas où (a) α =[3 5 4] et (b) α =[30 50 40]

4.2.2 Définition du processus de Dirichlet

Le processus de Dirichlet peut s'interpréter comme une généralisation de la loi de Dirichlet au cas où K est infini [16]. Il permet de définir une distribution sur un ensemble de distributions de probabilité. On définit maintenant le processus de Dirichlet [15] à partir d'un coefficient de précision $\alpha > 0$ et d'une mesure de base $\mathbb{G}_0$ qui est une loi de probabilité.

Définition 5. *Soit α un réel positif. Soit $(\Theta, \mathcal{A})$ un espace mesurable et $\mathbb{G}_0$ une mesure de probabilité sur $(\Theta, \mathcal{A})$. On dit qu'une distribution de probabilité $\mathbb{G}$ est distribuée selon un processus de Dirichlet de distribution de base $\mathbb{G}_0$ et de facteur d'échelle $\alpha > 0$ si pour toute partition mesurable $(A_1, \ldots, A_k)$ de Θ, le vecteur de probabilité aléatoire $[\mathbb{G}(A_1), \ldots, \mathbb{G}(A_k)]$ suit une distribution de Dirichlet standard :*

$$[\mathbb{G}(A_1), \ldots, \mathbb{G}(A_k)] \sim \mathcal{D}ir(\alpha\mathbb{G}_0(A_1), \ldots, \alpha\mathbb{G}_0(A_k)) \tag{4.7}$$

On le note :

$$\mathbb{G} \sim DP(\mathbb{G}_0, \alpha). \tag{4.8}$$

Le processus de Dirichlet est défini par deux paramètres : **la mesure de probabilité de base** $\mathbb{G}_0$ et **le paramètre d'échelle** α.

Propriété 2. *Soit $\mathbb{G}$ une mesure aléatoire distribuée suivant un processus de Dirichlet. Alors pour tout A élément de la tribu $\mathcal{A}$, la moyenne et la variance de $\mathbb{G}(A)$ sont les suivantes*

$$\mathrm{E}[\mathbb{G}(A)] = \mathbb{G}_0(A) \tag{4.9}$$

$$\mathrm{Var}[\mathbb{G}(A)] = \frac{\mathbb{G}_0(A)(1 - \mathbb{G}_0(A))}{\alpha + 1}. \tag{4.10}$$

Remarque :

1. *Si l'on reprend l'exemple du modèle de mélange Gaussien, on retrouve le fait que $\mathbb{G}_0$ est la loi normale Wishart, i.e. , une loi a priori jointe type sur l'espérance et la variance d'une Gaussienne.*

2. *La distribution de base $\mathbb{G}_0$ représente la valeur moyenne du processus de Dirichlet.*
3. *Plus α est grand, plus la variance est petite et le processus de Dirichlet est concentré autour de la distribution de base $\mathbb{G}_0$.*

4.2.3 Distribution *a posteriori*

La loi *a posteriori* d'un processus de Dirichlet après n observations est aussi un processus de Dirichlet. Supposons que $\mathbb{G}$ est distribuée selon un processus de Dirichlet de distribution de base $\mathbb{G}_0$ et de facteur d'échelle α :

$$\mathbb{G} \sim \text{DP}(\mathbb{G}_0, \alpha). \tag{4.11}$$

Soient n échantillons $\theta_1, \ldots, \theta_N$ échantillonnés de façon i.i.d selon la distribution $\mathbb{G}$.

$$\theta_i \mid \mathbb{G} \overset{i.i.d}{\sim} \mathbb{G} \text{ avec } i = 1, \ldots, n \tag{4.12}$$

La distribution $\mathbb{G}$ étant maintenant vue comme une variable aléatoire, on s'intéresse à déterminer la distribution *a posteriori* de $\mathbb{G}$ conditionnellement à $(\theta_1, \ldots, \theta_n)$. Comme évoqué ci-dessus, la distribution de Dirichlet conjuguée pour une loi Multinomiale. Selon l'équation (4.7), on peut montrer que [15] :

$$[\mathbb{G}(A_1), \ldots, \mathbb{G}(A_k)] \mid \theta_{1:n} \sim \mathcal{D}ir(\alpha\mathbb{G}_0(A_1) + \sum_{i=1}^{n} \delta_{\theta_i}(A_1), \ldots, \alpha\mathbb{G}_0(A_k) + \sum_{i=1}^{n} \delta_{\theta_i}(A_k)) \tag{4.13}$$

avec $\{A_1, \ldots, A_k\}$ une partition finie de Θ.

L'équation (4.13) est vraie pour toutes les partitions mesurables finies, par conséquent, d'après la définition 5, on peut en déduire que la distribution *a posteriori* est toujours distribuée selon un processus de Dirichlet :

$$\mathbb{G} \mid \theta_{1:n} \sim \text{DP}(\frac{\alpha}{\alpha + n}\mathbb{G}_0 + \frac{1}{\alpha + n}\sum_{i=1}^{n} \delta_{\theta_i}, \alpha + n). \tag{4.14}$$

4.2.4 Représentation de Backwell-Mac Queen

A partir de l'équation (4.14), il est possible d'obtenir analytiquement la loi marginale $\theta_{n+1} \mid \theta_{1:n}$ appelée aussi la distribution prédictive, obtenue en marginalisant par rapport à $\mathbb{G}$:

$$\theta_{n+1} \mid \theta_{1:n} \sim \frac{\alpha}{\alpha + n}\mathbb{G}_0 + \frac{1}{\alpha + n}\sum_{i=1}^{n} \delta_{\theta_i}. \tag{4.15}$$

L'équation (4.15) fournit un moyen pratique pour tirer un échantillon selon une mesure de probabilité aléatoire distribuée suivant un processus de Dirichlet, sans la construction explicite de la mesure $\mathbb{G}$. Ceci est appelé la représentation de Backwell-Mac Queen [66] qui est aussi connue comme une généralisation de la représentation en urne de Pólya.

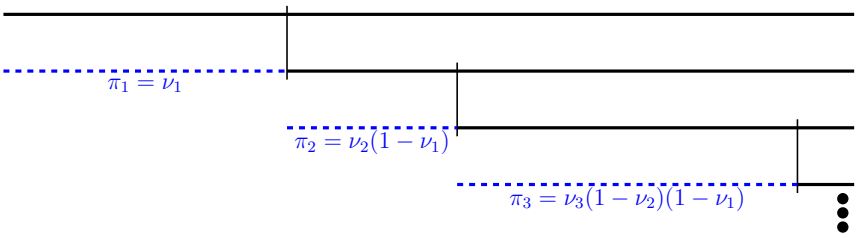


FIGURE 4.2 – La construction stick-breaking pour le processus de Dirichlet. On commence avec un bâton de longueur 1, et on casse récursivement des morceaux de longueurs $\pi_1, \pi_2, \ldots$

Urne de Pólya : Le modèle historique de l'urne de Pólya a été proposé par le mathématicien George (György) Pólya (1923). Une urne contient initialement R_0 boules rouges et N_0 boules noires. On tire une boule au hasard, les tirages de toutes les boules étant équiprobables. Puis on la replace dans l'urne et on ajoute une boule supplémentaire de la même couleur dans l'urne. La situation la plus simple considérée est celle d'un tirage uniforme. C'est-à-dire, à l'instant $n+1$, sachant R_n boules rouges et N_n boules noires dans l'urne, la probabilité de tirer une boule rouge est

$$\mathrm{P}(\theta_{n+1} = \text{"rouge"} \mid \theta_{1:n}) = \frac{R_n}{N_n + R_n} = \frac{\sum_{i=1}^{n} \delta_{\text{rouge}}(\theta_i)}{n + N_0 + R_0}. \tag{4.16}$$

Urne de Backwell-Mac Queen : Dans la représentation de Backwell-Mac Queen, on s'autorise une infinité (éventuellement non dénombrable) de couleurs de boule. Dans la métaphore de l'urne de Backwell-Mac Queen présentée équation (4.15), chaque valeur de Θ est une couleur unique. Les échantillons $\theta \sim \mathbb{G}$ sont les couleurs des boules tirées. En outre, une urne contient des boules vues précédemment. Au début, il n'y a pas de boules dans l'urne, et on choisit une couleur tirée selon la distribution de base $\mathbb{G}_0$, *i.e.* $\theta_1 \sim \mathbb{G}_0$. On prend une boule de cette couleur, et on la dépose dans l'urne. À l'étape $n+1$, on a deux possibilités. La première possibilité, avec probabilité $\frac{\alpha}{\alpha+n}$, on ajoute une boule de nouvelle couleur dans l'urne. Cette nouvelle couleur est distribuée (indépendamment) selon $\mathbb{G}_0$. La deuxième possibilité, avec une probabilité $\frac{n}{\alpha+n}$, est de tirer une boule au hasard dans l'urne, regarder sa couleur (échantillonner θ_{n+1} à partir de la distribution empirique). On prend une nouvelle boule avec la même couleur et l'on remet les deux boules dans l'urne.

Quelque soit la mesure de base $\mathbb{G}_0$ discrète ou continue, les échantillons θ sont tirés selon une mesure de probabilité aléatoire distribuée suivant un processus de Dirichlet qui est discret. De fait le nombre de couleurs tirées reste fini même quand le nombre d'observations tend vers l'infini. Un effet d'aggrégation des couleurs apparaît. Une même couleur peut être tirée plusieurs fois bien que la mesure de base $\mathbb{G}_0$ soit continue. Si l'on numérote les boules, on peut donc les partitionner suivant leur couleur. La distribution induite sur les partitions est appelée "processus du restaurant chinois" (voir partie 4.3). Elle est par exemple exploitée pour des tâches de *clustering*.

4.2.5 Construction stick-breaking

Une mesure de probabilité aléatoire issue d'un tirage suivant un processus de Dirichlet est presque sûrement une probabilité discrète [67]. On peut voir cette mesure comme un mélange infini dénombrable de mesures de Dirac [68]

$$\mathbb{G} = \sum_{j=1}^{\infty} \pi_k \delta_{\mathcal{C}_k} \tag{4.17}$$

avec $\mathcal{C}_k \sim \mathbb{G}_0$ et $\boldsymbol{\pi} \sim \text{GEM}(\alpha)$. La distribution de base $\mathbb{G}_0$ détermine les positions des composantes discrètes de $\mathbb{G}$. La séquence infinie $\boldsymbol{\pi}$ est définie par la distribution GEM, où les lettres signifient Griffiths, Engen et McCloskey [69]. La distribution GEM peut être construite via une procédure appelée stick-breaking que l'on va décrire. Soit $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots]$ une suite de réels construite vérifiant

$$\nu_k \overset{i.i.d}{\sim} \mathcal{B}(1, \alpha) \qquad k = 1, 2, \ldots \tag{4.18}$$

$$\pi_k = \nu_k \prod_{l=1}^{k-1} (1 - \nu_l) \qquad k = 1, 2, \ldots \tag{4.19}$$

où $\mathcal{B}$ désigne la distribution Beta et le facteur d'échelle α contrôle la variance des poids. Alors la suite $\boldsymbol{\pi}$ vérifie $\sum_{j=1}^{\infty} \pi_k = 1$ et définit bien une distribution de probabilité discrète. En raison de cette décomposition, on dit que les réalisations d'un processus de Dirichlet prennent la forme dite "stick-breaking" [68].

La figure 4.2 illustre la construction stick-breaking, *i.e.* la façon d'obtenir les poids π_k. Dans la métaphore de la construction stick-breaking, la somme de tous les poids se représente sous la forme d'un bâton de longueur initiale 1. À chaque instant k, nous cassons un morceau du bâton de longueur π_k qui correspond à une proportion aléatoire $\nu_k \sim \mathcal{B}(1, \alpha)$ du morceau du bâton restant.

Remarque : *L'ensemble des processus pour lesquels la mesure de probabilité aléatoire (Random Probability Measure,RPM) prend la forme de l'équation (4.17) avec $\mathcal{C}_k \sim \mathbb{G}_0$ et $\pi_k = \nu_k \prod_{l=1}^{k-1}(1 - \nu_l)$ s'appelle classe des stick-breaking. Les processus de cette classe diffèrent dans la façon d'obtenir le coefficient ν_k. Par exemple, pour le processus de Pitman-Yor (PYP) [70] (connu comme le processus de Poisson-Dirichlet à deux paramètres)*

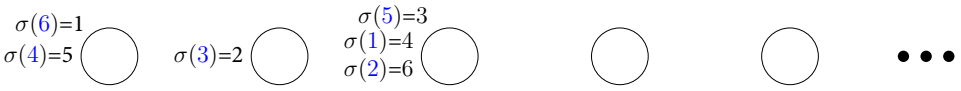
$$\nu_k \overset{i.i.d}{\sim} \mathcal{B}(1 - \varrho, \alpha + k\varrho) \text{ avec } k = 1, 2, \ldots$$

$\varrho \in [0, 1)$ et $\alpha \in (-\varrho, \infty)$. Quand $\varrho = 0$, on retrouve le processus de Dirichlet. Le PYP étant aussi connu comme une généralisation du processus de Dirichlet peut générer des lois de puissance via le paramètre ϱ [70].

4.3 Le processus du Restaurant Chinois (CRP)

Comme évoqué dans la partie 4.2.4, le processus de Dirichlet (DP) possède une deuxième représentation appelée processus du restaurant chinois *(Chinese Restaurant Process, CRP)*. La partie 4.4 reviendra en détails sur le lien entre DP et CRP. Le

(a) Une représentation du CRP après l'arrivée de 6 clients



(b) Une autre permutation du CRP de (a).
La permutation de client i est définie par $\sigma(i)$.

FIGURE 4.3 – Deux états possibles d'une représentation du processus du restaurant chinois après l'arrivée de 6 clients.

CRP a été appelé ainsi par Jim Pitman et Lester Dubins, sur la base d'une métaphore dans laquelle des clients apparentés à des observations, rentrent dans un restaurant et s'assoient chacun à une table. Ces tables sont apparentes à des classes (*clusters*).

Supposons qu'un restaurant chinois ait un nombre infini dénombrable de tables et que N clients rentrent un par un dans ce restaurant. La dynamique des clients est la suivante pour un paramètre $\alpha > 0$:

1. Le premier client s'assoit toujours à la première table.
2. Une fois que les n premiers clients sont entrés et occupent K tables, le client $n+1$ va choisir sa table de la façon suivante :
   - il choisit la table k parmi les K tables déja occupée avec probabilité $\frac{m_k}{n+\alpha}$ où m_k est le nombre de clients assistés à la table k.
   - sinon, il choisit une nouvelle table $K+1$ avec probabilité $\frac{\alpha}{n+\alpha}$.

Une fois que les N clients sont entrés, K tables sont occupées par un ou plusieurs clients. Chaque table forment un sous ensemble de $[1, N]$ et l'ensemble des tables forme une partition de $[1, N]$. En ce sens, ou dit que le processus du restaurant chinois est à valeur sur les partitions. Notons que le nombre de tables de ce restaurant chinois est potentiellement éventuellement infini.

La figure 4.3(a) illustre une réalisation du processus du restaurant chinois avec 6 clients où $\alpha = 1$. Le premier client choisit la table 1 avec une probabilité 1. Le deuxième client choisit de rejoindre le client 1 à la table 1 avec une probabilité de $1/2$. La troisième client décide de s'asseoir à la table 2 avec une probabilité $1/3$. Lorsque le septième client entre dans ce restaurant, il va rejoindre :

— les clients $1, 2, 5$ avec une probabilité $3/7$,
— les clients 3 avec une probabilité $1/7$,
— les clients $4, 6$ avec une probabilité $2/7$,
— et aller à une autre table avec une probabilité $1/7$.

Bien que l'histoire du CRP soit décrite en utilisant l'ordre des clients, la distribution sur les partitions définie par le CRP est invariante à l'ordre des données. Seul le nombre de classes compte dans la détermination de la probabilité de partition et pas les identités des clients. Cette propriété est connue sous le nom d'échangeabilité.

Définition 6. *Une suite finie de variables aléatoires $(\theta_1, \ldots, \theta_N)$, est dite échangeable si pour tout permutation σ de $\{1, \ldots, N\}$*

$$(\theta_1, \ldots, \theta_N) \overset{\text{loi}}{=} (\theta_{\sigma(1)}), \ldots, \theta_{\sigma(N)}).$$

Par extension, une suite infinie de variables aléatoires $(\theta_1, \theta_2, \ldots)$ est dite échangeable si quelque soit I un sous ensemble fini de $\mathbb{N}^*$, pour toute permutation finie σ de I

$$(\theta_i)_{i\in I} \overset{\text{loi}}{=} (\theta_{\sigma(i)})_{i\in I}.$$

Par exemple, la probabilité de la représentation sur la figure 4.3(a) est

$$\mathrm{P}(\{\{1,2,5\},\{3\},\{4,6\}\}) = \frac{\alpha}{\alpha}\frac{1}{\alpha+1}\frac{\alpha}{\alpha+2}\frac{\alpha}{\alpha+3}\frac{2}{\alpha+4}\frac{1}{\alpha+5}. \tag{4.20}$$

La figure 4.3(b) contient les même partitions que celles de la figure 4.3(a), mais une autre permutation (ordre des clients et ordre de tables). Par exemple $\sigma(3) = 2$ signifie que un client entre dans le restaurant en troisième sur la figure 4.3(a) et en deuxième sur la figure 4.3(b). La probabilité calculée avec la permutation de la figure 4.3(b) est :

$$\mathrm{P}(\{\{\sigma(6),\sigma(4)\},\{\sigma(3)\},\{\sigma(5),\sigma(1),\sigma(2)\}\}) = \frac{\alpha}{\alpha}\frac{\alpha}{\alpha+1}\frac{\alpha}{\alpha+2}\frac{1}{\alpha+3}\frac{1}{\alpha+4}\frac{2}{\alpha+5} \tag{4.21}$$

qui est égale à la probabilité dans l'équation (4.20).

En général, dans le cadre du modèle du CRP, quand on procède pour N clients , les dénominateurs sont simplement incrémentés de 1 à partir d'un α. Pour chaque choix de nouvelle table, on obtient un facteur α apparaît. Si à la fin, K clusters sont présents, on obtient un facteur de α^K. Enfin, pour chaque cluster $\mathcal{C}$ qui est présent à la fin, chaque fois qu'un client est ajouté à la table le numérateur est simplement le nombre de clients actuels à la table, de sorte que dans l'ensemble, on obtient un facteur de : $1 \times 2 \times \ldots \times (\#\mathcal{C} - 1) = (\#\mathcal{C} - 1)!$ où $\#\mathcal{C}$ est le nombre d'observations (clients) dans le cluster (la table). On obtient :

$$\mathrm{P}(C_{\sigma[1:N]}) = \frac{\alpha^K}{\prod_{n=1}^{N} \alpha + n - 1} \prod_{\mathcal{C}\in C_{\sigma[1:N]}} (\#\mathcal{C} - 1)!. \tag{4.22}$$

L'équation (4.22) montre que la probabilité d'une partition selon le processus du restaurant chinois est une fonction unique qui dépend du nombre d'observations et de l'ensemble des tailles des clusters formant la partition. Cette probabilité ne dépend pas de l'ordre des clients, ni des labels des tables. Ceci établit l'échangeabilité du CRP.

4.4 Du processus de Dirichlet au processus du restaurant chinois

Le processus du restaurant chinois peut être obtenu à partir du processus de Dirichlet. Pour rappel, soit une séquence $\theta_{1:n} = (\theta_1, \theta_2, ..., \theta_n)$ échantillonnée de façon i.i.d. selon une distribution $\mathbb{G}$. On suppose que $\mathbb{G}$ est distribuée *a priori* selon un processus de Dirichlet de distribution de base $\mathbb{G}_0$ et de facteur d'échelle α :

$$\theta_i \mid \mathbb{G} \overset{i.i.d}{\sim} \mathbb{G} \text{ avec } i = 1, \ldots, n$$
$$\mathbb{G} \sim \mathrm{DP}(\mathbb{G}_0, \alpha).$$

Alors, la distribution *a posteriori* de $\mathbb{G}$ conditionnellement à $(\theta_1, \ldots, \theta_n)$ est toujours distribuée selon un processus de Dirichlet :

$$\mathbb{G} \mid \theta_{1:n} \sim \text{DP}(\frac{\alpha}{\alpha+n}\mathbb{G}_0 + \frac{1}{\alpha+n}\sum_{i=1}^{n}\delta_{\theta_i}, \alpha+n).$$

Nous souhaitons obtenir la distribution de probabilité conditionnelle de θ_{n+1} donné $\theta_{1:n}$ en marginalisant selon $\mathbb{G}$. Ce qui nous donne la représentation de Backwell-Mac Queen [66]

$$\theta_{n+1} \mid \theta_{1:n} \sim \frac{\alpha}{\alpha+n}\mathbb{G}_0 + \frac{1}{\alpha+n}\sum_{i=1}^{n}\delta_{\theta_i}. \tag{4.23}$$

Le processus du restaurant chinois (CRP) peut être obtenu à partir du modèle d'urne de Blackwell-MacQueen en regroupant les valeurs de θ_i (boules) qui sont égales (de même couleur), et les considérant comme des clients à une même table dans le restaurant. Nous en supposons que $\mathbb{G}_0$ est lisse, de sorte que toutes les valeurs θ sont répétées en raison de la propriété discrétisation du DP et non en raison de $\mathbb{G}_0$ lui-même. Comme les valeurs des tirages sont répétées, soit $\mathcal{C}_1, \ldots, \mathcal{C}_K$ les valeurs uniques entre $(\theta_1, \theta_2, ..., \theta_n)$ et m_k le nombre de θ_i qui ont la même valeur $\mathcal{C}_k$, l'équation (4.23) peut également s'écrire de façon suivante :

$$\theta_{n+1} \mid \theta_{1:n} \sim \frac{\alpha}{\alpha+n}\mathbb{G}_0 + \frac{1}{\alpha+n}\sum_{k=1}^{K} m_k\delta_{\mathcal{C}_k}. \tag{4.24}$$

Notons que θ_{n+1} prend la valeur $\mathcal{C}_k$ avec une probabilité proportionnelle à m_k, le nombre de fois où $\mathcal{C}_k$ a déjà été tiré. Plus m_k est grand, plus cette probabilité croît. Ceci est une phénomène du type "des riches qui s'enrichissent" (*"rich get richer"*), où les grands clusters (un ensemble de θ_i avec des valeurs identiques $\mathcal{C}_k$ est considéré comme un cluster) grandissent plus vite. La probabilité que le nouvel échantillon $n+1$ soit identique à un échantillon précédent appartenant au cluster $\mathcal{C}_k$ est $\frac{m_k}{\alpha+n}$. La probabilité que le nouvel échantillon ait une autre valeur $\mathcal{C}_{K+1}$ est $\frac{\alpha}{\alpha+n}$, ce qui veut dire que le nouveau client choisit une autre table. On retrouve alors le processus du restaurant chinois.

L'équation (4.24) montre que α contrôle le nombre de clusters (tables). La probabilité que le client $n+1$ choisissant une nouvelle table est $\frac{\alpha}{\alpha+n}$, tandis que la probabilité de choisir une table occuppée est $\frac{n}{\alpha+n}$. Asymptotiquement, la probabilité de choisir une nouvelle table tend vers 0. De ce fait peu de nouvelles tables sont occupées si il y a déjà beaucoup de clients. On traduit ce comportement en regardant l'espérance du nombre de tables occupées par n clients

$$\sum_{i=1}^{n}\frac{\alpha}{\alpha+i} \approx \alpha\ln(n) \tag{4.25}$$

qui augmente logarithmiquement avec n.

4.5 Processus de Dirichlet à mélange

4.5.1 Description

Dans cette partie, on va exploiter l'intérêt d'utiliser le processus de Dirichlet dans cadre des modèles de mélange. Les données $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ sont supposées distribuées selon une densité de probabilité inconnue F

$$\mathbf{y}_i \sim F(\mathbf{y}_i), \forall i = 1, \ldots, N. \tag{4.26}$$

On peut souhaiter estimer la densité F à partir d'un jeu d'observations $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$. Dans ce cas, les processus de Dirichlet permettent de ne pas se limiter aux familles de distributions paramétriques. Cependant, on a vu précédement que les réalisations d'un processus de Dirichlet sont discrètes, ce qui ne correspond pas à une densité de probabilité. Une solution consiste à lisser la mesure discrète en utilisant un noyau $f(\mathbf{y} \mid \theta)$ [16].

$$F(\mathbf{y}) = \int_{\Theta} f(\mathbf{y} \mid \theta) \mathrm{d}\mathbb{G}(\theta). \tag{4.27}$$

Si l'on s'intéresse maintenant aux modèles de mélanges, le processus de Dirichlet $\mathbb{G}$ s'interprète comme la loi d'un modèle de mélange possédant une infinité dénombrable de composantes. Les variables $\theta \in \Theta$ représentent les paramètres latents propres au mélange. Le modèle peut être réécrit sous la forme hiérarchique suivante

$$\mathbf{y}_i \mid \theta_i \sim f(. \mid \theta_i) \tag{4.28}$$
$$\theta_i \mid \mathbb{G} \sim \mathbb{G} \tag{4.29}$$
$$\mathbb{G} \sim \mathrm{P}(\mathbb{G}) \tag{4.30}$$

Dans le cadre d'un mélange fini de densité gaussiennes par exemple, on suppose que la densité F est $\sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma_k)$ où les π_k sont des pondérations avec $\sum_{k=1}^{K} \pi_k = 1$. La densité F est entièrement caractérisée par les variables aléatoires π_k, μ_k et Σ_K inconnues avec $k = 1, \ldots, K$. Les densités forment une famille de densités paramétrisée par un nombre fini de paramètres. En ce sens, on parle de modèle paramétrique. θ=$\theta_1, \ldots, \theta_N$ sont des variables latentes où $\theta_i \in [1, K]$. Soit ϕ=$(\pi_{1:K}, \mu_{1:K}, \Sigma_{1:K})$, la densité *a priori* $p(\phi)$ de ϕ est définie dans un espace de dimension K, *i.e.* on fixe K classes (*clusters*).

Or, en pratique, K n'est pas toujours connu. Dans de nombreux cas, contraindre la densité de probabilité à prendre une certaine forme paramétrique donnée peut limiter l'inférence réalisée à partir de tels modèles [71]. C'est pourquoi on souhaite estimer aussi K en le considérant comme un hyperparamètre. Les modèles non paramétriques permettent de ne pas se restreindre à une famille finies de paramètres, tout en fournissant une pénalisation sous linéaire du nombre de composante.

4.5.2 Exemple

Reprenons l'exemple du mélange de gaussiennes, dans les modèles non paramétriques, la densité F s'écrit sous la forme $\sum_{k=1}^{\infty} \pi_k \mathcal{N}(\mu_k, \Sigma_k)$. Dans cette densité de pro-

babilité, les paramètres suivent une mesure de probabilité aléatoire issue d'un processus de Dirichlet à mélange (*Dirichlet Process Mixture, DPM*). Plus précisément, on l'appelle le processus de Dirichlet à mélange gaussiennes. Dans l'équation (4.29), les variables latentes θ sont distribuées selon $\mathbb{G}$. Dans l'équation (4.30), on choisit le processus de Dirichlet $DP(\mathbb{G}_0, \alpha)$ comme la distribution *a priori* pour $\mathbb{G}$. Dans le cadre des modèles de mélange, la distribution $\mathbb{G}_0$ est par exemple la loi Normale Wishart évoquée Section 4.1. Cela autorise théoriquement un nombre infini de clusters. En pratique le nombre de clusters sera lié au choix de α. Il est à noté que dans le cadre bayésien, α peut aussi être échantillonné. Dans la métaphore du processus du restaurant chinois, chaque table possède sa propre gaussienne. Les choix de table des clients et le nombre de tables sont donnés selon le processus du restaurant chinois. Certains d'entre eux ont des choix identiques ce qui produit le clustering. On trouve aussi la notion le modèle de mélange de processus du restaurant chinois.

L'inférence des modèles de mélange de processus de Dirichlet est souvent réalisée en utilisant les méthodes MCMC[72] ou les approches bayésiennes variationnelles [73]. Dans [74], l'algorithme EM (*Expectation-Maximization*) est proposé pour inférer les mélanges de processus de Dirichlet. Un stick-breaking prior (processus de Dirichlet) est mis sur les poids π_k. En pratique, on utilise une troncature en fixant un nombre K maximal de clusters très grand.

La figure 4.4 illustre une mélange de trois gaussiennes. A partir de ces données synthétiques, on cherche à regrouper les données. On peut penser à des méthodes de clustering par partitionnement comme K-moyennes (ou *K-means* en anglais) ou encore des méthodes de clustering par modélisation, *e.g.* l'algorithme EM. On obtient avec l'algorithme K-means ou EM dans un cadre paramétrique des résultats de clustering très variables suivant la valeur choisie de K. On peut choisir $K = 1, 2, 3$ ou 1000. Le choix de K est un problème difficile.

Pour contourner le problème du choix de K, un processus de Dirichlet à mélange de gaussiennes peut être utilisé. Les poids π_k des *clusters* sont distribués selon un processus de Dirichlet de type stick-breaking. Un algorithme EM (Expectation-Maximisation) a été proposé dans [74] pour inférer ce modèle un processus de Dirichlet à mélange de gaussiennes. En pratique, une troncature à $K = 100$ est utilisée.

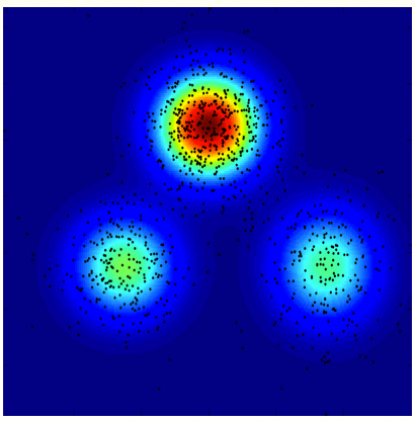
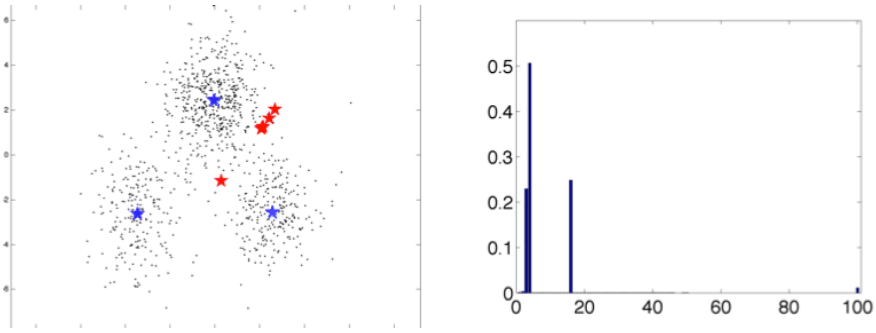


Figure 4.4 – Mélange de trois gaussiennes et échantillons



(a) Centres des clusters inférés (b) Poids des clusters

Figure 4.5 – Résultat de l'algorithme EM pour mélange de processus de Dirichlet.

Pour rappel, l'algorithme EM est un algorithme itératif proposé dans [75] en 1977. Dans le cadre des approches paramétriques, il s'agit d'une méthode d'estimation s'inscrivant dans le cadre général du maximum de vraisemblance. La figure 4.5 illustre les résultats obtenus issus du résultat de la dernière itération. La figure 4.5(a) affiche les étoiles bleues et rouges correspondant aux centres des clusters estimés. Parmi ces clusters, on retrouve bien les trois clusters souhaités en bleu. La figure 4.5(b) affiche l'estimateur des poids π_k des clusters. Les trois clusters qui ont des poids dominants [0.23 0.5 0.27] correspondent bien aux trois clusters illustrés par les étoiles bleues dans la figure 4.5(a). Les étoiles rouges représentent les autres clusters qui ont les poids négligeables, presque nuls. On peut les éliminer avec une étape de *post processing*, qui supprime par exemple tous les clusters dont les poids sont considérés comme négligeables.

4.6 Estimateurs bayésiens

Les estimateurs bayésiens d'un paramètre sont souvent construit à partir de sa loi *a posteriori* en minimisant d'une fonction de coût appropriée. Les estimateurs bayésiens proposés dans les applications de traitement du signal sont souvent :

- l'estimateur du maximum de vraisemblance ou *maximum likelihood estimation (MLE)*,
- l'estimateur **Minimum Mean Square Error, MMSE** minimise le coût quadratique. On l'appelle aussi la moyenne de la loi *a posteriori* du paramètre échantillonné,
- l'estimateur **Maximum A Posteriori, MAP** prend la forme de l'extremum de la distribution *a posteriori* du paramètre échantillonné.

Dans un cas paramétrique, les échantillons issus d'une méthode MCMC peuvent approximer ces estimateurs bayésiens. Par exemple, on peut calculer la moyenne de ces échantillons après l'étape de chauffe (*burn-in*) pour l'estimateur MMSE. Pour l'estimateur MAP, on prend l'échantillon qui maximise la loi *a posteriori* parmi les échantillons de la série. Toutefois, dans le cadre des approches non-paramétriques, la dimension de l'espace des paramètres n'est pas fixée à l'avance. Cette dimension peut varier au cours des itérations. La question de la définiton d'estimateur non-

paramétrique est une question difficile que nous n'approfondiiront pas dans ce travail.

4.7 Discussion

Le principe des approches non paramétriques est de travailler sur des mesures aléatoires. Dans le cas particulier du DPM, la loi *a priori* est une mesure aléatoire elle-même distribuée selon un processus de Dirichlet. Les modèles bayésiens non-paramétriques permettent de définir une distribution *a priori* sur des espaces fonctionnels (de dimension infinie) au lieu d'un espace de dimension finie habituellement. Un modèle non paramétrique peut être simplement considéré comme un modèle statistique avec un nombre infini de paramètres. Ceci évite de fixer la complexité ou l'ordre du modèle, le nombre de paramètres pouvant augmenter dynamiquement avec le nombre de données. Par exemple, le Processus de Dirichlet et le processus de Restaurant Chinois ont des applications en statistiques pour les modèles de mélange (*mixture models*) notamment. Le partie 4.5 nous présente l'intérêt du processus de Dirichlet dans un modèle de mélange et son inférence permet de retrouver les paramètres du modèle de mélange, incluant K le nombre de classes. Notons au passage que les problèmes de segmentation d'image se font parfois en s'appuyant sur des modèles de mélange (segmentation non supervisée ou clustering) pour lesquels des algorithmes EM ont souvent été proposés [76].

Le code de `DPpackage` de Jara [77] contient des fonctions pour effectuer l'inférence par simulation à partir des distributions *a posteriori* pour les modèles bayésiens non paramétriques. Les sources sont disponibles sous forme de package R sur le site du projet CRAN [1] [78].

Le chapitre 5 présentera le processus du buffet indien, une distribution non paramétrique utilisée dans l'apprentissage de dictionnaire, qui est aussi un des outils clefs de cette thèse.

1. https://cran.r-project.org/web/packages/DPpackage/index.html

CHAPITRE 5

Processus Beta et buffet indien

L'utilisation des méthodes bayésiennes non-paramétriques par l'apprentissage de dictionnaire nous permet de ne pas fixer à l'avance la taille du dictionnaire (nombre d'atomes). Le dictionnaire ainsi que sa taille sont échantillonnés lors de l'inférence. Chaque observation peut être une combinaison de plusieurs caractéristiques ou atomes. Il s'agit **des modèles à variables latentes** (*latent feature model*).
Or, dans les modèles de mélange ou dans les approches de classification présentées dans le chapitre 4, il n'est pas possible d'affecter plusieurs caractéristiques aux objets observés. Chaque observation ne peut être associée qu'à une seule classe. Par exemple, dans le processus du Restaurant Chinois, les clients ne peuvent s'asseoir qu'à une seule table.
Le processus du buffet indien introduit par Griffiths et Ghahramani [57, 58] en 2006 utilise les idées sous-jacentes des modèles de mélange infinis pour représenter les objets en termes d'**une infinité de fonctions latentes**. Dans le processus du buffet indien, les clients entrent dans un restaurant, mais au lieu de choisir une table à laquelle s'asseoir, ils choisissent les plats. Chaque client prend à la fois des plats déjà pris et des plats nouveaux. Le nombre total de plats choisis suit une loi Poisson.
Ce chapitre rapelle la construction du buffet indien et ses principes propriétés, ainsi que le lien avec les processus Beta.

5.1 Modèle à variables latentes et Processus du buffet indien

Les modèles à variables latentes (*latent feature models*) s'appliquent aux problèmes composés de plusieurs observations, où chacune des observations peut posséder un ensemble de caractéristiques (*features*) inconnues. L'affectation des caractéristiques aux observations est encodée par les variables latentes binaires. Ceci est représenté par une matrice binaire $\mathbf{Z}$. Dans ce chapitre, pour être cohérent avec les notations de Griffiths et Ghahramani [57, 58], la matrice $\mathbf{Z}$ est de taille $N \times K$. Les lignes de la

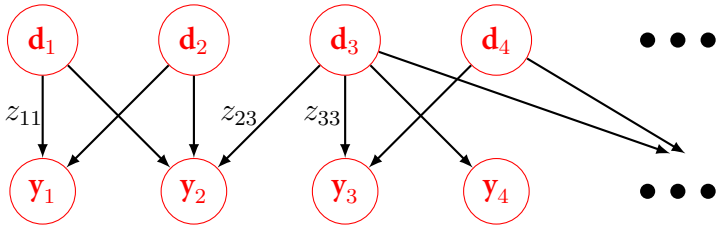


FIGURE 5.1 – Représentation graphique du modèle à variables latentes binaires. Chaque observation $\mathbf{y}_i$ est associée par une combinaison de caractéristiques $\mathbf{d}_k$.

matrice $\mathbf{Z}$ sont liées aux observations et les colonnes sont liées aux caractéristiques. Si l'observation i possède la caractéristique k alors $\mathbf{Z}(i,k)$=1, (0 si non). Notons que chaque donnée peut posséder plusieurs caractéristiques.

La figure 5.1 illustre la matrice des variables latentes comme des liaisons entre les caractéristiques et les observations. Ici, $\mathbf{d}_k$ représente la *valeur* de la $k^{\text{ième}}$ caractéristique. Une flèche reliant $\mathbf{d}_k$ à $\mathbf{y}_i$ indique que la caractéristique k est présente dans l'observation i. Les variables latentes binaires z_{ik} sont représentées par les flèches qui nous permettent de savoir si $\mathbf{d}_k$ est présente (reliée par une flèche) ou absente (non reliée) pour chaque observation $\mathbf{y}_i$. Par exemple, l'observation $\mathbf{y}_3$ contient des caractéristiques $\mathbf{d}_3$ et $\mathbf{d}_4$: la flèche relie $\mathbf{d}_3$ et $\mathbf{y}_3$ représente la variable latente $z_{33} = 1$, idem pour la flèche relie $\mathbf{d}_4$ et $\mathbf{y}_3$, $z_{34} = 1$. Par contre, il n'existe pas une flèche reliant $\mathbf{d}_1$ et $\mathbf{y}_3$, car $z_{31} = 0$.

Dans un cadre bayésien, on cherche à définir un *a priori* sur cette matrice binaire $\mathbf{Z}$. Le processus du buffet indien *(Indian Buffet Process, IBP)* a été initialement introduit par Griffiths et Ghahramani dans [57] et a été publié en revue dans [58]. L'IBP est une distribution non paramétrique sur les matrices binaires $\mathbf{Z}$ dans laquelle le nombre de caractéristiques (*features*) est potentiellement infini. Autrement dit, l'IBP peut être choisi comme *a priori* sur la matrice binaire $\mathbf{Z}$ dans le cas où le nombre de caractéristiques est inconnu. Les propriétés de l'IBP en font une distribution intéressante pour les applications aux modèles à variables latentes.

L'IBP ne limite pas le nombre de caractéristiques K. Cependant, en donnant un nombre fini N d'observations, la distribution assure que le nombre de caractéristiques K est fini avec probabilité un. Le comportement du processus est contrôlé par un seul paramètre α. Ce dernier règle l'*a priori* sur le nombre de caractéristiques observées. Le nombre de caractéristiques K attendues pour N observations est $O(\alpha \log(N))$, une valeur α faible favorisant peu de caractéristiques. Nous présentons dans la suite différentes façons d'obtenir l'IBP de paramètre α. Des versions plus générales de l'IBP à plusieurs paramètres pour contrôler en plus la popularité des caractéristiques et le comportement en loi de puissance seront décrites dans la section 5.6.

5.2 Métaphore du buffet indien

Dans la métaphore du buffet indien, les observations (données) sont symbolisées par les clients et les caractéristiques par des plats dans un buffet constitué d'une infinité de plats indiens. Le premier client qui entre dans le restaurant choisit Poisson(α) plats, qui vont constituer les premiers plat du buffet. Chaque client i choisit d'abord parmi les K premiers plats avec probabilité m_k/i où m_k est le nombre de fois où le plat k a été choisi par les clients précédents. Puis, ce client i choisit encore un nombre de nouveaux plats $k_{new} \sim$ Poisson(α/i). Cette étape permet d'enrichir progressivement l'ensemble des plats (caractéristiques) servis. Bien que le buffet soit infini, on peut montrer qu'une telle construction assure que chaque client dispose d'un nombre fini de plats avec probabilité un. Ainsi, pour un nombre fini d'observations, nous nous attendons à un nombre fini de caractéristiques. De plus, s'il y a une infinité d'observations, le nombre de caractéristiques reste dénombrable.

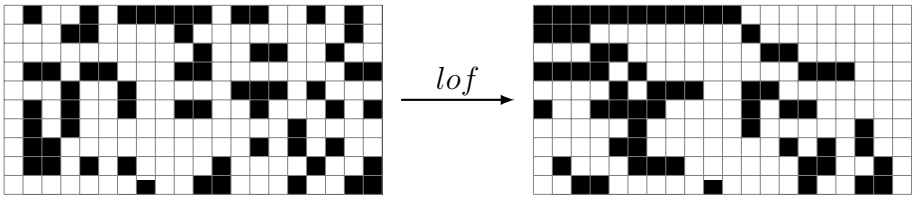


FIGURE 5.2 – Matrice binaire et sa forme ordonnée à gauche (*lof*). La couleur blanche et la couleur noire correspondent respectivement à 1 et 0.

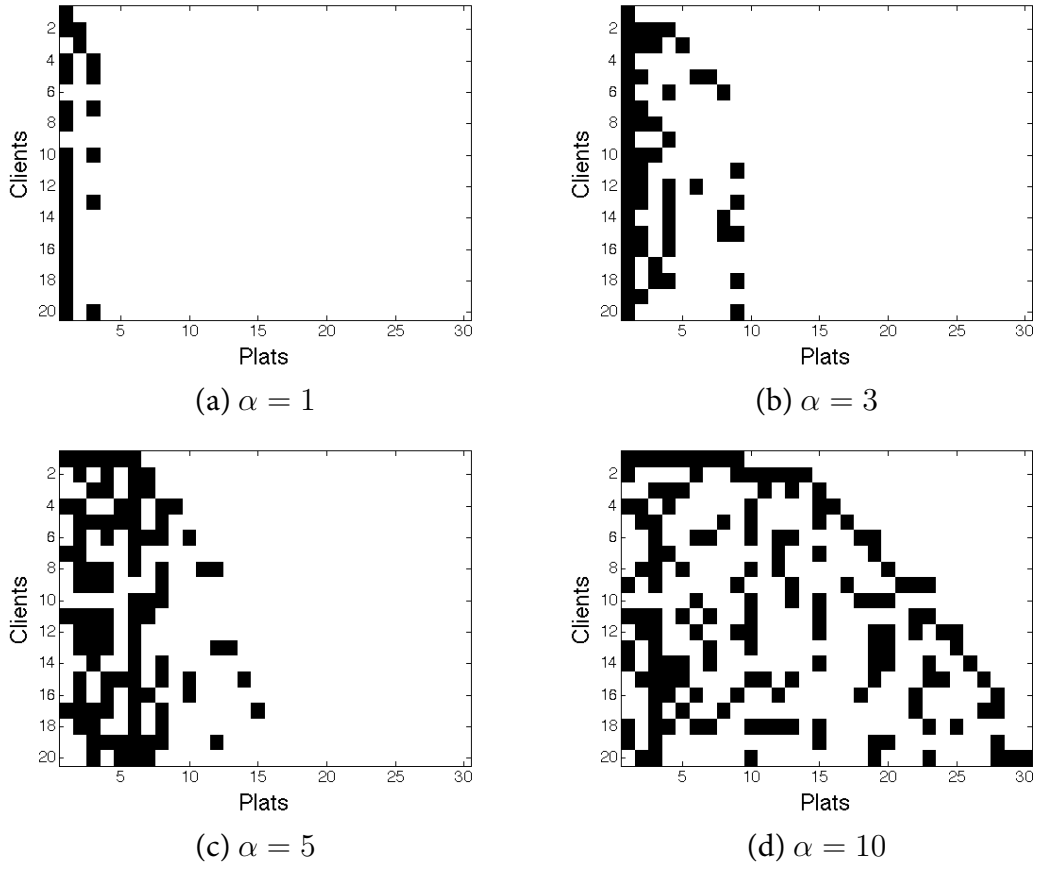
Deux propriétés importantes du processus du buffet indien sont mises en évidence. Premièrement, nous prévoyons que le nombre de plats (ou caractéristiques actives) augmente quand le nombre d'observations grandit. Deuxièmement, comme chaque client fait d'abord son choix parmi les plats choisis précédemment, nous nous attendons à ce qu'il existe quelques caractéristiques populaires qui se produisent dans de nombreuses observations et de nombreuses caractéristiques rares exprimées dans seulement quelques observations.

