الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

Université Mohamed Khider – Biskra

Faculté des Sciences et de la technologie

Département :…Génie électrique

Réf :……

جامعة محمد خيضر بسكرة
كلية العلوم و التكنولوجيا
قسم: الهندسة الكهربائية
المرجع:.....

Thèse présentée en vue de l'obtention
Du diplôme de

Doctorat en sciences en : Génie électrique.

Spécialité : Automatique

Utilisation des réseaux de neurones dans l'estimation et la prédiction des signaux.
Application à la séparation aveugle de sources

Présentée par :

Mostefa Mohamed TOUBA

Soutenue publiquement le ……

Devant le jury composé de :

Mohamed BOUMEHRAZ	**M.C.A**	**Président**	**Université de BISKRA**
Abdenacer TITAOUINE	**Professeur**	**Rapporteur**	**Université de BISKRA**
Abdelmalik TELEB-AHMED	**Professeur**	**Examinateur**	**Université de Valenciennes et du Hainaut Cambrésis- France**
Djamel SAÏGAA	**Professeur**	**Examinateur**	**Université de M'Sila**

Remerciements

Tout d'abord, je tiens à remercier très vivement Pr. TITAOUINE Abdenacer pour avoir accepté d'être rapporteur de cette thèse.

Je remercie très chaleureusement Pr. TITAOUINE Abdenacer Pr. Mellas MEKKI et Pr. Abdelmalik TALEB-AHMED pour leur soutien constant, leurs conseils et leurs encouragements qui m'ont permis de mener ce travail. Qu'ils trouvent ici l'expression de toute ma reconnaissance.

J'adresse mes sincères remerciements à Messieurs Mohamed BOUMEHRAZ, Djamel SAIGAA, et Ammar MEZAACHE pour l'intérêt qu'ils ont accordé à mon travail et pour avoir accepté de participer à ce jury.

Je voudrais remercier très chaleureusement Pr. Abdelmalik TALEB-AHMED qui fait preuve de sa gentillesse et patience, et pour son accueil distingué au sein du laboratoire LAMIH, université de valenciennes et du Hainaut Cambrésis-Valenciennes-France. Je voudrais également adresser un salut amical à mes amis avec qui j'ai partagé de bons moments.

Je termine par un grand merci à toute ma famille pour le soutien qu'elle m'a apporté tout au long de la préparation de cette thèse.

A ma famille

Résumé :

Dans cette thèse, nous proposons un nouvel algorithme de séparation aveugle de sources, basé sur l'optimisation de l'information mutuelle sous contraintes. Le problème d'optimisation sous contraintes est résolu par passage au problème dual.

L'estimateur proposé du gradient utilise l'estimation des densités de probabilité par maximum de vraisemblance est réseaux de neurones MLP pour des modèles de lois exponentielles choisis par minimisation du critère AIC. Ensuite, la méthode a été généralisée à l'ensemble des divergences entre densités de probabilité. Nous montrons que l'algorithme utilisant la modélisation neuronale de la loi de probabilité a de bonnes performances d'estimation des signaux sources.

Nous proposons aussi un algorithme de séparation aveugle de sources de mélange Post Non Linéaires (PNL) en utilisant un réseau de neurones multicouches. La procédure consiste à la fois à compenser les nonlinéarités du modèle PNL et d'estimer les sources tout en maximisant un critère d'entropie des signaux de sortie.

Nous illustrons les performances des algorithmes proposés pour des signaux simulés dans l'environnement MATLAB.

Mots-clés :

Réseaux de Neurones, Perceptron Multicouches, Séparation Aveugle de Sources, Analyse en Composantes Indépendantes, Information Mutuelle, Divergence de Kullback-Leibler.

Abstract :

In this thesis, we propose a new algorithm for blind source separation based on mutual information optimization under constraints. The constrained optimization problem is solved by passing to the dual problem.

The proposed gradient estimator uses density estimates by maximum likelihood and MLP neural networks on models of exponential laws chosen by minimizing the AIC criterion. Then, the method was generalized to all Divergences between probability densities. We show that the algorithm based on neural modeling of the probability densities have good performances in estimating the source signals.

We also propose an algorithm for blind source separation of Post Nonlinear mixtures using a multilayer neural network. The procedure consists in both compensating the nonlinearities in the PNL model and estimating the source signals while maximizing the entropy of the output signals.

We illustrate the performance of the proposed algorithms to simulated signals in MATLAB environment.

Keywords :

Neural Networks, Multi-layer Perceptron, Blind Source Separation, Independent Component Analysis, Mutual Information, Kullback-Leibler Divergence.

Table des matières

Notations et Abréviations

SAS : Séparation Aveugle de Sources

ACI : Analyse en Composantes Indépendantes

Pdf : Probability Density Function

$\boldsymbol{s}$: Vecteur source

$\boldsymbol{x}$: Vecteur mélange

$p_{\boldsymbol{s}}(\boldsymbol{s})$: Loi de probabilité d'un vecteur $\boldsymbol{s}$

$\boldsymbol{A}$: Matrice de mélange linéaire

$\boldsymbol{B}$: Matrice de séparation

$\boldsymbol{y}$: Vecteur des sources estimées

$E:$ Espérance mathématique

$\hat{E}$: Moyenne temporelle

$\boldsymbol{I}_n$: Matrice identité de dimension n

EVD : Eigen-Value Decomposition

$\| . \|$: Norme 2 d'un vecteur ou matrice

PNL : Post Non-Linéaire

$\boldsymbol{\mathcal{F}}$: Transformation nonlinéaire

N : Nombre d'échantillons

p : Nombre de sources

$\boldsymbol{G}$: Transformation inverse

i.i.d : indépendants et identiquement distribués

$\boldsymbol{P}$: Matrice de permutation

$\boldsymbol{D}$: Matrice d'échelle

$\boldsymbol{\mathcal{H}}$: Transformation non triviale

$\boldsymbol{\mathfrak{I}}$: Ensemble des transformations triviales

$\boldsymbol{\mathcal{M}}$: Ensemble particulier de transformations

EQM : Erreur quadratique moyenne

SNR : Rapport Signal sur Résidus

$kurt(.)$: Kurtosis

$H(.)$: Entropie

$ln(.)$: Logarithme népérien

IM : Information Mutuelle

$I(.)$: Information mutuelle

D_{KL} : Divergence de Kullback-Leibler

$\mathcal{L}(.)$: Fonction de vraisemblance

$det(.)$: Déterminant d'une matrice

$\psi(.)$: Fonction score marginale

MSF : Fonction score marginale

$\boldsymbol{\varphi}(.)$: Fonction score conjointe

JSF : fonction score conjointe

$\boldsymbol{\beta}(.)$: Différence des fonctions score

SFD : Différence des fonctions score

$k(.)$: Noyau

h : Paramètre de lissage

∇ : Opérateur gradient

$\sigma_{y_i}^2$: Variance de la variable aléatoire y_i

MLP : Multilayer Perceptron

$sgm(.)$: Fonction sigmoïde

d : Dimension du modèle exponentiel de pdf

Cdf : Fonction de répartition

$F_X(\quad)$: Fonction de répartition

$U[a,b]$: Loi uniforme sur l'intervalle [a,b]

$N(a,b)$: Loi normale de moyenne a et de variance b

sin : Fonction sinus

tri : Fonction triangle

$carré$: Fonction carré

$\boldsymbol{W_1}$: Matrice des poids de la couche de sortie du réseau MLP

$\boldsymbol{W_2}$: Matrice des poids de la couche d'entrée du réseau MLP

$\boldsymbol{g}(.)$: Fonction d'activation de la couche cachée

$\boldsymbol{\theta}$: Vecteur des biais de la couche cachée du réseau MLP

$\sigma(.)$: Fonction d'activation de la couche de sortie

η : Paramètre d'apprentissage du réseau MLP

$tanh(.)$: Fonction tangent hyperbolique

$signe$: Fonction signe

Liste des Figures

CHAPITRE 1

Introduction Générale

1.1. INTRODUCTION

La SAS est un problème général en traitement du signal, dont le principe consiste à retrouver un ensemble de signaux inobservables dits « signaux sources » à partir d'un ensemble de signaux observables dits « observations ». Ces observations sont souvent des mélanges de ces sources et proviennent de capteurs comme par exemple des microphones, sondes, accéléromètres, antennes, caméras, ... etc. Nous pouvons observer sur chaque capteur, la sortie d'un système réalisant le « mélange » des signaux sources. La nature du mélange et le milieu de propagation de ces sources sont généralement inconnus. Aucune information n'est donc disponible sur les sources ni sur les mélanges. Vu ces

ambigüités, il est difficile voire impossible de retrouver les sources sans faire quelques hypothèses

Dans cette thèse le thème abordé vise la puissance des réseaux neuronaux (voir [B1] pour plus de détails) qui réside dans le fait qu'ils peuvent être utilisés pour déduire une fonction à partir d'observations. L'apprentissage dans les réseaux de neurones est particulièrement utile dans les applications où la complexité des données ou des tâches rend la conception de ces fonctions manuellement non pratique. On peut classer les tâches des réseaux de neurones en 3 grandes catégories :

1. Approximation des fonctions, Analyse de régression y compris la prédiction et modélisation des séries temporelles.
2. Classification, qui contient principalement la reconnaissance des formes et des séquences.
3. Traitement des données, Filtrage, Clustering, et **séparation aveugle de sources.**

Ces 3 grandes catégories comprennent les applications telles que : Identification et contrôle de systèmes dynamiques, les systèmes radar, authentification de visages, reconnaissance vocale, reconnaissance de caractères, diagnostic médical, analyse des données financières, …etc.

1.2. ETAT DE l'ART

La Séparation Aveugle de Sources, **SAS** (**BSS** : Blind Source Separation), se réfère à une large classe de méthodes de traitement du signal et d'image dont le but est d'extraire des sources superposées d'un ensemble de mélanges sans presque aucune connaissance préalable ni sur les signaux sources ni sur l'ensemble

de mélange. Dans des applications biomédicales, la SAS est utilisée pour l'analyse des signaux électroencéphalographiques (EEG), magnéto-encéphalographiques (MEG), électrocardiographiques (ECG) et des signaux IRMf(Imagerie par résonnance Magnétique Fonctionnelle).

Ce problème se rencontre aussi dans plusieurs autres applications [L6, L7, L26, L34] : radio - communication, séparation des signaux sismiques, monitorage des réacteurs nucléaires, surveillance des aéroports, rehaussement de la parole …etc.

La Séparation Aveugle de Sources (SAS) ou Analyse en Composantes Indépendantes (ACI) est un sujet relativement nouveau en traitement du signal qui a été Introduit en milieu des années 80 par les travaux d'*Ans*, *Hérault et Jutten* [L8, L19, L18], alors qu'ils travaillaient sur un problème biologique (voir [L9] pour notes historiques sur la SAS). Le problème consiste à récupérer des signaux indépendants non observées à partir de mélanges de celles-ci, comme cité précédemment, sans connaissances préalables ni sur les signaux source d'origine, ni sur le système de mélange (d'où le terme aveugles). En dehors de l'invariance de la transformation au cours du temps, nous pouvons distinguer les contextes suivants :

- La nature de la transformation ($\mathcal{F}$) : Linéaire, non linéaire, linéaire convolutive, non linéaire convolutive, post non linéaire, …etc.
- L'instantanéité du système ($\mathcal{F}$) : instantané ou convolutif.

Diverses informations peuvent être utilisées dans un but d'identification de $\mathcal{F}$ ou de son inverse. Par exemple, des connaissances a priori peuvent être disponibles sur le système de mélange (modèle physique, . . .). Dans d'autres cas, il est possible de disposer à la fois des entrées et sorties du système $\mathcal{F}$. Dans le cas où aucune information a priori n'est disponible on se trouve dans le cadre dit aveugle, ce qui signifie, d'une part qu'aucune connaissance n'est disponible sur $\mathcal{F}$ en dehors des hypothèses de structure (c'est-à-dire de modèle), et d'autre part

que les sources sont inobservables. Ce défaut de connaissance est toutefois compensé par une hypothèse statistique forte qui est celle d'indépendance mutuelle des sources.

Les hypothèses sur les propriétés statistiques des sources sont généralement la base des algorithmes de Séparation [B2]. Certaines hypothèses sont :

- L'indépendance statistique des sources [L1, L16],
- Sources statistiquement orthogonales [L13],
- Sources non stationnaires [L22].

L'équivalence entre l'ACI et la SAS disparaît dès que l'on quitte le domaine linéaire. Nous citons l'exemple présenté dans [NL3]. L'exemple montre que les systèmes non-linéaires peuvent garantir l'indépendance des sorties sans les séparer. Bien que les mélanges non linéaires ne soient pas séparables dans le cas général, nous pouvons toujours trouver des sous-classes qui sont séparables. Un exemple est celui des mélanges Post-non linéaire (PNL), qui ont été introduit par *Taleb et Jutten* [NL3]. Dans ces mélanges, un mélange instantané linéaire est suivi par des non-linéarités inversibles.

Un mélange non linéaire (instantané), dans sa forme générale, est une transformation, $\boldsymbol{x} = \boldsymbol{\mathcal{F}}(\boldsymbol{s})$, où $\boldsymbol{x}$ et $\boldsymbol{s}$ désignent les vecteurs d'observation et de sources, respectivement. Il faut alors trouver une autre transformation $\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{x})$ telle que les composantes de $\boldsymbol{y}$ soient indépendantes (ACI). Cependant, dans les transformations non linéaires, l'indépendance ne signifie pas la séparation des sources (SAS). En effet, les mélanges non linéaires ne sont pas séparables, et l'hypothèse d'indépendance statistique des sources n'est pas assez forte pour aboutir à leur séparation. En d'autres termes, pour des mélanges non linéaires,

une solution totalement aveugle n'est pas possible. Il faut avoir d'autres informations supplémentaires, par exemple sur la structure du mélange.

Les mélanges non linéaires ont été beaucoup moins étudiés dans la littérature [NL8, NL15, NL12, NL3, NL13], et jusqu'à présent peu de résultats sont disponibles. Une des raisons est bien sûr la difficulté mathématique des systèmes non linéaires, mais une autre raison importante est le fait que les mélanges non linéaires ne sont pas séparables. Le problème remonte aux travaux de *Christian Jutten* [th1], quand il a utilisé des mélanges non linéaires afin d'évaluer la robustesse et les performances de l'algorithme HJ [L8].

Pajunen et al. [NL15] ont utilisé les Cartes auto-organisatrice de Kohonen (SOM : Self Organizing Maps) afin de séparer des mélanges non linéaires. Les SOM [B1] est une méthode très connue, qui est capable d'approximer une transformation non linéaire d'une manière non supervisée. L'une des difficultés de cette approche est de chercher à créer des sorties uniformément distribuées. Pour résoudre ce problème, dans [NL14] un mappage topographique génératif (GTP : Generative Topographic Mappings) a été utilisé comme une alternative au SOM [B2]. Dans cette approche, les sources peuvent avoir n'importe quelle distribution à condition d'être connue a priori.

Déco et Brauer [NL9] ont également abordé le problème, en imposant une condition de préservation de volume sur les transformations non linéaires. Yang *et al*. [NL12] ont étudié le problème pour des types spéciaux des mélanges non linéaires, avec l'hypothèse que les non-linéarités inverses peuvent être estimées à l'aide d'un perceptron à deux couches.

Valpola et al. [NL11, NL10] ont proposé une approche méthode de séparation où la transformation de $\boldsymbol{s}$ à $\boldsymbol{x}$ est modélisée par un perceptron multicouches

(MLP : Multi Layer Perceptron), et l'apprentissage de type Bayésien dans ce cas est non supervisé.

Les mélanges post non linéaires introduits par *A. Taleb et C. Jutten* [NL3, NL2] comme des mélanges réalistes, présentent clairement l'hypothèse de structure. Ces mélanges ont également été considérés dans d'autres travaux [NL21, NL18]. Dans [NL3], les auteurs ont utilisé l'information mutuelle des sorties comme critère de séparation et afin de compenser les non linéarités un MLP est utilisé pour modéliser les transformations inverses. Dans [NL2], une approche non paramétrique a été développée pour l'estimation des non linéarités. Cette approche a été étudiée en détail dans [NL21].

Dans notre travail nous considérons les deux types de mélanges : linéaire et le modèle Post Non Linéaire (PNL). L'Information Mutuelle (***IM***) (***MI*** : Mutual Information) des sorties est prise comme critère de séparation. En effet, l'information mutuelle est une mesure de dépendance de variables aléatoires, et ce n'est que la divergence de *Kullback-Leibler* entre la fonction de densité de probabilité (PDF : Probability Density Function) conjointe et le produit des densités de probabilités marginales de ces variables aléatoires. Les algorithmes de séparation de ce type de mélanges non linéaires utilisant l'information mutuelle comme critère d'optimisation ont fait l'objet de plusieurs travaux [NL31, NL32, NL33, NL34, NL35, NL36], et les paramètres du système de séparation sont déterminés en minimisant ce même critère. Il est important de noter que le gradient de l'***IM*** n'est rien que la différence des fonctions score (SFD : Score Function Difference). La SFD est calculée à partir de la fonction score marginale (MSF : Marginal Score Function) et la fonction score conjointe (JSF : Joint Score Function), et les deux nécessitent soit la connaissance préalable des lois de probabilité des sorties soit leurs estimations.

Afin de montrer leur puissance dans l'estimation des signaux dans le domaine de séparation aveugle de sources, les réseaux de neurones artificiels ont été utilisés dans toutes les étapes de séparation à partir de l'estimation des fonctions de densité de probabilités à l'estimation des transformations de compensation des nonlinéarités au niveau de l'étage de séparation. Cette idée présente un indice de performances très remarquable vis-à-vis la qualité d'estimation.

1.3. PLAN DU DOCUMENT

Ce manuscrit se divise en six chapitres et une annexe. Le présent chapitre constitue une introduction au problème traité, et un état le l'art est rapidement exposé.

- **Chapitre 2**

Ce chapitre est consacré dans sa totalité aux problèmes de séparation aveugle de sources et l'analyse en composantes indépendantes. Nous présentons les modèles mathématiques de quelques représentations rencontrées en littérature. Nous exposons aussi, les notions d'ambiguïtés et de prétraitement souvent présentes dans la procédure de séparation. Nous terminons ce chapitre par une présentation détaillée des deux types de mélanges : linéaire instantané et post-nonlinéaire. Nous discutons aussi la séparabilité de ces deux types de mélanges.

- **Chapitre 3**

Dans ce chapitre, nous présentons quelques critères de séparation de sources qui exploitent l'indépendance statistique tels que : la non-gaussianité, la fonction de vraisemblance, et l'information mutuelle. Nous introduisons aussi la notion de la divergence de *Kullback-Leibler* comme msure de distance entre fonction de probabilité.

- **Chapitre 4**

Dans ce chapitre nous évoquons, essentiellement, les techniques de la SAS basées sur le critère de l'information mutuelle. Pour cela, nous commencerons par introduire la notion de fonctions score qui constituent la base de plusieurs algorithmes de séparation. Ensuite, nous présentons la technique du gradient pour la minimisation de l'information mutuelle et ceci pour les types de mélanges : Linéaire instantané et Post-nonlinéaire.

- **Chapitre 5**

Ce chapitre comprend les contributions de ce travail. Nous présentons une méthode d'estimation des densités de probabilité (pddf) marginales dans le but d'estimer les fonctions score marginales (MSF). La méthode est basée sur un réseau de neurone multicouches (MLP) et un apprentissage non supervisé utilisant une modélisation de forme exponentielle des pdf.

Les estimées des fonctions score sont utilisées par la suite dans un algorithme de séparation de sources de mélange linéaire instantané en minimisant un critère d'information mutuelle sous contraintes.

Considérée comme une deuxième partie de ce chapitre, nous nous placerons dans une situation plus complexe où nous considérons un type de mélanges plus réaliste, qui est le mélange post-nonlinéaire. Pour ce type de mélanges, nous proposons une solution purement neuronale pour la résolution du problème de séparation de source. Le réseau MLP proposé est appris d'une manière non supervisée pour : l'estimation des nonlinéarités inverse d'une part, et l'estimation des signaux source d'autre part.

Les méthodes proposées sont validées par des résultats de simulation.

Nous terminons ce manuscrit par une conclusion générale récapitulant le contenu de cette thèse et les perspectives de futures recherches.

CHAPITRE 2

Séparation Aveugle de Sources (SAS) et Analyse en Composantes Indépendantes (ACI)

2.1. INTRODUCTION

La séparation aveugle de source consiste à estimer un jeu de p sources inconnues à partir d'un jeu de M observations. Ces observations sont des mélanges de ces sources et proviennent de capteurs (antennes, microphones, caméras par exemple). Le mélange entre ces sources, qui s'effectue pendant leur propagation jusqu'aux capteurs, est inconnu. La SAS est une discipline qui permet de nombreuses applications [B4] dans de nombreuses disciplines telles l'acoustique, les télécommunications, le génie biomédical [L70], l'astrophysique[L71].

Il existe dans la littérature plusieurs types de mélanges découpés en deux catégories : les mélanges linéaires et les mélanges nonlinéaires.

Il existe différentes méthodes permettant la séparation de sources dans le cadre des mélanges linéaires instantanés. Ces méthodes sont le plus souvent regroupées en trois catégories de méthodes. La première catégorie regroupe les méthodes fondées sur l'Analyse en Composantes Indépendantes (ACI) (Independent Component Analysis (ICA)). Dans la deuxième catégorie, on trouve les méthodes basées sur la Factorisation en Matrices Non-négatives (FMN) (Non-negative Matrix Factorization (NMF)). Enfin, les méthodes fondées sur l'Analyse en Composantes Parcimonieuses (ACPa) (Sparse Component Analysis (SCA)) sont regroupées dans la troisième et dernière catégorie.

Les contributions de cette thèse sont fondées sur la première catégorie qui est l'analyse en composantes indépendantes.

2.2. Analyse en Composantes Indépendantes (ACI)

L'analyse en Composantes Indépendantes (ACI), dans un contexte linéaire, d'un vecteur aléatoire, consiste à trouver une transformation linéaire qui minimise la dépendance statistique entre ses composantes.

Soit $x_1, x_2, \dots, x_n$ des combinaisons linéaires des n variables aléatoires latentes, $s_1, s_2, \dots, s_n$, telles que :

$$x_i = a_{i1}s_1 + a_{i2}s_2 + \ \dots + \ a_{in}s_{in} \ , \quad i = 1, 2, \dots, n \tag{2.1}$$

Où a_{ij} sont des coefficients réels inconnus.

Par définition, les variables aléatoires s_i sont mutuellement indépendantes. C'est la base du concept ACI. Donc, la fonction densité de probabilité (pdf : probability desity function) conjointe du vecteur $(s_1, s_2, \dots, s_n)^T$ est égale au produit des fonctions densité de probabilité marginales de chaque variable aléatoire s_i.

i.e.

$$p_{s_1,s_2,\dots,s_n}(s_1, s_2, \dots, s_n) = \prod_{i=1}^{n} p_{s_i}(s_i) \quad (2.2)$$

$p_{s_1,s_2,\dots,s_n}(s_1, s_2, \dots, s_n)$ est la pdf conjointe du vecteur $(s_1, s_2, \dots, s_n)^T$

$p_{s_i}(s_i)$ est la pdf marginale de la variable aléatoire s_i .

$(.)^T$: est le transposé d'un vecteur ou d'une matrice.

Sous forme matricielle (2.1) s'exprime :

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} \quad (2.3)$$

Où $\boldsymbol{A} = [a_{ij}]$ s'appelle la matrice de mélange (mixing matrix)

Dans ce cas, le but de l'ACI est d'estimer une matrice $\boldsymbol{B}$, appelée matrice de séparation, en disposant de la relation (2.2) de sorte à ce que les composantes du vecteur $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x}$ soient mutuellement indépendantes.

Où $\boldsymbol{y} = (y_1, y_2, \dots, y_n)^T$

2.2.1. Prétraitement pour l'ACI

Avant l'application de n'importe quel algorithme d'ACI, il est impératif de procéder à un préalable prétraitement des observations. Ce prétraitement consiste à centrer et à blanchir les variables aléatoires observées. L'intérêt d'un tel prétraitement est de permettre avantageusement de restreindre la recherche à l'espace vectoriel des matrices orthogonales.

- **Etape de centrage**

Le prétraitement de base est de centrer les observations, en retranchant du vecteur $\boldsymbol{x}$ sa valeur moyenne $\boldsymbol{m} = E[\boldsymbol{x}]$ afin de le rendre à valeur moyenne nulle.

$E[.]$ est l'opérateur de l'espérance mathématique.

$$\hat{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{m} \tag{2.4}$$

Après l'estimation de la matrice séparante $\boldsymbol{B}$, nous pouvons rajouter le terme soustrait aux estimées centrées de $\boldsymbol{s}$, qui est égal à : $\boldsymbol{A}^{-1}\boldsymbol{m}$

- **Etape de Blanchiment**

Une deuxième étape du prétraitement consiste au blanchiment du vecteur d'observations. Donc, après avoir centré les données, transformer le vecteur résultant à un autre vecteur, $\tilde{\boldsymbol{x}}$, dont les composantes sont non corrélées et de variances unité. Autrement dit, la matrice de covariance de $\tilde{\boldsymbol{x}}$ est égale à la matrice identité

i.e.

$$E[\tilde{\boldsymbol{x}}\,\tilde{\boldsymbol{x}}^{T}] = \boldsymbol{I}_n \tag{2.5}$$

Une des méthodes de blanchiment les plus populaires, est la décomposition en vecteurs et valeurs propres (EVD : Eigen-Value Decomposition) de la matrice de covariance.

$$E[\tilde{\boldsymbol{x}}\,\tilde{\boldsymbol{x}}^{\mathbf{T}}] = \boldsymbol{E}\boldsymbol{D}\boldsymbol{E}^{T} \tag{2.6}$$

Où $\boldsymbol{E}$ est une matrice orthogonale des vecteurs propres

$\boldsymbol{D}$ est une matrice diagonale des valeurs propres μ_i telle que :

$$\boldsymbol{D} = \begin{pmatrix} \mu_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mu_n \end{pmatrix} = diag(\mu_1, \mu_2, \dots, \mu_n) \tag{2.7}$$

Et dans ce cas $\tilde{\boldsymbol{x}}$ est obtenu par l'équation

$$\begin{aligned} \tilde{\boldsymbol{x}} &= \boldsymbol{D}^{-1/2}\boldsymbol{E}^{T}\hat{\boldsymbol{x}} \\ &= \boldsymbol{D}^{-1/2}\boldsymbol{E}^{T}\boldsymbol{A}\boldsymbol{s} = \tilde{\boldsymbol{A}}\boldsymbol{s} \end{aligned} \tag{2.8}$$

En effet l'importance du blanchiment est d'avoir une matrice de mélange $\widetilde{\boldsymbol{A}}$ orthogonale :

$$E[\widetilde{\boldsymbol{x}}\,\widetilde{\boldsymbol{x}}^{T}] = \widetilde{\boldsymbol{A}}E[\boldsymbol{s}\boldsymbol{s}^{\boldsymbol{T}}]\widetilde{\boldsymbol{A}}^{T} = \widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{A}}^{T} = \boldsymbol{I} \qquad (2.9)$$

Il est clair que le blanchiment réduit le nombre de paramètres à estimer. Au lieu d'avoir n^2 paramètres de la matrice originale $\boldsymbol{A}$, nous obtiendrons seulement $n(n-1)/2n$ paramètres de la matrice $\widetilde{\boldsymbol{A}}$.

- **Illustration**

Pour comprendre le concept ACI en terme statistique, prenons l'exemple suivant :

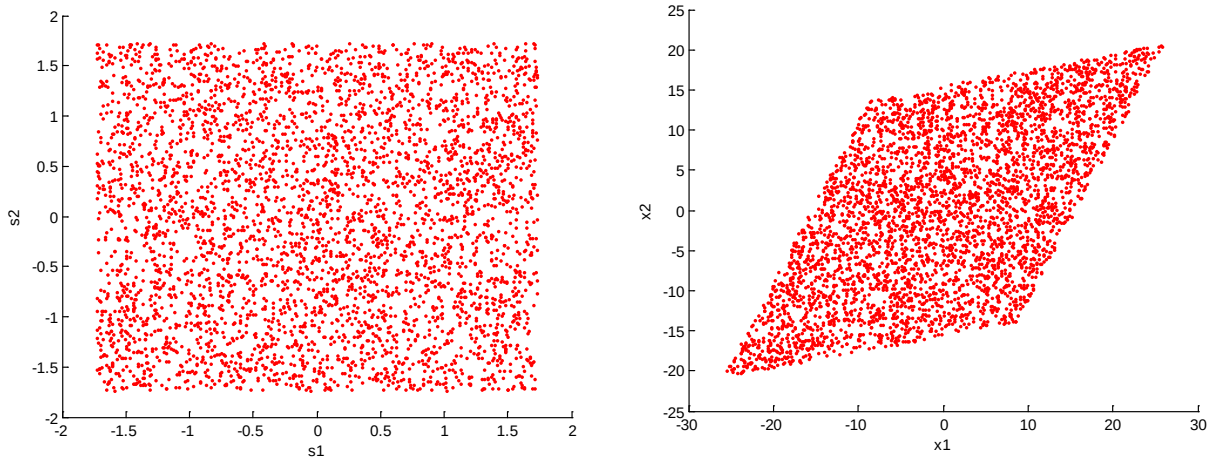
Soit s_1 et s_2 deux variables aléatoires indépendantes et uniformément distribuées sur l'intervalle $\left[-\sqrt{3}\,,\sqrt{3}\,\right]$. l'expression de leurs pdf s'écrit donc :

$$p_{s_i}(s_i) = \begin{cases} \frac{1}{2\sqrt{3}} & \text{pour} \quad |s_i| \leq \sqrt{3} \\ 0 & \text{ailleurs} \end{cases}, \qquad i = 1,2. \qquad (2.10)$$

Et soit $\boldsymbol{A}$ la matrice de mélange de s_1 et s_2

$$\boldsymbol{A} = \begin{pmatrix} 5 & 10 \\ 10 & 2 \end{pmatrix}$$

Ce qui produit d'autres variables x_1 et x_2 , et il est facile de calculer leurs distributions et de conclure qu'ils suivent une loi uniforme dans un parallélogramme, comme il est montré par la figure (2.1)



(a) (b)

Fig.2.1. Distribution conjointe de : (a) -(s_1 , s_2) et (b) - (x_1 , x_2).

