Université de Montréal

Analyse en composantes indépendantes avec une matrice de mélange éparse

par

Marc-Olivier Billette

Département de mathématiques et de statistique
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Statistique

juin 2013

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé

Analyse en composantes indépendantes avec une matrice de mélange éparse

présenté par

Marc-Olivier Billette

a été évalué par un jury composé des personnes suivantes :

Alejandro Murua

(président-rapporteur)

Pierre Lafaye de Micheaux

(directeur de recherche)

Martin Bilodeau

(membre du jury)

Mémoire accepté le:

20 juin 2013

SOMMAIRE

L'analyse en composantes indépendantes (ACI) est une méthode d'analyse statistique qui consiste à exprimer les données observées (mélanges de sources) en une transformation linéaire de variables latentes (sources) supposées non gaussiennes et mutuellement indépendantes. Dans certaines applications, on suppose que les mélanges de sources peuvent être groupés de façon à ce que ceux appartenant au même groupe soient fonction des mêmes sources. Ceci implique que les coefficients de chacune des colonnes de la matrice de mélange peuvent être regroupés selon ces mêmes groupes et que tous les coefficients de certains de ces groupes soient nuls. En d'autres mots, on suppose que la matrice de mélange est éparse par groupe. Cette hypothèse facilite l'interprétation et améliore la précision du modèle d'ACI. Dans cette optique, nous proposons de résoudre le problème d'ACI avec une matrice de mélange éparse par groupe à l'aide d'une méthode basée sur le LASSO par groupe adaptatif, lequel pénalise la norme $\mathfrak{l}_1$ des groupes de coefficients avec des poids adaptatifs. Dans ce mémoire, nous soulignons l'utilité de notre méthode lors d'applications en imagerie cérébrale, plus précisément en imagerie par résonance magnétique. Lors de simulations, nous illustrons par un exemple l'efficacité de notre méthode à réduire vers zéro les groupes de coefficients non-significatifs au sein de la matrice de mélange. Nous montrons aussi que la précision de la méthode proposée est supérieure à celle de l'estimateur du maximum de la vraisemblance pénalisée par le LASSO adaptatif dans le cas où la matrice de mélange est éparse par groupe.

Mots clés : Analyse en composantes indépendantes, matrice de mélange éparse, séparation aveugle de sources, LASSO par groupe adaptatif.

SUMMARY

Independent component analysis (ICA) is a method of statistical analysis where the main goal is to express the observed data (mixtures) in a linear transformation of latent variables (sources) believed to be non-Gaussian and mutually independent. In some applications, the mixtures can be grouped so that the mixtures belonging to the same group are function of the same sources. This implies that the coefficients of each column of the mixing matrix can be grouped according to these same groups and that all the coefficients of some of these groups are zero. In other words, we suppose that the mixing matrix is sparse per group. This assumption facilitates the interpretation and improves the accuracy of the ICA model. In this context, we propose to solve the problem of ICA with a sparse group mixing matrix by a method based on the adaptive group LASSO. The latter penalizes the norm l_1 of the groups of coefficients with adaptive weights. In this thesis, we point out the utility of our method in applications in brain imaging, specifically in magnetic resonance imaging. Through simulations, we illustrate with an example the effectiveness of our method to reduce to zero the non-significant groups of coefficients within the mixing matrix. We also show that the accuracy of the proposed method is greater than the one of the maximum likelihood estimator with an adaptive LASSO penalization in the case where the mixing matrix is sparse per group.

Keywords : Independent component analysis, Sparse mixing matrix, Blind source separation, Adaptive group LASSO.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

REMERCIEMENTS

Tout d'abord, j'aimerais remercier mon directeur de recherche Pierre Lafaye de Micheaux avec qui ce fut un plaisir de travailler sur ce mémoire. J'aimerais souligner sa disponibilité, son soutien ainsi que son acharnement pour le succès de ses étudiants. Je le remercie principalement de m'avoir encouragé tout au long de ce mémoire, pour finalement m'avoir donné le goût de la recherche. Les nombreuses discussions que nous avons eues durant ces dernières années m'ont beaucoup aidé en tant qu'étudiant et en tant que personne.

J'aimerais aussi remercier tout le personnel du département sans qui je n'aurais pas pu passer un aussi beau séjour à l'Université de Montréal. Je remercie également mes amis les collègues de travail pour les bons moments passés ensemble et pour les moments à venir.

Je remercie mes parents qui ont toujours tout mis sur mon chemin depuis que je suis tout petit afin que je puisse faire ce que j'aime dans la vie. Merci.

Je prends le temps de remercier spécialement ma copine Sandra pour tout ce qu'elle a fait pour moi durant ma maîtrise. Elle a su être là pour m'encourager et me supporter. Je l'apprécie infiniment.

Finalement, je remercie la vie pour tout ce qu'elle m'apporte.

INTRODUCTION

Depuis le début des années 1980, de plus en plus de chercheurs se sont penchés sur le problème dit de séparation aveugle de sources, notamment le célèbre problème de la soirée cocktail (*cockail party problem*), où il s'agit d'extraire les conversations individuelles des convives à partir d'enregistrements sonores contenant un mélange de ces conversations. Plusieurs méthodes statistiques permettant de résoudre ce problème ont fait surface telles que l'analyse en composantes indépendantes (ACI), la poursuite de projections ainsi que l'infomax. La modélisation sous-jacente en ACI consiste à exprimer les données observées, aussi appelées mélanges de sources (*mixtures*), en une transformation linéaire de variables latentes supposées non gaussiennes et mutuellement indépendantes. Le but de l'ACI est de retrouver ces variables latentes, communément appelées sources. On peut voir l'ACI comme étant une extension de l'analyse en composantes principales et de l'analyse de facteurs dans le sens où elle trouve des composantes qui sont mutuellement indépendantes au lieu d'être seulement mutuellement non-corrélées. L'ACI est couramment utilisée dans plusieurs domaines d'application, entre autres en imagerie cérébrale, en traitement d'images et en économétrie.

Dans la littérature, il existe plusieurs ouvrages traitant de l'analyse en composantes indépendantes. Parmi ces ouvrages, on retrouve notamment Comon (1994) et Hyvärinen et coll. (2001). Cependant, dans ces ouvrages, les auteurs ne considèrent pas le cas où la matrice de mélange est éparse, ce qui est fréquent dans de nombreuses applications. C'est ce qu'on appelle le problème d'ACI avec une matrice de mélange éparse. Afin de résoudre ce problème, Zhang et Chan (2006) ont proposé une vraisemblance pénalisée. Dans cet article, les auteurs utilisent la pénalisation LASSO, la pénalisation SCAD de même qu'une forme généralisée de cette dernière. Par contre, il existe deux inconvénients à leur approche. Premièrement, la pénalisation LASSO ne possède pas les propriétés oracles d'une bonne fonction de pénalité mentionnées dans Fan et Li (2001), c'est-à-dire entre autres que l'estimation des coefficients non-nuls est biaisée. Deuxièmement, puisque la

validation croisée ne peut s'appliquer dans un contexte d'ACI, alors il n'existe aucune façon de déterminer la valeur du multiplicateur de Lagrange qui apparaît dans la vraisemblance pénalisée. Suite à ces inconvénients, les auteurs Zhang et coll. (2009) ont proposé une vraisemblance pénalisée par le LASSO adaptatif. Ce dernier possède les propriétés oracles. De plus, dans cet article, les auteurs fixent le multiplicateur de Lagrange au coefficient qui effectuerait une sélection de variables semblable à celle du BIC. Ainsi, les auteurs obtiennent de très bons résultats dans le cas où la matrice de mélange est éparse.

Dans ce mémoire, nous proposons une version modifiée de l'approche de Zhang et coll. (2009) dans le cas où la matrice de mélange est éparse par groupe. Ainsi, nous proposons de résoudre le problème d'ACI avec une matrice de mélange éparse par une vraisemblance pénalisée par le LASSO par groupe adaptatif. Tout comme le LASSO adaptatif, le LASSO par groupe adaptatif possède les propriétés oracles. Ce dernier effectue l'estimation de la matrice de mélange tout en réduisant vers zéro les groupes de coefficients non-significatifs au sein de la matrice de mélange. Nous démontrons ensuite son utilité lors d'applications en imagerie cérébrale, plus précisément en imagerie par résonnance magnétique. Nous démontrons aussi son efficacité par simulation de petits et de grands jeux de données.

Le mémoire est organisé de la façon suivante. Dans le chapitre 1, on établit le contexte théorique de l'analyse en composantes indépendantes en définissant notamment le modèle et les objectifs de l'ACI. Puis, les chapitres 2 et 3 couvrent des notions utiles à la résolution du problème d'ACI, soit la théorie de l'information, l'analyse en composantes principales et le blanchiment des données. Le chapitre 4 explique les différentes méthodes d'ACI couramment utilisées. Ensuite, le chapitre 5 décrit plusieurs pénalisations dans un contexte de régression linéaire, dont la majorité sera appliquée dans un contexte d'ACI au chapitre suivant. À la dernière section du chapitre 6, nous proposons une méthode d'ACI avec une matrice de mélange éparse basée sur le LASSO par groupe adaptatif. Enfin, au chapitre 7, la méthode proposée est appliquée à divers jeux de données et les résultats obtenus sont dévoilés.

Chapitre 1

CONTEXTE THÉORIQUE

1.1. Séparation aveugle de sources

Le problème de la séparation aveugle de sources explique bien la genèse de l'analyse en composantes indépendantes (ACI). L'explication de ce problème aide à mieux comprendre le modèle d'ACI, qui sera expliqué à la section 1.2. Considérons une situation où il y a un certain nombre de signaux qui sont émis par des objets ou des sources physiques. Ces sources peuvent être, par exemple, des signaux électriques émis par une partie du cerveau (EEG), des enregistrements sonores de personnes ayant une discussion dans une même pièce ou bien des ondes radio émises par des téléphones cellulaires. Supposons aussi qu'il y a plusieurs capteurs ou récepteurs et que les « positions » de ces capteurs soient différentes, de façon à ce que chaque enregistrement représente un mélange des signaux sources avec des poids légèrement différents. Afin d'être plus explicite, considérons le fameux problème de la soirée cocktail où il s'agit d'extraire les conversations individuelles de 3 personnes à partir d'autant d'enregistrements sonores effectués simultanément pendant la soirée. Chacune d'entre elles émet des signaux vocaux, appelés sources, et leurs enregistrements sont dénotés respectivement par s_{1i}, s_{2i} et s_{3i}, $i = 1, \ldots, n$ où n représente le nombre d'observations. De même, les enregistrements sonores sont appelés mélanges de sources et sont dénotés par x_{1i}, x_{2i} et x_{3i}, $i = 1, \ldots, n$. On peut exprimer les réalisations des mélanges de sources x_{ij} comme étant issues d'une transformation linéaire des réalisations des sources s_{ij} dont les coefficients a_{ij} dépendent des distances qui séparent chacune des personnes des capteurs de sons. Ainsi, nous avons

$$\begin{aligned} x_{1i} &= a_{11}s_{1i} + a_{12}s_{2i} + a_{13}s_{3i} \\ x_{2i} &= a_{21}s_{1i} + a_{22}s_{2i} + a_{23}s_{3i} \\ x_{3i} &= a_{31}s_{1i} + a_{32}s_{2i} + a_{33}s_{3i}, \end{aligned} \tag{1.1.1}$$

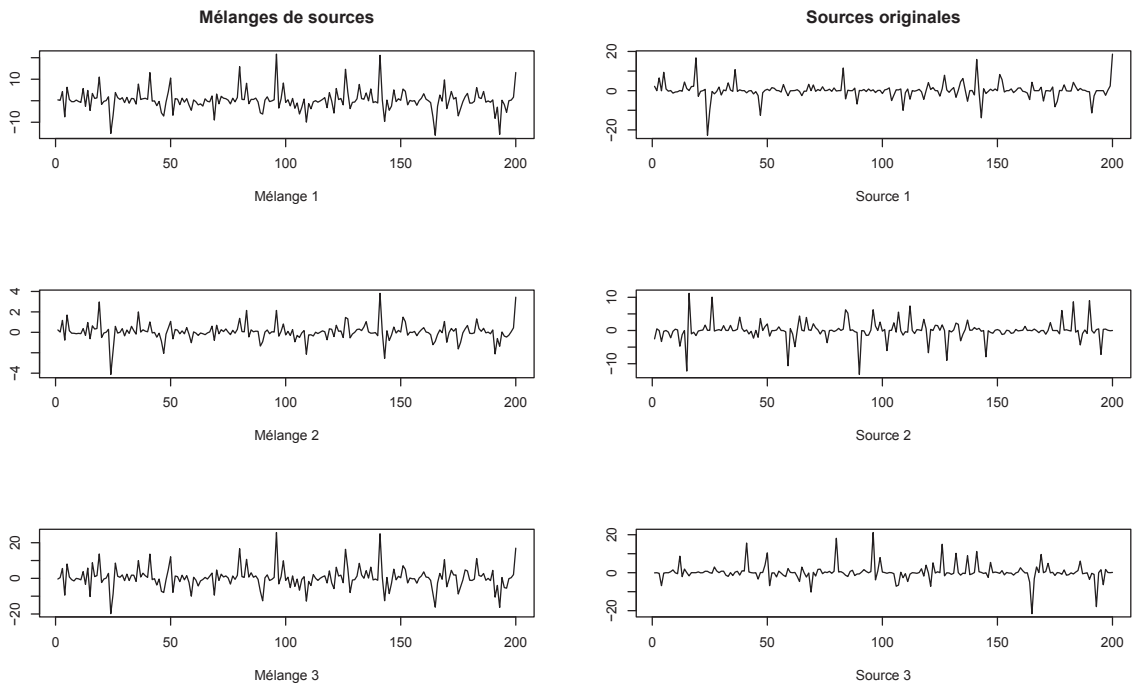

FIGURE 1.1. Réalisations des sources originales et leurs mélanges observés.

$\forall i = 1, \ldots, n$. Les a_{ij} sont des coefficients non aléatoires mais inconnus puisqu'on ne connait pas le modèle physique de mélange des sources. Les sources s_{ij} sont aussi inconnues puisqu'on ne les mesure pas directement. Ceci est le problème de la *séparation aveugle de sources*. On utilise le terme aveugle car on souhaite retrouver les pseudo-réalisations s_{ij} des sources uniquement à partir des mélanges x_{ij} de ces sources.

Les mélanges de sources observés ainsi que les réalisations des sources pourraient être, par exemple, ceux de la figure 1.1. On remarque que les mélanges de sources observés se ressemblent tous, ce qui est normal car ils sont tous fonction des mêmes sources. Il y a seulement les coefficients a_{ij} qui changent. À partir de ces mélanges de sources, le but est de retrouver les réalisations s_{ij} des sources de la figure 1.1.

1.2. Analyse en composantes indépendantes (ACI)

Dans le contexte de la séparation aveugle de sources, l'ACI permet d'extraire les sources à partir de mélanges de ces sources. Dans le modèle d'ACI, on considère les sources et les mélanges de sources comme étant des variables aléatoires. Le modèle s'écrit sous la forme suivante :

$$\mathbf{x} = \boldsymbol{\mathcal{A}}\mathbf{s}, \tag{1.2.1}$$

où $\mathbf{x} = (x_1, \ldots, x_p)^\intercal$ est un vecteur aléatoire observable $p \times 1$ à valeurs continues des p mélanges de sources, $\boldsymbol{\mathcal{A}} = (a_{lk})$ est une matrice de mélange inconnue non aléatoire de dimension $p \times p$ et $\mathbf{s} = (s_1, \ldots, s_p)^\intercal$ est un vecteur aléatoire non observable $p \times 1$, à valeurs continues, des p sources que l'on souhaite retrouver. À première vue, le problème semble impossible à résoudre puisque $\boldsymbol{\mathcal{A}}$ et $\mathbf{s}$ sont tous les deux inconnus. Afin de pouvoir identifier les sources s_j, l'ACI repose sur trois hypothèses fondamentales :

1. Les sources s_j sont supposées statistiquement indépendantes. C'est grâce à cette hypothèse que l'on obtient les estimations des sources, appelées composantes « indépendantes » et notées y_j.

2. Parmi les p sources, au plus une peut avoir une distribution gaussienne. Intuitivement, la distribution gaussienne est trop simple, car ses cumulants d'ordre supérieur à deux sont nuls. Cette hypothèse sera davantage expliquée à la section 1.8.

3. On suppose que la matrice de mélange $\boldsymbol{\mathcal{A}}$ est carrée et inversible et on note $\boldsymbol{\mathcal{W}}$ son inverse. On peut alors exprimer les sources en fonction des mélanges de sources selon le modèle d'ACI inverse tel que

$$\mathbf{s} = \boldsymbol{\mathcal{W}}\mathbf{x}. \tag{1.2.2}$$

On remarque que les sources sont une combinaison linéaire des mélanges de sources. Il suffit de trouver la bonne matrice de séparation $\boldsymbol{\mathcal{W}}$ qui va nous permettre de retrouver les sources.

1.3. Centrage des variables

Sans perte de généralité, on peut faire l'hypothèse que les mélanges de sources $\mathbf{x}$ de même que les sources $\mathbf{s}$ sont centrés en 0. Cette hypothèse simplifie grandement la théorie ainsi que les algorithmes. Cette hypothèse sera conservée tout au long de ce mémoire.

Si l'hypothèse n'est pas vérifiée, il est possible de centrer les variables comme méthode de pré-traitement. Le centrage des variables se fait en soustrayant l'espérance de chacune des variables. Ceci signifie que les mélanges de sources originaux, disons $\mathbf{x}'$, sont transformés par l'équation suivante

$$\mathbf{x} = \mathbf{x}' - \mathbb{E}\{\mathbf{x}'\} \tag{1.3.1}$$

avant d'effectuer l'analyse en composantes indépendantes. On remarque que par cette procédure, les sources sont aussi centrées en 0. Ceci se voit en calculant l'espérance de l'équation (1.2.2) :

$$\mathbb{E}\{\mathbf{s}\} = \boldsymbol{\mathcal{W}}\mathbb{E}\{\mathbf{x}\} = \boldsymbol{\mathcal{W}}\mathbb{E}\{\mathbf{x}' - \mathbb{E}\{\mathbf{x}'\}\} = 0. \tag{1.3.2}$$

La matrice de transformation, que ce soit $\boldsymbol{\mathcal{A}}$ ou $\boldsymbol{\mathcal{W}}$, demeure inchangée après cette méthode de pré-traitement ; on peut alors utiliser cette méthode sans affecter l'estimation du modèle. Après avoir estimé la matrice de séparation $\boldsymbol{\mathcal{W}}$ ainsi que les sources S_j, disons par $\hat{\boldsymbol{\mathcal{W}}}$ et Y_j respectivement, la moyenne qui a été soustraite peut être reconstruite en ajoutant $\hat{\boldsymbol{\mathcal{W}}}\mathbb{E}\{\mathbf{x}'\}$ aux composantes « indépendantes » Y_j.

En pratique, on possède un échantillon de données $\mathbf{x}'_1, \ldots, \mathbf{x}'_n$, provenant des vecteurs aléatoires indépendants et identiquement distribués $\mathbf{x}'_1, \ldots, \mathbf{x}'_n$, où chacun des vecteurs $\mathbf{x}'_i$ suit le modèle d'ACI (1.2.1). Les vecteurs de données $\mathbf{x}'_1, \ldots, \mathbf{x}'_n$ sont regroupés en lignes dans une matrice d'observations $\boldsymbol{\mathcal{X}}' : n \times p$. La matrice de données est d'abord centrée en soustrayant la moyenne échantillonale $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}'_i$ à chaque ligne de $\boldsymbol{\mathcal{X}}'$:

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}' - \mathbf{1}\bar{\mathbf{x}}^T, \tag{1.3.3}$$

où $\mathbf{1} : n \times 1$ est un vecteur contenant seulement des 1. Tout au long du mémoire, on suppose que la matrice de données $\boldsymbol{\mathcal{X}}$ est centrée. Ceci implique que chacune des composantes $\mathbf{x}_i, i = 1, \ldots, n$ sont aussi centrées.

1.4. Objectifs de l'ACI

En pratique, le modèle d'ACI s'écrit plutôt comme

$$\boldsymbol{\mathcal{X}}^T = \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{S}}^T, \tag{1.4.1}$$

où $\boldsymbol{\mathcal{S}}$ est une matrice $n \times p$ qui contient les réalisations des sources. Il y a deux objectifs principaux à l'ACI. Le premier objectif est d'estimer la densité p_{S_j} de chaque source S_j à partir de la matrice d'observations $\boldsymbol{\mathcal{X}}$ seulement. Le deuxième objectif est de retrouver les réalisations $\mathbf{s}_1, \ldots, \mathbf{s}_n$ comprises dans la matrice $\boldsymbol{\mathcal{S}}$, c'est-à-dire à les estimer à partir de la matrice d'observations $\boldsymbol{\mathcal{X}}$.

Chacun des objectifs passe d'abord par l'estimation de la matrice de séparation, notée $\hat{\boldsymbol{\mathcal{W}}}$. Une fois qu'on possède une estimation de $\boldsymbol{\mathcal{W}}$, l'estimation de la densité p_{S_j} de chaque source S_j se fait directement si on connait la densité $p_{\mathbf{X}}$ des mélanges de sources ou bien si cette dernière a été estimée à partir des données. De la même façon, l'estimation des $\mathbf{s}_i$ se fait directement en appliquant la formule

(1.2.2) du modèle d'ACI inverse. On dénote l'estimation de $\boldsymbol{\mathcal{S}}$ par la matrice $\boldsymbol{\mathcal{Y}}$ contenant les composantes « indépendantes » telle que

$$\boldsymbol{\mathcal{Y}}^{\intercal} = \hat{\boldsymbol{\mathcal{W}}}\boldsymbol{\mathcal{X}}^{\intercal}. \tag{1.4.2}$$

Chacune des méthodes d'analyse en composantes indépendantes se base sur des critères d'indépendance différents afin d'estimer la matrice $\boldsymbol{\mathcal{W}}$ pour ensuite obtenir des composantes $\mathbf{y}_1, \ldots, \mathbf{y}_p$, réalisations des variables aléatoires $Y_1, \ldots, Y_p$ les plus indépendantes possible au regard de ce critère.

1.5. Ambiguités de l'ACI

Dans le modèle d'ACI de l'équation (1.2.1), il est évident que les deux ambiguités suivantes sont présentes :

1.L'amplitude des sources ne peut être déterminée.
Puisque $\boldsymbol{\mathcal{A}}$ et $\mathbf{s}$ sont tous les deux inconnus, alors n'importe quel scalaire qui multiplie une des sources S_j peut être neutralisé en divisant par le même scalaire le vecteur colonne $\mathbf{a}_j$ correspondant de la matrice de mélange :

$$\mathbf{x} = \sum_{j=i}^{p} \mathbf{a}_j S_j = \sum_{j=i}^{p} \left(\frac{1}{\alpha_j}\mathbf{a}_j\right)(S_j\alpha_j). \tag{1.5.1}$$

Plusieurs auteurs fixent l'amplitude des sources en supposant que la variance de chacune des sources est unitaire : $\mathbb{E}(S_j^2) = 1$. Cependant, il reste toujours l'ambiguité du signe, c'est-à-dire qu'on peut multiplier une source par -1 et cela n'affecte pas le modèle. Heureusement, cette ambiguité est insignifiante dans la plupart des applications.

2. On ne peut pas déterminer l'ordre des composantes indépendantes.
Encore une fois, puisque $\boldsymbol{\mathcal{A}}$ et $\mathbf{s}$ sont inconnus, alors on peut considérer n'importe quelle composante « indépendante » comme étant la première source. Mathématiquement, on peut ajouter une matrice de permutation $\boldsymbol{\mathcal{P}}$ et son inverse dans le modèle (1.2.1) et on obtient $\mathbf{x} = \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{\mathcal{P}}\mathbf{s}$. Les éléments de $\boldsymbol{\mathcal{P}}\mathbf{s}$ sont les sources S_j disposées dans un nouvel ordre et la matrice $\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{P}}^{-1}$ devient une nouvelle matrice de mélange.

1.6. ACI et gaussianité

En analyse en composantes indépendantes, il est courant de maximiser des mesures de non-gaussianité afin de trouver des composantes « indépendantes ». Dans cette section, nous considérons seulement le cas où les sources sont identiquement distribuées.