Une autre propriété importante du processus du buffet indien est l'échangeabilité à la fois au niveaux des lignes (clients) que des colonnes (caractéristiques latentes). L'ordre dans lequel les clients assistent au buffet n'a pas d'impact sur la distribution de $\mathbf{Z}$ sur n'importe permutation des colonnes (l'ordre des plats) et les lignes (clients) sont également indépendantes.

Griffiths et Ghahramani[57] définissent une représentation canonique appelée la forme ordonnée à gauche (*left-ordered form, lof*) de $\mathbf{Z}$, écrit $[\mathbf{Z}] = lof(\mathbf{Z})$ par la suite. La figure 5.2 montre un exemple de la fonction lof d'une matrice binaire. La forme ordonnée à gauche prend d'abord la séquence binaire de 0 et de 1 pour chaque colonne (appelée *l'histoire* h) et convertit la séquence binaire en un nombre, en traitant le premier client (ligne ou donnée) comme le bit le plus significatif. Ainsi, chaque colonne (plat ou caractéristique) reçoit une valeur unique. On organise les colonnes par ordre décroissant de valeur. Plusieurs matrices binaires peuvent avoir la même forme ordonnée à gauche. Les deux matrices $\mathbf{Z}_1$ et $\mathbf{Z}_2$ sont lof-équivalentes si $\mathbf{Z}_1$ et $\mathbf{Z}_2$ ont même forme ordonnée à gauche : $lof(\mathbf{Z}_1) = lof(\mathbf{Z}_2)$. En revanche, il n'existe qu'une seule forme ordonnée à gauche pour chaque matrice binaire. On utilise lof pour définir un ensemble de classes d'équivalence. $[\mathbf{Z}] = lof(\mathbf{Z})$ désigne la classe d'équivalence pour la relation lof d'une matrice binaire $\mathbf{Z}$.

L'IBP est caractérisée par une distribution sur les classes d'équivalence de matrices binaires [57], c'est-à-dire que la distribution sur $\mathbf{Z}$ est invariante par rapport aux permutations des colonnes (l'ordre des plats). On déduit d'ailleurs de (5.1) l'échangeabilité des clients et l'invariance de l'ordre des plats, voir **Définition 6**. La probabilité de $[\mathbf{Z}]$ est donnée par

$$\mathrm{P}([\mathbf{Z}]) = \frac{1}{\prod_{h=1}^{2^N-1} K_h!} \exp(-\alpha \sum_{i=1}^{N} \frac{1}{i}) \prod_{k=1}^{K} \frac{(N-m_k)!(m_k-1)!}{N!} \tag{5.1}$$



(a) $\alpha = 1$ (b) $\alpha = 3$

(c) $\alpha = 5$ (d) $\alpha = 10$

FIGURE 5.3 – Réalisation selon un processus du buffet indien pour 20 observations et différentes valeurs du paramètre α.

où m_k est le nombre d'observations utilisant l'atome (plat) k, K le nombre d'atomes tels que $m_k > 0$, N le nombre de données et $\alpha > 0$ le paramètre de l'IBP. En particulier, nous notons que le nombre K d'atomes actifs n'est pas borné (fixé) dans l'équation (5.1). K_h est le nombre d'atomes avec la même *histoire* $\mathbf{Z}(:, k)$=h. Autrement dit, les plats (atomes) ont été choisis par le même ensemble de client. Le paramètre α quantifie le niveau de régularisation puisque $K \sim \text{Poisson}(\alpha H_N)$ avec $H_N = \sum_{j=1}^{N} \frac{1}{j}$ ce qui donne $\mathbb{E}[K_+] \approx \alpha \ln N$ puisque $\sum_{j=1}^{N} \frac{1}{j} \underset{N\to\infty}{\sim} \ln(N)$.

La figure 5.3 illustre des réalisations de ce processus pour un même nombre d'observations et différentes valeurs du paramètre α. Plus α est petit, plus la régularisation est forte, plus le nombre de caractéristiques est petit. Pour $N = 20$ observations et α égale à $1, 3, 5, 10$, le nombre de plats K est respectivement égale à $3, 9, 15, 30$. La figure 5.3 montre aussi l'effet de la croissance logarithmique du nombre de plats K avec le nombre de clients N. Certains plats sont souvent utilisés et d'autres le sont plus rarement. Par exemple, dans la figure 5.3(d), avec $\alpha = 10$, le troisième plat est utilisé par presque tous les clients tandis qu'un seul client choisit le plat 30. Cela montre un effet parcimonieux.

En bref, l'IBP génère des matrices binaires *parcimonieuses* et *potentiellement infinies.* Autrement dit, dans le cadre de l'apprentissage de dictionnaire, l'IBP permet à

la fois de varier la taille du dictionnaire (potentiellement infinie mais pénalisée) et de promouvoir la parcimonie de la représentation. En pratique, **Z** est de taille finie car il suffit de travailler sur les K atomes actifs, c'est-à-dire les atomes qui sont associés à au moins une observation.

5.3 Limite à l'infini d'un modèle fini

Griffiths et Ghahramani [57, 58] présentent aussi le processus du buffet indien comme la limite infinie d'un modèle fini avec K caractéristiques latentes. Le modèle fini attribue une probabilité π_k pour chaque caractéristique k. Chaque $\mathbf{Z}(i,k)$ est échantillonné comme une variable aléatoire indépendante d'une loi Bernoulli de paramètre π_k. Selon la métaphore dans 5.2, chaque client choisit maintenant un plat indépendamment des autres clients, donc, l'ordre des observations n'a pas d'impact sur la distribution de **Z**. La propriété d'échangeabilité du processus s'énonce plus clairement dans cette construction par la limite à l'infini. La probabilité d'une matrice **Z** sachant $\pi = \{\pi_1, \pi_2, ..., \pi_K\}$, est :

$$\mathrm{P}(\mathbf{Z} \mid \pi) = \prod_{k=1}^{K} \prod_{i=1}^{N} P(z_{ik} \mid \pi_k) = \prod_{k=1}^{K} \pi_k^{m_k} (1-\pi_k)^{N-m_k} \tag{5.2}$$

où m_k est le nombre de clients choisissant le plat k. Cependant, le nombre de paramètres π_k croît linéairement avec K, le nombre de plats. Pour résoudre ce problème, on peut considérer également les π_k comme des variables aléatoires. Chaque π_k suit la loi Bêta(r, c) qui est la loi conjuguée de la loi de Bernoulli (*cf.* Table 2.1). En particulier, les π_k peuvent être construites à partir d'une loi Bêta$(\frac{\alpha}{K}, 1)$. Le modèle de probabilité est maintenant défini par :

$$\pi_k \mid \alpha \sim \text{Bêta}(\frac{\alpha}{K}, 1) \tag{5.3}$$

$$z_{ik} \mid \pi_k \sim \text{Bernoulli}(\pi_k). \tag{5.4}$$

Notons que les π_k ne somment pas à 1.

Chaque z_{ik} est échantillonné indépendamment des autres, sachant π_k associé. Les π_k sont également générés de façon indépendante. Après avoir défini une loi *a priori*

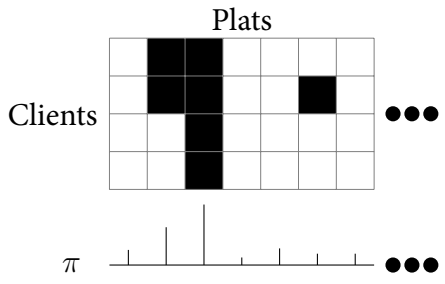


Figure 5.4 – Illustration de modèle génératif de l'IBP avec la limite à l'infini. Puisque les plats ne sont pas ordonnés par leur popularité, les plats initiaux peuvent n'être utilisés par aucun client. Les cases noires désignent l'utilisation d'un plat par un client.

sur π, nous pouvons simplifier ce modèle en intégrant sur toutes les valeurs de π_k. Ainsi, la probabilité marginale de la matrice binaire $\mathbf{Z}$ est :

$$\begin{aligned} \mathrm{P}(\mathbf{Z}) &= \prod_{k=1}^{K} \int \left(\prod_{i=1}^{N} P(z_{ik} \mid \pi_k) \right) p(\pi_k) d\pi_k \\ &= \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}. \end{aligned} \tag{5.5}$$

Sous le modèle fini, la probabilité de la forme ordonnée à gauche *left-ordered form (lof)* $[\mathbf{Z}]$ d'une matrice binaire $\mathbf{Z}$ est donnée par

$$\mathrm{P}([\mathbf{Z}]) = \frac{K!}{\prod_{h=0}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(N - m_k + 1)\Gamma(m_k + \frac{\alpha}{K})}{\Gamma(N + 1 + \frac{\alpha}{K})}. \tag{5.6}$$

où m_k est le nombre d'observations utilisant l'atome k, K_h est le nombre d'atomes avec la même *histoire* $\mathbf{Z}(:, k)$=h. La limite de l'équation (5.6) quand K tend vers l'infini devient l'équation (5.1) qui est la distribution de l'IBP.

On a $\pi_k \sim$ Bêta$(\frac{\alpha}{K}, 1)$, l'espérance $\mathbb{E}[\pi_k] = \frac{\alpha}{\alpha+K}$ tend vers 0 quand K est grand. Intuitivement, le poids π_k de chaque caractéristique est faible quand K est grand, cependant, quelques poids restent toujours grands.

La figure 5.4 montre une illustration du processus. La plupart des probabilités des caractéristiques π_k sont petites, très peu de caractéristiques sont exprimées dans l'ensemble de données fini. Comme chaque π_k est tiré de façon indépendante, chacune des (infinies) colonnes sont susceptibles d'être l'une des caractéristiques les plus populaires.

5.4 Stick-breaking

La construction de l'IBP par la limite à l'infini n'ordonne pas les caractéristiques. En effet, comme présenté dans la section 5.3, on voit bien que les caractéristiques ayant les faibles probabilités peuvent *se situer* devant celles ayant les grandes probabilités π_k, c'est pourquoi dans la figure 5.4, certains plats (caractéristiques) initiaux peuvent n'être utilisés par aucun client (observations). Teh et al. [79] déduisent une construction similaire qui trie les caractéristiques par ordre de leur popularité, ce qu'on appelle la construction par stick-breaking. Comme la construction par la limite à l'infini, la construction par stick-breaking attribue une probabilité π_k à chaque plat (colonne). Chaque client i choisit le plat k avec une probabilité π_k, *i.e.* $z_{ik} = \mathbf{Z}(i, k)$ est une variable aléatoire indépendante d'une loi Bernoulli de paramètre π_k.

Cependant, les π_k ne sont pas échantillonnées indépendamment comme dans la construction par la limite à l'infini. Au lieu de cela, une séquence de variables aléatoires indépendantes $\nu_1, \nu_2, \ldots$ est d'abord échantillonnée selon une loi Bêta$(1, \alpha)$.

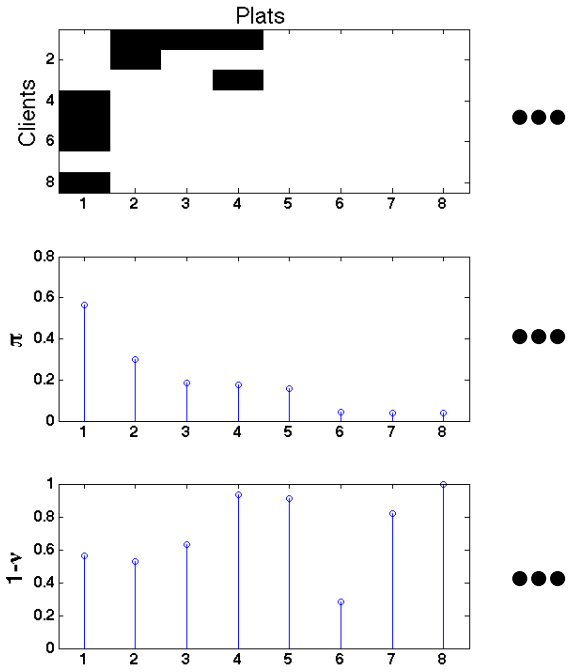


FIGURE 5.5 – Illustration une réalisation de l'IBP en utilisant la construction stick-breaking pour 8 clients (observations) et $\alpha = 2$.

En donnant l'ensemble de $\nu_1, \nu_2, \ldots$, la probabilité π_k d'un plat k est donnée par :

$$\pi_k = \prod_{\ell=1}^{k} (1 - \nu_\ell). \tag{5.7}$$

Remarque : *Il est à noter qu'ici, la somme des π_k dans la construction de l'IBP par stick-breaking n'est pas égale à 1 et que les π_k sont organisées par ordre décroissant. En comparant avec la construction stick-breaking du Processus de Dirichlet, présentée dans le chapitre 4, partie 4.2.5, la somme des π_k est égale à un mais il n'existe pas un ordre pour les π_k dans cette construction.*

La figure 5.5 illustre une réalisation de l'IBP via stick-breaking. On voit bien que $\pi_k > \pi_{k+1}$, car l'espérance de π_k est $\mathbb{E}[\pi_k] = \left(\frac{\alpha}{\alpha+1}\right)^k$. La probabilité qu'une des N observations soit décrite par la caractéristique k décroît exponentiellement avec k. La valeur de α joue aussi une rôle dans la décroissance de l'espérance. Les π_k diminue plus lentement dans l'espérance quand α est grand. Ainsi, plus la valeur de α est grande, plus le nombre de caractéristiques pour décrire les données sera grand.

5.5 Processus de Beta-Bernoulli

L'intérêt principal des méthodes bayésiennes non paramétriques est leur capacité à définir des distributions *a priori* sur les mesures aléatoires. Ces mesures aléatoires apparaissent comme des outils intéressants pour travailler avec des paramètres qui peuvent appartenir à des espaces de dimension potentiellement infinie. Par exemple,

on a vu dans le chapitre 4 que le processus de Dirichlet permet de traiter des problèmes de classification sans connaître le nombre de classes à l'avance.

Le processus du buffet indien (*IBP*) est une distribution non paramétrique sur les matrices binaires qui permet de traiter le problème de caractéristique latente (*latent feature*) sans connaître à l'avance le nombre de caractéristiques. La matrice binaire $\mathbf{Z}$ générée par l'IBP peut être présentée avec un nombre potentiellement infini de colonnes (caractéristiques). Modulo un ré-ordonnancement des colonnes, le processus du buffet indien peut être vu comme une distribution échangeable sur des matrices binaires pour n'importe quelle permutation de lignes (observations) σ, c'est-à-dire $\mathrm{P}(\mathbf{z}_{1,:}, \ldots, \mathbf{z}_{:,N})) = \mathrm{P}(\mathbf{z}_{\sigma(1),:}, \ldots, \mathbf{z}_{\sigma(N),:}))$.

En tenant compte de la propriété d'échangeabilité, Thibaux et Jordan [80] formalise l'IBP en étudiant la distribution sous-jacente qui rend la séquence conditionnellement indépendante, l'équivalent du processus de Dirichlet pour le processus du restaurant chinois. Le théorème de De Finetti indique que la distribution de n'importe quelle séquence infiniment échangeable peut s'écrire de façon suivante :

$$\mathrm{P}(\mathbf{z}_{1,:}, \ldots, \mathbf{z}_{:,N}) = \int \left(\prod_{i=1}^{N} \mathrm{P}(\mathbf{z}_{1,:} \mid B) \right) \mathrm{dP}(B) \tag{5.8}$$

où B est l'élément aléatoire qui rend les variables $\{\mathbf{z}_{1,:}\}$ conditionnellement indépendante. La distribution $\mathrm{P}(B)$ est connue comme *la distribution de mélange De Finetti*. Par exemple, pour le processus du restaurant chinois 4.3, la distribution de mélange de Finetti sous-jacente est le processus de Dirichlet.

Thibaux et Jordan [80] ont montré que le mélange de distribution de Finetti sous-jascent au processus de buffet indien est *le processus beta*. Ils montrent que IBP peut être obtenu en intégrant sur le processus Bêta dans un *processus de Bêta-Bernoulli*.

5.5.1 Processus Bêta

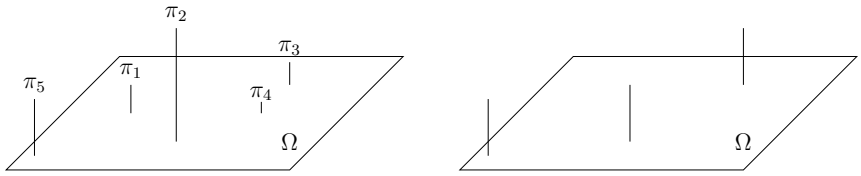
Nous étudions dans cette partie le processus Bêta, mesure de De Finetti associée à l'IBP. Comme évoqué dans le chapitre 4, le principe d'une approche bayésienne non paramétrique est d'introduire des modèles de lois *a priori* sur des espaces de mesures aléatoires. Ici, on travaille sur la mesure de probabilité aléatoires.

$$B = \sum_{k} \pi_k \delta_{\omega_k} \tag{5.9}$$

Définition 7. *Un processus Bêta B, noté $B \sim BP(c, B_0)$ est une distribution sur les mesures aléatoires positives sur un espace Ω (par exemple, sur $\mathbb{R}$). Le processus Bêta est défini par deux paramètres : c est une fonction positive sur Ω, appelée la fonction de concentration, et B_0 est une mesure fixe sur Ω, appelée la mesure de base. Dans le cas particulier où c est une constante, il sera appelé le paramètre de concentration.*

Définition 8. *Un processus stochastique à temps continu $L = \{L_t : t \geq 0\}$ est appelé processus de Lévy, si :*

— *L_0=0 presque sûrement,*



Processus Beta sur l'espace Ω Tirage selon processus Bernoulli associé

FIGURE 5.6 – Illustration de modèle Bêta-Bernoulli pour l'IBP

- $t \mapsto L_t$ *est presque sûrement continue à droite et limitée à gauche (Càdlàg),*
- *Pour tout* $s < t$, $X_t - X_s$ *est égale en loi à* X_{t-s} *(accroissements stationnaires),*
- *Pour tout* $0 \leq t_1 < t_2 < \cdots < t_n < \infty$, $X_{t_2} - X_{t_1}, X_{t_3} - X_{t_2}, \ldots, X_{t_n} - X_{t_{n-1}}$ *sont indépendants (accroissements indépendants).*

Un processus de bêta est un type particulier du processus de Lévy :

$$S \cap R = \varnothing \Longrightarrow B(S) \text{ et } B(R) \text{ sont indépendants .} \tag{5.10}$$

Le théorème de Lévy-Khintchine indique qu'un processus Lévy se caractérise par sa mesure de Lévy. La mesure Lévy du processus Bêta $BP(c, B_0)$ est :

$$\mu(\mathrm{d}\omega, \mathrm{d}\pi) = c(\omega)\pi^{-1}(1-\pi)^{c(\omega)-1}\mathrm{d}\pi B_0(\mathrm{d}\omega). \tag{5.11}$$

La mesure Lévy constitue une mesure sur $\Omega \times [0, 1]$, où Ω est par exemple l'espace des atomes (ou des caractéristiques), et $[0, 1]$ est l'espace des poids associés à ces atomes. Pour échantillonner B selon un processus Bêta, on tire un ensemble de couples (ω_k, π_k) à partir d'un processus de Poisson marqué sur $\Omega \times [0, 1]$, qui peut être représenté par :

$$B = \sum_k \pi_k \delta_{\omega_k} \tag{5.12}$$

où δ_{ω_k} est un Dirac sur un atome ω_k avec un poids π_k dans B. Les valeurs des π_k ne sont pas normalisées et la somme n'est en général pas égale à un. Cependant, elle est finie avec probabilité un puisque $\int \pi\mu(\mathrm{d}\omega, \mathrm{d}\pi) < \infty$.

Si B_0 contient des atomes, alors on peut les traiter séparément. Dans le cas où la mesure de base B_0 est une distribution discrète composée de couples atome-poids (ω_k, q_k) : $B_0 = \sum_k q_k \delta_{\omega_k}$ alors B a des atomes aux mêmes endroits $B = \sum_k \pi_k \delta_{\omega_k}$ avec $\pi_k \sim \text{Beta}(q_k, c(1 - q_k))$. Cela impose $q_k \in [0, 1]$. Et si B_0 est un mélange de lois continue et discrètes, via eq. (5.10), la réalisation globale de B est une somme des atomes pondérés issus de la composante continue et de la composante discrète de B_0. Par la suite B_0 sera choisie en lien avec la loi *a priori* conjuguée sur les atomes du dictionnaire.

5.5.2 Processus Bernoulli

On définit maintenant le processus de Bernoulli. Le processus Bernoulli correspondant échantillonne des atomes de B. Chaque tirage d'un processus de Bernoulli s'interprète comme un choix d'atomes (ou des caractéristiques) présents dans une observation particulière. Plus formellement, un processus Bernoulli $\mathbf{z} \sim BeP(B)$ prend la distribution B sur Ω comme mesure de base.

Si B est discrète, alors $B = \sum_k \pi_k \delta_{\omega_k}$ et $\mathbf{z} \sim BeP(B)$ sera de la forme :

$$\mathbf{z} = \sum_k b_k \delta_{\omega_k} \tag{5.13}$$

où $b_k \in \{0, 1\}$ est une variable aléatoire indépendante suivant une loi de Bernoulli de paramètre π_k. Si B est continue, $\mathbf{z}$ est tout simplement un processus de Poisson d'intensité B. Un processus de Bernoulli est également un type particulier de processus de Lévy. En ce qui concerne le processus Bêta, un processus de Bernoulli de mesure discrète et continue est la somme des deux contributions indépendantes. Ainsi, un tirage de $\mathbf{z}$ selon un processus de Bernoulli est un ensemble de caractéristiques. La concaténation de plusieurs tirages de $\mathbf{z}$ dans une matrice $\mathbf{Z}$ nous donne une matrice binaire qui représente l'affectation des caractéristiques.

Nous pouvons intuitivement voir Ω comme un espace de caractéristiques potentielles et $\mathbf{z}$ comme définissant les caractéristiques qu'un objet possède. La mesure aléatoire B encode la probabilité que $\mathbf{z}$ possède chaque caractéristique particulière. Dans la métaphore du buffet indien, $\mathbf{z}$ est un client et ses caractéristiques sont les plats qu'il goûte. Plus tard ces plats seront les atomes d'une dictionnaire.

La figure 5.6 illustre une visualisation de B comme un ensemble de poids π_k associés à des atomes ω_k de l'espace Ω, ici le plan.

5.5.3 Processus de Bêta-Bernoulli et processus du buffet indien

Le processus Bêta et le processus Bernoulli sont conjugués.

$$B \mid c, B_0 \sim \text{BP}(c, B_0) \tag{5.14}$$
$$\mathbf{z}_{i,:} \mid B \sim \text{BeP}(B) \text{ for } i = 1 \ldots, N \tag{5.15}$$

où $\mathbf{z}_{1,:}, \ldots, \mathbf{z}_{N,:}$ sont conditionnellement indépendants sachant B. La distribution *a posteriori* de B est aussi un processus Bêta :

$$B \mid \mathbf{z}_{1,:}, \ldots, \mathbf{z}_{N,:}, c, B_0 \sim \text{BP}\left(c + N, \frac{c}{c+N} B_0 + \frac{1}{c+N} \sum_{i=1}^{N} \mathbf{z}_{i,:}\right). \tag{5.16}$$

Une marginalisation sur B nous donne le lien avec le processus de buffet indien [80]. La distribution de $\mathbf{z}_{N+1,:}$ sachant $\{\mathbf{z}_{i,:}\}_{i=1\ldots N}$, c et B_0 est :

$$\mathbf{z}_{N+1,:} \mid \{\mathbf{z}_{i,:}\}_{i=1\ldots N}, c, B_0 \sim \text{BeP}\left(\frac{c}{c+N} B_0 + \frac{1}{c+N} \sum_{i=1}^{N} \mathbf{z}_{i,:}\right). \tag{5.17}$$

On retrouve implicitement dans l'équation (5.17) la quantité notion m_k, le nombre de fois où l'atome k a été sélectionné, car $\sum_{i=1}^{N} \mathbf{z}_{i,:} = \sum_k m_k \delta_{\omega_k}$.

Afin de faire le lien avec le processus du buffet indien, on suppose d'abord que c est une constante et B_0 est continue avec une masse totale finie valant $B_0(\Omega) = \alpha$. On observe maintenant ce qui se passe quand on génère les $\mathbf{z}$ en utilisant séquentiellement l'équation (5.17). $\mathbf{z}_{1,:} \sim \text{BeP}(B_0)$ et B_0 est continue, $\mathbf{z}_{1,:}$ est donc un processus de Poisson d'intensité B_0. En particulier, le nombre total de caractéristiques de $\mathbf{z}_{1,:}$ est $\mathbf{z}_{1,:}(\Omega) \sim \mathcal{P}(\alpha)$. Cela correspond au premier client essayant un nombre $\mathbf{z}_{1,:}(\Omega) \sim \mathcal{P}(\alpha)$ de plats. Comme on a vu dans la partie ci-dessus (5.5.2), un processus de Bernoulli est également un type particulier de processus de Lévy. Un processus de Bernoulli de mesure discrète et continue est la somme de ces deux contributions indépendantes. La mesure de base de l'équation (5.17) contient ces deux parties : continues et discrètes. On va réécrire $\mathbf{z}_{N+1,:}$ de l'équation (5.17) sous la forme d'une somme de deux processus de Bernoulli indépendants.

$$\mathbf{z}_{N+1,:} = U + V \tag{5.18}$$

où $U \sim \text{BeP}(\frac{c}{c+N} B_0)$ est un processus de Poisson d'intensité $\frac{c}{c+N} B_0$, générant un nombre de $\mathcal{P}(\frac{c}{c+N}\alpha)$ nouvelles caractéristiques; $V \sim \text{BeP}(\frac{1}{c+N} \sum_k m_k \delta_{\omega_k})$ a un atome à ω_k avec une probabilité $\frac{m_k}{c+N}$ où m_k est le nombre d'observations parmi N observations précédentes ayant choisi l'atome k. On retrouve alors le processus du buffet indien en fixant le paramètre de concentration à 1.

Remarque : *On retrouve le choix de $c = 1$ dans la construction de l'IBP par la limite à l'infini (voire section 5.3). Si on ne fixe pas l'hyperparamètre c de la loi Bêta conjuguée de la loi Bernoulli, l'équation (5.3) devient :*

$$\pi_k \mid \alpha, c \sim \textit{Beta}(\frac{\alpha c}{K}, c) \tag{5.19}$$

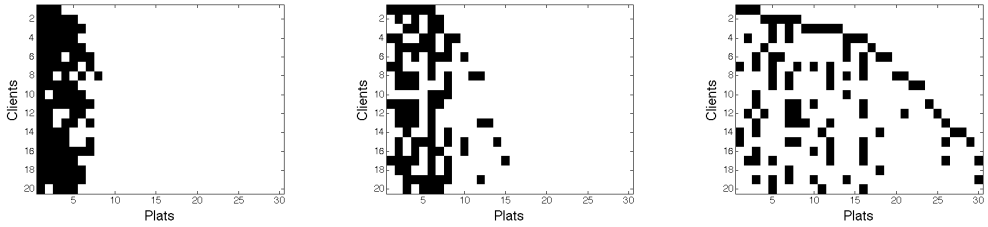
qui mène à des modèles plus généraux;

5.6 Modèles plus généraux de processus du buffet indien

5.6.1 Le processus du buffet indien à deux paramètres

Une extension de l'IBP à deux paramètres est présentée dans [81] en tenant en compte le paramètre de concentration c. Dans ce modèle, le client i choisit le plat k parmi les plats précédents avec probabilité $\frac{m_k}{i-1+c}$, puis Poisson($\frac{\alpha c}{i-1+c}$) nouveaux plats. Pour rappel, m_k est le nombre de fois où le plat k a été choisi par les clients précédents. On reprend la construction de l'IBP par la limite à l'infini en utilisant l'équation (5.19). La distribution jointe de la matrice $\mathbf{Z}$, quand K est fini, dans l'équation (5.5) devient :

$$\text{P}(\mathbf{Z}) = \prod_{k=1}^{K} \frac{\Gamma(\frac{\alpha c}{K} + c)}{\Gamma\frac{\alpha c}{K}\Gamma c} \frac{\Gamma(m_k + \frac{\alpha c}{K})\Gamma(N - m_k + c)}{\Gamma(N + c + \frac{\alpha c}{K})}. \tag{5.20}$$



(a) $c = 0.2$ (b) $c = 1$ (c) $c = 2.5$

Figure 5.7 – Réalisation selon un processus du buffet indien à deux paramètres pour 20 observations avec α=5 et (a) $c = 0.2$, (b) $c = 1$, (c) $c = 2.5$.

La distribution de probabilité correspondante sur les classes d'équivalence $[\mathbf{Z}]$, quand $K \to \infty$, est donnée par :

$$\mathrm{P}([\mathbf{Z}]) = \frac{(\alpha c)^K}{\prod_{h=1}^{2^N-1} K_h!} \exp(-\alpha c \sum_{i=1}^{N} \frac{1}{c+j-1}) \prod_{k=1}^{K} \beta(m_k, N - m_k + c) \qquad (5.21)$$

En fixant $c = 1$, on retrouve l'équation (5.1). L'espérance du nombre de caractéristique K est $\mathbb{E}[K] = \left(\alpha c \sum_{i=1}^{N} \frac{1}{i-1+c}\right)$. Pour un c fini et pour un nombre d'observations N très grand, le comportement asymptotique de K est $K \approx \alpha c \log(N)$. Cependant, si $c \gg 1$, le régime logarithmique est précédé par une croissance linéaire pour un nombre d'observations $N < c$.

Le paramètre α de l'IBP contrôle à la fois le nombre total K de caractéristiques et le nombre de caractéristiques possédées par l'objet (la parcimonie). L'IBP à deux paramètres a une flexibilité supplémentaire. Ces deux critères de la matrice $\mathbf{Z}$ sont contrôlés indépendamment . Le paramètre α contrôle le nombre total K de caractéristiques et c régularise la parcimonie. Plus c est grand, plus la matrice $\mathbf{Z}$ est parcimonieuse.

La figure 5.7 illustre trois matrices $\mathbf{Z}$ distribuées selon un processus du buffet indien à deux paramètres pour $N = 20$ avec le même α=5 et différents paramètres $c = 0.2$, 1 et 2.5 respectivement. Les trois matrices ont presque le même nombre d'éléments non nuls, mais le nombre K de caractéristiques utilisées varie considérablement. Pour la figure 5.7(a), on a 105 éléments non-nul pour $K = 8$, pour la figure5.7(b) 102 éléments non-nul avec $K = 15$ et sur la figure 5.7(c) $K = 30$ avec 106 éléments non-nul. Plus c est petit, plus la densité de chaque caractéristique est élevée. Plus c est grand, moins les caractéristiques sont utilisées.

5.6.2 Le processus du buffet indien à trois paramètres

La partie 5.5.3 a présenté le processus du buffet indien comme la marginale du processus Bêta. On peut établir une lien avec le chapitre 4. Dans la partie 4.4, le processus de Restaurant Chinois est la marginale du processus de Dirichlet. Pour

rappel, le processus de Dirichlet permet de générer un nombre potentiellement infini de *clusters*, et la vitesse à laquelle les nouveaux *clusters* sont générés est relativement lente. Or, de nombreux phénomènes sont caractérisés par des distributions en loi de puissance [82]. Ces lois de puissance se retrouvent notamment entre les fréquences d'apparition des différentes occurrences d'un phénomène et le rang de ces occurrences dans une suite ordonnée. La généralisation du processus de Dirichlet s'appelant le processus Pitman-Yor (PYP, *cf.* 4.2.5) permet de retrouver un comportement de la loi de puissance [69, 70]. Grâce à ses propriétés sur la loi de puissance, le PYP a eu de nombreuses applications, notamment, dans la modélisation des différents phénomènes linguistiques [83, 84], la segmentation d'image [85], et l'analyse PET [86].

Dans [87], Teh et Görür généralisent le processus bêta au processus stable-bêta et trouvent des relations intéressantes entre le processus stable-bêta et le processus Pitman-Yor. Une version de l'IBP à trois paramètres est proposée l'IBP en utilisant le processus stable-bêta. Dans cette extension, le comportement en loi de puissance de la fréquence d'utilisation des plats est contrôlé en plus via ϱ, un troisième paramètre de la stabilité (exposant). En utilisant la métaphore habituelle des clients entrant dans un restaurant du buffet indien et choisissant séquentiellement les plats à partir d'infinité de plats, la généralisation avec des paramètres $\alpha > 0$, $c > -\varrho$ et $\varrho \in [0, 1)$ se présente comme suit :

1. Le premier client choisit Poisson(α) plats .
2. Le client $n + 1$ choisit :
   - le plat k parmi les plats précédents avec probabilité $\frac{m_k - \varrho}{n + c}$ où $m_k > 0$ est le nombre de clients ayant déjà choisi le plat k.
   - Poisson($\alpha \frac{\Gamma(1+c)\Gamma(n+c+\varrho)}{\Gamma(n+1+c)\Gamma(c+\varrho)}$) nouveaux plats.

Quand $\varrho = 0$, on retrouve l'IBP à 2 paramètres.

5.7 Discussion

Au cours du chapitre 4 et 5, nous avons présenté deux types de distributions non paramétriques. Le processus de Dirichlet et sa marginale, le processus du restaurant chinois , qui ont des applications en statistiques dans ce qu'on appelle les modèles de mélange. Dans la famille des modèles à variables latentes, on trouve le processus Bêta ainsi que son dérivé, le processus du buffet indien.

Ce chapitre décrit quatre manières de construire le processus de buffet indien de paramètre α. Dans la construction utilisant la métaphore du buffet indien, les probabilités π_k des caractéristiques sont marginalisées. Dans les autres constructions, l'apparence des π_k est plus claire, et une procédure générative est décrite pour échantillonner les π_k. Dans la pratique, la construction par la métaphore et celle par stick-breaking ont tendance à être plus utilisées pour l'inférence. Cependant, les constructions par la limite à l'infini et le processus de Bêta-Bernoulli présentent explicitement l'échangeabilité et l'indépendance de la distribution. En passant par la construction par le processus Bêta-Bernoulli, une extension de l'IBP avec 2 paramètres est présentée dans [81]. Une autre version de l'IBP à trois paramètres a été introduite [87] : elle

permet de prendre en compte le comportement en loi de puissance de la fréquence d'utilisation des plats (atomes).

Retenons que le principe d'une approche bayésienne non paramétrique pour construire des modèles à nombre de degrés de liberté potentiellement infini est d'introduire des modèles de lois *a priori* sur des espaces de mesures aléatoires. Nous avons notamment détaillé les processus de Dirichlet et les processus Bêta. Ces approches fournissent une alternative à la sélection de modèle. Nous nous en servirons pour apprendre des dictionnaires de taille adaptative. Le chapitre 6 présente l'inférence de $\mathbf{Z}$ dont la loi a priori est de type IBP(α).

CHAPITRE 6

Inférence et processus du Buffet Indien

Dans ce chapitre, on examine le problème de l'inférence de $\mathbf{Z}$ distribuée selon un IBP(α), processus du buffet Indien de paramètre α. Pour cela, on choisit d'abord un contexte simplifié avec un modèle linéaire gaussien à variables latentes binaires. Ensuite, on introduit dans le chapitre 7 un algorithme d'inférence de $\mathbf{Z}$ dans un cas plus général. Chaque observation dans un modèle à variables latentes peut posséder un ensemble de caractéristiques (*features*). Dans une problématique d'apprentissage de dictionnaire, **les caractéristiques seront les atomes du dictionnaire**. Ce chapitre apporte une contribution bibliographique sur l'échantillonnage correct du modèle de buffet indien. Les calculs et formules sont détaillés dans l'Annexe A.1.

6.1 Modèle linéaire gaussien à variables latentes binaires

Dans le chapitre 5, les lignes de la matrice $\mathbf{Z}$ représentaient les clients et les colonnes des plats. Notons que dans ce chapitre et les suivants, $\mathbf{Z}$ est de taille $K \times N$, les lignes représentent maintenant des plats (caractéristiques) et les colonnes représentent des clients (observations). La figure 6.1 présente le modèle linéaire gaussien à variables latentes binaires suivant :

$$\mathbf{Y} = \mathbf{DZ} + \boldsymbol{\varepsilon} \tag{6.1}$$
$$\boldsymbol{\varepsilon}_i \sim \mathcal{N}(0, \sigma_{\boldsymbol{\varepsilon}}^2 \mathbb{I}_L), \forall 1 \leq i \leq N \tag{6.2}$$
$$\mathbf{d}_k \sim \mathcal{N}(0, \sigma_D^2 \mathbb{I}_L), \forall k \in \mathbb{N} \tag{6.3}$$
$$\mathbf{Z} \sim \text{IBP}(\alpha) \tag{6.4}$$

où $\mathbf{Y} \in \mathbb{R}^{L \times N}$ contient N observations, chaque colonne de $\mathbf{Y}$ représente une observation $\mathbf{y}_i$ de dimension L. $\boldsymbol{\varepsilon} = [\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_i, \ldots, \boldsymbol{\varepsilon}_N] \in \mathbb{R}^{L \times N}$ est le bruit, $\mathbf{D} \in \mathbb{R}^{L \times K}$ s'appelle le dictionnaire, $\mathbf{Z} \in \{0,1\}^{K \times N}$ est la matrice des variables latentes binaires. Chaque colonne k de $\mathbf{D}$ est un atome $\mathbf{d}_k$ de dimension L. Chaque ligne de $\mathbf{Z}$ représente les coefficients d'un atome pour l'ensemble des observations. Le nombre d'atomes est défini par le nombre de colonnes de $\mathbf{D}$ mais aussi par le nombre de lignes de $\mathbf{Z}$. De même, $\mathbf{Z}$ est une matrice avec un nombre potentiellement infini de lignes et $\mathbf{D}$ contient aussi un nombre potentiellement infini de colonnes. Lorsqu'une

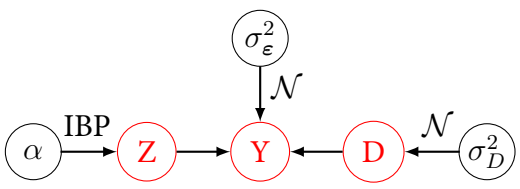


FIGURE 6.1 – Modèle linéaire gaussien avec les variables latentes binaires

ligne de **Z** ne contient que des zéros, cela revient à supprimer de **D** la colonne de l'atome correspondant à cette ligne de **Z**. Notons que le nombre K d'atomes sera contrôlé par **Z** grâce au processus du Buffet Indien et non pas par une loi *a priori* sur **D**.

Pour $i = 1 : N$
  Pour $k = 1 : K$
    Échantillonner z_{ki} : utilisation de l'atome k par l'observation i
  Échantillonner k_{inew} : nombre de nouveaux atomes proposé par i
  Mettre à jour K

Algorithme 7 : Pseudo-algorithme d'échantillonnage de $\mathbf{Z} \sim \text{IBP}(\alpha)$.

Notre objectif est de caractériser la distribution *a posteriori* $\text{p}(\mathbf{Z}|\mathbf{Y}, \mathbf{D}, \varepsilon)$ incluant implicitement le nombre d'atomes K. **Z** est une matrice avec un nombre potentiellement infini de lignes. En pratique on travaille avec les lignes non-nulles, autrement dit sur les atomes actifs. Un atome est dit actif lorsqu'au moins une observation l'utilise. Les deux étapes pour échantillonner **Z** proposées dans [57, 58] sont :

1. la mise à jour des $z_{ki} = \mathbf{Z}(k, i)$ pour les atomes k actifs ($\exists i \mid z_{ki} \neq 0$),
2. l'ajout de nouvelles lignes pour **Z** qui correspond à l'activation de nouveaux atomes dans le dictionnaire **D**.

L'algorithme 7 fait appel à ces deux étapes d'échantillonnage. Ce chapitre décrit les différents échantillonnages pour **Z** liés à ces deux étapes : l'échantillonnage de Gibbs usuel, l'échantillonnage de Gibbs marginalisé ou en anglais *Collapsed Gibbs Sampling (CGS)* [57, 58], l'échantillonnage de Gibbs marginalisé accéléré *(Accelerated Collapsed Gibbs Sampling, ACGS)* [88].

6.2 Échantillonnage de Gibbs

6.2.1 Utilisation de l'atome k par l'observation i

L'échantillonnage de Gibbs est une technique MCMC pour échantillonner suivant une distribution jointe sur plusieurs variables. Pour rappel, l'échantillonnage des atomes actifs est équivalent au choix de chaque client i devant les plats k choisis par les clients précédents dans la métaphore du buffet indien. Pour chaque observation i la distribution *a posteriori* de $\mathbf{z}_i$ est une distribution jointe sur $z_{1i}, z_{2i}, \ldots, z_{Ki}$. Dans cette approche, $\mathbf{z}_{ki}$ est échantillonné à partir de la distribution *a posteriori*

$$\text{P}(z_{ki} \mid \mathbf{y}_i, \mathbf{D}, \mathbf{z}_i(-k), \sigma_\varepsilon^2) \propto \text{p}(\mathbf{y}_i|\mathbf{D}, \mathbf{z}_i, \sigma_\varepsilon^2)\text{P}(z_{ki} \mid \mathbf{Z}_{-(ki)}). \tag{6.5}$$

z_{ki} est simulé selon une distribution Bernoulli pondérée par les vraisemblances pour chaque paire (observation i, atome k).

Comme l'IBP est une distribution échangeable, voir partie 5.2, on peut voir chaque client comme le dernier client qui arrive. Soit $m_{k,-i}$ le nombre d'observations sans compter l'observation i utilisant l'atome k. Le terme *a priori* est

$$\mathrm{P}(z_{ki} = 1|\mathbf{Z}_{-(ki)}) = \frac{m_{k,-i}}{N}. \tag{6.6}$$

La vraisemblance $\mathrm{p}(\mathbf{y}_i|\mathbf{D}, \mathbf{z}_i, \sigma_\varepsilon^2)$ est facilement calculée à partir du modèle Gaussien de la figure 6.1. Grâce à la règle de Bayes, on peut écrire :

$$\mathrm{p}(z_{ki}|\mathbf{y}_i, \mathbf{D}, \mathbf{Z}_{(-ki)}, \sigma_\varepsilon) \propto \mathcal{N}(\mathbf{y}_i|\mathbf{D}\mathbf{z}_i, \sigma_\varepsilon^2\mathbb{I}_L)\mathrm{P}(z_{ki}|\mathbf{Z}_{-(ki)}). \tag{6.7}$$

Les probabilités *a posteriori* de $z_{ki} = 0$ ou 1 sont proportionnelles à (p_0, p_1) définies par :

$$p_0 = 1 - \frac{m_{k,-i}}{N} \tag{6.8}$$

$$p_1 = \frac{m_{k,-i}}{N} \exp\left[-\frac{1}{2\sigma_\varepsilon^2}\left(\mathbf{d}_k^{\mathrm{T}}\mathbf{d}_k - 2\mathbf{d}_k^{\mathrm{T}}(\mathbf{y}_i - \sum_{j\neq k}^{K}\mathbf{d}_j z_{ji})\right)\right]. \tag{6.9}$$

z_{ki} peut être échantillonné à partir d'une distribution Bernoulli

$$z_{ki} \sim \text{Bernoulli}\left(\frac{p_1}{p_0 + p_1}\right). \tag{6.10}$$

Les calculs sont détaillés dans l'Annexe A.1.1.