2.2.2. Pourquoi les variables gaussiennes ne sont pas séparables par ACI

En réalité, une seule variable gaussienne au plus est permise. Pour illustrer cette hypothèse, supposons que s_1 et s_2 sont gaussiennes. Ceci implique que leur densité conjointe est donnée par l'expression suivante :

$$p_{s_1,s_2}(s_1,s_2) = \frac{1}{2\pi} exp\left(-\frac{s_1^2+s_2^2}{2}\right) = \frac{1}{2\pi} exp\left(-\frac{\|\boldsymbol{s}\|^2}{2}\right) \qquad (2.11)$$

$\|\,.\,\|$ est la norme 2 d'un vecteur.

En supposant que la matrice de mélange $\boldsymbol{A}$ est orthogonale, i.e., $\boldsymbol{A}^{-1} = \boldsymbol{A}^T$, et que $\det(\boldsymbol{A}) = 1$. alors la densité conjointe de $\boldsymbol{x} = (x_1, x_2)^T$ s'exprime :

$$p_{x_1,x_2}(x_1,x_2) = \frac{1}{2\pi} exp\left(-\frac{\|\boldsymbol{A}^T\boldsymbol{x}\|^2}{2}\right) |det(\boldsymbol{A}^T)| = \frac{1}{2\pi} exp\left(-\frac{\|\boldsymbol{x}\|^2}{2}\right) \qquad (2.12)$$

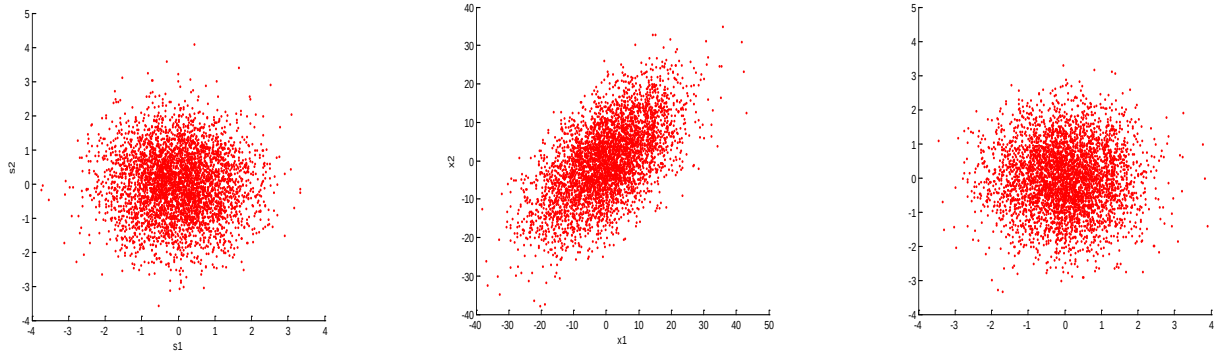
On peut voir que la distribution du vecteur $\boldsymbol{s}$ et identique à celle du vecteur mélange. Cela provient du fait que des variables gaussiennes orthogonales sont nécessairement indépendantes. Alors, la matrice de mélange $\boldsymbol{A}$, est non identifiable dans ce cas. Et le modèle ne peut être estimé qu'à une transformation orthogonale près.

- **illustration**

Soient s_1 et s_2 deux variables gaussiennes indépendantes de moyennes nulles et de variances 1. Et soit $\boldsymbol{x} = (x_1, x_2)^T$ un mélange linéaire de ces deux variables.

La figure (2.2) montre clairement que la pdf conjointe de $\boldsymbol{s}$ est identique à celle de $\boldsymbol{x}$ après blanchiment, ce qui coïncide avec (2.12).

L'équivalence entre le problème de la SAS et l'analyse en composantes indépendantes, est vérifiée pour les cas des mélanges linéaires instantanés et convolutifs, à condition qu'il y ait au plus une source gaussienne.



(a) (b) (c)

Fig.2.2. Distribution conjointe : (a) sources, (b) mélange sans blanchiment, (c) mélange avec blanchiment

2.3. MODELES DE MELANGES EN SAS

Le problème de séparation de sources comme précédemment définie, consiste à retrouver des signaux utiles provenant de plusieurs sources (par abus de langage on dit : signaux sources), qui ont été filtrées et additionnées (i.e. mélangées) en se propageant vers un ensemble de capteurs.

La propagation des signaux à séparer, de leurs sources productrices aux capteurs de mesure, a été initialement modélisée par les premiers fondateurs de cette nouvelle discipline à base d'un produit mathématique simple. Ceci correspond au cas appelé "***instantané***" de la séparation de sources.

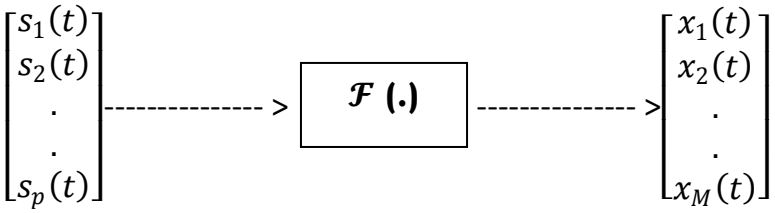
Plus tard, il s'est avéré que cette modélisation ne convenait pas à toutes les situations rencontrées dans la pratique. C'est pourquoi, des modélisations plus réalistes ont été proposées. L'une de ces modélisations interprète le phénomène de propagation comme une opération de filtrage, c'est-à-dire qu'elle suppose que l'environnement est caractérisé par une fonction mathématique dépendante du temps et réalise une opération plus complexe, qui est un produit de convolution, afin d'engendrer les mélanges dits convolutifs. Ceci correspond au cas plus intéressant de la séparation de sources appelé "***convolutifs***".

Les deux modèles ont été largement étudiés aux cours de plusieurs années [L12, L24, L36, L41, L33, LC8, LC6, LC14, LC19, LC10, LC9, LC4, LC18]. En revanche, le cas de mélanges non linéaires n'a été que très peu abordé. Une des raisons est la difficulté mathématique des systèmes non linéaires, mais une autre raison importante est le fait que les mélanges non linéaires ne sont pas séparables en se fondant sur la seule hypothèse d'indépendance statistique, et sans hypothèse supplémentaire sur les sources ou sur les mélanges, le problème a donc peu d'intérêt. . Mais, malgré la difficulté du modèle non linéaire, d'autres auteurs se sont penchés sur ce problème, et les contributions dans ce domaine sont rares et sont limités à l'hypothèse de structure [NL21, NL18, NL27, NL28, NL32-NL34].

Cette hypothèse traduit clairement l'intérêt que présente le mélange de type ***post non linéaire (PNL)***, dont le choix repose sur des modèles physiques assez réalistes.

Le problème de la SAS peut être exposé de la façon suivante :

Considérons p signaux sources $(s_i(t)\ ,\ i = 1, 2,\ \ldots p.\ et\ \ t = 0, 1, \ldots N-1.)$ non observables subissant une opération de mélange au travers d'un système $\boldsymbol{F}$ fournissant M observations $(x_j(t)\ ,\ j = 1, 2,\ \ldots M.)$



Sous une forme compacte nous pouvons écrire :

$$\boldsymbol{S} = \boldsymbol{\mathcal{F}}(\boldsymbol{X}) \qquad (2.13)$$

Où

$$\boldsymbol{S} = \left(\boldsymbol{s}_1\ , \boldsymbol{s}_2, \ldots\ \boldsymbol{s}_p\right)^T \qquad \text{et} \qquad \boldsymbol{X} = (\boldsymbol{x}_1\ , \boldsymbol{x}_2, \ldots\ \boldsymbol{x}_M)^T$$

où

$\boldsymbol{s}_i = \left(s_i(0), s_i(1), s_i(2), \ldots\ s_i(N-1)\right)$ est la $i^{ème}$ source

et $\boldsymbol{x}_j = \left(x_j(0), x_j(1), x_j(2), \ldots\ x_j(N-1)\right)$ est le $j^{ème}$ mélange (signal reçu au $j^{ème}$ capteur).

L'objectif à atteindre est de trouver le moyen d'annuler l'effet du système $\boldsymbol{\mathcal{F}}$, c'est-à-dire, trouver un inverse $\boldsymbol{G} = \boldsymbol{\mathcal{F}}^{-1}$ afin de restituer les signaux sources. Pour être en mesure de proposer des solutions, nous sommes en général amenés à

poser des hypothèses sur la structure du mélange. Nous pouvons distinguer les contextes suivants :

- Mélange linéaire instantané [L1, L5, L6, L7, L9, L11, L14, L16, L45, L47, L48]
- Mélange post non linéaire (PNL) [NL2, NL3, NL18, NL19, NL20, NL25]
- Mélange linéaire convolutif [LC1, LC2, LC7, LC11, LC19]
- Mélange post non linéaire convolutif (CPNL :Convolutive post-nonlinear mixture) [NL26, NL29, NL30]
- Mélange non linéaire général [NL1, NL5, NL9, NL11, NL14]

Sans perte de généralités, nous supposons dans ce qui suit que le nombre de signaux sources est égal au nombre de capteurs, c.à.d. $p = M$.

Dans le reste de ce travail nous nous intéressons au cas des mélanges instantanés.

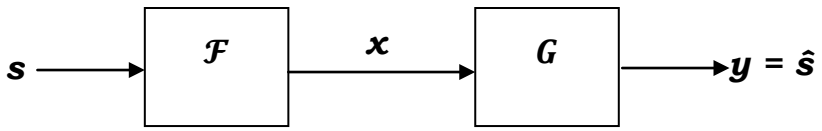


Fig. 2.3. Principe de la séparation aveugle de sources.

Dans le cas général, et au sens de l'analyse en composante indépendantes, le but de séparation est d'obtenir :

$$y_i(t) = g_i\left(s_{\sigma(i)}(t)\right) \quad , \qquad i = 1, \dots, p \tag{2.14}$$

Où σ est une permutation sur l'ensemble $\{1, 2, \dots, p\}$

Et g_i est une transformation inversible.

Cette équation montre explicitement que l'approche de séparation (ACI) contient des indéterminations d'estimation qui vont être exposées en détail dans la section suivante.

Afin d'accomplir la tâche de séparation des signaux, il est indispensable de citer les hypothèses principales sur lesquelles est basée la SAS :

H1. Les sources sont statistiquement mutuellement indépendantes.

H2. Dans le cas de sources i.i.d. (indépendantes et identiquement distribuées) une seule source au maximum peut être gaussienne.

H3. On suppose que le nombre de sources p est égal au nombre de capteurs M.

2.3.1. Mélanges Linéaires Instantanés

C'est le modèle le plus simple de la SAS, où p signaux reçus sont supposés être des mélanges linéaires instantanés de p signaux sources statistiquement indépendants. Le modèle s'écrit alors :

$$x_i(t) = \sum_{j=1}^{p} a_{ij}\, s_j(t)\,, \quad i = 1, \dots, p \tag{2.15}$$

Où a_{ij} sont des constantes inconnues.

Sous forme compacte, le modèle (2.3) s'écrit donc :

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t) \tag{2.16}$$

Ou plus encore :

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} \tag{2.17}$$

$\boldsymbol{A} \triangleq [a_{ij}]$ est la matrice de mélange.

Le problème de la SAS, dans ce cas, consiste à trouver une matrice de séparation, $\boldsymbol{B}$ de dimension$(p \times p)$, de sorte à ce que le vecteur de sortie $\boldsymbol{y} = \boldsymbol{Bx}$ soit un estimé du vecteur source $\boldsymbol{s}$.

a. Ambiguïtés de séparation

Afin de séparer les sources aveuglement à partir de leurs mélanges, il faut prendre en compte les indéterminations propres à une telle modélisation. En effet, nous avons une infinité de solutions [L45, L16, L9]. Common [L45] a montré que sans condition supplémentaire, les sources ne peuvent être estimées qu'à une permutation et un facteur d'échelle près.

a.1. Ambiguïté de permutation

C'set la première indétermination liée au problème de la SAS. Elle est liée à l'ordre arbitraire de restitution des signaux. Dans ce cas, la relation qui lie les deux vecteurs $\boldsymbol{s}$ et $\boldsymbol{x}$ sous forme vectorielle s'écrit :

$$\boldsymbol{x}(t) = \sum_{j=1}^{p} \boldsymbol{a}_j s_j(t)\,, \quad o\grave{u} \quad \boldsymbol{a}_j = \left(a_{1j}\,, a_{2j}\,, \ldots, a_{pj}\right)^T \qquad (2.18)$$

i.e. :

$$\boldsymbol{x}(t) = \boldsymbol{a}_1 s_1(t) + \boldsymbol{a}_2 s_2(t) + \cdots + \boldsymbol{a}_p s_p(t) \qquad (2.19)$$

$$= \boldsymbol{a}_2 s_2(t) + \boldsymbol{a}_1 s_1(t) + \cdots + \boldsymbol{a}_p s_p(t) \qquad (2.20)$$

Donc,

$$\boldsymbol{x}(t) = \begin{bmatrix} a_{12}s_2(t) \\ . \\ . \\ . \\ a_{p2}s_2(t) \end{bmatrix} + \begin{bmatrix} a_{11}s_1(t) \\ . \\ . \\ . \\ a_{p1}s_1(t) \end{bmatrix} + \cdots + \begin{bmatrix} a_{1p}s_p(t) \\ . \\ . \\ . \\ a_{pp}s_p(t) \end{bmatrix} \qquad (2.21)$$

Et enfin,

$$\boldsymbol{x}(t) = \begin{bmatrix} a_{12} & a_{11} & \dots & a_{1p} \\ a_{22} & a_{21} & \dots & a_{2p} \\ . & . & & . \\ . & . & & . \\ a_{p2} & a_{p1} & \dots & a_{pp} \end{bmatrix} \begin{bmatrix} s_2(t) \\ s_1(t) \\ . \\ . \\ s_p(t) \end{bmatrix} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{s}}(t) \qquad (2.22)$$

Où $\tilde{\boldsymbol{A}}$ et $\tilde{\boldsymbol{s}}$ sont respectivement la nouvelle matrice de mélange et le nouveau vecteur source. Cette permutation peut être modélisée par une matrice de permutation $\boldsymbol{P}$ telle que :

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t) = (\boldsymbol{A}\boldsymbol{P})\left(\boldsymbol{P}^{-1}\boldsymbol{s}(t)\right) = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{s}}(t) \qquad (2.23)$$

Il est clair, que la multiplication à droite de La matrice de mélange $\boldsymbol{A}$ par une matrice de permutation $\boldsymbol{P}$ modifie l'ordre des sources sans changer les mélanges.

a.2. Ambiguïté d'échelle

Il est impossible de déterminer les variances (énergies) des composantes indépendantes (sources). En effet, $\boldsymbol{s}$ et $\boldsymbol{A}$ étant inconnus, la multiplication d'une source s_i par un scalaire peut toujours être compensée (annulée) en divisant la colonne $\boldsymbol{a}_i$ de $\boldsymbol{A}$ par le même scalaire, soit α_i :

$$\boldsymbol{x}(t) = \sum_{j=1}^{p} \boldsymbol{a}_j s_j(t) = \sum_{j=1}^{p} \left(\frac{1}{\alpha_j}\boldsymbol{a}_j\right)\left(\alpha_j s_j(t)\right), \quad pour\ \ tout\ \alpha_j \neq 0 \qquad (2.24)$$

i.e.

$$\boldsymbol{x}(t) = \begin{bmatrix} \alpha_1 a_{11} & \alpha_2 a_{12} & \dots & \alpha_p a_{1p} \\ \alpha_1 a_{21} & \alpha_2 a_{22} & \dots & \alpha_p a_{2p} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \alpha_1 a_{p1} & \alpha_2 a_{p2} & \dots & \alpha_p a_{pp} \end{bmatrix} \begin{bmatrix} \frac{s_1(t)}{\alpha_1} \\ \frac{s_2(t)}{\alpha_2} \\ \cdot \\ \cdot \\ \frac{s_p(t)}{\alpha_p} \end{bmatrix} \tag{2.25}$$

Soit $\boldsymbol{D}$ la matrice diagonale constituée des éléments $\alpha_1, \alpha_2, \dots, \alpha_p$. Dans ce cas, nous pouvons écrire :

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t) = (\boldsymbol{A}\boldsymbol{D})\left(\boldsymbol{D}^{-1}\boldsymbol{s}(t)\right) = \check{\boldsymbol{A}}\check{\boldsymbol{s}}(t) \tag{2.26}$$

A partir de (2.25), nous concluons que l'amplitude des sources est indéterminée.

b. Algorithmes de séparation

A l'aide du théorème de Darmois [L46], Common [L45] a démontré que dans le cas $p \leq M$, l'analyse en composantes indépendantes est alors équivalente au problème de séparation aveugle de sources, bien sûr à condition qu'il y ait au plus une source gaussienne, et les sources pourront être restituées à une permutation et un facteur d'échelle près comme déjà vu dans la section précédente.

Common [L45] a montré aussi, que dans le cadre d'un mélange linéaire, les signaux $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x}$ sont indépendants si et seulement si la matrice séparante $\boldsymbol{B}$ est de la forme :

$$\boldsymbol{B} = \boldsymbol{D}\boldsymbol{P}\boldsymbol{A}^{-1} \tag{2.27}$$

Où $\boldsymbol{D}$ représente une matrice diagonale et $\boldsymbol{P}$ une matrice de permutation.

Dans ce contexte, de nombreuses approches ont été développées, parmi lesquelles nous citons :

- A partir des statistiques d'ordre supérieur [L48, L24]
- A partir de la néguentropie [B2]
- L'algorithme SOBI [L47]
- L'algorithme JADE [L48]
- L'InfoMax [L1]
- L'algorithme FastICA [B2]

2.3.2. Mélanges non linéaires

Un mélange non linéaire instantané dans sa forme générale s'écrit :

$$\boldsymbol{x(t)} = \boldsymbol{\mathcal{F}}\big(\boldsymbol{s(t)}\big) \tag{2.28}$$

Où $\boldsymbol{F}$est une transformation (mapping) non linéaire.

i.e.,

$$x_1 = f_1(s_1, s_2, \dots, s_p)$$
$$x_2 = f_2(s_1, s_2, \dots, s_p)$$
$$\vdots$$
$$x_p = f_p(s_1, s_2, \dots, s_p)$$

Où, $f_1, f_2, \dots, f_p$ sont des fonctions nonlinéaires inversibles.

D'une manière générale, le but de l'ACI non linéaire consiste à estimer une transformation $\boldsymbol{G} : \mathcal{R}^p \rightarrow \mathcal{R}^p$ qui produit un vecteur $\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{x})$ à composantes statistiquement indépendantes, et ceci en n'utilisant que les observations.

Mais, comme déjà mentionné dans le premier chapitre, l'indépendance ne signifie pas la séparation des sources, et une solution complètement aveugle n'est pas possible.

- **Indéterminations en ACI nonlinéaire**

Une transformation bijective $\boldsymbol{\mathcal{H}}$ est dite *triviale*, si elle transforme n'importe quel vecteur $\boldsymbol{s}$ à composantes indépendantes à un autre vecteur à composantes indépendantes. I.e., les transformations triviales préservent la propriété d'indépendance statistique de n'importe quel vecteur. L'ensemble des transformations triviale est noté par $\mathfrak{I}$.

Cette définition se traduit par la condition nécessaire et suffisante suivante : $\boldsymbol{\mathcal{H}}$ est *triviale* si et seulement si :

$$\mathcal{H}_i(u_1, u_2, \dots, u_n) = h_i\left(u_{\sigma(i)}\right), \quad i = 1,2, \dots, n \tag{2.29}$$

Où h_i sont des fonctions scalaires quelconques.

Ces transformations sont caractérisées par une matrice jacobienne diagonale à une permutation près. Ceci montre le lien entre l'hypothèse d'indépendance et l'objectif de séparation des signaux sources. En effet, cet objectif est, tout simplement, de rendre la transformation globale, $\boldsymbol{\mathcal{H}} = \boldsymbol{G} \circ \boldsymbol{\mathcal{F}}$ triviale en utilisant l'hypothèse d'indépendance statistique. Cependant, à partir de l'équation (2.29), il est tout à fait clair que la séparation est achevée à une permutation et une distorsion nonlinéaire près.

Soit $\boldsymbol{\mathcal{F}}(\boldsymbol{x}) = [f_1(\boldsymbol{x}),\ f_2(\boldsymbol{x}), \dots, f_n(\boldsymbol{x})]^T$, dont chaque composante est une fonction nonlinéaire scalaire, telle que :

$$f_i(\boldsymbol{x}) = f_i(x_i), \quad i = 1, \dots, n$$

Il est évident que :

Si $p_{\boldsymbol{x}}(\boldsymbol{u}) = \prod_{i=1}^{n} p_{x_i}(u_i)$ alors,

$$p_{\boldsymbol{\mathcal{F}}(\boldsymbol{x})}(\boldsymbol{v}) = \prod_{i=1}^{n} p_{f_i(x_i)}(v_i)$$

A partir des travaux de [L49], Darmois a conclu que pour les transformations nonlinéaires générales, la préservation de l'indépendance statistique n'était pas une hypothèse assez forte pour assurer la séparation des sources dans le sens de l'équation (2.29).

Dans leurs travaux, Jutten *et al*. [NL6], ont montré qu'il est facile de décomposer n'importe quel vecteur aléatoire à une transformation non triviale de variables aléatoires, et ceci en utilisant une procédure d'orthogonalisation similaire à celle de Gram-Schmidt.

Le résultat de Darmois montre qu'il existe des transformations *non triviales*, $\boldsymbol{\mathcal{H}}$, qui préservent l'indépendance statistiques des variables aléatoires mélangées. Donc, dans le cas général des transformations nonlinéaires, la séparation aveugle est tout simplement impossible en faisant appel à l'hypothèse d'indépendance statistique seule.

En effet, d'autres informations supplémentaires sont nécessaires, telle que l'information sur la structure du mélange.

- **Exemple :** soit $s_1 \in \mathcal{R}^+$ une variable aléatoire qui suit la loi de Rayleigh telle que :

$$p_{s_1}(s_1) = s_1 exp(-s_1^2/2)$$

et supposons qu'une autre variable aléatoire s_2 indépendante de s_1 est uniformément distribuée sur l'intervalle [0, 2π). Prenons la transformation nonlinéaire suivante :

$$\begin{cases} x_1 = s_1 cos(s_2) \\ x_2 = s_1 sin(s_2) \end{cases} \tag{2.30}$$

Le jacobien de cette transformation est alors :

$$\boldsymbol{J}(s_1, s_2) = \begin{bmatrix} cos(s_2) & -s_1 sin(s_2) \\ sin(s_2) & s_1 cos(s_2) \end{bmatrix} \tag{2.31}$$

Donc :

$$\begin{aligned} p_{x_1 x_2}(x_1,x_2) &= \frac{p_{s_1 s_2}(s_1,s_2)}{|J(s_1,s_2)|} = \frac{1}{2\pi} exp\left(-\frac{x_1^2+x_2^2}{2}\right) \\ &= \left(\frac{1}{\sqrt{2\pi}} exp\left(-\frac{x_1^2}{2}\right)\right)\left(\frac{1}{\sqrt{2\pi}} exp\left(-\frac{x_2^2}{2}\right)\right) \end{aligned} \tag{2.32}$$

Et par conséquent, x_1 et x_2 sont deux variables aléatoires gaussiennes indépendantes de moyenne nulle et de variance unité, malgré quelles sont non linéairement mélangées et le jacobien de la transformation n'est pas diagonale. D'autres exemples peuvent être trouvés dans la littérature [B2].

Afin de surmonter ce problème, on peut imposer une condition sur la structure du mélange, c'est à dire imposer à la transformation nonlinéaire d'appartenir à un ensemble particulier, $\mathcal{M}$, de transformations, ou ajouter des a priori sur les sources, ce qui peut réduire énormément les indéterminations prédites des travaux de Darmois.

Jutten et Karhunen [NL5] ont montré que les sources peuvent être restituées si la transformation triviale appartient à l'ensemble $\mathfrak{I} \cap \mathcal{M}$.

2.3.3. Mélanges Post-Nonlinéaires

Une des structures séparables les plus rencontrées en littérature [NL18, TH2, NL2, NL3, NL23, NL19, NL4, NL20] est celle des mélanges post-non linéaires (PNL) (information sur la structure du mélange).

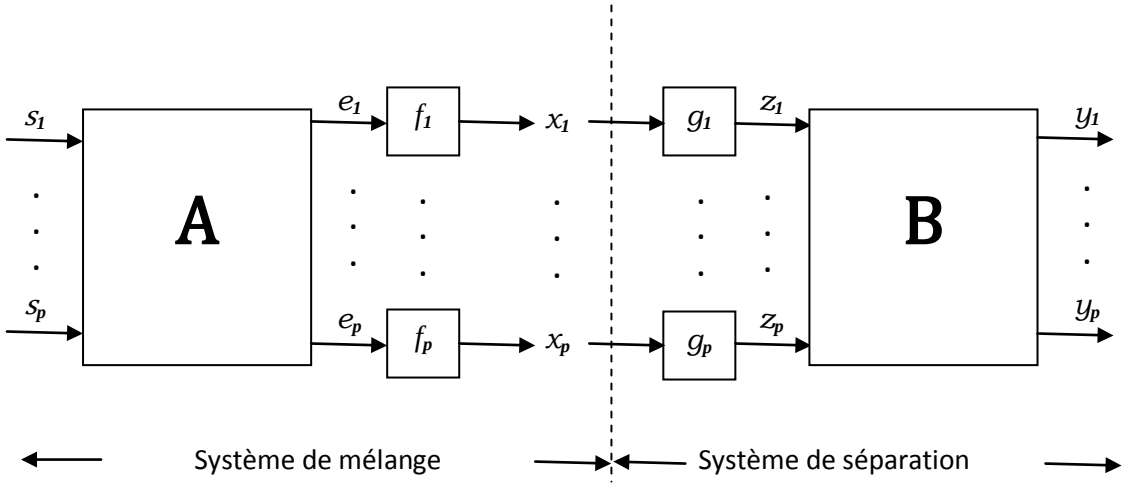


Fig.2.4. Schéma de mélange-séparation d'un mélange PNL.

Un mélange post-nonlinéaire de p sources $s_i(t)$ observées par p capteurs, comme présenté dans par la figure (2.4), est donné par l'expression :

$$x_i(t) = f_i\big(e_i(t)\big), \qquad i = 1, 2, \dots p \tag{2.33}$$

$$e_i(t) = \sum_{j=1}^{p} a_{ij}\, s_j(t)$$

Donc,

$$x_i(t) = f_i\left(\sum_{j=1}^{p} a_{ij}\, s_j(t)\right), \qquad i = 1, 2, \dots p \tag{2.34}$$

Les a_{ij} sont les coefficients réels de la matrice de mélange instantanée, et les f_i sont des fonctions nonlinéaires inversibles inconnues.

Ce choix de mélange nonlinéaire repose sur un modèle physique assez réaliste, dans lequel le canal entraine un mélange linéaire instantané, et les capteurs et leurs instrumentations (amplificateurs, etc.) sont responsables de la distorsion nonlinéaire.

Afin de séparer ce type de mélanges, nous avons besoin, en premier lieu, de compenser les nonlinéarités par les fonctions $g_1, g_2, \dots, g_p$, et par la suite, procéder à la séparation du modèle linéaire résultant (estimation de la matrice de séparation $\boldsymbol{B}$).

Dans ce type de mélange, la transformation globale $\mathcal{H}$ doit appartenir au sous espace $\mathcal{M}$ des transformations constituées : d'une transformation linéaire inversible (matrice régulière $\boldsymbol{A}$) suivie par des fonctions nonlinéaires scalaires inversibles après lesquelles nous trouvons encore une fois une autre transformation linéaire inversible (matrice régulière $\boldsymbol{B}$) , comme nous pouvons le voir sur la figure (2.4).

Où :

$(\boldsymbol{A}, \mathcal{F})$ est la structure du mélange PNL.

$(\boldsymbol{G}, \boldsymbol{B})$ est la structure séparante du mélange PNL.

- **Séparabilité des mélanges PNL**

Les mélanges PNL sont aveuglement séparable [NL3], avec au plus une source gaussienne, et avec les mêmes ambiguïtés des mélanges linéaires si :

- $\boldsymbol{A}$ est régulière avec deux éléments non nuls au minimum dans chaque ligne ou colonne.
- La pdf conjointe des sources est différentiable et sa dérivée est continue sur tout le support de définition [NL19].

2.3.4. Une classe de mélanges nonlinéaires séparables

En 1973 Kagan et al. [NL38], Ont étendu le théorème de Darmois-Skitovic à des mélanges nonlinéaires. Ces résultats ont été redécouverts dans le cadre de la séparation de sources par Eriksson et Koivunen [NL37]. L'idée de base est de considérer des mélanges particuliers $\boldsymbol{\mathcal{F}}$ qui satisfont un *théorème d'addition* au sens de la théorie des équations fonctionnelles.

Mathématiquement, ce théorème est vrai pour les transformations auxquelles existe une fonction inversible f qui satisfait la condition :

$$f(s_1 + s_2) = \boldsymbol{\mathcal{F}}[f(s_1), f(s_2)] \tag{2.35}$$

- **Exemple**

Prenons comme exemple la transformation suivante :

$$\begin{cases} x_1 = \frac{(s_1+s_2)}{(1+s_1 s_2)} \\ x_2 = \frac{(s_1-s_2)}{(1-s_1 s_2)} \end{cases} \tag{2.36}$$

Où s_1 et s_2 sont deux variables aléatoires indépendantes.

Et soit $u_1 = tan^{-1}(s_1)$ et $u_2 = tan^{-1}(s_2)$. L'équation (2.36) devient alors :

$$\begin{cases} x_1 = tan(u_1 + u_2) \\ x_2 = tan(u_1 - u_2) \end{cases} \tag{2.37}$$

En appliquant encore la fonction tan^{-1} à x_1 et x_2 (2.37) devient :

$$\begin{cases} v_1 = tan^{-1}(x_1) = u_1 + u_2 \\ v_2 = tan^{-1}(x_2) = u_1 - u_2 \end{cases} \quad (2.38)$$

Nous pouvons voir que (2.38) est maintenant un mélange linéaire des deux variables aléatoires indépendantes u_1 et u_2.

Ce résultat, est dû au fait que $tan(a+b)$ est une transformation $\boldsymbol{\mathcal{F}}\big(tan(a), tan(b)\big)$, et $f(.) = tan(.)$. Le même raisonnement se fait pour $tan(a-b)$.

Pour plus de détails sur les conditions sur $\boldsymbol{\mathcal{F}}$, voir [NL5].

2.4. CRITÈRES DE PERFORMANCES

Afin d'évaluer les performances des algorithmes de séparation, il est possible de mesurer la précision de chaque source estimées y_i en fonction de la vraie source s_i par deux différents critères : un critère de type erreur quadratique moyenne (EQM) ou rapport signal sur résidus.