Supposons que $\mathbf{x}$ suit le modèle d'ACI $\mathbf{x} = \boldsymbol{\mathcal{A}}\mathbf{s}$. L'estimation des sources se fait en trouvant la bonne combinaison linéaire des mélanges de sources puisqu'on peut inverser le modèle tel que $\mathbf{s} = \boldsymbol{\mathcal{A}}^{-1}\mathbf{x}$. Une composante indépendante est donnée par une certaine combinaison linéaire des mélanges de sources, qu'on peut écrire par $Y = \mathbf{b}^\mathsf{T}\mathbf{x} = \mathbf{b}^\mathsf{T}\boldsymbol{\mathcal{A}}\mathbf{s}$. La composante Y est alors une combinaison linéaire des sources S_j où les coefficients sont donnés par $\mathbf{q}^\mathsf{T} = \mathbf{b}^\mathsf{T}\boldsymbol{\mathcal{A}}$. On obtient

$$Y = \mathbf{b}^\mathsf{T}\mathbf{x} = \mathbf{q}^\mathsf{T}\mathbf{s} = \sum_{j=1}^{p} q_j S_j. \tag{1.6.1}$$

Si $\mathbf{b}^\mathsf{T}$ est une ligne de l'inverse de $\boldsymbol{\mathcal{A}}$, alors la combinaison linéaire $\mathbf{b}^\mathsf{T}\mathbf{x}$ est égale à l'une des sources S_j. Dans ce cas, le vecteur $\mathbf{q}$ correspondant a des zéros partout excepté un élément qui serait égal à 1.

La question est maintenant : comment déterminer $\mathbf{b}^\mathsf{T}$ de façon à ce qu'il soit égal à l'une des lignes de l'inverse de $\boldsymbol{\mathcal{A}}$? En pratique, on ne peut pas déterminer $\mathbf{b}$ exactement, mais on peut en obtenir une bonne approximation.

L'idée fondamentale repose sur une variante du théorème de la limite centrale proposée par Granger (1976). Le théorème est le suivant :

Théorème 1.6.1. *Soit* $S_1, \ldots, S_p$ *des variables aléatoires indépendantes et identiquement distribuées et soit* Y *une combinaison linéaire des* S_j *:*

$$Y = \sum_{j=1}^{p} q_j S_j. \tag{1.6.2}$$

Alors la distribution de Y *est plus près de la distribution gaussienne que chacune des distributions des variables* S_j*.*

Donc, à l'aide de ce théorème, on peut dire que la somme de deux variables aléatoires indépendantes et identiquement distribuées est plus gaussienne que les variables originales. Ceci implique que $Y = \mathbf{q}^\mathsf{T}\mathbf{s}$ est plus gaussienne que n'importe quelle source S_j et devient la moins gaussienne possible lorsqu'elle vaut une seule source telle que $Y = S_j$. Dans ce cas, tous les éléments de $\mathbf{q}$ sont évidemment 0 sauf le $j^{\text{ème}}$ qui est non nul.

En pratique, on ne connait pas la valeur de $\mathbf{q}$, mais cela importe peut, car $\mathbf{q}^\mathsf{T}\mathbf{s} = \mathbf{b}^\mathsf{T}\mathbf{x}$. On n'a qu'à faire varier $\mathbf{b}$ de façon à ce que $\mathbf{b}^\mathsf{T}\mathbf{x}$ maximise une certaine mesure de non-gaussianité.

Il est important de rappeler que cette démonstration est heuristique, dans le sens qu'elle s'applique seulement si l'hypothèse de départ est vraie, c'est-à-dire si les sources S_j sont identiquement distribuées. Il n'existe pas, à notre connaissance, de théorème équivalent pour des sources non identiquement distribuées. Un autre lien entre l'indépendance et la non-gaussianité sera donné à la section 4.3.1.

Pour plus d'informations à propos de ce sujet, on peut consulter Cardoso (2003). Dans cet article, l'auteur établi un lien entre l'indépendance, une certaine mesure de corrélation ainsi qu'une mesure de non-gaussianité et ce, peut importe si les sources sont identiquement distribuées ou non. Ce lien est établi à l'aide de l'équation suivante :

$$I(\mathbf{Y}) = C(\mathbf{Y}) - \sum_{j=1}^{p} G(Y_j) + \text{constante}, \tag{1.6.3}$$

où $I(\cdot)$ représente une mesure de dépendance, $C(\cdot)$ représente une mesure de corrélation et où $G(\cdot)$ est une mesure de non-gaussianité. La fonction $I(\mathbf{Y})$ est positive et vaut 0 si les composantes de $\mathbf{Y}$ sont indépendantes, la fonction $C(\mathbf{Y})$ est positive et vaut 0 si les composantes de $\mathbf{Y}$ sont non-corrélées et $G(Y_j)$ est une fonction positive qui vaut 0 si Y_j est gaussienne. Ainsi, minimiser la dépendance $I(\mathbf{Y})$ entre les composantes de $\mathbf{Y}$ est équivalent à optimiser un critère qui prend en compte également la corrélation des composantes de $\mathbf{Y}$ et l'opposé de la somme de leur non-gaussianité. En d'autres mots, les composantes de $\mathbf{Y}$ seront les plus indépendantes possibles si celles-ci sont les moins corrélées et les moins gaussiennes possible.

1.7. Indépendance et corrélation

En analyse en composantes indépendantes, l'hypothèse fondamentale est celle de l'indépendance des sources. Plusieurs mesures de dépendance peuvent être utilisées afin de retrouver des composantes les moins dépendantes possible. Dans cette section, quelques notions de dépendance seront couvertes.

Afin de définir le concept d'indépendance, considérons deux variables aléatoires Y_1 et Y_2. Les variables Y_1 et Y_2 sont dites indépendantes si Y_1 n'apporte aucune information sur Y_2 et vice versa. L'indépendance peut être définie à l'aide de la densité des variables aléatoires. Soit $p_{(Y_1,Y_2)}$ la densité conjointe de Y_1 et Y_2 et soit p_{Y_1} et p_{Y_2} leur densités marginales respectives obtenues par

$$p_{Y_1}(y_1) = \int p_{(Y_1,Y_2)}(y_1, y_2)\,dy_2 \quad \text{et} \quad p_{Y_2}(y_2) = \int p_{(Y_1,Y_2)}(y_1, y_2)\,dy_1. \tag{1.7.1}$$

Les variables aléatoires Y_1 et Y_2 sont indépendantes si et seulement si la densité conjointe peut être factorisée comme suit :

$$p_{(Y_1,Y_2)}(y_1, y_2) = p_{Y_1}(y_1)p_{Y_2}(y_2). \tag{1.7.2}$$

La définition d'indépendance peut être généralisée pour p variables aléatoires, auquel cas la densité conjointe doit être le produit des p densités marginales.

Une des propriétés importantes de l'indépendance peut être dérivée à partir de la définition mentionnée ci-haut. Les variables aléatoires Y_1 et Y_2 sont indépendantes si et seulement si

$$\mathbb{E}\{f_1(Y_1)f_2(Y_2)\} = \mathbb{E}\{f_1(Y_1)\}\mathbb{E}\{f_2(Y_2)\} \tag{1.7.3}$$

pour toutes fonctions Borel f_1 et f_2. Dans Coleman (2000), l'auteur montre que la définition de l'indépendance (1.7.3) est aussi vérifiée pour certaines classes de fonctions Borel, comme la classe des polynômes en Y_1 donnée par $\{1, Y_1, Y_1^2, \ldots\}$ lorsque Y_1 est bornée.

Une notion plus faible que l'indépendance est la non-corrélation. Deux variables aléatoires Y_1 et Y_2 sont dites non-corrélées si leur covariance est nulle :

$$\mathbb{C}\text{ov}\{Y_1, Y_2\} = \mathbb{E}\{Y_1Y_2\} - \mathbb{E}\{Y_1\}\mathbb{E}\{Y_2\} = 0, \tag{1.7.4}$$

où l'espérance $\mathbb{E}\{Y_1Y_2\}$ est le moment d'ordre 2 de la densité conjointe et les espérances $\mathbb{E}\{Y_1\}$ et $\mathbb{E}\{Y_2\}$ sont les premiers moments des densités marginales de Y_1 et Y_2 respectivement.

Il est important de remarquer que deux variables indépendantes sont forcément non-corrélées. Ceci se voit directement à partir de l'équation (1.7.3) en prenant $f_1(Y_1) = Y_1$ et $f_2(Y_2) = Y_2$. Par contre, deux variables aléatoires Y_1 et Y_2 non-corrélées ne sont pas forcément indépendantes. Par exemple, considérons la paire (Y_1, Y_2) de variables aléatoires discrètes et supposons que leur fonction de masse conjointe assigne la probabilité $1/4$ à chacune des valeurs suivantes : $(0, 1), (0, -1), (1, 0), (-1, 0)$. Alors, il est facile de voir que Y_1 et Y_2 sont non-corrélées. Cependant, on a aussi que

$$\mathbb{E}\{Y_1^2Y_2^2\} = 0 \neq \frac{1}{4} = \mathbb{E}\{Y_1^2\}\mathbb{E}\{Y_2^2\}. \tag{1.7.5}$$

La condition d'indépendance de l'équation (1.7.3) pour $f_1(Y_1) = Y_1^2$ et $f_2(Y_2) = Y_2^2$ ne tient pas ; les variables Y_1 et Y_2 ne sont donc pas indépendantes.

La corrélation peut aussi être vue dans un contexte vectoriel. Considérons un vecteur aléatoire centré $\mathbf{x} = (X_1, \ldots, X_p)^\top$ de dimension p. La matrice de covariance de $\mathbf{x}$ est donnée par

$$\boldsymbol{\mathcal{C}}_{\mathbf{X}} = \mathbb{E}\{\mathbf{x}\mathbf{x}^\top\}, \tag{1.7.6}$$

où l'élément (i, j) de la matrice représente la covariance entre les variables X_i et X_j. La matrice de covariance $\boldsymbol{\mathcal{C}}_{\mathbf{X}}$ est symétrique ($\boldsymbol{\mathcal{C}}_{\mathbf{X}} = \boldsymbol{\mathcal{C}}_{\mathbf{X}}^\top$) et semi-définie positive,

c'est-à-dire que $\mathbf{a}^T \mathcal{C}_\mathbf{X} \mathbf{a} \geq 0, \ \forall \mathbf{a} \in \mathbb{R}^p$. De plus, les valeurs propres de la matrice de covariance $\mathcal{C}_\mathbf{X}$ sont réelles et positives ou nulles.

Les composantes de $\mathbf{x}$ sont non-corrélées entre elles si et seulement si la matrice de covariance est de la forme suivante

$$\mathcal{C}_\mathbf{X} = \text{diag}(\sigma^2_{X_1}, \ldots, \sigma^2_{X_p}). \tag{1.7.7}$$

On dit qu'un vecteur est blanc si sa matrice de covariance est égale à la matrice identité $\mathcal{I} : p \times p$:

$$\mathcal{C}_\mathbf{X} = \mathcal{I}. \tag{1.7.8}$$

Encore une fois, les composantes d'un vecteur blanc sont non-corrélées, mais pas nécessairement indépendantes. La matrice de covariance de la transformation linéaire $\mathbf{z} = \mathcal{A}\mathbf{x}$ reste inchangée si la matrice $\mathcal{A}$ est orthogonale. En effet, on a

$$\mathbb{E}\left\{\mathbf{z}\mathbf{z}^T\right\} = \mathbb{E}\left\{\mathcal{A}\mathbf{x}\mathbf{x}^T\mathcal{A}^T\right\} = \mathcal{A}\mathbb{E}\left\{\mathbf{x}\mathbf{x}^T\right\}\mathcal{A}^T = \mathcal{A}\mathcal{A}^T = \mathcal{I}. \tag{1.7.9}$$

En pratique, on possède un échantillon de données $\mathbf{x}_1, \ldots, \mathbf{x}_n$ centré et disposé dans une matrice $\mathcal{X} : n \times p$. L'estimation de la matrice de covariance s'effectue en remplaçant l'espérance théorique par une moyenne échantillonale telle que

$$\hat{\mathcal{C}}_\mathbf{X} = \frac{1}{n}\mathcal{X}^T\mathcal{X}. \tag{1.7.10}$$

Dans le contexte d'analyse en composantes indépendantes, il est suffisant de trouver des composantes qui sont indépendantes par paires seulement. Ce qui suit démontre bien pourquoi l'indépendance par paires entraîne l'indépendance mutuelle des composantes indépendantes.

Les deux prochains lemmes sont utilisés dans la preuve du Théorème 1.7.1.

Lemme 1.7.1 (Lemme de Cramér (1937))**.** *Considérons* Y_1 *une variable aléatoire définie comme*

$$Y_1 = \sum_{i=1}^{n} a_j X_j, \tag{1.7.11}$$

et où les X_j *sont des variables aléatoires indépendantes. Si* Y_1 *est gaussienne, alors toutes les variables* X_j *pour lesquelles* $a_j \neq 0$ *sont gaussiennes.*

Lemme 1.7.2 (Lemme de Dugué (1951))**.** *La fonction* $\phi(u) = e^{P(u)}$*, où* $P(u)$ *est un polynôme de dégré* m*, peut être une fonction caractéristique si et seulement si* $m \leq 2$*.*

En utilisant les deux lemmes précédents, il est possible de démontrer les deux théorèmes suivants. Ces deux théorèmes sont dûs à Darmois.

Théorème 1.7.1 (Théorème de Darmois (1953))**.** *Soit deux variables aléatoires* Y_1 *et* Y_2 *définies comme suit*

$$Y_1 = \sum_{j=1}^{n} a_j X_j, \quad Y_2 = \sum_{j=1}^{n} b_j X_j \tag{1.7.12}$$

où $n \geq 2$ *est fixé et où les* X_j *sont des variables aléatoires indépendantes. Si* Y_1 *et* Y_2 *sont indépendantes, alors toutes les variables* X_j *pour lesquelles* $a_j b_j \neq 0$ *sont gaussiennes.*

Le théorème suivant est utile afin de démontrer le Théorème 1.7.3.

Théorème 1.7.2. *Soit* $\mathbf{x}$ *et* $\mathbf{z}$ *deux vecteurs aléatoires tels que* $\mathbf{z} = \boldsymbol{\mathcal{B}}\mathbf{x}$*,* $\boldsymbol{\mathcal{B}}$ *étant une matrice rectangulaire. Supposons en outre que les composantes de* $\mathbf{x}$ *sont indépendantes et que les composantes de* $\mathbf{z}$ *sont indépendantes par paires. Si* $\boldsymbol{\mathcal{B}}$ *a deux valeurs non-nulles dans la même colonne* j*, alors* X_j *est soit gaussien ou non aléatoire.*

Les théorèmes précédents sont utiles afin de démontrer le théorème principal de Comon (1994) qui montre que dans un contexte d'ACI, l'indépendance par paires est équivalente à l'indépendance mutuelle.

Théorème 1.7.3. *Soit* $\mathbf{s}$ *un vecteur aléatoire contenant des sources indépendantes parmi lesquelles au plus une est gaussienne. On suppose que la densité de chaque source n'est pas réduite à un point de masse. Soit* $\boldsymbol{\mathcal{W}} : p \times p$ *une matrice orthogonale et soit* $\mathbf{y}$ *définie par* $\mathbf{y} = \boldsymbol{\mathcal{W}}\mathbf{s}$*. Les propriétés suivantes sont équivalentes :*

(i) Les composantes Y_j *sont indépendantes par paires.*

(ii) Les composantes Y_j *sont mutuellement indépendantes.*

(iii) $\boldsymbol{\mathcal{W}} = \boldsymbol{\Sigma}\boldsymbol{\mathcal{P}}$ *avec* $\boldsymbol{\Sigma}$ *une matrice diagonale et* $\boldsymbol{\mathcal{P}}$ *une matrice de permutation.*

Il est important de noter que dans le Théorème 1.7.3, la matrice de séparation $\boldsymbol{\mathcal{W}}$ est supposée orthogonale. Ceci est nécessairement le cas si la matrice de données $\boldsymbol{\mathcal{X}}$ est blanchie préalablement. Le blanchiment des données sera vue à la section 3.2.

1.8. Pourquoi les sources gaussiennes sont-elles interdites ?

Comme il a été mentionné précédemment, parmi les p sources, au plus une peut avoir une distribution gaussienne. À titre de contre-exemple, supposons que $\mathbf{s} = (S_1, S_2)^T$ contient deux sources gaussiennes indépendantes suivant le modèle d'ACI $\mathbf{x} = \boldsymbol{\mathcal{A}}\mathbf{s}$.

La réponse à la question peut être amenée à l'aide du blanchiment des données qui sera vu à la section 3.2. Le blanchiment des données transforme linéairement

le vecteur de données $\mathbf{x}$ en un autre vecteur $\mathbf{z}$ de façon à ce que les composantes de ce dernier soient non-corrélées entre elles. Le modèle d'ACI devient le suivant :

$$\mathbf{z} = \widetilde{\mathcal{A}}\mathbf{s}. \tag{1.8.1}$$

Nous allons voir que le blanchiment des données fait en sorte que la solution de l'ACI est définie à une matrice orthogonale $\widetilde{\mathcal{A}}$ près. Selon le modèle d'ACI de l'équation (1.8.1), $\mathbf{z}$ est une combinaison linéaire des sources gaussiennes et indépendantes s_1 et s_2. Donc, ceci implique que la densité conjointe de z_1 et z_2 ainsi que les densités marginales de z_1 et de z_2 sont gaussiennes. Dans ce cas, la densité conjointe sera égale au produit des densités marginales si et seulement si les variables gaussiennes z_1 et z_2 sont non-corrélées. Or, puisque les composantes de $\mathbf{z}$ sont non-corrélées, alors elles sont aussi indépendantes. N'importe quelle matrice orthogonale $\widetilde{\mathcal{W}} = \widetilde{\mathcal{A}}^{-1}$ donnera des composantes $\mathbf{y} = \widetilde{\mathcal{W}}\mathbf{z}$ indépendantes puisque les composantes de $\mathbf{y}$ demeurent non-corrélées et gaussiennes. Il est donc impossible de déterminer la matrice de séparation $\widetilde{\mathcal{W}}$ en se basant uniquement sur l'indépendance dans le cas où le vecteur de sources $\mathbf{s}$ contient au moins deux sources gaussiennes.

On peut aussi le voir en termes de la densité des sources. Puisque les sources sont indépendantes, alors la densité conjointe est égale au produit des densités marginales :

$$p_{(S_1,S_2)}(s_1, s_2) = \frac{1}{2\pi} \exp\left(-\frac{s_1^2 + s_2^2}{2}\right) = \frac{1}{2\pi} \exp\left(-\frac{\|\mathbf{s}\|^2}{2}\right). \tag{1.8.2}$$

On peut exprimer la densité des mélanges de sources en faisant le changement de variable $\mathbf{s} = \widetilde{\mathcal{A}}^{-1}\mathbf{z} = \widetilde{\mathcal{W}}\mathbf{z}$. On obtient

$$p_{(Z_1,Z_2)}(z_1, z_2) = \frac{1}{2\pi} \exp\left(-\frac{\|\widetilde{\mathcal{W}}z\|^2}{2}\right) |\det \widetilde{\mathcal{W}}|. \tag{1.8.3}$$

Puisque $\widetilde{\mathcal{A}}$ est orthogonale, alors la matrice $\widetilde{\mathcal{W}} = \widetilde{\mathcal{A}}^{-1}$ est aussi orthogonale. Ceci implique que $\widetilde{\mathcal{W}}^T\widetilde{\mathcal{W}} = \mathcal{I}$ et $|\det \widetilde{\mathcal{W}}| = 1$. On obtient

$$p_{(Z_1,Z_2)}(z_1, z_2) = \frac{1}{2\pi} \exp\left(-\frac{\|z\|^2}{2}\right). \tag{1.8.4}$$

On remarque que la matrice de mélange $\widetilde{\mathcal{A}}$ ne change pas la distribution des mélanges de sources, puisqu'elle n'apparait pas dans la densité. La distribution des sources et celle des mélanges de sources sont identiques. Il est donc impossible de déduire la matrice de mélange à partir des mélanges de sources seulement dans le cas où il y a au moins deux sources gaussiennes.

Chapitre 2

THÉORIE DE L'INFORMATION

2.1. Entropie d'une variable aléatoire discrète

L'entropie de Shannon est le concept de base de la théorie de l'information. On définit l'entropie $H(X)$ d'une variable aléatoire discrète X comme suit :

$$H(X) = -\sum_i \mathbf{P}\left(X = a_i\right) \log \mathbf{P}\left(X = a_i\right), \tag{2.1.1}$$

où les a_i sont les valeurs possibles que X peut prendre. L'entropie d'une variable aléatoire représente le degré d'information donné par les observations de cette variable aléatoire X. L'entropie peut prendre plusieurs unités de mesure différentes selon la base du logarithme. Le logarithme à base 2 est habituellement utilisé et dans ce cas, l'unité de mesure est appelé le bit. Dans le reste du mémoire, la base du logarithme n'a pas d'importance puisque ça ne fait que changer l'échelle. On peut définir la fonction f telle que

$$f(p) = -p \log p, \text{ avec } 0 \leq p \leq 1. \tag{2.1.2}$$

La fonction est représentée à la figure 2.1. Cette fonction est positive et vaut 0 lorsque $p = 0$ ou $p = 1$.

On peut réécrire l'entropie à l'aide de cette fonction :

$$H(X) = \sum_i f(\mathbf{P}(X = a_i)). \tag{2.1.3}$$

On remarque que l'entropie d'une variable aléatoire est petite si les probabilités $\mathbf{P}(X = a_i)$ sont près de 0 ou de 1 et qu'elle est grande si les probabilités se situent

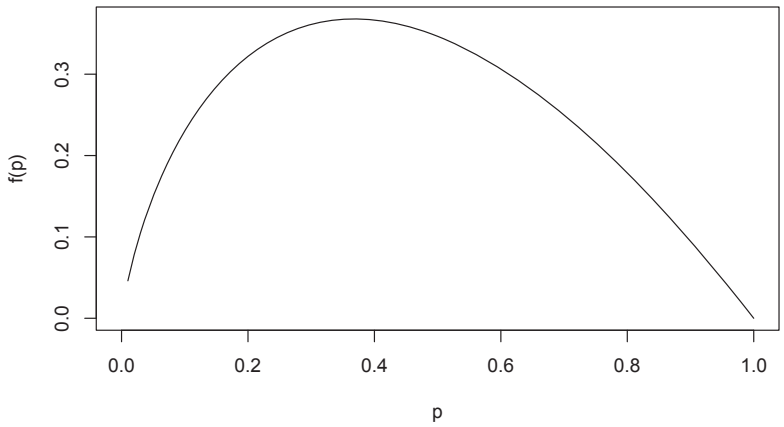

FIGURE 2.1. Graphique de la fonction $f(p) = -p\log(p)$.

entre les deux. Plus la variable aléatoire est imprévisible et non structurée, plus l'entropie sera grande. En effet, si toutes les probabilités sont près de 0 exceptée une qui est près de 1, alors la variable aléatoire est prévisible puisqu'elle prend souvent la même valeur. Cela se reflétera par une petite entropie.

2.2. Entropie d'une variable aléatoire continue

On peut généraliser l'entropie d'une variable aléatoire disrète à l'entropie d'une variable aléatoire continue X ayant comme densité $p_X(.)$ par

$$H(X) = -\int p_X(x)\log p_X(x)dx = \int f(p_X(x))dx. \tag{2.2.1}$$

L'entropie d'une variable continue mesure le caractère aléatoire de cette variable. Si la variable est concentrée dans un petit intervalle alors l'entropie sera petite. Contrairement à l'entropie d'une variable discrète, l'entropie d'une variable continue peut être négative. Ceci s'explique par le fait que la densité $p_X(.)$ peut prendre des valeurs qui sont supérieures à 1. Dans de tels cas, la fonction f est négative ce qui peut entraîner une entropie négative.