6.2.2 Nombre de nouveaux atomes proposé par l'observation i

Après avoir choisi parmi les plats déjà choisis par les clients précédents, le client i choisit en plus $k_{new} \sim \text{Poisson}(\frac{\alpha}{i})$ nouveaux plats (*cf.* partie 5.2,). Chaque client est considéré comme le dernier client arrivant grâce à la propriété d'échangeabilité de l'IBP. La loi *a priori* du nombre de nouveaux plats (atomes) est :

$$k_{inew} \sim \text{Poisson}\left(\frac{\alpha}{N}\right). \tag{6.11}$$

Il est à noter que le client i provoque l'ajout de k_{inew} nouveaux plats, donc k_{inew} lignes sont ajoutées à la matrice $\mathbf{Z}$. Dans ces k_{inew} lignes, les $\mathbf{z}_{inew}$, les nouveaux éléments de la colonnes i sont tous des 1 et les k_{inew} coefficients des autres colonnes sont des 0. La loi *a posteriori* de k_{inew} peut être calculée de la manière suivante :

$$\begin{aligned}\mathrm{p}(k_{inew} \mid \mathbf{y}_i, [\mathbf{D}, \mathbf{D}_{new}], \mathbf{z}_i, \sigma_\varepsilon^2, \alpha) \\ \propto \mathcal{P}\left(k_{inew} \mid \frac{\alpha}{N}\right) \mathrm{p}(\mathbf{y}_i \mid [\mathbf{D}, \mathbf{D}_{new}], [\mathbf{z}_i; \mathbf{z}_{inew}], \sigma_\varepsilon^2).\end{aligned} \tag{6.12}$$

L'ajout de nouvelles lignes à $\mathbf{Z}$ correspond à l'ajout de nouvelles colonnes $\mathbf{D}_{new}$ à $\mathbf{D}$. On l'appelle également l'activation de nouveaux atomes du dictionnaire. Un terme $\mathbf{D}_{new}$ apparaît quand on veut échantillonner le nombre de nouveaux atomes. La loi *a priori* de nouveaux atomes est une loi Gaussienne où pour chaque $k_{ajout} \in [1 : k_{inew}]$:

$$\mathbf{d}_{ajout} \sim \mathcal{N}(0, \sigma_D^2\mathbb{I}_L). \tag{6.13}$$

Il faut simuler maintenant non seulement k_{inew} mais aussi $\mathbf{D}_{new}$:

$$\begin{aligned}&\mathrm{p}(k_{inew}, \mathbf{D}_{new} \mid \mathbf{y}_i, \mathbf{D}, \mathbf{z}_i, \sigma_\varepsilon^2, \sigma_D^2, \alpha)\\ &\propto \mathrm{p}(\mathbf{y}_i \mid [\mathbf{D}, \mathbf{D}_{new}], [\mathbf{z}_i; \mathbf{z}_{inew}], \sigma_\varepsilon^2, \sigma_D^2)\mathcal{P}\left(k_{inew} \mid \frac{\alpha}{N}\right)\mathcal{N}(\mathbf{D}_{new} \mid 0^{L\times k_{inew}}, \sigma_D^2\mathbb{I}_L).\end{aligned} \tag{6.14}$$

Cette loi jointe n'est pas une loi identifiée suivant laquelle on sait simuler directement. On fait alors appel aux méthodes dites Métropolis-Hastings (MH), *cf.* partie 2.6.4. Ces méthodes peuvent être utilisées ici pour échantillonner le nombre de nouveaux atomes ainsi que les nouveaux atomes. Comme l'échantillonnage de Gibbs, l'échantillonneur Metropolis-Hastings est une méthode MCMC qui produit une séquence d'échantillons distribués suivant une loi cible à partir d'une distribution connue. Pour simplifier, on peut choisir la loi *a priori* comme loi de proposition.

$$k_{iprop} \sim \text{Poisson}\left(\frac{\alpha}{N}\right) \tag{6.15}$$

$$\mathbf{D}_{prop} \sim \prod_{k=1}^{k_{iprop}} \mathcal{N}(0, \sigma_D^2\mathbb{I}_L). \tag{6.16}$$

Dans ce cas, l'expression de la probabilité de mouvement devient le rapport des vraisemblances :

$$a = \min\left(1, \frac{\mathrm{p}(\mathbf{y}_i \mid [\mathbf{D}, \mathbf{D}_{prop}], [\mathbf{z}_i; \mathbf{z}_{iprop}], \sigma_\varepsilon^2)}{\mathrm{p}(\mathbf{y}_i \mid \mathbf{D}, \mathbf{z}_i, \sigma_\varepsilon^2)}\right) \tag{6.17}$$

Si $a > \mathcal{U}_{[0,1]}$ alors $k_{inew} = k_{iprop}$ et $\mathbf{D}_{new} = \mathbf{D}_{prop}$: on ajoute k_{inew} nouveaux atomes. Sinon on garde l'état initial, c'est-à-dire $k_{inew} = 0$, aucun nouvel atome n'est ajouté.

6.3 Échantillonnage de Gibbs marginalisé

6.3.1 Intérêt de la marginalisation

Un échantillonneur de Gibbs pourrait suffire si le modèle a un dictionnaire (le nombre K d'atomes) de taille fixe. Or, nous souhaitons nous affranchir de cette limitation. Le modèle utilisant l'IBP comme une loi *a priori* qui permet de traiter cette restriction en fournissant un modèle avec un nombre potentiellement infini d'atomes. Non seulement le dictionnaire est échantillonné mais le nombre de ces atomes aussi. L'étape où l'on échantillonne le nombre de nouveaux atomes nécessite une étape de Metropolis-Hastings car on ne sait pas simuler directement suivant la loi jointe. Dans cette étape, les nouveaux atomes doivent être proposés aussi, voir 6.2.2. Cependant, le mélange de la chaîne peut être lent puisqu'on explore un espace de dimension $L \times k_{inew}$, la dimension de l'atome fois le nombre de nouveaux atomes, surtout si la dimension L de l'atome est grande. Une solution naturelle apportée par les méthodes bayésiennes à ce type de problème consiste à marginaliser le dictionnaire dans cette étape. Un échantillonneur de Gibbs marginalisé propose d'intégrer la loi *a posteriori* par rapport à $\mathbf{D}$ [58] afin de réduire l'espace d'état et donc le temps de mélange. Cette technique constitue un avantage des approches bayésiennes et ne possède pas d'équivalent direct dans la famille des méthodes d'optimisation.

Vraisemblance marginalisée :

On calcule facilement une vraisemblance marginalisée par rapport au dictionnaire $\mathbf{D}$. Soit $\mathbf{M} = (\mathbf{Z}\mathbf{Z}^{\mathrm{T}} + \frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I}_K)^{-1}$,

$$\begin{aligned} \mathrm{p}(\mathbf{Y} \mid \mathbf{Z}, \sigma_\varepsilon^2, \sigma_D^2) &= \int \mathrm{p}(\mathbf{Y} \mid \mathbf{D}, \mathbf{Z}, \sigma_\varepsilon^2)\mathrm{p}(\mathbf{D} \mid \sigma_D)\mathrm{d}\mathbf{D} \\ &= \frac{|\sigma_\varepsilon^2\mathbf{M}|^{L/2}}{(2\pi)^{NL/2}\sigma_\varepsilon^{NL}\sigma_D^{KL}} \exp\left\{-\frac{1}{2\sigma_\varepsilon^2}\mathrm{tr}\left[\mathbf{Y}(\mathbb{I} - \mathbf{Z}^{\mathrm{T}}\mathbf{M}\mathbf{Z})\mathbf{Y}^{\mathrm{T}}\right]\right\}. \end{aligned} \tag{6.18}$$

Les détails du calcul sont dans l'Annexe A.1.2.

Échantillonnage de z_{ki} en utilisant la vraisemblance marginalisée :

La distribution *a posteriori* de chaque z_{ki} dans l'échantillonnage de Gibbs marginalisé est la suivante :

$$\mathrm{p}(z_{ki}|\mathbf{y}_i, \mathbf{Z}_{(-ki)}, \sigma_\varepsilon^2, \sigma_D^2) \propto \mathrm{p}(\mathbf{Y} \mid \mathbf{Z}, \sigma_\varepsilon^2, \sigma_D^2)\mathrm{P}(z_{ki}|\mathbf{Z}_{-(ki)}). \tag{6.19}$$

Par conséquent, la distribution Bernoulli dans l'équation (6.10) pour échantillonner z_{ki} dépend de

$$p_0 = \left(1 - \frac{m_{k,-i}}{N}\right)\mathrm{p}(\mathbf{Y} \mid \mathbf{Z}, \sigma_\varepsilon^2, \sigma_D^2) \tag{6.20}$$

$$p_1 = \frac{m_{k,-i}}{N}\mathrm{p}(\mathbf{Y} \mid \mathbf{Z}, \sigma_\varepsilon^2, \sigma_D^2). \tag{6.21}$$

Échantillonnage de k_{inew} :

La distribution *a posteriori* marginalisée de k_{inew} est la suivante :

$$\begin{aligned} &\mathrm{p}(k_{inew} \mid \mathbf{Y}, [\mathbf{Z}; \mathbf{Z}_{inew}], \sigma_\varepsilon^2, \sigma_D^2, \alpha) \\ &\qquad \propto \mathcal{P}\left(k_{inew} \mid \frac{\alpha}{N}\right)\mathrm{p}(\mathbf{Y} \mid [\mathbf{Z}; \mathbf{Z}_{inew}], \sigma_\varepsilon^2, \sigma_D^2). \end{aligned} \tag{6.22}$$

En marginalisant le dictionnaire $\mathbf{D}$, on n'a pas besoin de proposer les nouveaux atomes. L'avantage de l'échantillonneur de Gibbs marginalisé est que l'intégration de $\mathbf{D}$ donne à l'échantillonneur un taux de mélange plus rapide (grâce au théorème de Rao-Blackwell [42]).

Dans [58], Griffiths et Ghahramani ont proposé d'échantillonner k_{inew} en tronquant la distribution du nombre de nouveaux atomes à un certain niveau de troncature k_{max}. On calcule les poids dans (6.22) pour $0 \leq k_{inew} \leq k_{max}$. Ensuite, une loi Multinomiale permet d'échantillonner k_{inew}.

Une deuxième possibilité est d'utiliser l'algorithme Metropolis-Hastings (MH) proposé dans 6.2.2 pour échantillonner k_{inew}. En utilisant la loi *a priori* comme loi de proposition, l'expression de la probabilité de mouvement devient le rapport des vraisemblances marginalisées :

$$a = \min\left(1, \frac{\mathrm{p}(\mathbf{Y} \mid [\mathbf{Z}; \mathbf{Z}_{prop}], \sigma_\varepsilon^2, \sigma_D^2)}{\mathrm{p}(\mathbf{Y} \mid \mathbf{Z}, \sigma_\varepsilon^2, \sigma_D^2))}\right). \tag{6.23}$$

Dans 6.4.4, pour échantillonner k_{inew}, on utilise aussi l'algorithme MH en utilisant la loi *a priori* comme loi de proposition.

Toutefois, une remarque dans [58] signale que l'étape d'échantillonnage k_{inew} proposée dans [57, 58] pourrait poser problème. On en discutera dans la partie 6.5, ainsi que la façon de le corriger.

Remarque : *Le calcul de la marginalisation passe par une étape de calcul de la loi a posteriori de* $\boldsymbol{D}$ *(cf. Annnexe A.1.2). La loi a posteriori d'une vraisemblance gaussienne associée à loi la a priori gaussienne est aussi une loi gaussienne :*

$$\mathrm{p}(\boldsymbol{D} \mid \boldsymbol{Y}, \boldsymbol{Z}, \sigma_D^2) \propto \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{D}|\boldsymbol{Y},\boldsymbol{Z}}, \boldsymbol{\Sigma}_{\boldsymbol{D}|\boldsymbol{Y},\boldsymbol{Z}}). \tag{6.24}$$

Une possibilité pour mettre à jour le dictionnaire serait d'utiliser l'espérance de sa loi a posteriori :

$$E[\boldsymbol{D} \mid \boldsymbol{Y}, \boldsymbol{Z}] = \boldsymbol{\mu}_{\boldsymbol{D}|\boldsymbol{Y},\boldsymbol{Z}} = \boldsymbol{Y}\boldsymbol{Z}^T\boldsymbol{M} = \boldsymbol{Y}\boldsymbol{Z}^T(\boldsymbol{Z}\boldsymbol{Z}^T + \frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I}_K)^{-1}. \tag{6.25}$$

Dans l'équation (6.22), la marginalisation de $\boldsymbol{D}$ *nous permet de négliger la proposition de nouveaux atomes. Par contre, on retrouve dans la vraisemblance marginalisée de l'équation (6.18), le terme* $\boldsymbol{Y}\boldsymbol{Z}^T(\boldsymbol{Z}\boldsymbol{Z}^T + \frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I}_K)^{-1}$ *qui joue le rôle d'un dictionnaire intermédiaire correspondant à son espérance a posteriori :*

$$\begin{aligned}\mathrm{p}(\boldsymbol{Y} \mid \boldsymbol{Z}, \sigma_\varepsilon^2, \sigma_D^2) = & \frac{1}{(2\pi)^{NL/2}\sigma_\varepsilon^{(N-K)L}\sigma_D^{KL}|\boldsymbol{Z}\boldsymbol{Z}^T + \frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I}_K|^{L/2}} \\ & \exp\left\{-\frac{1}{2\sigma_\varepsilon^2}tr[\boldsymbol{Y}(\mathbb{I}_N - \boldsymbol{Z}^T(\boldsymbol{Z}\boldsymbol{Z}^T + \frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I}_K)^{-1}\boldsymbol{Z})\boldsymbol{Y}^T]\right\}.\end{aligned} \tag{6.26}$$

6.3.2 Problèmes numériques

Les principales limitations de l'échantillonnage de Gibbs marginalisé sont sa complexité numérique et son temps de calcul [88, 89]. La durée d'exécution de l'échantillonneur de Gibbs marginalisé est dominée par le calcul dans l'exposant du terme :

$$\mathbf{Y}(\mathbb{I}_N - \mathbf{Z}^{\mathrm{T}}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}} + \frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I}_K)^{-1}\mathbf{Z})\mathbf{Y}^{\mathrm{T}}. \tag{6.27}$$

Une première difficulté de cet échantillonnage de Gibbs marginalisé est la mise à jour de $\mathbf{M} = (\mathbf{Z}\mathbf{Z}^{\mathrm{T}} + \frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I}_K)^{-1}$. Pour chaque client i choisissant chaque plat k existant, la matrice $\mathbf{M}$ doit être recalculée pour deux possibilités $z_{ki} = 1$ et $z_{ki} = 0$. L'inversion de matrice est une des opérations coûteuses. La complexité d'inverser une matrice de taille $K \times K$ en utilisant la méthode de Gauss est $O(K^3)$. Toutefois, il est à noter que $\mathbf{Z}\mathbf{Z}^{\mathrm{T}}$ peut s'exprimer comme une somme de produits de variables latentes : $\sum_{i=1}^{N} \mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}$. Une méthode de *rank-one update*[58] est proposée en utilisant le lemme d'inversion de matrice pour ajouter ou supprimer facilement l'influence d'un seul $\mathbf{z}_i$ de $\mathbf{M}$.

Soit $\mathbf{M}_{-i} = (\sum_{j\neq i}^{N} \mathbf{z}_j\mathbf{z}_j^{\mathrm{T}} + \frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I}_K)^{-1}$,

$$\mathbf{M}_{-i} = (\mathbf{M}^{-1} - \mathbf{z}_i\mathbf{z}_i^{\mathrm{T}})^{-1} = \mathbf{M} - \frac{\mathbf{M}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\mathbf{M}}{\mathbf{z}_i^{\mathrm{T}}\mathbf{M}\mathbf{z}_i - 1} \tag{6.28}$$

$$\mathbf{M} = (\mathbf{M}_{-i}^{-1} + \mathbf{z}_i\mathbf{z}_i^{\mathrm{T}})^{-1} = \mathbf{M}_{-i} - \frac{\mathbf{M}_{-i}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\mathbf{M}}{\mathbf{z}_i^{\mathrm{T}}\mathbf{M}\mathbf{z}_i + 1}. \tag{6.29}$$

Grâce à la méthode *rank-one update* la complexité de l'inverse de $(\mathbf{Z}\mathbf{Z}^{\mathrm{T}} + \frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I}_K)$ est $O(K^2)$. Cependant, il est conseillé de calculer de temps en temps l'inverse complète afin d'éviter l'accumulation d'erreurs numériques.

Même si une méthode de *rank-one update* a été proposée pour mettre à jour $\mathbf{M}$, il reste encore le calcul de $\mathbf{Y}(\mathbb{I}_N - \mathbf{Z}^{\mathrm{T}}\mathbf{M}\mathbf{Z})\mathbf{Y}^{\mathrm{T}}$ dans l'équation (6.18). Les produits de matrice restants nécessitent $O(N^2 + 2NK^2 + 2N^2L)$ opérations respectivement. Ainsi, la complexité pour NK échantillonnages de z_{ki} est $O(NK(N^2 + 2NK^2 + 2N^2L)) = O(N^3K + 2N^2K^2 + 2N^3KL)$. Le terme dominant de cette complexité est de l'ordre de $O(N^3KL)$. Ce qui ne passe pas à l'échelle, surtout quand le nombre d'observations N est très grand. Pour la résolution de problèmes inverses en traitement d'image, N est le nombre des patches extraits à partir d'une image. Par exemple, pour une image de taille 512×512, N peut être égale à 255025, voir partie 3.2.

Dans l'échantillonnage de Gibbs usuel, le calcul de la vraisemblance $\mathrm{p}(\mathbf{y}_i|\mathbf{D}, \mathbf{z}_i, \sigma_\varepsilon^2)$ est dominé par le terme $\mathbf{D}\mathbf{z}_i$ qui a une complexité de $O(KL)$. Comme il existe NK z_{ki} à échantillonner, la complexité finale est $O(NK^2L)$. Sa complexité est moins importante que celle de l'échantillonneur de Gibbs marginalisé, mais le mélange est moins rapide.

6.4 Échantillonnage de Gibbs marginalisé accéléré

6.4.1 Description générale

Dans [88], Doshi-Velez et Ghahramani ont proposé un échantillonnage de Gibbs marginalisé accéléré (*Accelerated Collapsed Gibbs Sampling, ACGS*) pour $\mathbf{Z} \sim \mathrm{IBP}(\alpha)$ dans le modèle présenté dans la partie 6.1. Cette version accélérée permet de réduire la complexité à $O(N(KL + K^2))$ [88]. L'algorithme 8 décrit les étapes de ce nouvel échantillonnage.

La motivation de ce nouvel échantillonnage est de garder la rapidité de mélange de l'échantillonnage de Gibbs marginalisé mais en réduisant sa complexité. La complexité de l'échantillonnage de Gibbs marginalisé est coûteuse à cause du calcul de la vraisemblance marginalisée (6.18) qui dépend de l'ensemble des données. Sous les conditions d'indépendance, l'ensemble des données $\mathbf{Y}$ peut être divisé en deux sous-ensembles. Un sous-ensemble contient l'observation $\mathbf{y}_i$ correspondant à $\mathbf{z}_i$ et l'autre représente le reste. Pour obtenir la complexité la plus faible $O(N(KL + K^2))$ [88], on choisit de partitionner $\mathbf{Y}$ en $[\mathbf{y}_i, \mathbf{Y}_{-i}]$ correspondant à $\mathbf{Z} = [\mathbf{z}_i; \mathbf{Z}_{-i}]$.

Utiliser la statistique suffisante pour la loi *a posteriori* de $\mathbf{D}$ via eq.(6.38)
Pour $i = 1 : N$
  Enlever l'influence de donnée i sur la distribution *a posteriori* de $\mathbf{D}$ via eq.(6.41),(6.42)
  Pour $k = 1 : K$
    Échantillonner des caractéristiques actives $\mathbf{Z}(k, i)$ via eq.(6.47),(6.48)
  Inférer le nombre de nouveaux atomes
  Mettre à jour K
  Remettre l'influence de donnée i sur la distribution *a posteriori* de $\mathbf{D}$ via eq.(6.43),(6.44)

Algorithme 8 : Échantillonnage de Gibbs marginalisé accéléré de $\mathbf{Z} \sim \text{IBP}(\alpha)$.

Dans le cas de l'échantillonnage de Gibbs marginalisé, on intègre par rapport à la variable $\mathbf{D}$. Bien que la variable $\mathbf{D}$ ne se présente pas dans la vraisemblance marginalisée (6.26), il apparaît quand même un terme contenant les données $\mathbf{Y}$ qui joue le rôle d'un dictionnaire intermédiaire. L'échantillonnage de $\mathbf{z}_i$ en utilisant la vraisemblance marginalisée doit dépendre de toutes les données $\mathbf{y}_i$ et $\mathbf{Y}_{-i}$. À l'inverse, dans le cas de l'échantillonnage de Gibbs usuel, la vraisemblance contient la variable $\mathbf{D}$ mais l'échantillonnage de $\mathbf{z}_i$ ne dépend que des données $\mathbf{y}_i$.

L'échantillonnage de Gibbs marginalisé accéléré se base sur la marginalisation par rapport à $\mathbf{D}$, par contre, il propose de mettre à jour la distribution *a posteriori* de $\mathbf{D}$ afin de pouvoir l'utiliser pour échantillonner $\mathbf{z}_i$. Pour cela, l'idée est de soustraire l'influence de i sur la distribution *a posteriori* de $\mathbf{D}$. L'échantillonnage de $\mathbf{z}_i$ dépend de $\mathbf{y}_i$ et de la distribution *a posteriori* de $\mathbf{D}$ sans l'influence de $\mathbf{y}_i$ et $\mathbf{z}_i$. Une fois $\mathbf{z}_i$ est échantillonné, on réintègre l'influence de i sur la distribution *a posteriori* de $\mathbf{D}$. Cette démarche est détaillée ci-après.

6.4.2 Échantillonnage accéléré

L'échantillonnage de Gibbs marginalisé accéléré est effectué avec *une vraisemblance quasi-marginalisée*. On commence par procéder à une marginalisation comme dans l'équation (6.18)

$$\mathrm{p}(\mathbf{Y} \mid \mathbf{Z}, \sigma_\varepsilon^2, \sigma_D^2) \propto \int \mathrm{p}(\mathbf{Y} \mid \mathbf{D}, \mathbf{Z}, \sigma_\varepsilon^2)\mathrm{p}(\mathbf{D} \mid \sigma_D^2)d\mathbf{D}. \tag{6.30}$$

Par contre, la variable $\mathbf{D}$ n'est pas intégrée directement. On sépare d'abord les observations $\mathbf{Y} = [\mathbf{y}_i, \mathbf{Y}_{-i}]$ et les variables latentes $\mathbf{Z} = [\mathbf{z}_i, \mathbf{Z}_{-i}]$. Sous conditions d'indépendance entre les observations, on écrit l'équation (6.30) comme suit :

$$\begin{aligned}
&\mathrm{p}(\mathbf{y}_i \mid \mathbf{z}_i, \sigma_\varepsilon^2, \sigma_D^2)\mathrm{p}(\mathbf{Y}_{-i} \mid \mathbf{Z}_{-i}, \sigma_\varepsilon^2, \sigma_D^2) \\
&\qquad \propto \int \mathrm{p}([\mathbf{y}_i, \mathbf{Y}_{-i}] \mid \mathbf{D}, [\mathbf{z}_i, \mathbf{Z}_{-i}]\sigma_\varepsilon^2)\mathrm{p}(\mathbf{D} \mid \sigma_D^2)\mathrm{d}\mathbf{D} &(6.31)\\
&\qquad \propto \int \mathrm{p}(\mathbf{y}_i \mid \mathbf{D}, \mathbf{z}_i, \sigma_\varepsilon^2)\mathrm{p}(\mathbf{Y}_{-i} \mid \mathbf{D}, \mathbf{Z}_{-i}, \sigma_\varepsilon^2)\mathrm{p}(\mathbf{D} \mid \sigma_D^2)\mathrm{d}\mathbf{D}. &(6.32)
\end{aligned}$$

On utilise la règle de Bayes pour $\mathrm{p}(\mathbf{Y}_{-i} \mid \mathbf{D}, \mathbf{Z}_{-i}, \sigma_\varepsilon^2)\mathrm{p}(\mathbf{D} \mid \sigma_D^2)$. Comme ces deux distributions sont des gaussiennes, la distribution *a posteriori* $\mathrm{p}(\mathbf{D} \mid \mathbf{Y}_{-i}, \mathbf{Z}_{-i}, \sigma_\varepsilon^2, \sigma_D^2)$

est aussi une gaussienne. Voir la partie 6.4.3 pour les détails de la loi *a posteriori* de $\mathbf{D}$. L'équation (6.32) devient :

$$\begin{aligned} \mathrm{p}(\mathbf{y}_i \mid \mathbf{z}_i, \mathbf{Y}_{-i}, \mathbf{Z}_{-i}, \sigma_\varepsilon^2, \sigma_D^2) &\propto \int \mathrm{p}(\mathbf{y}_i \mid \mathbf{D}, \mathbf{z}_i, \sigma_\varepsilon^2)\mathrm{p}(\mathbf{D} \mid \mathbf{Y}_{-i}, \mathbf{Z}_{-i}, \sigma_\varepsilon^2, \sigma_D^2)\mathrm{d}\mathbf{D} \\ &\propto \int \mathrm{p}(\mathbf{y}_i \mid \mathbf{D}, \mathbf{z}_i, \sigma_\varepsilon^2)\mathcal{N}(\mathbf{D} \mid \boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}, \boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}})\mathrm{d}\mathbf{D}. \end{aligned} \tag{6.33}$$

La vraisemblance $\mathrm{p}(\mathbf{y}_i \mid \mathbf{D}, \mathbf{z}_i, \sigma_\varepsilon^2)$ est aussi une gaussienne.
Soit $\Delta_{-i} = \{\boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}, \boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}\}$, l'intégrale dans l'équation (6.33) devient :

$$\mathrm{p}(\mathbf{y}_i \mid \mathbf{z}_i, \sigma_\varepsilon^2, \Delta_{-i}) \propto \mathcal{N}(\mathbf{y}_i \mid \boldsymbol{\mu}_{\mathbf{y}i|\Delta_{-i}}, \boldsymbol{\Sigma}_{\mathbf{y}i|\Delta_{-i}}) \tag{6.34}$$

avec

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{y}i|\Delta_{-i}} &= \boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}\mathbf{z}_i \\ \boldsymbol{\Sigma}_{\mathbf{y}i|\Delta_{-i}} &= \mathbf{z}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}\mathbf{z}_i + \sigma_\varepsilon^2\mathbb{I}_K. \end{aligned} \tag{6.35}$$

Il reste à travailler sur les quantités $\boldsymbol{\mu}_{\mathbf{y}i|\Delta_{-i}}$ et $\boldsymbol{\Sigma}_{\mathbf{y}i|\Delta_{-i}}$ que l'on présentera dans la partie 6.4.3.

6.4.3 Mise à jour de la distribution *a posteriori* de D

La distribution *a posteriori* de $\mathbf{D}$ est une distribution gaussienne d'espérance $\boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}$ et de covariance $\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}$ (*cf.* Annexe A.1.2) :

$$\begin{aligned} \mathrm{p}(\mathbf{D} \mid \mathbf{Y}, \mathbf{Z}, \sigma_\varepsilon^2, \sigma_D^2) &\propto \mathrm{p}(\mathbf{Y} \mid \mathbf{D}, \mathbf{Z}, \sigma_\varepsilon^2)\mathrm{p}(\mathbf{D} \mid \sigma_D^2) \\ &\propto \mathcal{N}(\boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}, \boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}) \end{aligned} \tag{6.36}$$

avec $\mathbf{M} = (\mathbf{Z}\mathbf{Z}^{\mathrm{T}} + \frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I})^{-1}$ et

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} &= \sigma_\varepsilon^2\mathbf{M} \\ \boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} &= \mathbf{Y}\mathbf{Z}^{\mathrm{T}}\mathbf{M}. \end{aligned} \tag{6.37}$$

La distribution Gaussienne appartient à la famille exponentielle. Cela permet d'utiliser la *statistique suffisante* (ou *information form*) [89, 90] :

$$\begin{aligned} g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} &= \boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}^{-1} = (1/\sigma_\varepsilon^2)\mathbf{M}^{-1} \\ h_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} &= \boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}\, g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} = (1/\sigma_\varepsilon^2)\mathbf{Y}\mathbf{Z}^{\mathrm{T}}. \end{aligned} \tag{6.38}$$

où $\mathbf{M}^{-1} = (\sum_{i=1}^{N} \mathbf{z}_i\mathbf{z}_i^{\mathrm{T}} + \frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I})$.

L'avantage de la statistique suffisante est de rendre plus facile la manipulation sur l'influence d'une observation i dans la distribution *a posteriori* de $\mathbf{D}$. La distribution *a posteriori* $\mathrm{p}(\mathbf{D} \mid \mathbf{Y}_{-i}, \mathbf{Z}_{-i}, \sigma_\varepsilon^2, \sigma_D^2)$ tenant compte de toutes les données *sauf la donnée* i est aussi une distribution gaussienne. Son espérance $\boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}$ et sa covariance $\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}$ sont facilement déterminées grâce à l'utilisation de la statistique suffisante. On enlève l'influence de l'observation i de la manière suivante :

$$\begin{aligned} g_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} &= g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} - \sigma_\varepsilon^{-2}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}} \\ h_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} &= h_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} - \sigma_\varepsilon^{-2}\mathbf{y}_i\mathbf{z}_i^{\mathrm{T}}. \end{aligned} \tag{6.39}$$

On peut aussi la réintégrer comme suit :

$$\begin{aligned} g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} &= g_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} + \sigma_\varepsilon^{-2}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}} \\ h_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} &= h_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} + \sigma_\varepsilon^{-2}\mathbf{y}_i\mathbf{z}_i^{\mathrm{T}} \end{aligned} \tag{6.40}$$

En pratique, on a besoin de $\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} = g^{-1}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}$ plutôt que de $g_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}$ pour calculer la vraisemblance quasi-marginalisée (6.34). Comme une seule observation est traitée à la fois, on peut utiliser le lemme d'inversion matricielle pour ajouter ou supprimer facilement l'influence d'une seule donnée i sur la covariance de la loi *a posteriori* de $\mathbf{D}$. Ceci évite d'effectuer des inversions matricielles coûteuses. Pour enlever l'influence de la donnée i :

$$\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} = \boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} - \frac{\mathbf{z}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}\mathbf{z}_i - \sigma_{\varepsilon}^2}{\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}} \tag{6.41}$$

$$\boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} = (h_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} - \sigma_{\varepsilon}^{-2}\mathbf{y}_i\mathbf{z}_i^{\mathrm{T}})\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} \tag{6.42}$$

Pour ajouter l'influence de la donnée i pour calculer $\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}$:

$$\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} = \boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} - \frac{\mathbf{z}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}\mathbf{z}_i + \sigma_{\varepsilon}^2}{\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}} \tag{6.43}$$

$$\boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} = (h_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} + \sigma_{\varepsilon}^{-2}\mathbf{y}_i\mathbf{z}_i^{\mathrm{T}})\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} \tag{6.44}$$

Les détails des calculs ci-dessus sont dans l'Annexe A.1.3. Les quantités $\boldsymbol{\mu}_{\mathbf{y}i|\Delta_{-i}}$ et $\boldsymbol{\Sigma}_{\mathbf{y}i|\Delta_{-i}}$ sont ensuite utilisées dans le calcul de la vraisemblance quasi-marginalisée pour échantillonner $\mathbf{z}_i$. À la fin de l'échantillonnage de chaque $\mathbf{z}_i$, on réintègre facilement l'influence de $\mathbf{z}_i$ et $\mathbf{y}_i$ dans ces quantités afin de les mettre à jour.

6.4.4 Inférence de Z

Échantillonnage de z_{ki} :

z_{ki} peut être échantillonné à partir d'une distribution Bernoulli :

$$z_{ki} \sim \text{Bernoulli}\left(\frac{p_1}{p_0 + p_1}\right) \tag{6.45}$$

Soit $\Delta_{-i} = \{\boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}, \boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}\}$. Dans l'échantillonnage de Gibbs marginalisé accéléré (ACGS), le probabilité *a posteriori* de z_{ki} est :

$$\mathrm{P}(z_{ki} \mid \mathbf{y}_i, \mathbf{Z}_{(-ki)}, \sigma_{\varepsilon}^2, \Delta_{-i}) \propto \mathrm{P}(z_{ki} \mid \mathbf{Z}_{(-ki)})\mathrm{p}(\mathbf{y}_i \mid \mathbf{z}_i, \sigma_{\varepsilon}^2, \Delta_{-i}) \tag{6.46}$$

Les probabilités *a posteriori* de $z_{ki} = 0$ ou 1 sont proportionnelles à (p_0, p_1) définies par :

$$p_1 = \frac{m_{k,-i}}{N}\mathcal{N}(\mathbf{y}_i \mid \boldsymbol{\mu}_{\mathbf{y}i|\Delta_{-i}}, \boldsymbol{\Sigma}_{\mathbf{y}i|\Delta_{-i}}) \tag{6.47}$$

$$p_0 = \left(1 - \frac{m_{k,-i}}{N}\right)\mathcal{N}(\mathbf{y}_i \mid \boldsymbol{\mu}_{\mathbf{y}i|\Delta_{-i}}, \boldsymbol{\Sigma}_{\mathbf{y}i|\Delta_{-i}}). \tag{6.48}$$

On utilise d'abord la statistique suffisante pour la loi *a posteriori* de $\mathbf{D}$. Puis pour chaque donnée i, on utilise (6.41) et (6.42) pour supprimer l'influence des données i sur la distribution *a posteriori* de $\mathbf{D}$ et donc sur la distribution *a posteriori* de z_{ki}. Une fois que $\mathbf{z}_i$ est échantillonné, on remet l'influence de i dans la loi *a posteriori* . La complexité dominante de ces opérations est $O(NK^2)$. Lors de l'inférence de z_{ki}, le calcul de la vraisemblance est dominé par une complexité de $O(KL)$ L'échantillonnage de Gibbs marginalisé accéléré [88] peut réduire la complexité à $O(NK^2 + NK^2L))$, voir l'algorithme 8 page 68 .

Échantillonnage de k_{inew} :

Soit k_{inew} le nombre de nouveaux atomes. Lorsqu'un client i choisit k_{inew} atomes, on ajoute k_{inew} nouvelles lignes à la matrice $\mathbf{Z}$. Ces nouvelles lignes contiennent des zéros sauf la colonne i ne contenant que des 1. On appelle $\mathbf{Z}_{new}$, la matrice des k_{inew} lignes de variables latentes et $\mathbf{D}_{new}$ les nouveaux atomes correspondant à $\mathbf{Z}_{new}$. La matrice $\mathbf{Z}_{new,-i}$ est la matrice de k_{inew} lignes sauf la colonne i : elle ne contient que des zéros. Selon l'équation (6.37), la loi *a posteriori* de $\mathbf{D}_{new}$ sans l'observation i sera une loi normale de paramètres :

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{D}_{new}|\mathbf{Y}_{-i},\mathbf{Z}_{new,-i}} &= \mathbf{Y}_{-i}\mathbf{Z}_{new,-i}^{\mathrm{T}}(\mathbf{Z}_{new,-i}\mathbf{Z}_{new,-i}^{\mathrm{T}} + \frac{\sigma_{\boldsymbol{\varepsilon}}^2}{\sigma_D^2}\mathbb{I})^{-1} &&= 0^{L\times k_{inew}} \\ \boldsymbol{\Sigma}_{\mathbf{D}_{new}|\mathbf{Y}_{-i},\mathbf{Z}_{new,-i}} &= \sigma_{\boldsymbol{\varepsilon}}^2(\mathbf{Z}_{new,-i}\mathbf{Z}_{new,-i}^{\mathrm{T}} + \frac{\sigma_{\boldsymbol{\varepsilon}}^2}{\sigma_D^2}\mathbb{I})^{-1} &&= \sigma_D^2\mathbb{I}_{k_{inew}} \end{aligned} \tag{6.49}$$

qui est aussi la loi *a priori* de $\mathbf{D}$.

Soit $\mathbf{D}^* = \begin{bmatrix}\mathbf{D} & \mathbf{D}_{new}\end{bmatrix}$, $\mathbf{Z}^* = \begin{bmatrix}\mathbf{Z} & \mathbf{Z}_{new}\end{bmatrix}$ et $\Delta_{-i}^* = \{\boldsymbol{\mu}_{\mathbf{D}^*|\mathbf{Y}_{-i},\mathbf{Z}_{-i}^*}, \boldsymbol{\Sigma}_{\mathbf{D}^*|\mathbf{Y}_{-i},\mathbf{Z}_{-i}^*}\}$. La loi *a posteriori* $p(\mathbf{D}^* \mid \mathbf{Y}_{-i}, \mathbf{Z}_{-i}^*)$ est une gaussienne de paramètres :

$$\boldsymbol{\mu}_{\mathbf{D}^*|\mathbf{Y}_{-i},\mathbf{Z}_{-i}^*} = \begin{bmatrix}\boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} & \boldsymbol{\mu}_{\mathbf{D}_{new}|\mathbf{Y}_{-i},\mathbf{Z}_{new,-i}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} & 0^{L\times k_{inew}}\end{bmatrix} \tag{6.50}$$

$$\boldsymbol{\Sigma}_{\mathbf{D}^*|\mathbf{Y}_{-i},\mathbf{Z}_{-i}^*} = \begin{bmatrix}\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} & 0^{K\times k_{inew}} \\ 0^{k_{inew}\times K} & \boldsymbol{\Sigma}_{\mathbf{D}_{new}|\mathbf{Y}_{-i},\mathbf{Z}_{new,-i}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} & 0^{K\times k_{inew}} \\ 0^{k_{inew}\times K} & \sigma_D^2\mathbb{I}_{k_{inew}}\end{bmatrix} \tag{6.51}$$

La loi *a posteriori* de k_{inew} dans l'ACGS est calculée de la façon suivante :

$$\mathrm{p}(k_{inew} \mid \mathbf{y}_i, \mathbf{z}_i, \sigma_{\boldsymbol{\varepsilon}}^2, \alpha, \Delta_{-i}^*) \propto \mathcal{P}\left(k_{inew} \mid \frac{\alpha}{N}\right) \mathrm{p}(\mathbf{y}_i \mid \mathbf{z}_i, \sigma_{\boldsymbol{\varepsilon}}^2, \Delta_{-i}^*). \tag{6.52}$$

Pour échantillonner k_{inew}, on peut utiliser la méthode Métropolis-Hastings décrite dans la partie 6.3.1. En choisissant la loi *a priori* comme loi de proposition, l'expression de la probabilité de mouvement est le rapport des vraisemblances quasi-marginalisées. (*cf.* eq.(6.34)).

Comme évoqué dans la partie 6.3.1, l'étape de l'échantillonnage k_{inew} proposée dans [57, 58] peut poser problème. Une correction de ce problème est présentée ci-après. Notons que la structure de l'échantillonnage de $\mathbf{Z}$ reste identique, exceptée l'étape concernant l'échantillonnage de k_{inew}, correspondant à l'ajout de nouvelles lignes pour $\mathbf{Z}$.

6.5 Correction de l'échantillonnage de Z

Dans [57, 58], l'échantillonnage de $\mathbf{Z}$ consiste en deux étapes : l'étape d'échantillonnage de z_{ki} pour les atomes actifs et l'étape d'activation de nouveaux atomes. Cependant il faut tenir compte d'une connection entre l'échantillonnage les z_{ki} où l'atome k est seulement utilisé par la donnée i et l'activation de nouveaux atomes. Les auteurs ont signalé dans [58] que la méthode proposée jusqu'à présent pour échantillonner le nombre de nouveaux atomes dépend de l'état courant de la variable à échantillonner. Cet échantillonnage n'est pas conforme aux hypothèses de l'échantillonnage de Gibbs. Une nouvelle version d'échantillonnage de $\mathbf{Z}$ est proposée dans [91]. Nous la détaillons ici.

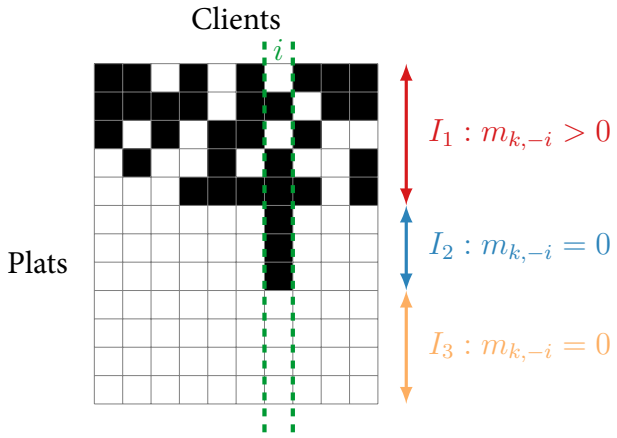


Figure 6.2 – Différents cas de l'inférence de z_{ki}.

6.5.1 Mise en évidence du problème

Une matrice **Z** dont la loi *a priori* est de type IBP(α) est une matrice avec une infinité de lignes. En pratique seulement les colonnes non nulles restent en mémoire. Cependant, les colonnes nulles doivent encore être prises en compte puisque le nombre d'atomes actifs peut changer lors de l'inférence. On s'intéresse maintenant à la façon d'échantillonner $\mathbf{z}_i$ proposée dans [57, 58, 88]. La figure 6.2 illustre les cas de l'inférence de $\mathbf{z}_i$. Pour rappel, $m_{k,-i} = \sum \mathbf{Z}(k, -i)$, on définit :

$$I_1 = \{z_{ki} \mid m_{k,-i} > 0\} \tag{6.53}$$
$$I_2 = \{z_{ki} \mid m_{k,-i} = 0 \;\&\; z_{ki} = 1\} \tag{6.54}$$
$$I_3 = \{z_{ki} \mid m_{k,-i} = 0 \;\&\; z_{ki} = 0\} \tag{6.55}$$

1. **Cas** $m_{k,-i} > 0$ **:**
   Pour échantillonner z_{ki}, on utilise le fait que la probabilité que $z_{ki} = 1$ est $m_{k,-i}/N$ où $m_{k,-i}$ est le nombre de données utilisant l'atome k sauf la donnée i. On n'a pas besoin de savoir l'état précédent de z_{ki} qui peut être 1 ou 0. L'échantillonnage ne dépend pas de l'état précédent de la variable z_{ki} que l'on est en train d'échan- tillonner.
2. **Cas** $m_{k,-i} = 0$ **:**
   Dans le cas où $m_{k,-i} = 0$, il existe deux situations, soit $z_{ki} = 1$, soit $z_{ki} = 0$. Dans les versions proposées dans [57, 58, 88], la façon d'échantillonner z_{ki} dans [57, 58] de ces deux situations est la suivante :
   — pour les k appartenant à I_2, $z_{ki} = 1$: seul le client i a choisi le plat k et non pas les autres. Dans l'état courant, la probabilité que le client i se resservie du plat k est $m_{k,-i}/N$ qui vaut ici 0, ce qui fera disparaître le plat.
   — pour les k appartenant à I_3, $z_{ki} = 0$: le client i va ajouter de nouveaux plats. Le nombre de nouveaux plats est déterminé par une loi de Poisson de paramètre α/N.

Quand on veut échantillonner une variable, il ne faut pas devoir tenir compte de l'état courant de cette variable. Or, dans le cas I_2 et I_3, lors de l'inférence, l'état courant de z_{ki} est pris en compte implicitement. Ensuite, l'échantillonnage dépend de la valeur de z_{ki} (Bernoulli si $z_{ki} = 1$ et Poisson si z_{ki}=0). L'échantillonnage de z_{ki} dans ces deux cas correspond aux noyaux différents de la chaîne de Markov en fonction

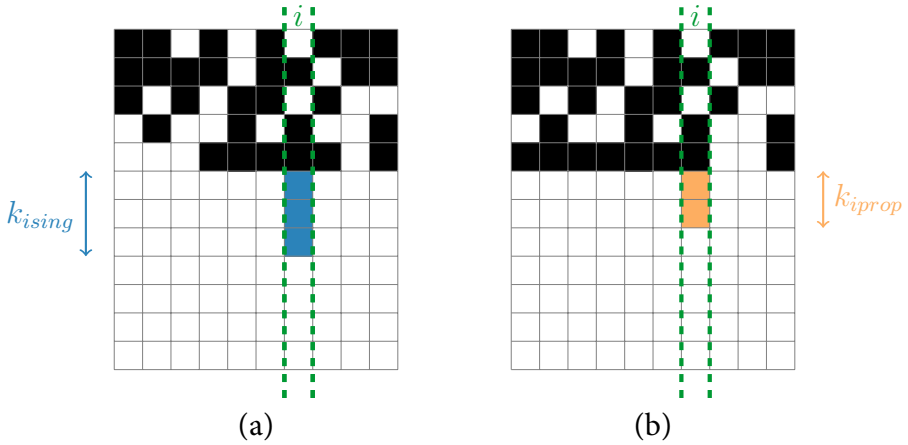


FIGURE 6.3 – Exemple : (a) état courant de **Z** et (b) proposition de **Z**.

de l'état courant de la variable qui est en cours d'échantillonnage. Cela ne permet pas de garantir le bon comportement de la chaîne.