2.4.1. Erreur quadratique moyenne (Mean Squared Error, MSE)

L'erreur quadratique moyenne (EQM) mesure la moyenne du carré de l'écart entre le signal source s_i et le signal estimé y_i, $i = 1,2,\dots,p$. Ce terme s'écrit comme suit

$$EQM_i = E_t[(s_i - y_i)^2] = \frac{1}{N}\sum_{n=1}^{N}\big(s_i(n) - y_i(n)\big)^2 \quad (2.39)$$

Où E_t désigne la moyenne temporelle et N le nombre d'échantillons utilisés.

La valeur moyenne de l'EQM sur toutes les sorties est

$$EQM = \frac{1}{p}\sum_{i=1}^{p} EQM_i = \frac{1}{p}\sum_{i=1}^{p}\frac{1}{N}\sum_{n=1}^{N}\big(s_i(n) - y_i(n)\big)^2 \qquad (2.40)$$

Plus l'EQM est petite plus la qualité de séparation est bonne.

2.4.2. Rapport signal sur résidus (SNR)

Le SNR est la mesure de performance la plus répandue dans la séparation de sources.

Dans le cas des mélanges instantanés, le SNR est défini comme le logarithme du rapport en décibels (dB) de la puissance de la source s_i sur celle de l'écart entre la composante s_i et son estimée y_i. En supposant qu'il n'a pas de permutation, le SNR s'écrit donc

$$SNR_i = 10log_{10}\left\{\frac{E_t(s_i^2)}{E_t[(s_i - y_i)^2]}\right\}, \quad i = 1, \dots, p \qquad (2.41)$$

La qualité de séparation s'apprécie avec une valeur du SNR la plus grande possible. Ce qui signifie qu'il n'y a pas une contribution importante provenant d'autres sources à cette sortie y_i.

2.5. CONCLUSION

Dans ce chapitre nous avons présenté le concept de séparation aveugle de sources tout en montrant son équivalence à l'analyse en composantes indépendantes dans le cas des mélanges linéaires. Nous avons montré que la séparation dans ce cas, est achevée avec deux types d'ambiguïtés : ambiguïté de permutation et ambiguïté d'échelle. Nous avons vu que l'équivalence entre l'ACI et

la SAS disparait dès qu'on quitte le domaine linéaire. En effet, la séparation dans le cas des mélanges nonlinéaires est impossible en s'appuyant sur l'hypothèse d'indépendance statistique seule. Une hypothèse de structure du mélange peut réduire considérablement les indéterminations rencontrées dans le cas nonlinéaire et la séparation peut être réalisée avec une permutation et une distorsion nonlinéaire près. Un des exemples réalistes de l'hypothèse de structure est celui du mélange post-nonlinéaire qui constitue une classe de mélanges nonlinéaires séparables.

CHAPITRE 3

Critères d'Indépendance

3.1. INTRODUCTION

Dans le cadre de l'analyse en composantes indépendantes, il est nécessaire de savoir si des variables aléatoires sont indépendantes. Nous verrons que pour certains mélanges de sources, une mesure de dépendance fournira directement un critère de séparation. C'est le cas en particulier de l'information mutuelle dont nous rappelons les propriétés dans les sections suivantes.

Qu'est ce que c'est indépendance ?

Pour définir le concept de l'indépendance statistique, prenons l'exemple de deux variables aléatoires réelles y_1 et y_2. Fondamentalement, les variables y_1 et y_2 sont dites indépendantes si la valeur de y_1 ne donne aucune information sur y_2 et vice versa.

Typiquement, l'indépendance statistique peut être définie par le biais de densités de probabilité des variables aléatoires. Notons par $p_{y_1 y_2}(y_1, y_2)$ la fonction de densité de probabilité (pdf) jointe de y_1 et y_2, et soient $p_{y_1}(y_1)$ et

$p_{y_2}(y_2)$ les fonctions de densité de probabilité marginales (ie. Lorsque chaque variable est prise toute seule) de y_1 et y_2 respectivement.

Alors :

$$p_{y_1}(y_1) = \int p_{y_1 y_2}(y_1, y_2)\mathrm{d}y_2 \tag{3.1}$$

Et

$$p_{y_2}(y_2) = \int p_{y_1 y_2}(y_1, y_2)\mathrm{d}y_1 \tag{3.2}$$

Dans ce cas, on dit que les deux variables y_1 et y_2 sont indépendantes si et seulement si leur pdf jointe vérifie l'équation suivante :

$$p_{y_1 y_2}(y_1, y_2) = p_{y_1}(y_1)p_{y_2}(y_2) \tag{3.3}$$

L'expression (3.3) s'étend naturellement à n'importe quel nombre n de variables aléatoires, et dans ce cas la pdf jointe devient le produit de n termes.

Propriété

Soient f_1 et f_2 deux fonctions scalaires. Il est toujours vrai d'écrire :

$$E[f_1(y_1)f_2(y_2)] = E[f_1(y_1)]E[f_2(y_2)] \tag{3.4}$$

Cette égalité peut être démontrée facilement :

$$E[f_1(y_1)f_2(y_2)] = \iint f_1(y_1)f_2(y_2)\, p_{y_1 y_2}(y_1, y_2)\mathrm{d}y_1\mathrm{d}y_2$$

$$= \iint f_1(y_1)f_2(y_2)p_{y_1}(y_1)p_{y_2}(y_2)\,\mathrm{d}y_1\mathrm{d}y_2$$

$$= \int f_1(y_1)p_{y_1}(y_1)\,\mathrm{d}y_1 \int f_2(y_2)p_{y_2}(y_2)\mathrm{d}y_2$$

$$= E[f_1(y_1)]E[f_2(y_2)] \quad (3.5)$$

3.2. MESURE D'INDEPENDANCE DANS LE CONTEXTE DE LA SAS

Puisque c'est l'idée de base derrière l'ACI, beaucoup d'auteurs se sont attachés à définir des critères de séparation qui se basent sur des mesures de dépendance et cela depuis les années 90.

Après avoir discuté le problème de l'identifiabilité du mélange à l'aide du critère d'indépendance, nous devons examiner comment mettre en œuvre l'indépendance statistique. Quoique le problème puisse être formulé de diverses manières : au sens du maximum de vraisemblance [NL14], ou à l'aide de fonctions de contraste [L48], ou avec un critère quadratique [NL39], nous nous concentrerons sur la divergence de *Kullback-Leibler*, qui donne un éclairage général à de nombreux algorithmes.

3.2.1. Non Gaussianité

Non gaussien implique indépendant, c'est d'ailleurs la clé pour l'estimation du modèle ACI. En effet, en l'absence de non Gaussianité des variables aléatoires l'estimation n'est pas possible.

Considérons le vecteur aléatoire $\boldsymbol{x} = (x_1, \dots, x_n)^T$ répondant au modèle d'ACI de l'équation (2.3), i.e. $\boldsymbol{x}$ est un mélange de variables aléatoires indépendantes, $\boldsymbol{s} = (s_1, \dots, s_n)^T$. Afin d'estimer une des composantes indépendantes, nous considérons une combinaison linéaire des x_i qui peut être écrite par l'équation suivante :

$$y = \boldsymbol{w}^T\boldsymbol{x} = \sum_{i=1}^{n} w_i x_i \quad (3.6)$$

Où $\boldsymbol{w}$ est un vecteur à déterminer.

Si $\boldsymbol{w}$ était une des lignes de la matrice inverse $\boldsymbol{A}^{-1}$, cette combinaison linéaire serait en fait égale à une des composantes indépendantes. La question qui se pose maintenant : Comment peut-on utiliser le théorème de la limite centrale pour estimer $\boldsymbol{w}$ de sorte à ce qu'il soit égal à une des lignes de $\boldsymbol{A}^{-1}$? En réalité, on ne peut pas déterminer exactement un tel $\boldsymbol{w}$, car nous n'avons aucune information à propos de la matrice de mélange $\boldsymbol{A}$, mais, on peut trouver un estimateur qui donne une bonne approximation.

Pour voir comment cela mène au principe de base de l'estimation de l'ACI, Faisons un changement de variables, en définissant les nouvelles variables suivantes :

$$\boldsymbol{z} = \boldsymbol{A}^T\boldsymbol{w} \ ,$$

alors nous aurons :

$$\boldsymbol{y} = \boldsymbol{w}^T\boldsymbol{x} = \boldsymbol{w}^T\boldsymbol{A}\boldsymbol{s} = \boldsymbol{z}^T\boldsymbol{s} \qquad (3.7)$$

Il est clair que y est une combinaison linéaire des composantes s_i du vecteur $\boldsymbol{s}$ avec les poids donnés par z_i. Puisque la somme de deux variables seulement est plus gaussienne que les variables d'origine, $\boldsymbol{z}^T\boldsymbol{s}$ est encore plus gaussienne que n'importe quel s_i et devient moins gaussienne quand elle est égale à l'une des s_i. Dans ce cas, il est clair qu'une seule composante z_i de $\boldsymbol{z}$ est non nulle (en supposant que les composantes s_i sont identiquement distribuées).

Donc, nous pouvons prendre comme $\boldsymbol{w}$ un vecteur qui maximise la *non-gaussianité* de $\boldsymbol{w}^T\boldsymbol{x}$. Un tel vecteur, correspondra forcément au vecteur $\boldsymbol{z}$ ayant une seule composante non nulle. Ceci signifie que $\boldsymbol{w}^T\boldsymbol{x} = \boldsymbol{z}^T\boldsymbol{s}$ est égale à une des composantes indépendantes.

Afin d'exploiter la non-gaussianité dans l'estimation d'ACI, nous devons avoir une mesure quantitative de la non-gaussianité d'une variable aléatoire. Pour simplifier les choses, supposons que la variable aléatoire y est centrée et de variance unité, ce qui correspond à la phase de prétraitement vue au chapitre précédent.

a. Kutosis

La méthode classique pour mesurer la non-gaussianité est bien *le cumulant d'ordre quatre*, dit aussi : le *kurtosis*. Le kurtosis de y s'écrit :

$$kurt(y) = E[y^4] - 3(E[y^2])^2 \tag{3.8}$$

Puisque nous avons supposé que y est de variance unité, l'équation (3.8) se simplifie à la suivante :

$$kurt(y) = E[y^4] - 3 \tag{3.9}$$

Ce qui montre que le kurtosis n'est rien qu'une version normalisée du moment d'ordre quatre, $E[y^4]$. Pour y gaussienne, le moment d'ordre quatre égal à $3(E[y^2])^2$, et par conséquent, le kurtosis de y est nul. En contre partie, pour la plupart des variables aléatoires non gaussiennes (mais non pas la totalité), le kurtosis est non nul.

La valeur du kurtosis peut être positive ou négative. Cependant, les variables aléatoires ayant un kurtosis négatif sont appelées *sous-gaussiennes*, en outre, celle possédant un kurtosis positif sont appelées sur-gaussiennes. Les variables sur-gaussiennes sont caractérisées par une pdf étroite avec des queues lourdes. Par contre les variables sous-gaussiennes possèdent une pdf plate.

Un exemple de variable sur-gaussienne, présentée dans la figure (3.1), est celle d'une distribution de *Laplace*, dont la pdf (variance unité) s'écrit :

$$p_y(y) = \frac{1}{\sqrt{2}} exp\left(\sqrt{2}|y|\right) \tag{3.10}$$

L'autre exemple de variable sous-gaussienne est bien celle possédant une pdf uniforme.

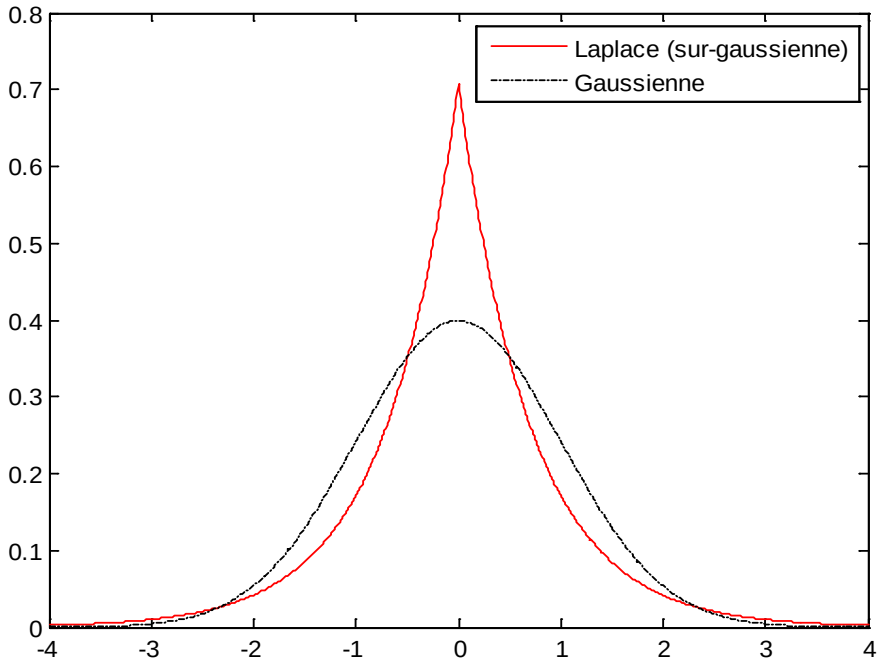


Fig. 3.1. Densité de probabilité de Laplace : densité sur-gaussienne typique. Pour comparaison : en discontinu une densité gaussienne. Les deux densités sont de variances unité.

La non-gaussianité est mesurée en prenant la valeur absolue du kurtosis. Le carré du kurtosis peut être également utilisé. En effet, le kurtosis ou plutôt sa valeur absolue ont été largement utilisés comme une mesure de non-gaussianité en ACI et les domaines d'application en lien. Pour son calcul, le kurtosis peut être estimé en utilisant tout simplement le moment d'ordre quatre des données.

Le kurtosis présente une propriété de linéarité très forte rendant son analyse théorique plus simple. Cette propriété peut être exprimée comme suit :

Si x_1 et x_2 sont deux variables aléatoires indépendantes, alors :

$$kurt(x_1 + x_2) = kurt(x_1) + kurt(x_2) \tag{3.11}$$

Et

$$kurt(\alpha x_1) = \alpha^4 kurt(x_1) \tag{3.12}$$

Où α est un scalaire.

Pour montrer la philosophie de l'application du kurtosis à l'optimisation et comment est ce que les composantes indépendantes sont déterminées, prenons un simple exemple dans le plan bidimensionnel où $\boldsymbol{x} = \boldsymbol{As}$. Supposons que les kurtosis des composantes indépendantes s_1 et s_2 , notés par $kurt(s_1)$ et $kurt(s_1)$ respectivement, sont non nuls.

D'après l'équation (3.7), nous avons :

$$\boldsymbol{y} = \boldsymbol{w}^T\boldsymbol{x} = \boldsymbol{w}^T\boldsymbol{As} = \boldsymbol{z}^T\boldsymbol{s} = z_1 s_1 + z_2 s_2$$

Et d'après les équations (3.11) et (3.12), nous obtenons :

$$kurt(y) = kurt(z_1 s_1 + z_2 s_2) = z_1^4 kurt(s_1) + z_2^4 kurt(s_2) \tag{3.13}$$

Sous contraintes que la variance de y est égale à 1, alors, $\boldsymbol{z}$ est mis sous la contrainte suivante :

$$E[y^2] = z_1^2 + z_2^2 = 1 \tag{3.14}$$

De point de vue géométrique, le vecteur $\boldsymbol{z}$ devrait appartenir au cercle unitaire. Le problème d'optimisation maintenant se pose par la question suivante :

> *Quelles sont les maxima de la fonction* $|kurt(y)| = |z_1^4 kurt(s_1) + z_2^4 kurt(s_2)|$ *appartenant au cercle unitaire* ?

Il n'est pas difficile de montrer [L50] que les maxima sont exactement les points correspondant au cas où une des valeurs du vecteur $\boldsymbol{z}$ est nulle et l'autre non nulle, et ceci est dû à la contrainte (3.13), ce qui force la valeur non nulle de $\boldsymbol{z}$ à prendre la valeur de 1 ou -1. Ces points prennent exactement ces valeurs lorsque y égale une des composantes indépendantes $\pm s_i$, et le problème est donc résolu.

En pratique, nous pourrions commencer à partir de certains vecteur poids $\boldsymbol{w}$, calculer la direction dans laquelle le kurtosis de $y = \boldsymbol{w}^T\boldsymbol{x}$ est croissant (dans le cas où le kurtosis est positif), ou décroissant (dans le cas où le kurtosis est négatif) à partir des données disponibles $\boldsymbol{x}(1), \dots, \boldsymbol{x}(N)$ du mélange, et appliquer une des méthode du gradient ou de ces variantes pour trouver le nouveau vecteur $\boldsymbol{w}$.

Le problème majeur du kurtosis, lorsqu'il doit être estimé à partir de données mesurées, est qu'il peut être très sensible aux perturbations [L51]. Sa valeur peut dépendre d'un petit nombre d'observations dans les queues de la distribution, qui peuvent être erronées ou non pertinentes. Autrement dit, le kurtosis n'est pas une mesure robuste de non-gaussianité. Ainsi, d'autres mesures de non-gaussianité peuvent être meilleurs que le kurtosis dans certaines situations.

b. Néguentropie

Une deuxième mesure très importante de non-gaussianité est donnée par la néguentropie. La néguentropie est basée sur la quantité d'information de l'entropie. L'entropie est le concept de base dans la théorie de l'information. L'entropie d'une variable aléatoire, peut être interprétée comme la quantité moyenne d'information présente dans une observation (réalisation) de cette variable.

L'entropie H d'une variable aléatoire discrète y est définie par :

$$H(y) = -\sum_i p(y = a_i) ln[p(y = a_i)] \tag{3.15}$$

Où les a_i sont les valeurs possibles de y.

Cette définition peut être étendue au cas continu des valeurs et vecteurs aléatoires, et dans ce cas, elle est connue sous le nom d'*entropie différentielle*.

Alors, l'entropie différentielle H d'un vecteur aléatoire $\boldsymbol{y}$ de densité de probabilité $f(\boldsymbol{y})$ est définie par l'expression [B3] :

$$H(\boldsymbol{y}) = -\int f(\boldsymbol{y}) ln[f(\boldsymbol{y})] d\boldsymbol{y} \tag{3.16}$$

$ln(.)$ est le logarithme népérien.

Un résultat fondamental de la théorie de l'information montre qu'une variable gaussienne possède la plus grande valeur d'entropie parmi toutes les variables aléatoires de mêmes variances [B3]. Cela veut dire que l'entropie peut être considérée comme une mesure de non-gaussianité. En effet, la distribution gaussienne est la plus aléatoire et la moins structurée parmi toutes les distributions. L'entropie est petite pour les distributions concentrées autour de certaines valeurs, i.e. quand la variable est clairement groupée, ou ayant une pdf très étroite.

Afin d'aboutir à une mesure de non-gaussianité qui soit zéro pour les variables gaussiennes et toujours positive, on utilise souvent une version légèrement modifiée de l'entropie différentielle, appelée *néguentropie*, qui s'écrit comme suit :

$$J(\boldsymbol{y}) = H(\boldsymbol{y}_{\boldsymbol{gauss}}) - H(\boldsymbol{y}) \tag{3.17}$$

Où $\boldsymbol{y}_{\boldsymbol{gauss}}$ est gaussien de même matrice de covariance que $\boldsymbol{y}$.

La néguentropie est toujours positive et s'annule pour des variables gaussiennes. Une propriété supplémentaire très intéressante de la néguentropie, réside dans le fait qu'elle est invariante pour des transformations linéaires inversibles [L52, L45].

En fait, néguentropie est en quelque sorte l'estimateur optimal de non-gaussianité. Le problème dans l'usage de la négentropie est, cependant, son calcul très difficile. Dans ce cas, l'estimation de la néguentropie en utilisant sa définition, pourrait nécessiter un estimateur (probablement non-paramétrique) de la pdf. Hors que, d'autres approximations plus simple peuvent être très utiles.

- **Approximations de la néguentropie**

Comme mentionné ci-dessus, l'estimation de la néguentropie est très difficile. En pratique, une méthode classique pour une telle estimation est celle utilisant les moments d'ordre supérieurs. Par exemple, Jones et Sibson [L53] ont proposé l'estimateur suivant :

$$J(y) \approx \frac{1}{12}(E[y^3])^2 + \frac{1}{48}\big(kurt(y)\big)^2 \tag{3.18}$$

La variable y est supposée être de moyenne nulle et de variance égale à 1. Malheureusement, cette approximation est limitée, et souffre en particulier de la non robustesse causé par le kurtosis.

Pour pallier le problème de (3.18), d'autres estimateurs ont été développées par Hyvärinen [L54]. Cette approximation est basée sur le principe de maximum-entropie. En général, l'estimation s'écrit :

$$J(y) \approx \sum_{i=1}^{p} k_i \{E[G_i(y)] - E[G_i(v)]\}^2 \qquad (3.19)$$

Où k_i sont des constantes positives, et v une variable gaussienne normale (de moyenne nulle et de variance unité).

y est aussi supposée de moyenne nulle et de variance unité, et les G_i sont quelques fonctions non quadratiques. Notons que même dans les cas où cette approximation n'est pas très précise, (3.19) peut être utilisée pour construire une mesure consistante de la non-gaussianité du fait qu'elle est toujours positive, et s'annule lorsque y est gaussienne.

Quand une seule fonction non quadratique G est utilisée, l'équation (3.19) devient :

$$J(y) \propto \{E[G(y)] - E[G(v)]\}^2 \qquad (3.20)$$

Pour pratiquement n'importe quelle fonction non quadratique G, il est clair que lorsque y est symétrique, (3.20) est une généralisation de l'approximation (3.18).

En prenant $G(y) = y^4$, on obtient exactement l'expression (3.18).

Avec un bon choix de G, on peut avoir une approximation meilleure que celle obtenue par (3.18).en particulier, lorsque G ne varie pas trop vite, on peut obtenir

un estimateur robuste. Un des choix de G, très rencontré en littérature, est donné par les expressions suivantes :

$$G_1(y) = \frac{1}{a_1} log[cosh(a_1 y)], \qquad G_2(y) = -exp\left(-\frac{y^2}{2}\right) \tag{3.21}$$

Où : $1 \leq a_1 \leq 2$ une constante.

3.2.2. Fonctions de Contraste

Le concept des fonctions de contraste pour la séparation de sources a été introduit par Comon [L45]. Une fonction de contraste, qui peut être vue comme une mesure d'indépendance, constitue un critère de séparation dans la mesure où sa maximisation résout le problème de séparation. Elle est définie de la façon suivante :

Une foction de contraste $\Psi(\,.\,)$ est une application à valeurs dans $\mathcal{R}$ définie sur l'espace de vecteurs aléatoires $\boldsymbol{y}$ de $\mathcal{R}^p$, ne dépendant que de la loi de probabilité de $\boldsymbol{y}$ et qui vérifie les propriétés suivantes :

- pour toute matrice de permutation $\boldsymbol{P}$

$$\Psi(\boldsymbol{Py}) = \Psi(\boldsymbol{y}) \tag{3.22}$$

- pour toute matrice diagonale $\boldsymbol{D}$

$$\Psi(\boldsymbol{Dy}) = \Psi(\boldsymbol{y})$$

- pour tout vecteur $\boldsymbol{x}$, de composantes indépendantes, et pour toute matrice $\boldsymbol{S}$ on a :

$$\Psi(\boldsymbol{Sy}) \leq \Psi(\boldsymbol{y}) \tag{3.23}$$

et

$$\Psi(\boldsymbol{Sy}) = \Psi(\boldsymbol{y}) \Longleftrightarrow \boldsymbol{S} = \boldsymbol{DP} \tag{3.24}$$

Où $\boldsymbol{D}$ est une matrice diagonale et $\boldsymbol{P}$ est une permutation quelconques.

Dans [L45], Comon a proposé entre autres, la maximisation d'un contraste défini comme la somme des modules au carré des kurtosis des sources estimées. Par ailleurs, Moreau et Pesquet [L55], ont introduit une classe de contrastes applicables aux mélanges convolutifs de sources centrées, indépendantes et identiquement distribuées (i.i.d), statistiquement mutuellement indépendantes et vérifiant certaines propriétés. De son côté Comon trouve une solution analytique, nommée COM1 (COntrast Maximization 1) [L58], au problème d'optimisation du contraste défini, au signe près, comme la somme des kurtosis des sources estimées. Ce contraste a été initialement présenté par Moreau *et al.* dans [L56, L57]. En outre, Moreau montre que ce critère est un contraste à la condition que les kurtosis des sources soient de même signe.

3.2.3. Maximum de Vraisemblance

L'objectif est de chercher les paramètres du mélange qui maximisent la probabilité d'occurrence des observations. Dans un mélange linéaire instantané, où le vecteur mélange $\boldsymbol{x}$ s'écrit $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$, on peut exprimer la pdf de $\boldsymbol{y}$ en fonction des pdfs des sources $\boldsymbol{s}$ et du déterminant de la matrice de séparation $\boldsymbol{B}$. i.e.

$$f_{\boldsymbol{x}}(\boldsymbol{x}) = |det(\boldsymbol{B})| f_{\boldsymbol{s}}(\boldsymbol{s}) = |det(\boldsymbol{B})| \prod_{i=1}^{p} f_{s_i}(s_i) \tag{3.29}$$

$f_{\boldsymbol{x}}(\boldsymbol{x})$ peut être aussi exprimée en fonction des lignes $\boldsymbol{b}_i$ de la matrice $\boldsymbol{B}$ par la relation :

$$f_{\boldsymbol{x}}(\boldsymbol{x}) = |det(\boldsymbol{B})| \prod_{i=1}^{p} f_{s_i}(\boldsymbol{b}_i \boldsymbol{x}) \tag{3.30}$$

Si nous prenons maintenant l'ensemble des N échantillons indépendants et de même loi, du vecteur $\boldsymbol{x}$, i.e. $\boldsymbol{x}(1), \boldsymbol{x}(2), \dots, \boldsymbol{x}(N)$. La vraisemblance $\mathcal{L}$ de l'obtention de cet ensemble peut s'écrire comme le produit de l'équation (3.30) évaluée à ces N points.

$$\mathcal{L}(\boldsymbol{B}) = \prod_{n=1}^{N}\left[|det(\boldsymbol{B})| \prod_{i=1}^{p} f_{s_i}\left(\boldsymbol{b}_i \boldsymbol{x}(n)\right)\right] \tag{3.31}$$

Il est souvent plus pratique d'utiliser le logarithme de la vraisemblance. On obtient donc,

$$ln\{\mathcal{L}(\boldsymbol{B})\} = \sum_{n=1}^{N}\sum_{i=1}^{p} ln\left\{f_{s_i}\left(\boldsymbol{b}_i \boldsymbol{x}(n)\right)\right\} + N\, ln(|det(\boldsymbol{B})|) \tag{3.32}$$

En divisant par le nombre d'échantillons N, nous avons,

$$\frac{1}{N} ln\{\mathcal{L}(\boldsymbol{B})\} = \sum_{i=1}^{p}\frac{1}{N}\sum_{n=1}^{N} ln\left\{f_{s_i}\left(\boldsymbol{b}_i \boldsymbol{x}(n)\right)\right\} + ln(|det(\boldsymbol{B})|) \tag{3.33}$$

D'autre part, par la loi des grands nombres, on a
$\frac{1}{N}\sum_{n=1}^{N} ln\left\{f_{s_i}\left(\boldsymbol{b}_i \boldsymbol{x}(n)\right)\right\} \approx E\left[ln\left\{f_{s_i}(\boldsymbol{b}_i \boldsymbol{x})\right\}\right]$ Lorsque N est suffisamment grand. D'où,

$$\frac{1}{N} ln\{\mathcal{L}(\boldsymbol{B})\} \approx \sum_{i=1}^{p} E\left[ln\left\{f_{s_i}(\boldsymbol{b}_i \boldsymbol{x})\right\}\right] + ln(|det(\boldsymbol{B})|) \tag{3.34}$$

Ce qui donne,

$$\frac{1}{N} ln\{\mathcal{L}(\boldsymbol{B})\} \approx -\sum_{i=1}^{p} H(\boldsymbol{b}_i \boldsymbol{x}) + ln(|det(\boldsymbol{B})|) \coloneqq \sum_{i=1}^{p} H(y_i) + ln(|det(\boldsymbol{B})|) \tag{3.35}$$

De nombreuses contributions exploitant le maximum de vraisemblance pour la SAS ont été présentées, citons par exemple : Belouchrani et Cardoso [L61, L62], Cardoso [L63], Amari [L64], et Moulines *et al.* [L65]. Il est important de noter que Gaeta et Lacoume [L66, L67] sont les premiers à avoir montré la possibilité d'utiliser le principe du maximum de vraisemblance dans le problème de séparation de sources pour le cas de mélanges linéaires instantanés ou convolutifs. Pham *et al.* [L68, L12] ont aussi utilisé le principe du maximum de vraisemblance dans la résolution du problème de la SAS.