L'entropie d'une variable aléatoire continue peut être généralisée au cas multidimensionnel. Soit $\mathbf{x}$ un vecteur aléatoire ayant comme densité $p_{\mathbf{X}}(.)$. L'entropie est maintenant définie comme

$$H(\mathbf{x}) = -\int p_{\mathbf{X}}(\mathbf{x})\log p_{\mathbf{X}}(\mathbf{x})d\mathbf{x} = \int f(p_{\mathbf{X}}(\mathbf{x}))d\mathbf{x}. \tag{2.2.2}$$

2.3. Entropie d'une transformation

Considérons une transformation inversible du vecteur aléatoire $\mathbf{x}$ telle que

$$\mathbf{y} = \mathbf{g}(\mathbf{x}). \tag{2.3.1}$$

Dans cette section, nous allons montrer la relation entre l'entropie de $\mathbf{y}$ et celle de $\mathbf{x}$. Dans le cas général, on obtient la relation suivante

$$H(\mathbf{y}) = H(\mathbf{x}) + \mathbb{E}\{\log|\det \mathbf{J}\mathbf{g}(\mathbf{x})|\}, \tag{2.3.2}$$

où $\mathbf{J}\mathbf{g}(\mathbf{x})$ représente la matrice jacobienne de la fonction $\mathbf{g}$. En d'autre mots, l'entropie de $\mathbf{x}$ est augmentée de $\mathbb{E}(\log|\det \mathbf{J}\mathbf{g}(\mathbf{x})|)$ lors de la transformation.

Un cas important dans l'analyse en composantes indépendantes est la transformation linéaire

$$\mathbf{y} = \boldsymbol{\mathcal{M}}\mathbf{x} \tag{2.3.3}$$

pour lequel on obtient

$$H(\mathbf{y}) = H(\mathbf{x}) + \log|\det \boldsymbol{\mathcal{M}}|. \tag{2.3.4}$$

Si l'on considère la transformation linéaire univariée, on obtient

$$H(\alpha x) = H(x) + \log|\alpha|. \tag{2.3.5}$$

On remarque donc que l'entropie varie lorsque l'échelle de la variable x varie aussi. C'est pour cette raison que l'échelle de x est souvent fixée avant de mesurer son entropie.

2.4. Information mutuelle

L'information mutuelle est une mesure d'information que les membres d'un ensemble de variables aléatoires ont sur les autres variables aléatoires de l'ensemble. L'information mutuelle de p variables aléatoires x_j, $j = 1, \ldots, p$ est

$$I(x_1, x_2, \ldots, x_p) = \sum_{j=1}^{p} H(x_j) - H(\mathbf{x}), \tag{2.4.1}$$

où $\mathbf{x} = (x_1, x_2, \ldots, x_p)^T$.

L'information mutuelle peut être exprimée comme une mesure de distance en utilisant la divergence de Kullback-Leibler. Celle-ci mesure la divergence entre deux densités à p dimensions $p_1(\cdot)$ et $p_2(\cdot)$ et est donnée par

$$\delta(p_1, p_2) = \int p_1(\mathbf{x}) \log \frac{p_1(\mathbf{x})}{p_2(\mathbf{x})} d\mathbf{x}. \tag{2.4.2}$$

Cette mesure est toujours positive et est nulle si et seulement si les deux densités sont égales. Appliquons maintenant la divergence de Kullback-Leibler dans notre cas. Si les variables aléatoires X_j sont indépendantes, alors leur densité conjointe est égale au produit des densités marginales. On peut donc mesurer l'indépendance des X_j en mesurant la divergence de Kullback-Leibler entre les densités $p_1(\cdot) = p_{\mathbf{X}}(\cdot)$ et $p_2(\cdot) = p_{X_1}(\cdot) \dots p_{X_p}(\cdot)$. On obtient

$$\begin{aligned}
\delta(p_{\mathbf{X}}, p_{X_1} \dots p_{X_p}) &= \int p_{\mathbf{X}}(\boldsymbol{x}) \log \frac{p_{\mathbf{X}}(\boldsymbol{x})}{p_{X_1}(x_1) \dots p_{X_p}(x_p)} d\boldsymbol{x} \\
&= \int p_{\mathbf{X}}(\boldsymbol{x}) \log p_{\mathbf{X}}(\boldsymbol{x}) d\boldsymbol{x} - \sum_{i=1}^{p} \int p_{\mathbf{X}}(\boldsymbol{x}) \log p_{X_i}(x_i) d\boldsymbol{x} \\
&= -H(\mathbf{X}) - \sum_{j=1}^{p} \int p_{X_j}(x_j) \log p_{X_j}(x_j) dx_j. \qquad (2.4.3)
\end{aligned}$$

Ceci correspond à la définition de l'information mutuelle. On en déduit que l'information mutuelle est toujours positive et est nulle si et seulement si les variables aléatoires X_j sont indépendantes. L'information mutuelle est donc une mesure de dépendance ; c'est pour cela qu'elle est beaucoup utilisée en analyse en composantes indépendantes.

2.5. Distributions du maximum d'entropie

Dans cette section, on s'intéresse aux distributions qui maximisent l'entropie sous certaines contraintes. Ces contraintes sont habituellement des contraintes sur les moments de la variable aléatoire X :

$$\int p_X(x) F_i(x) dx = \mathbb{E}(F_i(X)) = c_i, \ i = 1, \dots, m, \qquad (2.5.1)$$

où les c_i sont des constantes fixées. Par exemple, si $F_1(X) = X$, alors cela veut dire qu'on cherche la distribution qui maximise l'entropie sous la contrainte que l'espérance de X soit fixée, c'est-à-dire que $\mathbb{E}\{X\} = c_1$.

La question qu'on se pose est maintenant : quelle est la densité p_0 qui maximise l'entropie parmi les densités qui satisfont aux contraintes de l'équation (2.5.1) ? La densité du maximum d'entropie peut être vue comme étant la densité qui est compatible avec les données et qui fait le moins d'hypothèses sur celles-ci. Ceci est vrai, car l'entropie peut être vue comme une mesure du caractère aléatoire de la densité. Selon Cover et coll. (1991), la densité p_0 qui satisfait les contraintes

(2.5.1) et qui a le maximum d'entropie est de la forme

$$p_0(x) = A \exp\left(\sum_{i=1}^{m} a_i F_i(x)\right). \tag{2.5.2}$$

Les constantes A et a_i sont fonction des contraintes (2.5.1) et de la contrainte $\int p_0(x)dx = 1$. Pour les trouver, il suffit de résoudre un système de $m+1$ équations non linéaires qui peut être difficile à résoudre. La solution est généralement trouvée numériquement.

Considérons maintenant les variables qui ont comme domaine les réels $\mathbb{R}$, une espérance nulle $\mathbb{E}\{x\} = 0$ ainsi qu'une variance fixée $\mathbb{V}\text{ar}\{x\} = 1$. La distribution du maximum d'entropie qui satisfait ces contraintes est la suivante :

$$p_0(x) = A \exp(a_1 x^2 + a_2 x). \tag{2.5.3}$$

Cette distribution correspond à la distribution normale. Nous avons donc le résultat qu'une variable gaussienne possède la plus grande entropie parmi toutes les variables aléatoires de variance fixée. En d'autres mots, la distribution gaussienne est la distribution la plus « aléatoire » parmi toutes ces distributions. Ce résultat est aussi valide dans le cas multidimensionnel, c'est-à-dire que la distribution gaussienne a le maximum d'entropie parmi toutes les distributions avec une matrice de covariance fixée.

2.6. Néguentropie

Les résultats de la section 2.5 nous montrent qu'on peut utiliser l'entropie pour bâtir une mesure de non normalité. Cette mesure est appelée la néguentropie. Elle vaut 0 pour une variable gaussienne et est toujours positive. Elle est définie comme suit

$$J(\mathbf{x}) = H(\mathbf{x}_{\text{gauss}}) - H(\mathbf{x}), \tag{2.6.1}$$

où $\mathbf{x}_{\text{gauss}}$ est une variable gaussienne ayant la même matrice de covariance que $\mathbf{x}$. Son entropie peut être évaluée par

$$H(\mathbf{x}_{\text{gauss}}) = \frac{1}{2} \log|\det \boldsymbol{\Sigma}| + \frac{n}{2}\left[1 + \log(2\pi)\right]. \tag{2.6.2}$$

2.7. Approximation de l'entropie

Il est difficile de travailler avec l'entropie puisqu'on doit calculer l'intégrale de la définition (2.2.1). En pratique, on préfère travailler avec des approximations de l'entropie qui réduisent considérablement le temps de calcul. Dans cette section, nous allons approximer l'entropie à l'aide de l'expansion de Gram-Charlier. On peut montrer que l'approximation de l'entropie est la suivante

$$H(x) \approx -\int \phi(x) \log \phi(x) dx - \frac{\kappa_3(x)^2}{2 \times 3!} - \frac{\kappa_4(x)^2}{2 \times 4!}, \tag{2.7.1}$$

où $\phi(.)$ représente la densité gaussienne. Le cumulant d'ordre 3 est $\kappa_3(x)$ et vaut $\mathbb{E}\{x^3\}$ pour une variable aléatoire x centrée. De la même façon, le cumulant d'ordre 4 est $\kappa_4(x)$ et vaut $\mathbb{E}\{x^4\} - 3\mathbb{E}\{x^2\}^2$ pour une variable aléatoire centrée.

Une approximation de la néguentropie est obtenue directement et est

$$J(x) \approx \frac{\kappa_3(x)^2}{12} + \frac{\kappa_4(x)^2}{48}. \tag{2.7.2}$$

En pratique, cette approximation de la néguentropie n'est pas robuste aux valeurs aberrantes. Or, cette approximation peut dépendre de quelques valeurs éloignées seulement. De plus, ces observations peuvent être erronées ce qui résulte en une estimation de la néguentropie erronée. Il est possible d'obtenir une approximation plus robuste en approximant la néguentropie par des fonctions non quadratiques telles que mentionnées dans Hyvärinen et coll. (2001). L'approximation générale est donnée par

$$J(x) \approx \left(\mathbb{E}\{g(x)\} - \mathbb{E}\{g(\nu)\}\right)^2, \tag{2.7.3}$$

où g est une fonction non quadratique et où $\nu \sim N(0, 1)$. Dans cette approximation, on suppose que x est symétrique et que sa variance est unitaire. On choisit habituellent une fonction g parmi les suivantes :

$$g_1(x) = \log\cosh(x) \quad \text{ou bien} \quad g_2(x) = -\exp\left(\frac{-x^2}{2}\right). \tag{2.7.4}$$

Ces fonctions sont robustes aux valeurs aberrantes. Par exemple, si on choisit la fonction g_2 alors l'approximation de la néguentropie devient

$$J(x) \approx \left(\mathbb{E}\left\{\exp\left(\frac{-x^2}{2}\right)\right\} - \frac{1}{\sqrt{2}}\right)^2. \tag{2.7.5}$$

Chapitre 3

ANALYSE EN COMPOSANTES PRINCIPALES ET BLANCHIMENT

3.1. Analyse en composantes principales

L'analyse en composantes principales (ACP) est une méthode statistique d'analyse de données souvent utilisée en extraction d'images ainsi qu'en compression de données. À partir d'un ensemble de données multivariées provenant de p variables, le but de l'ACP est de trouver un ensemble de $m < p$ variables contenant moins de redondance tout en conservant un maximum d'information. En d'autres mots, le but de l'ACP est de trouver un ensemble de m variables non-corrélées maximisant la variance. Le but de l'ACP est semblable à celui de l'ACI dans le sens qu'en ACP, la redondance est mesurée à l'aide de la corrélation alors qu'en ACI, elle est mesurée à l'aide d'une notion beaucoup plus large, soit l'indépendance. De plus, contrairement à l'ACP, peu d'importance est donnée à la réduction de l'espace en ACI.

L'ACP est souvent utilisée comme méthode de pré-traitement en ACI, car elle fabrique des composantes non-corrélées. Or, la non-corrélation est une étape nécessaire à l'indépendance.

En pratique, il est essentiel que les éléments de l'échantillon soient mutuellement corrélés afin de rendre possible la compression. Si les éléments sont indépendants, alors la transformation par ACP ne modifiera pas les données.

3.1.1. ACP par maximisation de la variance

Considérons d'abord la combinaison linéaire suivante :

$$Y_1 = \boldsymbol{w}_1^T \mathbf{x} = \sum_{j=1}^{p} w_{j1} X_j. \tag{3.1.1}$$

On cherche à maximiser la variance de Y_1 en fonction de $\boldsymbol{w}_1$. Si la variance de Y_1 est maximale, alors on dit que Y_1 est la première composante principale de $\mathbf{x}$. On remarque que la variance dépend de la norme du vecteur $\boldsymbol{w}_1$ et qu'elle peut augmenter autant que la norme augmente. C'est pour cette raison qu'en pratique, on fixe la norme de $\boldsymbol{w}_1$ à 1. On rappelle que les variables Y et $\mathbf{x}$ sont centrées. On cherche donc le vecteur $\boldsymbol{w}_1$ qui maximise la variance

$$\mathbb{V}\mathrm{ar}\{Y_1\} = \mathbb{E}\{Y_1^2\} = \mathbb{E}\{(\boldsymbol{w}_1^T\mathbf{x})^2\} = \boldsymbol{w}_1^T\mathbb{E}\{\mathbf{x}\mathbf{x}^T\}\boldsymbol{w}_1 = \boldsymbol{w}_1^T\boldsymbol{\mathcal{C}}_{\mathbf{X}}\boldsymbol{w}_1 \text{ tel que } \|\boldsymbol{w}_1\|_2 = 1, \tag{3.1.2}$$

où la norme $\|\boldsymbol{w}_1\|_r$ est définie par

$$\|\boldsymbol{w}_1\|_r = \left(\sum_{j=1}^{p} |w_{1j}|^r\right)^{1/r}. \tag{3.1.3}$$

Il est connu que la solution de l'ACP est donnée en termes des vecteurs propres unitaires $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_p$ de la matrice $\boldsymbol{\mathcal{C}}_{\mathbf{X}}$. Les vecteurs propres sont associés aux valeurs propres $d_1, \ldots, d_p$ telles que $d_1 \geq d_2 \geq \ldots \geq d_p$. La solution qui maximise (3.1.2) est

$$\boldsymbol{w}_1 = \boldsymbol{e}_1. \tag{3.1.4}$$

La première composante principale de $\mathbf{x}$ est donc $Y_1 = \boldsymbol{e}_1^T\mathbf{x}$.

Le critère de l'équation (3.1.2) peut être généralisé pour m composantes principales. La $m^{\text{ième}}$ composante principale de $\mathbf{x}$ sera donnée par Y_m si cette dernière maximise la variance sous la contrainte qu'elle soit non-corrélée avec les $m-1$ composantes principales trouvées précédemment :

$$\mathbb{E}\{Y_m Y_k\} = \mathbb{E}\{(\boldsymbol{w}_m^T\mathbf{x})(\boldsymbol{w}_k^T\mathbf{x})\} = \boldsymbol{w}_m^T\boldsymbol{\mathcal{C}}_{\mathbf{X}}\boldsymbol{w}_k = 0, \ \forall k < m. \tag{3.1.5}$$

Pour la deuxième composante principale, on obtient

$$\boldsymbol{w}_2^T\boldsymbol{\mathcal{C}}_{\mathbf{X}}\boldsymbol{w}_1 = \boldsymbol{w}_2^T\boldsymbol{\mathcal{C}}_{\mathbf{X}}\boldsymbol{e}_1 = \boldsymbol{w}_2^T\boldsymbol{e}_1 d_1 = 0. \tag{3.1.6}$$

On cherche donc $\boldsymbol{w}_2$ tel que la variance $\mathbb{E}\{Y_2^2\} = \mathbb{E}\{(\boldsymbol{w}_2^T\mathbf{x})^2\}$ soit maximale dans un espace orthogonal à $\boldsymbol{e}_1$. La solution est donnée par

$$\boldsymbol{w}_2 = \boldsymbol{e}_2. \tag{3.1.7}$$

De la même façon, on trouve

$$\boldsymbol{w}_k = \boldsymbol{e}_k. \tag{3.1.8}$$

La $k^{\text{ième}}$ composante principale est alors donnée par $Y_k = \boldsymbol{e}_k^T\mathbf{x}$.

3.1.2. Choix du nombre de composantes principales

Une application importante en ACP est la compression de données. Le jeu de données originales contient p variables. On souhaite réduire l'espace à $m < p$ variables non-corrélées tout en conservant un maximum d'information. Le vecteur original $\mathbf{x}$ peut être approximé par une expansion d'ACP tronquée telle que

$$\hat{\mathbf{x}} = \sum_{i=1}^{m} Y_i \boldsymbol{e}_i. \tag{3.1.9}$$

La question est maintenant : comment déterminer le nombre m de composantes principales ? On peut répondre à cette question en regardant l'erreur quadratique moyenne (EQM) :

$$\mathrm{EQM}\{\hat{\mathbf{x}}; \mathbf{x}\} = \mathbb{E}\left\{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2\right\} = \mathbb{E}\left\{\|\mathbf{x} - \sum_{j=1}^{m} (\boldsymbol{e}_j^T \mathbf{x})\boldsymbol{e}_j\|_2^2\right\}. \tag{3.1.10}$$

On peut écrire l'EQM sous la forme suivante dû à l'orthogonalité des vecteurs propres :

$$\begin{aligned}\mathrm{EQM}\{\hat{\mathbf{x}}; \mathbf{x}\} &= \mathbb{E}\{\|\mathbf{x}\|_2^2\} - \mathbb{E}\left\{\sum_{j=1}^{m} (\boldsymbol{e}_j^T \mathbf{x})^2\right\} \\ &= \mathrm{trace}\{\boldsymbol{\mathcal{C}}_{\mathbf{X}}\} - \sum_{j=1}^{m} \boldsymbol{e}_j^T \boldsymbol{\mathcal{C}}_{\mathbf{X}} \boldsymbol{e}_j. \end{aligned} \tag{3.1.11}$$

On peut finalement montrer que l'EQM est égale à

$$\mathrm{EQM}\{\hat{\mathbf{x}}; \mathbf{x}\} = \sum_{j=m+1}^{p} d_j. \tag{3.1.12}$$

Puisque les valeurs propres de la matrice de covariance $\boldsymbol{\mathcal{C}}_{\mathbf{X}}$ sont toutes positives, on remarque que l'EQM diminue lorsque le nombre m de composantes principales augmente et est nulle lorsque $m = p$. On doit faire un compromis entre l'erreur d'approximation et la compression des données. En pratique, la séquence de valeurs propres $d_1, \ldots, d_p$ de la matrice de covariance décroît rapidement. Il est possible d'établir un seuil à partir duquel les valeurs propres sont non significatives.

3.2. Blanchiment des données

Le problème d'analyse en composantes indépendantes est grandement simplifié lorsque le vecteur contenant les mélanges de sources $\mathbf{x}$ est blanchi en un autre vecteur, disons $\mathbf{z}$. Le blanchiment des données est une décorrélation des données

suivit d'un changement d'échelle. C'est pour cette raison que l'ACP peut être utilisé dans ce contexte et que le blanchiment peut être effectué par une combinaison linéaire. Considérons un vecteur aléatoire $\mathbf{x}$. Le problème du blanchiment des données est de trouver une transformation linéaire $\boldsymbol{\mathcal{V}}$ de façon à ce que la résultante

$$\mathbf{z} = \boldsymbol{\mathcal{V}}\mathbf{x} \tag{3.2.1}$$

soit blanche.

Le problème a une solution qui est directement liée aux termes de l'expansion de l'ACP. Soit $\boldsymbol{\mathcal{E}} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_p)$ la matrice contenant les vecteurs propres unitaires de la matrice $\boldsymbol{\mathcal{C}}_{\mathbf{X}}$. Soit $\boldsymbol{\mathcal{D}} = \mathrm{diag}(d_1, \ldots, d_p)$ la matrice diagonale contenant les valeurs propres. Une transformation linéaire qui effectue le blanchiment des données est fournie par

$$\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{D}}^{-1/2}\boldsymbol{\mathcal{E}}^{\mathsf{T}}. \tag{3.2.2}$$

On peut vérifier que cette transformation nous donne bien une résultante $\mathbf{z}$ blanche. La matrice de covariance peut être exprimée en fonction des vecteurs et des valeurs propres : $\boldsymbol{\mathcal{C}}_{\mathbf{X}} = \boldsymbol{\mathcal{E}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{E}}^{\mathsf{T}}$. On obtient

$$\mathbb{E}\{\mathbf{z}\mathbf{z}^{\mathsf{T}}\} = \boldsymbol{\mathcal{V}}\mathbb{E}\{\mathbf{x}\mathbf{x}^{\mathsf{T}}\}\boldsymbol{\mathcal{V}}^{\mathsf{T}} = \boldsymbol{\mathcal{D}}^{-1/2}\boldsymbol{\mathcal{E}}^{\mathsf{T}}\boldsymbol{\mathcal{E}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{E}}^{\mathsf{T}}\boldsymbol{\mathcal{E}}\boldsymbol{\mathcal{D}}^{-1/2} = \boldsymbol{\mathcal{I}}. \tag{3.2.3}$$

La matrice de covariance de $\mathbf{z}$ est unitaire ; $\mathbf{z}$ est donc blanc. Il est important de noter que la transformation $\boldsymbol{\mathcal{V}}$ n'est pas unique. En effet, on peut voir que n'importe quelle matrice $\boldsymbol{\mathcal{U}}\boldsymbol{\mathcal{V}}$, où $\boldsymbol{\mathcal{U}}$ est une matrice orthogonale, est aussi une transformation qui effectue le blanchiment des données. De la même façon, on obtient

$$\mathbb{E}\{\mathbf{z}\mathbf{z}^{\mathsf{T}}\} = \boldsymbol{\mathcal{U}}\boldsymbol{\mathcal{V}}\mathbb{E}\{\mathbf{x}\mathbf{x}^{\mathsf{T}}\}\boldsymbol{\mathcal{V}}^{\mathsf{T}}\boldsymbol{\mathcal{U}}^{\mathsf{T}} = \boldsymbol{\mathcal{I}}. \tag{3.2.4}$$

On remarque que peu importe la matrice orthogonale $\boldsymbol{\mathcal{U}}$, le vecteur $\mathbf{z}$ reste blanc. Or, le blanchiment des données signifie que les composantes de $\mathbf{z}$ sont non-corrélées entre elles. La non-corrélation est un pas vers l'indépendance.

Le nouveau modèle d'ACI pour des données blanchies est le suivant :

$$\mathbf{z} = \boldsymbol{\mathcal{V}}\boldsymbol{\mathcal{A}}\mathbf{s} \equiv \widetilde{\boldsymbol{\mathcal{A}}}\mathbf{s}. \tag{3.2.5}$$

Considérons la notation $\widetilde{\boldsymbol{\mathcal{W}}} = \widetilde{\boldsymbol{\mathcal{A}}}^{-1}$. En ACI, l'objectif principal est d'estimer le vecteur contenant les sources indépendantes $\mathbf{s}$ par le vecteur $\mathbf{y} = \widetilde{\boldsymbol{\mathcal{W}}}\mathbf{z}$ contenant des composantes les plus indépendantes possible. Ceci implique qu'on cherche des composantes les moins corrélées possible. On cherche aussi des composantes ayant une variance unitaire afin d'éviter l'ambiguité de l'amplitude des sources.

Ainsi, les composantes indépendantes recherchées sont blanches et la matrice de covariance est donnée par l'équation suivante :

$$\mathbb{E}\{\mathbf{Y}\mathbf{Y}^\mathsf{T}\} = \widetilde{\mathcal{W}}\mathbb{E}\{\mathbf{Z}\mathbf{Z}^\mathsf{T}\}\widetilde{\mathcal{W}}^\mathsf{T} = \widetilde{\mathcal{W}}\widetilde{\mathcal{W}}^\mathsf{T}. \tag{3.2.6}$$

On remarque que les composantes $\mathbf{Y}$ seront blanches si et seulement si $\widetilde{\mathcal{W}}\widetilde{\mathcal{W}}^\mathsf{T} = \mathcal{I}$. En d'autres mots, les composantes seront blanches si et seulement si la matrice de séparation $\widetilde{\mathcal{W}}$ est orthogonale.