6.5.2 Prise en compte des singletons par Metropolis-Hastings

Dans [91], Knowles et Ghahramani proposent une nouvelle version de l'échantillonnage des z_{ki} dans les cas I_2 et I_3 qui respecte les hypothèses d'un échantillonnage correct en utilisant l'algorithme Métropolis-Hastings. L'algorithme 9 présente les étapes de ce nouvel échantillonnage.

Dans le cas où $m_{k,-i} > 0$, l'échantillonnage de z_{ki} peut être effectué selon une distribution Bernoulli comme dans les versions présentées dans 6.2, 6.3 et 6.4. Les cas où un plat est utilisé par un seul client s'appellent les "singletons". La figure 6.3 illustre les deux cas où l'on a des singletons. Soit k_{ising} le nombre de singletons dans le cas I_2, soit k_{iprop} le nombre de nouveaux plats proposé dans le cas I_3. k_{iprop} est proposé par une loi de proposition $\mathrm{q}(\cdot)$. La loi *a priori* de k_{ising} est toujours $\text{Poisson}(k_{ising}; \frac{\alpha}{N})$.

La proposition est acceptée avec une probabilité $\min(1, a_{k_{ising} \to k_{iprop}})$ où :

$$a_{k_{ising} \to k_{iprop}} = \frac{\mathrm{p}(\mathbf{Y}|k_{iprop}, -)\text{Poisson}\left(k_{iprop}; \frac{\alpha}{N}\right)\mathrm{q}(k_{ising})}{\mathrm{p}(\mathbf{Y}|k_{ising}, -)\text{Poisson}\left(k_{ising}; \frac{\alpha}{N}\right)\mathrm{q}(k_{iprop})} \tag{6.56}$$

pour $i = 1 : N$
- $k_{ising} \leftarrow \{k \mid m_{k,-i} = 0\}$
- **pour** $k \in \{k \mid m_{k,-i} > 0\}$
  - Échantillonner z_{ki} dans le cas non singletons (*cf.* parties 6.2, 6.3 et 6.4)
- Proposer k_{iprop} via eq.(6.57)
- $k_{inew} \leftarrow$ Metropolis-Hastings(k_{ising}, k_{iprop}) eq.(6.58)
- Mettre à jour K

Algorithme 9 : Pseudo-algorithme d'échantillonnage de $\mathbf{Z} \sim \text{IBP}(\alpha)$ en tenant compte des singletons.

Nous pouvons proposer simplement :

$$\mathrm{q}(k_{iprop}) = \text{Poisson}\left(k_{iprop}; \frac{\alpha}{N}\right) \tag{6.57}$$

Dans ce cas :

$$a_{k_{ising} \to k_{iprop}} = \frac{\mathrm{p}(\mathbf{Y}|k_{iprop}, -)}{\mathrm{p}(\mathbf{Y}|k_{ising}, -)}. \tag{6.58}$$

Si $a_{k_{ising} \to k_{iprop}} > \mathcal{U}_{[0,1]}$ alors $k_{inew} = k_{iprop}$, sinon $k_{inew} = k_{ising}$. $\mathrm{p}(\mathbf{Y}|k_{iprop}, -)$ et $\mathrm{p}(\mathbf{Y}|k_{ising}, -)$ sont des vraisemblances. Dans le cas où l'on utilise l'échantillonnage de Gibbs marginalisé, ce seront des vraisemblances marginalisées (voir eq. (6.18)), ou des vraisemblances quasi-marginalisées dans la version accélérée (voir eq. (6.34)).

Dans la suite, l'échantillonnage de $\mathbf{Z}$ sera effectué en respectant les hypothèses de l'échantillonnage. L'étape de l'ajout de nouvelles lignes de $\mathbf{Z}$ correspondant aux nouveaux atomes de $\mathbf{D}$ prend en compte les singletons par Métropolis-Hastings.

6.6 Discussion

Nous avons étudié la littérature pour détailler plusieurs méthodes pour échantillonner le processus de buffet indien. Il faut toutefois faire attention à respecter les hypothèses d'un échantillonnage de Gibbs correct. Cet échantillonnage ne doit pas dépendre de la valeur courante du paramètre que l'on échantillonne. On a vu que l'échantillonneur de Gibbs est pratique ici car il s'appuie sur des lois conjuguées, y compris après marginalisation. On rappelle que la marginalisation permet de "cacher" un paramètre du modèle via une intégrale. Les étapes de marginalisation permettent de réduire l'espace des solutions à explorer. Notons que ces étapes se formalisent naturellement dans un contexte Bayésien en intégrant directement la loi jointe *a posteriori* du modèle. Cette étape consiste à proposer une nouvelle vraisemblance (marginalisée) qui tient compte du comportement moyen au sens de la loi *a priori* du paramètre caché. La neg-log vraisemblance marginalisée obtenue à partir d'un modèle Bayésien peut servir à construire de nouvelles fonctions de coûts pour les méthodes d'optimisation. En revanche, il n'existe pas d'outils systématiques dans la famille des méthodes d'optimisation pour construire de nouvelles fonctions de coût permettant d'intégrer l'influence d'un paramètre.

Ce chapitre présente un travail bibliographique de synthèse sur la question essentielle d'un échantillonnage correct du modèle de buffet indien. Ce travail n'est présenté de façon synthétique que dans nos publications [26, 27]. Les algorithmes décrits dans ce chapitre sont utilisés par la suite en version accélérée en prenant en compte les singletons par Metropolis-Hastings. Cependant, dans le cadre de l'apprentissage de dictionnaire exploré ici, on ne sait pas simuler la loi jointe associée au nouveau nombre d'atomes induite par l'IBP. On s'est donc proposé d'échantillonner cette loi en utilisant une étape de Metropolis-Hastings. Les méthodes de Metropolis-Hastings permettent de simuler suivant n'importe quelle loi à partir d'une loi de proposition arbitrairement choisie. Cependant, l'efficacité des méthodes de Metropolis-Hastings dépend du choix de la loi de proposition. Ici, le dictionnaire a été marginalisé afin de devoir simplement proposer une loi sur le nombre de nouveaux atomes à ajouter. La

loi *a priori* sur le nombre de nouveaux atomes a ensuite été choisie comme loi de proposition. Le design d'une loi de proposition plus adaptée fait partie des perspectives de ce travail, voir 7.2.1.2.

Vue la subtilité de ces algorithmes, les codes Matlab et C seront mis à disposition dans une logique de recherche reproductible. Cette démarche favorisera peut-être la popularisation des méthodes bayésiennes non paramétriques qui peuvent paraître un peu délicates à manipuler au premier abord.

CHAPITRE 7

Processus du buffet indien pour l'apprentissage de dictionnaire

L'apprentissage de dictionnaire pour la représentation parcimonieuse est maintenant bien connu dans le cadre de la résolution de problèmes inverses en traitement d'image [4]. En général, le nombre d'atomes du dictionnaire est fixé à l'avance. Dans cette thèse, nous proposons une méthode qui apprend automatiquement un dictionnaire de taille adaptée grâce à un modèle bayésien non paramétrique de type Buffet Indien. Les autres paramètres comme le niveau de bruit, la parcimonie sont aussi estimés avec précision, de sorte que presqu'aucun réglage des paramètres n'est nécessaire.
Le modèle IBP-DL est élaboré pour les problèmes inverses linéaires tels que le débruitage, l'inpainting et l'acquisition compressée (*compressed sensing, CS*). Différentes méthodes d'échantillonnage sont aussi étudiées. Un estimateur maximum *a posteriori* marginalisé est dérivé. L'essentiel de ce chapitre est publié dans [26, 27] et un article est en cours de révision [28]. Ce chapitre est assez technique. Il contient tous les détails de l'algorithme du modèle proposé. Les calculs et formules sont détaillés dans l'Annexe A.2.

7.1 Présentation du modèle IBP-DL

Le modèle IBP-DL est un modèle bayésien non paramétrique (BNP). Le support de la représentation parcimonieuse est contrôlé par une matrice binaire $\mathbf{Z}$. Le processus Buffet Indien (IBP) [57, 58] présenté dans les chapitres 5 et 6 est proposé comme loi *a priori* non paramétrique sur ce support de représentation. L'acronyme IBP-DL signifie *Indian Buffet Process for Dictionary Learning*. Le modèle peut être décrit $\forall\ 1 \leq i \leq N$ par :

$$\mathbf{y}_i = \mathbf{H}_i\mathbf{x}_i + \boldsymbol{\varepsilon}_i, \tag{7.1}$$
$$\mathbf{x}_i = \mathbf{D}\mathbf{w}_i \text{ où } \mathbf{w}_i = \mathbf{z}_i \odot \mathbf{s}_i, \tag{7.2}$$
$$\mathbf{d}_k \sim \mathcal{N}(0, L^{-1}\mathbb{I}_L), \forall k \in \mathbb{N}, \tag{7.3}$$
$$\mathbf{Z} \sim \text{IBP}(\alpha), \tag{7.4}$$
$$s_{ki} \sim \mathcal{N}(0, \sigma_s^2), \forall k \in \mathbb{N}, \tag{7.5}$$
$$\boldsymbol{\varepsilon}_i \sim \mathcal{N}(0, \sigma_{\boldsymbol{\varepsilon}}^2\mathbb{I}_L). \tag{7.6}$$

où $\odot$ est le produit Hadamard (terme à terme), $\mathcal{N}$ est la distribution Gaussienne, $\mathbb{I}$ représente la matrice identité.

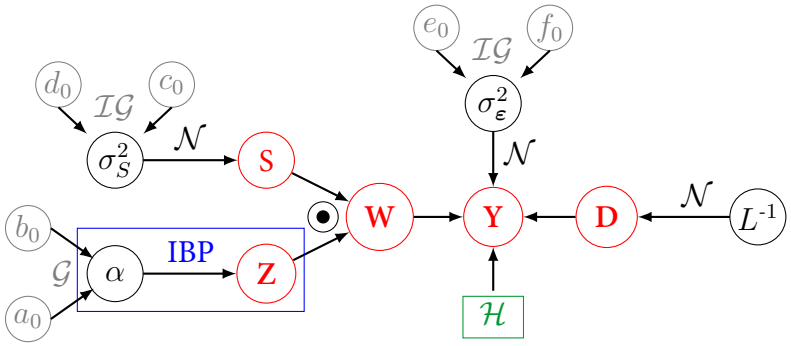


FIGURE 7.1 – IBP-DL pour les problèmes inverses linéaires avec un bruit gaussien.

La figure 7.1 illustre le modèle graphique de IBP-DL. La matrice d'observation $\mathbf{Y}$ contient N vecteurs colonnes $\mathbf{y}_i$ de dimension L, et seulement $Q \leq L$ dans le cas de l'acquisition compressée (*compressed sensing, CS*). Le vecteur $\mathbf{z}_i \in \{0,1\}^K$ encode l'utilisation des K colonnes de $\mathbf{D}$ correspondants aux K atomes pour représenter $\mathbf{y}_i$. La matrice $\mathbf{Z}$ est de taille $K \times N$. Le vecteur $\mathbf{s}_i \in \mathbb{R}^K$ représente les coefficients utilisés pour cette représentation. La matrice $\mathbf{S}$ a la même taille que $\mathbf{Z}$. Les coefficients de la représentation sont définis comme $\mathbf{w}_i = \mathbf{z}_i \odot \mathbf{s}_i$, dans l'esprit d'un modèle Bernoulli-gaussien paramétrique, voir chapitre 2 partie 2.6.7.4. Si $z_{ki} = 1$ alors la donnée i s'associe à l'atome k avec un coefficient $\mathbf{W}(k,i) = \mathbf{S}(k,i)$ et 0 si non. Notons que chaque donnée peut posséder un ensemble de caractéristiques (atomes). La parcimonie de $\mathbf{W}$ est induite par celle de la matrice binaire $\mathbf{Z}$ grâce à l'*a priori* de type IBP(α).

Le modèle IBP-DL offre un nombre potentiellement infini d'atomes $\mathbf{d}_k$ de sorte que la taille du dictionnaire n'est pas limitée *a priori* . Quand on parle de nombre d'atomes, on parle de nombre de colonnes de $\mathbf{D}$ ou de manière équivalente du nombre de lignes de $\mathbf{Z}$, voir partie 6.1. La loi *a priori* de type processus du buffet indien joue le rôle de la régularisation qui pénalise le nombre K de lignes de $\mathbf{Z}$ correspondant au nombre d'atomes actifs. Pour rappel, un atome dit actif lorsqu'au moins une observation l'utilise. L'espérance de K est $\mathbb{E}[K] \simeq \alpha \log(N)$. Lorsque $N \to \infty$ K peut être aller à l'infini mais sous linéaire, ce qui pénalise la taille de $\mathbf{D}$.

La variance σ_D^2 est fixée à $1/L$ en raison de problème l'indétermination du couple ($\mathbf{D}$,$\mathbf{W}$) à un facteur multiplicatif près. Notons que fixer σ_D^2 à $1/L$ revient à écrire que l'énergie contenue dans chaque atome k vaut, en espérance, $\mathbb{E}\left[\mathbf{d}_k^{\mathrm{T}}\mathbf{d}_k\right] = \sum_{i=1}^{L} 1/L = 1$. Il s'agit donc d'une manière douce de les normaliser. Les distributions Gamma sont utilisées comme les lois *a priori* conjuguées pour les paramètres $\boldsymbol{\theta} = (1/\sigma_S^2, 1/\sigma_\epsilon^2, \alpha)$. En pratique, on choisit de faibles valeurs pour les hyperparamètres de précisions (c_0=d_0=e_0=f_0=10^{-6}) afin d'obtenir des hyperpriors non-informatives. α est associé à une loi de Poisson dans l'IBP. On choisit donc un *a priori* de type $\mathcal{G}(1,1)$. Nous soulignons que tous les paramètres de $\boldsymbol{\theta}$ sont estimés lors de l'inférence. La distribution Gamma est définie comme $\mathcal{G}(x;a,b) = x^{a-1}b^a \exp(-bx)/\Gamma(a)$ pour $x > 0$.

$\mathcal{H}$ est l'ensemble des opérateurs linéaires d'observation $\{\mathbf{H}_i\}_{i=1:N}$. L'opérateur linéaire $\mathbf{H}_i$ qui endommage l'observation $\mathbf{y}_i$, peut être une matrice non inversible. Le

problème le plus simple est le débruitage quand $\mathbf{H}_i = \mathbb{I}_L$ [25]. Dans le problème de l'inpainting, $\mathbf{H}_i$ est une matrice binaire diagonale de taille $L \times L$ où les zéros indiquent les pixels manquants. Dans le cas du *compressed sensing*, $\mathbf{H}_i$ est une matrice rectangulaire (et aléatoire) dite de projection, de taille $Q \times L$ ($Q \ll L$). Dans ce qui suit, nous décrivons l'algorithme Markov Chain Monte Carlo (MCMC) pour générer des échantillons en fonction de la distribution *a posteriori* $\mathrm{p}(\mathbf{Z}, \mathbf{D}, \mathbf{S}, \boldsymbol{\theta} \mid \mathbf{Y}, \mathcal{H}, \sigma_D^2)$ issue du modèle IBP-DL.

7.2 Algorithmes MCMC

Cette section décrit les stratégies d'échantillonnage MCMC pour échantillonner la distribution *a posteriori* $\mathrm{p}(\mathbf{Z}, \mathbf{D}, \mathbf{S}, \boldsymbol{\theta} \mid \mathbf{Y}, \mathcal{H}, \sigma_D^2)$. L'algorithme 10 résume les étapes principales de l'inférence du modèle IBP-DL. Les hypothèses de l'échantillonnage lors de l'inférence sont bien respectées, *cf.* partie 6.5. Les différentes méthodes d'échantillonnage de Gibbs sont proposées. L'échantillonnage de Gibbs usuel peut être utilisé pour les trois cas du débruitage, de l'inpainting et de l'acquisition compressée. Nous dérivons l'échantillonneur de Gibbs marginalisé et sa version accélérée pour l'inpainting. Notons que cet échantillonneur marginalisé fonctionne aussi pour le débruitage. Ces types d'échantillonneurs n'ont pas encore proposés dans le cas de l'inpainting, à notre connaissance. Certaines étapes essentielles sont décrites dans les algorithmes 11, 12 et 13. Pour rappel, $\mathbf{W} = \mathbf{Z} \odot \mathbf{S}$. Les variables et paramètres du modèle IBP-DL sont échantillonnés alternativement selon :

$$\mathrm{p}(\mathbf{Z} \mid \mathbf{Y}, \mathcal{H}, \mathbf{D}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}, \alpha) \propto \mathrm{IBP}(\alpha) \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i \mid \mathbf{H}_i \mathbf{D} \mathbf{w}_i, \sigma_{\boldsymbol{\varepsilon}}^2 \mathbb{I}_{\|\mathbf{H}_i\|_0}) \tag{7.7}$$

$$\mathrm{p}(\mathbf{D} \mid \mathbf{Y}, \mathcal{H}, \mathbf{W}\sigma_{\boldsymbol{\varepsilon}}) \propto \prod_{k=1}^{K} \mathcal{N}(\mathbf{d}_k; 0, L^{-1}\mathbb{I}_L) \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i \mid \mathbf{H}_i \mathbf{D} \mathbf{w}_i, \sigma_{\boldsymbol{\varepsilon}}^2 \mathbb{I}_{\|\mathbf{H}_i\|_0}) \tag{7.8}$$

$$\mathrm{p}(\mathbf{S} \mid \mathbf{Y}, \mathcal{H}, \mathbf{D}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_S) \propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{s}_i; 0, \sigma_S^2 \mathbb{I}_K) \mathcal{N}(\mathbf{y}_i \mid \mathbf{H}_i \mathbf{D} \mathbf{w}_i, \sigma_{\boldsymbol{\varepsilon}}^2 \mathbb{I}_{\|\mathbf{H}_i\|_0}) \tag{7.9}$$

$$\mathrm{p}(\sigma_{\boldsymbol{\varepsilon}}^{-2} \mid \mathbf{Y}, \mathcal{H}, \mathbf{D}, \mathbf{W}) \propto \mathcal{G}(\sigma_{\boldsymbol{\varepsilon}}^{-2} \mid c_0, d_0) \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i \mid \mathbf{H}_i \mathbf{D} \mathbf{w}_i, \sigma_{\boldsymbol{\varepsilon}}^2 \mathbb{I}_{\|\mathbf{H}_i\|_0}) \tag{7.10}$$

$$\mathrm{p}(\sigma_S^{-2} \mid \mathbf{S}) \propto \mathcal{G}(\sigma_S^{-2} \mid e_0, f_0) \prod_{i=1}^{N} \mathcal{N}(\mathbf{s}_i \mid 0, \sigma_S^2 \mathbb{I}_K) \tag{7.11}$$

$$\mathrm{p}(\alpha \mid K) \propto \mathcal{G}(\alpha \mid 1, 1) \mathcal{P}(K \mid \alpha \sum_{j=1}^{N} \frac{1}{j}). \tag{7.12}$$

7.2.1 Échantillonnage de la matrice des variables latentes binaires

La matrice $\mathbf{Z}$ a un nombre de lignes potentiellement infinie. En pratique seulement des lignes non nulles (atomes actifs) sont gardées en mémoire. Soit $m_{k,-i}$ le nombre d'observations sauf l'observation i utilisant l'atome k. Un atome est dit actif lorsqu'au moins une observation l'utilise. Les cas où l'atome est utilisé par une seule observa-

Init. : $K = 0$, $\mathbf{Z}=\emptyset$, $\mathbf{D}=\emptyset$, α=1, $\sigma_D^2=L^{-1}$, σ_S^2=1, σ_ε
Résultat : $\mathbf{D} \in \mathbb{R}^{P\times K}$, $\mathbf{Z} \in \{0;1\}^{K\times L}$, $\mathbf{S} \in \mathbb{R}^{K\times L}$, σ_ε
pour *chaque itération t*
  pour *donnée* $i = 1 : N$
    $m_{-i} \in \mathbb{N}^{K\times 1} \leftarrow \sum \mathbf{Z}(:,-i)$;
    $k_{ising} \leftarrow \{k \mid m_{-i}(k) = 0\}$;
    pour $k \in \{k \mid m_{-i}(k) > 0\}$
      Échantillonner z_{ki} du cas non-singletons , voir 7.2.1 ;
    Proposer k_{iprop} ;
    Échantillonner le nombre de nouveaux atomes via MH, voir 7.2.1 ;
    %% inclus des nouveaux coefficients de $\mathbf{s}_i$ et/ou des nouveaux atomes de D
    et les nouveaux coefficients **pour** *atome* $k = 1 : K$
      Échantillonner $\mathbf{d}_k$, voir 7.2.2 ;
      Échantillonner $\mathbf{s}_k$, voir 7.2.3 ;
  Échantillonner σ_ε, σ_S, α, voir 7.2.4 ;

Algorithme 10 : Pseudo-algorithme décrivant l'inférence du modèle IBP-DL.

tion s'appelle *un singleton*, voir partie 6.5.2. Les deux étapes de l'échantillonnage de $\mathbf{Z}$ sont :

— échantillonnage *des non-singletons* $z_{ki} = \mathbf{Z}(k,i)$ dans le cas des atomes actifs et $m_{k,-i} > 0$,
— échantillonnage du nombre de nouveaux atomes correspondant aux *nouveaux singletons*.

Pour échantillonner $\mathbf{Z}$, trois méthodes d'échantillonnage de Gibbs (usuel, marginalisé, accéléré) se basant sur ces deux étapes sont proposées dans la suite .

7.2.1.1 Échantillonnage de Gibbs usuel

Dans cette approche, $\mathbf{Z}$ est échantillonnée à partir de la distribution *a posteriori* :

$$\mathrm{P}(\mathbf{Z} \mid \mathbf{Y}, \mathcal{H}, \mathbf{D}, \mathbf{S}, \sigma_\varepsilon^2, \alpha) \propto \mathrm{p}(\mathbf{Y}|\mathcal{H}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_\varepsilon^2)\mathrm{p}(\mathbf{Z} \mid \alpha) \tag{7.13}$$

Échantillonnage de z_{ki} pour les cas non-singletons :

Comme l'IBP est une distribution échangeable (voir partie 5.2) on peut voir chaque client comme le dernier client, ce qui permet d'écrire la probabilité *a priori* de z_{ki} comme suit :

$$\mathrm{P}(z_{ki} = 1|\mathbf{Z}_{k,-i}) = \frac{m_{k,-i}}{N}. \tag{7.14}$$

La vraisemblance $\mathrm{p}(\mathbf{Y}|\mathcal{H}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_\varepsilon^2)$ est facilement calculée conformément au modèle présenté dans la section 7.1. En utilisant la règle de Bayes, on peut écrire :

$$\mathrm{p}(z_{ki}|\mathbf{Y}, \mathcal{H}, \mathbf{D}, \mathbf{Z}_{-ki}, \mathbf{S}, \sigma_\varepsilon) \propto \mathcal{N}(\mathbf{y}_i|\mathbf{H}_i\mathbf{D}(\mathbf{z}_i \odot \mathbf{s}_i), \sigma_\varepsilon^2)\mathrm{P}(z_{ki}|\mathbf{Z}_{-ki}). \tag{7.15}$$

Les probabilités *a posteriori* de $z_{ki} = 0$ ou 1 sont proportionnelles à (p_0, p_1) définis par :

$$p_0 = 1 - m_{k,-i}/N \tag{7.16}$$

$$p_1 = \frac{m_{k,-i}}{N} \exp\left[-\frac{1}{2\sigma_\varepsilon^2}\left(s_{ki}^2 \mathbf{d}_k^T \mathbf{H}_i^T \mathbf{H}_i \mathbf{d}_k - 2 s_{ki} \mathbf{d}_k^T \mathbf{H}_i^T (\mathbf{y}_i - \mathbf{H}_i \sum_{j \neq k}^{K} \mathbf{d}_j z_{ji} s_{ji})\right)\right], \tag{7.17}$$

voir Annexe A.2.1 pour le calcul détaillé. Par conséquent, z_{ki} peut être simulé selon une distribution Bernoulli :

$$z_{ki} \sim \text{Bernoulli}\left(\frac{p_1}{p_0 + p_1}\right). \tag{7.18}$$

Échantillonnage du nombre de nouveaux atomes :

D'après [91], on utilise la méthode Metropolis-Hastings pour échantillonner le nombre k_{inew} de nouveaux atomes (nouveaux singletons), voir partie 6.5.2. On veut échantillonner le nombre k_{inew} de nouvelles lignes à ajouter dans $\mathbf{Z}$. Dans ces k_{inew} lignes, seuls les nouveaux éléments $\mathbf{z}_{inew}$ de la colonnes i sont tous des 1, les autres colonnes sont des 0. On parle de nouveaux singletons. Dans le modèle IBP-DL, cette étape provoque non seulement l'ajout de nouveaux atomes $\mathbf{D}_{new}$ mais aussi de nouveaux coefficients à $\mathbf{s}_{inew}$ associés à ces atomes et à $\mathbf{z}_{inew}$. On souhaite maintenant échantillonner $\zeta_{new} = \{k_{inew}, \mathbf{D}_{new}, \mathbf{s}_{inew}\}$.

Soit k_{ising} le nombre de singletons dans la matrice $\mathbf{Z}$, $\mathbf{D}_{sing}$ les atomes de k_{ising} et $\mathbf{s}_{ising}$ les coefficients correspondant à k_{ising}. Soit $k_{iprop} \in \mathbb{N}$ le nombre de nouveaux atomes proposé, soit $\mathbf{D}_{prop}$ les k_{iprop} nouveaux atomes proposés et $\mathbf{s}_{iprop}$ les nouveaux coefficients proposés correspondant à $\mathbf{D}_{prop}$. Ainsi, on a la proposition $\zeta_{prop} = \{k_{iprop}, \mathbf{D}_{prop}, \mathbf{s}_{iprop}\}$. Soit $\mathrm{q}(\cdot)$ la distribution de proposition, on propose un mouvement $\zeta_{sing} \to \zeta_{prop}$ avec une probabilité ayant la forme :

$$\mathrm{q}(\zeta_{prop}) = \mathrm{q}_K(k_{iprop}) \mathrm{q}_D(\mathbf{D}_{prop}) \mathrm{q}_S(\mathbf{s}_{iprop}). \tag{7.19}$$

Soit $u \in (0, 1)$ une variable aléatoire Uniforme. Si u vérifie la condition :

$$u \leq \min\left(1, a_{\zeta_{sing} \to \zeta_{prop}}\right) \tag{7.20}$$

alors la proposition est acceptée, et $\zeta_{new} = \zeta_{prop}$. On appelle $a_{\zeta_{sing} \to \zeta_{prop}}$, l'expression de la probabilité de mouvement qui s'écrit sou la forme :

$$a_{\zeta_{sing} \to \zeta_{prop}} = \frac{P(\zeta_{prop} \mid \mathbf{Y}, rest) J(\zeta_{sing})}{P(\zeta_{sing} \mid \mathbf{Y}, rest) J(\zeta_{prop})} = \frac{\mathrm{p}(\mathbf{Y} | \zeta_{prop}, rest)}{\mathrm{p}(\mathbf{Y} | \zeta_{sing}, rest)} a_K a_D a_S \tag{7.21}$$

où

$$a_K = \frac{\text{Poisson}(k_{iprop}; \alpha/N) \mathrm{q}_K(k_{ising})}{\text{Poisson}(k_{ising}; \alpha/N) \mathrm{q}_K(k_{iprop})} \tag{7.22}$$

$$a_D = \frac{\mathcal{N}(\mathbf{D}_{prop}; 0, \sigma_D^2) \mathrm{q}_D(\mathbf{D}_{sing})}{\mathcal{N}(\mathbf{D}_{sing}; 0, \sigma_D^2) \mathrm{q}_D(\mathbf{D}_{prop})} \tag{7.23}$$

$$a_S = \frac{\mathcal{N}(\mathbf{s}_{iprop}; 0, \sigma_S^2) \mathrm{q}_S(\mathbf{s}_{ising})}{\mathcal{N}(\mathbf{s}_{ising}; 0, \sigma_S^2) \mathrm{q}_S(\mathbf{s}_{iprop})}. \tag{7.24}$$

Si on utilise la loi *a priori* comme loi de proposition sur ζ_{prop}, l'expression de la probabilité de mouvement devient simplement le rapport des vraisemblances :

$$a_{\zeta_{sing}\to\zeta_{prop}} = \frac{\mathrm{p}(\mathbf{y}_i|\zeta_{prop}, rest)}{\mathrm{p}(\mathbf{y}_i|\zeta_{sing}, rest)} \tag{7.25}$$

puisque $a_K = a_D = a_S = 1$ dans ce cas.

Cet échantillonneur sera utilisé ensuite pour résoudre le problème de compressed sensing dont les résultats sont présentés dans la partie 8.5 du chapitre 8.

7.2.1.2 Échantillonnage de Gibbs marginalisé pour l'inpainting

Comme évoqué dans 6.3.1, les approches bayésiennes permettent d'effectuer la marginalisation afin de réduire l'espace des solutions à explorer, ce qui est intéressant dans le cas non-paramétrique. Dans l'étape de l'échantillonnage de k_{inew}, on doit proposer à la fois les nouveaux atomes $\mathbf{D}_{new} \in \mathbb{R}^{L\times k_{inew}}$ et les nouveaux coefficients associés $\mathbf{s}_{inew} \in \mathbb{R}^{k_{inew}}$. Il est naturel de souhaiter les marginaliser afin de avoir un taux de mélange plus rapide (grâce au théorème de Rao-Blackwell [42]). Il est facile d'intégrer soit par rapport aux nouveaux atomes $\mathbf{D}_{new}$ du dictionnaire, soit par rapport les nouveaux coefficients $\mathbf{s}_{inew}$ de la matrice de coefficients, mais pas les deux. Un calcul détaillé sur la marginalisation par rapport à $\mathbf{S}$ est effectué dans le cas où les $\mathbf{H}_i = \mathbb{I}$, voir Annexe A.3. Cependant, l'espace à explorer de $\mathbf{D}_{new}$ est plus grand que celui de $\mathbf{s}_{inew}$: nous choisissons de marginaliser par rapport à $\mathbf{D}$ et incluons $\mathbf{s}_{inew}$ dans le cadre de la proposition.

Le calcul détaillé dans le cas de l'inpanting se trouve dans l'Annexe A.2.2. Notons que lorsqu'on intègre une variable quelque part, et que cette variable apparaît dans une autre distribution *a posteriori*, elle doit être échantillonnée avant de la réutiliser [92]. Par conséquent, la variable $\mathbf{D}$ doit être échantillonnée immédiatement après la variable $\mathbf{Z}$ et ensuite on échantillonne $\mathbf{S}$ et les paramètres $\boldsymbol{\theta} = \{\sigma_\varepsilon, \sigma_S, \alpha\}$.

Échantillonnage de z_{ki} :

Dans cette approche, $\mathbf{Z}$ est échantillonné à partir d'une distribution *a posteriori* marginalisé par rapport à $\mathbf{D}$ comme expliqué dans le chapitre 6 :

$$P(\mathbf{Z} \mid \mathbf{Y}, \mathcal{H}, \mathbf{S}, \sigma_\varepsilon^2, \sigma_D^2, \alpha) \propto \mathrm{p}(\mathbf{Y} \mid \mathcal{H}, \mathbf{Z}, \mathbf{S}, \sigma_\varepsilon^2, \sigma_D^2) P(\mathbf{Z} \mid \alpha). \tag{7.26}$$

Dans le cas de l'inpainting, $\mathbf{H}_i$ est une matrice diagonale binaire de taille $L \times L$. À cause de la présence des masques binaires, on ne peut plus marginaliser directement tout le dictionnaire $\mathbf{D}$. La marginalisation sera effectuée par rapport aux lignes de $\mathbf{D}$. Cela correspond à la valeur d'un même pixel dans chacun des atomes. Si chaque $\mathbf{H}_i$ est le masque endommageant chaque observation $\mathbf{y}_i = \mathbf{Y}(:, i)$, $\mathbf{F}_\ell$ est le masque endommageant les pixels à l'emplacement ℓ des données $\mathbf{y}_{\ell,:} = \mathbf{Y}(\ell, :)$. $\mathbf{F}_\ell$ est une matrice diagonale binaire de taille N. $\mathbf{F}_\ell(i, i)$ indique si le pixel à l'emplacement ℓ du patch i est observé ou non, ainsi $\mathbf{F}_\ell(i, i)$=$\mathbf{H}_i(\ell, \ell)$=$H_{i,\ell}$. Soient $\mathcal{F} = \{\mathbf{F}_\ell\}_{\ell=1,\dots,L}$, $\mathbf{W} = \mathbf{Z} \odot \mathbf{S}$. Comme présenté dans la partie 6.3.1 du chapitre 6, la marginalisation

est effectuée via une intégrale :

$$\begin{aligned} &\mathrm{p}(\mathbf{Y} \mid \mathcal{H}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2) = \mathrm{p}(\mathbf{Y} \mid \mathcal{F}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2) \\ &= \int \mathrm{p}(\mathbf{Y} \mid \mathcal{F}, \mathbf{D}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}}) \mathrm{p}(\mathbf{D} \mid \sigma_D) d\mathbf{D} \\ =&\frac{1}{(2\pi)^{\|\mathbf{Y}\|_0/2} \sigma_{\boldsymbol{\varepsilon}}^{\|\mathbf{Y}\|_0 - KL} \sigma_D^{KL}} \prod_{\ell=1}^{L} |\mathbf{M}_\ell|^{1/2} \exp\left[-\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2}(\mathbf{Y}_\ell(\mathbb{I} - \mathbf{F}_\ell^T \mathbf{W}^T \mathbf{M}_\ell \mathbf{W} \mathbf{F}_\ell)\mathbf{Y}_\ell^T\right] \end{aligned} \tag{7.27}$$

où $\mathbf{M}_\ell = (\mathbf{W}\mathbf{F}_\ell\mathbf{F}_\ell^T\mathbf{W}^T + \frac{\sigma_{\boldsymbol{\varepsilon}}^2}{\sigma_D^2}\mathbb{I}_K)^{-1}$. Le calcul est détaillé dans l'Annexe A.2.2.

La distribution Bernoulli dans l'équation (7.18) pour échantillonner z_{ki} dépend maintenant de :

$$p_0 = \left(1 - \frac{m_{k,-i}}{N}\right) \mathrm{p}(\mathbf{Y} \mid \mathcal{F}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2) \tag{7.28}$$

$$p_1 = \frac{m_{k,-i}}{N} \mathrm{p}(\mathbf{Y} \mid \mathcal{F}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2). \tag{7.29}$$

Échantillonnage des nouveaux singletons :

Comme le dictionnaire $\mathbf{D}$ est marginalisé, on n'a pas besoin de proposer de nouveaux atomes. Toutefois, une loi de proposition du nombre des nouveaux atomes et des nouveaux coefficients est nécessaire. Soit $\zeta_{prop} = \{k_{iprop}, \mathbf{s}_{iprop}\}$ la proposition. En utilisant la loi *a priori* comme loi de proposition, la probabilité de mouvement est simplement régie par le rapport des vraisemblances marginalisées, voir eq. (7.27) :

$$a_{\zeta_{sing} \to \zeta_{prop}} = \frac{p(\mathbf{Y}|\zeta_{prop}, rest)}{p(\mathbf{Y}|\zeta_{sing}, rest)}. \tag{7.30}$$

Nouvelle loi de proposition :

Comme discuté dans partie 6.6, le choix de la loi de proposition influence l'efficacité des méthodes de Metropolis-Hatings. En prenant la loi *a priori* comme loi de proposition, le nombre de nouveaux atomes est distribué selon une loi Poisson de paramètre α/N. Lorsque N est grand, la proposition selon la loi *a priori* propose rarement de nouveaux atomes. Par exemple, pour la résolution de problèmes inverses en traitement d'image, N est le nombre des patches extraits à partir d'une image. Pour une image de taille 512×512, N peut être égal à 255025, voir partie 3.2, d'où si $\alpha = 1$, $\alpha/N \simeq 5.10^{-6}$.

Par conséquent, on souhaite modifier la distribution de proposition du nombre k_{iprop} de nouveaux atomes. L'idée est de distinguer le choix des nombres de nouveaux atomes entre 0 et k_{max}. On propose d'utiliser la distribution suivante :

$$\begin{aligned} \mathrm{q}_K(k_{iprop}) =& \pi \mathbb{1}_{(k_{iprop} > k_{max})} \mathcal{P}(k_{iprop}; \frac{\alpha}{N}) \\ &+ (1 - \pi) \mathbb{1}_{(k_{iprop} \in [0:k_{max}])} \mathcal{M}(\mathrm{p}_k(0 : k_{max})) \end{aligned} \tag{7.31}$$

$$\begin{aligned} \text{avec} \quad \mathrm{p}_k(x) &= \mathcal{P}(x; \tfrac{\alpha}{N}) \mathcal{N}(\mathbf{y}_i; \mu_{\mathbf{y}i}, \Sigma_{\mathbf{y}i}) = \mathcal{P}\left(k; \tfrac{\alpha}{N}\right) \prod_{l=1}^{L} \mathcal{N}(\mathbf{y}_i(l); \mu_{\mathbf{y}il}, \Sigma_{\mathbf{y}il}) \\ \pi &= \mathrm{P}(k > k_{max}; \tfrac{\alpha}{N}) = \textstyle\sum_{k=k_{max}+1}^{\infty} \mathcal{P}\left(k; \tfrac{\alpha}{N}\right) \end{aligned} \tag{7.32}$$

et $\mathcal{M}$ est une distribution Multinomiale , $\mathcal{P}$ est une distribution Poisson.

Cependant, cette loi de proposition sur mesure s'est empiriquement avérée non adapté en raison du temps d'exécution trop important. On utilisera pour cette raison la loi a priori comme loi de proposition dans la suite de cette thèse.

7.2.1.3 Échantillonnage de Gibbs marginalisé accéléré pour l'inpainting

Comme évoqué dans la partie 6.3.2, la limite numérique de l'échantillonneur de Gibbs marginalisé est sa durée d'exécution à cause du calcul de l'exponentielle dans la vraisemblance marginalisée (7.27). Une version accélérée est proposée en mettant à jour régulièrement la loi *a posteriori* de $\mathbf{D}$, voir partie 6.4. Cela permet à l'échantillonnage de $\mathbf{z}_i$ de ne pas dépendre de toutes les données $\mathbf{Y}$. Il ne dépend alors plus que de $\mathbf{y}_i$ et la distribution *a posteriori* de $\mathbf{D}$ sans l'influence de $\mathbf{y}_i$ et $\mathbf{w}_i = \mathbf{z}_i \odot \mathbf{s}_i$. La statistique suffisante sera utilisée en pratique afin d'enlever et de réintégrer facilement l'influence de i sur la loi *a posteriori* de $\mathbf{D}$. Les calculs de cet échantillonneur sont détaillés dans l'Annexe A.2.3. L'algorithme 11 présente les étapes principales de la version accélérée.

Init. : K=0, $\mathbf{Z}=\emptyset$, $\mathbf{D}=\emptyset$, α=1, $\sigma_D^2=L^{-1}$, σ_S=1, σ_ε
Résultat : $\mathbf{D} \in \mathbb{R}^{P\times K}$, $\mathbf{Z} \in \{0;1\}^{K\times L}$, $\mathbf{S} \in \mathbb{R}^{K\times L}$, σ_ε
pour *chaque itération t*
- Utiliser la statistique suffisante pour la loi *a posteriori* de $\mathbf{D}$ selon eq.(7.35)
- **pour** *donnée* $i = 1 : N$
  - Enlever l'influence de i sur la loi *a posteriori* de $\mathbf{D}$ selon eq.(7.36)
  - $m_{-i} \in \mathbb{N}^{K\times 1} \leftarrow \sum \mathbf{Z}(:,-i)$
  - $k_{ising} \leftarrow \{k \mid m_{-i}(k) = 0\}$ `%% Chercher les singletons`
  - **pour** *atome* $k \in \{k \mid m_{-i}(k) > 0\}$
    - Échantillonner z_{ki}, voir **Algo. 12**
  - Échantillonner $\zeta_{new} = \{k_{inew}, \mathbf{s}_{inew}\}$ voir **Algo. 13**
  - Mettre à jour K
  - Remettre l'influence de i sur la loi *a posteriori* de $\mathbf{D}$ selon eq.(7.36)
- **pour** *atome* $k = 1 : K$
  - Sample $\mathbf{d}_k$ selon eq. (7.48)
  - Sample $\mathbf{s}_k$ selon eq. (7.51)

Algorithme 11 : Algorithme de l'échantillonnage de Gibbs marginalisé accélérée de z_{ki} dans le cas de l'inpainting, voir aussi **l'Algo. 12** et **13** pour les détails.

Distribution ***a posteriori*** de **D** :

Dans 7.2.1.2, $\mathbf{Z}$ peut être échantillonnée en marginalisant $\mathbf{D}$. Le calcul dans l'Annexe A.2.2 montre que la distribution *a posteriori* de $\mathbf{D}$ a une forme gaussienne. Rappelons que $\mathbf{F}_\ell$ est une matrice diagonale binaire de taille N et représente la présence des pixels à

l'emplacement ℓ. Soit $\mathbf{W} = \mathbf{Z} \odot \mathbf{S}$. La loi *a posteriori* de $\mathbf{D}$ est :

$$\mathrm{p}(\mathbf{D} \mid \mathbf{Y}, \mathcal{F}, \mathbf{W}, \sigma_\varepsilon^2, \sigma_D^2) \propto \prod_{\ell=1}^{L} \mathcal{N}(\mathbf{D}(\ell,:); \boldsymbol{\mu}_{\mathbf{D}(\ell,:)}, \boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)}) \tag{7.33}$$

$$\begin{aligned} \text{où} \quad & \boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)} = \sigma_\varepsilon^2 \mathbf{M}_\ell \\ & \boldsymbol{\mu}_{\mathbf{D}(\ell,:)} = \mathbf{Y}(\ell,:)\mathbf{F}_\ell^T \mathbf{W}^T \mathbf{M}_\ell \\ & \mathbf{M}_\ell = (\mathbf{W}\mathbf{F}_\ell \mathbf{F}_\ell^T \mathbf{W}^T + \frac{\sigma_\varepsilon^2}{\sigma_D^2} \mathbb{I}_K)^{-1} \end{aligned} \tag{7.34}$$

L'idée est de travailler sur $\mathbf{D}$ *ligne par ligne* (dimension) à la place de colonnes (atomes) comme d'habitude. La loi *a posteriori* de $\mathbf{D}$ compte tenu de toutes les données *sauf les données* i est également facilement déterminée grâce à l'utilisation de la *statistique suffisante* :

$$\begin{aligned} g_{\mathbf{D}(\ell,:)} &= \boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)}^{-1} = (1/\sigma_\varepsilon^2)\mathbf{M}_\ell^{-1} \\ h_{\mathbf{D}(\ell,:)} &= \boldsymbol{\mu}_{\mathbf{D}(\ell,:)} g_{\mathbf{D}(\ell,:)} = (1/\sigma_\varepsilon^2)\mathbf{Y}(\ell,:)\mathbf{F}_\ell^T \mathbf{W}^T. \end{aligned} \tag{7.35}$$

Cela rend plus facile le traitement de l'influence d'une observation i à part. En effet, on peut définir :

$$\begin{aligned} g_{\mathbf{D}(\ell,:),\pm i} &= g_{\mathbf{D}(\ell,:)} \pm \sigma_\varepsilon^{-2} H_{i,\ell}\, \mathbf{w}_i \mathbf{w}_i^{\mathrm{T}} \\ h_{\mathbf{D}(\ell,:),\pm i} &= h_{\mathbf{D}(\ell,:)} \pm \sigma_\varepsilon^{-2} H_{i,\ell}\, y_i(\ell)\, \mathbf{w}_i^{\mathrm{T}}. \end{aligned} \tag{7.36}$$

En pratique, nous avons besoin de $\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)}$, voire $\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:),-i} = g_{\mathbf{D}(\ell,:),-i}^{-1}$ plutôt que $g_{\mathbf{D}(\ell,:)}$ ou $g_{\mathbf{D}(\ell,:),-i}$. Grâce au lemme d'inversion matricielle, on peut facilement ajouter ou supprimer l'influence d'une seule donnée i sur $\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)}$, voir l'Annexe A.2.3.2.