3.2.4. Information Mutuelle

L'information mutuelle (IM) est un critère permettant la mesure de la dissimilarité entre la densité conjointe d'un ensemble de signaux et le produit des densités marginales de chacune de ses composantes. Pour un vecteur aléatoire, $\boldsymbol{y} = \left(y_1, \dots, y_p\right)^T$, l'IM $I(\boldsymbol{y})$, s'écrit :

$$I(\boldsymbol{y}) = \int_{\mathcal{R}^p} f_{\boldsymbol{y}}(\boldsymbol{y}) ln \frac{f_{\boldsymbol{y}}(\boldsymbol{y})}{\prod_{i=1}^{p} f_{y_i}(y_i)} d\boldsymbol{y} \tag{3.25}$$

$f_{\boldsymbol{y}}(\boldsymbol{y})$ est la densité jointe du vecteur aléatoire $\boldsymbol{y}$, et $f_{y_i}(y_i)$ la densité marginale des variable aléatoires y_i , $i = 1, \dots, p$. La divergence de *Kullback-Leibler* entre les deux densités $f_{\boldsymbol{y}}(\boldsymbol{y})$ et $\prod_{i=1}^{p} f_{y_i}(y_i)$ est définie par :

$$\begin{aligned} D_{KL}\left(f_{\boldsymbol{y}}(\boldsymbol{y}) \middle| \prod_{i=1}^{p} f_{y_i}\right) &= -\int_{\mathcal{R}^p} f_{\boldsymbol{y}}(\boldsymbol{y}) ln \frac{\prod_{i=1}^{p} f_{y_i}(y_i)}{f_{\boldsymbol{y}}(\boldsymbol{y})} d\boldsymbol{y} \\ &= \int_{\mathcal{R}^p} f_{\boldsymbol{y}}(\boldsymbol{y}) ln \frac{f_{\boldsymbol{y}}(\boldsymbol{y})}{\prod_{i=1}^{p} f_{y_i}(y_i)} d\boldsymbol{y} \end{aligned} \tag{3.26}$$

D'où la relation suivante entre l'information mutuelle et la divergence de *Kullback-Leibler* :

$$I(\boldsymbol{y}) = D_{KL}\left(f_{\boldsymbol{y}}(\boldsymbol{y}) \middle| \prod_{i=1}^{p} f_{y_i}\right) \tag{3.27}$$

L'IM peut être aussi exprimée en fonction de l'entropie, H (équation 3.16).

$$\int_{\boldsymbol{y}} f_{\boldsymbol{y}}(\boldsymbol{y}) ln \frac{f_{\boldsymbol{y}}(\boldsymbol{y})}{\prod_{i=1}^{p} f_{y_i}(y_i)} d\boldsymbol{y} = \int_{\boldsymbol{y}} f_{\boldsymbol{y}}(\boldsymbol{y}) ln\, f_{\boldsymbol{y}}(\boldsymbol{y}) d\boldsymbol{y} - \int_{\boldsymbol{y}} f_{\boldsymbol{y}}(\boldsymbol{y}) ln \prod_{i=1}^{p} f_{y_i}(y_i)\, d\boldsymbol{y}$$

$$= \sum_i H(y_i) - H(\boldsymbol{y}) \tag{3.28}$$

La divergence de *Kullback-Leibler* entre deux lois de probabilité f et g possède les propriétés suivantes :

1. $D_{KL}(f|g) \geq 0$, et s'annule seulement quand $f = g$,
2. $D_{KL}(f|g)$ est invariante pour les transformations sur $\boldsymbol{y}$, suivantes :
   a. Permutation des éléments de $\boldsymbol{y}$
   b. Transformation nonlinéaires monotones

Ces propriétés font de l'information mutuelle un critère d'indépendance très intéressant pour le problème de la SAS. Beaucoup de travaux ont fait l'objet du problème de minimisation de l'IM dans le cadre de la SAS et l'ACI [NL13, L16, L59, NL3, L60]. En effet, la SAS basée sur la minimisation de l'IM tend asymptotiquement à la méthode du maximum de vraisemblance (ML : Maximum Likelihood) [L16, Th2].

3.3. CONCLUSION

Dans ce chapitre, nous avons exposé les plus importants critères d'indépendance utilisés en SAS. En effet, afin de restituer les signaux sources en se basant sur l'hypothèse maitresse, indépendance statistique des sources, nous devons faire appel à l'analyse en composantes indépendantes, et dans ce cas, il est indispensable de trouver une mesure d'indépendance statistique. Cependant, la non-gaussianité, l'information mutuelle, et le maximum de vraisemblance, peuvent constituer une mesure d'indépendance statistique à partir de laquelle le problème de la SAS peut être résolu.

CHAPITRE 4

SAS basée sur l'IM

4.1. INTRODUCTION

Nous avons vu dans le chapitre précédent que l'information mutuelle (IM) entre deux variables aléatoires n'est rien que la divergence de *Kullback-Leibler* entre leurs densités de probabilité. Cette approche possède plusieurs avantages par rapport aux autres approches. Parmi ces avantages nous citons :

1. Pour les mélanges linéaires instantanés, la qualité de séparation par l'IM converge asymptotiquement au maximum de vraisemblance (ML) [L16]
2. Contrairement à quelques critères d'indépendance (cumulant s'ordre 4, par exemple), L'IM n'a aucune approximation, i.e. s'annule si et seulement si les signaux sont indépendants. Par conséquent, elle peut être utilisée pour des types de mélanges plus complexes (mélanges nonlinéaires par exemple)
3. L'IM peut être l'origine d'autres approches plus améliorés pour la séparation de différents autres modèles séparables [NL28]

4. Dans [NL28], les auteurs ont montré que l'IM n'a pas de minima locaux.

Pour la minimisation de l'IM pour les deux types de mélanges : linéaires instantanés et post nonlinéaires, nous avons besoin de calculer l'expression analytique de son gradient. Cependant, une des méthodes du calcul du gradient de l'IM passe par l'estimation des fonctions score [L32].

Dans ce qui suit nous présentons la notion de fonctions score, leurs propriétés les plus importantes, exemples de méthodes de leur estimation, et les principes de calcul du gradient de l'IM pour les deux types de mélanges : linéaire instantané et post-nonlinéaire.

4.2. FONCTIONS SCORE

Les fonctions score sont utiles pour l'évaluation de l'expression du gradient de l'IM. Cependant, dans ce qui suit nous présentons les différents types des fonctions score.

4.2.1. Fonction Score d'une variable aléatoire

Pour une variable aléatoire x, la fonction score $\psi_x(x)$ est définie par l'expression suivante :

$$\psi_x(x) \triangleq -\frac{d\ ln(p_x(x))}{d\ x} = -\frac{\acute{p}_x(x)}{p_x(x)} \qquad (4.1)$$

Où : $p_x(x)$ est la pdf de x, et $\acute{p}_x(x)$ sa dérivée par rapport à x.

4.2.2. Fonction Score Marginale (MSF)

Par définition, la fonction score marginale d'un vecteur aléatoire, $\boldsymbol{x} = \left(x_1, \dots, x_p\right)^{\mathrm{T}}$, est le vecteur des fonctions score de ses composantes. i.e.

$$\boldsymbol{\psi_x}(\boldsymbol{x}) \triangleq \left(\psi_{x_1}(x_1), \dots, \psi_{x_p}\left(x_p\right)\right)^T$$

Où :

$$\psi_{x_i}(x_i) \triangleq -\frac{d\ ln\left(p_{x_i}(x_i)\right)}{d\ x_i} = -\frac{\dot{p_{x_i}}(x_i)}{p_{x_i}(x_i)} \tag{4.2}$$

4.2.3. Fonction Score Conjointe (JSF)

La fonction score conjointe (Joint Score Function, JSF), $\boldsymbol{\varphi_x}$, du vecteur aléatoire $\boldsymbol{x}$, est le gradient de la fonction : $-ln\left(p_{\boldsymbol{x}}(\boldsymbol{x})\right)$, i.e.

$$\boldsymbol{\varphi_x}(\boldsymbol{x}) = \left(\varphi_1(\boldsymbol{x}), \dots, \varphi_p(\boldsymbol{x})\right)^T \tag{4.3}$$

Où :

$$\varphi_i(\boldsymbol{x}) \triangleq -\frac{\partial\ ln\left(p_{\boldsymbol{x}}(\boldsymbol{x})\right)}{\partial\ x_i} = -\frac{\frac{\partial}{\partial x_i}p_{\boldsymbol{x}}(\boldsymbol{x})}{p_{\boldsymbol{x}}(\boldsymbol{x})} \tag{4.4}$$

4.2.4. Différence des Fonctions Score (SFD)

On définit la différence des fonctions score, $\boldsymbol{\beta_x}$, du vecteur aléatoire $\boldsymbol{x}$, comme la différence entre la fonction score marginale et la fonction score conjointe du même vecteur x. I.e. :

$$\boldsymbol{\beta_x}(\boldsymbol{x}) \triangleq \boldsymbol{\psi_x}(\boldsymbol{x}) - \boldsymbol{\varphi_x}(\boldsymbol{x}) \tag{4.5}$$

Cette fonction score contient beaucoup d'information relative à l'indépendance statistique des éléments du vecteur aléatoire $\boldsymbol{x}$.

4.2.5. PROPRIÉTÉS DE LA SFD

Comme déjà mentionné ci-dessus, la SFD possède plusieurs propriétés qui peuvent être exploitées dans le cadre de l'estimation des composantes indépendantes d'un vecteur aléatoire.

Propriété 1

Les Composantes du vecteur $\boldsymbol{x} = (x_1, \dots, x_p)^T$ sont indépendantes si et seulement si $\boldsymbol{\beta_x}(\boldsymbol{x}) \equiv \mathbf{0}$, i.e. :

$$\boldsymbol{\varphi_x}(\boldsymbol{x}) = \boldsymbol{\psi_x}(\boldsymbol{x}) \tag{4.6}$$

Propriété 2

Pour un vecteur aléatoire, $\boldsymbol{x} = (x_1, \dots, x_p)^T$, nous avons :

$$\beta_i(\boldsymbol{x}) = \frac{\partial}{\partial x_i} ln\left[p\left(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_p | x_i\right)\right] \tag{4.7}$$

telle que : $\beta_i(\boldsymbol{x})$ est la i$^{\text{ème}}$ composante de $\boldsymbol{\beta_x}(\boldsymbol{x})$.

Par exemple, dans le cas bidimensionnelle, $\boldsymbol{x} = (x_1,\ x_2)^T$, nous avons :

$$\boldsymbol{\beta_x}(\boldsymbol{x}) = [\beta_1(x_1, x_2),\ \beta_2(x_1, x_2)]^T$$

Donc, à partir de (4.7) nous pouvons écrire :

$$\beta_1(x_1, x_2) = \frac{\partial}{\partial x_1} ln[p(x_2|x_1)] \tag{4.8}$$

$$\beta_2(x_1, x_2) = \frac{\partial}{\partial x_2} ln[p(x_1|x_2)] \tag{4.9}$$

Autrement dit, $\beta_1(x_1, x_2) = 0$, $si\ \ p(x_2|x_1)$ ne dépend pas de x_1 , (x_1 et x_2 sont indépendants). Dans le cas de dimension p, $\beta_i(\boldsymbol{x}) = \mathbf{0}$ si x_i est indépendante des autres composantes de $\boldsymbol{x}$, i.e.

$$p\big(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_p|x_i\big) = p\big(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_p\big) \tag{4.10}$$

Ou bien

$$p(\boldsymbol{x}) = p\big(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_p\big)p(x_i) \tag{4.11}$$

<u>Propriété 3</u>

Soit $\boldsymbol{x}$ un vecteur de densité $p_{\boldsymbol{x}}(\boldsymbol{x})$ et une JSF $\boldsymbol{\varphi_x}(\boldsymbol{x})$, et si $f(\boldsymbol{x})$ est une fonction multivariable dont les dérivées partielles $\frac{\partial f}{\partial x_i}(\boldsymbol{x})$ sont continues, et si

$$\lim_{x_i \to \pm\infty} \int_{x_1,\dots,x_{i-1},x_{i+1},\dots,x_p} f(\boldsymbol{x})\, p_{\boldsymbol{x}}(\boldsymbol{x}) dx_1, \dots, dx_{i-1}, dx_{i+1}, \dots, dx_p = 0 \tag{4.12}$$

Alors nous avons,

$$E\{f(\boldsymbol{x})\varphi_i(\boldsymbol{x})\} = E\left\{\frac{\partial f}{\partial x_i}(\boldsymbol{x})\right\} \tag{4.13}$$

Il est à noter que l'équation (4.12) est vraie pour la plupart des signaux physiques où $p_{\boldsymbol{x}}(\boldsymbol{x})$ décroit rapidement quand $\|\boldsymbol{x}\|$ tend vers l'infini. En effet, la majorité des signaux réels sont bornés, ce qui vérifie, donc, l'équation (4.13).

Propriété 4

Pour un vecteur aléatoire, $\boldsymbol{x} = \left(x_1, \dots, x_p\right)^T$ nous avons :

$$\psi_i(x) = E\{\varphi_i(\boldsymbol{x}) | x_i = x\} \tag{4.14}$$

Sans perte de généralité, prenons $(i = 1)$, pour démontrer la relation (4.14). Dans ce cas, nous écrivons :

$$\begin{aligned} E\{\varphi_1(\boldsymbol{x})|x_1\} &= \int_{x_2,\dots,x_p} \varphi_1(\boldsymbol{x}) p\left(x_2, \dots, x_p | x_1\right) dx_2 \dots dx_p \\ &= -\int_{x_2,\dots,x_p} \frac{\frac{\partial p_{\boldsymbol{x}}(x)}{\partial x_1}}{p_{\boldsymbol{x}}(\boldsymbol{x})} \frac{p_{\boldsymbol{x}}(\boldsymbol{x})}{p_{x_1}(x_1)} dx_2 \dots dx_p \\ &= -\frac{1}{p_{x_1}(x_1)} \frac{\partial}{\partial x_1} \int_{x_2,\dots,x_p} p_{\boldsymbol{x}}(\boldsymbol{x})\, dx_2 \dots dx_p \\ &= -\frac{1}{p_{x_1}(x_1)} \frac{\partial}{\partial x_1} p_{x_1}(x_1) \\ &= \psi_1(x_1) \end{aligned} \tag{4.15}$$

Ce qui démontre l'équation (4.14). □

A partir de la propriété (1), nous savons que si la composante, x_i , du vecteur $\boldsymbol{x}$, est indépendante des autres variables alors $\varphi_i(x_i) = \psi_i(x_i)$. Cependant, la propriété (4) montre que si les autres variables dépendent de x_i , $\varphi_i(x_i)$ n'est plus égale à $\psi_i(x_i)$, mais sa moyenne reste toujours égale à $\psi_i(x_i)$. C'est-à-dire que la dépendance statistique peut entrainer des fluctuations dans $\varphi_i(x_i)$, mais qui sont autour de sa moyenne. L'exemple suivant montre l'utilité de la propriété (4).

- **Exemple**

Soit s_1 et s_2 deux variables aléatoires indépendantes uniformément distribuées sur l'intervalle [-0.5, 0.5]. On définit les variables aléatoires x_1 et x_2 par :

$$\begin{cases} x_1 = s_1 \\ x_2 = s_2 + ks_1 \end{cases} \tag{4.16}$$

Pour $k = 0$, x_1 et x_2 sont indépendantes. Mais, si on varie k, $\psi_1(x_1)$ ne change pas puisqu'elle ne dépend pas de k, tandis que, $\varphi_1(x_1)$ varie autour de $\psi_1(x_1)$, qui est sa moyenne constante. La figure (4.1), montre $\varphi_1(x_1)$ et $\psi_1(x_1)$ estimées pour $k = 0.5$.

Dans ce cas, on peut dire que la SFD est, en fait, une mesure des variations de la JSF autour de sa moyenne (valeur lissée : smoothed value).

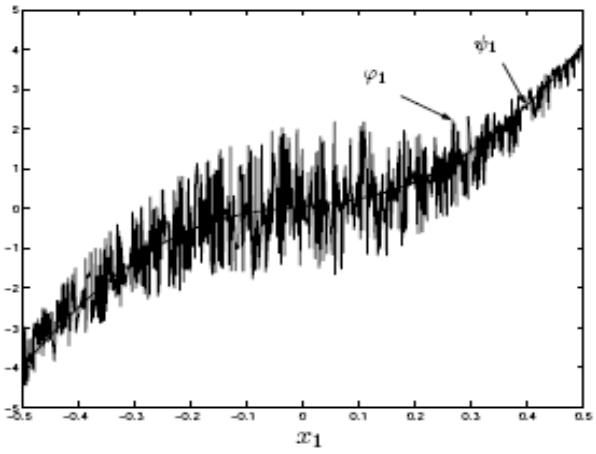


Fig. 4.1. $\varphi_1(x_1)$ et $\psi_1(x_1)$ en cas de dépendance statistique.

Propriété 5

Pour $= \boldsymbol{Bx}$, où $\boldsymbol{x}$ et $\boldsymbol{y}$ sont deux vecteurs aléatoires, et $\boldsymbol{B}$ une matrice carrée non-singulière, alors

$$\boldsymbol{\varphi_y(y) = B^{-T}\varphi_x(x)} \tag{4.17}$$

4.3. ESTIMATION DES FONCTIONS SCORE

Nous avons vu dans les sections précédentes, que le gradient de l'IM est équivalent à la différence des fonctions score (SFD). Cependant, pour évaluer cette SFD, nous avons besoin d'estimer les deux fonctions score : MSF et JSF.

4.3.1. Estimation de JSF

Dans ses travaux de doctorat, Babaie-Zadeh [TH2, NL40] a présenté deux méthodes d'estimation de $\boldsymbol{\varphi_x(x)}$. la première méthode est l'estimateur à noyau. Dans la deuxième méthode, il suppose que $\varphi_i(\boldsymbol{x})$ est une combinaison linéaire de fonctions multivariables.

- **Estimateur à noyau**

Dans cette méthode, on définit un noyau multivariable de la forme :$k(\boldsymbol{x}) = k(x_1, \dots, x_p)$ comme une pdf d'un vecteur aléatoire à moyenne nulle. Dans ce cas, $k(\boldsymbol{x})$ est dit un noyau si et seulement s'il vérifie les conditions suivantes :

a) $\forall \boldsymbol{x} \in \mathcal{R}^p, k(\boldsymbol{x}) \geq 0$

b) $\int_{\mathcal{R}^p} k(\boldsymbol{x}) d\boldsymbol{x} = 1$

c) $\int_{\mathcal{R}^p} \boldsymbol{x}\, k(\boldsymbol{x})\, d\boldsymbol{x} = 0$

En effet, cet estimateur est désigné pour l'estimation de la pdf des observations à partir de combinaisons linéaires des fonctions noyau. Ces noyaux sont centrés aux valeurs des échantillons.

Les noyaux sont souvent des densités de probabilités symétriques et unimodales. Pour une variable aléatoire X, l'estimateur à noyau de $p_x(x)$ à partir des observations $\{x_1, \dots, x_N\}$ s'écrit comme suit:

$$\hat{p}_x(x) = \frac{1}{N}\sum_{i=1}^{N} k_h(x - x_i) = \frac{1}{Nh}\sum_{i=1}^{N} k\left(\frac{x - x_i}{h}\right) \qquad (4.18)$$

Où h est la largeur du noyau (appelé aussi : paramètre de lissage)
$k(.)$ n'importe quelle pdf symétrique par rapport à zéro, et $k_h(x) = \frac{1}{h}k(x/h)$

Les noyaux les plus utilisés sont :

- *Noyau d'Epanechnikov*

$$k(x) = \frac{3}{4}(1 - x^2) \qquad |x| \leq 1$$

- *Noyau à pondération triple (triweight)*

$$k(x) = \frac{35}{32}(1 - x^2)^3 \qquad |x| \leq 1$$

- *Noyau Gaussien*

$$k(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \qquad -\infty < x < +\infty$$

D'autres noyaux peuvent aussi être utilisés tels que : noyau uniforme, noyau triangulaire, etc. La figure (4.2) présente l'allure de quelques noyaux monovariable.

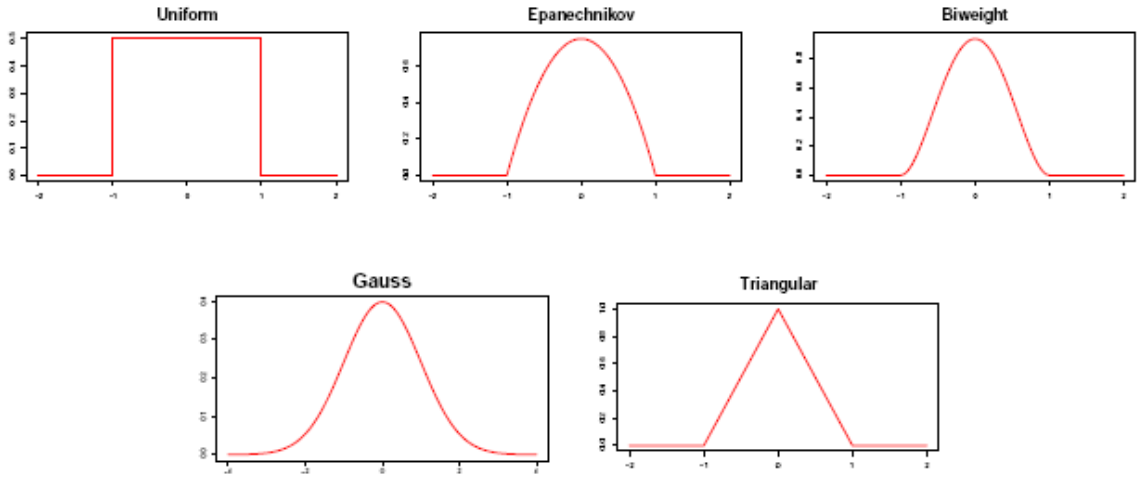


Fig. 4.2. Exemples de Noyaux

Dans le cas multivariable, le noyau gaussien de dimension p s'écrit :

$$k(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi h^p}} e^{-\frac{1}{h^2}\boldsymbol{x}^T\boldsymbol{x}} \tag{4.19}$$

Et dans ce cas l'estimateur de la pdf à partir des observations blanchies $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ est donné par :

$$\hat{p}_{\boldsymbol{x}}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} k_h(\boldsymbol{x} - \boldsymbol{x}_i) \tag{4.20}$$

Le rôle du blanchiment des observations, est de considérer le même h dans toutes les directions $\boldsymbol{x}_i$, et par conséquent, l'estimateur est dit isotropique.

La figure (4.3) montre l'idée derrière l'estimation à noyau de la densité de probabilité dans le cas monovariable en utilisant le noyau gaussien ainsi que l'effet du choix du paramètre de lissage h. Il est à noter que la forme du noyau n'est pas très déterminante pour la qualité de l'estimation contrairement à la valeur de h . en fait, si h est choisi trop petit, la pdf estimée sera trop fluctuante, et s'il est choisi trop grand, on aura une mauvaise forme de la pdf. Donc, il faut choisir une valeur qui vérifie certains critères d'optimalité.

Pour des données blanchies, une expression heuristique de h est donnée par :

$$h = c\, N^{-1/(p+4)} \tag{4.21}$$

c est une constante qui dépend du type du noyau utilisé. Pour un noyau gaussien l'expression de c est donnée par :

$$c = \left(\frac{4}{2p+1}\right)^{\frac{1}{p+4}} \tag{4.22}$$

Alors,

Pour $p = 1$, $c = 1.06$, alors :

$$h \approx 1.06\sigma_x N^{-1/5} \tag{4.23}$$

où σ_x est l'écart-type de X.

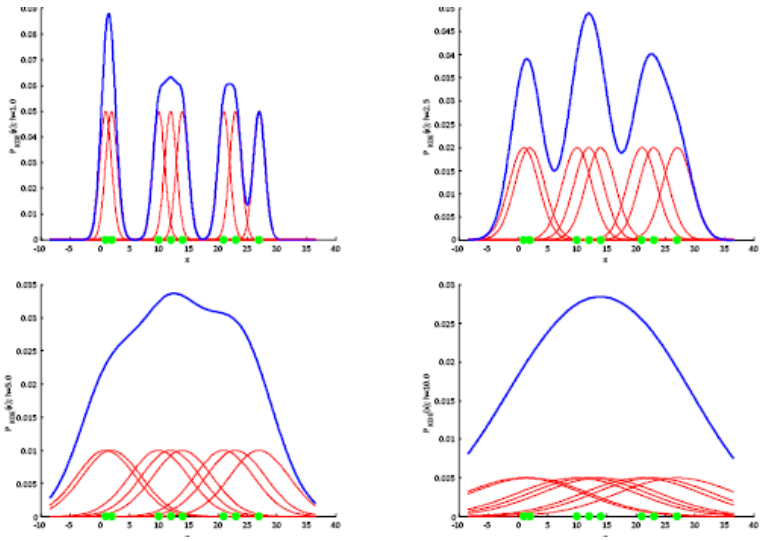


Fig.4.3. Estimateur à noyau de densité de probabilité. Influence du choix de h.

Puisque l'estimation des fonctions score est liée à l'estimation des pdf des observations comme montré par l'équation (4.4), l'estimateur à noyau de la i^{ème} composante $\varphi_i(\boldsymbol{x})$ est donné par:

$$\hat{\varphi}_i(\boldsymbol{x}) \overset{\text{def}}{=} -\frac{\frac{\partial}{\partial x_i}\hat{p}_{\boldsymbol{x}}(\boldsymbol{x})}{\hat{p}_{\boldsymbol{x}}(\boldsymbol{x})} = -\frac{\sum_{r=1}^{N}\frac{\partial}{\partial x_i}k(\boldsymbol{x}-\boldsymbol{x}_r)}{\sum_{r=1}^{N}k(\boldsymbol{x}-\boldsymbol{x}_r)} \tag{4.24}$$

D'autres estimateurs peuvent être utilisés pour l'estimation des pdf, tels que : l'histogramme, l'estimateur naïf, …etc.

- **Estimateur à Erreur Quadratique Moyenne Minimale (MMSE)**

Cette estimateur est basée sur propriété 3, en considérant une modélisation paramétrique de $\varphi_i(\boldsymbol{x})$. dans ce cas, $\varphi_i(\boldsymbol{x})$ est mise sous forme de combinaison linéaire de fonctions multivariables $\{k_1(\boldsymbol{x}), \dots, k_r(\boldsymbol{x})\}$, i.e.

$$\hat{\varphi}_i(\boldsymbol{x}) = \sum_{j=1}^{r} w_j k_j(\boldsymbol{x}) = \boldsymbol{k}^T(\boldsymbol{x})\boldsymbol{w} \tag{4.25}$$

Où, $\boldsymbol{k}^T(\boldsymbol{x}) = \big(k_1(\boldsymbol{x}), \dots, k_r(\boldsymbol{x})\big)$ et $\boldsymbol{w} = (w_1, \dots, w_r)^T$

Le vecteur des paramètres,$\boldsymbol{w}$, peut être obtenu en minimisant le critère d'erreur suivant :

$$\epsilon = E\{[\varphi_i(\boldsymbol{x}) - \hat{\varphi}_i(\boldsymbol{x})]^2\} \tag{4.26}$$

A partir du principe d'orthogonalité [B1], nous avons :

$$E\{\boldsymbol{k}(\boldsymbol{x})[\varphi_i(\boldsymbol{x}) - \hat{\varphi}_i(\boldsymbol{x})]\} = 0 \tag{4.27}$$

En utilisant propriété 3 (équation 4.23) on obtient

$$E\{\boldsymbol{k}(\boldsymbol{x})\boldsymbol{k}^T(\boldsymbol{x})\}\boldsymbol{w} = E\left\{\frac{\partial \boldsymbol{k}}{\partial x_i}(\boldsymbol{x})\right\} \tag{4.28}$$

4.3.2. Estimation de SFD

La fonction score différentielle peut être estimées à partir des estimées de MSF et JSF indépendamment, i.e. estimer MSF et estimer JSF puis estimer la SFD par la relation :

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{x}}(\boldsymbol{x}) = \widehat{\boldsymbol{\psi}}_{\boldsymbol{x}}(\boldsymbol{x}) - \widehat{\boldsymbol{\varphi}}_{\boldsymbol{x}}(\boldsymbol{x}) \tag{4.29}$$

Mais, la minimisation de $E\{[\varphi_i(\boldsymbol{x}) - \hat{\varphi}_i(\boldsymbol{x})]^2\}$ et $E\left\{\left[\psi_i(\boldsymbol{x}) - \hat{\psi}_i(\boldsymbol{x})\right]^2\right\}$ indépendamment, ne minimise forcément pas $E\left\{\left[\beta_i(\boldsymbol{x}) - \hat{\beta}_i(\boldsymbol{x})\right]^2\right\}$. Donc, il est important d'estimer $\boldsymbol{\beta}_{\boldsymbol{x}}(\boldsymbol{x})$ directement. Une méthode pour ce faire, est d'estimer MSF à partir du lissage de l'estimée de JSF et puis utiliser (4.29) pour l'estimation de SFD.

Dans la section suivante, nous allons présenter les principes de minimisation de l'information mutuelle basés sur l'estimation des fonctions score dans le cadre de séparation aveugle de sources.

4.4. MINIMISATION DE L'IM

Dans ce qui suit nous allons voir comment est ce que la minimisation de l'IM peut être utilisée pour l'estimation et la séparation de mélanges instantanés linéaires et post nonlinéaires.

4.4.1. Mélanges linéaires instantanés

Comme montré dans le chapitre précédent, pour un vecteur aléatoire $\boldsymbol{y} = \left(y_1, \ldots, y_p\right)^T$, l'IM est donnée par l'expression

$$\begin{aligned} I(\boldsymbol{y}) &= \int_{\boldsymbol{y}} p_{\boldsymbol{y}}(\boldsymbol{y}) ln\left\{\frac{p_{\boldsymbol{y}}(\boldsymbol{y})}{\prod_{i=1}^{p} p_{y_i}(y_i)}\right\} d\boldsymbol{y} \\ &= \sum_i H(y_i) - H(\boldsymbol{y}) \end{aligned}$$

Où H désigne l'entropie, i.e. $H(.) \triangleq -E\left\{ln\left(p_{(.)}(.)\right)\right\}$

Nous savons que $I(\boldsymbol{y})$ est toujours non négative, et s'annule quand

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = \prod_{i=1}^{p} p_{y_i}(y_i),$$

C'est-à-dire : lorsque $y_1, \ldots, y_p$ sont indépendants.

Afin de trouver le minimum de $I(\boldsymbol{y})$, on peut prendre l'algorithme de descente du gradient pour le chercher. Pour un mélange linéaire instantané, la matrice de séparation $\boldsymbol{B}$, doit être estimée en minimisant l'IM de $\boldsymbol{y} = \boldsymbol{Bx}$, et dans ce cas, la mise à jour de $\boldsymbol{B}$ s'écrit

$$\boldsymbol{B} \leftarrow \boldsymbol{B} - \mu \frac{\partial I(\boldsymbol{y})}{\partial \boldsymbol{B}} \tag{4.30}$$

Ce qui nécessite l'estimation de $\frac{\partial I(\boldsymbol{y})}{\partial \boldsymbol{B}}$

A partir de la relation

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = \frac{p_{\boldsymbol{x}}(\boldsymbol{x})}{|det\,(\boldsymbol{B})|} \tag{4.31}$$

Et en prenant le logarithme des deux membres, nous aurons

$$ln\left(p_{\boldsymbol{y}}(\boldsymbol{y})\right) = ln\left(p_{\boldsymbol{x}}(\boldsymbol{x})\right) - ln(|det(\boldsymbol{B})|) \tag{4.32}$$

Le résultat intéressant est que dans l'équation (4.32), la pdf conjointe s'exprime par la somme d'un terme fixe (qui ne dépend pas du système de séparation) et un terme qui ne dépend pas de la pdf conjointe.

En combinant les équations (4.32), celle de $I(\boldsymbol{y})$ et celle de $H(.)$, on obtient

$$I(\boldsymbol{y}) = \sum_i H(y_i) - H(\boldsymbol{x}) - ln(|det(\boldsymbol{B})|) \tag{4.33}$$

Puisque $H(\boldsymbol{x})$ ne dépend pas de $\boldsymbol{B}$, et donc disparait en dérivant $I(\boldsymbol{y})$, alors $\frac{\partial I(\boldsymbol{y})}{\partial \boldsymbol{B}}$ s'écrit

$$\frac{\partial I(\boldsymbol{y})}{\partial \boldsymbol{B}} = E\{\psi_{\boldsymbol{y}}(\boldsymbol{y})\boldsymbol{x}^T\} - \boldsymbol{B}^{-T} \tag{4.34}$$

Où $\psi_{\boldsymbol{y}}(\boldsymbol{y}) = \left(\psi_{y_1}(y_1), \ldots, \psi_{y_p}(y_p)\right)^T$ est le vecteur des fonctions score marginales.