Dans le contexte d'ACI, on dit souvent que le blanchiment des données résout le problème d'ACI à une matrice de mélange orthogonale $\widetilde{\mathcal{A}}$ près. Ceci est vrai, car

$$\widetilde{\mathcal{A}}\widetilde{\mathcal{A}}^\mathsf{T} = \widetilde{\mathcal{W}}^{-1}\left(\widetilde{\mathcal{W}}^{-1}\right)^\mathsf{T} = \widetilde{\mathcal{W}}^\mathsf{T}\widetilde{\mathcal{W}} = \mathcal{I}. \tag{3.2.7}$$

Le blanchiment des données est souvent utilisé comme étape de pré-traitement en ACI, car la nouvelle matrice de mélange $\widetilde{\mathcal{A}} = \mathcal{V}\mathcal{A}$ est orthogonale. Ceci veut dire qu'on peut restreindre l'espace de la matrice de mélange à l'espace des matrices orthogonales seulement. De plus, on peut montrer qu'une matrice orthogonale a seulement $p(p-1)/2$ degrés de liberté au lieu de p^2 pour une matrice générale.

En pratique, on possède un échantillon de données $\mathbf{x}_1, \ldots, \mathbf{x}_n$ disposé en lignes dans une matrice d'observations centrées $\mathcal{X} : n \times p$. Afin d'effectuer le blanchiment, on doit d'abord estimer la matrice de covariance $\mathcal{C}_\mathbf{X} = \mathbb{E}\{\mathbf{X}\mathbf{X}^\mathsf{T}\}$. L'estimation s'effectue en remplaçant l'espérance théorique par une moyenne échantillonale comme mentionné à l'équation (1.7.10) :

$$\hat{\mathcal{C}}_\mathbf{X} = \frac{1}{n}\mathcal{X}^\mathsf{T}\mathcal{X} = \widetilde{\mathcal{E}}\widetilde{\mathcal{D}}\widetilde{\mathcal{E}}^\mathsf{T}. \tag{3.2.8}$$

Le blanchiment des données peut aussi être effectué en considérant la décomposition en valeurs singulières (DVS) de la matrice de données $\mathcal{X}$ telle que

$$\mathcal{X} = \mathcal{U}\Sigma\mathcal{V}^\mathsf{T} \tag{3.2.9}$$

avec $\mathcal{U} : n \times n$ et $\mathcal{V} : p \times p$ des matrices orthogonales et $\Sigma : n \times p$ une matrice contenant les valeurs singulières σ_j sur sa diagonale. Dans le cas où $p < n$, la DVS peut aussi s'écrire sous la forme :

$$\mathcal{X} = \widetilde{\mathcal{U}}\widetilde{\Sigma}\mathcal{V}^\mathsf{T}, \tag{3.2.10}$$

où $\widetilde{\mathcal{U}} : n \times p$ est constituée des p premières colonnes de $\mathcal{U}$ et $\widetilde{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$ des p premières lignes de Σ. Dans ce cas, $\widetilde{\mathcal{U}}^\mathsf{T}\widetilde{\mathcal{U}} = \mathcal{I}_p$ tient toujours. On obtient alors la matrice de covariance suivante :

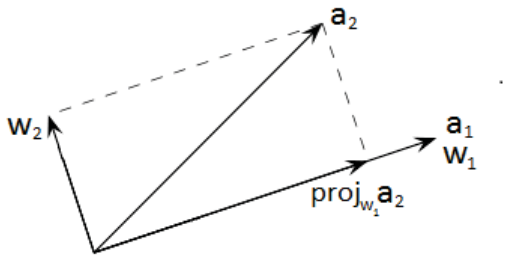

FIGURE 3.1. Les deux premières étapes de la méthode d'orthogonalisation de Gram-Schmidt dans le cas bidimensionnel.

$$\frac{1}{n}\mathcal{X}^T\mathcal{X} = \frac{1}{n}\mathcal{V}\widetilde{\Sigma}^T\widetilde{\mathcal{U}}^T\widetilde{\mathcal{U}}\widetilde{\Sigma}\mathcal{V}^T = \frac{1}{n}\mathcal{V}\widetilde{\Sigma}^2\mathcal{V}^T. \tag{3.2.11}$$

On peut faire le lien entre la décomposition en valeurs propres et la décomposition en valeurs singulières en comparant les matrices de covariance respectives. On remarque que $\widetilde{\mathcal{E}} = \mathcal{V}$ et que $\mathcal{D} = \frac{1}{n}\widetilde{\Sigma}^2$. La matrice $\mathcal{Z}$ obtenue par le blanchiment des données devient alors

$$\mathcal{Z}^T = \mathcal{D}^{-1/2}\widetilde{\mathcal{E}}^T\mathcal{X}^T = \mathcal{D}^{-1/2}\widetilde{\mathcal{E}}^T\mathcal{V}\widetilde{\Sigma}^T\widetilde{\mathcal{U}}^T = \sqrt{n}\widetilde{\mathcal{U}}^T. \tag{3.2.12}$$

3.3. ORTHOGONALISATION

L'orthogonalisation est souvent utile en analyse en composantes indépendantes. Tel qu'on vient de le voir à la section 3.2, le blanchiment des données résout le problème d'ACI à une matrice orthogonale près. Cependant, les algorithmes d'ACI ne produisent pas toujours des matrices orthogonales. C'est pourquoi on orthogonalise souvent les vecteurs entre chacune des itérations de l'algorithme.

Le problème est le suivant : étant donné un ensemble de vecteurs $\mathbf{a}_1, \ldots, \mathbf{a}_p$, le but est alors de trouver un autre ensemble de vecteurs $\mathbf{w}_1, \ldots, \mathbf{w}_p$ couvrant le même sous-espace mais étant orthogonaux ou orthonormaux. L'orthogonalisation est une transformation linéaire, c'est-à-dire que les $\mathbf{w}_i$ sont une combinaison linéaire des $\mathbf{a}_j$.

L'approche classique est la méthode d'orthogonalisation de Gram-Schmidt :

$$\begin{aligned} \mathbf{w}_1 &= \mathbf{a}_1 \\ \mathbf{w}_j &= \mathbf{a}_j - \sum_{i=1}^{j-1} \frac{\mathbf{w}_i^T\mathbf{a}_j}{\mathbf{w}_i^T\mathbf{w}_i}\mathbf{w}_i. \end{aligned} \tag{3.3.1}$$

La figure 3.1 montre les deux premières étapes de la méthode d'orthogonalisation de Gram-Schmidt dans le cas où les vecteurs sont dans $\mathbb{R}^2$. On peut vérifier que $\boldsymbol{w}_i^T \boldsymbol{w}_j = 0$ pour $i \neq j$. On obtient bel et bien des vecteurs orthogonaux. Si les vecteurs étaient ensuite divisés par leur norme, on obtiendrait alors des vecteur orthonormaux. Cette méthode d'orthogonalisation est une procédure d'orthogonalisation séquentielle, c'est-à-dire que l'orthogonalisation se fait dans un premier lieu par rapport à $\boldsymbol{w}_1$, puis ensuite par rapport à $\boldsymbol{w}_2$ et ainsi de suite. Un problème avec ce type d'orthogonalisation est l'accumulation des erreurs.

Il existe aussi des méthodes d'orthogonalisation symétrique. Dans ces méthodes, aucun vecteurs $\boldsymbol{a}_i$ n'est traité différemment des autres. On forme d'abord la matrice $\boldsymbol{\mathcal{A}} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p)$ dans laquelle on souhaite orthogonaliser les vecteurs colonnes. Ensuite, on applique la transformation suivante

$$\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{A}}^T \boldsymbol{\mathcal{A}})^{-1/2}, \tag{3.3.2}$$

où $(\boldsymbol{\mathcal{A}}^T \boldsymbol{\mathcal{A}})^{-1/2}$ est obtenu par la décomposition en valeurs propres de la matrice $\boldsymbol{\mathcal{A}}^T \boldsymbol{\mathcal{A}}$. La matrice résultante $\boldsymbol{\mathcal{W}}$ est bien orthogonale, car $\boldsymbol{\mathcal{W}}^T \boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{I}}$. L'orthogonalisation symétrique de $\boldsymbol{\mathcal{A}}$ n'est pas unique ; on peut multiplier la matrice $\boldsymbol{\mathcal{W}}$ par n'importe quelle matrice orthogonale $\boldsymbol{\mathcal{U}}$ et $\boldsymbol{\mathcal{U}}\boldsymbol{\mathcal{W}}$ reste orthogonale. Cette méthode d'orthogonalisation devrait être utilisée dans les algorithmes d'ACI permettant de trouver tous les vecteurs simultanément.

Il est important de noter que les deux méthodes d'orthogonalisation sont différentes ; elles ne mènent pas aux mêmes résultats.

Chapitre 4

ANALYSE EN COMPOSANTES INDÉPENDANTES

4.1. ACI par maximisation de la non normalité

Tel que vu à la section 1.6, la maximisation de la non normalité mène à des composantes indépendantes. Afin d'utiliser la non gaussianité pour résoudre le problème d'ACI, nous avons besoin de mesures quantitatives de non normalité pour une variable aléatoire. Une des mesures classiques de non normalité en ACI est le kurtosis. Ce dernier est le nom donné au cumulant d'ordre 4 d'une variable aléatoire.

Le kurtosis d'une variable aléatoire centrée Y est défini comme suit

$$\mathrm{kurt}(\mathrm{Y}) = \mathbb{E}\{\mathrm{Y}^4\} - 3(\mathbb{E}\{\mathrm{Y}^2\})^2. \tag{4.1.1}$$

Pour simplifier encore plus la forme du kurtosis, on peut supposer que Y a été normalisée de façon à ce que sa variance soit égale à 1 : $\mathbb{E}\{\mathrm{Y}^2\} = 1$. Le kurtosis s'écrit alors $\mathbb{E}\{\mathrm{Y}^4\}-3$. On remarque que le kurtosis est tout simplement une version normalisée du quatrième moment $\mathbb{E}\{\mathrm{Y}^4\}$. Le kurtosis vaut zéro pour une variable gaussienne et est non nul pour la grande majorité des variables aléatoires.

Il est important de noter que le kurtosis peut être positif ou négatif. Les variables aléatoires avec un kurtosis négatif sont appelées subgaussiennes et celles avec un kurtosis positif sont appelées supergaussiennes. Les variables aléatoires supergaussiennes ont généralement une densité pointue et des ailes relevées.

Une mesure de non gaussianité est donnée par la valeur absolue du kurtosis. Le carré du kurtosis peut aussi être utilisé. Cette nouvelle mesure est nulle pour une variable gaussienne alors qu'elle est positive pour la plupart des variables non gaussiennes. Le kurtosis, ou plutôt sa valeur absolue, a été largement utilisé comme mesure de non gaussianité en ACI du fait de sa simplicité.

Avant d'effectuer l'analyse en composantes indépendantes, les mélanges de sources $\mathbf{x}$ sont préalablement centrés et blanchis de façon à ce que $\mathcal{C}_{\mathbf{X}} = \mathcal{I}$. De plus, la matrice de séparation $\mathcal{W}$ cherchée est maintenant orthogonale.

L'estimation des sources par la maximisation du kurtosis se fait une composante $Y_k = \boldsymbol{w}_k^T \mathbf{x}$ à la fois. On trouve d'abord $\boldsymbol{w}_1$ qui maximise le kurtosis de Y_1, puis ensuite on trouve $\boldsymbol{w}_2$ qui maximise le kurtosis de Y_2 sous la contrainte d'orthogonalité avec $\boldsymbol{w}_1$ et ainsi de suite. Mathématiquement, on peut exprimer $\tilde{\boldsymbol{w}}_k$ comme étant celui qui maximise le kurtosis à la $k^{\text{ième}}$ étape :

$$\tilde{\boldsymbol{w}}_k = \arg\max_{\boldsymbol{w}} \left\{ \left| \left(\mathbb{E}\{\boldsymbol{w}^T \mathbf{x}\} \right)^4 - 3 \right| \right\} \tag{4.1.2}$$

sous les contraintes $\|\boldsymbol{w}\|^2 = 1$ et $\boldsymbol{w}^T \tilde{\boldsymbol{w}}_i = 0,\ \forall i = 1, \ldots, k-1$.

En pratique, on possède un échantillon de données $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ disposé dans une matrice $\boldsymbol{\mathcal{X}} : n \times p$. On estime le kurtosis à l'aide du quatrième moment échantillonal :

$$\widehat{\text{kurt}}(y) = \widehat{\text{kurt}}(\boldsymbol{w}^T \boldsymbol{\mathcal{X}}^T) = \left| \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{w}^T \mathbf{x}_i \right)^4 - 3 \right|. \tag{4.1.3}$$

L'ACI peut aussi être effectuée à l'aide des estimations de la néguentropie définies en (2.7.2) et (2.7.5). Dans le cas où la néguentropie est approximée par les cumulants, on obtient

$$\tilde{\boldsymbol{w}}_k = \arg\max_{\boldsymbol{w}} \left\{ \frac{\kappa_3 (\boldsymbol{w}^T \mathbf{x})^2}{12} + \frac{\kappa_4 (\boldsymbol{w}^T \mathbf{x})^2}{48} \right\}, \tag{4.1.4}$$

sous les contraintes $\|\boldsymbol{w}\|^2 = 1$ et $\boldsymbol{w}^T \tilde{\boldsymbol{w}}_i = 0,\ \forall i = 1, \ldots, k-1$. En pratique, la néguentropie est estimée par

$$\hat{J}(y) = \hat{J}(\boldsymbol{w}^T \boldsymbol{\mathcal{X}}^T) = \frac{\left(\frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{w}^T \mathbf{x}_i)^3 \right)^2}{12} + \frac{\left(\frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{w}^T \mathbf{x}_i)^4 - 3 \right)^2}{48}. \tag{4.1.5}$$

De la même façon, en utilisant l'approximation (2.7.5), on obtient

$$\tilde{\boldsymbol{w}}_k = \arg\max_{\boldsymbol{w}} \left\{ \left(\mathbb{E}\left[\exp\left(-\frac{1}{2} \left(\boldsymbol{w}^T \mathbf{x} \right)^2 \right) \right] - \frac{1}{\sqrt{2}} \right)^2 \right\} \tag{4.1.6}$$

sous les contraintes $\|\boldsymbol{w}\|^2 = 1$ et $\boldsymbol{w}^T \tilde{\boldsymbol{w}}_i = 0,\ \forall i = 1, \ldots, k-1$. En pratique, on estime cette approximation de la néguentropie par

$$J(\widehat{y}) = J(\boldsymbol{w}^{\widehat{T}} \boldsymbol{\mathcal{X}}^T) = \left(\frac{1}{n} \sum_{i=1}^{n} \left[\exp\left(-\frac{1}{2} \left(\boldsymbol{w}^T \mathbf{x}_i \right)^2 \right) \right] - \frac{1}{\sqrt{2}} \right)^2. \tag{4.1.7}$$

Les méthodes vues ci-haut représentent seulement quelques méthodes classiques d'ACI par maximisation de la non normalité. N'importe quelle mesure de non normalité peut permettre de résoudre le problème d'ACI.

4.2. ACI par maximum de vraisemblance

Nous allons d'abord exprimer la densité $p_{\mathbf{X}}(.)$ des mélanges de sources en fonction de la densité des sources s_j puisque l'hypothèse principale du modèle d'ACI repose sur l'indépendance des sources s_j. On obtient le résultat suivant en effectuant le changement de variable $\mathbf{x} = \boldsymbol{\mathcal{A}}\mathbf{s}$:

$$p_{\mathbf{X}}(\mathbf{x}) = |\det \boldsymbol{\mathcal{W}}| p_{\mathbf{S}}(\mathbf{s}) = |\det \boldsymbol{\mathcal{W}}| \prod_{j=1}^{p} p_{S_j}(s_j), \tag{4.2.1}$$

où $\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{A}}^{-1}$. On peut exprimer (4.2.1) en fonction de $\boldsymbol{\mathcal{W}}$ et $\mathbf{x}$ seulement

$$p_{\mathbf{X}}(\mathbf{x}) = |\det \boldsymbol{\mathcal{W}}| \prod_{j=1}^{p} p_{S_j}(\boldsymbol{w}_j^T \mathbf{x}). \tag{4.2.2}$$

Soit un échantillon de données $\mathbf{x}_1, \ldots, \mathbf{x}_n$ provenant des variables aléatoires $\mathbf{x}_1, \ldots, \mathbf{x}_n$ disposé dans la matrice $\boldsymbol{\mathcal{X}} : n \times p$. La vraisemblance de l'échantillon est donnée par

$$L(\mathbf{x}_1, \ldots, \mathbf{x}_n | \boldsymbol{\mathcal{W}}) = \prod_{i=1}^{n} p_{\mathbf{X}_i}(\mathbf{x}_i) = \prod_{i=1}^{n} |\det \boldsymbol{\mathcal{W}}| \prod_{j=1}^{p} p_{S_j}(\boldsymbol{w}_j^T \mathbf{x}_i). \tag{4.2.3}$$

Il est plus facile de travailler avec la log-vraisemblance. Elle est donnée par

$$\log L(\mathbf{x}_1, \ldots, \mathbf{x}_n | \boldsymbol{\mathcal{W}}) = \sum_{i=1}^{n} \sum_{j=1}^{p} \log p_{S_j}(\boldsymbol{w}_j^T \mathbf{x}_i) + n \log |\det \boldsymbol{\mathcal{W}}|. \tag{4.2.4}$$

La question qui se pose maintenant est : comment estimer la densité des sources p_{S_j} ? Une méthode très utilisée en ACI est l'approximation de la densité des sources par une famille de densités qui est spécifiée par un nombre fini de paramètres. Dans le contexte d'ACI, on peut montrer qu'il est suffisant d'estimer les sources parmi deux classes de densités seulement. Pour chacune des composantes indépendantes, il suffit simplement de déterminer laquelle des deux approximations est la meilleure. La validité de cette approche est justifiée par le théorème suivant qui se trouve dans Hyvärinen et coll. (2001) :

Théorème 4.2.1. *Notons* $\tilde{p}_{S_i}$ *la densité de la source* S_i *et définissons*

$$g_{S_i}(s_i) = \frac{\partial}{\partial s_i} \log \tilde{p}_{S_i} = \frac{\tilde{p}'_{S_i}}{\tilde{p}_{S_i}}. \tag{4.2.5}$$

Supposons que les composantes « indépendantes » $Y_i = \boldsymbol{w}_i^T \mathbf{x}$ *soient non-corrélées et de variance unitaire. L'estimateur du maximum de vraisemblance est localement convergeant si la densité* $\tilde{p}_{S_i}$ *vérifie la condition suivante*

$$\mathbb{E}\{S_i g_{S_i}(s_i) - g'_{S_i}(s_i)\} > 0 \tag{4.2.6}$$

pour tout i.

Ce théorème nous montre qu'on peut estimer la densité des sources par n'importe quelle densité $\tilde{p}_{S_i}$, en autant que cette densité satisfasse la condition (4.2.6). Si la condition est vérifiée, alors l'estimateur du maximum de vraisemblance sera localement convergeant.

Ce théorème nous montre aussi comment choisir les deux familles de densités. On n'a qu'à choisir deux densités de façon à ce que la condition (4.2.6) soit toujours respectée pour l'une d'entre elles. Par exemple, on peut utiliser les log-densités suivantes :

$$\log \tilde{p}^+_{S_i}(s_i) = \alpha_1 - 2 \log \cosh(s_i) \tag{4.2.7}$$

$$\log \tilde{p}^-_{S_i}(s_i) = \alpha_2 - \left(\frac{s_i^2}{2} - \log \cosh(s_i) \right) \tag{4.2.8}$$

où α_1 et α_2 sont des paramètres positifs qui sont fixés de façon à ce que les fonctions soient des densités de probabilité. Ces constantes peuvent être ignorées dans ce qui suit.

La motivation derrière le choix de ces densités est que la condition (4.2.6) est vérifiée pour l'une ou l'autre de ces densités. En effet, le terme à gauche de l'inégalité de la condition (4.2.6) pour $\log \tilde{p}^+_{S_i}(s_i)$ est

$$2\mathbb{E}\left\{ -\tanh(S_i)S_i + \left(1 - \tanh(S_i)^2\right) \right\} \tag{4.2.9}$$

alors qu'il est

$$\mathbb{E}\left\{ \tanh(S_i)S_i - \left(1 - \tanh(S_i)^2\right) \right\} \tag{4.2.10}$$

pour $\log \tilde{p}^-_{S_i}(s_i)$. Les résultats sont obtenus en fixant $\mathbb{E}\{S_i^2\} = 1$. On remarque que le terme de l'équation (4.2.9) est égal à celui de l'équation (4.2.10) si on multiplie ce dernier par -2 ; le signe de chacun des termes est différent. Donc, l'estimation de la densité des sources se fait simplement en choisissant la densité parmi (4.2.7) et (4.2.8) qui satisfait la condition (4.2.6).

Maintenant qu'on peut estimer la log-vraisemblance de l'équation (4.2.4), on peut estimer la matrice de séparation $\mathcal{W}$ par l'estimateur du maximum de vraisemblance. Avant d'effectuer l'analyse en composantes indépendantes, la matrice contenant les données est préalablement centrée et blanchie selon les équations (1.3.3) et (3.2.12) respectives. Puisque les données sont blanchies, alors la matrice de séparation $\mathcal{W}$ que l'on cherche est orthogonale.

Un estimateur du maximum de vraisemblance de $\mathcal{W}$ est alors donné par

$$\hat{\mathbf{W}}_{\mathrm{MV}} = \arg\max_{\mathcal{W}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{p} \log p_{S_j}(\boldsymbol{w}_j^{\mathsf{T}} \mathbf{x}_i) + n \log |\det \mathcal{W}| \right\} \tag{4.2.11}$$

sous la contrainte que $\mathcal{W}^{\mathsf{T}}\mathcal{W} = \mathcal{I}$.

4.3. ACI par information mutuelle

Nous avons vu à la section 2.4 que l'information mutuelle est une mesure de dépendance entre des variables aléatoires. Elle est toujours positive et vaut 0 si et seulement si les variables sont indépendantes. On peut donc utiliser l'information mutuelle comme critère de dépendance afin de retrouver des composantes $Y_1, \ldots, Y_p$ les plus indépendantes possible.

Supposons que les mélanges de sources ont été blanchis préalablement de façon à ce que $\mathcal{C}_{\mathbf{X}} = \mathcal{I}$. À l'aide de l'entropie d'une transformation décrite à l'équation (2.3.4), on peut exprimer l'information mutuelle des composantes indépendantes comme

$$I(Y_1, Y_2, \ldots, Y_p) = \sum_{i=1}^{p} H(Y_i) - H(\mathbf{Y}) = \sum_{i=1}^{p} H(Y_i) - H(\mathbf{x}) - \log |\det \mathcal{W}|. \tag{4.3.1}$$

Nous avons vu que le blanchiment résout le problème d'ACI à une matrice orthogonale près. Puisque les mélanges de sources ont été blanchies préalablement, ceci implique que la matrice $\mathcal{W}$ est orthogonale. Or, la valeur absolue du déterminant d'une matrice orthogonale vaut toujours 1. Le dernier terme à droite de l'équation vaut donc 0. De plus, l'entropie des mélanges de sources blanchis $H(\mathbf{x})$ est une constante. On obtient alors

$$I(Y_1, Y_2, \ldots, Y_p) = \sum_{i=1}^{p} H(Y_i) - \text{constante}. \tag{4.3.2}$$

Puisque l'information mutuelle est toujours positive, on n'a qu'à minimiser le terme $\sum_{i=1}^{p} H(Y_i)$ afin d'obtenir des composantes les plus indépendantes possible.

4.3.1. Information mutuelle et non gaussianité

On se rappelle que la néguentropie est définie comme

$$J(Y) = H(Y_{gauss}) - H(Y). \tag{4.3.3}$$

On peut alors exprimer l'information mutuelle de l'équation (4.3.2) comme

$$I(Y_1, Y_2, \ldots, Y_p) = \sum_{j=1}^{p} [H(Y_{j,gauss}) - J(Y_j)] - \text{constante}. \tag{4.3.4}$$

L'entropie $H(Y_{j,gauss})$ est constante. On obtient alors

$$I(Y_1, Y_2, \ldots, Y_p) = \text{constante} - \sum_{j=1}^{p} J(Y_j). \tag{4.3.5}$$

Ceci démontre bien la relation entre l'information mutuelle et la néguentropie. On constate que trouver la transformation linéaire et inversible $\boldsymbol{\mathcal{W}}$ qui minimise l'information mutuelle est sensiblement la même chose que trouver les directions qui maximisent la néguentropie. Nous avons vu que la néguentropie est une mesure de non gaussianité. Or, l'équation (4.3.5) nous montre bien que l'ACI par la minimisation de l'information mutuelle est équivalent à la maximisation de la somme de mesures de non gaussianité des composantes indépendantes.