Échantillonnage de z_{ki} :

La mise en œuvre de cette étape est décrite dans l'algorithme 12. Dans l'esprit de la partie 6.4 du chapitre 6. Les données sont réparties en deux sous-ensembles $\mathbf{Y} = [\mathbf{y}_i, \mathbf{Y}_{-i}]$, $\mathcal{H} = [\mathbf{H}_i, \mathcal{H}_{-i}]$ et $\mathbf{W} = [\mathbf{w}_i, \mathbf{W}_{-i}]$ sous certains conditions indépendantes. Soit $\Delta_{-i} = \{\boldsymbol{\mu}_{\mathbf{D}(\ell,:),-i}, \boldsymbol{\Sigma}_{\mathbf{D}(\ell,:),-i}\}$. La loi *a posteriori* de z_{ki} dans les cas non-singletons est :

$$\begin{aligned} & \mathrm{P}(z_{ki} \mid \mathbf{y}_i, \mathbf{H}_i, \mathbf{z}_i(-k), \mathbf{s}_i, \sigma_\varepsilon, \Delta_{-i}) \\ & \propto \mathrm{P}(z_{ki} \mid \mathbf{Z}_{-(ki)}) \int \mathrm{p}(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{D}, \mathbf{w}_i) \mathrm{p}(\mathbf{Y}_{-i} \mid \mathcal{H}_{-i}, \mathbf{D}, \mathbf{W}_{-i}) p(\mathbf{D} \mid \sigma_D) d\mathbf{D} \\ & \propto \mathrm{P}(z_{ki} \mid \mathbf{Z}_{-(ki)}) \int \mathrm{p}(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{D}, \mathbf{w}_i) \prod_{\ell=1}^{L} \mathcal{N}(\mathbf{D}(\ell,:); \boldsymbol{\mu}_{\mathbf{D}(\ell,:),-i}, \boldsymbol{\Sigma}_{\mathbf{D}(\ell,:),-i}) d\mathbf{D}. \end{aligned} \tag{7.37}$$

$$\mathrm{P}(z_{ki} \mid \mathbf{y}_i, \mathbf{H}_i, \mathbf{z}_i(-k), \mathbf{s}_i, \sigma_\varepsilon, \Delta_{-i}) \propto \mathrm{P}(z_{ki} \mid \mathbf{Z}_{-(ki)}) \prod_{\ell=1}^{L} \mathcal{N}(\mathbf{y}_i(\ell); \mu_{yi\ell}, \sigma_{yi\ell}) \tag{7.38}$$

$$\text{où} \quad \mu_{yi\ell} = H_{i,\ell}\boldsymbol{\mu}_{\mathbf{D}(\ell,:),-i}\mathbf{w}_i \tag{7.39}$$

$$\sigma_{yi\ell} = H_{i,\ell}\mathbf{w}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:),-i}\mathbf{w}_i + \sigma_{\varepsilon}^2. \tag{7.40}$$

Les calculs détaillés se trouvent dans l'Annexe A.2.3.1.

z_{ki} peut être échantillonné à partir d'une distribution Bernoulli

$$z_{ki} \sim \text{Bernoulli}\left(\frac{p_1}{p_0 + p_1}\right). \tag{7.41}$$

où

$$p_0 = (1 - \frac{m_{k,-i}}{N}) \prod_{\ell=1}^{L} \mathcal{N}(\mathbf{y}_i(\ell); \mu_{yi\ell}, \sigma_{yi\ell}) \tag{7.42}$$

$$p_1 = \frac{m_{k,-i}}{N} \prod_{\ell=1}^{L} \mathcal{N}(\mathbf{y}_i(\ell); \mu_{yi\ell}, \sigma_{yi\ell}). \tag{7.43}$$

Une fois les non singletons z_{ki} échantillonnés, on passe à l'étape d'échantillonnage des singletons de $\mathbf{z}_i$ (le nombre ds nouveaux atomes).

Calcul de la vraisemblance dans le cas z_{ki}=1 %% *cf.* eq. (7.43)
- **si** $\mathbf{w}_i(k) = 0$
  - $\mathbf{w}_i(k) \sim \mathcal{N}(0, \sigma_S^2)$
- $tmp \leftarrow \mathbf{w}_i(k)$
- $\boldsymbol{\mu}_{yi} \leftarrow \mathbf{H}_i\boldsymbol{\mu}_{D,-i}\mathbf{w}_i$ %% eq.(7.39)
- **pour** *dimension* $\ell = 1 : L$ %% position d'un pixel
  - $\Sigma_{yi}(\ell) \leftarrow H_{i,\ell}\mathbf{w}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}-i}\{l\}\mathbf{w}_i + \sigma_{\varepsilon}^2$ %% eq.(7.40)
- $p_1 \leftarrow \frac{m_{-i}(k)}{N} \prod_{\ell=1}^{L} \mathcal{N}(\mathbf{y}_i(\ell); \boldsymbol{\mu}_{yi}(\ell), \Sigma_{yi}(\ell))$ %% eq.(7.38)

Calcul de la vraisemblance dans le cas z_{ki}=0 %% *cf.* eq. (7.42)
- $\mathbf{w}_i(k) \leftarrow 0$
- $\boldsymbol{\mu}_{yi} \leftarrow \mathbf{H}_i\boldsymbol{\mu}_{D,-i}\mathbf{w}_i$ %% eq.(7.39)
- **pour** *dimension* $\ell = 1 : L$ %% position d'un pixel
  - $\Sigma_{yi}(\ell) \leftarrow H_{i,\ell}\mathbf{w}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}-i}\{l\}\mathbf{w}_i + \sigma_{\varepsilon}^2$ %% eq.(7.40)
- $p_0 \leftarrow \left(1 - \frac{m_{-i}(k)}{N}\right) \prod_{\ell=1}^{L} \mathcal{N}(\mathbf{y}_i(\ell); \boldsymbol{\mu}_{yi}(\ell), \Sigma_{yi}(\ell))$ %% eq. (7.38)

$z_{ki} \sim \text{Bernoulli}\left(\frac{p_1}{p_1 + p_0}\right)$

si $z_{ki} = 1$
- $\mathbf{w}_i(k) \leftarrow tmp$;

Algorithme 12 : Algorithme d'échantillonnage en version accélérée de z_{ki} de la méthode IBP-DL pour l'inpainting, voir **l'Algo. 11**

Anciens singletons : $\zeta_{sing} = \{k_{ising}, s_{ising}\}$ %% voir l'Algo.11

$\boldsymbol{\mu}_{yi} \leftarrow \mathbf{H}_i \boldsymbol{\mu}_{D,-i} \mathbf{w}_i$

pour *dimension* $\ell = 1 : L$

$\Sigma_{yi}(\ell) \leftarrow H_{i,\ell} \mathbf{w}_i^{\mathrm{T}} \boldsymbol{\Sigma}_{\mathbf{D}-i}\{l\} \mathbf{w}_i + \sigma_{\varepsilon}^2$

$p_{sing} \leftarrow \prod_{\ell=1}^{L} \mathcal{N}(\mathbf{y}_i(\ell); \boldsymbol{\mu}_{yi}(\ell), \Sigma_{yi}(\ell))$

%% la vraisemblance dans (7.38) avec les singletons ζ_{ising}

Nouveaux singletons proposés : $\zeta_{prop} = \{k_{iprop}, s_{iprop}\}$

$k_{iprop} \sim \text{Poisson}(\alpha/N)$

$\mathbf{w}_{prop} \leftarrow \mathbf{w}_i$

$\mathbf{w}_{prop}(k_{ising}) \leftarrow 0$ %% Enlever les anciens singletons

$\mathbf{s}_{iprop} \in \mathbb{R}^{k_{iprop} \times 1} \sim \mathcal{N}(0, \sigma_S^2)$ %% Propose new singletons

$\boldsymbol{\mu}_{yi} \leftarrow \mathbf{H}_i \boldsymbol{\mu}_{D,-i} \mathbf{w}_{prop}$

pour *dimension* $\ell = 1 : L$

$\Sigma_{yi}(l) \leftarrow H_{i,\ell} \mathbf{w}_{prop}^{\mathrm{T}} \boldsymbol{\Sigma}_{\mathbf{D}-i}\{l\} \mathbf{w}_{prop} + \sigma_{\varepsilon}^2 + H_{i,\ell} \mathbf{s}_{iprop}^T \sigma_D^2 \mathbf{s}_{iprop}$

$p_{prop} \leftarrow \prod_{\ell=1}^{L} \mathcal{N}(\mathbf{y}_i(\ell); \boldsymbol{\mu}_{yi}(\ell), \Sigma_{yi}(\ell))$

%% la vraisemblance dans (7.38) avec les ζ_{iprop} et sans les ζ_{ising}

si $\min\left(\frac{p_{prop}}{p_{sing}}, 1\right) > \mathcal{U}_{[0,1]}$

$\mathbf{w}_i = [\mathbf{w}_{prop}; \mathbf{s}_{iprop}]$

pour *dimension* $\ell = 1 : L$ %% position d'un pixel

$\boldsymbol{\Sigma}_{\mathbf{D}-i}\{l\} \leftarrow \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{D}-i}\{l\} & 0 \\ 0 & \sigma_D^2 \mathbb{I}_{k_{iprop}} \end{bmatrix}$

$h_{D,-i} \leftarrow [h_{D,-i} \;\; zeros(P, k_{iprop})]$

Algorithme 13 : Algorithme de Metropolis-Hastings choisissant la loi *a priori* comme loi de proposition pour inférer le nombre de nouveaux atomes et les nouveaux coefficients en version accélérée de la méthode IBP-DL pour l'inpainting, voir **l'Algo. 11**

Échantillonnage des singletons :

L'algorithme 13 décrit la mise en œuvre de cette étape. Lors de l'échantillonnage du nombre de nouveaux atomes, la loi de proposition peut être soit la loi *a priori* Poisson(α/N) ou la loi (7.31) que nous avons proposée dans 7.2.1.2. Lorsqu'une donnée i propose k_{inew} singletons, les nouveaux atomes $\mathbf{D}_{new}$ apparaissent. Sa loi *a posteriori sans l'influence de* i est une gaussienne d'espérance $\boldsymbol{\mu}_{\mathbf{D}_{new}(\ell,:),-i} = 0^{k_{inew}}$ et de covariance $\boldsymbol{\Sigma}_{\mathbf{D}_{new}(\ell,:),-i} = \sigma_D^2 \mathbb{I}_{k_{inew}}$ selon (7.34).

Soit $\mathbf{D}^* = \begin{bmatrix} \mathbf{D} & \mathbf{D}_{new} \end{bmatrix}$, $\mathbf{W}^* = \begin{bmatrix} \mathbf{W} & \mathbf{W}_{new} \end{bmatrix}$. La loi *a posteriori* de $\mathbf{D}^*$ sans l'influence de i est une gaussienne

$$p(\mathbf{D}^* \mid \mathbf{Y}_{-i}, \mathcal{H}_{-i}, \mathbf{W}^*_{-i}, \sigma_{\varepsilon}^2, \sigma_D^2) \propto \prod_{\ell=1}^{L} \mathcal{N}(\mathbf{D}^*(\ell,:); \boldsymbol{\mu}_{\mathbf{D}^*(\ell,:),-i}, \boldsymbol{\Sigma}_{\mathbf{D}^*(\ell,:),-i}) \quad (7.44)$$

avec

$$\boldsymbol{\mu}_{\mathbf{D}^*(\ell,:),-i} = \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{D}(\ell,:),-i} & \boldsymbol{\mu}_{\mathbf{D}_{new}(\ell,:),-i} \end{bmatrix} \tag{7.45}$$

$$\boldsymbol{\Sigma}_{\mathbf{D}^*(\ell,:),-i} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{D}(\ell,:),-i} & 0^{K \times k_{inew}} \\ 0^{k_{inew} \times K} & \boldsymbol{\Sigma}_{\mathbf{D}_{new}(\ell,:),-i} \end{bmatrix} \tag{7.46}$$

En conséquence, l'algorithme 13 utilise les distributions *a priori* comme loi de proposition pour échantillonner du nombre de nouveaux atomes ainsi que des nouveaux coefficients.

7.2.2 Échantillonnage du dictionnaire

Le dictionnaire $\mathbf{D}$ peut être échantillonné par l'échantillonnage de Gibbs. La loi *a posteriori* de chaque atome $\mathbf{d}_k$ est donnée par :

$$\mathrm{p}(\mathbf{d}_k|\mathbf{Y}, \mathcal{H}, \mathbf{Z}, \mathbf{S}, \mathbf{D}_{-k}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_D) \propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i; \mathbf{H}_i \mathbf{D} \mathbf{w}_i, \sigma_{\boldsymbol{\varepsilon}}^2 \mathbb{I}_{\|\mathbf{H}_i\|_0}) \mathcal{N}(\mathbf{d}_k; 0, L^{-1}\mathbb{I}_L). \tag{7.47}$$

Alors,

$$\mathrm{p}(\mathbf{d}_k|\mathbf{Y}, \mathcal{H}, \mathbf{Z}, \mathbf{S}, \mathbf{D}_{-k}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_D) \propto \mathcal{N}(\boldsymbol{\mu}_{\mathbf{d}_k} \boldsymbol{\Sigma}_{\mathbf{d}_k}) \tag{7.48}$$

$$\begin{cases} \boldsymbol{\Sigma}_{\mathbf{d}_k} = (\sigma_D^{-2}\mathbb{I}_L + \sigma_{\boldsymbol{\varepsilon}}^{-2} \sum_{i=1}^{N} w_{ki}^2 \mathbf{H}_i^T \mathbf{H}_i)^{-1} \\ \boldsymbol{\mu}_{\mathbf{d}_k} = \sigma_{\boldsymbol{\varepsilon}}^{-2} \boldsymbol{\Sigma}_{\mathbf{d}_k} \sum_{i=1}^{N} w_{ki} (\mathbf{H}_i^T \mathbf{y}_i - \mathbf{H}_i^T \mathbf{H}_i \sum_{j \neq k}^{K} \mathbf{d}_j w_{ji}). \end{cases} \tag{7.49}$$

Les calculs sont détaillés dans l'Annexe A.2.4.

7.2.3 Échantillonnage de la matrice des coefficients

La distribution *a posteriori* de chaque élément s_{ki} de $\mathbf{S}$ est donnée dans (7.51).

$$\mathrm{p}(s_{ki}|\mathbf{Y}, \mathcal{H}, \mathbf{D}, \mathbf{Z}, \mathbf{S}_{k,-i}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_S) \propto \mathcal{N}(\mathbf{y}_i; \mathbf{H}_i \mathbf{D}(\mathbf{s}_i \odot \mathbf{z}_i), \sigma_{\boldsymbol{\varepsilon}}^2 \mathbb{I}_{\|\mathbf{H}_i\|_0}) \mathcal{N}(\mathbf{s}_i; 0, \sigma_S^2 \mathbb{I}_K). \tag{7.50}$$

Alors,

$$\mathrm{p}(s_{ki}|\mathbf{Y}, \mathbf{H}_i, \mathbf{D}, \mathbf{Z}, \mathbf{S}_{k,-i}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_S) \propto \mathcal{N}(\mu_{s_{ki}}, \Sigma_{s_{ki}}) \tag{7.51}$$

$$z_{ki} = 1 \Rightarrow \begin{cases} \Sigma_{s_{ki}} = (\sigma_{\boldsymbol{\varepsilon}}^{-2} \mathbf{d}_k^T \mathbf{H}_i^T \mathbf{H}_i \mathbf{d}_k + \sigma_S^{-2})^{-1} \\ \mu_{s_{ki}} = \sigma_{\boldsymbol{\varepsilon}}^{-2} \Sigma_{s_{ki}} \mathbf{d}_k^T (\mathbf{H}_i^T \mathbf{y}_i - \mathbf{H}_i^T \mathbf{H}_i \sum_{j \neq k}^{K} \mathbf{d}_j w_{ji}) \end{cases} \tag{7.52}$$

$$z_{ki} = 0 \Rightarrow \begin{cases} \Sigma_{s_{ki}} = \sigma_S^2 \\ \mu_{s_{ki}} = 0. \end{cases} \tag{7.53}$$

Les calculs sont détaillés dans l'Annexe A.2.5.

7.2.4 Échantillonnage des autres paramètres

Les autres paramètres sont échantillonnés en fonction de leur distribution *a posteriori* qui est facilement obtenue grâce aux propriétés de conjugaison :

$$\begin{aligned} \mathrm{p}(\sigma_{\varepsilon}^{-2} \mid \mathbf{Y}, \mathcal{H}, \mathbf{D}, \mathbf{W}) &\propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i \mid \mathbf{H}_i\mathbf{D}\mathbf{w}_i, \sigma_{\varepsilon}^2 \mathbb{I}_{\|\mathbf{H}_i\|_0}) \mathcal{G}(\sigma_{\varepsilon}^{-2} \mid c_0, d_0) \\ \sigma_{\varepsilon}^{-2} \mid \mathbf{Y}, \mathcal{H}, \mathbf{D}, \mathbf{W} &\sim \mathcal{G}\left(c_0 + \tfrac{1}{2}\sum_{i=1}^{N} \|\mathbf{H}_i\|_0, d_0 + \tfrac{1}{2}\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_2^2\right) \end{aligned} \tag{7.54}$$

$$\begin{aligned} \mathrm{p}(\sigma_S^{-2} \mid \mathbf{S}) &\propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{s}_i \mid 0, \sigma_S^2 \mathbb{I}_K) \Gamma(\sigma_S^{-2} \mid e_0, f_0) \\ \sigma_S^{-2} \mid \mathbf{S} &\sim \mathcal{G}\left(e_0 + \tfrac{KN}{2}, f_0 + \tfrac{1}{2}\sum_{i=1}^{N} \mathbf{s}_i^T \mathbf{s}_i\right). \end{aligned} \tag{7.55}$$

Notons que l'on peut échantillonner le paramètre α du processus du buffet indien :

$$\begin{aligned} \mathrm{p}(\alpha \mid K) &\propto \mathcal{P}(K \mid \alpha \sum_{j=1}^{N} \tfrac{1}{j}) \mathcal{G}(\alpha \mid 1, 1) \\ \alpha \mid K &\sim \mathcal{G}\left(1 + K, 1 + \sum_{j=1}^{N} 1/j\right). \end{aligned} \tag{7.56}$$

En conséquence, l'approche bayésienne non paramétrique proposée est également non paramétrique au sens où il n'y a aucun paramètre à régler. Les hyperparamètres ont tous des valeurs très faibles (c_0=d_0=e_0=f_0=10^{-6}) servent à construire des loi sur les paramètres les moins informatives possible. La sensibilité aux conditions initiales de l'algorithme est assez faible. Tous les détails concernant les calculs de cette partie peuvent trouvée dans l'Annexe A.2.6.

7.3 Estimateur du maximum *a posteriori* marginalisé

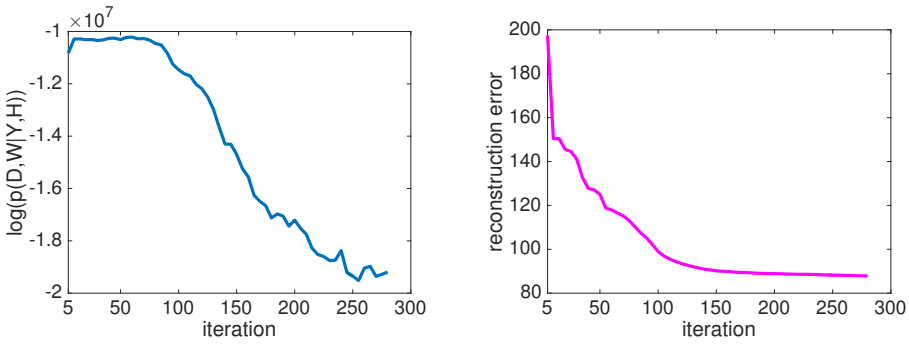
Soit $\boldsymbol{\theta} = (\sigma_{\varepsilon}, \sigma_S, \alpha)$. Une séquence $\{\mathbf{D}^{(t)}, \mathbf{Z}^{(t)}, \mathbf{S}^{(t)}, \boldsymbol{\theta}^{(t)}\}_{t=1}^{T_{MCMC}}$ est échantillonnée par l'algorithme MCMC. Le but final de tout ce travail est de restaurer $\mathbf{X}$ à partir de $(\mathbf{D}, \mathbf{W} = \mathbf{Z} \odot \mathbf{S})$, voir 7.1. Le but de cette section est de définir une estimation pertinente de $(\mathbf{D}, \mathbf{W})$ pour une utilisation pratique dans la résolution de problèmes inverses. On calcule la distribution *a posteriori* $\mathrm{p}(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \mathcal{H})$ qui est le résultat de la marginalisation de la distribution *a posteriori* conjointe sur les paramètres de nuisance $\boldsymbol{\theta}$:

$$\mathrm{p}(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \mathcal{H}) = \int \mathrm{p}(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \mathcal{H}, \boldsymbol{\theta}) \mathrm{p}(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{7.57}$$

où $\mathrm{p}(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \mathcal{H}, \boldsymbol{\theta}) \propto \mathrm{p}(\mathbf{Y} \mid \mathcal{H}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_{\varepsilon}) \mathrm{p}(\mathbf{S} \mid \sigma_S) \mathrm{p}(\mathbf{Z} \mid \alpha) \mathrm{p}(\mathbf{D} \mid \sigma_D)$.
On obtient :

$$\mathrm{p}(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \mathcal{H}) = \int \mathrm{p}(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \mathcal{H}, \boldsymbol{\theta}) \mathrm{p}(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{7.58}$$

$$\propto \frac{1}{\sigma_D^{LK}} \exp\left(-\frac{\|\mathbf{D}\|_{\mathrm{F}}^2}{\sigma_D^2}\right) \left(\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_{\mathrm{F}}^2\right)^{-N_0/2} \frac{\Gamma(NK/2)}{\pi^{NK/2}\|\mathbf{S}\|_{\mathrm{F}}^{NK}} \frac{K!}{(H_N+1)^{K+1}\prod_{h=1}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{(N-m_k)!(m_k-1)!}{N!} \tag{7.59}$$



(a) (b)

Figure 7.2 – (a) L'évolution du logarithme de la loi *a posteriori* marginalisée; (b) L'évolution de l'erreur de reconstruction sur le résultat du segment Barbara avec 50% de pixels manquants et $\sigma_\varepsilon = 15$.

où $N_0 = \sum_{i=1}^{N} \|\mathbf{H}_i\|_0$, $H_N = \sum_{j=1}^{N} 1/j$. Voir l'Annexe A.4 pour les détails des calculs. Ensuite on peut définir l'estimateur maximum *a posteriori* marginalisé (mMAP)

$$(\mathbf{D}_{mMAP}, \mathbf{W}_{mMAP}) = \underset{\{\mathbf{D}^{(t)},\mathbf{Z}^{(t)}\}_{t=1}^{T_{MCMC}}}{\operatorname{argmax}} \log \mathrm{p}(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \mathcal{H}). \tag{7.60}$$

La figure 7.2(a) montre un exemple de l'évolution de cette loi *a posteriori* marginalisée issue d'une expérience de l'inpainting, voir partie 8.4.2, avec l'image Barbara avec 50% de pixels manquants et $\sigma_\varepsilon = 15$. La figure 7.2(b) montre l'évolution de l'erreur de reconstruction quadratique moyenne entre itérations Gibbs. Il est à noter que la distribution *a posteriori* marginalisée est presque toujours en progression alors que l'erreur de reconstruction diminue. Cela conduit en pratique à considérer simplement la dernière itération comme une bonne approximation de l'estimation mMAP. Le résultat à la dernière itération de l'exemple dans la figure 7.2 nous donne un dictionnaire de 58 atomes associé à 343036 coefficients non nuls (9.54%).

7.4 Discussion

Certaines méthodes d'optimisation basées sur K-SVD comme EK-SVD[11], Sub clustering K-SVD[12], Stagewise K-SVD[13] ou encore DLENE[14] proposent des améliorations afin d'obtenir un dictionnaire de taille adaptative. Cependant, l'estimation de la taille du dictionnaire pour ces méthodes est souvent basée sur des heuristiques. La construction du modèle IBP-DL est dans le même esprit que le modèle Bernoulli-gaussien. Chaque coefficient associé à un atome a ainsi une probabilité non nulle de prendre la valeur zéro, et sera Gaussien dans le cas contraire. On propose une méthode où l'*a priori* de type Bernoulli est remplacé par un processus du buffet Indien. IBP-DL est une méthode d'apprentissage de dictionnaire non paramétrique qui permet d'échantillonner le dictionnaire ainsi que sa taille. De plus IBP-DL permet également d'échantillonner le niveau de bruit ainsi que la parcimonie, et aucun paramètre de réglage n'est nécessaire. En ce sens, il s'agit vraiment d'une méthode

non paramétrique. La pertinence de la méthode sera illustrée numériquement sur différents problèmes en traitement d'image au chapitre 8.

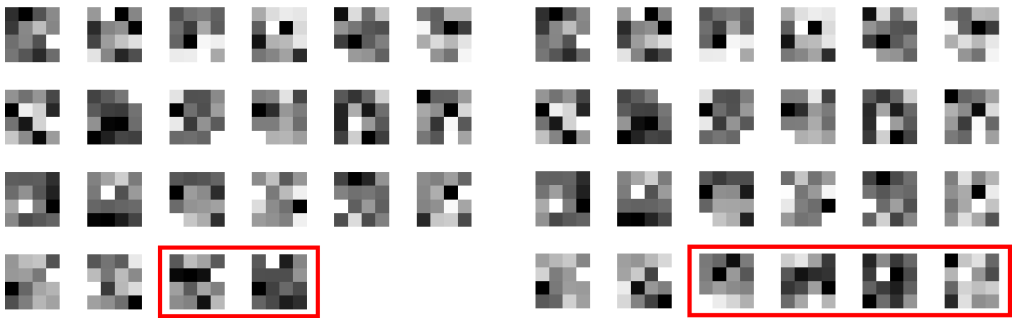
CHAPITRE 8

Applications : problèmes inverses en traitement d'image

Ce chapitre présente les résultats obtenus en appliquant le modèle IBP-DL pour résoudre des problèmes inverses en traitement d'image : débruitage, inpainting, compressive sensing. Les résultats obtenus sont comparés avec ceux d'autres méthodes de l'état de l'art afin d'évaluer la pertinence de l'approche IBP-DL. Une partie des résultats de ce chapitre sont publiés dans [24–27].

8.1 Modèle jouet

Dans une première expérience, nous préparons un ensemble de données synthétiques de $N = 10000$ échantillons en dimension $L = 16$ générées à partir du modèle génératif proposé dans la partie 7.1. Chaque échantillon peut être vu comme une imagette de taille 4×4. Tout d'abord, une matrice binaire $\mathbf{Z}$ est simulée selon un IBP(α) avec $\alpha = 2$, ce qui conduit à un nombre total K de 22 caractéristiques ($\mathbf{Z}$ est de taille $K \times N$). Il est à noter que $K = 22$ est proche de $\mathbb{E}[K] \simeq \alpha \log N \simeq 18$. Ensuite, un dictionnaire de $K = 22$ atomes est construit de manière aléatoire selon une loi normale avec une variance $\sigma_D^2 = 1/L$. Les coefficients $\mathbf{S}$ sont des variables gaussiennes i.i.d. avec $\sigma_S^2 = 1$. L'ensemble de données est corrompu par un bruit blanc gaussien additif avec $\sigma_\varepsilon = 0.1$. Finalement, l'ensemble de données est construit à



(a) (b)

FIGURE 8.1 – Comparaison entre (a) le dictionnaire original de $K = 22$ atomes et (b) le dictionnaire de $K_{ech} = 24$ atomes estimé par IBP-DL. Les rectangles rouge indiquent les atomes pour lesquels aucune corrélation $> 0,55$ n'a été trouvée entre les atomes originaux et estimés : 20 atomes sont inférés correctement à un niveau de corrélation de 0.55.

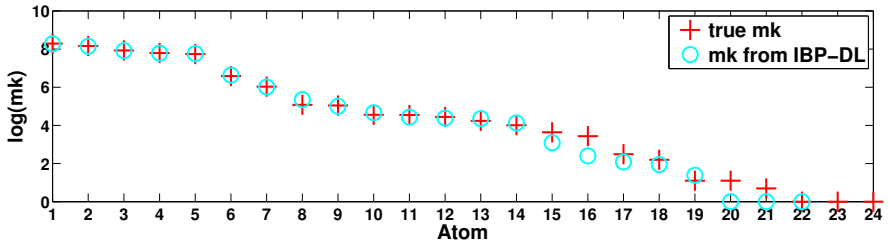
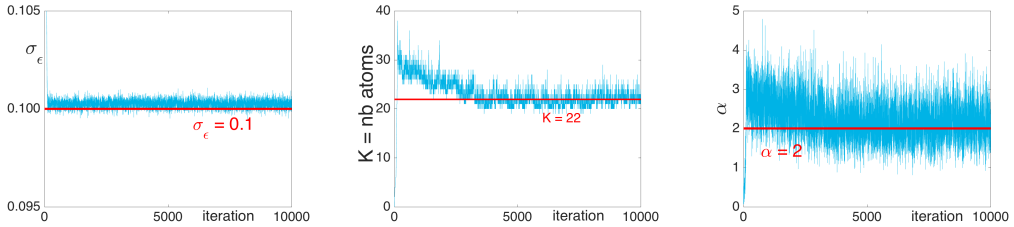


FIGURE 8.2 – Comparaison sur la distribution de m_k, le nombre d'observations utilisant l'atome k : entre les vrais m_k de l'ensemble de données synthétique et les m_k obtenus à partir de l'IBP-DL.



(a) (b) (c)

FIGURE 8.3 – Evolution des paramètres échantillonnés à travers les itérations de IBP-DL : (a) le niveau de bruit σ_ε, (b) le nombre K d'atomes, (c) le paramètre α de la loi *a priori* IBP.

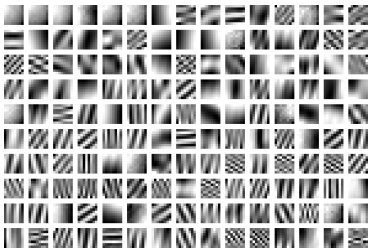
partir de :

$$\mathbf{Y} = \mathbf{D}(\mathbf{Z} \odot \mathbf{S}) + \boldsymbol{\varepsilon} \tag{8.1}$$

La figure 8.1 montre la comparaison entre le dictionnaire original, utilisé pour synthétiser l'ensemble de données, et le dictionnaire échantillonné à la dernière itération de l'IBP-DL, ici après 10000 itérations. Les atomes ont été réorganisés pour rendre la correspondance plus visible. On peut observer que 20 atomes sur 22 atomes sont récupérés à un niveau de corrélation de 0.55 (18 atomes à un niveau 0.9 et 14 atomes à un niveau de 0.99). Seulement deux atomes du dictionnaire d'origine, qui sont bien corrélés aux autres atomes dans le dictionnaire original, ne sont pas identifiés. Ceci montre que l'algorithme se comporte très bien sur ce modèle jouet.

La figure 8.2 montre la distribution des m_k rangés par ordre décroissant, obtenue à partir du jeu de données synthétiques, comparée à la distribution des m_k inférée par IBP-DL. Les atomes sont très similaires, surtout les 14 premiers atomes qui atteignent un niveau de corrélation 0.99. Moins les atomes sont utilisés, plus ils sont difficiles à inférer. Dans le jeu de données synthétiques, les 4 atomes étant utilisés par moins de 10 observations sur 10000 correspondent aux 4 atomes qui ne sont pas identifiés à un niveau de corrélation de 0.55.

La figure 8.3 illustre le comportement des paramètres (σ_ε,K,α) échantillonnés au fil des itérations. On peut voir que le niveau de bruit σ_ε, le nombre d'atomes K et le paramètre α de la loi *a priori* IBP fluctuent rapidement autour de leurs valeurs


(a)Dictionnaire : 150 atomes


(b) Initiale


(c) Restaurée , PSNR=44.94dB

Figure 8.4 – Illustration de la restauration sans bruit obtenue en utilisant IBP-DL sur un segment de l'image Barbara. (a) La taille de chaque atome est 8×8. (b) et (c) La taille du segment est 256×256.

attendues après environ 3000 itérations correspondant à l'étape de chauffe (burn in) de l'échantillonneur.

8.2 Exemple de reconstruction sans bruit

On évalue d'abord la performance de l'IBP-DL à partir d'un segment de taille 256×256 de l'image Barbara sans bruit. L'ensemble de 62001 patches qui se chevauchent est utilisé pour reconstruire ce segment.

La méthode IBP-DL infère un dictionnaire adapté de 150 atomes. La reconstruction est très précise avec PSNR $= 44.94$dB. La figure 8.4 illustre le résultat de la restauration d'image sans bruit en utilisant IBP-DL. À titre de comparaison, la méthode K-SVD [7] apprend un dictionnaire de taille fixée à 256 et produit une erreur de reconstruction plus grande que celle de IBP-DL, PSNR$= 43.97$dB. La méthode bayésienne BPFA proposée dans [51] avec un dictionnaire de taille 256 au départ obtient un PSNR$= 42.92$dB. IBP-DL restaure l'image avec un nombre encore plus petit d'atomes et une meilleure qualité d'approximation.

8.3 Débruitage d'image

8.3.1 IBP-DL et l'état de l'art

On aborde maintenant le problème de débruitage, le problème inverse le plus commun en traitement d'image. Dans cette expérience, 9 images (512×512 ou 256×256) codées sur 8 bits sont traitées pour 2 niveaux de buit $\sigma_\epsilon = 25$ et 40, corespondant respectivement à PSNR $= 20.17$dB et 16.08dB. Chaque image a $(512-7)^2 = 255025$ (ou $(256-7)^2 = 62001$) imagettes chevauchants (*overlapping patches*). Toutefois, à cause du problème de temps d'exécution, IBP-DL apprend juste avec 16129 (ou 3969) patches soit 50% de recouvrement (50% overlapping). La valeur initiale de $\widehat{\sigma}_\epsilon$ est réglée sur une valeur de deux fois plus grande que la vraie, voir l'algorithme 11. Nous comparerons les performances d'IBP-DL à celles de NL-means [93], K-SVD [7], BPFA [51], BM3D [94] et DLENE [14].

	PSNR $\simeq 20.17$dB, $\sigma_\epsilon = 25$			PSNR $\simeq 16.08$dB, $\sigma_\epsilon = 40$		
	PSNR [dB]	# atomes	$\hat{\sigma}_\epsilon$	PSNR [dB]	# atomes	$\hat{\sigma}_\epsilon$
Barbara	29.06	100	25.86	26.34	58	40.76
Boat	28.92	91	25.82	26.75	44	40.64
Carmeraman	28.57	413	26.10	26.24	121	41.16
Fingerprint	26.72	34	25.79	23.99	20	40.90
GoldHill	28.80	54	25.89	26.93	19	40.70
House	31.55	60	25.53	29.11	28	40.46
Lena	31.12	62	25.45	28.78	31	40.24
Mandrill	24.59	169	27.57	22.29	80	42.50
Peppers	29.46	116	25.76	27.06	55	40.58

Table 8.1 – Résultats de l'IBP-DL pour le débruitage appliqué sur 9 images. La moyenne de niveau de bruit estimé est 25.97 en utilisant σ_{init} = 51, et 40.87 en utilisant $\sigma_{init} = 76.5$, quand les vrais niveaux étaient 25 et 40, respectivement.

Le tableau 8.1 rassemble les performances numériques de IBP-DL sur le débruitage, ainsi que la taille du dictionnaire et le niveau de bruit estimé. En utilisant une approche vraiment non-paramétrique comme IBP-DL, il apparaît que la taille du dictionnaire peut considérablement varier d'une image à l'autre, par exemple de plusieurs dizaines à des centaines pour le même niveau de bruit, voir le tableau 8.1. Le niveau de bruit σ_ϵ est aussi estimé avec une bonne précision. La figure 8.5 affiche des résultats typiques de débruitage obtenus en utilisant IBP-DL sur plusieurs exemples du tableau 8.1. Les images débruitées sont de bonne qualité.

La pertinence de l'IBP-DL est illustrée en comparant les résultats de débruitage avec ceux d'autres méthodes de l'état de l'art. Dans un premier temps, IBP-DL produit de meilleurs résultats que NL-means [93], une méthode de référence de débruitage . Pour un niveau de bruit $\sigma_\epsilon = 25$, le PSNR moyen de IBP-DL sur les 9 images est 28.75dB et celui de NL-means est 27.91dB; pour un niveau de bruit de $\sigma_\epsilon = 40$, IBP-DL donne un PSNR moyen de 26.39dB et NL-means donne 25.89dB.

Les résultats de IBP-DL sont ensuite comparés avec plusieurs méthodes basées sur K-SVD [7, 56], et BPFA [51], une approche bayésienne qui peut être considérée comme une approximation paramétrique de l'IBP. Il existe potentiellement de meilleurs méthodes de l'état de l'art pour le débruitage, par exemple BM3D [94]. Une méthode standard pour comparer la pertinence des méthodes d'apprentissage de dictionnaire est de comparer leurs performances de débruitage. Toutefois, BM3D n'est pas une méthode d'apprentissage de dictionnaire. Les résultats de BM3D [94] sont utilisés comme référence que nous ne nous attendons pas à battre. Les expériences actuelles visent à vérifier la pertinence des dictionnaires obtenus à partir de la méthode IBP-DL. Dans la suite, on compare IBP-DL avec DLENE [14], une approche adaptative pour apprendre les dictionnaires avec un nombre d'atomes variable.

(a) (b) (c)

Figure 8.5 – Résultats de débruitage obtenus en utilisant IBP-DL. De haut en bas on trouve le dictionnaire IBP-DL, l'image bruitée, l'image débruitée et l'image originale ; (a) Lena, PSNR de 20.17 dB à 31.12 dB, (b) Boat, PSNR de 20.17 dB à 28.92 dB, (c) Barbara, PSNR de 16.08 dB à 26.34 dB.

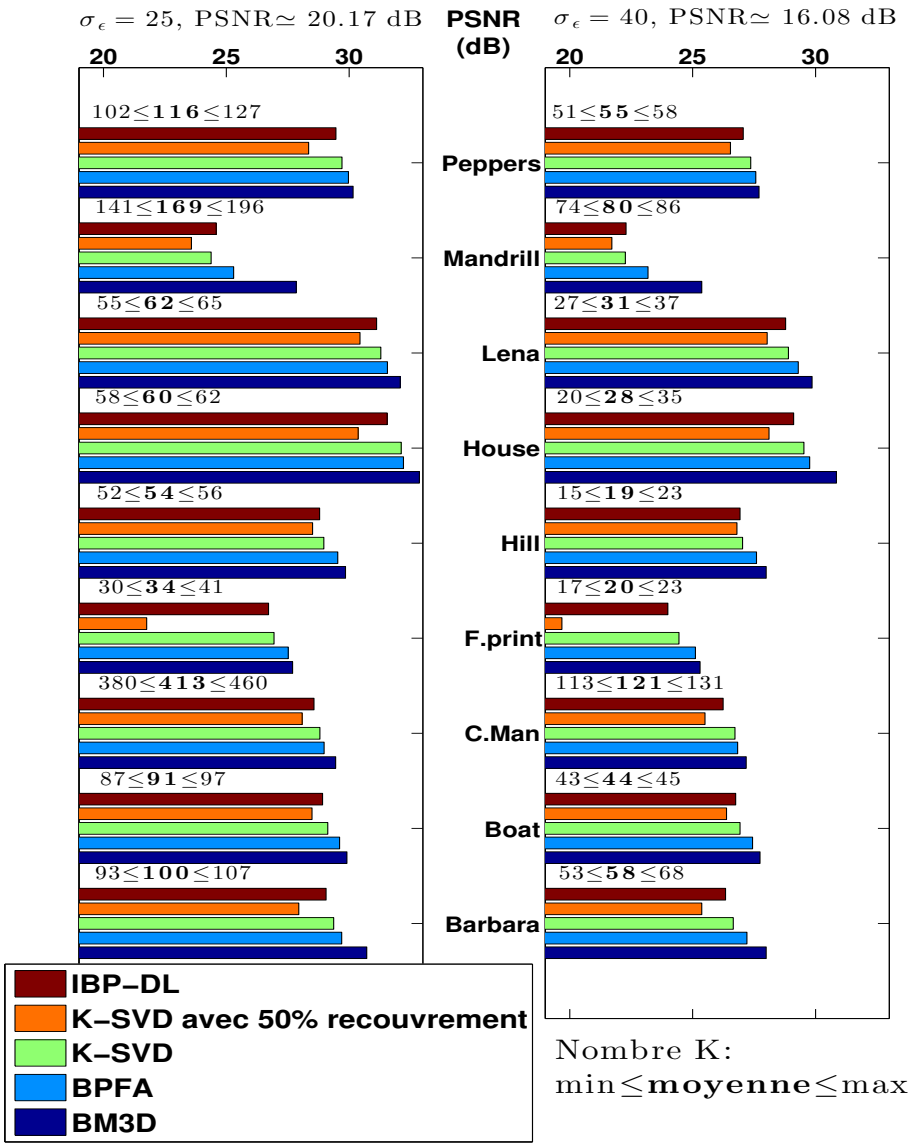


FIGURE 8.6 – Résultats de débruitage de IBP-DL : PSNR et taille K du dictionnaire pour un niveau de bruit (à gauche) $\sigma_\epsilon = 25$, (à droite) $\sigma_\epsilon = 40$. Le texte au-dessus chaque groupe de barres indique la taille dictionnaire inférée par IBP-DL. De haut en bas de chaque groupe de barres, on trouve le PSNR en utilisant IBP-DL appris à partir d'un ensemble de patches réduit, K-SVD avec 256 atomes appris à partir de même ensemble de patches réduit, K-SVD avec 256 atomes appris à partir d'un ensemble de patches complet, BPFA avec un nombre d'atomes initial $K = 256$ ou 512 appris à partir d'un ensemble de patches complet et BM3D.

La figure 8.6 résume la comparaison entre IBP-DL avec les méthodes suivantes :

1. K-SVD avec K=256 appris à partir de même ensemble de patches réduit que IBP-DL (50% overlapping).
2. K-SVD avec K=256 appris à partir de l'ensemble complet de patches,
3. BPFA avec un nombre d'atomes initial $K = 256$ ou 512 dépendant de la taille de l'image,
4. BM3D comme référence de l'état de l'art.

Pour être cohérent avec la méthode de débruitage dans [7, 56] utilisant *Orthogonal Matching Pursuit* et le dictionnaire K-SVD, on utilise les même initialisations [1]. La méthode OMP pour débruiter les images est initialisée avec une tolérance d'erreur maximale de représentaion de $1.15\sigma_\varepsilon$ et un multiplicateur Lagrangien $\lambda = 30/\sigma_\varepsilon$. Les performances de IBP-DL sont au moins comparables aux méthodes basées sur K-SVD. On remarque au passage la sensibilité de K-SVD à l'ensemble d'apprentissage. La figure 8.6 montre que les résultats de K-SVD entraîné sur 16129 (ou 3969) patches comme IBP-DL sont nettement moins bons que ceux de K-SVD entraîné sur 255025 (ou 62001) patches. Notons que IBP-DL propose souvent un dictionnaire de taille K<64 qui n'est pas toujours un dictionnaire redondant bien que la performance de débruitage reste comparable. On compare aussi nos résultats avec DLENE [14], une méthode récente qui adapte également la taille du dictionnaire en visant un compromis entre l'erreur de reconstruction et la parcimonie. Pour Barbara avec σ_ε=25, DLENE donne $\text{PSNR}_{\text{DLENE}}$= 28.82 dB et $\text{PSNR}_{\text{IBP-DL}}$=29.06 dB ; pour Peppers on obtient $\text{PSNR}_{\text{DLENE}}$=27.27dB et $\text{PSNR}_{\text{IBP-DL}}$=27.07dB avec σ_ε=40. En général, l'IBP-DL est aussi performant que DLENE.