Afin d'avoir des performances indépendantes du mélange (principe d'équivariance, [L67]), on utilise le gradient naturel (relatif), $\nabla_{\boldsymbol{B}} I$ dont l'expression s'écrit :

$$\frac{\partial I}{\partial \boldsymbol{B}} = E\{\boldsymbol{\beta}_{\boldsymbol{y}}(\boldsymbol{y})\boldsymbol{x}^T\} \tag{4.35}$$

$$\nabla_{\boldsymbol{B}} I \triangleq \frac{\partial I}{\partial \boldsymbol{B}}\boldsymbol{B}^T = E\{\boldsymbol{\beta}_{\boldsymbol{y}}(\boldsymbol{y})\boldsymbol{y}^T\} \tag{4.36}$$

La mise à jour de $\boldsymbol{B}$ devient donc :

$$\boldsymbol{B} = (\boldsymbol{I}_d - \mu\nabla_{\boldsymbol{B}} I)\boldsymbol{B} \tag{4.37}$$

$\boldsymbol{I}_d$ est la matrice identité.

La figure (4.4) présente les étapes principales de l'algorithme de séparation

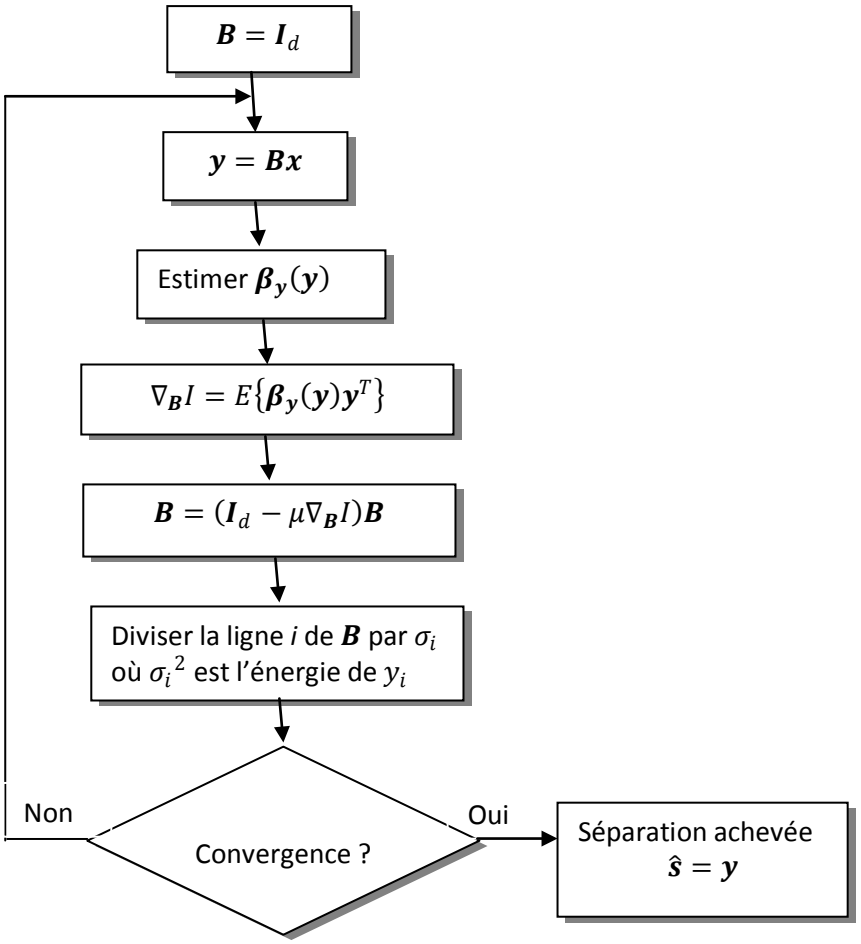


Fig. 4.4. Algorithme de séparation dans le cas linéaire instantané

4.4.2. Mélanges post-nonlinéaires

Pour la séparation de ces mélanges on adopte la structure présentée dans la figure (4.5) (déjà vue dans le chapitre 2).

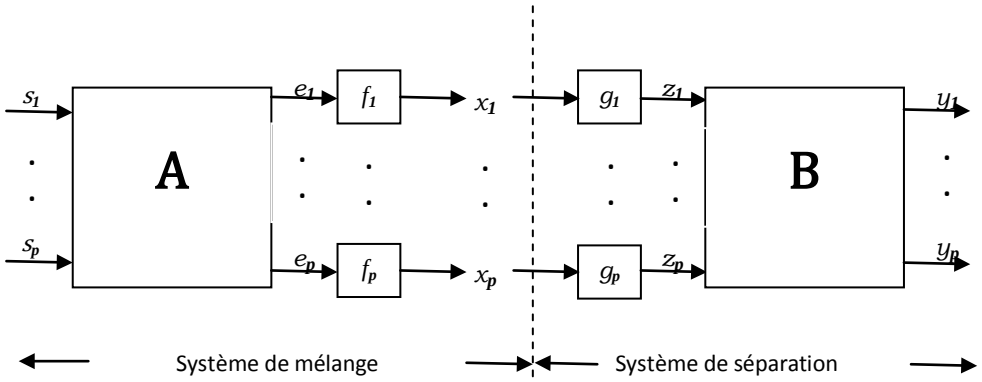


Fig.4.5. Structure de séparation dans un mélange post-nonlinéaire

Pour la minimisation de l'IM, la même idée est utilisée dans le cas du mélange post-nonlinéaire. En effet, pour ce type de mélanges, la relation entre $p_{\boldsymbol{y}}(\boldsymbol{y})$ et $p_{\boldsymbol{x}}(\boldsymbol{x})$ (voir Fig. 4.5) est multiplicative

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = \frac{p_{\boldsymbol{x}}(\boldsymbol{x})}{|det\,(\boldsymbol{B}) \prod_i \acute{g}_i(x_i)|} \tag{4.38}$$

Où $\boldsymbol{x}$ est le vecteur observé, et $\acute{g}_i(x_i)$ sont les dérivées des fonctions nonlinéaires inverses.

L'information mutuelle des sorties s'écrit alors sous la forme suivante :

$$I(\boldsymbol{y}) = \sum_i H(y_i) - H(\boldsymbol{x}) - ln(|det(\boldsymbol{B})|) - \sum_i E\{ln|\acute{g}_i(x_i)|\} \tag{4.39}$$

Afin d'évaluer le gradient de l'IM, supposons que $\boldsymbol{x}$ est un vecteur aléatoire borné, et soit $\boldsymbol{\Delta}$ un vecteur aléatoire petit de même dimension que $\boldsymbol{x}$, donc [L32] :

$$I(\boldsymbol{x}+\boldsymbol{\Delta}) - I(\boldsymbol{x}) = E\{\boldsymbol{\Delta}^T \boldsymbol{\beta}_{\boldsymbol{x}}(\boldsymbol{x})\} + o(\boldsymbol{\Delta}) \quad (4.40)$$

Où $\boldsymbol{\beta}_{\boldsymbol{x}}$ est la différence des fonctions score (SFD : Score Function Difference) de , et $o(\boldsymbol{\Delta})$ sont les termes d'ordre supérieurs de $\boldsymbol{\Delta}$.

Rappelons que pour une fonction multivariable dérivable $f(\boldsymbol{x})$ nous avons :

$$f(\boldsymbol{x}+\boldsymbol{\Delta}) - f(\boldsymbol{x}) = \boldsymbol{\Delta}^T . \big(\nabla f(\boldsymbol{x})\big) + o(\boldsymbol{\Delta}) \quad (4.41)$$

En comparant cette équation avec l'équation (4.40), nous pouvons remarquer que la SFD est le gradient stochastique de l'IM.

Le point le plus important est que les sources sont séparées à un facteur et une permutation près avec la disparition de l'ambiguïté nonlinéaire (sans distorsion nonlinéaire), à condition que les sources soient réellement mélangées à la sortie de la partie linéaire : i.e. il y a au moins deux éléments non nuls dans chaque ligne ou colonne de la matrice de mélange.

Contrairement au cas linéaire instantané, nous avons besoin de calculer les gradients de l'IM par rapport aux paramètres du système de séparation : la matrice $\boldsymbol{B}$ et les fonctions g_i . et puis utiliser l'algorithme à descente du gradient sur ces paramètres pour la minimisation de $(\boldsymbol{y})$.

Le calcul du gradient par rapport à $\boldsymbol{B}$ est déjà donné précédemment :

$$\frac{\partial I}{\partial \boldsymbol{B}} = E\{\boldsymbol{\beta}_{\boldsymbol{y}}(\boldsymbol{y})\boldsymbol{z}^T\} \quad (4.42)$$

Le gradient naturel s'exprime donc par (voir Equation 4.37) :

$$\nabla_{\boldsymbol{B}} I = E\{\boldsymbol{\beta}_{\boldsymbol{y}}(\boldsymbol{y})\boldsymbol{y}^T\} \tag{4.43}$$

La deuxième étape consiste à calculer le gradient de $I(\boldsymbol{y})$ par rapport aux fonctions g_i . Pour ce faire, supposons qu'il y a des petites perturbations dans ces fonctions : i.e.

$$\hat{g}_i = g_i + \varepsilon_i \circ g_i \tag{4.44}$$

Où ε_i indique une fonction petite. (4.44) est équivalente à :

$$\hat{z}_i = z_i + \varepsilon_i(z_i) = z_i + \delta_i \tag{4.45}$$

Où $\delta_i \triangleq \varepsilon_i(z_i)$, et par conséquent :

$$\hat{\boldsymbol{y}} \triangleq \boldsymbol{B}\hat{\boldsymbol{z}} = \boldsymbol{y} + \boldsymbol{\delta} \tag{4.46}$$

Où $\boldsymbol{\delta} \triangleq \left(\delta_1 , \dots , \delta_p\right)^{\boldsymbol{T}}$.

Dans ce cas, l'effet de cette perturbation sur $I(\boldsymbol{y})$ est de la forme [TH2] :

$$I(\hat{\boldsymbol{y}}) - I(\boldsymbol{y}) = E\{\boldsymbol{\delta}^T \boldsymbol{B}^T \boldsymbol{\beta}_{\boldsymbol{y}}(\boldsymbol{y})\} \tag{4.47}$$

Soit

$$\boldsymbol{\alpha}(\boldsymbol{y}) \triangleq \boldsymbol{B}^T \boldsymbol{\beta}_{\boldsymbol{y}}(\boldsymbol{y}) \tag{4.48}$$

Alors

$$I(\hat{\boldsymbol{y}}) - I(\boldsymbol{y}) = E\{\boldsymbol{\delta}^T \boldsymbol{\alpha}(\boldsymbol{y})\}$$

$$= \sum_i E\{\varepsilon_i(z_i)\alpha_i(\boldsymbol{y})\}$$

$$= \sum_i E\{\varepsilon_i(z_i) E\{\alpha_i(\boldsymbol{y})|z_i\}\} \quad (4.49)$$

$$= \sum_i \int \varepsilon_i(z) E\{\alpha_i(\boldsymbol{y})|z_i = z\} p_{z_i}(z) dz$$

Finalement, à partir de (4.48), le gradient naturel de $I(\boldsymbol{y})$ par rapport à g_i est alors :

$$\nabla_{g_i} I(z) = E\{\alpha_i(\boldsymbol{y})|z_i = z\} \quad (4.50)$$

La règle de Descente du gradient pour les paramètres du système s'écrit donc :

$$\begin{cases} \boldsymbol{B} = (\boldsymbol{I}_d - \mu_1 \nabla_{\boldsymbol{B}} I)\boldsymbol{B} \\ z_i = z_i - \mu_2 \nabla_{g_i} I(z_i) \end{cases} \quad (4.51)$$

Les équations (4.43) et (4.50) sont utilisées pour le calcul de (4.51), où l'espérance dans (4.45) est remplacée par la moyenne empirique. Cependant, pour le calcul de l'espérance conditionnelle dans (4.50) on peut utiliser soit une méthode paramétrique (régression en fonction de $(z_i\ ,\ \alpha_i)$, ou bien une méthode non paramétrique et dans ce cas on peut utiliser un lissage par *splines*.

La figure (4.6) présente l'algorithme de séparation pour le mélange PNL.

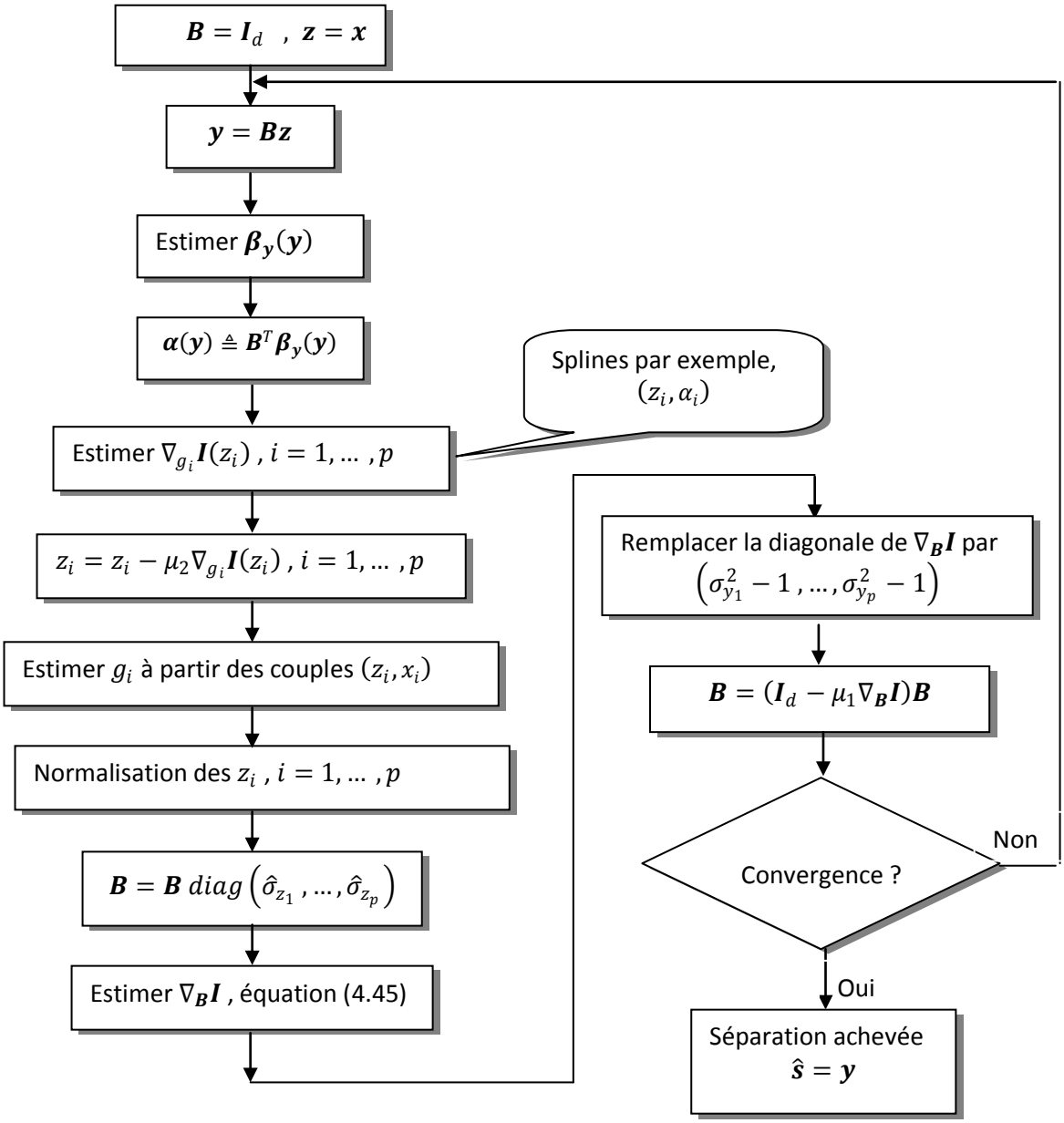


Fig. 4.6. Approche du gradient de l'IM pour la séparation des mélanges PNL

4.5. CONCLUSION

Dans ce chapitre, nous avons présenté les principes de séparation de sources pour les deux types de mélanges : mélanges linéaires et nonlinéaires particuliers. Ces modèles non linéaires appelés post-non linéaires, sont physiquement plausibles et présentent des propriétés de séparabilité sans distorsion très intéressantes. La méthode de séparation repose sur la minimisation d'un critère d'information mutuelle, qui nécessite l'estimation des dérivées logarithmique des pdf inconnues des sorties (fonctions score).

CHAPITRE 5

Réseaux de neurones en SAS

5.1. INTRODUCTION

Dan ce chapitre, nous présentons les contributions de ce travail. En effet, les principaux éléments de ce chapitre se résument par les points suivants :

- l'estimation des fonctions score marginales en estimant les pdf marginales par un réseau de neurones multicouches (MLP) [L69].
- nous proposons un algorithme de séparation de sources de mélanges linéaires instantanés en utilisant une minimisation sous contraintes de l'IM et en intégrant l'estimateur neuronal des fonctions score marginales dans l'algorithme [L69].
- Nous présentons aussi un algorithme de séparation de sources de mélanges PNL en utilisant une modélisation neuronale de l'étage de séparation, où les expressions du gradient ainsi que les équations de

mise à jour des poids et des biais sont exposées dans un contexte d'apprentissage non-supervisé [NL34].

5.2. MINIMISATION SOUS CONTRAINTES

Il est important de déduire, à partir de la divergence de *Kullback-Leibler*, que l'information mutuelle est toujours positive et n'est nulle que si et seulement si $p_{\boldsymbol{y}} = \prod_i p_{y_i}$, c'est-à-dire, ssi les composantes y_i sont indépendantes. En fonction de l'IM, cela s'écrit :

Pour des mélanges linéaires instantanés

$$\min_{\boldsymbol{B}} I(\boldsymbol{y}) = 0$$

Et pour des mélanges post-nonlinéaires instantanés :

$$\min_{\boldsymbol{B},g} I(\boldsymbol{y}) = 0$$

Cette minimisation est discutée en détail dans les sections précédentes. Cependant, pour remédier au problème de l'ambiguïté d'échelle, et stabiliser l'algorithme de minimisation, une minimisation sous la contrainte de normalisation des puissances des sources estimées est proposée.

En procédant à mettre les puissances des entrées égales à 1, la minimisation sous cette contrainte devient :

- Minimiser l'IM du vecteur du vecteur sortie $\boldsymbol{y}$

$$I(\boldsymbol{y}) = \int_{\boldsymbol{y}} p_{\boldsymbol{y}}(\boldsymbol{y}) ln\left\{\frac{p_{\boldsymbol{y}}(\boldsymbol{y})}{\prod_{i=1}^{p} p_{y_i}(y_i)}\right\} d\boldsymbol{y} \qquad (5.1)$$

Sous la contrainte

$$\left(\sigma_{y_i}^2 - 1\right)^2 = 0\,, \quad i = 1, \dots, p \tag{5.2}$$

Où $\sigma_{y_i}^2 = E\{(y_i - m_i)^2\}$, $m_i = E\{y_i\}$

Pour résoudre le problème (5.1), on utilise le Lagrangien associé :

$$\ell(\boldsymbol{\theta}) = I(\boldsymbol{y}) + \boldsymbol{\lambda}^T \boldsymbol{\sigma_y} \tag{5.3}$$

Où

$$\boldsymbol{\lambda} = \left(\lambda_1, \dots, \lambda_p\right)^T \quad et \quad \boldsymbol{\sigma_y} = \left(\left(\sigma_{y_1}^2 - 1\right)^2, \dots, \left(\sigma_{y_p}^2 - 1\right)^2\right)^T$$

λ_i sont des constantes réelles positives.

$\boldsymbol{\theta}$ est le vecteur des paramètres de minimisation qui dépend du mélange.

Si nous notons

$$\mathrm{C} = \boldsymbol{\lambda}^T \boldsymbol{\sigma_y} = \left(\lambda_1, \dots, \lambda_p\right) \begin{pmatrix} \left(\sigma_{y_1}^2 - 1\right)^2 \\ \cdot \\ \cdot \\ \cdot \\ \left(\sigma_{y_p}^2 - 1\right)^2 \end{pmatrix} = \sum_{i=1}^{p} \lambda_i \left(\sigma_{y_i}^2 - 1\right)^2$$

Alors

$$\ell(\boldsymbol{\theta}) = I(\boldsymbol{y}) + \mathrm{C}(\boldsymbol{\theta}) \tag{5.4}$$

La solution du problème (5.4) devient équivalente à celle du problème d'optimisation sans contraintes suivant

$$optimum\{\ell(\boldsymbol{\theta})\} = optimum\{I(\boldsymbol{y}) + \mathrm{C}(\boldsymbol{\theta})\} \tag{5.5}$$

Puisque (5.5) dépend du modèle du mélange, nous allons considérer dans ce qui suit le mélange linéaire instantané.

La solution peut être calculée par utilisation de l'algorithme de descente du gradient.

Dans ce cas (5.5) s'écrit

$$\max_{\lambda_1 \dots \lambda_p} \min_{\boldsymbol{B}} \{\ell(\boldsymbol{B}, \lambda_1, \dots, \lambda_p)\} = \max_{\lambda_1 \dots \lambda_p} \min_{\boldsymbol{B}} \{I(\boldsymbol{y}) + \mathrm{C}(\boldsymbol{y}, \lambda_1, \dots, \lambda_p)\} \tag{5.6}$$

A l'optimum, nous avons

$$\frac{d\ell}{d\boldsymbol{B}} = 0\,, \quad \frac{\partial \ell}{\partial \lambda_i} = 0 \quad , \qquad i = 1, \dots, p$$

Où

$$\frac{d\ell}{d\boldsymbol{B}} = \frac{dI(\boldsymbol{y})}{d\boldsymbol{B}} + \frac{d\mathrm{C}(\boldsymbol{y}, \lambda_1, \dots, \lambda_p)}{d\boldsymbol{B}} \tag{5.7}$$

Et

$$\frac{\partial\, \ell}{\partial\, \lambda_i} = \left(\sigma_{y_i}^2 - 1\right)^2 \;\; , \; i = 1, \dots, p \tag{5.8}$$

Puisque dans un mélange linéaire instantané, comme déjà vue au chapitre(2), le vecteur de sortie s'écrit $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x}$, et son information mutuelle, $I(\boldsymbol{y})$,est donnée par la relation

$$I(\boldsymbol{y}) = \sum_{i=1}^{p} H(y_i) - H(\boldsymbol{x}) - ln(|det(\boldsymbol{B})|)$$

Dans ce cas, puisque $H(\boldsymbol{x})$ est indépendante de la matrice $\boldsymbol{B}$, on a

$$\frac{dI(\boldsymbol{y})}{d\boldsymbol{B}} = \sum_{i=1}^{p} \frac{dH(y_i)}{d\boldsymbol{B}} - \frac{dln(|det(\boldsymbol{B})|)}{d\boldsymbol{B}} \tag{5.9}$$

Or, d'une part :

$$\sum_{i=1}^{p} \frac{dH(y_i)}{d\boldsymbol{B}} = -\sum_{i=1}^{p} E\left\{\frac{d\ ln\left(p_{y_i}(y_i)\right)}{d\boldsymbol{B}}\right\}$$

$$= -\sum_{i=1}^{p} E\left\{\frac{\acute{p}_{y_i}(y_i)}{p_{y_i}(y_i)}\frac{dy_i}{d\boldsymbol{B}}\right\}$$

$$= E\{\boldsymbol{\psi_y}(\boldsymbol{y})\boldsymbol{x}^T\} \tag{5.10}$$

Où $\boldsymbol{\psi_y}(\boldsymbol{y})$ est le vecteur des fonctions score marginales de $\boldsymbol{y}$

D'autre part, on peut montrer que:

$$\frac{dln(|det(\boldsymbol{B})|)}{d\boldsymbol{B}} = \boldsymbol{B}^{-T} \tag{5.11}$$

Et enfin l'expression de $\frac{dI(\boldsymbol{y})}{d\boldsymbol{B}}$ devient

$$\frac{dI(\boldsymbol{y})}{d\boldsymbol{B}} = E\{\boldsymbol{\psi_y}(\boldsymbol{y})\boldsymbol{x}^T\} - \boldsymbol{B}^{-T} \tag{5.12}$$

Le gradient du terme $\mathrm{C}(\boldsymbol{y}, \lambda_1, \ldots, \lambda_p)$ par rapport à $\boldsymbol{B}$, est donné comme suit :

$$\frac{d\mathrm{C}(\boldsymbol{y},\lambda_1,\ldots,\lambda_p)}{d\boldsymbol{B}} = 4\, diag(\lambda_1, \ldots, \lambda_p) E\{\boldsymbol{w}\boldsymbol{x}^T\}, \quad pour\ \ m_i = 0,\ i = 1, \ldots, p \qquad (5.13)$$

Où

$$\boldsymbol{w} = (\omega_1, \ldots, \omega_p)^T \quad \text{avec} \qquad \omega_i = (E[y_i^2] - 1) y_i$$

Finalement, l'expression du gradient par rapport à $\boldsymbol{B}$ peut s'écrire sous la forme suivante :

$$\frac{d\ell}{d\boldsymbol{B}} = E\{\boldsymbol{\psi}_{\boldsymbol{y}}(\boldsymbol{y})\boldsymbol{x}^T\} - \boldsymbol{B}^{-T} + 4\, diag(\lambda_1, \ldots, \lambda_p) E\{\boldsymbol{w}\boldsymbol{x}^T\} \qquad (5.14)$$

et

$$\frac{\partial\, \ell}{\partial\, \lambda_i} = \left(\sigma_{y_i}^2 - 1\right)^2 \ ,\ i = 1, \ldots, p \qquad (5.15)$$

Il est clair dans (5.14), que nous avons besoin d'estimer les densités de probabilité marginales $p_{y_i}(y_i)$ afin de calculer par la suite les fonctions score marginales $\psi_{y_i}(y_i)$ correspondantes. Le paragraphe suivant a pour objet d'exposer notre méthode d'estimation des pdf marginales. Dans la section suivante, nous présentons une méthode d'estimation de pdf basée sur réseau de neurone et plus précisément sur un réseau multicouche (MLP : Multi-Layer Perceptron).

5.2.1. Estimation de la fonction de densité de probabilité

L'approximation des fonctions de densité de probabilité par des réseaux feedforward a été initialement introduite par White [pdf1]. Pour un problème défini dans $\mathcal{R}^p$, l'architecture du réseau consiste en p élément d'entrée, H élément dans la couche cachée et un élément dans la couche de sortie, comme il est présenté par la figure suivante :

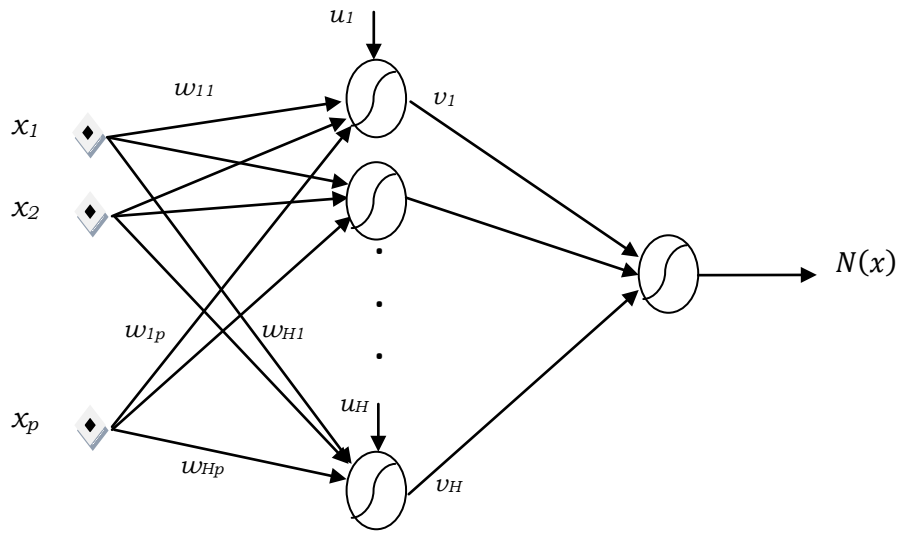


Fig. 5.1. MLP Adopté pour l'estimation de densité de probabilité

A partir de la figure ci-dessus, nous avons les équations suivantes

$$N(x) = exp\left(\sum_{j=1}^{H} v_j \, sgm\left(c_j\right)\right) \tag{5.16}$$

Où $sgm(\,.\,)$ est la fonction logistique (sigmoid), i.e.

$$sgm(c_j) = \frac{1}{1 + e^{-c_j}}$$

Et c_i est l'entrée du i^{ème} élément de la couche cachée.

$$c_j = \sum_{i=1}^{p} w_{ji}\, x_i + u_j \qquad (5.17)$$

Le réseau est adapté en ajustant l'ensemble des paramètres w_{ji}, u_i et v_i $(i = 1, \dots, p)$, et $(j = 1, \dots, H)$

pour être plus précis, notons l'ensemble des paramètres du réseau par $\boldsymbol{\theta}$, et par conséquent la sortie du réseau devient $N(x, \boldsymbol{\theta})$.

L'apprentissage du réseau se fait en minimisant la fonction de vraisemblance logarithmique négative des données par rapport aux paramètres du réseau [pdf2].

Dans le cas multidimensionnel $\boldsymbol{x}(k) \in \mathcal{R}^p$, $(k = 1, \dots, n)$, où n est la taille de l'échantillon. L'échantillon est supposé suivre une loi de probabilité $p(\boldsymbol{x})$ inconnue, que nous voulons approximer.

Le modèle paramétrique, $p_N(\boldsymbol{x}, \boldsymbol{\theta})$, de la densité de $\boldsymbol{x}$ est donnée par la fonction suivante :

$$p_N(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{N(x, \boldsymbol{\theta})}{\int_{\mathcal{R}^p} N(\boldsymbol{y}, \boldsymbol{\theta}) d\boldsymbol{y}} \qquad (5.18)$$

Où le vecteur des paramètres $\boldsymbol{\theta}$ est ajusté en minimisant le logarithme du maximum de vraisemblance négatif de $p_N(\boldsymbol{x}, \boldsymbol{\theta})$, i.e.