Ceci ajoute du poids à l'idée de maximiser des mesures de non normalité. Puisque l'entropie est une mesure de non normalité, on en conclut que la maximisation de la non normalité mène à l'indépendance des composantes « indépendantes ».

Par contre, en pratique, il y a une différence importante entre les deux méthodes de résolution du problème d'ACI. La néguentropie, ainsi que les autres mesures de non gaussianité, nous permet de trouver des composantes « indépendantes » l'une après l'autre, puisque nous trouvons le maximum de non gaussianité dans une seule direction $\boldsymbol{w}$ à la fois. Ceci n'est pas possible avec l'information mutuelle.

4.3.2. Information mutuelle et vraisemblance

L'information mutuelle et la vraisemblance sont directement liées. Rappelons d'abord la vraisemblance du modèle d'ACI :

$$\log L(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n | \boldsymbol{\mathcal{W}}) = \sum_{i=1}^{n} \sum_{j=1}^{p} \log p_{S_j}(\boldsymbol{w}_j^T \boldsymbol{x}_i) + n \log |\det \boldsymbol{\mathcal{W}}|. \tag{4.3.6}$$

Afin de mieux voir la connection entre l'information mutuelle et la vraisemblance, nous pouvons exprimer la vraisemblance de l'équation précédente sous la forme d'une espérance. Le tout est obtenu en divisant par n de chaque côté de l'équation et en considérant la moyenne échantillonale comme une espérance :

$$\frac{1}{n} \log L(\mathbf{x}_1, \ldots, \mathbf{x}_n | \boldsymbol{\mathcal{W}}) = \sum_{i=1}^{p} \mathbb{E}\{\log p_{S_i}(\boldsymbol{w}_i^T \mathbf{x})\} + \log |\det \boldsymbol{\mathcal{W}}|. \tag{4.3.7}$$

Nous pouvons aussi exprimer l'entropie sous la forme d'une espérance tel que $H(Y_i) = -\mathbb{E}\{\log p_{Y_i}(y_i)\} = -\mathbb{E}\{\log p_{Y_i}(\boldsymbol{w}_i^T \mathbf{x})\}$. Si les densités des composantes indépendantes p_{Y_i} étaient les mêmes que les densités des sources p_{S_i} alors on pourrait exprimer la vraisemblance comme

$$\frac{1}{n} \log L(\mathbf{x}_1, \ldots, \mathbf{x}_n | \boldsymbol{\mathcal{W}}) = \text{constante} - \sum_{i=1}^{p} H(Y_i), \tag{4.3.8}$$

ce qui est identique au signe près à l'équation (4.3.2). Dans le contexte de maximum de vraisemblance, on cherche $\boldsymbol{\mathcal{W}}$ de façon à ce que la vraisemblance soit maximale alors que dans le contexte d'information mutuelle, on cherche $\boldsymbol{\mathcal{W}}$ de façon à ce que l'information mutuelle soit minimale. Ceci explique la différence du signe.

En pratique, la connection entre l'information mutuelle et la vraisemblance est encore plus forte, car on ne connait pas la densité des sources. Dans les deux cas, on doit estimer la densité par la densité des composantes « indépendantes ». Il y a donc aucune différence entre les deux.

Chapitre 5

PÉNALITÉS PRODUISANT DES COEFFICIENTS ÉPARSES EN RÉGRESSION LINÉAIRE

Dans ce chapitre, plusieurs fonctions de pénalité seront couvertes. Certaines pénalités seront utilisées dans le chapitre 6 - ACI avec une matrice de transformation éparse.

5.1. Propriétés oracles d'une fonction de pénalité

Considérons d'abord le problème d'estimation et de sélection de variables dans un modèle de régression linéaire :

$$\mathbf{Y} = \boldsymbol{\mathcal{X}}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{5.1.1}$$

où $\mathbf{Y} : n \times 1$ est la variable réponse, $\boldsymbol{\mathcal{X}} = [\mathbf{x}_1, \ldots, \mathbf{x}_p] : n \times p$ est la matrice contenant les variables explicatives et où $\boldsymbol{\epsilon} : n \times 1$ est le bruit du modèle. On suppose généralement que $\boldsymbol{\mathcal{X}}$ est de plein rang et que $\boldsymbol{\epsilon} \sim N(0, \sigma^2\boldsymbol{\mathcal{I}})$. Ici, on s'intéresse à une représentation éparse de $\boldsymbol{\beta}$, c'est-à-dire que certaines composantes de $\boldsymbol{\beta}$ sont nulles. L'estimateur classique de $\boldsymbol{\beta}$ est obtenu en minimisant les moindres carrés :

$$\hat{\boldsymbol{\beta}}_{MC} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{Y} - \boldsymbol{\mathcal{X}}\boldsymbol{\beta}\|_2^2 = \left(\boldsymbol{\mathcal{X}}^T\boldsymbol{\mathcal{X}}\right)^{-1}\boldsymbol{\mathcal{X}}^T\mathbf{Y}. \tag{5.1.2}$$

Les procédures de sélection de variables traditionnelles comprennent principalement les procédures pas-à-pas ainsi que les procédures optimisant un certain critère tel que R^2, AIC et BIC. Or, selon Breiman (1996), ces procédures sont instables, ce qui résulte en un modèle ayant une mauvaise précision dans ses prédictions. Pour remédier aux inconvénients de ces procédures, les statisticiens ont récemment proposé plusieurs méthodes de pénalisation des moindres carrés

afin de sélectionner et d'estimer le modèle simultanément. Fan et Li (2001) affirment qu'une bonne fonction de pénalité devrait avoir les propriétés oracles suivantes :

(1) *Sans biais* : Les estimateurs des paramètres significativement différents de zéro doivent être sans biais pour la vraie valeur des paramètres.

(2) *Parcimonie* : La fonction de pénalité doit créer une représentation éparse, c'est-à-dire que les paramètres non significativement différents de zéro doivent être automatiquement mis à 0.

(3) *Continuité* : L'estimateur résultant doit être continu par rapport aux données afin d'éviter une variation brusque dans l'estimation.

Ce qui suit présente un aperçu des pénalités qui sont couramment utilisées dans un contexte de régression linéaire.

5.2. Pénalisation SCAD

La pénalisation SCAD (*Smoothly Clipped Absolute Deviation*) a été proposé par Fan et Li (2001). Cette pénalisation a été construite de façon à ce qu'elle possède les propriétés oracles mentionnées ci-haut. L'estimateur de $\boldsymbol{\beta}$ par la pénalisation SCAD, noté $\widehat{\boldsymbol{\beta}}_{\text{SCAD}}$, est le suivant :

$$\widehat{\boldsymbol{\beta}}_{\text{SCAD}} = \arg\min_{\boldsymbol{\beta}} \left\{ \|\mathbf{Y} - \boldsymbol{\mathcal{X}}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^{p} P_\lambda(\beta_j) \right\}, \tag{5.2.1}$$

où $P_\lambda(\beta_j)$ est une pénalisation qui dépend d'un paramètre λ et qui peut être différente pour différents β_j. Sa dérivée par rapport à β_j est :

$$P'_\lambda(\beta_j) = \lambda \left(I(\beta_j \leq \lambda) + \frac{(a\lambda - \beta_j)_+}{(a-1)\lambda} I(\beta_j > \lambda) \right), \tag{5.2.2}$$

où $(\cdot)_+$ représente la partie positive et où $I(\cdot)$ est une fonction assignant 1 si la condition est respectée et 0 sinon. La valeur des paramètres λ et a est habituellement fixée à $\sqrt{2\log(p)}$ et 3.7 respectivement.

5.3. Pénalisation LASSO et LASSO adaptatif

Le LASSO (*Least Absolute Shrinkage and Selection Operator*) a été proposé par Tibshirani (1996). Le LASSO est une pénalisation de la vraisemblance par la norme l_1 des coefficients de la régression. L'estimateur de $\boldsymbol{\beta}$ par le LASSO, noté $\widehat{\boldsymbol{\beta}}_{\text{L}}$, est le suivant :

$$\widehat{\boldsymbol{\beta}}_{\text{L}} = \arg\min_{\boldsymbol{\beta}} \left\{ \|\mathbf{Y} - \boldsymbol{\mathcal{X}}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \right\}. \tag{5.3.1}$$

Le LASSO effectue l'estimation du modèle simultanément avec une sélection de variables qui se traduit par le rétrécissement des coefficients non-significatifs vers 0 selon un choix de λ approprié. Par contre, le LASSO ne possède pas les propriétés oracles d'une bonne fonction de pénalité mentionnées ci-haut, car il a été démontré qu'asymptotiquement, le LASSO a un biais non négligeable pour l'estimation des coefficients significativement différents de 0. De plus, le LASSO a les limitations suivantes. Dans le cas où $p > n$, le LASSO choisit au plus n variables dans le modèle. Aussi, s'il y a un groupe de variables dans lesquelles la corrélation par paire est très élevée, alors le LASSO a tendance à sélectionner seulement une variable parmi ce groupe.

Le LASSO adaptatif a été proposé par Zou (2006). L'estimateur de $\boldsymbol{\beta}$ par le LASSO adaptatif, noté $\hat{\boldsymbol{\beta}}_{LA}$, est le suivant :

$$\hat{\boldsymbol{\beta}}_{LA} = \arg\min_{\boldsymbol{\beta}} \left\{ \|\mathbf{Y} - \boldsymbol{\mathcal{X}}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{|\hat{\beta}_j|^{\gamma}} \right\}, \tag{5.3.2}$$

où $\hat{\boldsymbol{\beta}}$ est un estimateur $\sqrt{n}$-convergent du vrai $\boldsymbol{\beta}$ et où γ est une constante positive. Zou a démontré qu'avec un choix approprié de λ, le LASSO adaptatif possède les propriétés oracles. Les méthodes de pénalisation sur la norme l_1 ne fonctionnent pas très bien lorsque les variables explicatives sont grandement corrélées et ce problème est courant dans les modèles ayant une grande dimension.

5.4. Pénalisation elastic net et élastic net adaptatif

Dans le cas où les variables explicatives sont indépendantes et que la dimension est grande, il a été démontré que le maximum de la corrélation échantillonnale peut tout de même être élevé. C'est pourquoi il serait bien d'avoir une fonction de pénalité qui s'adapte bien dans le cas de grandes dimensions. Zou et Hastie (2005) ont proposé l'elastic net, une version améliorée du LASSO dans le cas de grandes dimensions. L'estimateur de $\boldsymbol{\beta}$ par l'elastic net, noté $\hat{\boldsymbol{\beta}}_{EN}$, est le suivant :

$$\hat{\boldsymbol{\beta}}_{EN} = \left(1 + \frac{\lambda_2}{n}\right) \left[\arg\min_{\boldsymbol{\beta}} \left\{ \|\mathbf{Y} - \boldsymbol{\mathcal{X}}\boldsymbol{\beta}\|_2^2 + \lambda_2 \|\boldsymbol{\beta}\|_2^2 + \lambda_1 \|\boldsymbol{\beta}\|_1 \right\} \right]. \tag{5.4.1}$$

Si les variables explicatives sont standardisées, alors le coefficient $\left(1 + \frac{\lambda_2}{n}\right)$ devrait être changé par $(1 + \lambda_2)$. La norme l_1 de l'elastic net effectue automatiquement une sélection de variables alors que la norme l_2 stabilise la solution et donc améliore la prédiction. L'elastic net ne possède pas les propriétés oracles d'une bonne fonction de pénalité. Dans un plan orthogonal où le LASSO est optimal, l'elastic net se réduit au LASSO. Par contre, lorsque la corrélation entre

les covariables est élevée, l'elastic net peut grandement améliorer l'efficacité de la prédiction du LASSO.

Le LASSO adaptatif et l'elastic net améliorent le LASSO dans deux directions différentes. Le LASSO adaptatif atteint les propriétés oracles alors que l'elastic net prend en compte la colinéarité. Il est sensé de croire qu'on peut combiner les deux pénalités afin de pouvoir améliorer le LASSO dans les deux directions. Cette idée de Zou et Zhang (2009) a donné place à l'elastic net adaptatif. L'estimateur de $\boldsymbol{\beta}$ par l'elastic net adaptatif, noté $\widehat{\boldsymbol{\beta}}_{\text{ENA}}$, est le suivant :

$$\widehat{\boldsymbol{\beta}}_{\text{ENA}} = \left(1 + \frac{\lambda_2}{n}\right)\left[\arg\min_{\boldsymbol{\beta}}\left\{\|\mathbf{Y} - \boldsymbol{\mathcal{X}}\boldsymbol{\beta}\|_2^2 + \lambda_2\|\boldsymbol{\beta}\|_2^2 + \lambda_1\sum_{j=1}^{p}\frac{|\beta_j|}{|\widehat{\beta}_j|^\gamma}\right\}\right]. \quad (5.4.2)$$

Encore une fois, si les variables explicatives sont standardisées, alors le coefficient $\left(1 + \frac{\lambda_2}{n}\right)$ devrait être changé par $(1 + \lambda_2)$. Comparativement à l'elastic net, l'elastic net adaptatif possède les propriétés oracles d'une bonne fonction de pénalité. De plus, il conserve la propriété intéressante de l'elastic net, c'est-à-dire qu'il est très efficace dans le cas où les variables explicatives sont corrélées.

5.5. Pénalisation LASSO par groupe et LASSO par groupe adaptatif

Les pénalisations vues précédemment ont été conçues pour sélectionner les variables explicatives individuellement. Par contre, il y a des situations où il est préférable de choisir les variables de façon groupée. À cette fin, Yuan et Lin (2006) ont développé le LASSO par groupe. Ce dernier pénalise les coefficients groupés de manière similaire au LASSO. Le modèle de régression linéaire peut être réécrit en considérant des coefficients groupés :

$$\mathbf{Y} = \boldsymbol{\mathcal{X}}\boldsymbol{\beta} + \boldsymbol{\epsilon} = \sum_{j=1}^{m}\boldsymbol{\mathcal{X}}_j\boldsymbol{\beta}_j + \boldsymbol{\epsilon}. \quad (5.5.1)$$

Dans ce cas-ci, on suppose que $\boldsymbol{\mathcal{X}}$ peut être groupé en m facteurs $\boldsymbol{\mathcal{X}} = (\boldsymbol{\mathcal{X}}_1, \ldots, \boldsymbol{\mathcal{X}}_m)$, où $\boldsymbol{\mathcal{X}}_j = (\mathbf{x}_1, \ldots, \mathbf{x}_{p_j})$ est un groupe de p_j variables. On note que $\sum_{j=1}^{m} p_j = p$. De la même façon, on peut découper $\boldsymbol{\beta}$ en m composantes $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\intercal, \ldots, \boldsymbol{\beta}_m^\intercal)^\intercal$, où $\boldsymbol{\beta}_j$ est de taille p_j. L'estimateur de $\boldsymbol{\beta}$ par le LASSO par groupe, noté $\widehat{\boldsymbol{\beta}}_{\text{LG}}$, est le suivant :

$$\widehat{\boldsymbol{\beta}}_{\text{LG}} = \arg\min_{\boldsymbol{\beta}}\left\{\|\mathbf{Y} - \boldsymbol{\mathcal{X}}\boldsymbol{\beta}\|^2 + \lambda\sum_{j=1}^{m}\|\boldsymbol{\beta}_j\|_2\right\}. \quad (5.5.2)$$

Cette procédure agit comme le LASSO au niveau des groupes ; selon un λ approprié, tous les coefficients d'un groupe seront mis à zéro simultanément. Cependant, le LASSO par groupe ne produit pas de parcimonie au sein d'un groupe. Il est important de noter que si la taille de chaque groupe est de 1 (c'est-à-dire $p_j = 1 \; \forall \, j$), alors le LASSO par groupe se réduit au LASSO. De plus, le LASSO par groupe ne possède pas les propriétés oracles puisqu'il pénalise tous les groupes de la même façon, ce qui résulte en une estimation biaisée des coefficients significativement différents de zéro. Afin d'éviter ce problème, Wang et Leng (2008) ont proposé le LASSO par groupe adaptatif. L'estimateur de $\boldsymbol{\beta}$ par le LASSO par groupe adaptatif, noté $\hat{\boldsymbol{\beta}}_{\text{LGA}}$, est le suivant :

$$\hat{\boldsymbol{\beta}}_{\text{LGA}} = \arg\min_{\boldsymbol{\beta}} \left\{ \|\mathbf{Y} - \boldsymbol{\mathcal{X}}\boldsymbol{\beta}\|^2 + \lambda \sum_{j=1}^{m} \frac{\|\boldsymbol{\beta}_j\|_2}{\|\hat{\boldsymbol{\beta}}_j\|_2^{\gamma}} \right\}, \tag{5.5.3}$$

où $\hat{\boldsymbol{\beta}}$ est un estimateur $\sqrt{n}$-convergent du vrai $\boldsymbol{\beta}$ et où γ est une constante positive. Contrairement au LASSO par groupe, le LASSO par groupe adaptatif possède les propriétés oracles définies précédemment.

5.6. Choix des paramètres

En régression, le choix des paramètres λ_1 et λ_2 se fait généralement par la validation croisée. La validation croisée est une méthode d'estimation de la fiabilité d'un modèle fondée sur une technique d'échantillonnage. On choisit habituellement une grille de valeurs pour les paramètres à estimer, puis, pour chacune des valeurs du paramètre, on met en oeuvre la procédure suivante.

On divise d'abord l'échantillon en k sous-échantillons. On sélectionne ensuite un des k sous-échantillons comme ensemble de validation et les $k-1$ autres sous-échantillons constitueront l'ensemble d'apprentissage. L'ensemble d'apprentissage nous permet de calculer l'erreur du modèle, qui est mesurée à l'aide de l'erreur quadratique moyenne. Puis on répète l'opération en sélectionnant un autre sous-échantillon de validation parmi les $k-1$ sous-échantillons qui n'ont pas encore été utilisés pour la validation du modèle. L'opération se répète ainsi k fois pour qu'en fin de compte chaque sous-échantillon ait été utilisé exactement une fois comme ensemble de validation. La moyenne des k erreurs quadratiques moyennes est enfin calculée pour estimer l'erreur de prédiction.

Finalement, les paramètres sont fixés à la valeur parmi la grille de valeurs qui offre la plus petite erreur de prédiction.

Chapitre 6

ACI AVEC UNE MATRICE DE TRANSFORMATION ÉPARSE

6.1. Avantages de l'ACI avec une matrice de transformation éparse

Considérons d'abord le modèle de génération de données supposé en ACI tel que mentionné précédemment :

$$\mathbf{x} = \boldsymbol{\mathcal{A}}\mathbf{s}. \tag{6.1.1}$$

En ACI, le but est de retrouver des composantes $\mathbf{y} = \boldsymbol{\mathcal{W}}\mathbf{x}$ les plus indépendantes possible. En outre, dans certaines applications, plusieurs coefficients de la matrice de transformation, que ce soit $\boldsymbol{\mathcal{A}}$ ou $\boldsymbol{\mathcal{W}}$, valent 0. Il est possible d'estimer la matrice de transformation tout en rétrécissant les coefficients non significatifs vers 0 à l'aide de ce qu'on appelle l'ACI avec une matrice de transformation éparse.

Avant d'aller plus loin, il est important de faire la distinction entre l'*ACI éparse* et l'*ACI avec une matrice de transformation éparse*. En ACI avec une matrice de transformation éparse, l'hypothèse de parcimonie est faite sur la matrice de transformation alors qu'en ACI éparse, l'hypothèse de parcimonie est posée sur les sources. Dans ce dernier cas, on se sert surtout de l'aspect géométrique des sources afin de retrouver ces dernières. Par exemple, si chacune des sources s_j est composée de seulement 10% d'entrées non nulles, alors il y a de fortes chances que la majorité des entrées des mélanges de sources x_j provienne seulement d'une source. Pour plus d'informations, on peut consulter Bronstein et coll. (2005).

Il y a plusieurs raisons pour lesquelles on souhaiterait retrouver une matrice de transformation éparse.

Premièrement, considérons le cas où la dimension p des données est grande et où les vraies valeurs de certaines entrées de la matrice de transformation sont nulles. Si on peut automatiquement cibler et mettre ces entrées à 0, alors le modèle devient beaucoup plus simple et l'estimation des paramètres devient plus fiable.

Deuxièmement, une matrice de transformation éparse signifie que les mélanges de sources x_j sont une combinaison linéaire d'un plus petit ensemble de sources indépendantes s_j. L'interprétation du modèle d'ACI devient alors beaucoup moins compliquée lorsque la matrice de transformation est éparse.

Troisièmement, la parcimonie de la matrice de transformation permet de résoudre le modèle LiNGAM (modèle acyclique, linéaire et non gaussien) de Shimizu et coll. (2006). Dans le modèle LiNGAM, les variables observées $x_i, i \in \{1, \ldots, n\}$, peuvent être arrangées dans un ordre causal de façon à ce que chaque variable ne dépende que des variables précédentes. Cet ordre causal est dénoté par $k(i)$. Le modèle LiNGAM est défini comme suit :

$$x_i = \sum_{k(j)<k(i)} b_{ij} x_j + \epsilon_i + c_i, \tag{6.1.2}$$

où c_i est un terme constant associé à x_i et où ϵ_i représente le bruit du modèle. On considère que les ϵ_i sont des variables aléatoires continues et indépendantes ayant des distributions non gaussiennes de variances non nulles. De plus, on suppose qu'on peut observer plusieurs vecteurs de données $\boldsymbol{x}$ (comprenant les composantes x_i), où chacun d'entre eux ont les mêmes coefficients b_{ij}, les mêmes constantes c_i et où les ϵ_i sont échantillonnées indépendamment à partir des mêmes distributions. Le modèle LiNGAM peut être réécrit sous la forme suivante, en considérant que les variables observées x_i sont préalablement centrées :

$$\boldsymbol{x} = \boldsymbol{\mathcal{B}}\boldsymbol{x} + \epsilon. \tag{6.1.3}$$

Si on connaissait l'ordre causal $k(i)$, alors la matrice $\boldsymbol{\mathcal{B}}$ pourrait être permutée de façon à être triangulaire inférieure et à avoir des 0 sur la diagonale. L'équation (6.1.2) peut être résolue par rapport à $\boldsymbol{x}$:

$$\boldsymbol{x} = \boldsymbol{\mathcal{A}}\epsilon, \tag{6.1.4}$$

où $\boldsymbol{\mathcal{A}} = (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{B}})^{-1}$. Encore une fois, si on connaissait l'ordre causal, alors la matrice $\boldsymbol{\mathcal{A}}$ pourrait être permutée de façon à être triangulaire inférieure. On remarque que ce modèle peut être traité avec l'ACI vu précédemment, puisque les composantes de ϵ sont indépendantes et proviennent de distributions non gaussiennes. Cependant, si on sait que la matrice de mélange $\boldsymbol{\mathcal{A}}$ contient beaucoup

d'éléments nuls, alors on pourrait utiliser cette information supplémentaire afin d'augmenter la précision de l'estimation de $\boldsymbol{\mathcal{A}}$. C'est pour cette raison que l'ACI avec une matrice de transformation éparse peut être utilisée dans ce contexte.