8.3.2 IBP-DL et BPFA

On compare plus précisément les résultats de IBP-DL à ceux de BPFA [51], une méthode bayésienne implémentée en utilisant l'échantillonnage de Gibbs. Malgré d'une connexion avec le processus du buffet indien, cette approche n'est pas vraiment une approche non paramétrique. Malgré le titre de l'article qui annonce une approche non paramétrique, il s'agit en fait d'une approximation paramétrique de l'IBP, car elle fonctionne avec un nombre d'atomes fixé à l'avance. La taille initiale (K = 256 ou 512) du dictionnaire de BPFA dépend de la taille de l'image. Ensuite, un sous-ensemble d'atomes est utilisé qui est légèrement inférieur à la taille initiale (environ 250 ou 500). Dans cette comparaison, les deux approches IBP-DL et BPFA sont entraînées sur les même ensembles de données complets : 62001 overlapping patches pour les images *House* et *Peppers* ; 255025 overlapping patches pour les images *Barbara* et *Lena*. L'approche BPFA [2] est initialisée à $K = 256$ pour les images *House* et *Peppers* ; et à $K = 512$ pour les images *Barbara* et *Lena*. Le tableau 8.2 illustre les résultats de IBP-DL et BPFA avec 2 niveaux de bruit σ_ε=25 et 40. Pour l'image *House* avec σ_ε=25, $\text{PSNR}_{\text{IBP-DL}}$=31.95dB avec 57 atomes et BPFA donne $\text{PSNR}_{\text{BPFA}}$= 32.14dB. Pour l'image *Barbara* avec σ_ε=40, $\text{PSNR}_{\text{BPFA}}$=26.34 dB

1. Le code Matlab de R. Rubinstein est disponible à http://www.cs.technion.ac.il/~ronrubin/software.html
2. Le code Matlab de M. Zhou est disponible à http://mingyuanzhou.github.io/Code.html

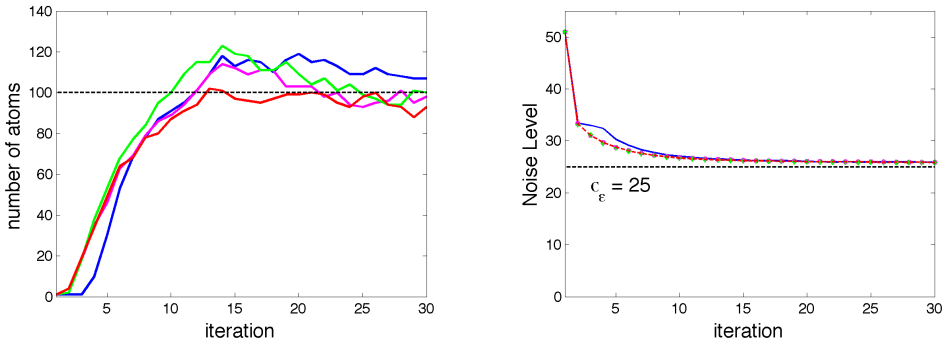
	PSNR $\simeq$ 20.17dB, $\sigma_\epsilon = 25$					PSNR $\simeq$ 16.08dB, $\sigma_\epsilon = 40$				
	IBP-DL			BPFA		IBP-DL			BPFA	
	# atomes	PSNR	$\hat{\sigma}_\epsilon$	PSNR	$\hat{\sigma}_\epsilon$	# atomes	PSNR	$\hat{\sigma}_\epsilon$	PSNR	$\hat{\sigma}_\epsilon$
House	57	31.95	25.37	32.14	25.43	40	29.47	40.43	29.73	40.54
Peppers	191	29.40	25.48	29.88	25.50	163	27.15	40.34	27.06	40.67
Barbara	134	29.31	25.66	29.79	25.45	107	26.81	40.33	26.34	40.15
Lena	173	31.43	25.19	31.58	25.32	111	29.15	40.04	29.27	40.19

Table 8.2 – Résultats de IBP-DL et BPFA quand les vrais niveaux de bruit sont 25 et 40. Pour chauqe niveau de bruit, de gauche à droite, on trouve la taille K du dictionnaire IBP-DL, son PSNR de débruitage (dB) et son niveau de bruit estimé ; puis le PSNR (dB) et le niveau de bruit estimé de BPFA.

alors qu'on obtient $\text{PSNR}_{\text{IBP-DL}}$= 26.81 dB avec 107 atomes. Les performances IBP-DL sont comparables à BPFA [51] alors que les tailles des dictionnaires inférées par IBP-DL sont souvent relativement plus petites que la méthode BPFA. Ces observations permettent de confirmer expérimentalement l'intérêt des approches non paramétriques qui s'adaptent mieux au contenu de l'image. On note surtout une réelle différence de comportement entre une approche vraiment non paramétrique comme IBP-DL et une approximation paramétrique comme BPFA.

8.3.3 Comportement de l'algorithme

Un résultat essentiel de cette approche est l'estimation avec une bonne précision du niveau de bruit. L'erreur d'estimation varie entre 2%- 10% pour σ_ϵ=25 et entre 1%- 6% pour σ_ϵ=40. Cette estimation est un résultat essentiel et rend notre approche



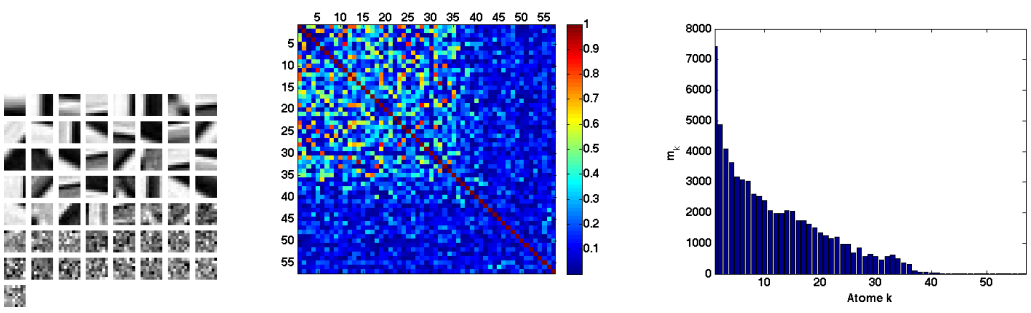
(a) (b)

Figure 8.7 – (a) Evolution du nombre d'atomes de dictionnaire au cours des itérations de IBP-DL sur *Barbara* pour 4 différentes réalisations avec $\sigma_\epsilon = 25$. (b) Evolution du niveau de bruit échantillonné par IBP-DL au cours des itérations sur *Barbara* quand $\sigma_\epsilon = 25$ and $\sigma_{init} = 51$.

pratiquement non paramétrique au sens où il n'y a pas de paramètres à régler. La figure 8.7(a) montre l'évolution de la taille K du dictionnaire dans les itérations. La taille finale (environ 100 dans cet exemple) est atteinte après environ 15 itérations seulement. Cela signifie implicitement que α converge à environ $K/\log(N) \simeq 10$. La figure 8.7(b) montre l'évolution de σ_ε échantillonné au cours des itérations. Après 15 itérations, la valeur échantillonnée a convergé très près de la vraie valeur. Une limite de l'approche IBP-DL est son coût de calcul dû à l'échantillonnage de Gibbs malgré l'implémentation d'une version accélérée (voir partie 7.2.1.3). Par exemple, sur un ensemble de données réduit de l'image *Barbara*, l'apprentissage coûte environ 1 heure pour 30 itérations en utilisant Matlab_R2013b sur un ordinateur portable personnel. C'est pourquoi un ensemble réduit de patches a été utilisé. Un autre type d'inférence est envisagé pour réduire le temps de calcul. Les futurs travaux comprendra une étude des méthodes d'inférence variationnelle [95] qui sont plus prometteuses pour améliorer le temps de calcul (voir les perspectives dans le chapitre 9).

8.3.4 Dictionnaire obtenu

Une autre observation importante est que le dictionnaire inféré par IBP-DL n'est pas toujours un dictionnaire redondant dans le sens où la taille de dictionnaire est plus grande que la dimension des données. L'IBP-DL propose souvent un dictionnaire de taille K plus petite ou seulement un peu plus grande que 64 tandis que K-SVD et BPFA fixent souvent la taille du dictionnaire à 256 ou 512. Néanmoins, le nombre de coefficients non nuls reste petit lors de l'utilisation de l'IBP-DL. Par exemple, pour l'image *House* avec un niveau de bruit $\sigma = 25$, nous avons constaté que BPFA a conduit à 116380 coefficients non nuls (0.73% de parcimonie) en utilisant un dictionnaire de $K = 256$ atomes tandis que IBP-DL donne un dictionnaire de taille $K = 57$ associé à 67028 coefficients non nuls (1.9% parcimonie). La performance de débruitage de IBP-DL reste comparable à celle de K-SVD et BPFA. La figure 8.8(a) illustre les 57 atomes du dictionnaire de l'image *House* avec $\sigma = 25$ dans l'ordre décroissant de leur utilisation (au sens de l'énergie apportée par les coefficients $\mathbf{W}$). La figure 8.8(b) illustre la corrélation entre ces atomes. En particulier, on observe de fortes corrélations entre les 40 premiers atomes. La figure 8.8(c) montre que les 17 derniers atomes qui sont décorrélés avec les autres atomes ressemblent plus au bruit. Ils sont très peu utilisés.



(a) (b) (c)

FIGURE 8.8 – (a) 57 atomes du dictionnaire IBP-DL appris à patir de l'image *House* avec $\sigma_\varepsilon = 25$. (b) Sa matrice de corrélation. (c) Le nombre d'observations utilisant chaque atome

Missing	Cameraman	House	Peppers	Lena
80%	75 - **24.02**	46 - **31.00**	86 - **26.05**	24 - **30.98**
	22.87 - 24.11	28.38 - 30.12	23.51 - 25.92	28.57 - 31.00
50%	87 - **29.02**	52 - **37.86**	93 - **32.66**	84 - **36.66**
	26.56 - 28.90	33.40 - 38.02	28.36 - 32.58	33.25 - 36.94
20%	75 - **35.14**	56 - **42.37**	90 - **37.58**	44 - **39.20**
	27.56 - 34.70	34.66 - 43.03	30.09 - 37.73	34.37 - 41.27
Missing	Baboon	Boat	F.print	Hill
80%	63 - **21.93**	29 - **27.86**	44 - **26.52**	98 - **29.33**
	21.24 -21.47	25.95 - 27.81	21.01 - 26.03	27.88 - 29.33
50%	48 - **25.70**	84 - **33.39**	45 - **33.74**	71 - **33.82**
	24.16 - 25.98	30.34 - 33.78	27.56 - 33.53	31.61 - 34.23
20%	65 - **29.48**	62 - **37.54**	86 - **39.88**	68 - **37.34**
	25.36 - 31.67	31.48 - 39.50	29.04 - 40.17	32.67 - 38.75

Table 8.3 – Inpainting appliqué sur des images en niveaux de gris : (en haut) la taille K du dictionnaire appris par IBP-DL, le **PSRN** (dB) de restauration de IBP-DL ; (en bas) (à gauche) le PSNR (dB) issue de K-SVD et (à droite) de BPFA .

8.4 Inpainting d'image

8.4.1 Inpainting sans bruit

Dans cette expérience, 6 images en niveaux de gris sont traitées pour trois niveaux de pixels manquants : 80%,50% et 20%. IBP-DL apprend les dictionnaires à partir de ces images observées. Chaque image a 255025 (ou 62001) imagettes se chevauchant (*overlapping patches*).

La figure 8.9 illustre les résultats de l'inpainting avec IBP-DL pour trois images, et trois configurations différentes. En particulier, les pourcentages de pixels manquants 80%, 50% et 20% donnent respectivement des performances de 26.05 dB, 33.82 dB et 35.14 dB. À ces performances quantitives s'ajoute un bonne aspect visuel des images.

Le tableau 8.3 décrit les résultats de IBP-DL comparés à ceux de BPFA [51] sur un échantillon d'images. Les deux méthodes ont des performances similaires, et les écarts sont de l'ordre de 0.1dB. Cependant, des écarts de performances plus importants sont observés pour 20% de pixels manquants, le plus souvent en faveur de BPFA. Sur l'image *House* pour 80% de pixels manquants, IBP-DL et BPFA atteignent respectivement des performances de 31.0 dB et 30.12 dB, et donc en faveur de IBP-DL. Les tendances des résultats de l'inpainting sont similaires malgré un nombre d'atomes inféré par IBP-DL ($39 \leq K \leq 150$) en général plus petit que le nombre d'atomes utilise par BPFA (256 ou 512).

Le tableau 8.3 compare aussi les résultats de IBP-DL à ceux obtenus par des méthodes K-SVD. Il est à noter qu'ici, K-SVD apprend le dictonnaire à partir de l'image

(a) (b) (c)

Figure 8.9 – Illustration des résultats typiques de l'inpainting obtenus en utilisant IBP-DL. De haut en bas, on a le dictionnaire inféré par IBP-DL, l'image masqué et l'image démasqué ; (a) *Peppers* (80% de pixels manquants), PSNR de 6.53 dB à 26.05 dB, (b) *Hill* (50% pixels manquants) PSNR de 8.70 dB à 33.82 dB, (c) *Cameraman* (20% de pixels manquants) PSNR de 12.48 dB à 35.14 dB.

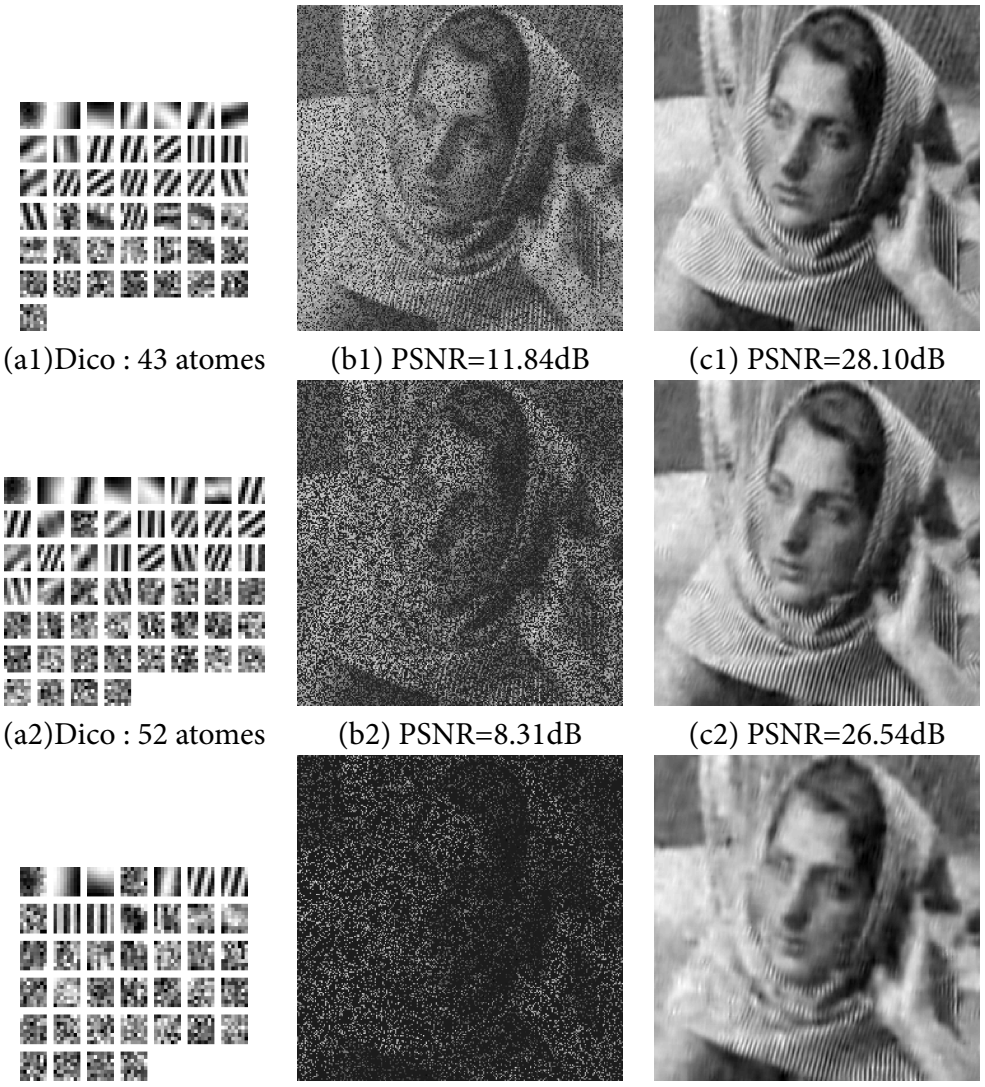
initiale, non masquée. Puis à partir de ce dictionnaire, la méthode OMP est utilisée pour restaurer les patches avant de les recombiner pour reconstruire l'image endommagée. A contrario IBP-DL apprend le dictionnaire à partir de l'image observée (image avec des pixels manquants). On observe que IBP-DL bat K-SVD aussi souvent que K-SVD bat IBP-DL.

8.4.2 Inpainting en présence de bruit

On applique maintenant la méthode IBP-DL à l'inpainting en présence de bruit. Cette expérience est effectuée sur un segment de taille 256×256 de l'image *Barbara* avec les pourcentages de 80%, 50%, 20%, 0% de pixels manquants et les niveaux de bruit de σ_ε=0, 15 et 25.

σ_ε \ Missing		80%	50%	20%	0%
0	IBP-DL	57 - **27.49**	47 - **35.40**	40 - **38.87**	150 - **44.94**
	BPFA	26.87	35.60	40.12	42.94
15	IBP-DL	62 - **25.28**	58 - **28.90**	45 - **30.68**	121 - **31.87**
	BPFA	25.17	29.31	29.93	32.14
25	IBP-DL	39 - **23.74**	52 - **26.54**	43 - **28.10**	67 - **28.90**
	BPFA	23.49	26.79	27.58	29.30

Table 8.4 – Résultats de la restauration sur un segment de *Barbara* en niveau de gris. (En haut) la taille K du dictionnaire IBP-DL et le PSNR (dB) : K - **PSNR**. (En bas) le PSNR (dB) en utilisant BPFA avec 256 atomes pour restaurer l'image.



(a1)Dico : 43 atomes (b1) PSNR=11.84dB (c1) PSNR=28.10dB

(a2)Dico : 52 atomes (b2) PSNR=8.31dB (c2) PSNR=26.54dB

(a3)Dico : 39 atomes (b3) PSNR=6.37dB (c3) PSNR=23.74dB

Figure 8.10 – La restauration du segment de *Barbara*. De gauche à droite, on a le dicitonnaire IBP-DL, l'image observée, l'image restaurée. De haut en bas, on a les restaurations de σ_ε=25 avec 20%, 50% et 80% de pixels manquants.

La figure 8.10 illustre plusieurs exemples de l'inpainting sur un segment de *Barbara*. Les atomes sont affichés par l'ordre décroissant de leur utilisation. Cette expérience montre la pertinence de l'IBP-DL qui propose un dictionnaire adapté et efficace pour l'inpainting, même en présence de bruit supplémentaire.

Le tableau 8.4 regroupe plusieurs exemples de restauration sur le segment de l'image *Barbara*. Il présente la taille K du dictionnaire et le PSNR obtenus en utilisant IBP-DL pour la restauration, pour diverses proportions de pixels manquants (de 0% à 80%) et divers niveaux de bruit (0, 15, 25) pour une image en niveaux de gris vont de 0 à 255 (8 bits). A titre de référence minimale, il est à noter qu'en utilisant uniquement l'atome constant pour la restauration, ce qui est équivalent à un filtre de moyenne locale (ou une interpolation du plus proche voisin), on obtient un PSNR de 22 dB : IBP-DL apporte au moins une amélioration significative par rapport à cette méthode de base. Pour 80% de pixels manquants sans bruit, BPFA donne un PSNR de 26.87 dB lorsque IBP-DL donne un PSNR de 27.49 dB avec 57 atomes. Pour 50% de pixels manquants et σ_ε=25, $\text{PSNR}_{\text{BPFA}}$=26.79 dB et $\text{PSNR}_{\text{IBP-DL}}$=26.54 dB avec K=52.

8.4.3 Inpainting avec masque non aléatoire

La dernière expérience d'inpainting est effectuée sur une image rayée et non avec un masque aléatoire. Les rayures corespondent à 30.55% de pixels manquants. La figure 8.11 illustre l'image *House* rayée avec un PSNR à 10.04 dB. Un dictionnaire de 46 atomes est inféré en utilisant IBP-DL. On obtient finalement une image restaurée à PSNR=38.97 dB. Encore une fois, la pertinence du dictionnaire obtenu à partir de la méthode IBP-DL est de nouveau validée.

(a)PSNR=10.04 dB (b)PSNR=38.97 dB (c)Dico=46 atomes

FIGURE 8.11 – Exemple de l'inpanting pour l'image *House* rayée

8.5 Acquisition compressée

Dans le cadre de l'acquisition compressée, on utilise l'image *Castle* en niveaux de gris (de taille 481×321). Chaque patch $\mathbf{x}_i$ est compressé avec la même matrice de projection gaussienne aléatoire $\mathbf{H}$. Ensuite, nous utilisons un échantillonnage de Gibbs usuel (voir 7.2.1.1) pour l'inférence selon le modèle de la section 7.1. L'image *Castle* a 148836 imagettes (patches) de taille 8×8 qui se chevauchent (*overlapping*), par conséquent en dimension $L = 64$. La matrice de projection $\mathbf{H} \in \mathbb{R}^{Q \times L}$, $Q \leq L$,

(a) (b) (c)

Figure 8.12 – (a) Image *Castle* initiale ; (b) image restorée à partir d'une compression de 50% ($Q = L/2$) avec une erreur de reconstruction relative obtenue par IBP-DL : SNR = 23.9 dB, PSNR = 32.9 dB ; (c) le dictionnaire estimé contient seulement 26 atomes.

est aléatoire avec les coefficients $\mathbf{H}(q,l) \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$. La figure 8.12 affiche la restauration de l'image *Castle* avec un taux de 50% qui est pour $Q = L/2$. Le dictionnaire estimé est constitué de 26 atomes seulement. Ces atomes n'ont pas un bon aspect visuel. Cependant, l'erreur quadratique relative est de 0.004 correspondant à un $SNR = 23.9$ dB et un PSNR $= 32.9$ dB ce qui signifie une très bonne performance de restauration.

8.6 Discussion

Les résultats expérimentaux obtenus avec le modèle IBP-DL sur différents problèmes inverses sont comparables avec les autres méthodes. En particulier, les résultats de reconstruction d'image sont visuellement satisfaisants. Les performances de reconstruction d'image de l'IBP-DL illustrent la pertinence des dictionnaires appris. De plus, on remarque que le bruit est généralement bien estimé. On retient que le nombre d'atomes K appris est expérimentalement souvent plus petit ou légèrement supérieur à la dimension des données (64 ici), ce qui est plutôt surprenant. Toutefois, la taille des dictionnaires ne semble pas si critique vis-à-vis des performances de restauration. Cette observation est à contre courant par rapport aux autres méthodes paramétriques qui fixent généralement la taille du dictionnaire à un nombre grand devant la dimension (typiquement 256).

CHAPITRE 9

Conclusion et Perspectives

Nous résumons dans ce chapitre l'ensemble des contributions et questions ouvertes qui ont été abordées au cours de ce travail. Nous proposons aussi quelques directions de travail et perspectives à court et à moyen termes.

9.1 Conclusion

L'intérêt des méthodes bayésiennes non paramétriques est de plus en plus reconnu pour les méthodes d'apprentissage statistiques (machine learning). Cependant cette thématique n'a pas été beaucoup explorée dans la communauté française. L'objectif de nos travaux est précisément d'explorer le fort potentiel des méthodes bayésiennes non paramétriques de manière générale et pour l'apprentissage de dictionnaire en particulier. Il s'agit d'explorer le potentiel d'un nouveau cadre pour le traitement statistique du signal et des images qui est encore très peu utilisé. Cette thèse est l'une des premières thèses à l'interface entre les modèles bayésiens non paramétriques et le traitement de signal et des images. De façon non exhaustive, voici les thèses sur cette thématique qui ont déjà été soutenues sont [96–98] et la plus récente est la thèse de A.Todeschini [99].

Les représentations parcimonieuses permettent de résoudre les problèmes inverses où l'on est à la limite de l'identifiabilité. Elles consistent à utiliser *un petit nombre d'atomes* dans un dictionnaire pour la reconstruction. Ce dictionnaire peut être appris à partir d'un ensemble de données de référence, ce qu'on appelle l'apprentissage de dictionnaire. Cette thèse s'est attachée à répondre à la question suivante "Comment apprendre un dictionnaire sans fixer sa taille à l'avance ?". Pour cela, une étude des approches bayésiennes non paramétriques a été effectuée afin de proposer une nouvelle méthode d'apprentissage de dictionnaire bayésienne non paramétrique nommée IBP-DL, pour résoudre les problèmes inverses notamment en traitement d'image.

Ce manuscrit peut être partitionné en deux parties : après l'introduction du chapitre 1, les chapitres 2 à 6 ont présenté les notions de parcimonie, d'apprentissage de dictionnaire et les approches bayésiennes non paramétriques. Les chapitres 6, 7 et 8 ont présenté nos contributions originales : le modèle IBP-DL, les algorithmes pour l'échantillonnage et des illustrations expérimentales sur des problèmes inverses en traitement d'image.

Le chapitre 2 s'est intéressé aux représentations parcimonieuses et a présenté quelques méthodes d'optimisation et les bases des méthodes d'échantillonnage aléatoire aussi appelées méthodes de Monte Carlo.

Le chapitre 3 a présenté un état de l'art des méthodes d'apprentissage de dictionnaire pour les représentations parcimonieuses et précisé l'intérêt des approches bayésiennes, en particulier non paramétriques.

Le chapitre 4 s'est intéressé au processus de Dirichlet, et à son utilisation pour les modèles de mélange en particulier. Le lien entre le processus de Dirichlet et les autres distributions non paramétriques comme la représentation de Backwell-Mac Queen ou le processus de restaurant chinois a été aussi présenté. Les propriétés et avantages de ceux-ci on été développés en détail. Même si nous n'avons pas utilisé les modèles de mélange dans notre travail, le chapitre 4 a permis d'introduire les approches bayésiennes non paramétriques qui construisent des lois *a priori* sur des espaces de mesures aléatoires.

Le chapitre 5 a poursuivi avec une deuxième famille d'approches bayésiennes non paramétriques appliquée aux modèles à variables latentes. Ces modèles sont plus adaptés pour l'apprentissage de dictionnaire. On s'est intéressé particulièrement au processus de Buffet Indien qui est un des outils clés de cette thèse. Plusieurs façons de construire le processus du Buffet Indien ont été présentées. Une lien entre le processus Bêta et buffet indien a aussi été présenté.

Le chapitre 6 s'appuie sur une synthèse bibliographique portant sur l'échantillonnage correct du modèle de buffet indien. Ce travail de synthèse n'est détaillé de façon complète que dans nos publications [26, 27].

Dans le chapitre 7, nous avons proposé la méthode IBP-DL en utilisant une approche bayésienne non-paramétrique de type Buffet Indien présentée dans le chapitre 5. L'essentiel de ce travail est publié dans [25–27] et un article est en cours de révision [28]. IBP-DL nous permet d'apprendre un dictionnaire de taille adaptative à partir d'un dictionnaire vide. Par conséquent, un problème de factorisation de matrice est résolu d'une manière non paramétrique. En outre, nous avons formulé le problème de l'apprentissage de dictionnaire dans le contexte des problèmes inverses linéaires avec un bruit Gaussien. Le bruit et le niveau de parcimonie sont aussi inférés. La méthode IBP-DL est pratiquement non paramétrique car aucun paramètre de réglage n'est nécessaire contrairement à la plupart des méthodes d'optimisation ou paramétriques. La sensibilité aux conditions initiales de l'algorithme est assez faible. Les hyperparamètres ont tous des valeurs très faibles et servent à construire des lois les moins informatives possible sur les paramètres. Différents algorithmes MCMC ont été proposés en se basant sur un échantillonnage correct étudié dans le chapitre 6. En particulier, nous avons présenté un échantillonneur de Gibbs marginalisé ainsi qu'un échantillonneur de Gibbs accéléré pour résoudre le problème de l'inpainting d'image. Nous avons également déterminé un estimateur maximum *a posteriori* marginalisé (mMAP) pour le couple dictionnaire et jeu de coefficients.

Dans le chapitre 8, les expériences numériques ont montré la pertinence de l'approche proposée dans le traitement d'image pour le débruitage, l'inpainting ainsi que l'acquisition compressée. La taille des dictionnaires obtenus est très variée. Elle est souvent plus petite ou un peu plus grande que 64 (la dimension des données), ce qui est plutôt surprenant. En effet, les tailles classiquement proposées dans la littérature

sont souvent de 256 ou plus. Il apparaît aussi que la taille du dictionnaire ne semble pas être un paramètre très sensible quant à la qualité de la restauration des images. Toutefois, les performances de IBP-DL restent comparables aux autres méthodes de l'état de l'art qui fixent à l'avance une grande taille de dictionnaire. En plus, le niveau de bruit σ_ϵ est aussi estimé avec une bonne précision. Cela illustre le fort potentiel des méthodes bayésiennes non paramétriques pour l'apprentissage de dictionnaire.

Dans une démarche de recherche reproductible, les codes Matlab et C sont mis à disposition.

9.2 Problèmes ouverts et perspectives

9.2.1 Vers des méthodes d'optimisation

Une des limites de notre algorithme est son coût de calcul dû à l'échantillonnage de Gibbs. Un autre type d'inférence est envisagé pour réduire le temps de calcul. On pense d'abord à *l'approximation bayésienne variationnelle*. L'idée de l'approximation bayésienne variationnelle (BV) est de chercher une loi $\mathrm{q}^{opt}(.)$ la plus proche de la loi *a posteriori* ciblée $\mathrm{p}(.\,|y)$ qui est difficile à calculer. On peut choisir librement la forme de la loi $\mathrm{q}^{opt}(.)$. Cette loi approchant $\mathrm{q}^{opt}(.)$ doit être la plus proche possible de $\mathrm{p}(.\,|y)$ au sens où elle doit minimiser une mesure de dissemblance. Un choix naturel de cette mesure est la divergence de Kullback-Leibler (KL) qui est une mesure de la différence entre 2 densités de probabilité. Une étude de l'inférence variationnelle pour le processus de buffet indien a été présentée dans [95]. À courts termes nous souhaitons explorer ces approximations variationnelles afin de réduire les temps de calculs. Cependant notons au passage que comme ils s'agit d'une approximation, l'algorithme peut ne converger que vers un maximum local. Pour cela, une étude se basant sur les travaux de Doshi *et al.*[95] est nécessaire afin de pouvoir proposer une approximation pertinente pour le modèle IBP-DL.

On pense aussi à explorer de nouveaux modèles conduisant à un jonglage entre les méthodes bayésiennes et d'optimisation. Par exemple, une méthode d'optimisation en utilisant des outils bayésiens a été proposé dans [100]. Nous avons commencé à travailler sur une autre méthode qui se base sur *Small Variance Asymptotics* (SVA) [101, 102]. L'idée consiste à regarder le comportement de l'échantillonneur de Gibbs quand la variance du bruit tend vers 0. Accessoirement, afin de factoriser la vraisemblance, il est nécessaire de coupler certains paramètres, par exemple dans notre cas α et σ_ϵ. À la fin, à partir d'un MAP (Maximum A Posteriori), on espère obtenir une fonction du coût similaire à celle obtenue avec les méthodes d'optimisation. Ce travail est en cours.

Dans un premier temps, on envisage de poser un nouveau modèle liant les méthodes bayésiennes et d'optimisation en utilisant SVA. Cela pourrait nous permettre de trouver les réponses aux heuristiques utilisées dans les méthodes d'optimisation. À moyen terme, on souhaite proposer un algorithme pour minimiser la fonction de coût qui apparaît dans ce nouveau modèle, tout en conservant les intérêts des approches non paramétriques. Cette perspective nécessite un investissement à la fois théorique et expérimental. Nous pensons que cette contribution est importante car

à notre connaissance aucune étude de ce type n'a été menée en apprentissage de dictionnaire.

9.2.2 Comportement en loi de puissance

Dans le chapitre 5, nous avons présenté une version IBP à trois paramètres qui peut prendre en compte un comportement en loi de puissance liant le nombre d'observations utilisant un atome et le nombre d'atomes utilisés par chaque observation. Un autre modèle plus général est proposé par Caron [103] s'appuyant sur les réseaux bipartites ou les réseaux d'affiliation ou de collaboration. Dans ce type de réseaux, les éléments sont divisés en deux types A et B, et seules les connexions entre les éléments de types différents sont autorisées. Des exemples de ce genre peuvent être des acteurs de cinéma jouant dans le même film, des acheteurs choisissant le même produit, des internautes postant un message sur le même forum, des personnes qui lisent le même livre ou écoutent la même musique, etc. Les méthodes BNP permettent de modéliser les relations entre deux types d'entités. Un étude sur ces modèles a été commencée avec F.Caron pendant un séjour à Oxford au printemps 2015.

Dans [99, 103], un modèle bayésien non paramétrique (BNP) a été proposé où chaque élément possède son propre paramètre de sociabilité permettant de capturer le comportement en loi de puissance observé dans les graphes bipartites réels et reproduire des propriétés statistiques fines des données. Dans ce modèle, les éléments de type A sont des lecteurs et les éléments de type B sont des livres. Les méthodes BNP nous permettent de ne pas fixer à l'avance l'ensemble des livres disponibles $\{\theta_j\}$, il peut augmenter à mesure que de nouveaux lecteurs sont ajoutés, sa taille étant potentiellement infinie. L'ensemble des livres lu par le lecteur i peut être représenté par le processus ponctuel suivant :

$$\mathbf{z}_i = \sum_{j=1}^{\infty} z_{ij}\delta_{\theta_j} \tag{9.1}$$

où $z_{ij} = 1$ quand le lecteur i a lu le livre j et 0 sinon. La collection de mesures binaires $(\mathbf{z}_1, ..., \mathbf{z}_n)$ définit l'ensemble des relations entre les lecteurs et les livres. Le modèle BNP de [103] propose :

$$\mathrm{p}(z_{ij} = 1|\omega_j, \gamma_i) \sim Ber(1 - exp(-\gamma_i\omega_j)) \tag{9.2}$$

où les $\omega_j > 0$, (ω_j, θ_j) sont issus d'une mesure complètement aléatoire (CRM) et où chaque lecteur possède son propre paramètre d'intérêt pour la lecture $\gamma_i > 0$. Dans la métaphore du buffet Indien, chaque client i a son propre appétit γ_i et chaque plat j a sa propre popularité ω_j. Ce modèle plus flexible permet une distribution des degrés des lecteurs non Poissonnienne, tout en conservant les propriétés de conjugaison et un processus génératif similaire à l'IBP à trois paramètres (stable).

Le temps d'exécution en Matlab des algorithmes proposés limite la complexification de nos algorithmes. Nous n'avons pas encore pu utiliser ces modèles dans notre méthode d'apprentissage de dictionnaire. A présent, l'ensemble des codes a été réimplémenté en C. A court terme, nous envisageons de retravailler sur ces modèles afin de les implémenter dans l'apprentissage de dictionnaire, notamment en traitement d'image. Nous espérons que l'utilisation de ces modèles permettra de prendre

en compte le comportement en loi puissance de la fréquence d'utilisation des atomes. On obtient en plus un lien entre la texture de chaque patch et le nombre de patches utilisés. On peut imaginer un modèle où les patches ayant une texture pauvre sont représentés par peu d'atomes. Réciproquement, les patches ayant une texture riche seront représentés par de nombreux atomes. Nous pensons que ces modèles ouvre de nouvelles perspectives sur l'apprentissage de dictionnaire et méritent d'être étudier.

9.2.3 Application dans l'image couleur

Une perspective au niveau applicatif a aussi été envisagée à court terme. Pour l'instant, l'application de la méthode IBP-DL en traitement d'image concerne les images en niveaux de gris. Nous voulons ensuite l'implémenter dans le cadre des images couleur. Les images couleurs peuvent être construites par la superposition de 3 couches, par exemple RGB ou encore YCbCr. Dans les approches paramétriques et d'optimisation, les trois couches ne peuvent pas être pas traitées indépendamment. Comme les 256 atomes sont fixés, les patches au même emplacement sur les 3 couches doivent s'associer aux mêmes atomes. Une question sur l'avantage de BNP pour l'image couleur est posée. Comme K n'est pas fixé, les 3 couches peut être traitées séparément. On peut apprendre un dictionnaire $\mathbf{D}$, incluant un jeu de coefficients $\mathbf{W}$ pour chaque couche et les combiner à la fin. Cette perspective devra être explorée au travers d'expériences numériques : celles-ci pourront être développées à partir du code C existant.

9.2.4 Le nombre d'atomes et les données

Nous observons dans le chapitre 8 que le dictionnaire inféré par IBP-DL n'est pas toujours redondant dans le sens où la taille de dictionnaire K est souvent plus petite ou seulement un peu plus grande que la dimension des données (64 ici). On a observé aussi que quand le niveau de bruit est petit on obtient plus d'atomes que quand le niveau de bruit est élevé. Certaines images ont des nombres d'atomes similaires. Certaines images ont plus atomes que les autres. Une fois que les modèles d'apprentissage de dictionnaire utilisant l'IBP à trois paramètres et les réseaux bipartites seront mis en place, les résultats obtenus nous donneront un premier avis sur cette observation. On souhaite ensuite étudier le lien entre le nombre d'atomes et l'information apportée par l'image. Pour l'instant, l'entropie de l'image au sens de Shannon est calculée pixels à pixels. Nous souhaitons introduire un autre type d'entropie utilisant les patches. Cela sera une perspective à moyen terme, voire long terme.