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{k=1}^{n} ln\{p_N(\boldsymbol{x}(k), \boldsymbol{\theta})\} \tag{5.19}$$

En remplaçant $p_N(\boldsymbol{x}(k), \boldsymbol{\theta})$ par son expression (5.18), on obtient

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= -\sum_{k=1}^{n} ln\{N(\boldsymbol{x}(k), \boldsymbol{\theta})\} + n\ ln\left\{\int_{\mathcal{R}^p} N(\boldsymbol{x}, \boldsymbol{\theta}) d\boldsymbol{x}\right\} \\ &= -\sum_{k=1}^{n} ln\{N(\boldsymbol{x}(k), \boldsymbol{\theta})\} + n\ ln(I_{\boldsymbol{\theta}})\end{aligned} \tag{5.20}$$

avec

$$I_{\boldsymbol{\theta}} = \int_{\mathcal{R}^p} N(\boldsymbol{x}, \boldsymbol{\theta}) d\boldsymbol{x} \tag{5.21}$$

En pratique, en remplace $\mathcal{R}^p$ de (5.21) par le support de $\boldsymbol{x}$ qu'on définit par :

$$\boldsymbol{S_x} = \left\{\boldsymbol{x} = \left(x_1, \dots, x_p\right)^T \in \mathcal{R}^p\ ,\ a_i \leq x_i \leq b_i\ , \forall i\right\} \tag{5.22}$$

Puisque le support de $\boldsymbol{x}$ est inconnu, il est estimé par l'estimateur suivant :

$$\left[\hat{a}_i\ , \hat{b}_i\ \right] = \left[\min_{k=1,\dots,n} \boldsymbol{x}_i(k)\ , \max_{k=1,\dots,n} \boldsymbol{x}_i(k)\right] \tag{5.23}$$

Dans ce cas, l'équation (5.18) devient comme suit :

$$p_N(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{N(x, \boldsymbol{\theta})}{\int_{\boldsymbol{S_x}} N(\boldsymbol{y}, \boldsymbol{\theta}) d\boldsymbol{y}} \tag{5.24}$$

L'apprentissage du réseau $N(x, \boldsymbol{\theta})$ s'effectue en cherchant l'optimum $\boldsymbol{\theta}^*$, du vecteur des paramètres $\boldsymbol{\theta}$ conduisant à la valeur minimale $\mathcal{L}(\boldsymbol{\theta}^*)$ de $\mathcal{L}(\boldsymbol{\theta})$. Cette recherche peut être effectuée en utilisant une des méthodes de descente du gradient (rétropropagation du gradient, quasi-Newton, …etc.).

Cependant, la minimisation de (5.20), nécessite le calcul de son gradient par rapport aux paramètres du réseau θ_j, $\frac{\partial \mathcal{L}}{\partial \theta_j}$.

Le gradient de (5.20) s'écrit alors :

$$\frac{\partial \mathcal{L}}{\partial \theta_j} = -\sum_{k=1}^{n} \frac{\partial N(\boldsymbol{x}(k), \boldsymbol{\theta})/\partial \theta_j}{N(\boldsymbol{x}(k), \boldsymbol{\theta})} + n \, \frac{\partial I_{\boldsymbol{\theta}}/\partial \theta_j}{I_{\boldsymbol{\theta}}} \tag{5.25}$$

L'équation (5.25) contient deux membres :

- La première partie de la formule du gradient peut être explicitement et facilement calculée, $\partial N(\boldsymbol{x}(k), \boldsymbol{\theta})/\partial \theta_j$
- Par contre, le terme $\partial I_{\boldsymbol{\theta}}/\partial \theta_j$ nécessite le calcul de l'intégrale, et par conséquent choisir une méthode numérique pour ce faire.

Afin d'éviter le calcul de l'intégrale $I_{\boldsymbol{\theta}}$, on propose la modélisation de la densité des données par une densité de loi exponentielle. Dans le cas monodimensionnel, cette modélisation s'écrit :

$$p_d(x, \boldsymbol{\delta}) = \frac{N_e(x, \boldsymbol{\delta})}{\int_a^b N_e(y, \boldsymbol{\delta}) dy} = \frac{exp(\delta_1 x + \cdots + \delta_d x^d)}{\int_a^b exp(\delta_1 y + \cdots + \delta_d y^d) dy} \tag{5.26}$$

Où

d est la dimension du modèle,

$(\delta_i\ ,\ i = 1, \dots, d)$ les paramètres du modèle.

Cette modélisation est utilisée pour les avantages suivants :

1. Les densités exponentielles s'adaptent pour la modélisation de la plupart des densités usuelles (normale, uniforme, exponentielles, …etc.)
2. Cette modélisation est utile pour une bonne initialisation du réseau MLP (Fig. 5.2).

Dans le but d'estimer le modèle (5.26) on propose l'architecture de la figure ci-dessous

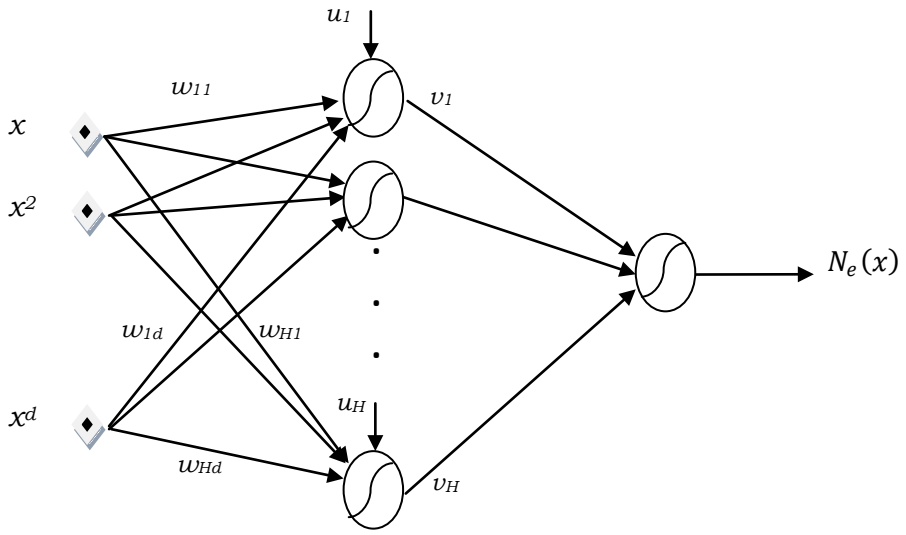


Fig. 5.2. MLP utilisé pour l'estimation du modèle (5.26)

On peut penser qu'au lieu de chercher à estimer directement la fonction de densité de probabilité, on estime tout d'abord la fonction de répartition (cdf) correspondante et cela pour les deux raisons suivantes :

1. La fonction de répartition est plus facile à estimer grâce à sa forme
2. L'utilisation de la fonction d'activation logistique dont la sortie est d'une forme très semblable à une fonction de répartition (voir Fig. 5.3).

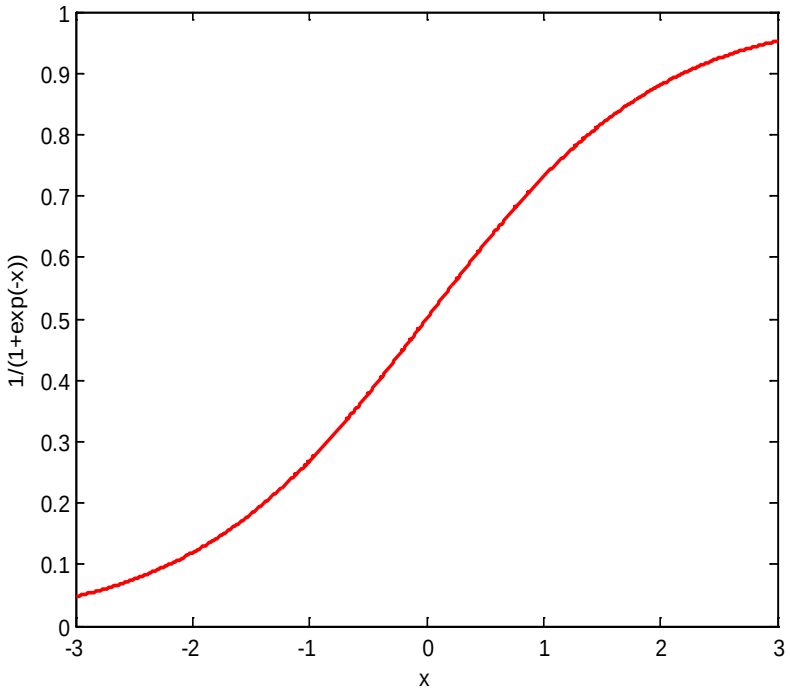


Fig. 5.3. Fonction logistique

Rappelons que la fonction de répartition F_X d'une variable aléatoire continue X de densité de probabilité de loi p_X se définit par l'équation suivante :

$$F_X(x) = p_X(X \leq x) = \int_{-\infty}^{x} p_X(y)dy \qquad (5.27)$$

Dans le cas discret

$$F_X(x_i) = p_X(X \leq x_i) = \sum_{j=1}^{i} p_X(x_j) \ , \ x_1 < x_2 < \cdots < x_i \qquad (5.28)$$

Pour finir avec l'initialisation on adopte une des méthodes de modélisation non paramétrique telles que l'histogramme, la méthode à noyau, …etc.

Après l'apprentissage du réseau, la fonction de densité de probabilité peut être calculée de la manière suivante :

Nous avons

$$p_X(x) = \frac{dF_X(x)}{dx} \qquad (5.29)$$

et

$$\hat{F}_X(x) = N_e(x) = \sum_{j=1}^{H} v_j \left(\frac{1}{1+exp\{-(\sum_{i=1}^{d} w_{ji}\, x^i + u_j)\}} \right) = \sum_{j=1}^{H} v_j\, z_j \qquad (5.30)$$

tel que : $z_j = \frac{1}{1+exp\{-(\sum_{i=1}^{d} w_{ji}\, x^i + u_j)\}}$

Par conséquent

$$\hat{p}_X(x) = \frac{d\hat{F}_X(x)}{dx} = \frac{dN_e(x)}{dx} = \sum_{j=1}^{H} v_j \frac{dz_j}{dx} \qquad (5.31)$$

avec

$$\frac{dz_j}{dx} = \acute{L}_j(x)\big(1 - z_j\big)z_j \tag{5.32}$$

et

$$\acute{L}_j(x) = -\frac{d\left(\sum_{i=1}^{d} w_{ji}\, x^i + u_j\right)}{dx} = -\sum_{l=1}^{d} w_{jl}\ l\ x^{l-1} \tag{5.33}$$

En fin

$$\hat{p}_X(x) = \sum_{j=1}^{H} v_j\, \acute{L}_j(x) z_j\big(1 - z_j\big) \tag{5.34}$$

Grace à la forme de la fonction sigmoïde, Figure (5.3), le réseau de la figure (5.2) se réduit au réseau de la figure ci-dessous (Fig.5.4)

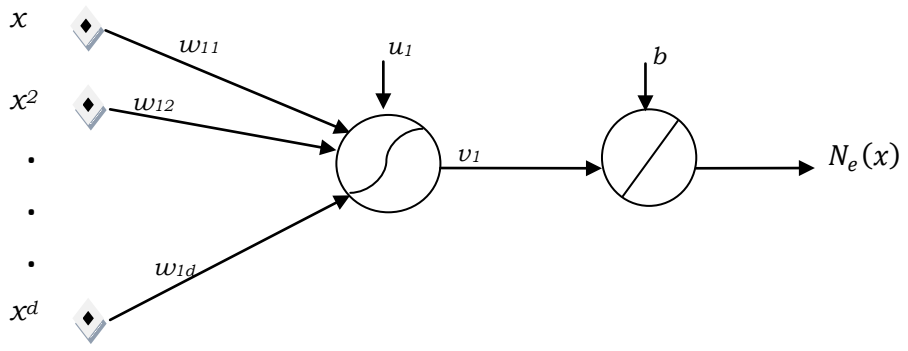
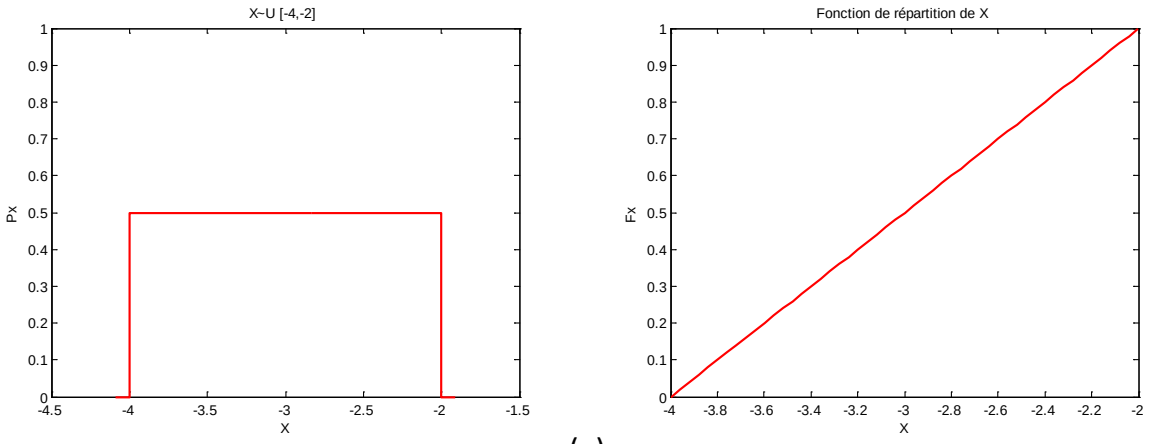


Fig. 5.4. Architecture utilisée pour l'estimation
de la fonction de répartition

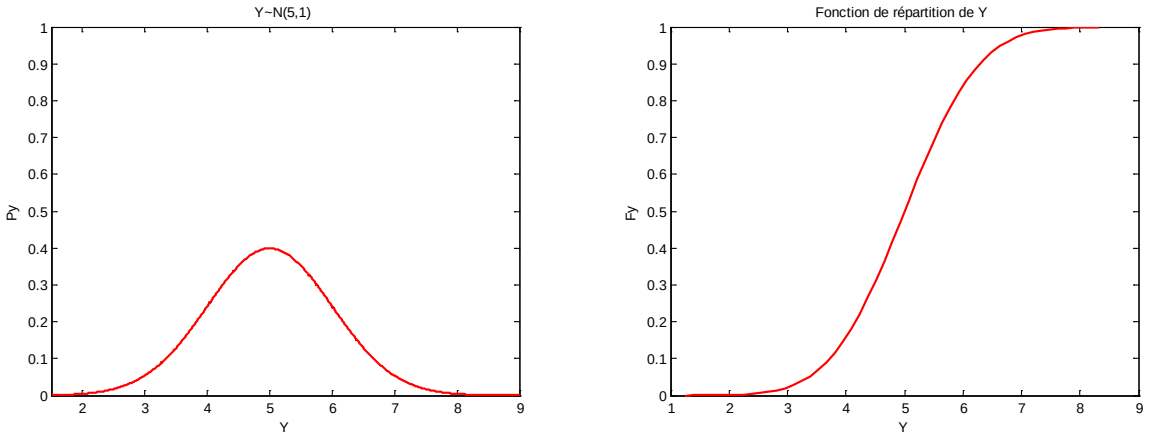
5.2.2. Exemples

- **Exemple 1**

Soit X et Y deux variables aléatoires scalaires telle que $X \sim U[-4,-2]$: i.e. une variable aléatoire uniforme dans l'intervalle [-4,-2], et $Y \sim N(5,1)$: i.e. une variable aléatoire gaussienne de moyenne 5 et de variance 1. La figure (5.11) présente les fonctions de répartition ainsi que les fonctions de densité de probabilité théoriques correspondantes.



(a)



(b)

Fig. 5.5. Fonctions de densité de Probabilité (à gauche) et Fonctions de répartition (à droite) - (a) : $X \sim U[-4,-2]$, (b) : $Y \sim N(5,1)$

En prenant un nombre d'échantillons $N = 2000$, le réseau est adapté pour différentes valeurs de d en utilisant $M = 50$ échantillons uniformément répartis sur le support de définition. Les paramètres du réseau sont adaptés en utilisant l'erreur quadratique moyenne (MSE) comme critère de performance.

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(e_i)^2 = \frac{1}{M}\sum_{i=1}^{M}\left(F_i - N_e(x_i)\right)^2 \qquad (5.35)$$

Où

F_i est la cible (estimateur non paramétrique)

$N_e(x_i)$ est la sortie du réseau.

la figure (5.6) présente les résultats obtenus des fonctions de répartition estimées ainsi que les pdf correspondantes pour ($d = 1$).

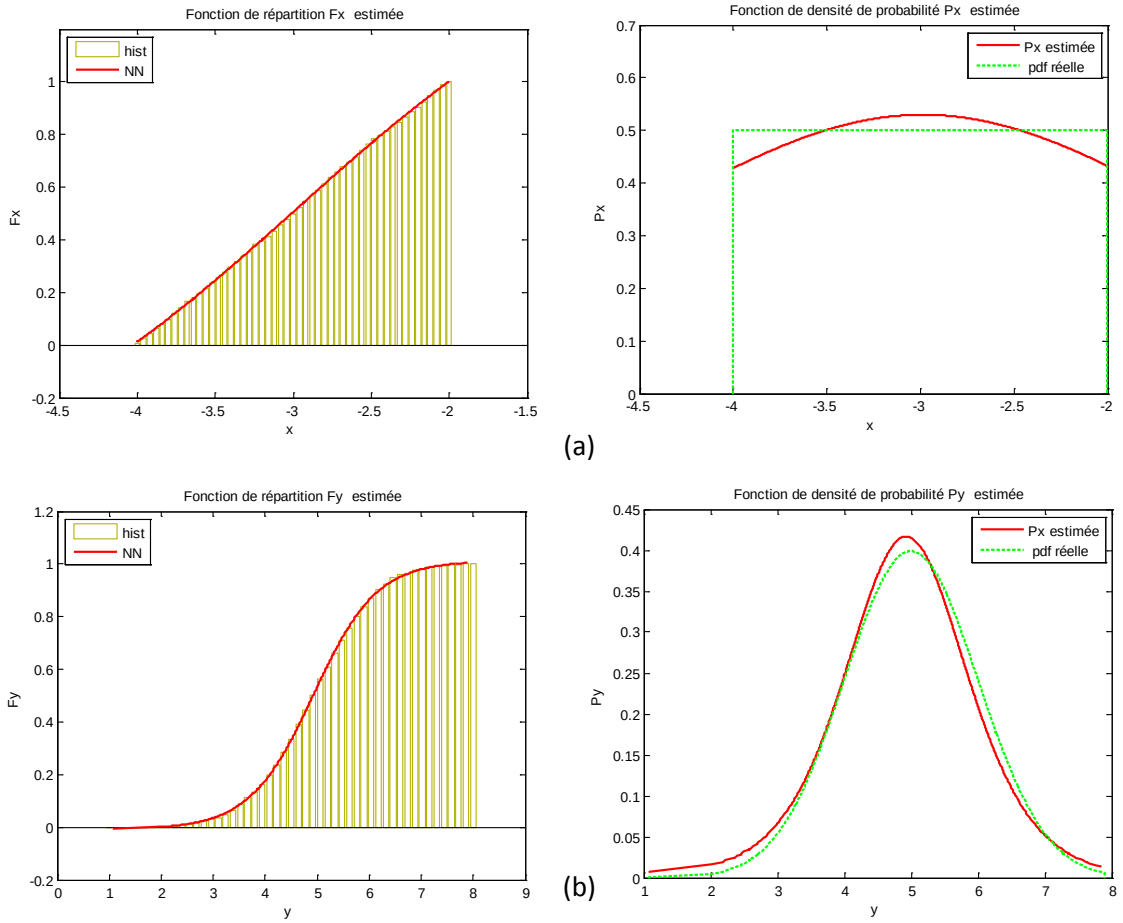


Fig. 5.6. cdf et pdf estimées et réelles (d=1, N=2000) - (a) : $X \sim U[-4,-2]$, (b) : $Y \sim N(5,1)$

La figure suivante montre l'effet de la dimension d sur la qualité d'estimation :

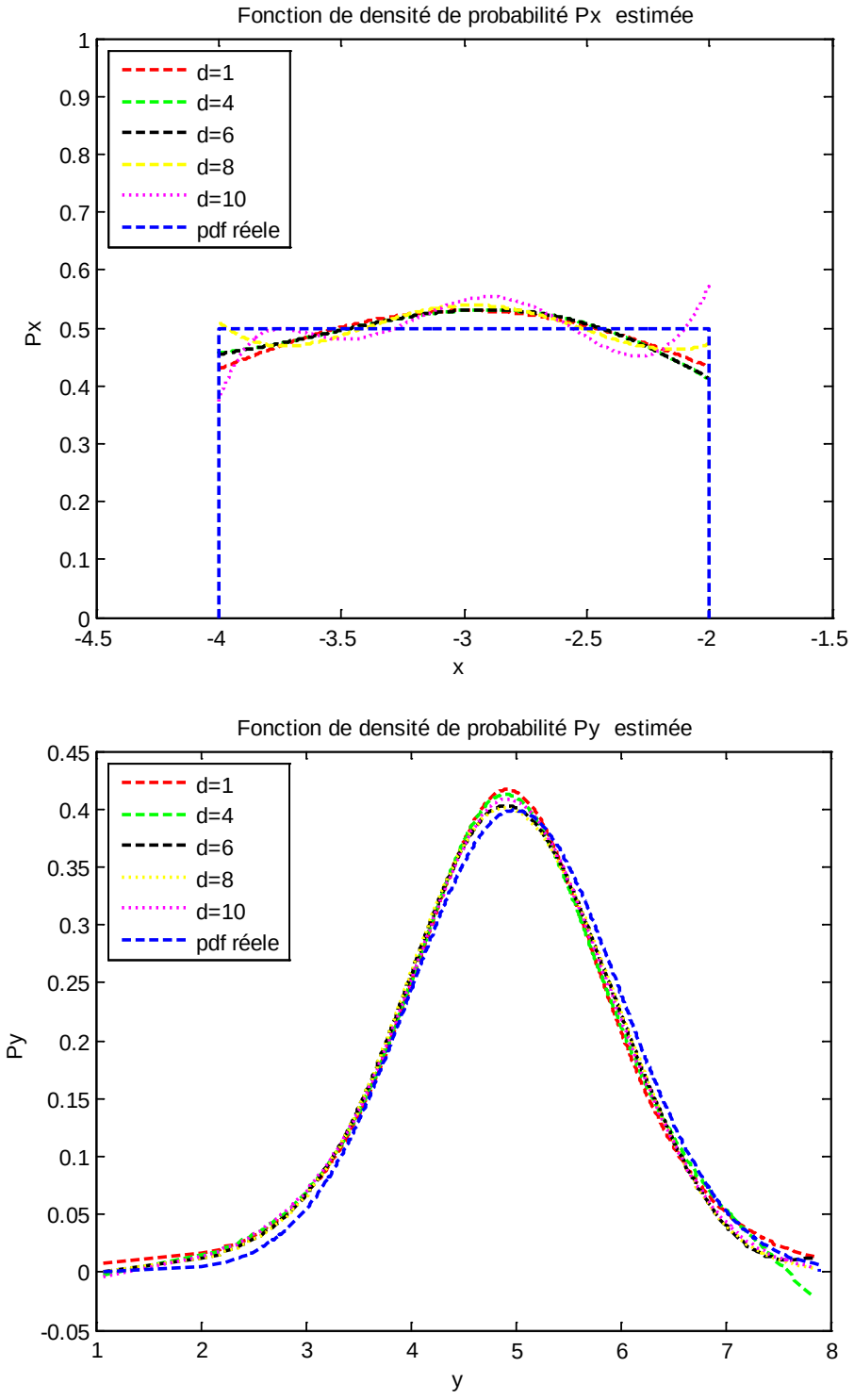


Fig. 5.7. pdf estimées en fonction de d

A partir de la figure (5.7) nous pouvons voir que le paramètre d influe surtout au niveau des bornes du support d'une part et d'autre part en augmentant d on augmente le nombre d'ondulations.

La figure suivante montre l'amélioration qu'on peut avoir en procédant à l'estimation de la pdf directement, et surtout pour des fonctions non lisses telles que la loi uniforme, triangulaire, …etc.

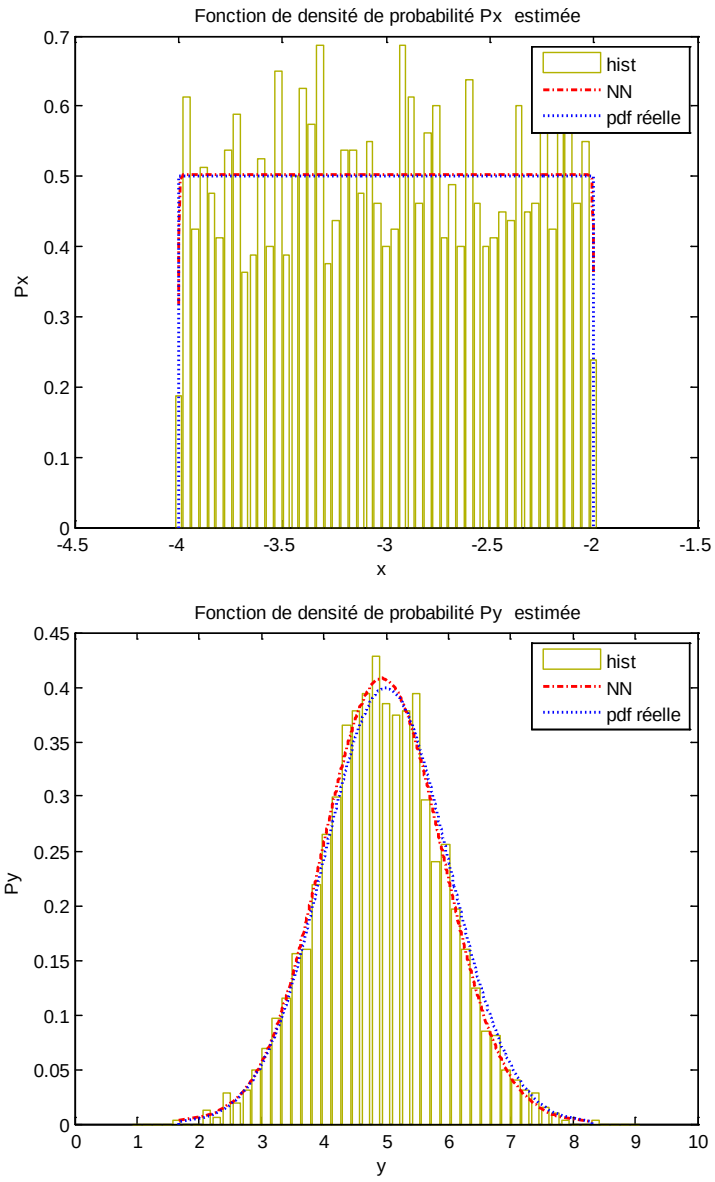


Fig. 5.8. pdf estimées $(d = 2)$ pour les lois $U[-4,-2]$ et $N(5,1)$ respectivement

- **Exemple 2**

Dans cet exemple nous prenons un cas plus complexe de densité de probabilité. Dans ce cas on considère une loi de probabilité, $g(x)$, d'un mélange de lois de densités standards. i.e.,

1- $g_1(x) = 0.5\, U[-4,-2] + 0.5\, N(5,1)$

2- $g_2(x) = 0.25\, U[-2,-1] + 0.25\, N\left(5,\frac{1}{4}\right) + 0.25U[1,2] + 0.25\, N\left(5,\frac{1}{4}\right)$

La forme de $g(x)$ est présentée par les figures ci-dessous

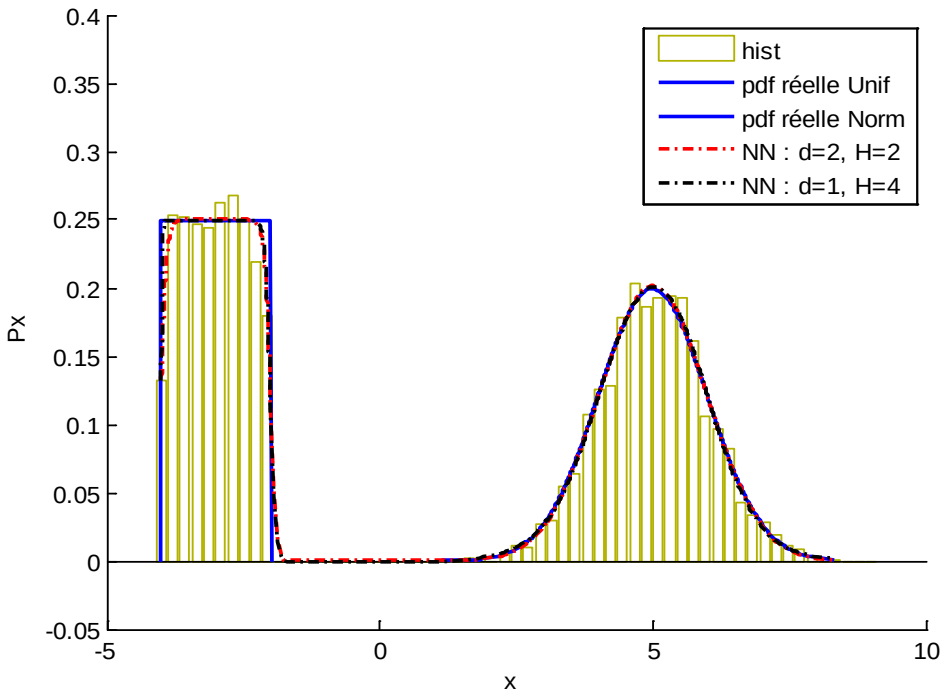


Fig. 5.9. Pdf estimée ($g_1(x)$) pour différentes valeurs de d et H

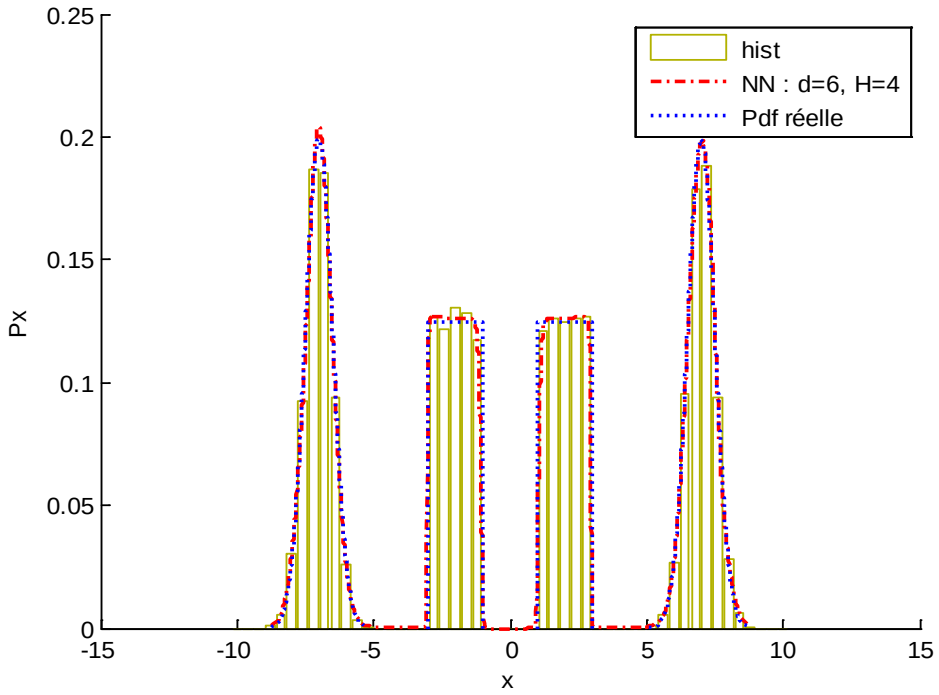


Fig. 5. 10. Pdf estimée ($g_2(x)$) pour les valeurs : $d = 6$ et $H = 4$ ($f = sgm(.)$, comme fonction d'activation en couche de sortie)

5.2.3. Algorithme de l'IM sous contraintes

Les résultats trouvés dans la section précédente sont appliqués directement dans l'algorithme de minimisation sous contraintes de l'information mutuelle, et ceci en calculant les estimés des fonctions score marginales à partir des pdf estimées. L'expression est donnée par l'équation suivante (voir Chapitre (4))

$$\hat{\psi}_{\boldsymbol{y}}(\boldsymbol{y}) = \left(\hat{\psi}_{y_1}(y_1), \dots, \hat{\psi}_{y_p}(y_p)\right)^T \quad \text{où} \qquad \hat{\psi}_{y_1}(y_1) = -\frac{\hat{p}'_{y_i}(y_i)}{\hat{p}_{y_i}(y_i)}$$

- **Algorithme**

- *Initialisation* : $\boldsymbol{B} = I_p$, λ_1 , λ_2 , $\boldsymbol{y} = \boldsymbol{Bx}$
- *while* (objectif non atteint),
  1. $\boldsymbol{y} = \boldsymbol{Bx}$
  2. $\boldsymbol{w} = \left((E[y_1^2] - 1)y_1, \dots, \left(E\left[y_p^2\right] - 1\right)y_p\right)^T$
  3. Calculer $\hat{p}_{y_i}(y_i) = N_e(y_i)$, $i = 1, \dots, p.$
  4. Calculer $\hat{\psi}_{y_i}(y_i) = -\frac{\hat{p}'_{y_i}(y_i)}{\hat{p}_{y_i}(y_i)}$, $i = 1, \dots, p.$
  5. $\frac{d\ell}{d\boldsymbol{B}} = \hat{E}\{\hat{\boldsymbol{\psi}}_{\boldsymbol{y}}(\boldsymbol{y})\boldsymbol{x}^T\} - \boldsymbol{B}^{-T} + 4diag\left(\lambda_1, \dots, \lambda_p\right) \hat{E}\{\boldsymbol{wx}^T\}$
  6. $\boldsymbol{B}^{(k+1)} = \boldsymbol{B}^{(k)} - \mu \frac{d\ell}{d\boldsymbol{B}}$
  7. *For* $i = 1 : p$

     $\lambda_i^{(k+1)} = \lambda_i^{(k)} + \mu[var(y_i) - 1]^2$

     End for
- *End while.*

5.2.4. Résultats de Simulation

- **Simulation 1**

Pour tester cet algorithme sur des mélanges linéaires, prenons l'exemple de deux signaux source $s = (s_1, s_2)^T$ et deux mélanges tels que :

$$\begin{cases} s_1 = sin(5t) \\ s_2 \sim U[-1, 1] \end{cases}$$

Et la matrice de mélange : $\boldsymbol{A} = \begin{pmatrix} -2.29 & 0.49 \\ 1.84 & 0.41 \end{pmatrix}$

Le nombre des échantillons est pris : $N = 2000$.