6.2. Résolution du problème d'ACI avec une matrice de transformation éparse

Selon Zhang et Chan (2006), sous certaines conditions faibles, les estimateurs du maximum de vraisemblance sont asymptotiquement normaux. Ceci implique que les pénalités mentionnées au chapitre précédent s'appliquent au modèle de régression linéaire tout comme ils s'appliquent aux modèles pouvant s'exprimer sous la forme de vraisemblance. Or, nous avons vu à la section 4.2 que le problème d'analyse en composantes indépendantes peut être résolu à l'aide du maximum de vraisemblance. On peut alors utiliser une vraisemblance pénalisée afin de résoudre le problème d'ACI avec une matrice de transformation éparse. Rappelons d'abord la log-vraisemblance du modèle d'ACI :

$$\log L(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n | \boldsymbol{\mathcal{A}}) = \sum_{i=1}^{n} \sum_{j=1}^{p} \log p_{S_j}(\boldsymbol{w}_j^T \boldsymbol{x}_i) + n \log |\det \boldsymbol{\mathcal{W}}|. \tag{6.2.1}$$

La parcimonie de la matrice de transformation, que ce soit $\boldsymbol{\mathcal{A}}$ ou bien $\boldsymbol{\mathcal{W}}$, sera atteinte en pénalisant les entrées de la matrice à l'aide des pénalisations vues au chapitre 5. Ici, la pénalisation se fera seulement sur la matrice de mélange $\boldsymbol{\mathcal{A}}$. Une pénalisation sur la matrice de séparation $\boldsymbol{\mathcal{W}}$ peut très bien se faire de façon similaire. La log-vraisemblance pénalisée est de la forme suivante :

$$l_p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n | \boldsymbol{\mathcal{A}}) = \sum_{i=1}^{n} \sum_{j=1}^{p} \log p_{S_j}(\boldsymbol{w}_j^T \boldsymbol{x}_i) + n \log |\det \boldsymbol{\mathcal{W}}| - \sum_{i,j=1}^{p} P_\lambda(a_{ij}), \tag{6.2.2}$$

où $P_\lambda(a_{ij})$ est une fonction de pénalité qui dépend de λ et qui peut être différente pour chacun des coefficients a_{ij}. Cette fonction de pénalité peut être, par exemple, la pénalisation SCAD, la pénalisation LASSO (ou LASSO adaptatif), la pénalisation elastic net (ou elastic net adaptatif) ou bien la pénalisation LASSO par groupe (ou LASSO par groupe adaptatif).

Il est important de noter qu'en ACI avec une matrice de mélange éparse, la densité des sources ne peut plus être estimée par les deux familles de densité vues à la section 4.2 puisque le terme de pénalisation n'est pas constant. Les densités doivent alors être estimées différemment. Dans notre cas, nous utilisons

l'estimation rapide de la densité par un noyau laplacien tel qu'utilisé dans l'article de Chen (2006).

De plus, dans le contexte d'ACI, le paramètre λ ne peut pas être choisi à l'aide de la validation croisée, car on connait seulement le vecteur contenant les mélanges de sources $\mathbf{x}$. Ainsi, l'erreur quadratique moyenne ne peut pas être calculée. De ce fait, aucun indice ne nous permet de valider l'estimation de notre modèle.

Dans l'article de Zhang et Chan (2006), les auteurs utilisent les pénalisations LASSO, SCAD et une forme généralisée du SCAD. Ainsi, par exemple, la vraisemblance pénalisée pour le LASSO est de la forme suivante :

$$l_p(\mathbf{x}_1, \ldots, \mathbf{x}_n|\boldsymbol{\mathcal{A}}) = \sum_{i=1}^{n}\sum_{j=1}^{p} \log p_{S_j}(\boldsymbol{w}_j^T \mathbf{x}_i) + n \log|\det \boldsymbol{\mathcal{W}}| - \lambda \sum_{i,j=1}^{p} |a_{ij}|. \quad (6.2.3)$$

Dans une des simulations de cet article, les auteurs fixent le paramètre λ de façon expérimentale. Par contre, lorsque les auteurs essaient d'estimer un modèle LiNGAM, ils choisissent le λ qui convient le mieux au modèle LiNGAM. Pour y arriver, les auteurs considèrent une grille de valeurs pour λ et choisissent celui qui vérifie le mieux l'hypothèse du modèle LiNGAM, c'est-à-dire qu'il existe une permutation telle que $\boldsymbol{\mathcal{W}}$ soit triangulaire inférieure. Ils y parviennent en utilisant l'algorithme B de Shimizu et coll. (2006).

Dans l'article de Zhang et coll. (2009), les auteurs utilisent plutôt la pénalisation LASSO adaptatif avec $\gamma = 1$ (voir l'équation 5.3.2). La vraisemblance pénalisée du modèle d'ACI devient alors :

$$l_p(\mathbf{x}_1, \ldots, \mathbf{x}_n|\boldsymbol{\mathcal{A}}) = \sum_{i=1}^{n}\sum_{j=1}^{p} \log p_{S_j}(\boldsymbol{w}_j^T \mathbf{x}_i) + n \log|\det \boldsymbol{\mathcal{W}}| - \lambda \sum_{i,j=1}^{p} \frac{|a_{ij}|}{|\hat{a}_{ij}|}, \quad (6.2.4)$$

où $\hat{a}_{ij}$ correspond à l'élément (i, j) de la matrice $\hat{\boldsymbol{\mathcal{A}}}$. Cette matrice est un estimateur $\sqrt{n}$-convergent de $\boldsymbol{\mathcal{A}}$ et est habituellement l'estimateur du maximum de vraisemblance. Dans leur article, les auteurs fixent $\lambda = \lambda_{BIC} = \frac{1}{2}\log(n)$, où λ_{BIC} correspond au λ qui effectuerait une sélection de variables semblable au critère BIC (*Bayesian information criterion*).

6.3. Méthode d'ACI avec une matrice de mélange éparse basée sur le LASSO par groupe adaptatif

Dans cette section, nous proposons une nouvelle méthode d'ACI avec une matrice de mélange éparse. Cette méthode pénalise les coefficients groupés de la

matrice de mélange $\mathcal{A}$ avec des poids adaptatifs. Nous avons des raisons de croire que dans certaines applications, le LASSO par groupe adaptatif peut grandement faciliter l'interprétation du modèle d'ACI. De plus, tel que vu au chapitre 5, le LASSO par groupe adaptatif possède les propriétés oracles d'une bonne fonction de pénalité. Ainsi, l'estimation des coefficients groupés significativement différents de 0 demeure sans biais. De plus, si nous avons des raisons de croire que certains coefficients groupés de la matrice de mélange sont nuls, alors la précision de l'estimation $\mathcal{A}$ ne peut qu'être améliorée.

Avant d'expliquer la méthode plus en détails, il est important de définir la disposition des coefficients groupés au sein de la matrice de mélange $\mathcal{A}$. On se rappelle que le modèle de génération de données supposé en ACI peut être écrit en fonction des vecteurs colonnes $\mathbf{a}_j$ de la matrice de mélange $\mathcal{A}$:

$$\mathbf{x} = \mathcal{A}\mathbf{s} = \sum_{j=1}^{p} \mathbf{a}_j s_j. \tag{6.3.1}$$

Dans cette écriture, on voit bien que la j^e source influence les mélanges de sources $\mathbf{x}$ à travers l'intensité des coefficients contenus dans le vecteur colonne $\mathbf{a}_j = (a_{1j}, \ldots, a_{pj})^T$. Dans certaines applications, chacun des vecteurs colonnes $\mathbf{a}_j$ peut être découpé en m composantes $\mathbf{a}_j = (\mathbf{a}_{1j}^T, \ldots, \mathbf{a}_{mj}^T)^T$, où $\mathbf{a}_{ij} = (a_{ij1}, \ldots, a_{ijd_i})^T \in \mathbb{R}^{d_i}$. Chacun des vecteurs colonnes est découpé de la même façon, c'est-à-dire que la taille d_i de la composante (i, j) est la même pour chaque vecteur colonne. On note que $\sum_{i=1}^{m} d_i = p$.

La log-vraisemblance du modèle d'ACI avec la pénalisation LASSO par groupe adaptatif est de la forme suivante :

$$l_p(\mathbf{x}_1, \ldots, \mathbf{x}_n|\mathcal{A}) = \sum_{i=1}^{n}\sum_{j=1}^{p} \log p_{S_j}(\mathbf{w}_j^T \mathbf{x}_i) + \log|\det \mathcal{W}| - \lambda \sum_{i=1}^{m}\sum_{j=1}^{p} \frac{\|\mathbf{a}_{ij}\|_2}{\|\hat{\mathbf{a}}_{ij}\|_2^{\gamma}}. \tag{6.3.2}$$

où $\hat{\mathbf{a}}_{ij}$ représente le vecteur contenant le groupe (i, j) de coefficients de la matrice $\hat{\mathcal{A}}$, qui est habituellement l'estimateur du maximum de vraisemblance. L'estimateur de $\mathcal{A}$ que nous proposons, noté $\hat{\mathcal{A}}_{\text{LGA}}$, est celui qui maximise l'équation (6.3.2) :

$$\hat{\mathcal{A}}_{\text{LGA}} = \arg\max_{\mathcal{A}} \left\{ \sum_{i=1}^{n}\sum_{j=1}^{p} \log p_{S_j}(\mathbf{w}_j^T \mathbf{x}_i) + \log|\det \mathcal{W}| - \lambda \sum_{i=1}^{m}\sum_{j=1}^{p} \frac{\|\mathbf{a}_{ij}\|_2}{\|\hat{\mathbf{a}}_{ij}\|_2^{\gamma}} \right\}. \tag{6.3.3}$$

Il est important de noter que la disposition des groupes doit être telle que définie précédemment. En effet, l'ambiguité 2 mentionnée à la section 1.5 stipule

qu'on ne peut pas déterminer l'ordre des composantes indépendantes. En d'autres mots, ceci est équivalent à dire qu'on ne peut pas déterminer l'ordre des colonnes de la matrice de mélange $\boldsymbol{\mathcal{A}}$. Ainsi, si on veut que l'estimateur du maximum de vraisemblance $\widehat{\boldsymbol{\mathcal{A}}}$ corresponde à la matrice $\boldsymbol{\mathcal{A}}$ qui maximise l'équation (6.3.2), alors les groupes doivent être définis au sein des colonnes de $\boldsymbol{\mathcal{A}}$ et non entre celles-ci. Dans notre cas, nous utilisons $\lambda = \lambda_{\text{BIC}} = \frac{1}{2}\log(n)$ tel que proposé par Zhang et coll. (2009) et nous fixons $\gamma = 2$.

Afin d'éviter l'ambiguité de l'amplitude des sources, il y a deux solutions possibles. Premièrement, on peut standardiser les composantes indépendantes Y_j après chaque itération de façon à ce que $\mathbb{E}\{Y_j^2\} = 1$. Deuxièmement, on peut effectuer le blanchiment des données suivant

$$\mathbf{z} = \boldsymbol{\mathcal{V}}\mathbf{x} = \boldsymbol{\mathcal{V}}\boldsymbol{\mathcal{A}}\mathbf{s} = \widetilde{\boldsymbol{\mathcal{A}}}\mathbf{s}, \tag{6.3.4}$$

où $\boldsymbol{\mathcal{V}}$ correspond à la matrice de blanchiment. Puis, parmi les matrices orthogonales, on cherche la matrice $\widetilde{\boldsymbol{\mathcal{A}}}$ qui maximise l'équation suivante :

$$l_p(z_1, \ldots, z_n|\widetilde{\boldsymbol{\mathcal{A}}}) = \sum_{i=1}^{n}\sum_{j=1}^{p} \log p_{S_j}(\tilde{\boldsymbol{w}}_j^T z_i) + \log|\det \widetilde{\boldsymbol{\mathcal{W}}}| - \lambda \sum_{i=1}^{m}\sum_{j=1}^{p} \frac{\|\boldsymbol{a}_{ij}\|_2}{\|\hat{\boldsymbol{a}}_{ij}\|_2^{\gamma}}, \tag{6.3.5}$$

où $\widetilde{\boldsymbol{\mathcal{W}}} = \widetilde{\boldsymbol{\mathcal{A}}}^{-1}$ et où $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{V}}^{-1}\widetilde{\boldsymbol{\mathcal{A}}}$. Il est important de noter que l'hypothèse de parcimonie est faite sur la matrice de mélange $\boldsymbol{\mathcal{A}}$. Ainsi, peut importe le changement de variables effectué, la pénalisation doit toujours se faire sur la matrice de mélange $\boldsymbol{\mathcal{A}}$.

Chapitre 7

APPLICATION DE L'ACI AVEC UNE MATRICE DE MÉLANGE ÉPARSE PAR LE LASSO PAR GROUPE ADAPTATIF

7.1. Imagerie par résonance magnétique (IRM)

L'analyse en composantes indépendantes est beaucoup appliquée en imagerie cérébrale, notamment en imagerie par résonance magnétique (IRM). L'IRM permet de détecter les variations du taux d'oxygène sanguin du cerveau d'un sujet se trouvant à l'intérieur d'un appareil d'IRM, en réponse à divers stimuli ou tâches cognitives. L'objectif principal de l'IRM est d'explorer, de façon reproductible, les réseaux corticaux impliqués dans des tâches de stimulation prédéfinies. Les données résultant d'expériences d'IRM sont en général une représentation 3D du cerveau d'un sujet à différents temps, où chacun des voxels représente le taux d'oxygène sanguin du cerveau du sujet en question. Les données 3D sont tout d'abord « déroulées » de façon à obtenir, à chaque temps, un vecteur contenant tous les voxels du cerveau.

7.2. Modèle d'ACI dans le contexte d'IRM

Afin de retrouver le stimulus de l'expérience en question à partir des données obtenues par IRM, on effectue une analyse en composantes indépendantes (ACI). Pour ce faire, on suppose que la mesure d'un voxel à un temps donné est une combinaison linéaire de certaines sources que l'on peut retrouver dans le corps humain, notamment le stimulus de l'expérience, les battements du coeur ainsi que la respiration. On suppose aussi que les sources, à un temps donné, sont indépendantes et proviennent de lois non gaussiennes. Dans cas-ci, $\mathfrak{n}$ représente le nombre d'observations par voxel, $\mathfrak{p}$ représente le nombre de sources et $\mathfrak{q}$ représente le nombre de voxels. Les voxels à chaque temps sont regroupés en lignes dans la

matrice d'observations $\boldsymbol{\mathcal{X}} : n \times q$. On peut alors exprimer les voxels en fonction des sources selon le modèle d'ACI :

$$\boldsymbol{\mathcal{X}}^\intercal = \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{S}}^\intercal, \tag{7.2.1}$$

où $\boldsymbol{\mathcal{A}} : q \times p$ est la matrice de mélange et où $\boldsymbol{\mathcal{S}} : n \times p$ est la matrice contenant les observations des sources. On remarque que, contrairement à ce qui a été vu précédemment, la matrice de mélange n'est pas carrée. Or, une alternative est d'effectuer préalablement une analyse en composantes principales et de sélectionner seulement les p premières composantes. Ceci est équivalent à effectuer le blanchiment des données en ne considérant que les p premières lignes de la matrice de blanchiment de l'équation (3.2.2), soit $\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{D}}^{-1/2}\boldsymbol{\mathcal{E}}^\intercal : q \times q$. On note $\widetilde{\boldsymbol{\mathcal{V}}} : p \times q$ la matrice $\boldsymbol{\mathcal{V}}$ tronquée. Le modèle devient alors

$$\boldsymbol{\mathcal{Z}}^\intercal = \widetilde{\boldsymbol{\mathcal{V}}}\boldsymbol{\mathcal{X}}^\intercal = \widetilde{\boldsymbol{\mathcal{V}}}\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{S}}^\intercal = \widetilde{\boldsymbol{\mathcal{A}}}\boldsymbol{\mathcal{S}}^\intercal. \tag{7.2.2}$$

On remarque que la nouvelle matrice de mélange $\widetilde{\boldsymbol{\mathcal{A}}} = \widetilde{\boldsymbol{\mathcal{V}}}\boldsymbol{\mathcal{A}}$ est désormais carrée $p \times p$. Ainsi, on obtient un modèle d'ACI tel que vu précédemment. On peut alors résoudre le problème d'ACI à l'aide de l'estimateur du maximum de vraisemblance par exemple.

7.3. Hypothèse de parcimonie et structure de la matrice de mélange

Dans ce contexte, il est censé de croire que certaines sources ne se retrouvent que dans certaines parties du cerveau. Par exemple, si le stimulus de l'expérience est de type visuel, alors on s'attend à ce qu'il se retrouve principalement dans le cortex visuel du cerveau. Les différentes structures cérébrales peuvent être localisées à l'aide du référentiel de Talairach. Ce dernier est un système de coordonnées permettant de repérer la position de n'importe quel point dans le cerveau d'un individu quelconque en référence à un atlas publié par les médecins Jean Talairach et Pierre Tournoux.

À l'aide du référentiel de Talairach, les voxels de la représentation 3D du cerveau peuvent être regroupés en m structures cérébrales. Ceci est équivalent à dire que chaque colonne de la matrice de mélange $\boldsymbol{\mathcal{A}}$ contenant les voxels du cerveau peut être groupée en m groupes, où chacun d'entre eux représente une structure cérébrale. Ainsi, il est maintenant aisé d'effectuer l'ACI avec une matrice de mélange éparse obtenue par le LASSO par groupe adaptatif. On n'a qu'à choisir la matrice $\widetilde{\boldsymbol{\mathcal{A}}}$ maximisant la vraisemblance sous la contrainte de parcimonie au sein des groupes de la matrice de mélange $\boldsymbol{\mathcal{A}}$, c'est-à-dire à choisir $\widetilde{\boldsymbol{\mathcal{A}}}$ maximisant

l'équation (6.3.5). On note que la matrice $\boldsymbol{\mathcal{V}}$ n'est pas carrée, elle n'est donc pas inversible. Dans ce cas, on exprime la matrice $\boldsymbol{\mathcal{A}}$ à l'aide de la matrice pseudo-inverse de $\boldsymbol{\mathcal{V}}$ tel que $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{V}}^{+}\widetilde{\boldsymbol{\mathcal{A}}}$, où $\boldsymbol{\mathcal{V}}^{+}$ est la matrice inverse de Moore-Penrose.

7.4. Simulations

Dans cette section, nous allons inspecter par simulation la performance de la nouvelle méthode d'ACI que nous avons proposée, soit la méthode basée sur le LASSO par groupe adaptatif. La procédure de simulation des jeux de données est sensiblement la même que celle proposée par Zhang et Chan (2006), à quelques exceptions près.

Simulation #1

Tout d'abord, nous appliquons la méthode proposée sur un petit jeu de données afin de constater qu'elle simplifie grandement l'interprétation du modèle en réduisant les coefficients groupés non significatifs vers 0. Dans cette simulation, on fixe $p = 5$ sources, $q = 20$ voxels et $n = 200$ observations par voxel. Ceci implique que la matrice de mélange $\boldsymbol{\mathcal{A}}$ est de taille 20×5. Chaque colonne de $\boldsymbol{\mathcal{A}}$ est divisée en $m = 5$ groupes, chacun de taille $p_i = 4, i = 1, \ldots, m$. La matrice $\boldsymbol{\mathcal{A}}$ a été fixée de la façon suivante : tous les coefficients de certains groupes choisis aléatoirement sont mis à 0 alors que les coefficients des autres groupes sont générés à partir d'une loi Uniforme$(0.1, 1)$. Les sources $s_j, j = 1, \ldots, 5$, sont simulées à partir de lois gaussiennes que nous avons élevées à une puissance se situant aléatoirement entre 1.5 et 2. La variance de chacune des sources est choisie aléatoirement entre 0.2 et 1. Enfin, les observations sont générées à partir du modèle $\mathbf{x} = \boldsymbol{\mathcal{A}}\mathbf{s}$. Le but est principalement d'estimer la matrice de mélange $\boldsymbol{\mathcal{A}}$ à partir de $\mathbf{x}$ seulement.

Les résultats sont affichés à la table 7.1, où $\widehat{\boldsymbol{\mathcal{A}}}_{\text{MV}}$ représente l'estimation du maximum de vraisemblance et où $\widehat{\boldsymbol{\mathcal{A}}}_{\text{LGA}}$ représente l'estimation du maximum de la vraisemblance pénalisée par le LASSO par groupe adaptatif. Les colonnes de chaque matrice ont d'abord été divisées par le maximum absolu de la colonne correspondante. Puis, les colonnes des matrices de $\widehat{\boldsymbol{\mathcal{A}}}_{\text{MV}}$ et de $\widehat{\boldsymbol{\mathcal{A}}}_{\text{LGA}}$ ont été permutées de façon à correspondre aux colonnes de la matrice $\boldsymbol{\mathcal{A}}$. On remarque que la matrice $\widehat{\boldsymbol{\mathcal{A}}}_{\text{LGA}}$ est quasi identique à $\boldsymbol{\mathcal{A}}$. En effet, les coefficients groupés non significatifs ont bel et bien été réduits à 0, excepté certains coefficients d'amplitude 0.01 dans le deuxième groupe de la deuxième colonne. Même les coefficients du dernier groupe de la 5^{e} colonne ont été mis à 0, dans lequel un des coefficients avait une amplitude de 0.140 dans la matrice $\widehat{\boldsymbol{\mathcal{A}}}_{\text{MV}}$.

TABLE 7.1. Résultats de la simulation #1. À gauche, on retrouve la matrice de mélange, au centre, l'estimation par maximum de vraisemblance (MV) et à droite, l'estimation par maximum de la vraisemblance pénalisée par le LASSO par groupe adaptatif (LGA).