ANNEXE A

Annexes

A.1 Modèle linéaire gaussien avec les variables latentes binaires

A.1.1 Echantillonnage de Gibbs

On calcule la distribution *a posteriori* sur z_{ki} pour l'atome k active, *cf.* partie 6.2.1.

$$\begin{aligned}
&\mathrm{p}(z_{ki}|\mathbf{Y},\mathbf{D},\mathbf{Z}_{-ki},\sigma_\varepsilon) \propto \mathcal{N}(\mathbf{y}_i|\mathbf{D}\mathbf{z}_i,\sigma_\varepsilon^2)P(z_{ki}|\mathbf{Z}_{-ki}) \\
&\propto \exp\left[-\frac{1}{2\sigma_\varepsilon^2}((\mathbf{y}_i-\mathbf{D}\mathbf{z}_i)^{\mathrm{T}}(\mathbf{y}_i-\mathbf{D}\mathbf{z}_i)\right]P(z_{ki}|\mathbf{Z}_{-ki}) \\
&\propto \exp\left[\frac{-1}{2\sigma_\varepsilon^2}(z_{ki}^2\mathbf{d}_k^{\mathrm{T}}\mathbf{d}_k-2z_{ki}\mathbf{d}_k^{\mathrm{T}}(\mathbf{y}_i-\sum_{\substack{j=1\\j\neq k}}^{K}\mathbf{d}_j z_{ji})\right]P(z_{ki}|\mathbf{Z}_{-ki})
\end{aligned}$$

A.1.2 Echantillonnage de Gibbs marginalisé

On cherche à calculer la vraisemblance marginalisée par rapport au dictionnaire $\mathbf{D}$, *cf.* partie 6.3.1.

$$\mathrm{p}(\mathbf{Y}\mid\mathbf{Z},\sigma_D,\sigma_\varepsilon)=\int\mathrm{p}(\mathbf{Y}\mid\mathbf{D},\mathbf{Z},\sigma_\varepsilon)\mathrm{p}(\mathbf{D}\mid\sigma_D)\mathrm{d}\mathbf{D} \tag{A.1}$$

Nous avons :

$$\begin{aligned}
\mathrm{p}(\mathbf{Y}\mid\mathbf{D},\mathbf{Z},\sigma_\varepsilon) &= \frac{1}{(2\pi\sigma_\varepsilon^2)^{NL/2}}\exp(-\frac{1}{2\sigma_\varepsilon^2}\mathrm{tr}[(\mathbf{Y}-\mathbf{D}\mathbf{Z})^{\mathrm{T}}(\mathbf{Y}-\mathbf{D}\mathbf{Z})]) \\
\mathrm{p}(\mathbf{D}\mid\sigma_D) &= \frac{1}{(2\pi\sigma_D^2)^{KL/2}}\exp\left\{-\frac{1}{2\sigma_D^2}\mathrm{tr}[\mathbf{D}^T\mathbf{D}]\right\}
\end{aligned}$$

Le produit de ces deux équations est l'exponentielle de la trace de :

$$\begin{aligned}
&\sigma_\varepsilon^{-2}(\mathbf{Y}-\mathbf{D}\mathbf{Z})(\mathbf{Y}-\mathbf{D}\mathbf{Z})^{\mathrm{T}}+\frac{1}{\sigma_D^2}\mathbf{D}\mathbf{D}^{\mathrm{T}} \\
=&\sigma_\varepsilon^{-2}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}-\sigma_\varepsilon^{-2}\mathbf{Y}\mathbf{Z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}-\sigma_\varepsilon^{-2}\mathbf{D}\mathbf{Z}\mathbf{Y}^{\mathrm{T}}+\sigma_\varepsilon^{-2}\mathbf{D}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}+\frac{1}{\sigma_D^2}\mathbf{D}\mathbf{D}^{\mathrm{T}} \\
=&\sigma_\varepsilon^{-2}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}-\sigma_\varepsilon^{-2}\mathbf{Y}\mathbf{Z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}-\sigma_\varepsilon^{-2}\mathbf{D}\mathbf{Z}\mathbf{Y}^{\mathrm{T}}+\mathbf{D}(\sigma_\varepsilon^{-2}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}+\frac{1}{\sigma_D^2}\mathbb{I})\mathbf{D}^{\mathrm{T}} \\
=&\sigma_\varepsilon^{-2}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}-\sigma_\varepsilon^{-2}\mathbf{Y}\mathbf{Z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}-\sigma_\varepsilon^{-2}\mathbf{D}\mathbf{Z}\mathbf{Y}^{\mathrm{T}}+\mathbf{D}(\sigma_\varepsilon^2\mathbf{M})^{-1}\mathbf{D}^{\mathrm{T}}
\end{aligned} \tag{A.2}$$

$$\text{Avec } \mathbf{M}=(\mathbf{Z}\mathbf{Z}^{\mathrm{T}}+\frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I})^{-1} \tag{A.3}$$

Afin de faciliter l'intégrale sur $\mathbf{D}$, on montre $\mathbf{D}$ sachant $\mathbf{Y}$ et $\mathbf{Z}$ suit une loi Normale. On cherche à réécrire l'équation (A.2) sous la forme :

$$\begin{aligned}&(\mathbf{D}-\boldsymbol{\mu}_{D|Y,Z})\boldsymbol{\Sigma}_{D|Y,Z}^{-1}(\mathbf{D}-\boldsymbol{\mu}_{D|Y,Z})^{\mathrm{T}}+\mathbf{C}\\ =&\mathbf{D}\boldsymbol{\Sigma}_{D|Y,Z}^{-1}\mathbf{D}^{\mathrm{T}}-\mathbf{D}\boldsymbol{\Sigma}_{D|Y,Z}^{-1}\boldsymbol{\mu}_{D|Y,Z}^{\mathrm{T}}-\boldsymbol{\mu}_{D|Y,Z}\boldsymbol{\Sigma}_{D|Y,Z}^{-1}\mathbf{D}^{\mathrm{T}}+\boldsymbol{\mu}_{D|Y,Z}\boldsymbol{\Sigma}_{D|Y,Z}^{-1}\boldsymbol{\mu}_{D|Y,Z}^{\mathrm{T}}+\mathbf{C}\end{aligned} \tag{A.4}$$

L'idée est d'indentifier terme à terme (A.2) et (A.4) :

$$\begin{array}{lcll}\mathbf{D}\boldsymbol{\Sigma}_{D|Y,Z}^{-1}\mathbf{D}^{\mathrm{T}} & \equiv & \mathbf{D}(\sigma_\varepsilon^2\mathbf{M})^{-1}\mathbf{D}^{\mathrm{T}} & \Rightarrow \boldsymbol{\Sigma}_{D|Y,Z}=\sigma_\varepsilon^2\mathbf{M}\\ \boldsymbol{\mu}_{D|Y,Z}\boldsymbol{\Sigma}_{D|Y,Z}^{-1}\mathbf{D}^{\mathrm{T}}=\boldsymbol{\mu}_{D|Y,Z}(\sigma_\varepsilon^2\mathbf{M})^{-1}\mathbf{D}^{\mathrm{T}} & \equiv & \sigma_\varepsilon^{-2}\mathbf{Y}\mathbf{Z}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}} & \Rightarrow \boldsymbol{\mu}_{D|Y,Z}=\mathbf{Y}\mathbf{Z}^{\mathrm{T}}\mathbf{M}\\ \boldsymbol{\mu}_{D|Y,Z}\boldsymbol{\Sigma}_{D|Y,Z}^{-1}\boldsymbol{\mu}_{D|Y,Z}^{\mathrm{T}}+\mathbf{C} & \equiv & \sigma_\varepsilon^{-2}\mathbf{Y}\mathbf{Y}^{\mathrm{T}} & \\ \Leftrightarrow \mathbf{Y}\mathbf{Z}^{\mathrm{T}}\mathbf{M}(\sigma_\varepsilon^2\mathbf{M})^{-1}\mathbf{M}\mathbf{Z}\mathbf{Y}^{\mathrm{T}}+\mathbf{C} & \equiv & \sigma_\varepsilon^{-2}\mathbf{Y}\mathbf{Y}^{\mathrm{T}} & \Rightarrow \mathbf{C}=\sigma_\varepsilon^{-2}(\mathbf{Y}(\mathbb{I}-\mathbf{Z}^{\mathrm{T}}\mathbf{M}\mathbf{Z})\mathbf{Y}^{\mathrm{T}})\end{array}$$

On obtient donc : (A.2) = $(\mathbf{D}-\mathbf{Y}\mathbf{Z}^{\mathrm{T}}\mathbf{M})^{\mathrm{T}}(\sigma_\varepsilon^2\mathbf{M})^{-1}(\mathbf{D}-\mathbf{Y}\mathbf{Z}^{\mathrm{T}}\mathbf{M})+\sigma_\varepsilon^{-2}(\mathbf{Y}(\mathbb{I}-\mathbf{Z}^{\mathrm{T}}\mathbf{M}\mathbf{Z})\mathbf{Y}^{\mathrm{T}})$
L'intégrale extrait la dépendance en $\mathbf{D}$ est :

$$\begin{aligned}&\mathrm{p}(\mathbf{Y}\mid\mathbf{Z},\sigma_D,\sigma_\varepsilon)=\int\mathrm{p}(\mathbf{Y}\mid\mathbf{D},\mathbf{Z},\sigma_\varepsilon)\mathrm{p}(\mathbf{D}\mid\sigma_D)\mathrm{d}\mathbf{D}\\ =&\frac{1}{(2\pi)^{(N+K)L/2}\sigma_\varepsilon^{NL}\sigma_D^{KL}}\exp\{-\frac{1}{2\sigma_\varepsilon^2}\mathrm{tr}[\mathbf{Y}(\mathbb{I}-\mathbf{Z}^{\mathrm{T}}\mathbf{M}\mathbf{Z})\mathbf{Y}^{\mathrm{T}}]\}\\ &\int\exp\{-\frac{1}{2}\mathrm{tr}[(\mathbf{D}-\mathbf{Y}\mathbf{Z}^{\mathrm{T}}\mathbf{M})^{\mathrm{T}}(\sigma_\varepsilon^2\mathbf{M})^{-1}(\mathbf{D}-\mathbf{Y}\mathbf{Z}^{\mathrm{T}}\mathbf{M})]\}\mathrm{d}\mathbf{D}\\ =&\frac{1}{(2\pi)^{(N+K)L/2}\sigma_\varepsilon^{NL}\sigma_D^{KL}}\exp\{-\frac{1}{2\sigma_\varepsilon^2}\mathrm{tr}[\mathbf{Y}(\mathbb{I}-\mathbf{Z}^{\mathrm{T}}\mathbf{M}\mathbf{Z})\mathbf{Y}^{\mathrm{T}}]\}(2\pi)^{KL/2}|\sigma_\varepsilon^2\mathbf{M}|^{L/2}\end{aligned} \tag{A.5}$$

$$=\frac{|\sigma_\varepsilon^2\mathbf{M}|^{L/2}}{(2\pi)^{NL/2}\sigma_\varepsilon^{NL}\sigma_D^{KL}}\exp\{-\frac{1}{2\sigma_\varepsilon^2}\mathrm{tr}[\mathbf{Y}(\mathbb{I}-\mathbf{Z}^{\mathrm{T}}\mathbf{M}\mathbf{Z})\mathbf{Y}^{\mathrm{T}}]\} \tag{A.6}$$

A.1.3 Echantillonnage de Gibbs marginalisé accéléré

On applique le lemme d'inversion matricielle sur la *statistique suffisante* de la loi *a posteriori* de $\mathbf{D}$ afin d'enlever et ajouter facilement l'influence de i, *cf.* 6.4.2. Pour rappel, le lemme d'inservion matricielle est l'équation suivante :

$$(A-BD^{-1}C)^{-1}=A^{-1}+A^{-1}B(D-CA^{-1}B)^{-1}CA^{-1} \tag{A.7}$$

La *statistique suffisante* ou *information form* a la forme suivante :

$$\begin{aligned}g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}&=\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}^{-1}\\ h_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}&=\boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}\,g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}\end{aligned} \tag{A.8}$$

On enlève l'influence de $\mathbf{z}_i et \mathbf{y}_i$ de la façon suivante :

$$\begin{aligned}g_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}&=g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}-\sigma_\varepsilon^{-2}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\\ h_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}&=h_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}-\sigma_\varepsilon^{-2}\mathbf{y}_i\mathbf{z}_i^{\mathrm{T}}\end{aligned} \tag{A.9}$$

En pratique, on veut récupérer $\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}=g_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}^{-1}$. Pour éviter des inversions matricelles qui coûtent chères, on utilise le lemme d'inversion matricielle :

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}&=(g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}-\sigma_\varepsilon^{-2}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}})^{-1}=g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}^{-1}-\frac{\mathbf{z}_i^{\mathrm{T}}g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}^{-1}\mathbf{z}_i-\sigma_\varepsilon^2}{g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}^{-1}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}^{-1}}\\ &=\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}-\frac{\mathbf{z}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}\mathbf{z}_i-\sigma_\varepsilon^2}{\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}}\end{aligned} \tag{A.10}$$

$$\boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}=h_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}g_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}^{-1}=(h_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}-\sigma_\varepsilon^{-2}\mathbf{y}_i\mathbf{z}_i^{\mathrm{T}})\boldsymbol{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} \tag{A.11}$$

La remise de i est comme suit :

$$\begin{aligned} g_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} &= g_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} + \sigma_{\varepsilon}^{-2}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}} \\ h_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} &= h_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} + \sigma_{\varepsilon}^{-2}\mathbf{y}_i\mathbf{z}_i^{\mathrm{T}} \end{aligned} \tag{A.12}$$

La covariance $\mathbf{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}$ peut être récupéré facilement sans inverser la matrice $g^{-1}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}}$:

$$\begin{aligned} \mathbf{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} &= (g_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} + \sigma_{\varepsilon}^{-2}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}})^{-1} = g^{-1}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} - \frac{\mathbf{z}_i^{\mathrm{T}} g^{-1}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}\mathbf{z}_i + \sigma_{\varepsilon}^2}{g^{-1}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}} g_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}} \\ &= \mathbf{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} - \frac{\mathbf{z}_i^{\mathrm{T}}\mathbf{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}\mathbf{z}_i + \sigma_{\varepsilon}^2}{\mathbf{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\mathbf{\Sigma}_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}}} \end{aligned} \tag{A.13}$$

$$\boldsymbol{\mu}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} = h_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} g^{-1}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} = (h_{\mathbf{D}|\mathbf{Y}_{-i},\mathbf{Z}_{-i}} + \sigma_{\varepsilon}^{-2}\mathbf{y}_i\mathbf{z}_i^{\mathrm{T}})\mathbf{\Sigma}_{\mathbf{D}|\mathbf{Y},\mathbf{Z}} \tag{A.14}$$

A.2 Modèle IBP-DL

A.2.1 Echantillonnage de Gibbs

On calcule la postérieure de z_{ki} pour l'atome actif k, voir l'équation (7.15).

$$\begin{aligned} &\mathrm{p}(z_{ki}|\mathbf{Y},\mathbf{H},\mathbf{D},\mathbf{Z}_{-ki},\mathbf{S},\sigma_{\varepsilon}) \propto \mathcal{N}(\mathbf{y}_i|\mathbf{H}_i\mathbf{D}(\mathbf{z}_i\odot\mathbf{s}_i),\sigma_{\varepsilon}^2)P(z_{ki}|\mathbf{Z}_{-ki}) \\ &\propto \exp\left[-\frac{1}{2\sigma_{\varepsilon}^2}((\mathbf{y}_i-\mathbf{H}_i\mathbf{D}(\mathbf{z}_i\odot\mathbf{s}_i))^{\mathrm{T}}(\mathbf{y}_i-\mathbf{H}_i\mathbf{D}(\mathbf{z}_i\odot\mathbf{s}_i))\right]P(z_{ki}|\mathbf{Z}_{-ki}) \\ &\propto \exp\left[\frac{-1}{2\sigma_{\varepsilon}^2}(-2z_{ki}s_{ki}\mathbf{d}_k^{\mathrm{T}}\mathbf{H}_i^{\mathrm{T}}\mathbf{y}_i + z_{ki}s_{ki}\mathbf{d}_k^{\mathrm{T}}\mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i z_{ki}s_{ki}\mathbf{d}_k + 2z_{ki}s_{ki}\mathbf{d}_k^{\mathrm{T}}\mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i\sum_{\substack{j=1\\j\neq k}}^{K}\mathbf{d}_j\mathbf{z}_{ji}\mathbf{s}_{ji})\right]P(z_{ki}|\mathbf{Z}_{-ki}) \\ &\propto \exp\left[\frac{-1}{2\sigma_{\varepsilon}^2}((z_{ki}s_{ki})^2\mathbf{d}_k^{\mathrm{T}}\mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i\mathbf{d}_k - 2z_{ki}s_{ki}\mathbf{d}_k^{\mathrm{T}}\mathbf{H}_i^{\mathrm{T}}(\mathbf{y}_i-\mathbf{H}_i\sum_{\substack{j=1\\j\neq k}}^{K}\mathbf{d}_j z_{ji}s_{ji})\right]P(z_{ki}|\mathbf{Z}_{-ki}) \end{aligned}$$

A.2.2 Echantillonnage de Gibbs marginalisé pour l'inpainting

On calcul la vraisemblance marfginalisée $\mathrm{p}(\mathbf{Y} \mid \{\mathbf{H}_i\},\mathbf{Z},\mathbf{S},\sigma_{\varepsilon},\sigma_D)$ en intégrant le dictionnaire $\mathbf{D}$ dans l'équation (7.27). L'intégration doit être réalisée par rapport aux lignes de $\mathbf{D}$ en raison de la présence de la masque binaire $\mathbf{H}_i$. On rappelle que $\mathcal{F} = \{\mathbf{F}_\ell\}_{\ell=1,\ldots,L}$ est l'ensemble de matrices diagonales binaires de taille N. Si chaque $\mathbf{H}_i$ est associé à chaque $\mathbf{Y}(:,i)$, $\mathbf{F}_\ell$ est associé à chaque $\mathbf{Y}(\ell,:)$,la dimension ℓ des données. $\mathbf{F}_\ell(i,i)$ indique si le pixel à l'emplacement ℓ du patch i est observé ou pas, ainsi $\mathbf{F}_\ell(i,i)$=$\mathbf{H}_i(\ell,\ell)$=$H_{i,\ell}$.

$$\begin{aligned} \mathrm{p}(\mathbf{Y} \mid \{\mathbf{H}_i\},\mathbf{Z},\mathbf{S},\sigma_{\varepsilon},\sigma_D) &= \mathrm{p}(\mathbf{Y} \mid \mathcal{F},\mathbf{Z},\mathbf{S},\sigma_{\varepsilon},\sigma_D) \\ &= \int \mathrm{p}(\mathbf{Y} \mid \mathcal{F},\mathbf{D},\mathbf{Z},\mathbf{S},\sigma_{\varepsilon})\mathrm{p}(\mathbf{D} \mid \sigma_D)\mathrm{d}\mathbf{D} \end{aligned} \tag{A.15}$$

Soit $\mathbf{y}_{\ell,:} = \mathbf{Y}(\ell,:)$, $\mathbf{d}_{\ell,:} = \mathbf{D}(\ell,:)$, on obtient :

$$\begin{aligned} \mathrm{p}(\mathbf{Y} \mid \mathcal{F},\mathbf{D},\mathbf{W},\sigma_{\varepsilon}) &= \frac{1}{(2\pi\sigma_{\varepsilon}^2)^{\|\mathbf{Y}\|_0/2}}\exp\left[-\frac{1}{2\sigma_{\varepsilon}^2}\sum_{l=1}^{P}(\mathbf{y}_{\ell,:}-\mathbf{d}_{\ell,:}\mathbf{W}\mathbf{F}_\ell)(\mathbf{y}_{\ell,:}-\mathbf{d}_{\ell,:}\mathbf{W}\mathbf{F}_\ell)^{\mathrm{T}}\right] \\ \mathrm{p}(\mathbf{D} \mid \sigma_D^2) &= \frac{1}{(2\pi\sigma_D^2)^{KL/2}}\exp\left[-\frac{1}{2\sigma_D^2}\sum_{l=1}^{P}\mathbf{D}(\ell,:)\mathbf{D}(\ell,:)^{\mathrm{T}}\right] \end{aligned}$$

La produit dans l'intégrale de l'équation (A.15) devient

$$\mathrm{p}(\mathbf{Y} \mid \mathcal{F}, \mathbf{D}, \mathbf{W}, \sigma_\varepsilon)\mathrm{p}(\mathbf{D} \mid \sigma_D) = \frac{1}{(2\pi)^{(\|\mathbf{Y}\|_0+KL)/2}\sigma_\varepsilon^{\|\mathbf{Y}\|_0}\sigma_D^{KL}} \tag{A.16}$$
$$\exp\left[-\frac{1}{2}\sum_{l=1}^{L}\left(\sigma_\varepsilon^{-2}(\mathbf{y}_{\ell,:} - \mathbf{d}_{\ell,:}\mathbf{W}\mathbf{F}_\ell)(\mathbf{y}_{\ell,:} - \mathbf{d}_{\ell,:}\mathbf{W}\mathbf{F}_\ell)^{\mathrm{T}} + \frac{1}{\sigma_D^2}\mathbf{d}_{\ell,:}\mathbf{d}_{\ell,:}^{\mathrm{T}}\right)\right]$$

$$\begin{aligned}
&\sigma_\varepsilon^{-2}(\mathbf{y}_{\ell,:} - \mathbf{d}_{\ell,:}\mathbf{W}\mathbf{F}_\ell)(\mathbf{y}_{\ell,:} - \mathbf{d}_{\ell,:}\mathbf{W}\mathbf{F}_\ell)^{\mathrm{T}} + \frac{1}{\sigma_D^2}\mathbf{d}_{\ell,:}\mathbf{d}_{\ell,:}^{\mathrm{T}} \\
=&\sigma_\varepsilon^{-2}\mathbf{y}_{\ell,:}\mathbf{y}_{\ell,:}^{\mathrm{T}} + \sigma_\varepsilon^{-2}\mathbf{d}_{\ell,:}\mathbf{W}\mathbf{F}_\ell\mathbf{F}_\ell^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{d}_{\ell,:}^{\mathrm{T}} - \frac{2}{\sigma_\varepsilon^2}\mathbf{y}_{\ell,:}\mathbf{F}_\ell^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{d}_{\ell,:}^{\mathrm{T}} + \frac{1}{\sigma_D^2}\mathbf{d}_{\ell,:}\mathbf{d}_{\ell,:}^{\mathrm{T}} \\
=&\sigma_\varepsilon^{-2}\mathbf{y}_{\ell,:}\mathbf{y}_{\ell,:}^{\mathrm{T}} + \mathbf{d}_{\ell,:}(\sigma_\varepsilon^{-2}\mathbf{W}\mathbf{F}_\ell\mathbf{F}_\ell^{\mathrm{T}}\mathbf{W}^{\mathrm{T}} + \frac{1}{\sigma_D^2}\mathbb{I}_K)\mathbf{d}_{\ell,:}^{\mathrm{T}} - \frac{2}{\sigma_\varepsilon^2}\mathbf{y}_{\ell,:}\mathbf{F}_\ell^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{d}_{\ell,:}^{\mathrm{T}} \\
=&\sigma_\varepsilon^{-2}\mathbf{y}_{\ell,:}\mathbf{y}_{\ell,:}^{\mathrm{T}} + \mathbf{d}_{\ell,:}(\sigma_\varepsilon^2\mathbf{M}_\ell)^{-1}\mathbf{d}_{\ell,:}^{\mathrm{T}} - \frac{2}{\sigma_\varepsilon^2}\mathbf{y}_{\ell,:}\mathbf{F}_\ell^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{d}_{\ell,:}^{\mathrm{T}} \\
=&(\mathbf{d}_{\ell,:} - \mathbf{y}_{\ell,:}\mathbf{F}_\ell^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{M}_\ell)(\sigma_\varepsilon^2\mathbf{M}_\ell)^{-1}(\mathbf{d}_{\ell,:} - \mathbf{y}_{\ell,:}\mathbf{F}_\ell^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{M}_\ell)^{\mathrm{T}} + \sigma_\varepsilon^{-2}\boldsymbol{\Upsilon}_{\boldsymbol{\ell}}
\end{aligned}$$

où

$$\mathbf{M}_\ell = (\mathbf{W}\mathbf{F}_\ell\mathbf{F}_\ell^{\mathrm{T}}\mathbf{W}^{\mathrm{T}} + \frac{\sigma_\varepsilon^2}{\sigma_D^2}\mathbb{I}_K)^{-1}, \tag{A.17}$$
$$\boldsymbol{\Upsilon}_{\boldsymbol{\ell}} = \mathbf{y}_{\ell,:}(\mathbb{I} - \mathbf{F}_\ell^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{M}_\ell\mathbf{W}\mathbf{F}_\ell)\mathbf{y}_{\ell,:}^{\mathrm{T}}. \tag{A.18}$$

On peut montrer que $\mathbf{d}_{\ell,:} = \mathbf{D}(\ell,:)$ peut être distribué selon une distribution gaussienne :

$$\begin{aligned}
\mathrm{p}(\mathbf{d}_{\ell,:} \mid \mathbf{y}_{\ell,:}, \mathbf{F}_\ell, \mathbf{Z}, \mathbf{S}, \sigma_\varepsilon, \sigma_D) &\propto \mathcal{N}(\boldsymbol{\mu}_{\mathbf{D}(\ell,:)}, \boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)}) \\
\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)} &= \sigma_\varepsilon^2\mathbf{M}_\ell \\
\boldsymbol{\mu}_{\mathbf{D}(\ell,:)} &= \mathbf{y}_{\ell,:}\mathbf{F}_\ell^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{M}_\ell
\end{aligned} \tag{A.19}$$

et

$$\mathrm{p}(\mathbf{D} \mid \mathbf{Y}, \mathcal{F}, \mathbf{Z}, \mathbf{S}, \sigma_\varepsilon, \sigma_D) \propto \prod_{\ell=1}^{L}\mathrm{p}(\mathbf{d}_{\ell,:} \mid \mathbf{y}_{\ell,:}, \mathbf{F}_\ell, \mathbf{Z}, \mathbf{S}, \sigma_\varepsilon, \sigma_D) \tag{A.20}$$

L'intégrale dans l'équation (A.15) donne l'équation (7.27)

$$\begin{aligned}
\mathrm{p}(\mathbf{Y} \mid \mathcal{F}, \mathbf{Z}, \mathbf{S}, \sigma_\varepsilon^2, \sigma_D^2) &= \frac{1}{(2\pi)^{(\|\mathbf{Y}\|_0+KL)/2}\sigma_\varepsilon^{\|\mathbf{Y}\|_0}\sigma_D^{KL}}\exp\left[-\frac{1}{2}\sum_{l=1}^{P}\sigma_\varepsilon^{-2}\boldsymbol{\Upsilon}_{\boldsymbol{\ell}}\right] \\
&\times \int\exp\left[-\frac{1}{2}\sum_{l=1}^{P}\left((\mathbf{d}_{\ell,:} - \boldsymbol{\mu}_{\mathbf{D}(\ell,:)})\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)}^{-1}(\mathbf{d}_{\ell,:} - \boldsymbol{\mu}_{\mathbf{D}(\ell,:)})^{\mathrm{T}}\mathrm{d}\mathbf{D}\right)\right] \\
&= \frac{\prod_{\ell=1}^{P}(2\pi)^{K/2}|\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)}|^{1/2}}{(2\pi)^{(\|\mathbf{Y}\|_0+KL)/2}\sigma_\varepsilon^{\|\mathbf{Y}\|_0}\sigma_D^{KL}}\prod_{\ell=1}^{P}\exp\left[-\frac{1}{2\sigma_\varepsilon^2}\boldsymbol{\Upsilon}_{\boldsymbol{\ell}}\right] \\
&= \frac{1}{(2\pi)^{\|\mathbf{Y}\|_0/2}\sigma_\varepsilon^{\|\mathbf{Y}\|_0-KL}\sigma_D^{KL}}\prod_{\ell=1}^{P}|\mathbf{M}_\ell|^{1/2}\exp\left[-\frac{1}{2\sigma_\varepsilon^2}\boldsymbol{\Upsilon}_{\boldsymbol{\ell}}\right]
\end{aligned} \tag{A.21}$$

A.2.3 Version échantillonnage accélérée pour l'inpainting

Cette partie détaille les calculs utilisés dans la partie 7.2.1.3.

A.2.3.1 Vraisemblance marginalisée

On détaille maintenant l'intégrale utiliée dans l'équation (7.37).

$$\mathrm{p}(\mathbf{Y} \mid \mathcal{H}, \mathbf{W}, \sigma_\varepsilon, \sigma_D) = \int \mathrm{p}(\mathbf{Y} \mid \mathcal{H}, \mathbf{D}, \mathbf{W}, \sigma_\varepsilon)\mathrm{p}(\mathbf{D} \mid \sigma_D)\mathrm{d}\mathbf{D} \tag{A.22}$$

Les données sont réparties en deux sous-ensembles selon $\mathbf{Y} = [\mathbf{y}_i, \mathbf{Y}_{-i}]$, $\mathbf{W} = [\mathbf{w}_i, \mathbf{W}_{-i}]$ et $\mathcal{H} = \{\mathbf{H}_i, \mathcal{H}_{-i}\}$.

$$\begin{aligned}
&\mathrm{p}([\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{w}_i, \sigma_\varepsilon, \sigma_D)\mathrm{p}(\mathbf{Y}_{-i} \mid \mathcal{H}_{-i}, \mathbf{W}_{-i}, \sigma_\varepsilon, \sigma_D) && \text{(A.23)}\\
&\propto \mathrm{p}([\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{w}_i, \sigma_\varepsilon, \mathcal{H}_{-i}, \mathbf{Y}_{-i}, \mathbf{W}_{-i}, \sigma_D) && \text{(A.24)}\\
&\propto \int \mathrm{p}(\mathbf{y}_i, \mathbf{Y}_{-i} \mid \mathbf{H}_i, \mathcal{H}_{-i}, \mathbf{D}, \mathbf{w}_i, \mathbf{W}_{-i}, \sigma_\varepsilon)\mathrm{p}(\mathbf{D} \mid \sigma_D)\mathrm{d}\mathbf{D} && \text{(A.25)}\\
&\propto \int \mathrm{p}(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{D}, \mathbf{w}_i, \sigma_\varepsilon)\mathrm{p}(\mathbf{Y}_{-i} \mid \mathcal{H}_{-i}, \mathbf{D}, \mathbf{W}_{-i}, \sigma_\varepsilon)\mathrm{p}(\mathbf{D} \mid \sigma_D)\mathrm{d}\mathbf{D}
\end{aligned}$$

La vraisemblance $\mathrm{p}(\mathbf{Y}_{-i} \mid \mathcal{H}_{-i}, \mathbf{W}_{-i}, \mathbf{D}, \sigma_\varepsilon)$ et la loi *a priori* $\mathrm{p}(\mathbf{D} \mid \sigma_D)$ sont toutes des gaussiennes. On applique la règle de bayes. La loi *a posteriori* $\mathrm{p}(\mathbf{D} \mid \mathbf{Y}_{-i}, \mathcal{H}_{-i}, \mathbf{W}_{-i}, \sigma_\varepsilon, \sigma_D)$ est aussi une gaussienne d'espérance $\boldsymbol{\mu}_{\mathbf{D}(\ell,:),-i}$ et de covariance $\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:),-i}$

$$\begin{aligned}
&\mathrm{p}([\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{w}_i, \sigma_\varepsilon, \boldsymbol{\mu}_{\mathbf{D}(\ell,:),-i}, \boldsymbol{\Sigma}_{\mathbf{D}(\ell,:),-i})\\
&\propto \int \mathrm{p}(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{D}, \mathbf{w}_i, \sigma_\varepsilon)\mathrm{p}(\mathbf{D} \mid \mathbf{Y}_{-i}, \mathcal{H}_{-i}, \mathbf{W}_{-i}, \sigma_\varepsilon, \sigma_D)\mathrm{d}\mathbf{D}\\
&\propto \int \mathrm{p}(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{D}, \mathbf{z}_i, \mathbf{s}_i, \sigma_\varepsilon) \prod_{\ell=1}^{L} \mathcal{N}(\mathbf{D}(\ell,:); \boldsymbol{\mu}_{\mathbf{D}(\ell,:),-i}, \boldsymbol{\Sigma}_{\mathbf{D}(\ell,:),-i})\mathrm{d}\mathbf{D} && \text{(A.26)}
\end{aligned}$$

$\mathrm{p}(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{D}, \mathbf{w}_i, \sigma_\varepsilon)$ et $\mathrm{p}(\mathbf{D} \mid \mathbf{Y}_{-i}, \mathcal{H}_{-i}, \mathbf{W}_{-i}, \sigma_\varepsilon, \sigma_D)$ sont des gaussiennes alors l'integrale dans l'équation (A.26) donne :

$$\mathrm{p}([\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{w}_i, \sigma_\varepsilon, \boldsymbol{\mu}_{\mathbf{D}(\ell,:),-i}, \boldsymbol{\Sigma}_{\mathbf{D}(\ell,:),-i}) \propto \prod_{\ell=1}^{L} \mathcal{N}(\mathbf{y}_i(\ell); \mu_{yi\ell}, \sigma_{yi\ell}) \tag{A.27}$$

$$\text{où} \quad \begin{aligned} \mu_{yi\ell} &= H_{i,\ell}\boldsymbol{\mu}_{\mathbf{D}(\ell,:),-i}\mathbf{w}_i \\ \sigma_{yi\ell} &= H_{i,\ell}\mathbf{w}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:),-i}\mathbf{w}_i + \sigma_\varepsilon^2 \end{aligned} \tag{A.28}$$

A.2.3.2 Lemme d'inversion matricielle

On a besoin de calculer l'inversion de $g_{\mathbf{D}(\ell,:)}$ et enlever or remettre l'influence de chaque donnée i, voir (7.36).

1. Pour enlever l'influence de donnée i, on a besoin de $\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:),-i} = g^{-1}_{\mathbf{D}(\ell,:),-i}$:

$$\begin{aligned}
g^{-1}_{\mathbf{D}(\ell,:),-i} &= \left(g_{\mathbf{D}(\ell,:)} - \sigma_\varepsilon^{-2} H_{i,\ell}\mathbf{w}_i\mathbf{w}_i^{\mathrm{T}}\right)^{-1}\\
&= g^{-1}_{\mathbf{D}(\ell,:)} + \frac{g^{-1}_{\mathbf{D}(\ell,:)} H_{i,\ell}\mathbf{w}_i\mathbf{w}_i^{\mathrm{T}} g^{-1}_{\mathbf{D}(\ell,:)}}{\sigma_\varepsilon^2 - \mathbf{w}_i^{\mathrm{T}} g^{-1}_{\mathbf{D}(\ell,:)} H_{i,\ell}\mathbf{w}_i}\\
&= g^{-1}_{\mathbf{D}(\ell,:)} - \frac{H_{i,\ell}}{H_{i,\ell}\mathbf{w}_i^{\mathrm{T}} g^{-1}_{\mathbf{D}(\ell,:)}\mathbf{w}_i - \sigma_\varepsilon^2} g^{-1}_{\mathbf{D}(\ell,:)}\mathbf{w}_i\mathbf{w}_i^{\mathrm{T}} g^{-1}_{\mathbf{D}(\ell,:)} && \text{(A.29)}
\end{aligned}$$

2. Remetre l'influence de donnée i pour récupérer $\mathbf{\Sigma}_{\mathbf{D}(\ell,:)} = g^{-1}_{\mathbf{D}(\ell,:)}$ à partir de $\mathbf{\Sigma}_{\mathbf{D}(\ell,:),-i}$:

$$\begin{aligned} g^{-1}_{\mathbf{D}(\ell,:)} &= \left(g_{\mathbf{D}(\ell,:),-i} + \sigma_{\boldsymbol{\varepsilon}}^{-2} H_{i,\ell}\mathbf{w}_i\mathbf{w}_i^{\mathrm{T}}\right)^{-1} \\ &= g^{-1}_{\mathbf{D}(\ell,:),-i} - \frac{g^{-1}_{\mathbf{D}(\ell,:),-i} H_{i,\ell}\mathbf{w}_i\mathbf{w}_i^{\mathrm{T}} g^{-1}_{\mathbf{D}(\ell,:),-i}}{\sigma_{\boldsymbol{\varepsilon}}^2 + \mathbf{w}_i^{\mathrm{T}} g^{-1}_{\mathbf{D}(\ell,:),-i} H_{i,\ell}\mathbf{w}_i} \\ &= g^{-1}_{\mathbf{D}(\ell,:),-i} - \frac{H_{i,\ell}}{H_{i,\ell}\mathbf{w}_i^{\mathrm{T}} g^{-1}_{\mathbf{D}(\ell,:),-i}\mathbf{w}_i + \sigma_{\boldsymbol{\varepsilon}}^2} g^{-1}_{\mathbf{D}(\ell,:),-i}\mathbf{w}_i\mathbf{w}_i^{\mathrm{T}} g^{-1}_{\mathbf{D}(\ell,:),-i} \end{aligned} \tag{A.30}$$

A.2.4 Echantillonnage du dictionnaire

On détaille le calcul de l'équation (7.48). La loi *a posteriori* de $\mathbf{d}_k$ est calculée suivante :

$$\begin{aligned} \mathrm{p}(\mathbf{d}_k|\mathbf{Y},\mathcal{H},\mathbf{Z},\mathbf{S},\mathbf{D}_{-k},\sigma_{\boldsymbol{\varepsilon}},\sigma_D) &\propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i;\mathbf{H}_i\mathbf{D}(\mathbf{s}_i\odot\mathbf{z}_i),\sigma_{\boldsymbol{\varepsilon}}^2\mathbb{I}_{\|\mathbf{H}_i\|_0})\mathcal{N}(\mathbf{d}_k;0,\sigma_D^2\mathbb{I}_L) \\ &\propto \prod_{i=1}^{N} \exp\left[-\frac{1}{2}(\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i)^{\mathrm{T}}\sigma_{\boldsymbol{\varepsilon}}^{-2}(\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i)\right]\exp\left[-\frac{1}{2}(\mathbf{d}_k^{\mathrm{T}}\sigma_D^{-2}\mathbb{I}_L\mathbf{d}_k)\right] \\ &\propto \exp\left[-\frac{1}{2}\left(\sum_{i=1}^{N}(\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i)^{\mathrm{T}}\sigma_{\boldsymbol{\varepsilon}}^{-2}(\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i)+\mathbf{d}_k^{\mathrm{T}}\sigma_D^{-2}\mathbb{I}_L\mathbf{d}_k\right)\right] \end{aligned} \tag{A.31}$$

Le produit des lois normales nous donne une loi normale.
On montrera que $p(\mathbf{d}_k|-) \equiv \mathcal{N}(\boldsymbol{\mu}_{\mathbf{d}_k}, \mathbf{\Sigma}_{\mathbf{d}_k})$.
Maintenant, on travaille avec l'exponentielle dans l'équation (A.31) :

$$\begin{aligned} &\sum_{i=1}^{N}(\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i)^{\mathrm{T}}\sigma_{\boldsymbol{\varepsilon}}^{-2}\mathbb{I}(\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i)+\mathbf{d}_k^{\mathrm{T}}\sigma_D^{-2}\mathbb{I}_L\mathbf{d}_k \\ =&\mathbf{d}_k^{\mathrm{T}}\sigma_D^{-2}\mathbb{I}_L\mathbf{d}_k+\sigma_{\boldsymbol{\varepsilon}}^{-2}\left(\sum_{i=1}^{N}\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_i-2\mathbf{w}_i^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{H}_i^{\mathrm{T}}\mathbf{y}_i+\mathbf{w}_i^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i\mathbf{D}\mathbf{w}_i\right) \\ =&\sigma_{\boldsymbol{\varepsilon}}^{-2}\sum_{i=1}^{N}\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_i-2\left(w_{ki}\mathbf{d}_k^{\mathrm{T}}+\sum_{\substack{j=1\\j\neq k}}^{K}w_{ji}\mathbf{d}_j^{\mathrm{T}}\right)\mathbf{H}_i^{\mathrm{T}}\mathbf{y}_i+ \\ &\left(w_{ki}\mathbf{d}_k^{\mathrm{T}}+\sum_{\substack{j=1\\j\neq k}}^{K}w_{ji}\mathbf{d}_j^{\mathrm{T}}\right)\mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i\left(w_{ki}\mathbf{d}_k+\sum_{\substack{j=1\\j\neq k}}^{K}\mathbf{d}_j w_{ji}\right)+\mathbf{d}_k^{\mathrm{T}}\sigma_D^{-2}\mathbb{I}_L\mathbf{d}_k \end{aligned} \tag{A.32}$$

Il faut identifier l'équation (A.32) à la forme : $\boxed{\mathbf{d}_k^{\mathrm{T}}\mathbf{\Sigma}_{\mathbf{d}_k}^{-1}\mathbf{d}_k - 2\mathbf{d}_k^{\mathrm{T}}\mathbf{\Sigma}_{\mathbf{d}_k}^{-1}\boldsymbol{\mu}_{\mathbf{d}_k} + \textbf{Cst de normalisation}}$.

En simplifiant les termes qui ne contiennent pas $\mathbf{d}_k$ dans eq. (A.32), on obtient :

$$\begin{aligned}
&\mathbf{d}_k^{\mathrm{T}}\sigma_D^{-2}\mathbb{I}_L\mathbf{d}_k + \sigma_{\boldsymbol{\varepsilon}}^{-2}\sum_{i=1}^{N} -2w_{ki}\mathbf{d}_k^{\mathrm{T}}\mathbf{H}_i^{\mathrm{T}}\mathbf{y}_i + w_{ki}\mathbf{d}_k^{\mathrm{T}}\mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i w_{ki}\mathbf{d}_k + 2w_{ki}\mathbf{d}_k^{\mathrm{T}}\mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i\sum_{\substack{j=1\\j\neq k}}^{K}\mathbf{d}_j w_{ji} \\
=&\mathbf{d}_k^{\mathrm{T}}\left(\sigma_D^{-2}\mathbb{I}_L + \sigma_{\boldsymbol{\varepsilon}}^{-2}\sum_{i=1}^{N} w_{ki}^2\mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i\right)\mathbf{d}_k - 2\sigma_{\boldsymbol{\varepsilon}}^{-2}\sum_{i=1}^{N}\mathbf{d}_k^{\mathrm{T}}w_{ki}\left(\mathbf{H}_i^{\mathrm{T}}\mathbf{y}_i - \mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i\sum_{\substack{j=1\\j\neq k}}^{K}\mathbf{d}_j w_{ji}\right) \\
=&\underbrace{\mathbf{d}_k^{\mathrm{T}}\left(\sigma_D^{-2}\mathbb{I}_L + \sigma_{\boldsymbol{\varepsilon}}^{-2}\sum_{i=1}^{N} w_{ki}^2\mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i\right)}_{\boldsymbol{\Sigma}_{\mathbf{d}_k}^{-1}}\mathbf{d}_k - 2\mathbf{d}_k^{\mathrm{T}}\overbrace{\boldsymbol{\Sigma}_{\mathbf{d}_k}^{-1}\boldsymbol{\Sigma}_{\mathbf{d}_k}}^{\mathbb{I}_K}\sigma_{\boldsymbol{\varepsilon}}^{-2}\underbrace{\sum_{i=1}^{N} w_{ki}(\mathbf{H}_i^{\mathrm{T}}\mathbf{y}_i - \mathbf{H}_i^{\mathrm{T}}\mathbf{H}_i\sum_{\substack{j=1\\j\neq k}}^{K}\mathbf{d}_j w_{ji})}_{\boldsymbol{\mu}_{\mathbf{d}_k}}
\end{aligned} \tag{A.33}$$

$$\begin{aligned}
&\mathrm{p}(\mathbf{d}_k|\mathbf{Y},\mathcal{H},\mathbf{Z},\mathbf{S},\mathbf{D}_{-k},\sigma_{\boldsymbol{\varepsilon}},\sigma_D) \propto \mathcal{N}(\boldsymbol{\mu}_{\mathbf{d}_k},\boldsymbol{\Sigma}_{\mathbf{d}_k}) \\
&\boldsymbol{\Sigma}_{\mathbf{d}_k} = (\sigma_D^{-2}\mathbb{I}_L + \sigma_{\boldsymbol{\varepsilon}}^{-2}\sum_{i=1}^{N} w_{ki}^2\mathbf{H}_i^T\mathbf{H}_i)^{-1} \\
&\boldsymbol{\mu}_{\mathbf{d}_k} = \sigma_{\boldsymbol{\varepsilon}}^{-2}\boldsymbol{\Sigma}_{\mathbf{d}_k}\sum_{i=1}^{N} w_{ki}(\mathbf{H}_i^T\mathbf{y}_i - \mathbf{H}_i^T\mathbf{H}_i\sum_{j\neq k}^{K}\mathbf{d}_j w_{ji})
\end{aligned} \tag{A.34}$$

A.2.5 Echantillonnage des coefficients

La loi *a posteriori* de s_{ki} dans l'équation (7.51) peut être calculée de la façon suivante :

$$\begin{aligned}
\mathrm{p}(s_{ki}|\mathbf{Y},\mathbf{H}_i,\mathbf{D},\mathbf{Z},\mathbf{S}_{k,-i},\sigma_{\boldsymbol{\varepsilon}},\sigma_S) &\propto \mathcal{N}(\mathbf{y}_i;\mathbf{H}_i\mathbf{D}(\mathbf{s}_i\odot\mathbf{z}_i),\sigma_{\boldsymbol{\varepsilon}}^2\mathbb{I}_{\|\mathbf{H}_i\|_0})\mathcal{N}(\mathbf{s}_i;0,\sigma_S^{-2}\mathbb{I}_K)) \\
&\propto \exp\left[-\frac{1}{2}\left(\sigma_{\boldsymbol{\varepsilon}}^{-2}(\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i)^T(\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i)] + \sigma_S^{-2}\sum_{j=1}^{K}s_{ij}^2\right)\right]
\end{aligned} \tag{A.35}$$

Idem que $\mathbf{d}_k$, on cherche à montrer $\mathrm{p}(s_{ki} \mid -) \propto \mathcal{N}(\mu_{s_{ki}},\Sigma_{s_{ki}})$. On simplifie les termes qui ne contient pas s_{ki} afin d'identifier la partie dans l'exponentielle de l'équation (A.35) à la forme : $s_{ki}^T\Sigma_{s_{ki}}^{-1}s_{ki} - 2s_{ki}^T\Sigma_{s_{ki}}^{-1}\mu_{s_{ki}} + \textbf{Cst de normalisation}$, on obtient :

$$\begin{aligned}
&\sigma_{\boldsymbol{\varepsilon}}^{-2}(w_{ki}^2\mathbf{d}_k^T\mathbf{H}_i^T\mathbf{H}_i\mathbf{d}_k - 2w_{ki}\mathbf{d}_k^T(\mathbf{H}_i^T\mathbf{y}_i - \mathbf{H}_i^T\mathbf{H}_i\sum_{\substack{j=1\\j\neq k}}^{K}\mathbf{d}_j w_{ji})) + \sigma_S^{-2}s_{ki}^2 \\
=&\sigma_{\boldsymbol{\varepsilon}}^{-2}z_{ki}^2s_{ki}^2\mathbf{d}_k^T\mathbf{H}_i^T\mathbf{H}_i\mathbf{d}_k + \sigma_S^{-2}s_{ki}^2 - 2\sigma_{\boldsymbol{\varepsilon}}^{-2}z_{ki}s_{ki}\mathbf{d}_k^T(\mathbf{H}_i^T\mathbf{y}_i - \mathbf{H}_i^T\mathbf{H}_i\sum_{\substack{j=1\\j\neq k}}^{K}\mathbf{d}_j w_{ji}) \\
=&s_{ki}^2\underbrace{(\sigma_{\boldsymbol{\varepsilon}}^{-2}z_{ki}^2\mathbf{d}_k^T\mathbf{H}_i^T\mathbf{H}_i\mathbf{d}_k + \sigma_S^{-2})}_{\Sigma_{s_{ki}}^{-1}} - 2s_{ki}\underbrace{\overbrace{\Sigma_{s_{ki}}^{-1}\Sigma_{s_{ki}}}^{1}\sigma_{\boldsymbol{\varepsilon}}^{-2}z_{ki}\mathbf{d}_k^T(\mathbf{H}_i^T\mathbf{y}_i - \mathbf{H}_i^T\mathbf{H}_i\sum_{\substack{j=1\\j\neq k}}^{K}\mathbf{d}_j w_{ji})}_{\mu_{s_{ki}}}
\end{aligned}$$

$$
\begin{aligned}
&\mathrm{p}(s_{ki}|\mathbf{Y},\mathbf{H}_i,\mathbf{D},\mathbf{Z},\mathbf{S}_{k,-i},\sigma_{\boldsymbol{\varepsilon}},\sigma_S) \propto \mathcal{N}(\mu_{s_{ki}},\Sigma_{s_{ki}}) \\
&z_{ki}=1 \Rightarrow \begin{cases} \Sigma_{s_{ki}} = (\sigma_{\boldsymbol{\varepsilon}}^{-2}\mathbf{d}_k^T\mathbf{H}_i^T\mathbf{H}_i\mathbf{d}_k + \sigma_S^{-2})^{-1} \\ \mu_{s_{ki}} = \sigma_{\boldsymbol{\varepsilon}}^{-2}\Sigma_{s_{ki}}\mathbf{d}_k^T(\mathbf{H}_i^T\mathbf{y}_i - \mathbf{H}_i^T\mathbf{H}_i\sum_{j\neq k}^{K}\mathbf{d}_j w_{ji}) \end{cases} \\
&z_{ki}=0 \Rightarrow \begin{cases} \Sigma_{s_{ki}} = \sigma_S^2 \\ \mu_{s_{ki}} = 0 \end{cases}
\end{aligned} \tag{A.36}
$$

où $\Sigma_{s_{ki}}$, $\mu_{s_{ki}}$ et s_{ki} sont des scalaires.