La figure ci-dessous présente les signaux sources, les mélanges et les signaux estimés.

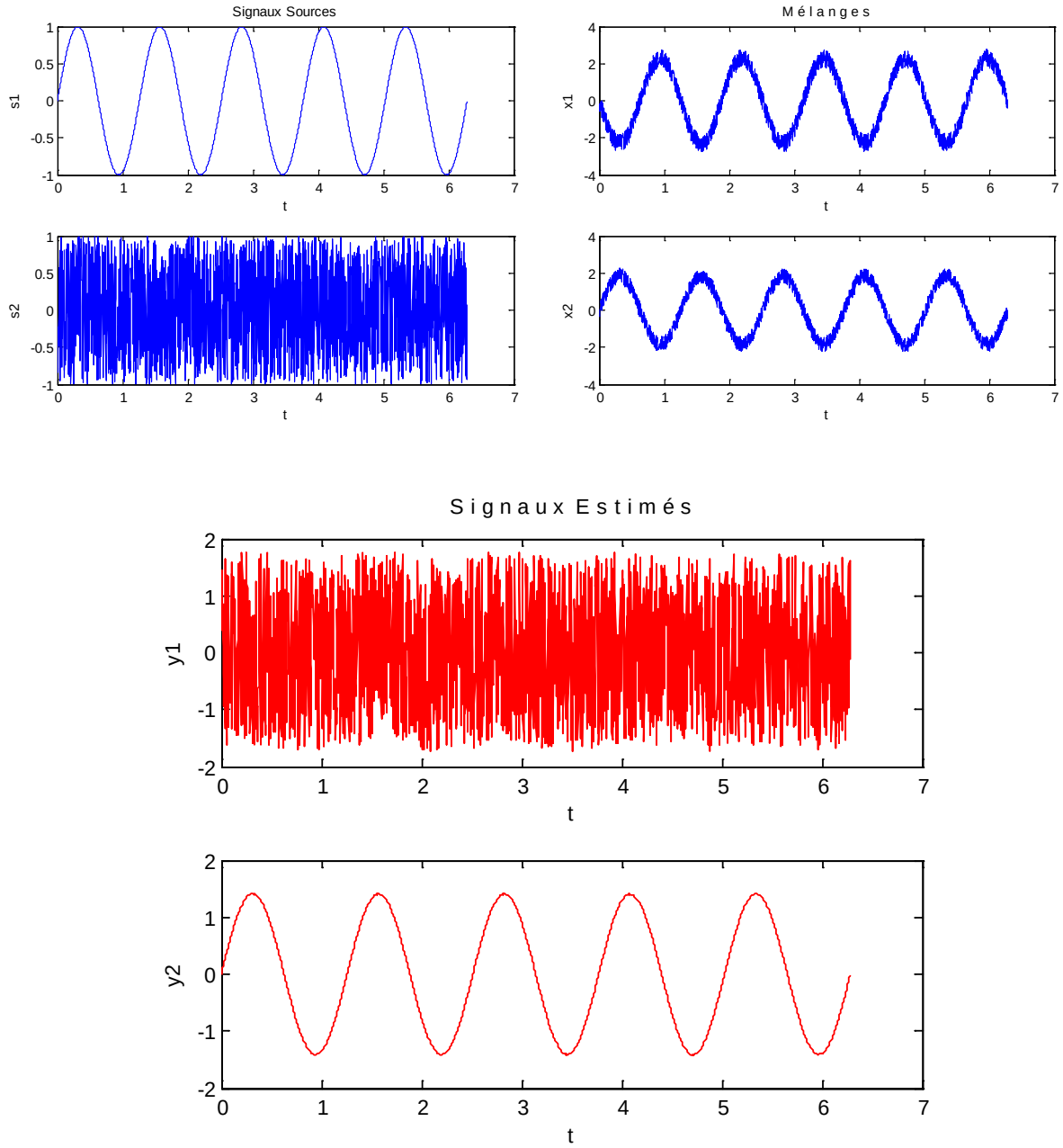


Fig. 5.11. Résultats de séparation : ($H = 2, d = 2$ pour l'estimation de la pdf)

Fig. 5.12 montre les distributions des différents signaux. On peut remarquer que les signaux estimés sont indépendants.

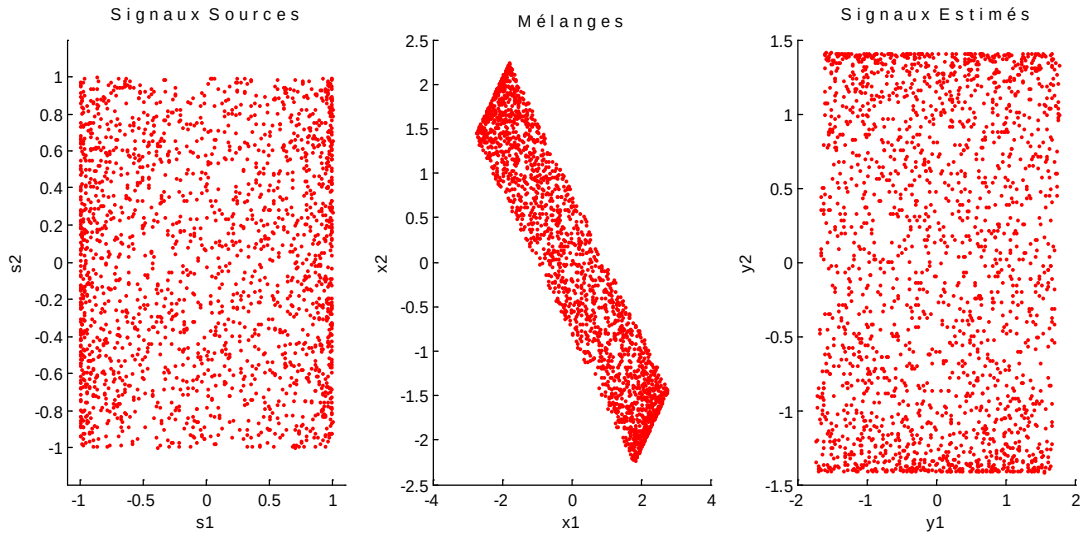


Fig. 5.12. Distributions des différents signaux : sources, mélanges, et signaux estimés.

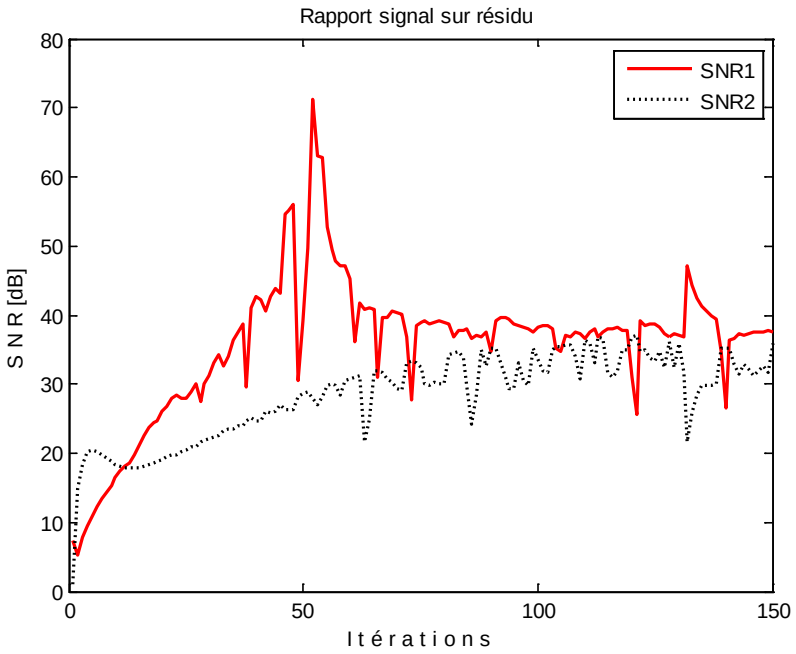


Fig. 5.13. Les SNRs des deux sources en fonction du nombre d'itérations (SNR1 : Source1, SNR2 : Source2).

- **Simulation 2**

Dans cet exemple nous prenons deux autres signaux pour $N = 2000$.

$$\begin{cases} s_1 = sin(5t) \\ s_2 = tri(4t) \end{cases}$$

Et la matrice de mélange : $\boldsymbol{A} = \begin{pmatrix} 1 & 0.5 \\ 1 & 0.7 \end{pmatrix}$

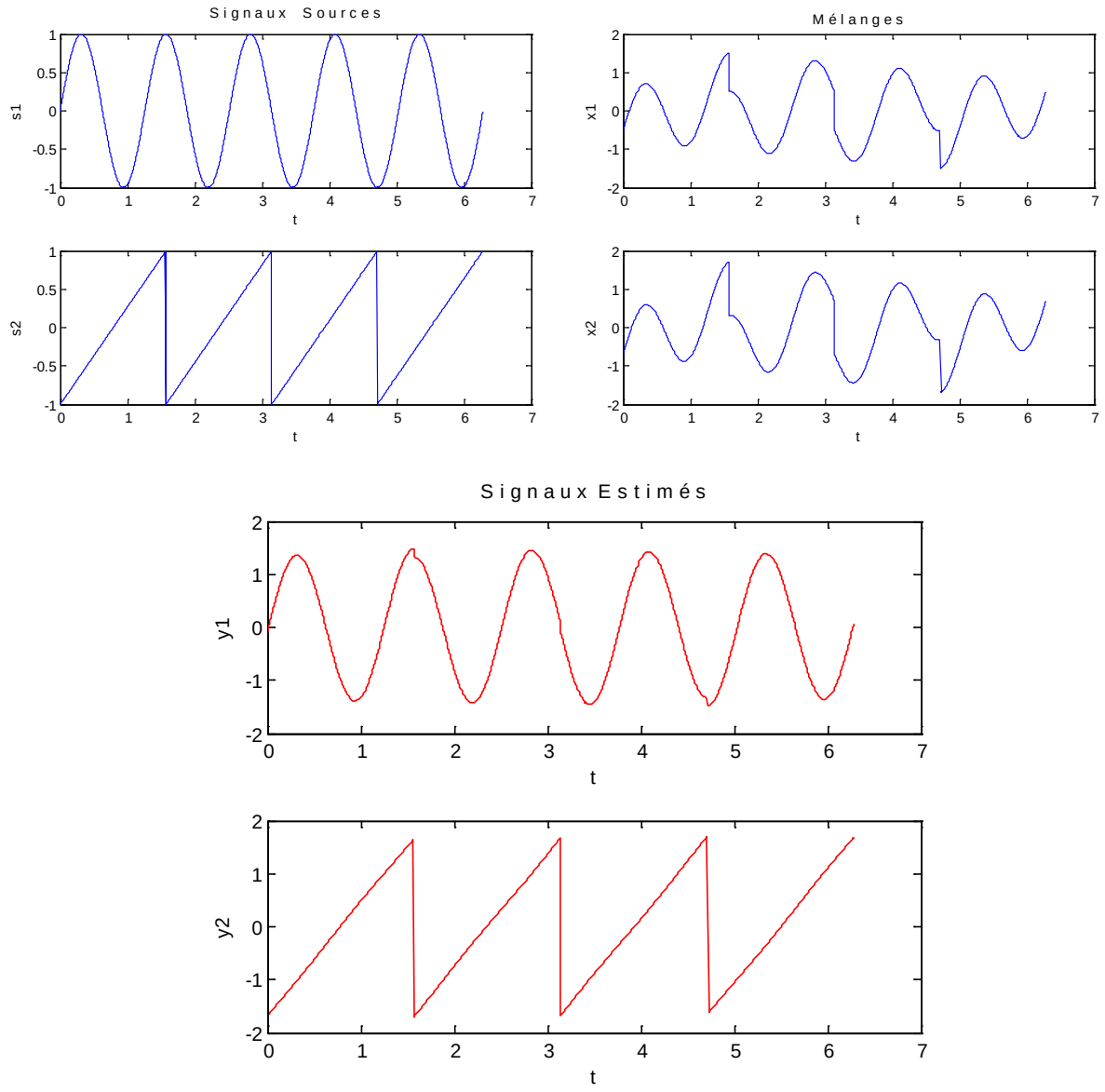


Fig. 5.14. Résultats de séparation : ($H = 2, d = 2$ pour l'estimation de la pdf)

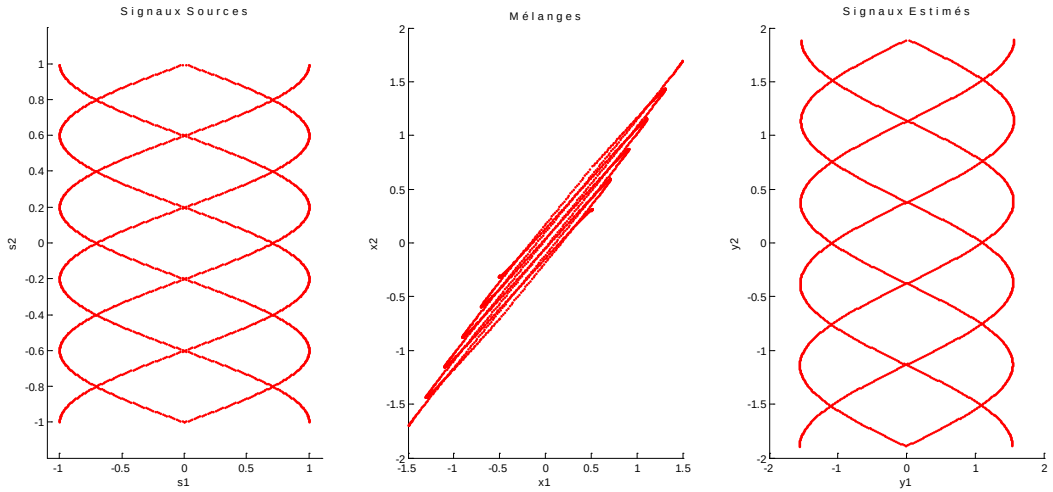


Fig. 5.15. Distributions des différents signaux : sources, mélanges, et signaux estimés.

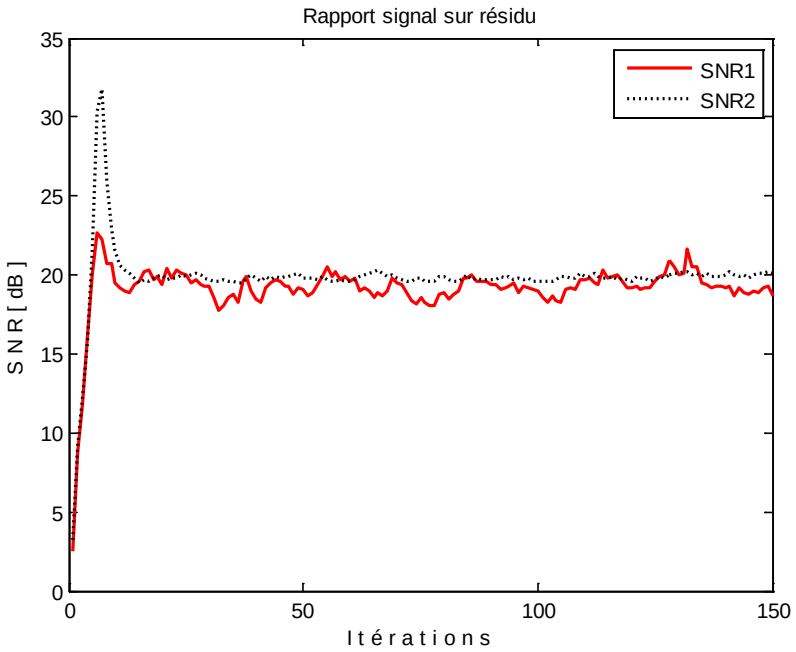


Fig. 5.16. Les SNRs des deux sources en fonction du nombre d'itérations (SNR1 : Source1, SNR2 : Source2).

- **Simulation 3**

Nous considérons un mélange de 03 sources, telles que :

$$\begin{cases} s_1 = sin(5t) \\ s_2 = tri(4t) \\ s_3 = carré(3t) \end{cases}$$

La matrice de mélange dans cet exemple est la suivante : $\boldsymbol{A} = \begin{pmatrix} 1 & 0.6 & 0.5 \\ 0.7 & 1 & 0.5 \\ 0.5 & 1 & 0.8 \end{pmatrix}$

Fig. 5.17 montre les résultats de séparation en présentant : les signaux sources, les mélanges, et enfin les signaux estimés.

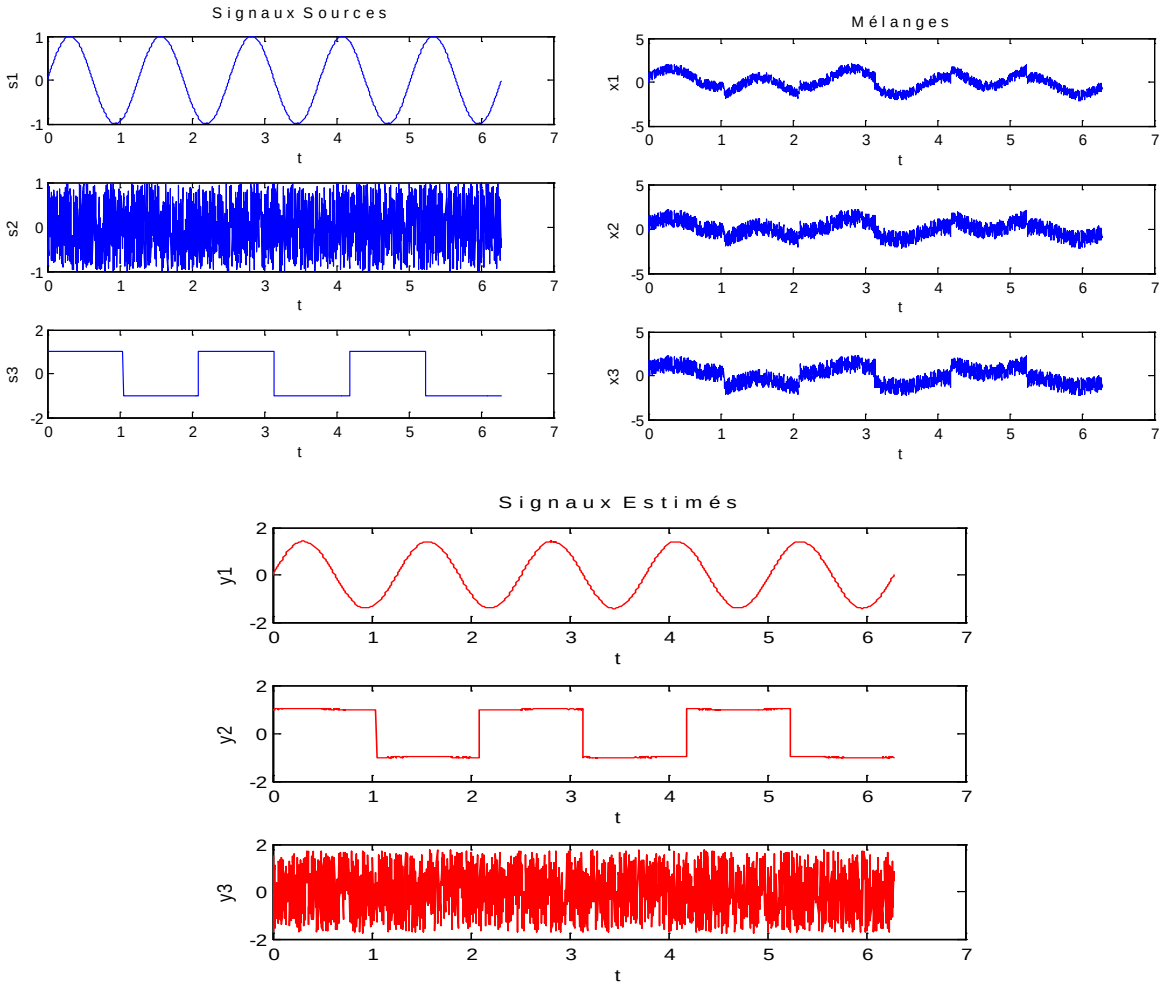


Fig. 5.17. Résultats de séparation : ($H = 2, d = 2$ pour l'estimation de la pdf)

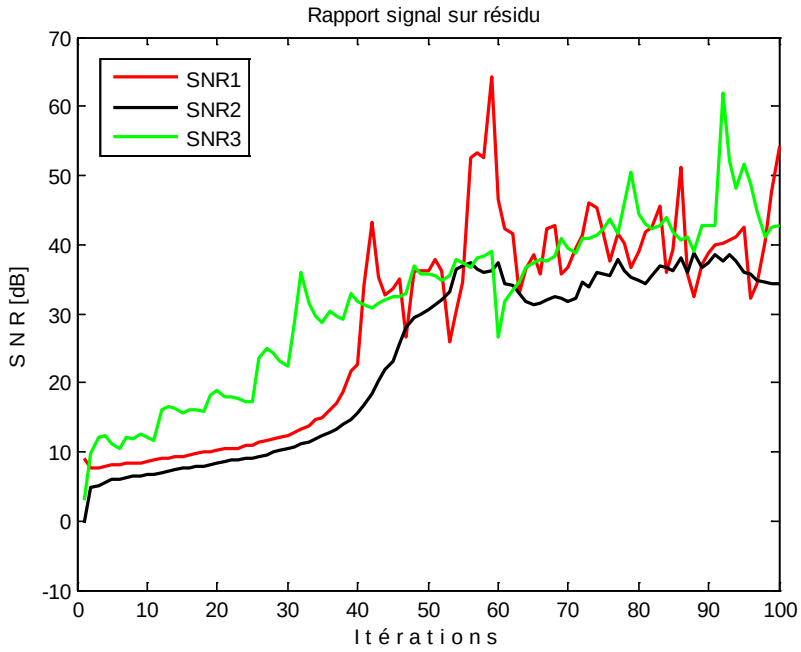


Fig. 5.18. Les SNRs des trois sources en fonction du nombre d'itérations (SNR1 : Source1, SNR2 : Source2, SNR3 : Source3).

A partir de la figure (5.18), nous pouvons dire que l'algorithme converge vers les trois signaux sources à partir de l'itération 55, et dans ce cas les SNRs sont de l'ordre de 35 $[dB]$.

Contrairement aux mélanges linéaires où la séparation des sources consiste à estimer une matrice $\boldsymbol{B}$ telle que $= \boldsymbol{Bx}$, les mélanges post-nonlinéaires, comme déjà vu, nécessite l'estimation de la transformation inverse G, telle que $\boldsymbol{z} = \boldsymbol{G}(\boldsymbol{x})$, avant de procéder à une séparation d'un mélange linéaire. L'étape d'estimation des nonlinéarités inverse est souvent appelée : Compensation des nonlinéarités.

5.3. SEPARATION DES MELANGES PNL

Pour pouvoir estimer la transformation non linéaire inverse $\boldsymbol{G}$ de $\boldsymbol{\mathcal{F}}$ conduisant à l'estimation des signaux sources, on est obligé à choisir une fonction cout qui dépend seulement de la fonction de densité de probabilité des sources. L'information mutuelle, l'entropie, la néguentropie, les cumulants d'ordre supérieur, sont des fonctions cout largement utilisées en littérature (voir Chapitre (4)).

L'inverse des fonctions non linéaires dans Le modèle PNL peut être estimé en utilisant un perceptron multicouche comprenant une seule couche cachée qui contient un nombre de neurones égal au nombre de sources à restituer.

Pour ce type de mélanges le signal observé s'écrit :

$$\boldsymbol{x(t)} = \boldsymbol{\mathcal{F}}\big(\boldsymbol{A}\, \boldsymbol{s(t)}\big)$$

$\boldsymbol{x(t)} = \big(x_1\,, x_2\,, \dots, x_p\big)^T$: vecteur des observations à l'instant t

$\boldsymbol{s(t)} = \big(s_1\,, s_2\,, \dots, s_p\big)^T$: vecteur des signaux sources à l'instant t

$\boldsymbol{A} = [a]_{ij}$: matrice de mélange de la partie linéaire

$\boldsymbol{\mathcal{F}}$: transformation non linéaire inversible de la partie non linéaire

Dans ce travail, on suppose que la transformation $\boldsymbol{\mathcal{F}}$ est inversible, et que son inverse $\boldsymbol{\mathcal{F}^{-1}}$ peut être approximé par un perceptron à deux couches (cachée + sortie) tel que le vecteur de sortie $\boldsymbol{y}$ s'exprime (Fig.5.19) :

$$\boldsymbol{y} = \boldsymbol{W_1 g(W_2 x + \theta)} \approx \boldsymbol{\mathcal{F}^{-1}(x)} \qquad (5.36)$$

$\boldsymbol{W_1}$ et $\boldsymbol{W_2}$ sont des matrices carré qui contiennent les poids des connexions.

$\boldsymbol{\theta}$ vecteur comprenant les biais de la couche cachée.

$\boldsymbol{g(u)} = (u_1, \ldots, u_p)^T$ est une fonction sigmoïde.

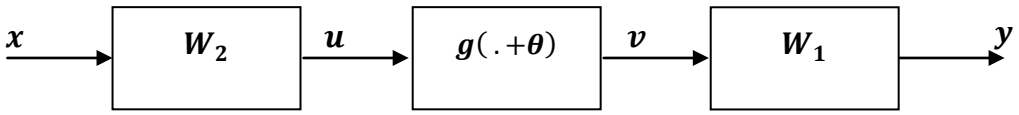


Fig. 5.19. Etage de séparation

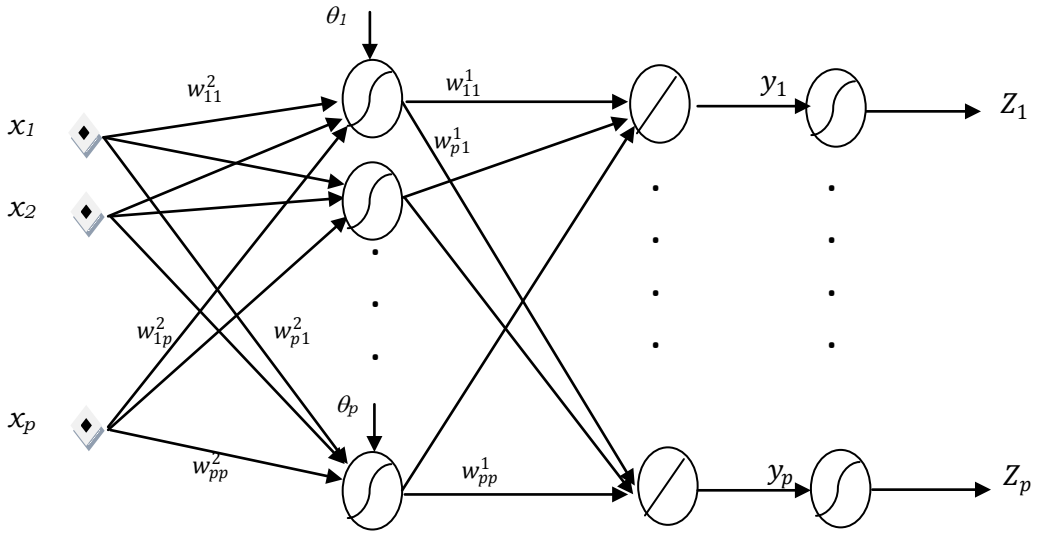


Fig. 5.20. MLP modélisant l'étage de séparation (mélange PNL – p source)

Les poids des connexions sont mise à jour en optimisant une fonction cout qui dépend directement de la pdf de $\boldsymbol{y}$. Cependant, il a été démontré dans la littérature que l'estimation des sources peut être achevée en maximisant leur entropie [L1], ou en minimisant le critère de l'information mutuelle [NL12].