$\mathcal{A}$					$\hat{\mathcal{A}}_{MV}$					$\hat{\mathcal{A}}_{LGA}$				
0.950	0.000	0.000	0.000	0.549	0.949	-0.001	-0.042	-0.115	0.513	0.953	0.000	0.000	0.000	0.549
0.624	0.000	0.000	0.000	0.301	0.624	0.000	-0.027	-0.073	0.283	0.626	0.000	0.000	0.000	0.301
0.908	0.000	0.000	0.000	0.712	0.906	-0.006	-0.042	-0.117	0.664	0.911	0.000	0.000	0.000	0.712
0.322	0.000	0.000	0.000	0.740	0.319	-0.014	-0.019	-0.061	0.685	0.323	0.000	0.000	0.000	0.740
0.289	0.000	0.475	0.337	0.170	0.296	0.010	0.485	0.316	0.174	0.296	0.000	0.478	0.337	0.151
0.971	0.000	0.272	0.996	0.914	0.990	0.012	0.244	0.944	0.852	0.995	-0.001	0.277	0.996	0.860
0.817	0.000	0.968	0.708	0.503	0.830	0.021	0.978	0.638	0.501	0.832	0.001	0.973	0.708	0.465
0.253	0.000	0.956	1.000	0.805	0.271	0.012	0.988	1.000	0.772	0.273	0.000	0.963	1.000	0.750
0.000	0.000	0.000	0.000	0.505	-0.003	-0.012	-0.004	-0.020	0.466	0.000	0.000	0.000	0.000	0.505
0.000	0.000	0.000	0.000	0.931	-0.005	-0.022	-0.008	-0.037	0.859	0.000	0.000	0.000	0.000	0.931
0.000	0.000	0.000	0.000	1.000	-0.005	-0.024	-0.009	-0.039	0.922	0.000	0.000	0.000	0.000	1.000
0.000	0.000	0.000	0.000	0.824	-0.004	-0.020	-0.007	-0.032	0.760	0.000	0.000	0.000	0.000	0.824
0.813	0.536	0.314	0.139	0.865	0.812	0.521	0.287	0.031	0.885	0.817	0.536	0.312	0.139	0.858
0.725	0.360	0.859	0.933	0.365	0.744	0.380	0.870	0.904	0.413	0.744	0.360	0.864	0.933	0.314
0.675	1.000	0.456	0.420	0.107	0.684	1.000	0.448	0.379	0.248	0.683	1.000	0.453	0.420	0.084
0.883	0.584	1.000	0.232	0.955	0.882	0.575	1.000	0.099	1.000	0.886	0.585	1.000	0.232	0.943
1.000	0.380	0.821	0.000	0.000	1.000	0.393	0.816	-0.123	0.088	1.000	0.382	0.821	0.000	0.000
0.729	0.893	0.486	0.000	0.000	0.729	0.890	0.475	-0.081	0.140	0.729	0.893	0.482	0.000	0.000
0.768	0.333	0.180	0.000	0.000	0.769	0.338	0.156	-0.079	0.056	0.770	0.333	0.178	0.000	0.000
0.254	0.217	0.937	0.000	0.000	0.252	0.224	0.967	-0.055	0.066	0.251	0.219	0.938	0.000	0.000

Simulation #2

La méthode proposée est maintenant effectuée sur un jeu de données plus représentatif de données pouvant provenir d'une expérience par IRM. Dans cette simulation, on fixe $p = 10$ sources, $q = 1000$ voxels et $n = 500$ observations par voxels. De plus, chaque colonne de $\mathcal{A}$ est divisée en $m = 100$ groupes, chacun de taille $p_i = 10, i = 1, \ldots, m$. La matrice $\mathcal{A}$ est simulée une seule fois selon la procédure de la simulation précédente. Puis, les sources s_j sont eux aussi simulées selon la procédure de la simulation précédente et cette procédure est reproduite lors de $B = 100$ essais différents. À chaque essai, on mesure l'erreur Frobenius au carré entre les sources et les composantes indépendantes de même que l'erreur Amari. L'erreur quadratique moyenne entre les sources et les composantes indépendantes est définie de la façon suivante :

$$\text{EQM}(\mathbf{Y};\mathbf{S}) = \mathbb{E}\left\{\left(d_F\left(\mathbf{Y};\mathbf{S}\right)\right)^2\right\} = \mathbb{E}\left\{\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\mathbf{Y}_{ij} - \mathbf{S}_{ij}\right)^2\right\}, \tag{7.4.1}$$

où $d_F\left(\mathbf{Y};\mathbf{S}\right)$ représente l'erreur Frobenius entre les matrices $\mathbf{Y}$ et $\mathbf{S}$. En pratique, l'erreur quadratique moyenne est calculée comme suit :

$$\widehat{\mathrm{EQM}}(\mathbf{Y};\mathbf{S}) = \frac{1}{B}\sum_{b=1}^{B}\left(d_F^{(b)}\left(\boldsymbol{\mathcal{Y}}^{(b)};\boldsymbol{\mathcal{S}}^{(b)}\right)\right)^2 = \frac{1}{B}\sum_{b=1}^{B}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\boldsymbol{\mathcal{Y}}_{ij}^{(b)} - \boldsymbol{\mathcal{S}}_{ij}^{(b)}\right)^2, \tag{7.4.2}$$

où $d_F^{(b)}\left(\boldsymbol{\mathcal{Y}}^{(b)};\boldsymbol{\mathcal{S}}^{(b)}\right)$ représente l'erreur Frobenius du $b^{\text{ième}}$ essai calculée entre les matrices $\boldsymbol{\mathcal{Y}}^{(b)}$ et $\boldsymbol{\mathcal{S}}^{(b)}$. Note : les composantes indépendantes sont d'abord permutées de façon à correspondre aux sources, puis les sources et les composantes indépendantes sont transformées de façon à ce qu'elles soient comprises dans l'intervalle $[-1, 1]$. L'erreur Amari (EA) a été proposée par Amari et coll. (1996) et se calcule à l'aide de l'équation suivante :

$$\mathrm{EA}(\hat{\boldsymbol{\mathcal{W}}};\widetilde{\boldsymbol{\mathcal{A}}}) = \frac{1}{2p(p-1)}\left[\sum_{i=1}^{p}\left(\sum_{j=1}^{p}\frac{|r_{ij}|}{\max_k|r_{ik}|} - 1\right) + \sum_{j=1}^{p}\left(\sum_{i=1}^{p}\frac{|r_{ij}|}{\max_k|r_{kj}|} - 1\right)\right], \tag{7.4.3}$$

où $\hat{\boldsymbol{\mathcal{W}}}$ est l'estimation de $\widetilde{\boldsymbol{\mathcal{W}}} = \widetilde{\boldsymbol{\mathcal{A}}}^{-1}$ et où r_{ij} est l'élément (i, j) de la matrice $\boldsymbol{\mathcal{R}} = \hat{\boldsymbol{\mathcal{W}}}\widetilde{\boldsymbol{\mathcal{A}}}$. Avant de mesurer l'erreur Amari, on normalise d'abord chaque ligne des matrices $\hat{\boldsymbol{\mathcal{W}}}$ et $\widetilde{\boldsymbol{\mathcal{A}}}$ puis, on s'assure que la plus grande valeur absolue de chaque ligne soit de signe positif. L'erreur Amari est une sorte de distance entre la matrice $\boldsymbol{\mathcal{R}}$ et la matrice identité. Plus l'erreur Amari est petite, plus l'estimation de la matrice $\widetilde{\boldsymbol{\mathcal{W}}}$ est précise.

Lors de la simulation #2, nous avons pu constater que l'algorithme du maximum de vraisemblance n'a pas convergé 3 fois parmi les 100 essais, car la vraisemblance évaluée en $\hat{\boldsymbol{\mathcal{A}}}$ était moins élevée que la vraisemblance évaluée en $\boldsymbol{\mathcal{A}}$. Ces essais n'ont donc pas été considérés dans les résultats qui sont affichés à la figure 7.1 et à la table 7.2. La figure 7.1 montre les boxplots représentant les erreurs Frobenius au carré et les erreurs Amari du maximum de vraisemblance, du LASSO adaptatif et du LASSO par groupe adaptatif. On remarque que les erreurs Frobenius au carré ainsi que les erreurs Amari de la méthode proposée sont nettement inférieures à celles du maximum de vraisemblance et du LASSO adaptatif. En effet, on voit bien à la table 7.2 que les EQM calculées selon les différentes méthodes d'ACI sont significativement différentes puisque les intervalles de confiance ne se croisent pas. Donc, on peut conclure que la méthode proposée améliore significativement la précision de l'estimation des sources lorsque la matrice de mélange est éparse par groupe. De la même façon, on constate que la moyenne des erreurs Amari du LASSO par groupe adaptatif est significativement inférieure à la

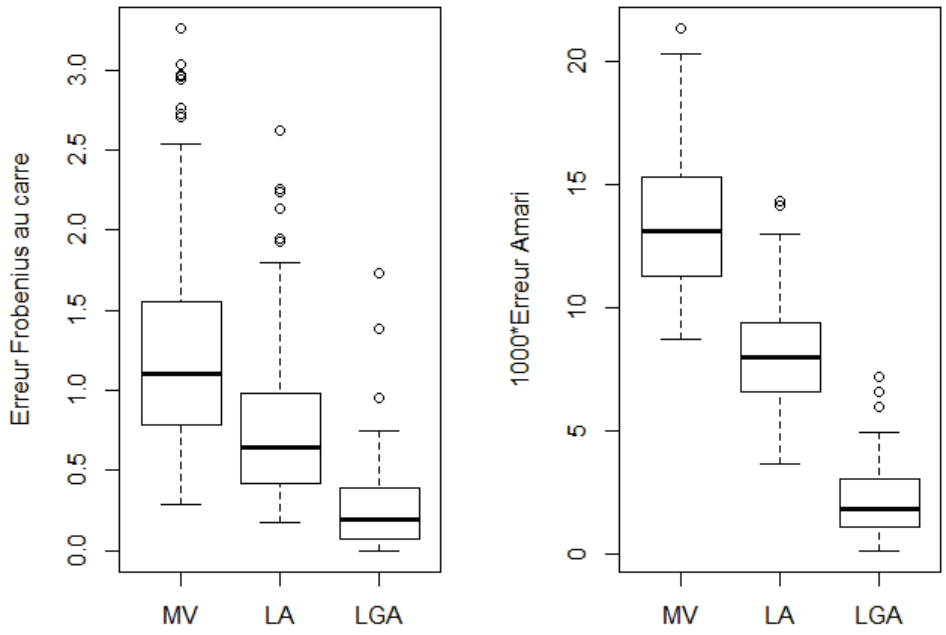

FIGURE 7.1. Résultats de la simulation #2. Boxplots représentant les erreurs Frobenius au carré et les erreurs Amari du maximum de vraisemblance (MV), du LASSO adaptatif (LA) et du LASSO par groupe adaptatif (LGA).

TABLE 7.2. Résultats de la simulation #2. Intervalles de confiance 95% de l'EQM et de la vraie moyenne des erreurs Amari×1000 du maximum de vraisemblance (MV), du LASSO adaptatif (LA) et du LASSO par groupe adaptatif (LGA).

	Moyenne	Erreur standard	$IC_{95\%}$
$d_F(\boldsymbol{\mathcal{Y}}_{MV};\boldsymbol{\mathcal{S}})^2$	1.286	0.071	[1.148, 1.425]
$d_F(\boldsymbol{\mathcal{Y}}_{LA};\boldsymbol{\mathcal{S}})^2$	0.772	0.052	[0.671, 0.873]
$d_F(\boldsymbol{\mathcal{Y}}_{LGA};\boldsymbol{\mathcal{S}})^2$	0.282	0.029	[0.225, 0.339]
$EA(\hat{\boldsymbol{\mathcal{W}}}_{MV};\widetilde{\boldsymbol{\mathcal{A}}})$	13.289	0.258	[12.784, 13.794]
$EA(\hat{\boldsymbol{\mathcal{W}}}_{LA};\widetilde{\boldsymbol{\mathcal{A}}})$	8.134	0.217	[7.708, 8.559]
$EA(\hat{\boldsymbol{\mathcal{W}}}_{LGA};\widetilde{\boldsymbol{\mathcal{A}}})$	2.122	0.142	[1.849, 2.400]

moyenne des erreurs Amari du maximum de vraisemblance et du LASSO adaptatif. Donc, on peut conclure que la méthode proposée améliore significativement la précision de la matrice de mélange carrée du modèle d'ACI lorsque la matrice de mélange est éparse par groupe.

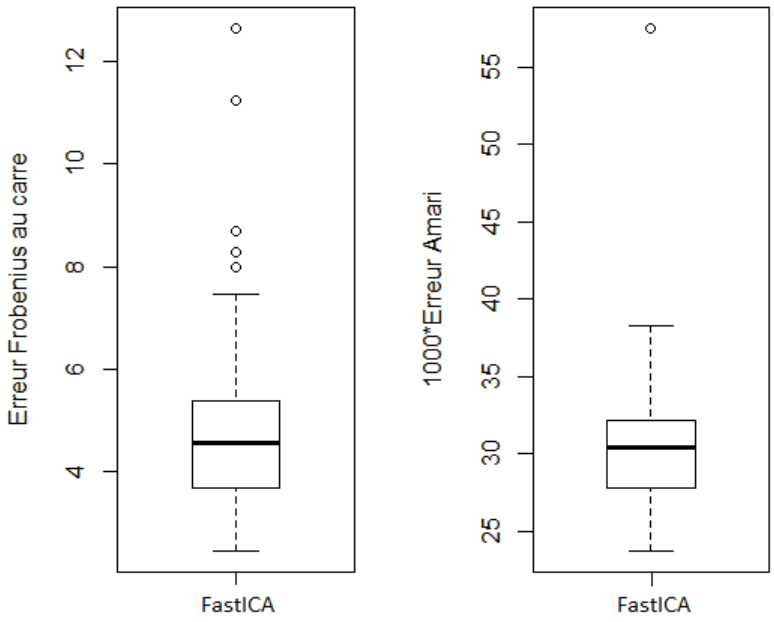

FIGURE 7.2. Résultats de la simulation #2. Boxplots représentant les erreurs Frobenius au carré et les erreurs Amari de l'algorithme FastICA. Une erreur Frobenius de 59.16 n'a pas été affichée sur le boxplot de gauche.

Les modèles d'ACI générés lors de la simulation #2 ont aussi été résolus à l'aide de l'algorithme FastICA. Cet algorithme est très populaire en ACI et se retrouve notamment dans Hyvärinen et coll. (2001). Cet algorithme maximise une des mesures de non gaussianité mentionnées à l'équation (2.7.4). Dans notre cas, nous avons utilisé la première fonction avec une orthogonalisation parallèle. À titre informatif, la médiane des erreurs Frobenius au carré et des erreurs Amari$\times$1000 obtenues par l'algorithme FastICA sont respectivement de 4.61 et de 30.44. Les boxplots de ces erreurs sont montrés à la figure 7.2. On remarque que les méthodes d'ACI utilisées à la figure 7.1 sont nettement plus efficaces que l'algorithme FastICA si on se fie à l'échelle des boxplots.

Chapitre 8

CONCLUSION

Dans ce mémoire, nous avons revu les notions de base de l'analyse en composantes indépendantes, notamment la théorie de l'information, l'analyse en composantes principales et le blanchiment des données. Nous avons décrit et expliqué les méthodes d'ACI classiques telles que l'ACI par maximisation de la non normalité, l'ACI par maximum de vraisemblance ainsi que l'ACI par minimisation de l'information mutuelle. Ensuite, nous avons vu certaines pénalisations dans un contexte de régression linéaire dont certaines ont été appliquées dans un contexte d'analyse en composantes indépendantes avec une matrice de mélange éparse.

Nous avons proposé une nouvelle méthode basée sur le LASSO par groupe adaptatif afin de résoudre le problème d'analyse en composantes indépendantes avec une matrice de mélange éparse par groupe. Nous avons expliqué que cette nouvelle méthode peut avoir des applications en imagerie cérébrale, plus précisément en imagerie par résonance magnétique. Nous avons démontré par une simulation que l'ACI par maximum de la vraisemblance pénalisée par le LASSO par groupe adaptatif simplifie grandement l'interprétation du modèle en réduisant vers zéro les groupes de coefficients non-significatifs au sein de la matrice de mélange. De plus, nous avons montré que la précision de la méthode proposée est nettement supérieure à celle du maximum de la vraisemblance pénalisée par le LASSO adaptatif.

Au cours de la simulation #2 de ce mémoire, un des problèmes que nous avons rencontré est la non convergence de certains essais. Afin de résoudre ce problème, il serait intéressant de considérer des algorithmes plus sophistiqués lors de futures recherches.

Dans de futurs travaux, on pourrait envisager une pénalisation LASSO par groupe adaptatif dans laquelle la pénalisation se fait aussi au sein des groupes. Ce serait un mélange du LASSO adaptatif et du LASSO par groupe adaptatif.

Il serait aussi intéressant de considérer une régression linéaire dont la distribution des erreurs est non gaussienne. Dans ce cas, on pourrait envisager une régression linéaire robuste en maximisant une approximation robuste de la néguentropie des résidus.

Finalement, il serait bien d'essayer la méthode d'ACI proposée sur un jeu de données réelles provenant d'une expérience d'IRM. Dans le cas où l'on essaie de retrouver le stimulus de l'expérience, on s'attend à ce que celui-ci soit bien visible dans quelques structures cérébrales et qu'il soit invisible dans les autres structures du cerveau.

Bibliographie

Amari, S.-i., Cichocki, A., Yang, H. H. et coll. (1996). A new learning algorithm for blind signal separation. *Advances in neural information processing systems*, 757–763.

Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *The annals of statistics*, **24**, 2350–2383.

Bronstein, A. M., Bronstein, M. M., Zibulevsky, M. et Zeevi, Y. Y. (2005). Sparse ica for blind separation of transmitted and reflected images. *International Journal of Imaging Systems and Technology*, **15**, 84–91.

Cardoso, J.-F. (2003). Dependence, correlation and gaussianity in independent component analysis. *The Journal of Machine Learning Research*, **4**, 1177–1203.

Chen, A. (2006). Fast kernel density independent component analysis. *Independent Component Analysis and Blind Signal Separation*, 24–31.

Coleman, J. O. (2000). Generalize higher-order moments in independent component analysis. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 1, 153–156. IEEE.

Comon, P. (1994). Independent component analysis, a new concept ? *Signal processing*, **36**, 287–314.

Cover, T., Thomas, J., Proakis, J. G., Salehi, M. et Morelos-Zaragoza, R. H. (1991). *Elements of information theory. telecommunications*. Wiley series.

Cramér, H. (1937). *Random variables and probability distributions*. Cambridge University Press.

Darmois, G. (1953). Analyse générale des liaisons stochastiques : etude particulière de l'analyse factorielle linéaire. *Revue de l'Institut international de statistique*, 2–8.

Dugué, D. (1951). Analycité et convexité des fonctions caractéristiques. In *Annales de l'institut Henri Poincaré*, vol. 12, 45–56. Presses universitaires de France.

Fan, J. et Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, **96**,

1348–1360.

Granger, C. W. (1976). Tendency towards normality of linear combinations of random variables. *Metrika*, **23**, 237–248.

Hyvärinen, A., Karhunen, J. et Oja, E. (2001). *Independent component analysis*, vol. 26. Wiley-interscience.

Shimizu, S., Hoyer, P. O., Hyvärinen, A. et Kerminen, A. (2006). A linear non-gaussian acyclic model for causal discovery. *The Journal of Machine Learning Research*, **7**, 2003–2030.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

Wang, H. et Leng, C. (2008). A note on adaptive group lasso. *Computational Statistics & Data Analysis*, **52**, 5277–5286.

Yuan, M. et Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, **68**, 49–67.

Zhang, K. et Chan, L.-W. (2006). Ica with sparse connections. *Intelligent Data Engineering and Automated Learning–IDEAL 2006*, 530–537.

Zhang, K., Peng, H., Chan, L. et Hyvärinen, A. (2009). Ica with sparse connections : Revisited. *Independent Component Analysis and Signal Separation*, 195–202.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, **101**, 1418–1429.

Zou, H. et Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, **67**, 301–320.

Zou, H. et Zhang, H. H. (2009). On the adaptive elastic-net with a diverging number of parameters. *Annals of statistics*, **37**, 1733.

Annexe A

CODES *R*

A.1. Fonctions utiles

A.1.1. Fonction *ordre*

La fonction *ordre* trouve la permutation et le signe des composantes indépendantes correspondant le mieux aux sources.

```
ordre <- function(p, y, s){
distance <- matrix( rep(0,2*p^2), nrow=p )
ymoins <- -y

#Transforme les variables en 0-1
for( i in 1:p )
{
s[i,] <- s[i,]*2/(max(s[i,])-min(s[i,]))
s[i,] <- s[i,]-max(s[i,])+1
y[i,] <- y[i,]*2/(max(y[i,])-min(y[i,]))
y[i,] <- y[i,]-max(y[i,])+1
ymoins[i,] <- ymoins[i,]*2/(max(ymoins[i,])-min(ymoins[i,]))
ymoins[i,] <- ymoins[i,]-max(ymoins[i,])+1
}

#On calcule toutes les distances possibles
for( i in 1:p){
 for( j in 1:p ){
 distance[i,j] <- sum(abs(s[i,] - y[j,]))
 distance[i,j+p] <- sum(abs(s[i,] - ymoins[j,]))
}}

p.indice <- 1:p
indice <- 1:(2*p)
position <- matrix( rep(0,2*p), nrow= p )

for( k in 1:p ){
#On regarde la mesure qui correspond a la plus petite des mesures restantes
```

```
29 for( i in p.indice ) {
   for( j in indice ) {
31   if( distance[i, j] == min(distance[i,indice]) )
     {
33   if( j <= p ) position[i,] <- c(j,1)
     if ( j > p ) position[i,] <- c(j-p, -1)
35   }}}

37 #On enleve l'indice qui correspond a la plus petite
   miny <- rep(0,2*p)
39 minTot <- min(distance[p.indice,indice])
   for( i in 1:p ){
41 if(min(distance[i,indice])==minTot) p.indice <- p.indice[p.indice!=i]
   }
43 for( i in indice ) {
   if(min(distance[,i])==minTot){ indice <- indice[indice!=i]
45 if( i <=p ) {
    indice <- indice[indice!=(i+p)]
47  } else {
   indice <- indice[indice!=(i-p)]
49 }}}}

51 #On change l'ordre du y initial et on multiplie par le signe correspondant
   y.res <- y[position[,1],]
53 for( i in 1:p) {
   y.res[i,] <- y.res[i,] * position[i,2]
55 }
   return(list(y=y.res, s=s,position=position))
57 }
```

A.1.2. Fonction *MSE*

La fonction *MSE* retourne l'erreur Frobenius entre les composantes indépendantes et les sources.

```
1 MSE <- function(p, x, W, s)
  {
3 y <- W %*% x
  y.res <- ordre(p,y,s)$y
5 s.res <- ordre(p,y,s)$s
  MSE <- sum((y.res - s.res)^2)
7 return(MSE)
  }
```

A.1.3. Fonction *mesure*

La fonction *mesure* standardise les matrices $\boldsymbol{\mathcal{A}}$ et $\hat{\boldsymbol{\mathcal{A}}}$, puis permute et ajuste le signe des colonnes de $\hat{\boldsymbol{\mathcal{A}}}$ de façon à correspondre à $\boldsymbol{\mathcal{A}}$.

```
mesure <- function(p, A, Achapeau, position, group){
for( i in 1:p ){
A[,i] <- A[,i] / max(abs(A[,i]))
Achapeau[,i] <- Achapeau[,i] / max(abs(Achapeau[,i]))
}
Achapeau <- Achapeau[,position[,1]]
for( i in 1:p ){
Achapeau[,i] <- Achapeau[,i] * position[i,2]
}
return(list(A=A, Achapeau=Achapeau))
}
```

A.1.4. Fonction *vraisemblance*

La fonction *vraisemblance* retourne la valeur de la log-vraisemblance de l'équation (4.2.4) évaluée en une certaine matrice de séparation $\widetilde{\mathcal{W}}$.

```

vraisemblance <- function(n, p, x, Wstar){
W <- matrix(Wstar,p)
y <- W %*% x
for( i in 1:p )
{
y[i,] <- sort(y[i,])
}
h <- rep(0,p)
for( i in 1:p )
{
h[i] <- 0.6*(var(y[i,]))^(1/2)*n^(-1/5)
}
tmoins <- matrix(rep(0,n*p), nrow=p)
tplus <- matrix(rep(0,n*p), nrow=p)

for( i in 1:p ){
tmoins[i,1] <- exp(y[i,1]/h[i])
tplus[i,n] <- 0
for( j in 2:n ){
tmoins[i,j] <- tmoins[i,j-1] + exp( y[i,j]/h[i] )
tplus[i,n-j+1] <- tplus[i,n-j+2] + exp( -y[i,n-j+2]/h[i] )
}}

logp <- matrix(rep(0,p*n),p)
for ( i in 1:p ){
for( j in 1:n ){
logp[i,j] <- log( 1/(2*n*h[i])*( tmoins[i,j]* exp(- y[i,j]/h[i]) +
tplus[i,j]*exp(y[i,j]/h[i]) ) )
}}
vary <- rep(0,p)
for( i in 1:p ){
```

```
33 vary[i] <- 1/n*sum(y[i,]^2)
   }
35 vraisemblance <- sum(logp) + n*log(abs(det(W))) - sum((vary - 1)^2)
   return(-vraisemblance)
37 }
```