A.2.6 Echantillonnage des autres paramètres

La loi *a posteriori* de $\sigma_{\boldsymbol{\varepsilon}}^{-2}$ (voir eq.(7.54)) est :

$$
\begin{aligned}
&\mathrm{p}(\sigma_{\boldsymbol{\varepsilon}}^{-2}|-) \propto \prod_{i=1}^{N}\mathcal{N}(\mathbf{y}_i;\mathbf{H}_i\mathbf{D}(\mathbf{s}_i\odot\mathbf{z}_i),\sigma_Y^2\mathbb{I}_{\|\mathbf{H}_i\|_0})\mathcal{G}(\sigma_{\boldsymbol{\varepsilon}}^{-2};c_0,d_0) \\
&\propto \left(\sigma_{\boldsymbol{\varepsilon}}^{-2}\right)^{\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{H}_i\|_0}\exp\left[-\frac{1}{2\sigma_Y^2}\sum_{i=1}^{N}\|\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_2^2\right]\frac{f_0^{c_0}\left(\sigma_{\boldsymbol{\varepsilon}}^{-2}\right)^{c_0-1}\exp[-d_0\sigma_{\boldsymbol{\varepsilon}}^{-2}]}{\Gamma(c_0)} \\
&\propto \frac{d_0^{c_0}\left(\sigma_{\boldsymbol{\varepsilon}}^{-2}\right)^{c_0+\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{H}_i\|_0-1}\exp[-\sigma_{\boldsymbol{\varepsilon}}^{-2}(d_0+\frac{1}{2\sigma_Y^2}\sum_{i=1}^{N}\|\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_2^2)]}{\Gamma(c_0)} \\
&\Longrightarrow \sigma_{\boldsymbol{\varepsilon}}^{-2}|- \sim \mathcal{G}\left(c_0+\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{H}_i\|_0, d_0+\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i]\|_2^2\right)
\end{aligned} \tag{A.37}
$$

La loi *a posteriori* de σ_S^{-2} (voir eq.(7.55)) est :

$$
\begin{aligned}
&\mathrm{p}(\sigma_S^{-2}|-) \propto \prod_{i=1}^{N}\mathcal{N}(\mathbf{s}_i;0,\sigma_S^2\mathbb{I}_K)\mathcal{G}(\sigma_S^{-2};e_0,f_0) \\
&\propto \prod_{i=1}^{N}(2\pi)^{-\frac{K}{2}}|\sigma_S^{-2}\mathbb{I}_K|^{\frac{1}{2}}\exp[-\frac{1}{2}(\mathbf{s}_i^T\sigma_S^{-2}\mathbb{I}_K\mathbf{s}_i)]\frac{f_0^{e_0}\left(\sigma_S^{-2}\right)^{e_0-1}\exp[-f_0\sigma_S^{-2}]}{\Gamma(e_0)} \\
&\propto \left((2\pi)^{-\frac{KN}{2}}\left(\sigma_S^{-2}\right)^{\frac{KN}{2}}\exp[-\frac{1}{2\sigma_S^2}\sum_{i=1}^{N}(\mathbf{s}_i^T\mathbf{s}_i)]\right)\frac{f_0^{e0}\left(\sigma_S^{-2}\right)^{e_0-1}\exp[-f_0\sigma_S^{-2}]}{\Gamma(e_0)} \\
&\propto \frac{(2\pi)^{-\frac{KN}{2}}f_0^{e_0}\left(\sigma_S^{-2}\right)^{e_0+\frac{KN}{2}-1}\exp[-\sigma_S^{-2}(f_0+\frac{1}{2}\sum_{i=1}^{N}\mathbf{s}_i^T\mathbf{s}_i)]}{\Gamma(e_0)} \\
&\Longrightarrow \sigma_S^{-2}|- \sim \mathcal{G}\left(e_0+\frac{KN}{2}, f_0+\frac{1}{2}\sum_{i=1}^{N}\mathbf{s}_i^T\mathbf{s}_i\right)
\end{aligned} \tag{A.38}
$$

Soit $\mathrm{H}_N = \sum_{j=1}^{N}$, la loi *a posteriori* de α (voir eq.(7.56)) est :

$$\begin{aligned}
&\mathrm{p}(\alpha|-) \propto \mathcal{P}(K \mid \alpha \mathrm{H}_N)\mathcal{G}(\alpha; a_0, b_0) \\
&\exp(-\alpha \mathrm{H}_N)\frac{(\alpha \mathrm{H}_N)^K}{K!} b_0^{a_0} \alpha^{a_0-1} \exp(-b_0\alpha) \\
&\frac{b_0^{a_0}\mathrm{H}_N^K}{K!}\alpha^{a_0+K-1}\exp(-\alpha(\mathrm{H}_N + b_0)) \\
&\Longrightarrow \alpha|- \sim \mathcal{G}\left(a_0 + K, b_0 + \mathrm{H}_N\right)
\end{aligned} \tag{A.39}$$

A.3 Marginalisation par rapport des coefficients

On cherche à calculer la vraisemblance marginalisée par rapport aux coefficients S.

$$\mathrm{p}(\mathbf{Y} \mid \mathbf{Z}, \mathbf{D}, \sigma_S, \sigma_\varepsilon) = \int \mathrm{p}(\mathbf{Y} \mid \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_\varepsilon)\mathrm{p}(\mathbf{S} \mid \sigma_S)\mathrm{dS} \tag{A.40}$$

Nous avons :

$$\begin{aligned}
\mathrm{p}(\mathbf{Y} \mid \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_\varepsilon) &= \frac{1}{(2\pi\sigma_\varepsilon^2)^{NL/2}} \exp(-\frac{1}{2\sigma_\varepsilon^2}\mathrm{tr}[(\mathbf{Y} - \mathbf{D}(\mathbf{Z} \odot \mathbf{S}))^{\mathrm{T}}(\mathbf{Y} - \mathbf{D}(\mathbf{Z} \odot \mathbf{S}))]) \\
\mathrm{p}(\mathbf{S} \mid \sigma_S) &= \frac{1}{(2\pi\sigma_S^2)^{KN/2}} \exp\left\{-\frac{1}{2\sigma_S^2}\mathrm{tr}[\mathbf{S}^T\mathbf{S}]\right\}
\end{aligned}$$

Soit $\hat{\mathbf{D}}_i$ les atomes k du dictionnaire que le donnée i sélectionne.
Soit $\hat{\mathbf{s}}_i$ les coefficients des atomes k que le donnée i utilise effectivement ($z_{ki} = 1$).
Soit m_i le nombre d'atomes que le donnée i utilise. On a :

$$\begin{aligned}
&\mathrm{p}(\mathbf{Y} \mid \mathbf{Z}, \mathbf{D}, \sigma_S, \sigma_\varepsilon) = \frac{1}{2\pi^{(NL+KN)/2}\sigma_\varepsilon^{NL}\sigma_S^{KN}} \\
&\int \exp\left(\sum_{i=}^{N} -\frac{1}{2\sigma_\varepsilon^2}\left((\mathbf{y}_i - \mathbf{D}(\mathbf{z}_i \odot \mathbf{s}_i))^{\mathrm{T}}(\mathbf{y}_i - \mathbf{D}(\mathbf{z}_i \odot \mathbf{s}_i))\right) - \frac{1}{2\sigma_S^2}\mathbf{s}_i^{\mathrm{T}}\mathbf{s}_i\right)\mathrm{dS} \\
=&\frac{1}{2\pi^{(NL+KN)/2}\sigma_\varepsilon^{NL}\sigma_S^{KN}} \exp\left(\sum_{i=1}^{N} -\frac{1}{2\sigma_\varepsilon^2}\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_i\right) \\
&\int \exp\left(-\frac{1}{2}\sum_{i=1}^{N} -\frac{1}{\sigma_\varepsilon^2}\mathbf{y}_i^{\mathrm{T}}\hat{\mathbf{D}}_i\hat{\mathbf{s}}_i - \frac{1}{\sigma_\varepsilon^2}\hat{\mathbf{s}}_i^{\mathrm{T}}\hat{\mathbf{D}}_i^{\mathrm{T}}\mathbf{y}_i + \hat{\mathbf{s}}_i^{\mathrm{T}}\left(\frac{1}{\sigma_\varepsilon^2}\hat{\mathbf{D}}_i^{\mathrm{T}}\hat{\mathbf{D}}_i + \frac{1}{\sigma_S^2}\mathbb{I}_{m_i}\right)\hat{\mathbf{s}}_i\right)\mathrm{d}S
\end{aligned} \tag{A.41}$$

Afin de faciliter l'intégrale sur S, on essaie de montrer la loi *a posteriori* de $\hat{\mathbf{s}}_i$ est une gaussienne $\mathcal{N}(\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i)$. L'idée est d'indentifier les termes dans l'exponentielle de l'équation (A.41) avec

$$\hat{\mathbf{s}}_i^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_i^{-1}\hat{\mathbf{s}}_i - \hat{\mathbf{s}}_i^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_i^{-1}\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}_i^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_i^{-1}\hat{\mathbf{s}}_i + \hat{\boldsymbol{\mu}}_i^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_i^{-1}\hat{\boldsymbol{\mu}}_i + \hat{c}_i$$

Soit $\hat{\mathbf{M}}_i = \left(\hat{\mathbf{D}}_i^{\mathrm{T}}\hat{\mathbf{D}}_i + \frac{\sigma_\varepsilon^2}{\sigma_S^2}\mathbb{I}_{m_i}\right)^{-1}$. On obtient :

$$\hat{\boldsymbol{\Sigma}}_i = \sigma_\varepsilon^2\hat{\mathbf{M}}_i \tag{A.42}$$

$$\hat{\boldsymbol{\mu}}_i = \hat{\mathbf{M}}_i\hat{\mathbf{D}}_i^{\mathrm{T}}\mathbf{y}_i \tag{A.43}$$

$$\hat{c}_i = -\frac{1}{\sigma_\varepsilon^2}\mathbf{y}_i^{\mathrm{T}}\hat{\mathbf{D}}_i\hat{\mathbf{M}}_i^{\mathrm{T}}\hat{\mathbf{D}}_i^{\mathrm{T}}\mathbf{y}_i \tag{A.44}$$

L'équation (A.41) devient :

$$\frac{1}{2\pi^{(NL+KN)/2}\sigma_{\boldsymbol{\varepsilon}}^{NL}\sigma_S^{KN}}\exp\left(\sum_{i=1}^{N}-\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2}\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_i\right)\exp\left(\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2}\sum_{i=1}^{N}\mathbf{y}_i^{\mathrm{T}}\hat{\mathbf{D}}_i\hat{\mathbf{M}}_i^{\mathrm{T}}\hat{\mathbf{D}}_i^{\mathrm{T}}\mathbf{y}_i\right)$$
$$\int\exp\left(-\frac{1}{2}\sum_{i=1}^{N}(\hat{\mathbf{s}}_i-\hat{\boldsymbol{\mu}}_i)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_i^{-1}(\hat{\mathbf{s}}_i-\hat{\boldsymbol{\mu}}_i)\right)\mathrm{d}\hat{\mathbf{s}}_i \tag{A.45}$$

On obtient finalement :

$$\begin{aligned}&\mathrm{p}(\mathbf{Y}\mid\mathbf{Z},\mathbf{D},\sigma_S,\sigma_{\boldsymbol{\varepsilon}})\\ =&\frac{1}{2\pi^{(NL+KN)/2}\sigma_{\boldsymbol{\varepsilon}}^{NL}\sigma_S^{KN}}\exp\left(\sum_{i=1}^{N}-\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2}\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_i\right)\exp\left(\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2}\sum_{i=1}^{N}\mathbf{y}_i^{\mathrm{T}}\hat{\mathbf{D}}_i\hat{\mathbf{M}}_i^{\mathrm{T}}\hat{\mathbf{D}}_i^{\mathrm{T}}\mathbf{y}_i\right)\\ &\prod_{i=1}^{N}2\pi^{m_i/2}\mid\sigma_{\boldsymbol{\varepsilon}}^2\hat{\mathbf{M}}_i\mid^{1/2}\end{aligned} \tag{A.46}$$

A.4 Estimateur du maximum *a posteriori* marginalisé

Soit $\boldsymbol{\theta}=(\sigma_{\boldsymbol{\varepsilon}},\sigma_S,\alpha)$. On cherche :

$$\mathrm{p}(\mathbf{D},\mathbf{Z},\mathbf{S}\mid\mathbf{Y},\mathcal{H})=\int\mathrm{p}(\mathbf{D},\mathbf{Z},\mathbf{S}\mid\mathbf{Y},\mathcal{H},\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}. \tag{A.47}$$

On a :

$$\mathrm{p}(\mathbf{D},\mathbf{Z},\mathbf{S}\mid\mathbf{Y},\boldsymbol{\theta})=\mathrm{p}(\mathbf{Y}\mid\mathcal{H},\mathbf{D},\mathbf{Z},\mathbf{S},\boldsymbol{\theta})\mathrm{p}(\mathbf{D})\mathrm{p}(\mathbf{Z})\mathrm{p}(\mathbf{S}) \tag{A.48}$$
$$\mathrm{p}(\boldsymbol{\theta})=\mathrm{p}(\alpha)\mathrm{p}(\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2})\mathrm{p}(\frac{1}{\sigma_S^2}) \tag{A.49}$$

On a aussi :

$$\mathrm{p}(\mathbf{Y}\mid\mathbf{D},\mathbf{Z},\mathbf{S},\sigma_{\boldsymbol{\varepsilon}})=\frac{1}{(2\pi\sigma_{\boldsymbol{\varepsilon}}^2)^{N_0/2}}\exp\left(-\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2}\sum_{i=1}^{N}\|\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_F^2\right) \tag{A.50}$$

$$\mathrm{p}(\mathbf{D}\mid\sigma_D)=\prod_{k=1}^{K}\frac{1}{(2\pi\sigma_D^2)^{L/2}}\exp\left(-\frac{1}{2\sigma_D^2}\|\mathbf{d}_k\|_2^2\right) \tag{A.51}$$

$$\mathrm{p}(\mathbf{Z}\mid\alpha)=\frac{\alpha^K}{\prod_{h=1}^{2^N-1}K_h!}\exp\left(-\alpha H_N\right)\prod_{k=1}^{K}\frac{(N-m_k)!(m_k-1)!}{N!} \tag{A.52}$$

$$\mathrm{p}(\mathbf{S}\mid\sigma_S)=\prod_{i=1}^{N}\prod_{k=1}^{K}\frac{1}{(2\pi\sigma_S^2)^{1/2}}\exp\left(-\frac{1}{2\sigma_S^2}s_{ki}\right) \tag{A.53}$$

$$\mathrm{p}(\alpha)=\mathcal{G}(1,1) \tag{A.54}$$
$$\mathrm{p}(1/\sigma_{\boldsymbol{\varepsilon}}^2)=\mathcal{G}(c_0,d_0) \tag{A.55}$$
$$\mathrm{p}(1/\sigma_S^2)=\mathcal{G}(e_0,f_0) \tag{A.56}$$

où $\mathcal{G}(x;a,c)=x^{a-1}b^a\exp(-bx)/\Gamma(a)$; $H_N=\sum_{j=1}^{N}\frac{1}{j}$; $N_0=\sum_{i=1}^{N}\|\mathbf{H}_i\|_0$

Marginalisation par rapport à α : Pour effectuer le calcul de l'équation (A.47), on intègre tout d'abord α. Dans ce calcul, on s'intéresse juste les termes contenant α qui se trouvent dans les distributions (A.52) et (A.54). On a $\alpha \sim \mathcal{G}(1,1) = \exp(-\alpha)$.

$$\int_0^\infty \alpha^K \exp(-\alpha H_N) \exp(-\alpha) \mathrm{d}\alpha = \int_0^\infty \alpha^K \exp(-\alpha(H_N+1)) \mathrm{d}\alpha \tag{A.57}$$

On a : $\Gamma(x) = \int_0^\infty t^{x-1} \exp(-t) \mathrm{d}t$

$$\begin{aligned} \int_0^\infty \alpha^K \exp(-\alpha(H_N+1)) \mathrm{d}\alpha &= \frac{\int [(H_N+1)\alpha]^K \exp(-\alpha(H_N+1))(H_N+1)\mathrm{d}\alpha}{(H_N+1)^{K+1}} \\ &= \frac{\Gamma(K+1)}{(H_N+1)^{K+1}} = \frac{K!}{(H_N+1)^{K+1}} \; K \in \mathbb{N} \end{aligned} \tag{A.58}$$

Alors,

$$\begin{aligned} &\int \mathrm{p}(\mathbf{Z}) \mathrm{p}(\alpha) \mathrm{d}\alpha \\ &= \frac{K!}{(H_N+1)^{K+1}} \frac{1}{\prod_{h=1}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{(N-m_k)!(m_k-1)!}{N!} \end{aligned} \tag{A.59}$$

Remarque : $\forall h : K_h = 1$ *avec une forte probabilité surtout quand* N *est très grand.*

Marginalisation par rapport à σ_ε : On cherche ensuite à marginaliser par rapport à la variance du bruit σ_ε^2. Les termes qui contiennet cette variable se trouve dans la vraisemblance (A.50) et la loi *a priori*(A.55). Soit $N_0 = \sum_{i=1}^{N} \|\mathbf{H}_i\|_0$.

$$\begin{aligned} &\int_0^\infty \mathrm{p}(\mathbf{Y} \mid \mathcal{H}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_\varepsilon) \mathrm{p}(1/\sigma_\varepsilon^2) \mathrm{d}\frac{1}{\sigma_\varepsilon^2} \\ &\propto \int \exp\left[-\frac{1}{\sigma_\varepsilon^2} \left(d_0 + \frac{\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{H}_i \mathbf{D} \mathbf{w}_i\|_F^2}{2} \right) \right] \left(\frac{1}{\sigma_\varepsilon^2} \right)^{N_0/2 + c_0 - 1} \mathrm{d}\frac{1}{\sigma_\varepsilon^2} \end{aligned} \tag{A.60}$$

En pratique, on choisit de faibles valeurs pour les hyperparamètres de précisions (c_0=d_0=e_0=f_0=10^{-6}) afin d'obtenir des hyperpriors non-informatives. Alors,

$$\begin{aligned}
&\int_0^\infty \mathrm{p}(\mathbf{Y} \mid \mathcal{H}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_\varepsilon)\mathrm{p}(1/\sigma_\varepsilon^2)\mathrm{d}\frac{1}{\sigma_\varepsilon^2} \\
&\propto \int \exp\left[-\frac{1}{\sigma_\varepsilon^2}\left(\frac{\sum_{i=1}^{N}\|\mathbf{y}_i - \mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_F^2}{2}\right)\right]\left(\frac{1}{\sigma_\varepsilon^2}\right)^{N_0/2-1}\mathrm{d}\frac{1}{\sigma_\varepsilon^2} \\
&\propto \frac{\int \exp\left[-\frac{1}{\sigma_\varepsilon^2}\left(\frac{\sum_{i=1}^{N}\|\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_F^2}{2}\right)\right]\left(\frac{\sum_{i=1}^{N}\|\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_F^2}{2}\right)^{N_0/2-1}\left(\frac{1}{\sigma_\varepsilon^2}\right)^{N_0/2-1}\left(\frac{\sum_{i=1}^{N}\|\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_F^2}{2}\right)\mathrm{d}\frac{1}{\sigma_\varepsilon^2}}{\left(\frac{\sum_{i=1}^{N}\|\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_F^2}{2}\right)^{N_0/2}} \\
&\propto \frac{1}{\left(\frac{\sum_{i=1}^{N}\|\mathbf{y}_i-\mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_F^2}{2}\right)^{N_0/2}}\Gamma(\frac{N_0}{2}) \\
&\propto \left(\sum_{i=1}^{N}\|\mathbf{y}_i - \mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_F\right)^{-N_0/2}
\end{aligned} \tag{A.61}$$

Marginalisation par rapport à σ_S : Idem pour marginaliser par rapport à σ_S :

$$\begin{aligned}
&\int_0^\infty \mathrm{p}(\mathbf{S} \mid \sigma_S)\mathrm{p}(1/\sigma_S^2)\mathrm{d}\frac{1}{\sigma_S^2} \\
&\propto \int \frac{1}{(2\pi)^{NK/2}}\exp\left[-\frac{1}{\sigma_S^2}\left(f_0 + \frac{\|\mathbf{S}\|_F^2}{2}\right)\right]\left(\frac{1}{\sigma_\varepsilon^2}\right)^{NK/2+e_0-1}\mathrm{d}\frac{1}{\sigma_S^2} \\
&\propto \frac{1}{(2\pi)^{NK/2}}\frac{1}{\left(\frac{\|\mathbf{S}\|_F^2}{2}\right)^{NK/2}}\Gamma(\frac{NK}{2}) \\
&\propto \frac{1}{(\pi)^{NK/2}}\frac{1}{(\|\mathbf{S}\|_F)^{NK}}\Gamma(\frac{NK}{2})
\end{aligned} \tag{A.62}$$

On combine les trois équations (A.59),(A.61),(A.62) et (A.51) on obtient :

$$\begin{aligned}
\mathrm{p}(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}) \propto\ & \frac{K!}{(H_N+1)^{K+1}}\frac{1}{\prod_{h=1}^{2^N-1}K_h!}\prod_{k=1}^{K}\frac{(N-m_k)!(m_k-1)!}{N!} \\
& \frac{1}{(2\pi\sigma_D)^{LK}}\exp\left(-\frac{\|\mathbf{D}\|_{\mathrm{F}}^2}{\sigma_D^2}\right)\frac{1}{\left(\sum_{i=1}^{N}\|\mathbf{y}_i - \mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_{\mathrm{F}}^2\right)^{N_0/2}}\frac{1}{(\pi)^{NK/2}}\frac{1}{(\|\mathbf{S}\|_F)^{NK}}\Gamma(\frac{NK}{2})
\end{aligned} \tag{A.63}$$

où $N_0 = \sum_{i=1}^{N}\|\mathbf{H}_i\|_0$, $H_N = \sum_{j=1}^{N}1/j$.

Bibliographie

1. A.E. Hoerl et R. Kennard. Biased estimation for nonorthogonal problems. *Technometrics*, **12** : 55–67, 1970. (cf. p. 2)
2. B.A. Olshausen et D.J. Field. Emergence of simple-cell receptive properties by learning a sparse code for natural images. *Nature*, **381** : 607–609, 1996. (cf. p. 2, 25, 26)
3. R. Gribonval et M. Nielsen. Sparse representations in unions of bases. *IEEE Transactions on Information Theory*, **49** : 3320–3325, 2003. (cf. p. 2)
4. Ivana Tosic et Pascal Frossard. Dictionary Learning : What is the right representation for my signal. *IEEE Signal Processing Magazine*, **28** : 27–38, 2011. (cf. p. 2–4, 25, 31, 77)
5. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society*, **58** : 267–288, 1996. (cf. p. 3, 12)
6. S.S. Chen, D.L. Donoho et M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, **20** : 33–61, 1998. (cf. p. 3, 12, 13)
7. M. Aharon, M. Elad et A. Bruckstein. K-SVD : An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, **54** : 4311–4322, 2006. (cf. p. 3, 4, 25, 27, 31, 95, 96, 99)
8. J. Mairal, F. Bach, J. Ponce, G. Sapiro et A. Zisserman. "Discriminative learned dictionaries for local image analysis" in : *IEEE Conf. on Computer Vision and Pattern Recognition*. 2008. 1–8 (cf. p. 3)
9. B. Ophir, M. Lustig et M. Elad. Multi-scale dictionary learning using wavelets. *IEEE Journal of Selected Topics in Signal Processing*, **5** : 1014–1024, 2011. (cf. p. 3)
10. Gabriel Peyré. A review of adaptive image representations. *IEEE Journal of Selected Topics in Signal Processing*, **5** : 896–911, 2011. (cf. p. 3)
11. R. Mazhar et P.D. Gader. EK-SVD : Optimized dictionary design for sparse representations. *in International Conference on Pattern Recognition*, 1–4, 2008. (cf. p. 4, 28, 90)
12. Jianzhou Feng, Li Song, Xiaokang Yang et Wenjun Zhang. Sub clustering K-SVD : Size variable dictionary learning for sparse representations. *IEEE International Conference on Image Processing*, 2149–2152, 2009. (cf. p. 4, 28, 90)
13. C. Rusu et B. Dumitrescu. Stagewise K-SVD to Design Efficient Dictionaries for Sparse Representations. *IEEE Signal Processing Letters*, **19** : 631–634, 2012. (cf. p. 4, 28, 90)

14. M. MARSOUSI, K. ABHARI, P. BABYN et J. ALIREZAIE. An Adaptive Approach to Learn Overcomplete Dictionaries With Efficient Numbers of Elements. *IEEE Transactions on Signal Processing*, **62** : 3272–3283, 2014. (cf. p. 4, 28, 90, 95, 96, 99)

15. TS FERGUSON. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, **1** : 209–230, 1973. (cf. p. 5, 33, 35, 36)

16. Y. W. TEH. "Dirichlet Processes" in : *Encyclopedia of Machine Learning*. Springer, 2010. (cf. p. 5, 33, 35, 42)

17. F. WOOD et T. L. GRIFFITHS. "Particle filtering for nonparametric Bayesian matrix factorization" in : *Advances in Neural Information Processing Systems 19*. 2007. (cf. p. 6)

18. Ruslan SALAKHUTDINOV et Andriy MNIH. "Bayesian Probabilistic Matrix Factorization using Markov chain Monte Carlo" in : *Proceedings of the 25th International Conference on Machine Learning*. 200. 880–887 (cf. p. 6)

19. S. NAKAJIMA et M. SUGIYAMA. Theoretical Analysis of Bayesian Matrix Factorization. *Journal of Machine Learning Research*, **12** : 2583–2648, 2011. (cf. p. 6)

20. M. GÖNEN, S.A. KHAN et S. KASKI. "Kernelized Bayesian Matrix Factorization" in : *In Proceedings of ICML 2013, the 30th International Conference on Machine Learning*. 2013. 864–872 (cf. p. 6)

21. T.S. AHN, A. KORATTIKARA, N. LIU, S. RAJAN et M. M. WELLING. "Large Scale Distributed Bayesian Matrix Factorization using Stochastic Gradient MCMC" in : *Proceedings of Int. Conf. on Knowledge Discovery and Data Mining, KDD*. 2015. (cf. p. 6)

22. Y. HUANG, J. PAISLEY, Q. LIN, X. DING, X. FU et X. P. ZHANG. Bayesian Nonparametric Dictionary Learning for Compressed Sensing MRI. *IEEE Transactions on Image Processing*, **23** : 5007–5019, 2014. (cf. p. 6)

23. S. SERTOGLU et J. PAISLEY. "Scalable Bayesian nonparametric dictionary learning" in : *Proc. of EUSIPCO*. 2015. (cf. p. 6)

24. H-P. DANG et P. CHAINAIS. "Approche bayésienne non paramétrique dans l'apprentissage du dictionnaire pour adapter le nombre d'atomes" in : *Proceedings of the French National Conference GRETSI*. 2015. (cf. p. 7, 93)

25. H-P. DANG et P. CHAINAIS. A Bayesian non parametric approach to learn dictionaries with adapted numbers of atoms. *IEEE 25th International Workshop on Machine Learning for Signal Processing*, 1–6, 2015. (cf. p. 7, 25, 78, 93, 108)

26. Hong Phuong DANG et Pierre CHAINAIS. Towards Dictionaries of Optimal Size : A Bayesian Non Parametric Approach. *Journal of Signal Processing Systems*, 1–12, 2016. (cf. p. 7, 74, 77, 93, 108)

27. Hong Phuong DANG et Pierre CHAINAIS. Indian Buffet process dictionary learning for image inpainting. *in IEEE Statistical Signal Processing Workshop (SSP)*, 2016. (cf. p. 7, 74, 77, 93, 108)

28. H-P. DANG et P. CHAINAIS. Indian Buffet Process Dictionary Learning : algorithms and applications to image processing. *International Journal of Approximate Reasoning*, preprint. (cf. p. 7, 77, 108)

29. B. K. Natarajan. Sparse Approximate Solutions to Linear Systems. *SIAM Journal on Computing*, **24** : 227–234, 1995. (cf. p. 10)

30. S.G. Mallat et Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, **41** : 3397–3415, 1993. (cf. p. 10)

31. Y. C. Pati, R. Rezaiifar et P. S. Krishnaprasad. Orthogonal matching pursuit : Recursive function approximation with applications to wavelet decomposition. *in : Conference Record of The Twenty-Seventh Asilomar Conference on signals, systems and computers*, 40–44, 1993. (cf. p. 10)

32. D. L. Donoho, Y. Tsaig, I. Drori et J. L. Starck. Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit. *IEEE Transactions on Information Theory*, **58** : 1094–1121, 2012. (cf. p. 11)

33. D. Needell et R. Vershynin. Signal Recovery From Incomplete and Inaccurate Measurements Via Regularized Orthogonal Matching Pursuit. *IEEE Journal of Selected Topics in Signal Processing*, **4** : 310–316, 2010. (cf. p. 11)

34. D. Needell et J. A. Tropp. CoSaMP : Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, **26** : 301–321, 2009. (cf. p. 11)

35. David L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, **41** : 1955. (cf. p. 13)

36. J.J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, **93** : 273–299, 1965. (cf. p. 13)

37. W. J. Fu. Penalized Regressions : The Bridge Versus the Lasso. *Journal of Computational and Graphical Statistics*, 397–416, 1998. (cf. p. 13)

38. I. Daubechies, M. Defrise et C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, **57** : 1413–1457, 2004. (cf. p. 13)

39. Jerome Friedman, Trevor Hastie, Holger Höfling et Robert Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, **2** : 302–332, 2007. (cf. p. 13)

40. Jerome Friedman, Trevor Hastie et Robert Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, **33** : 1–22, 2010. (cf. p. 13)

41. Bradley Efron, Trevor Hastie, Iain Johnstone et Robert Tibshirani. Least angle regression. *The Annals of Statistics*, **32** : 407–499, 2004. (cf. p. 14)

42. C.P. Robert et G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004. (cf. p. 14–16, 65, 82)

43. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth et A.H. Teller ans E. Teller. Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, **21** : 1087–1092, 1953. (cf. p. 16)

44. W.K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, **57** : 97–109, 1970. (cf. p. 16)

45. H. JEFFREYS. An Invariant Form for the Prior Probability in Estimation Problems. *Proceedings of the Royal Society of London A : Mathematical, Physical and Engineering Sciences*, **186** : 453–461, 1946. (cf. p. 18)

46. B.OLSHAUSEN et K.MILLMAN. Learning sparse codes with a mixture-of-gaussians prior. *Advances in Neural Information Processing Systems*, **12** : 841–847, 2000. (cf. p. 20)

47. I. JOLLIFFE. *Principal Component Analysis*. Springer Verlag, 1986. (cf. p. 23)

48. T. HASTIE, R. TIBSHIRANI et J. FRIEDMAN. *The Elements of Statistical Learning (2nd edition)*. Springer-Verlag, 2009. (cf. p. 24)

49. J.F CARDOSO. Blind signal separation : statistical principles. *Proceedings of the IEEE*, **9** : 2009–2025, 1998. (cf. p. 25)

50. Aapo HYVARINEN, Juha KARHUNEN et Erkki OJA. *Independent Component Analysis*. Adaptive and learning systems for signal processing, communications, and control John Wiley, 2001. (cf. p. 25)

51. M. ZHOU, H. CHEN, J. PAISLEY, L. REN, L. LI, Z. XING, D. DUNSON, G. SAPIRO et L. CARIN. Nonparametric Bayesian Dictionary Learning for Analysis of Noisy and Incomplete Images. *IEEE Signal Processing*, **21** : 130 –144, 2012. (cf. p. 25, 29, 31, 32, 95, 96, 99, 100, 102)

52. B.A. OLSHAUSEN et D.J. FIELD. Natural Image Statistics and Efficient Coding. *Network Computation in Neural Systems*, **7** : 333–339, 1996. (cf. p. 26)

53. K. ENGAN, S. O. AASE et J. H. HUSOY. Method of optimal directions for frame design. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, **5** : 2443–2446, 1999. (cf. p. 27)

54. K. ENGAN, B. D. RAO et K. KREUTZ-DELGADO. Frame design using FOCUSS with method of optimal directions (MOD). *Proceedings of Nordic Signal Processing Symposium*, **99** : 1999. (cf. p. 27)

55. K. ENGAN, S. O. AASE et J. H. HUSOY. Multi-frame compression : Theory and design. *Signal Processing*, **80** : 2121–2140, 2000. (cf. p. 27)

56. M. ELAD et M. AHARON. Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries. *IEEE Transactions on Image Processing*, **15** : 3736–3745, 2006. (cf. p. 27, 28, 96, 99)

57. T. L. GRIFFITHS et Z. GHAHRAMANI. "Infinite latent feature models and the Indian buffet process" in : *Advances in NIPS 18*. MIT Press, 2006. 475–482 (cf. p. 32, 47–49, 51, 62, 66, 71, 72, 77)

58. T. L. GRIFFITHS et Z. GHAHRAMANI. The Indian Buffet Process : An Introduction and Review. *Journal of Machine Learning Research*, **12** : 1185–1224, 2011. (cf. p. 32, 47, 48, 51, 62, 64–66, 71, 72, 77)

59. Cédric FÉVOTTE, Nancy BERTIN et Jean-Louis DURRIEU. Nonnegative matrix factorization with the Itakura-Saito divergence : With application to music analysis. *Neural computation*, 2009. (cf. p. 32)

60. C. FÉVOTTE et J. IDIER. Algorithms for Nonnegative Matrix Factorization with the β-Divergence. *Neural Computation*, **23** : 2421–2456, 2011. (cf. p. 32)

61. Parent ÉRIC et Bernier JACQUES. *Le raisonnement bayésien : Modélisation et Inférence*. Spinger, 2007. (cf. p. 33)

62. Carl RASMUSSEN. The Infinite Gaussian Mixture Model. *Advances in Neural Information Processing Systems*, 2000. (cf. p. 33)

63. J. GRIFFIN et M.STELL. Order-based dependent Dirichlet processes. *Journal of the American Statistical Association*, **101** : 179–194, 2006. (cf. p. 33)

64. K. OTA, E. DUFLOS, P. VANHEEGHE et M. YANAGIDA. Speech recognition with speech density estimation by the Dirichlet Process Mixture. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008. (cf. p. 33)

65. M. DAVY et J.Y. TOURNERET. Generative Supervised Classification Using Dirichlet Process Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32** : 1781–1794, 2010. (cf. p. 33)

66. David BLACKWELL et James B. MACQUEEN. Ferguson Distributions Via Polya Urn Schemes. *The Annals of Statistics*, **1** : 1973. (cf. p. 36, 41)

67. D. BLACKWELL. Discreteness of Ferguson selection. *The Annals of Statistics*, **1** : 356–358, 1973. (cf. p. 38)

68. Jayaram SETHURAMAN. A Constructive Definition of Dirichlet Priors. *Statistica Sinica*, **4** : 639–650, 1994. (cf. p. 38)

69. J. PITMAN *Combinatorial stochastic processes* rapp. tech. Department of Statistics, University of California at Berkeley, 2002, 227–242 (cf. p. 38, 59)

70. J. PITMAN et M.YOR. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, **25** : 855–900, 1997. (cf. p. 38, 59)

71. P. MÜLLER et F. A. QUINTANA. Nonparametric Bayesian data analysis. *Statistical Science*, **19** : 95–110, 2004. (cf. p. 42)

72. H. ISHWARAN et LF. JAMES. Gibbs sampling methods for stick-breaking prior. *Journals American Statistical Association*, **96** : 161–173, 2001. (cf. p. 43)

73. David M. BLEI et Michael I. JORDAN. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, **1** : 121–143, 2006. (cf. p. 43)

74. T. KIMURA, T. TOKUDA, Y. NAKADA, T. NOKAJIMA, T. MATSUMOTO et A. DOUCET. Expectation-maximization algorithms for inference in Dirichlet processes mixture. *Pattern Analysis and Applications*, 1–13, 2011. (cf. p. 43)

75. N. M. Laird A. P. DEMPSTER et D. B. RUBIN. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, **39** : 1–38, 1977. (cf. p. 44)

76. Gilles CELEUX, Florence FORBES et Nathalie PEYRARD. EM procedures using mean field-like approximations for Markov model-based image segmentation. *Pattern Recognition*, **36** : 131–144, 2003. (cf. p. 45)

77. A. JARA. Applied Bayesian Non- and Semi-parametric Inference Using DPpackage. *Rnews*, 17–26, 2007. (cf. p. 45)

78. Nils Lid HJORT, Chris HOLMES et Peter Müller ET AL. *Bayesian Nonparametrics : Principles and Practice*. Cambridge series in statistical and probabilistic mathematics Cambridge, New York : Cambridge University Press, 2010. (cf. p. 45)

79. Yee W TEH, Dilan GÖRÜR et Zoubin GHAHRAMANI. "Stick-breaking construction for the Indian buffet process" in : *International Conference on Artificial Intelligence and Statistics*. 2007. 556–563 (cf. p. 52)

80. Romain THIBAUX et Michael I. JORDAN. Hierarchical beta processes and the Indian buffet process. *Practical Nonparametric and Semiparametric Bayesian Statistics*, **2007** : 227–242, 2007. (cf. p. 54, 56)

81. Z. GHAHRAMANI, T. L. GRIFFITHS et P. SOLLICH. Bayesian nonparametric latent feature models. *Bayesian Statistic*, **8** : 2007. (cf. p. 57, 59)

82. M. E. NEWMAN. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, **46** : 323–351, 2005. (cf. p. 59)

83. S. GOLDWATER, T. GRIFFITHS et M. JOHNSON. Interpolating between types and tokens by estimating power-law generetors. *Advances in Neural Information Processing Systems*, **18** : 2006. (cf. p. 59)

84. T. COHN, P. BLUNSOM et S. GOLDWATER. Inducing tree-substitution grammars. *Journal of Machine Learning Research*, 3053–3096, 2010. (cf. p. 59)

85. Erik B. SUDDERTH et Michael I. JORDAN. "Shared Segmentation of Natural Scenes Using Dependent Pitman-Yor Processes" in : *Advances in Neural Information Processing Systems 21*. sous la dir. de D. KOLLER, D. SCHUURMANS, Y. BENGIO et L. BOTTOU Curran Associates, Inc., 2009. 1585–1592 (cf. p. 59)

86. M.D. FALL, E. BARAT, A. MOHAMMAD-DJAFARI et C. COMTAT. "Spatial emission tomography reconstruction using Pitman-Yor process" in : *Proceedings of the 29th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*. t. 1193 2009. 194–201 (cf. p. 59)

87. Y.W. TEH et D. GÖRÜR. "Indian Buffet Processes with Power-law Behavior" in : *NIPS*. 2009. (cf. p. 59)

88. Finale DOSHI-VELEZ et Zoubin GHAHRAMANI. "Accelerated sampling for the Indian buffet process" in : *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*. 2009. 273–280 (cf. p. 62, 66, 67, 70, 72)

89. David ANDRZEJEWSKI *Accelerated Gibbs Sampling for Infinite Sparse Factor Analysis* rapp. tech. Lawrence Livermore National Laboratory (LLNL), Livermore, CA, 2011 (cf. p. 66, 69)

90. Chirstopher M. BISHOP. *Pattern Recognition and Machine Learning*. Spinger, 2007. (cf. p. 69)

91. D A KNOWLES et Z GHAHRAMANI. Nonparametric Bayesian sparse factor models with application to gene expression modeling. *The Annals of Applied Statistics*, **5** : 1534–1552, 2011. (cf. p. 71, 73, 81)

92. David A VAN DYK et Taeyoung PARK. Partially Collapsed Gibbs Samplers. *Journal of the American Statistical Association*, **103** : 790–796, 2008. (cf. p. 82)

93. A. BUADES, B. COLL et J.M. MOREL. A Review of Image Denoising Algorithms, with a New One. *Multiscale Modeling & Simulation*, **4** : 490–530, 2005. (cf. p. 95, 96)

94. K. DABOV, A. FOI, V. KATKOVNIK et K. EGIAZARIAN. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Trans. Image Process.*, **16** : 2080–2095, 2007. (cf. p. 95, 96)

95. Finale Doshi, Kurt Miller, Jurgen Van Gael et Yee Whye Teh. "Variational Inference for the Indian Buffet Process" in : *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009*. 2009. 137–144 (cf. p. 101, 109)

96. Francois. Caron. *Inférence bayésienne pour la détermination et la sélection de modèles stochastiques*. thèse de doct. Ecole Centrale de Lille; Université des Sciences et Technologie de Lille-Lille I, 2006. (cf. p. 107)

97. Mame Diarra Fall. *Modélisation stochastique de processus pharmaco-cinétiques, application à la reconstruction tomographique par émission de positrons (TEP) spatio-temporelle*. thèse de doct. Université Paris Sud, 2012. (cf. p. 107)

98. Nouha Jaoua. *Estimation bayésienne non paramétrique de systèmes dynamiques en présence de bruits alpha-stables*. thèse de doct. Ecole Centrale de Lille, 2013. (cf. p. 107)

99. Adrien Todeschini. *Probabilistic and Bayesian nonparametric approaches for recommender systems and networks*. thèse de doct. INRIA Bordeaux- Sud-Ouest ; Institut de Mathématiques de Bordeaux ; Université de Bordeaux, 2016. (cf. p. 107, 110)

100. Onur Dikmen et Cédric Févotte. Maximum marginal likelihood estimation for nonnegative dictionary learning in the Gamma-Poisson Model. *IEEE Transactions on Signal Processing*, **60** : 5163–5175, 2012. (cf. p. 109)

101. Ke Jiang, Brian Kulis et Michael I. Jordan. "Small-Variance Asymptotics for Exponential Family Dirichlet Process Mixture Models" in : *Advances in Neural Information Processing Systems 25* sous la dir. de F. Pereira, C. J. C. Burges, L. Bottou et K. Q. Weinberger. Curran Associates, Inc., 2012. 3158–3166 (cf. p. 109)

102. Tamara Broderick, Brian Kulis et Michael Jordan. "MAD-Bayes : MAP-based Asymptotic Derivations from Bayes" in : *Proceedings of The 30th International Conference on Machine Learning*. 2013. 226–234 (cf. p. 109)

103. François Caron. "Bayesian nonparametric models for bipartite graphs" in : *Proc. of NIPS*. 2012. (cf. p. 110)

Approches bayésiennes non paramétriques et apprentissage de dictionnaire pour les problèmes inverses en traitement d'image

L'apprentissage de dictionnaire pour la représentation parcimonieuse est bien connu dans le cadre de la résolution de problèmes inverses. Les méthodes d'optimisation et les approches paramétriques ont été particulièrement explorées. Ces méthodes rencontrent certaines limitations, notamment liées au choix de paramètres. En général, la taille de dictionnaire doit être fixée à l'avance et une connaissance des niveaux de bruit et éventuellement de parcimonie sont aussi nécessaires. Les contributions méthodologiques de cette thèse concernent l'apprentissage conjoint du dictionnaire et de ses paramètres, notamment pour les problèmes inverses en traitement d'image. Nous étudions et proposons la méthode IBP-DL (Indien Buffet Process for Dictionary Learning) en utilisant une approche bayésienne non paramétrique. Une introduction sur les approches bayésiennes non paramétriques est présentée. Le processus de Dirichlet et son dérivé, le processus du restaurant chinois, ainsi que le processus Bêta et son dérivé, le processus du buffet indien, sont décrits. Le modèle proposé pour l'apprentissage de dictionnaire s'appuie sur un a priori de type Buffet Indien qui permet d'apprendre un dictionnaire de taille adaptative. Nous détaillons la méthode de Monte-Carlo proposée pour l'inférence. Le niveau de bruit et celui de la parcimonie sont aussi échantillonnés, de sorte qu'aucun réglage de paramètres n'est nécessaire en pratique. Des expériences numériques illustrent les performances de l'approche pour les problèmes du débruitage, de l'inpainting et de l'acquisition compressée. Les résultats sont comparés avec l'état de l'art. Le code source en Matlab et en C est mis à disposition.
Mots-clés : représentations parcimonieuses, apprentissage de dictionnaire, prolèmes inverses, bayésien non paramétrique, processus du Buffet Indien, Monte-Carlo par chaînes de Markov

Bayesian nonparametric approaches and dictionary learning for inverse problems in image processing

Dictionary learning for sparse representation has been widely advocated for solving inverse problems. Optimization methods and parametric approaches towards dictionary learning have been particularly explored. These methods meet some limitations, particularly related to the choice of parameters. In general, the dictionary size is fixed in advance, and sparsity or noise level may also be needed. In this thesis, we show how to perform jointly dictionary and parameter learning, with an emphasis on image processing. We propose and study the Indian Buffet Process for Dictionary Learning (IBP-DL) method, using a bayesian nonparametric approach. A primer on bayesian nonparametrics is first presented. Dirichlet and Beta processes and their respective derivatives, the Chinese restaurant and Indian Buffet processes are described. The proposed model for dictionary learning relies on an Indian Buffet prior, which permits to learn an adaptive size dictionary. The Monte-Carlo method for inference is detailed. Noise and sparsity levels are also inferred, so that in practice no parameter tuning is required. Numerical experiments illustrate the performances of the approach in different settings : image denoising, inpainting and compressed sensing. Results are compared with state-of-the art methods is made. Matlab and C sources are available for sake of reproducibility.
Keywords : sparse representations, dictionary learning, inverse problems, Bayesian nonparametric, Indian Buffet Process, Markov chain Monte Carlo