A.1.5. Fonction *vrai.deriv*

La fonction *vrai.deriv* retourne la dérivée de la fonction *vraisemblance* par rapport à $\widetilde{\boldsymbol{\mathcal{W}}}$.

```
1
   vrai.deriv <- function(n,p,x,W){
3 W <- matrix(W,p)
   y <- W %*% x
5 ordre <- matrix( rep(0,n*p),p)
   for( i in 1:p ){
7 ordre[i,] <- order(y[i,])
   }
9 for( i in 1:p ) y[i,] <- sort(y[i,])
   h <- rep(0,p)
11 for( i in 1:p ) h[i] <- 0.6*(var(y[i,]))^(1/2)*n^(-1/5)
   h.deriv <- matrix(rep(0,p*p),p)
13 var.deriv <- matrix(rep(0,p*p),p)

15 for( i in 1:p ){
   for( m in 1:p ){
17 h.deriv[i,m] <- (0.6 * n^(-1/5) / 2 * var(y[i,])^(-1/2) * (2/(n-1) * sum(t(W[i,]) %*%
                 (x[,ordre[i,]])[,1:n]* (x[,ordre[i,]])[m,1:n])))
19 var.deriv[i,m] <- -4/n*(mean(y[i,]^2)-1)*sum(y[i,]*(x[,ordre[i,]])[m,])
   }}
21
   tmoins <- matrix(rep(0,n*p), nrow=p)
23 tplus <- matrix(rep(0,n*p), nrow=p)
   tmoins.deriv <- matrix(rep(0,n*p), nrow=p)
25 tplus.deriv <- matrix(rep(0,n*p), nrow=p)

27 for( i in 1:p ){
   tmoins[i,1] <- exp(y[i,1]/h[i])
29 tplus[i,n] <- 0
   for( j in 2:n ){
31 tmoins[i,j] <- tmoins[i,j-1] + exp( y[i,j]/h[i] )
   tplus[i,n-j+1] <- tplus[i,n-j+2] + exp( -y[i,n-j+2]/h[i] )
33 }}

35 vrai.deriv <- matrix(rep(0,p*p),p)
   for( i in 1:p ){
37 for( m in 1:p ){
      tmoins.deriv[i,1] <- exp(y[i,1]/h[i])*( (x[,ordre[i,]])[m,1]*h[i]
```

```
39      - y[i,1]*h.deriv[i,m])/h[i]^2
        tplus.deriv[i,n] <- 0
41      for( l in 2:n ) {
        tmoins.deriv[i,l] <- tmoins.deriv[i,l-1] + exp(y[i,l]/h[i])*
43      ( (x[,ordre[i,]])[m,l]*h[i] - y[i,l]*h.deriv[i,m] )/h[i]^2
        tplus.deriv[i,n-l+1] <- tplus.deriv[i,n-l+2] + exp(-y[i,n-l+2]/h[i])*
45      -( (x[,ordre[i,]])[m,n-l+2]*h[i] - y[i,n-l+2]*h.deriv[i,m])/h[i]^2
        }
47  for( j in 1:n ) {
    densiteprime <- 1/(2*n)*(h[i]*
49      ( tmoins.deriv[i,j]*exp(-y[i,j]/h[i]) +
        tmoins[i,j]*exp(-y[i,j]/h[i])*-( (x[,ordre[i,]])[m,j]*h[i]  -
51      y[i,j]*h.deriv[i,m] )/h[i]^2 + tplus.deriv[i,j]*exp(y[i,j]/h[i]) +
        tplus[i,j]*exp(y[i,j]/h[i])*( (x[,ordre[i,]])[m,j]*h[i]  -
53      y[i,j]*h.deriv[i,m])/h[i]^2) - h.deriv[i,m]* ( tmoins[i,j]* exp(-
        y[i,j]/h[i]) + tplus[i,j]*exp(y[i,j]/h[i]) ))/h[i]^2
55  psi <- densiteprime/(1/(2*n*h[i])*( tmoins[i,j]* exp(- y[i,j]/h[i]) +
          tplus[i,j]*exp(y[i,j]/h[i]) ))
57  vrai.deriv[i,m] <- vrai.deriv[i,m] + psi
    }
59  vrai.deriv[i,m] <- vrai.deriv[i,m] + var.deriv[i,m]
    }}
61  vrai.deriv <- vrai.deriv + n*solve(t(W))
    return(-vrai.deriv)
63  }
```

A.1.6. Fonction *ICA_vraisemblance*

La fonction *ICA_vraisemblance* maximise la log-vraisemblance par rapport à $\widetilde{\boldsymbol{\mathcal{W}}}$ puis retourne les estimations des matrices $\boldsymbol{\mathcal{A}}$ et $\widetilde{\boldsymbol{\mathcal{A}}}$.

```
1  ICA_vraisemblance <- function(n, p, q, x){
   X <- x
3  for( i in 1:q ) X[i,] <- x[i,] - mean(x[i,])
   mat <- 1/n*(X%*%t(X))
5  Ddemi <- diag(eigen(mat)$values^(-1/2))
   V <- (Ddemi %*% t(eigen(mat)$vectors))[1:p,]
7  svd <- svd(V)
   Vpseudo <- svd$v %*% diag(1/svd$d) %*% t(svd$u)
9  z <- V %*% X

11 f <- function(Wstar) vraisemblance(n, p, z, Wstar)
   gr2 <- function(W) vrai.deriv(n,p,z,W)
13 Wtilde <- matrix(optim(JADE(t(z),p,maxiter=1000)$W, f, method ="BFGS",
             gr = gr2, control=list(trace=1, maxit=1000))$par, nrow=p)
15 Achapeau <- Vpseudo %*% solve(Wtilde)
   return(list(Atilde=solve(Wtilde), A=Achapeau,z=z))
17 }
```

A.1.7. Fonction *vraisemblance_pen*

La fonction *vraisemblance_pen* retourne la vraisemblance pénalisée par le LASSO, le LASSO adaptatif ou bien le LASSO par groupe adaptatif évaluée en $\widetilde{\mathcal{A}}$.

```
1
  vraisemblance_pen <- function(n, p, q, z, Atilde, Vpseudo,
3                   pen="NA", Achapeau=NA, lambda1=0, group=NA)
  {
5 Atilde <- matrix(Atilde,p)
  A <- Vpseudo %*% Atilde
7 Wtilde <- solve(Atilde)
  y <- Wtilde %*% z
9 for( i in 1:p ) y[i,] <- sort(y[i,])
  h <- rep(0,p)
11 for( i in 1:p ) h[i] <- 0.6*(var(y[i,]))^(1/2)*n^(-1/5)

13 tmoins <- matrix(rep(0,n*p), nrow=p)
  tplus <- matrix(rep(0,n*p), nrow=p)
15 for( i in 1:p ){
  tmoins[i,1] <- exp(y[i,1]/h[i])
17 tplus[i,n] <- 0
  for( j in 2:n ){
19 tmoins[i,j] <- tmoins[i,j-1] + exp( y[i,j]/h[i] )
  tplus[i,n-j+1] <- tplus[i,n-j+2] + exp( -y[i,n-j+2]/h[i] )
21 }}

23 logp <- matrix(rep(0,p*n),p)
  for ( i in 1:p ){
25 for( j in 1:n ){
  logp[i,j] <- log( 1/(2*n*h[i])*( tmoins[i,j]* exp(- y[i,j]/h[i]) +
27              tplus[i,j]*exp(y[i,j]/h[i]) ) )
  }}
29
  vary <- rep(0,p)
31 for( i in 1:p ) vary[i] <- 1/n*sum(y[i,]^2)

33 if( pen == "NA") vraisemblance <- sum(logp) + n*log(abs(det(Wtilde))) -
                                    sum((vary - 1)^2)
35 if( pen == "lasso") vraisemblance <- sum(logp) + n*log(abs(det(Wtilde))) -
                                      lambda1*sum(abs(A)) - sum((vary - 1)^2)
37 if( pen == "lassoA") vraisemblance <- sum(logp) + n*log(abs(det(Wtilde))) -
                                  sum((vary - 1)^2) - lambda1*sum(abs(A/Achapeau))
39 if( pen == "groupLassoA" ) {
  group_pen <- 0
41 for( i in 1:length(group) ){
  group_pen <- group_pen + sqrt(sum(A[group[[i]]]^2))/sqrt(sum(Achapeau[group[[i]]]^2))^2
43 }
  vraisemblance <- sum(logp) + n*log(abs(det(Wtilde))) - sum((vary - 1)^2) - lambda1*group_pen
```

```
45 }
   return(-vraisemblance)
47 }
```

A.1.8. Fonction *vrai.deriv_pen*

La fonction *vrai.deriv_pen* retourne la dérivée de la fonction *vraisemblance_pen* par rapport à $\widetilde{\boldsymbol{\mathcal{A}}}$.

```
1  vrai.deriv_pen <- function(n, p, q, z, Atilde, V, Vpseudo, pen="NA",
                              Achapeau=matrix(runif(p*p),p), lambda1=0, group=NA)
3  {
   Atilde <- matrix(Atilde,p)
5  A <- Vpseudo %*% Atilde
   Wtilde <- solve(Atilde)
7  y2 <- Wtilde %*% z

9  y <- matrix( rep(0,n*p),p)
   ordery <- matrix( rep(0,n*p),p)
11 tmoins <- matrix(rep(0,n*p), nrow=p)
   tplus <- matrix(rep(0,n*p), nrow=p)
13 h <- rep(0,p)

15 for( i in 1:p ){
   ordery[i,] <- order(y2[i,])
17 y[i,] <- sort(y2[i,])
   h[i] <- 0.6*(var(y[i,]))^(1/2)*n^(-1/5)
19 tmoins[i,1] <- exp(y[i,1]/h[i])
   tplus[i,n] <- 0
21 for( j in 2:n ){
   tmoins[i,j] <- tmoins[i,j-1] + exp( y[i,j]/h[i] )
23 tplus[i,n-j+1] <- tplus[i,n-j+2] + exp( -y[i,n-j+2]/h[i] )
   }}
25
   vrai.deriv <- matrix(rep(0,p*p),p)
27 tmoins.deriv <- matrix(rep(0,n*p), nrow=p)
   tplus.deriv <- matrix(rep(0,n*p), nrow=p)
29 h.deriv <- rep(0,p)

31 for( k in 1:p ){
   for( m in 1:p ){
33 var.deriv <- rep(0,p)
   delta <- matrix(rep(0,p*p), p)
35 delta[k,m] <- 1
   y.der <- matrix(rep(0,p*n),p)
37 for( i in 1:p ){
   z2_i <- z[,ordery[i,]]
39 y.der[i,] <- (-Wtilde %*% delta %*% Wtilde %*% z2_i)[i,]
   var.deriv[i] <- mean(y[i,]^2-1)*sum(y[i,]*y.der[i,])
```

```
41 h.deriv[i] <- 0.6 * n^(-1/5) / (n-1) *(1/(n-1)*sum(y[i,]^2))^(-1/2)*
                (sum(y[i,]*y.der[i,]))
43 tmoins.deriv[i,1] <- exp(y[i,1]/h[i])*( h[i]*y.der[i,1] - y[i,1]*h.deriv[i] )/h[i]^2
   tplus.deriv[i,n] <- 0
45 for( l in 2:n ){
   tmoins.deriv[i,l] <- tmoins.deriv[i,l-1] + exp(y[i,l]/h[i])*( h[i]*y.der[i,l] -
47                      y[i,l]*h.deriv[i] )/h[i]^2
   tplus.deriv[i,n-l+1] <- tplus.deriv[i,n-l+2] + exp(-y[i,n-l+2]/h[i])*-
49                         ( h[i]*y.der[i, n-l+2] - y[i,n-l+2]*h.deriv[i])/h[i]^2
   }
51 for( j in 1:n ){
   densiteprime <- 1/(2*n)*(h[i]*
53    ( tmoins.deriv[i,j]*exp(-y[i,j]/h[i]) +
      tmoins[i,j]*exp(-y[i,j]/h[i])*-( y.der[i,j]*h[i]  - y[i,j]*h.deriv[i] )/h[i]^2 +
55    tplus.deriv[i,j]*exp(y[i,j]/h[i]) +
      tplus[i,j]*exp(y[i,j]/h[i])*( y.der[i,j]*h[i]  - y[i,j]*h.deriv[i])/h[i]^2) -
57    h.deriv[i]* ( tmoins[i,j]* exp(- y[i,j]/h[i]) + tplus[i,j]*exp(y[i,j]/h[i]) ))/h[i]^2
   psi <- densiteprime/(1/(2*n*h[i])*( tmoins[i,j]* exp(- y[i,j]/h[i]) + tplus[i,j]
59        *exp(y[i,j]/h[i]) ))
   vrai.deriv[k,m] <- vrai.deriv[k,m] + psi
61 }}
   vrai.deriv[k,m] <- vrai.deriv[k,m] - 4/n*sum(var.deriv)
63 }}

65 if( pen == "NA") vrai.deriv <- vrai.deriv - n*solve(t(Atilde))
   if( pen == "lasso"){
67 pen.deriv <- matrix(rep(0,p*p),p)
   for( k in 1:p ){
69 for( m in 1:p ){
   for( i in 1:q ){
71 pen.deriv[k,m] <- pen.deriv[k,m] + sign(A[i,m])*Vpseudo[i,k]
   }}}
73 vrai.deriv <- vrai.deriv - n*solve(t(Atilde)) - lambda1*pen.deriv
   }
75 if( pen == "lassoA"){
   pen.deriv <- matrix(rep(0,p*p),p)
77 for( k in 1:p ){
   for( m in 1:p ){
79 for( i in 1:q ){
   pen.deriv[k,m] <- pen.deriv[k,m] + sign(A[i,m])/abs(Achapeau[i,m])*Vpseudo[i,k]
81 }}}
   vrai.deriv <- vrai.deriv - n*solve(t(Atilde)) - lambda1*pen.deriv
83 }
   if( pen == "groupLassoA"){
85 group_pen.deriv <- matrix( rep(0,p*p),p)
   for( k in 1:p ){
87 for( m in 1:p ){
   for( i in 1:length(group) ){
89 for( j in group[[i]] ){
```

```
    mod <- j %% q
91  if( mod == 0 ) mod <- q
    if( ((j-mod)/q + 1) == m ){
93      if( length(group[[i]]) == 1 ){
        group_pen.deriv[k,m] <- group_pen.deriv[k,m] + sign(A[mod , (j-mod)/q + 1])*
95                              Vpseudo[mod,k]/sqrt(sum(Achapeau[group[[i]]]^2))^2
        } else {
97      group_pen.deriv[k,m] <- group_pen.deriv[k,m] + sum(A[group[[i]]]^2)^(-1/2)*
          A[mod , (j-mod)/q + 1]*Vpseudo[mod,k]/sqrt(sum(Achapeau[group[[i]]]^2))^2
99 }}}}}}
   vrai.deriv <- vrai.deriv - n*solve(t(Atilde)) - lambda1*group_pen.deriv
101 }
   return(-vrai.deriv)
103 }
```

A.1.9. Fonction *ICA_algo_pen*

Finalement, la fonction *ICA_algo_pen* maximise la vraisemblance pénalisée par rapport à $\widetilde{\boldsymbol{\mathcal{A}}}$ et retourne les matrices de mélange $\boldsymbol{\mathcal{A}}$ et $\widetilde{\boldsymbol{\mathcal{A}}}$.

```
 1 ICA_algo_pen <- function(n, p, q, x, pen="NA", Achapeau= 1, lambda1=0, group=NA){
   X <- x
 3 for( i in 1:q ) X[i,] <- x[i,] - mean(x[i,])
   mat <- 1/n*(X%*%t(X))
 5 eigen <- eigen(mat)
   Ddemi <- diag(eigen$values^(-1/2))
 7 V <- (Ddemi %*% t(eigen$vectors))[1:p,]
   svd <- svd(V)
 9 Vpseudo <- svd$v %*% diag(1/svd$d) %*% t(svd$u)
   z <- V %*% X
11 eta <- matrix(rep(1,p*p),p)
   u <- 1.1
13 d <- 0.5
   ite <- 0
15 coeff <- 1
   alpha <- 1/100
17 if( length(Achapeau)==1 ){ Astar <- diag(1,p)} else{Astar <- V %*% Achapeau}
   print(c(pen,"starting value", vraisemblance_pen(n, p, q, z, Astar, Vpseudo=Vpseudo,
19                              pen=pen, Achapeau=Achapeau, lambda1=lambda1, group=group)))
   grad1 <-  vrai.deriv_pen(n, p, q, z, Astar, V=V, Vpseudo=Vpseudo, pen=pen,
21                          Achapeau=Achapeau, lambda1=lambda1, group=group)/100
   AstarOld <- Astar
23 Astar <- Astar - alpha/10000000*grad1
   gradOld <- grad1
25 tol <- 1
   ite <- ite + 1
27 print(valeur <- vraisemblance_pen(n, p, q, z, Astar, Vpseudo=Vpseudo,
                    pen=pen, Achapeau=Achapeau, lambda1=lambda1, group=group))
29
```

```
  while(tol > 0.000005 && ite < 1000){
31 grad1 <-  vrai.deriv_pen(n, p, q, z, Astar, V=V, Vpseudo=Vpseudo,
                 pen=pen, Achapeau=Achapeau, lambda1=lambda1, group=group)
33 for( i in 1:p ){
  for( j in 1:p ){
35 eta[i,j] <- (grad1[i,j]*gradOld[i,j] >0)*eta[i,j]*u + (grad1[i,j]
            *gradOld[i,j] <0 )*eta[i,j]*d
37 grad1[i,j] <- grad1[i,j]*eta[i,j]
  }}
39
  AstarOld <- Astar
41 Astar <- Astar - alpha*grad1
  gradOld <- grad1
43 valeurOld <- valeur
  valeur <- vraisemblance_pen(n, p, q, z, Astar, Vpseudo=Vpseudo, pen=pen,
45                          Achapeau=Achapeau, lambda1=lambda1, group=group)
  coeff2 <- 1
47
  while( (valeurOld > 0 && valeur > valeurOld*1.02) || (valeurOld <=0 &&
49                                              valeur > valeurOld*0.95) ){
  coeff2 <- coeff2/2
51 Astar <- AstarOld - alpha*grad1*coeff2
  valeur <- vraisemblance_pen(n, p, q, z, Astar, Vpseudo=Vpseudo,
53          pen=pen, Achapeau=Achapeau, lambda1=lambda1, group=group)
  valeur
55 }

57 if( ite > 30 ) coeff <- coeff*0.95
  tol <- sum(abs(AstarOld-Astar))*coeff
59 ite <- ite + 1
  if( ite < 10 ) tol <- 1
61 print(vraisemblance_pen(n, p, q, z, Astar, Vpseudo=Vpseudo,
    pen=pen, Achapeau=Achapeau, lambda1=lambda1, group=group))
63 }
  print(c("converged in", ite, "iterations"))
65 Astar2 <- Vpseudo %*% Astar
  return(list(Atilde=Astar, A=Astar2,z=z))
67 }
```

A.2. Simulations

A.2.1. Simulations # 1

Code *R* qui effectue la simulation # 1. La simulation ne sera pas la même, car il n'y a pas de *seed*. Cependant, les conclusions seront les mêmes.

```
  n <- 200
2 q <- 20
```

```
p <- 5
m <- 5
nb.group <- p*q/m

group <- vector("list", nb.group)
for( i in 1:nb.group ) group[[i]] <- seq(m*(i-1) + 1,m*(i-1)+m)
A <- matrix(rep(0,q*p),q)
for( i in 1:nb.group ) if( runif(1) < 0.6 ) A[group[[i]]] <- runif(m,0.1,1)

s <- matrix( rep(0,p*n), nrow=p)
for( i in 1:p ){
s[i,] <- rnorm(n,0,runif(1, 0.2,1))
s[i,] <- sign(s[i,])*abs(s[i,])^(runif(1,1.5,2))
}
x <- A %*% s
for( i in 1:q ) x[i,] <- x[i,] - mean(x[i,])

res1 <- ICA_vraisemblance(n, p, q, x)
Achapeau <- res1$A
Atilde <- res1$Atilde
Wtilde <- solve(Atilde)
MSE_Achapeautetest <- MSE(p,res1$z, Wtilde, s)

lambda1 <- 1/2*log(n)
res2 <- ICA_algo_pen(n, p, q, x, pen="groupLassoA", Achapeau= Achapeau,
        lambda1=lambda1, group=group)
A_groupLasso <- res2$A
Atilde_groupLasso <- res2$Atilde
Wtilde_groupLasso <- solve(Atilde_groupLasso)
MSE_AGLAtest <- MSE(p,res2$z, Wtilde_groupLasso, s)

A <- mesure(p, A, Achapeau, ordre(p,Wtilde%*%res1$z,s)$position)$A
Achapeau <- mesure(p, A, Achapeau, ordre(p,Wtilde%*%res1$z,s)$position)$Achapeau
A_groupLasso <- mesure(p, A, A_groupLasso, ordre(p,Wtilde_groupLasso%*%res1$z,s)$
                position)$Achapeau
```

A.2.2. Simulations # 2

Code *R* qui effectue la simulation # 2.

```
require(JADE)

simulation <- function(n,p,q,A,group,m,seed)
{
set.seed(seed)
s <- matrix( rep(0,p*n), nrow=p)
for( i in 1:p ){
s[i,] <- rnorm(n,0,runif(1, 0.2,1))
s[i,] <- sign(s[i,])*abs(s[i,])^(runif(1,1.5,2))
```

```
}
x <- A %*% s
for( i in 1:q ) x[i,] <- x[i,] - mean(x[i,])

res1 <- ICA_vraisemblance(n, p, q, x)
Achapeau <- res1$A
Atilde <- res1$Atilde
Wtilde <- solve(Atilde)
MSE_Achapeau <- MSE(p,res1$z, Wtilde, s)

lambda1 <- 1/2*log(n)
res2 <- ICA_algo_pen(n, p, q, x, pen="groupLassoA", Achapeau= Achapeau,
        lambda1=lambda1, group=group)
A_groupLasso <- res2$A
Atilde_groupLasso <- res2$Atilde
Wtilde_groupLasso <- solve(Atilde_groupLasso)
MSE_AGLA <- MSE(p,res2$z, Wtilde_groupLasso, s)

res3 <- ICA_algo_pen(n, p, q, x, pen="lassoA", Achapeau= Achapeau,
        lambda1=lambda1, group=group)
A_lassoA <- res3$A
Atilde_lassoA <- res3$Atilde
Wtilde_lassoA <- solve(Atilde_lassoA)
MSE_AlassoA <- MSE(p,res3$z, Wtilde_lassoA, s)

X <- x
for( i in 1:q ) X[i,] <- x[i,] - mean(x[i,])
mat <- 1/n*(X%*%t(X))
Ddemi <- diag(eigen(mat)$values^(-1/2))
V <- (Ddemi %*% t(eigen(mat)$vectors))[1:p,]
svd <- svd(V)
Vpseudo <- svd$v %*% diag(1/svd$d) %*% t(svd$u)
z <- V %*% X
AA <- V %*% A
WW <- solve(AA)
vraisemblanceAchapeau <- vraisemblance(n, p, res1$z, Wtilde)
vraisemblanceA <- vraisemblance(n, p, res1$z, WW)

fastICA.res <- fastICA(t(z), p)
MSE_fastICA <- MSE(p,z,t(fastICA.res$W)%*%t(fastICA.res$K), s)
amari_fastICA <- amari.error(t(fastICA.res$W)%*%t(fastICA.res$K), AA, standardize = T)

amari_vrai <- amari.error(Wtilde, AA, standardize = T)
amari_vrai.pen <- amari.error(Wtilde_groupLasso, AA, standardize = T)
amari_lassoA <- amari.error(Wtilde_lassoA, AA, standardize = T)

return(list(MSE_Achapeau = MSE_Achapeau, MSE_AGLA = MSE_AGLA, MSE_AlassoA = MSE_AlassoA,
vraisemblanceAchapeau = vraisemblanceAchapeau,vraisemblanceA = vraisemblanceA,
amari_vrai.pen=amari_vrai.pen, amari_vrai = amari_vrai, amari_lassoA = amari_lassoA,
```

```
   MSE_fastICA = MSE_fastICA, amari_fastICA = amari_fastICA))
60 }

62 set.seed(1)

64 n <- 500
   q <- 1000
66 p <- 10
   m <- 10
68 nb.group <- p*q/m

70 group <- vector("list", nb.group)
   for( i in 1:nb.group ) group[[i]] <- seq(m*(i-1) + 1,m*(i-1)+m)
72 A <- matrix(rep(0,q*p),q)
   for( i in 1:nb.group ) if( runif(1) < 0.6 ) A[group[[i]]] <- runif(m,0.1,1)
74
   repetitions <- 100
76
   MSE_Achapeau <- rep(0,repetitions)
78 MSE_AGLA <- rep(0,repetitions)
   MSE_AlassoA <- rep(0,repetitions)
80 MSE_fastICA <- rep(0,repetitions)
   vraisemblanceAchapeau <- rep(0,repetitions)
82 vraisemblanceA <- rep(0,repetitions)
   amari_vrai.pen <- rep(0,repetitions)
84 amari_vrai <- rep(0,repetitions)
   amari_lassoA <- rep(0,repetitions)
86 amari_fastICA <- rep(0,repetitions)
   seed <- 0
88
   for( i in 1:repetitions )
90 {
   resultat <- simulation(n,p,q,A,group,m, seed+i)
92 MSE_Achapeau[i] <- resultat$MSE_Achapeau
   MSE_AGLA[i] <- resultat$MSE_AGLA
94 MSE_AlassoA[i] <- resultat$MSE_AlassoA
   MSE_fastICA[i] <- resultat$MSE_fastICA
96 vraisemblanceAchapeau[i] <- resultat$vraisemblanceAchapeau
   vraisemblanceA[i] <- resultat$vraisemblanceA
98 amari_vrai.pen[i] <- resultat$amari_vrai.pen
   amari_vrai[i] <- resultat$amari_vrai
100 amari_lassoA[i] <- resultat$amari_lassoA
   amari_fastICA[i] <- resultat$amari_fastICA
102 }
```