5.3.1. Maximisation de l'entropie

L'entropie différentielle, $H_{\boldsymbol{y}}(\boldsymbol{y}) = -E\{\ln p_{\boldsymbol{y}}(\boldsymbol{y})\}$, des sorties du système de séparation (Fig.5.19) est généralement non bornée, et donc la recherche d'un maximum peut être impossible. Alors, afin de chercher un maximum d'une fonction bornée et être sure que ce maximum existe, on applique la fonction sigmoïde aux sorties $\boldsymbol{y}$:

$$\boldsymbol{Z} = \left(\sigma_1(y_1), \dots, \sigma_p(y_p)\right)^T \tag{5.37}$$

$\sigma_i(y_i)$ est une fonction sigmoïde dérivable deux fois.

Donc

$$H_{\boldsymbol{Z}}(\boldsymbol{Z}) = H_{\boldsymbol{v}}(\boldsymbol{v}) + ln|\boldsymbol{W}_1| + \sum_{i=1}^{p} E\{ln\, \sigma_i^{'}(y_i)\} \tag{5.38}$$

A partir des deux relations :

$$\boldsymbol{v} = \left(g_1(u_1 + \theta_1), \dots, g_p(u_p + \theta_p)\right)^T = \boldsymbol{g}(\boldsymbol{u} + \boldsymbol{\theta}) \tag{5.39}$$

et

$$\boldsymbol{u} = \boldsymbol{W}_2\boldsymbol{x} \tag{5.40}$$

Nous avons

$$H_{\boldsymbol{v}}(\boldsymbol{v}) = H(\boldsymbol{x}) + ln|\boldsymbol{W}_2| + \sum_{i=1}^{p} E\{ln\, g_i^{'}(u_i + \theta_i)\} \tag{5.41}$$

Notons les matrice des poids par $\boldsymbol{W}_k = \left[w_{ij}^k\right],\ pour\ \ k = 1, 2.$ dans ce cas, le gradient stochastique $\frac{\partial H_Z(\boldsymbol{Z})}{\partial w_{ij}^k}$ peut être calculé de la manière suivante :

A partir de l'équation (5.90) nous avons :

$$\frac{\partial H_Z(\boldsymbol{Z})}{\partial w_{ij}^1} = [\boldsymbol{W}_1^{-T}]_{ij} + \frac{\sigma_i^{''}(y_i)}{\sigma_i^{'}(y_i)} v_i \tag{5.42}$$

$[\boldsymbol{W}_1^{-T}]_{ij}$ est l'élément de la i^{ème} ligne et la j^{ème} colonne de la matrice $\boldsymbol{W}_1^{-T} = [\boldsymbol{W}_1^{-1}]^T$

En forme compacte, l'équation (5.94) s'écrit

$$\frac{\partial H_Z(\boldsymbol{Z})}{\partial \boldsymbol{W}_1} = \boldsymbol{W}_1^{-T} - \boldsymbol{\Psi}_1(\boldsymbol{y})\boldsymbol{v^T} \tag{5.43}$$

où

$$\boldsymbol{\Psi}_1(\boldsymbol{y}) = -\left(\frac{\sigma_1^{''}(y_1)}{\sigma_1^{'}(y_1)}, \ldots, \frac{\sigma_p^{''}(y_p)}{\sigma_p^{'}(y_p)}\right)^T$$

Comme déjà vu dans la section précédente (Fig. 5.9), si nous supposons que :

$F_i(y) = \gamma_1 \sigma_i(y) + \gamma_2 \qquad pour\ \gamma_1 \neq 0$, alors : $\frac{F_i^{''}(y)}{F_i^{'}(y)} = \frac{\sigma_i^{''}(y)}{\sigma_i^{'}(y)}$

Donc, sans perte de généralité, on peut supposer que : $\sigma_i(y) = \int_{-\infty}^{y} p_i(r)\, dr$, où p_i sont quelques pdf. Cependant, $\boldsymbol{\Psi}_1(\boldsymbol{y})$ peut être exprimée par la relation suivante :

$$\boldsymbol{\Psi}_1(\boldsymbol{y}) = -\left(\frac{p_1^{'}(y_1)}{p_1(y_1)}, \dots, \frac{p_p^{'}(y_p)}{p_p(y_p)}\right)^T = \left(\psi_1(y_1), \dots, \psi_p(y_p)\right)^T$$

Où $\psi_i(y_i)$ sont les fonctions score marginales.

A partir des expressions (5.90) et (5.93) nous avons

$$H_Z(\boldsymbol{Z}) = H_x(\boldsymbol{x}) + ln|\boldsymbol{W}_1| + ln|\boldsymbol{W}_2| + \sum_{i=1}^{p} E\{ln\, g_i^{'}(u_i + \theta_i)\} + \sum_{i=1}^{p} E\{ln\, \sigma_i^{'}(y_i)\} \tag{5.44}$$

$$\frac{\partial H_Z(\boldsymbol{Z})}{\partial \boldsymbol{W}_2} = \boldsymbol{W}_2^{-T} - \boldsymbol{\Psi}_2(\boldsymbol{u})\boldsymbol{x}^{\boldsymbol{T}} + \sum_{i=1}^{p} \frac{\partial\, ln\, \sigma_i^{'}(y_i)}{\partial\, \boldsymbol{W}_2} \tag{5.45}$$

Où $\boldsymbol{\Psi}_2(\boldsymbol{u}) = -\left(\frac{g_1^{''}(u_1+\theta_1)}{g_1^{'}(u_1+\theta_1)}, \dots, \frac{g_p^{''}(u_p+\theta_p)}{g_p^{'}(u_p+\theta_p)}\right)^T$

et

$$\frac{\partial\, y_i}{\partial\, w_{kl}^2} = w_{ik}^1\, g_k^{'}(u_k + \theta_k) x_l \tag{5.46}$$

Finalement

$$\frac{\partial H_Z(\boldsymbol{Z})}{\partial \boldsymbol{W}_2} = \boldsymbol{W}_2^{-T} - \boldsymbol{\Psi}_2(\boldsymbol{u})\boldsymbol{x}^{\boldsymbol{T}} - \boldsymbol{D}(\boldsymbol{u})\boldsymbol{W}_1^T\boldsymbol{\Psi}_1(\boldsymbol{y})\boldsymbol{x}^{\boldsymbol{T}} \tag{5.47}$$

Avec : $\boldsymbol{D}(\boldsymbol{u}) = diag\left(g_1^{'}(u_1 + \theta_1), \dots, g_p^{'}(u_p + \theta_p)\right)$

Et pour finir, on calcule l'expression $\frac{\partial H(\boldsymbol{Z})}{\partial \boldsymbol{\theta}}$.

$$\frac{\partial\, y_k}{\partial\, \theta_i} = w_{ki}^1 g_i^{'}(u_i + \theta_i) \tag{5.48}$$

Et donc à partir des deux derniers termes de l'expression (5.96) on obtient

$$\frac{\partial H_Z(\boldsymbol{Z})}{\partial \theta_i} = \frac{g_i^{''}(u_i+\theta_i)}{g_i^{'}(u_i+\theta_i)} + \sum_{k=1}^{p} \frac{\sigma_k^{''}(y_k)}{\sigma_k^{'}(y_k)} \frac{\partial y_k}{\partial \theta_i} \tag{5.49}$$

En remplaçant $\frac{\partial\, y_k}{\partial\, \theta_i}$ par son expression de (5.48) on obtient :

$$\frac{\partial H_Z(\boldsymbol{Z})}{\partial \theta_i} = \frac{g_i^{''}(u_i+\theta_i)}{g_i^{'}(u_i+\theta_i)} + \sum_{k=1}^{p} \frac{\sigma_k^{''}(y_k)}{\sigma_k^{'}(y_k)}\, w_{ki}^1 g_i^{'}(u_i + \theta_i) \tag{5.50}$$

Sous forme compacte $\frac{\partial H_Z(\boldsymbol{Z})}{\partial \boldsymbol{\theta}}$ est donné par l'expression suivante

$$\frac{\partial H_Z(\boldsymbol{Z})}{\partial \boldsymbol{\theta}} = -\boldsymbol{\Psi}_2(\boldsymbol{u}) - \boldsymbol{D}(\boldsymbol{u})\boldsymbol{W}_1^T\boldsymbol{\Psi}_1(\boldsymbol{y}) \tag{5.51}$$

L'algorithme d'apprentissage non-supervisé se résume par les équations finales suivantes :

$$\begin{cases} d\boldsymbol{W}_1 = \eta\{\boldsymbol{W}_1^{-T} - \boldsymbol{\Psi}_1(\boldsymbol{y})\boldsymbol{v}^{\boldsymbol{T}}\} \\ d\boldsymbol{W}_2 = \eta\{\boldsymbol{W}_2^{-T} - \boldsymbol{\Psi}_2(\boldsymbol{u})\boldsymbol{x}^{\boldsymbol{T}} - \boldsymbol{D}(\boldsymbol{u})\boldsymbol{W}_1^T\boldsymbol{\Psi}_1(\boldsymbol{y})\boldsymbol{x}^{\boldsymbol{T}}\} \\ d\boldsymbol{\theta} = -\eta\{\boldsymbol{\Psi}_2(\boldsymbol{u}) + \boldsymbol{D}(\boldsymbol{u})\boldsymbol{W}_1^T\boldsymbol{\Psi}_1(\boldsymbol{y})\} \end{cases} \tag{5.52}$$

Notons bien que cet ensemble d'équations forme la solution d'un problème d'optimisation sans contraintes.

Pour le calcul de $\boldsymbol{\Psi}_1(\boldsymbol{y})$, nous utilisons le bagage développé dans la section (5.2), par contre, $\boldsymbol{\Psi}_2(\boldsymbol{u})$ est calculée de la manière suivante :

Si nous prenons la fonction d'activation de la couche cachée telle que :

$v = g(u) = tanh(u)$, alors

$$g'(u) = tanh'(u) = 1 - tanh^2(u) = 1 - v^2 \quad (5.53)$$

Et

$$g''(u) = tanh''(u) = 2tanh(u)(tanh^2(u) - 1)$$

$$= 2v(v^2 - 1) \quad (5.54)$$

Et enfin

$$\frac{g''(u)}{g'(u)} = -2v \quad (5.55)$$

La même chose pour la fonction logistique :

$v = g(u) = logsig(u)$, alors

$$g'(u) = logsig'(u) = (1 - v)v \quad (5.56)$$

Et

$$g''(u) = logsig''(u) = (1 - 2v)(1 - v)v \quad (5.57)$$

Et donc

$$\frac{g''(u)}{g'(u)} = 1 - 2v \quad (5.58)$$

Pour valider le développement précédent, prenons des exemples de simulation où nous pouvons voir les mélanges PNL ainsi que leurs transformations inverses.

5.3.2. Résultats de simulations (mélanges PNL)

- **Simulation 4**

Dans cet exemple nous considérons les mêmes signaux sources que dans la simulation 1 de la section précédente

$$\begin{cases} s_1 = sin(5t) \\ s_2 \sim U[-1, 1] \end{cases}$$

Le mélange PNL est de la forme :

$$\boldsymbol{x} = tanh(\boldsymbol{As})$$

Où

$\boldsymbol{A}$ est la matrice du mélange de la partie linéaire dont les éléments sont choisis de manière aléatoire d'une loi uniforme sur l'intervalle $[-0.5, 0.5]$.

La figure (5.21) présente les signaux sources, les mélanges PNL, ainsi que l'espace des différents types de signaux.

Dans la figure (5.22) nous présentons les signaux estimés.

La figure (5.23) montre les inverses estimés des nonlinéarités pour les deux signaux mélanges en présentant aussi les inverses réels. On peut voir que les estimés des deux fonctions inverses sont proche de celles réelles.

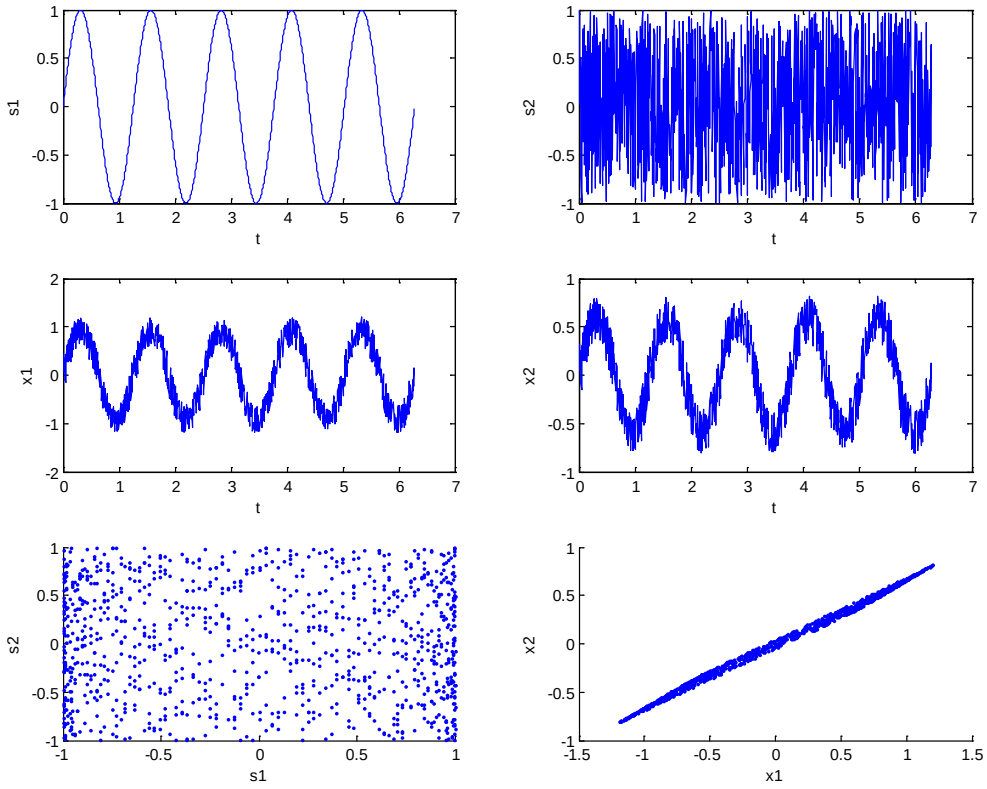


Fig. 5. 21. De haut en bas : signaux sources, mélanges,
et espace des signaux sources et mélanges

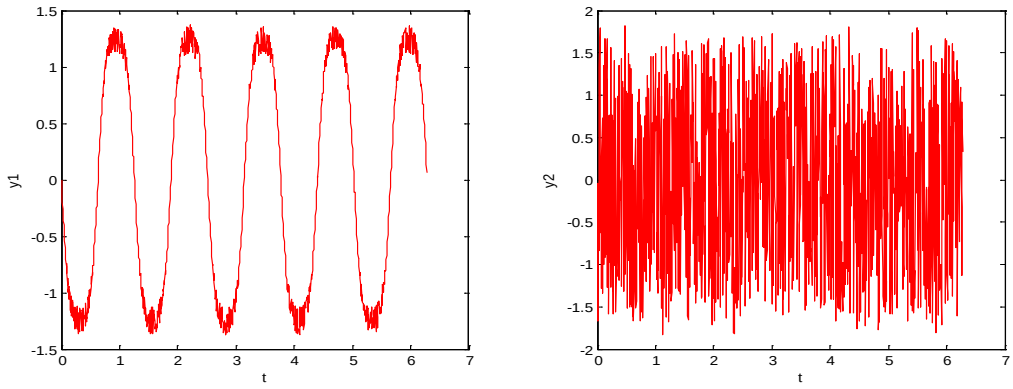


Fig. 5. 22. Signaux estimés

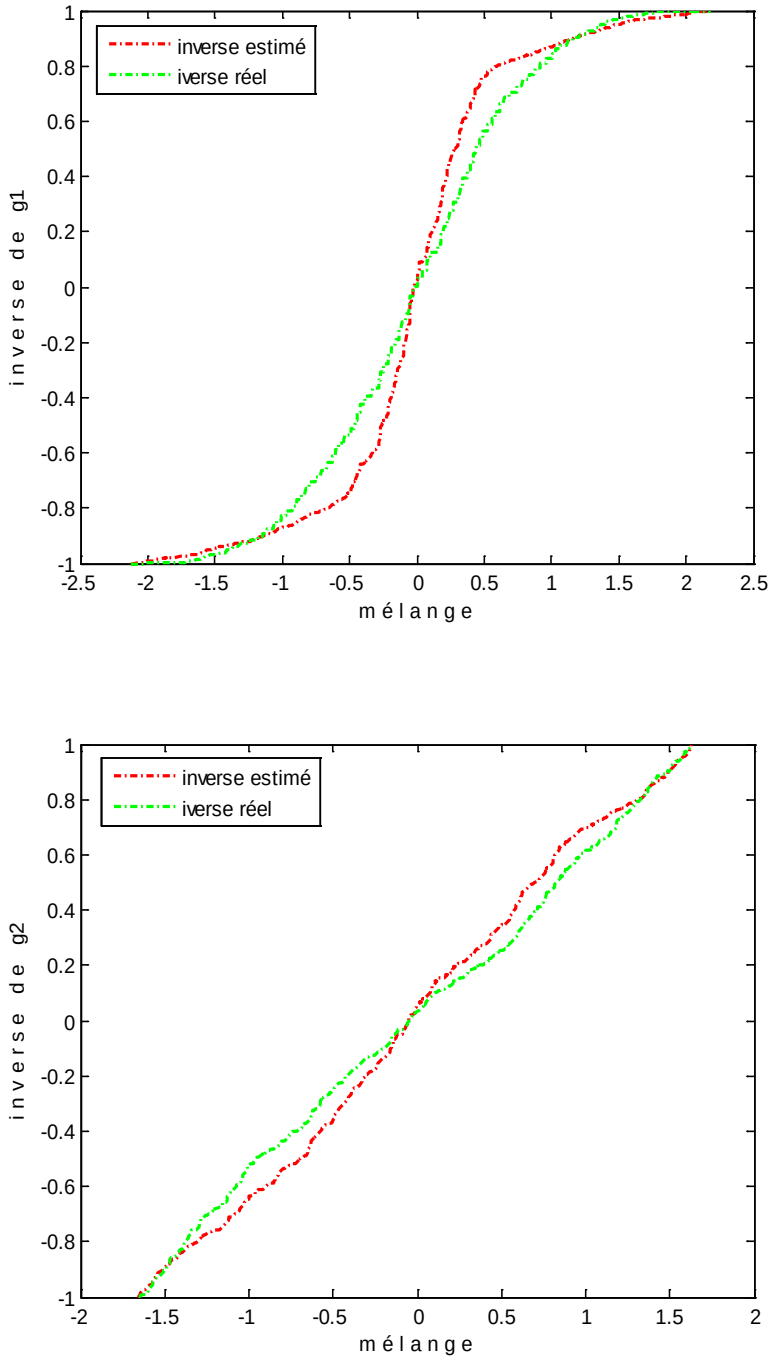


Fig. 5. 23. Nonlinéarités inverses : estimées et réelles

- **Simulation 5**

Dans cet exemple de simulation nous prenons les mêmes signaux sources mélangés par la transformation PNL suivante et nous allons comparer notre algorithme avec celui de FastICA :

$$x_1 = 0.1e_1 + e_1^3$$

$$x_2 = 0.3e_2 + tanh(3e_2)$$

Où

$$\boldsymbol{e} = \boldsymbol{A}\boldsymbol{s}$$

Telle que : $$\boldsymbol{A} = \begin{pmatrix} -2.29 & 0.49 \\ 1.84 & 0.41 \end{pmatrix}$$

La figure (5.24) présente la distribution (dispersion dans l'espace signal) des signaux sources ainsi que leurs mélanges.

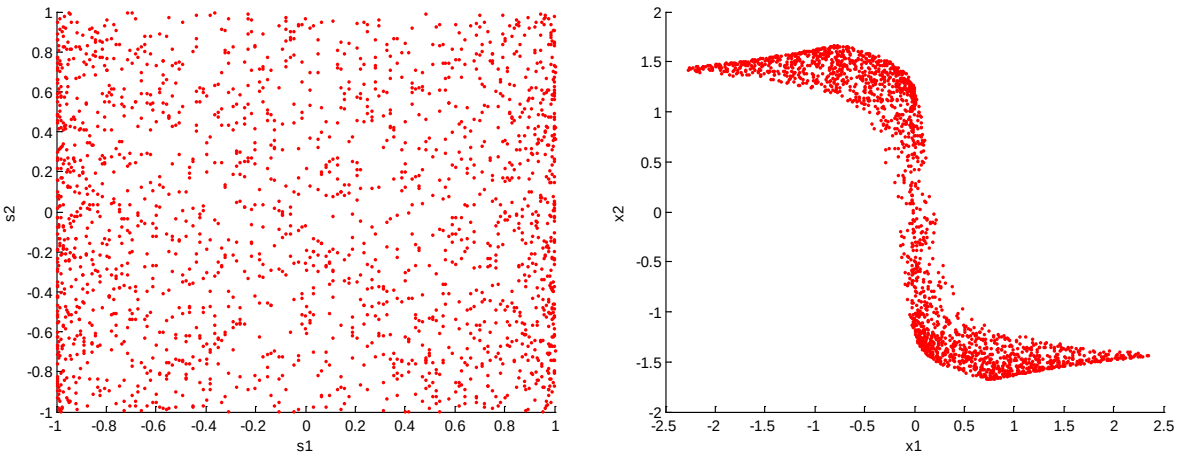


Fig. 5. 24. Distribution des signaux sources et mélanges

Après apprentissage du réseau, nous avons eu les résultats suivants :

La figure (5.25) montre que les signaux estimés sont une copie des signaux sources avec une très légère erreur d'estimation, contrairement à l'algorithme FastICA qui échoue à la restitution des signaux.

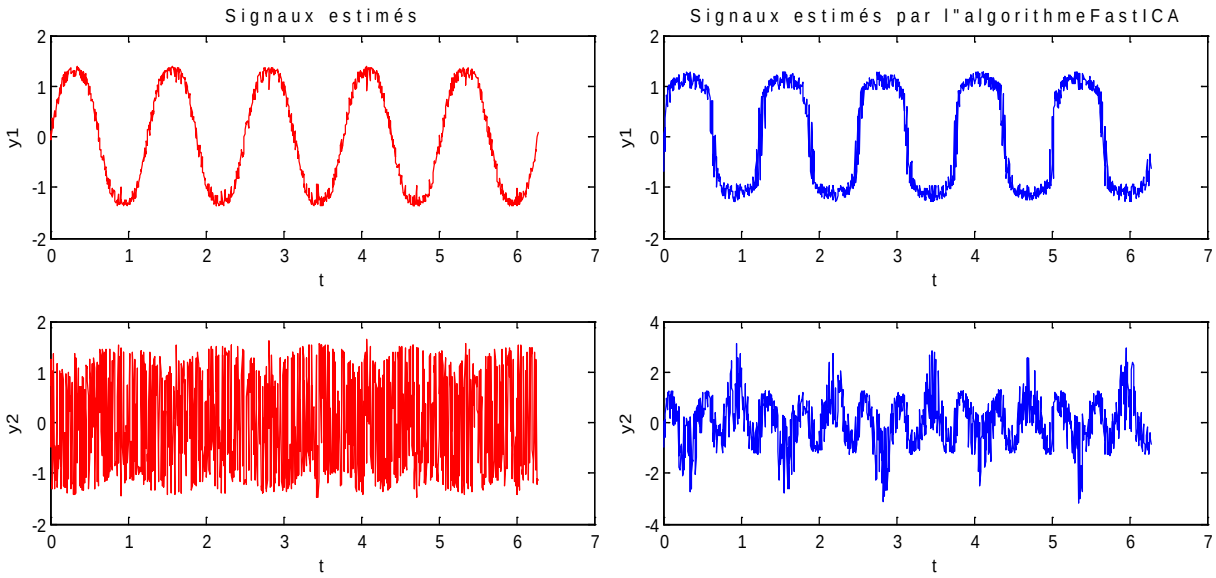


Fig. 5.25 Signaux estimés : à gauche MLP et à droite par l'algorithme FastICA

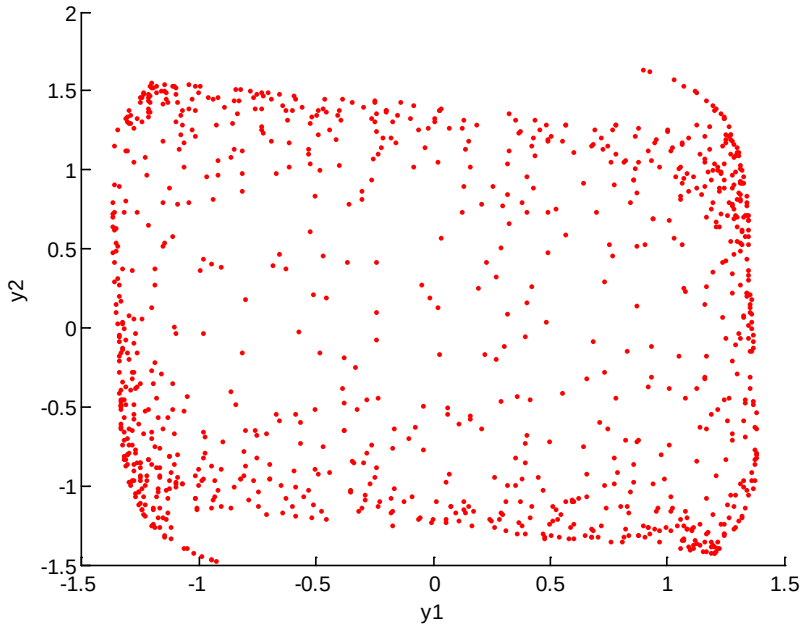


Fig. 5.26. Distribution des signaux estimés

On peut clairement voir, à partir de la figure (5.26), que les deux signaux estimés sont pratiquement statistiquement indépendants. La figure (5.27) présente les fonctions nonlinéaires inverses estimées ainsi que celles réelles.

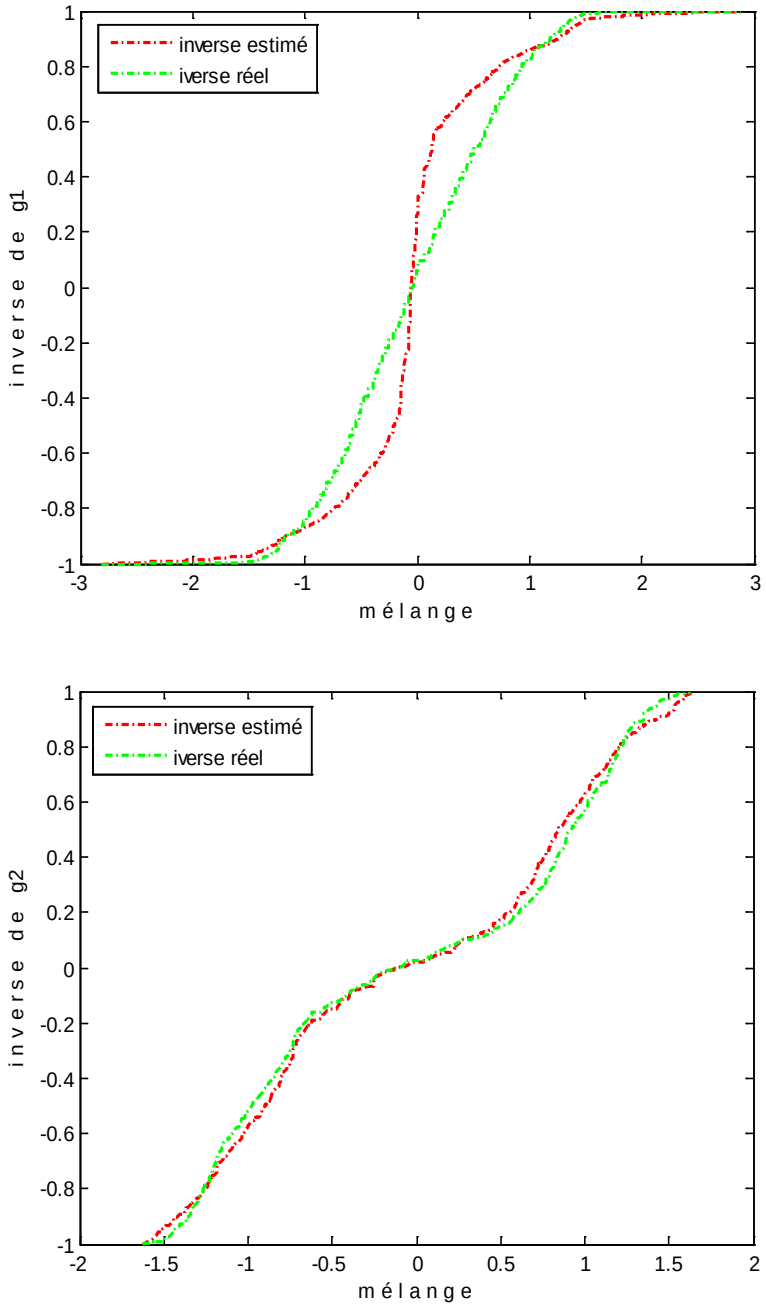


Fig. 5.27. Nonlinéarités inverses : estimées et réelles

- **Simulation 6**

Pour terminer, soit un dernier exemple, où quatre signaux sources sont mélangés de sorte à avoir un mélange PNL. Dans la partie linéaire la matrice de mélange est la suivante

$$\boldsymbol{A_2} = \begin{pmatrix} 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1 \end{pmatrix}$$

Et $\boldsymbol{x} = \boldsymbol{A_1} tanh(\boldsymbol{A_2 s})$

Où les éléments de $\boldsymbol{A_1}$ sont uniformément distribués sur l'intervalle [-1,1].

$$\boldsymbol{s} = \begin{cases} s_1 = sin(5t) \\ s_2 = tri(4t) \\ s_3 = signe\big(sin(3t)\big) \\ s_4 \sim U[-1,1] \end{cases}$$

Dans la figure (5.28) sont présentés les quatres signaux sources ainsi que les quatres différents mélanges.

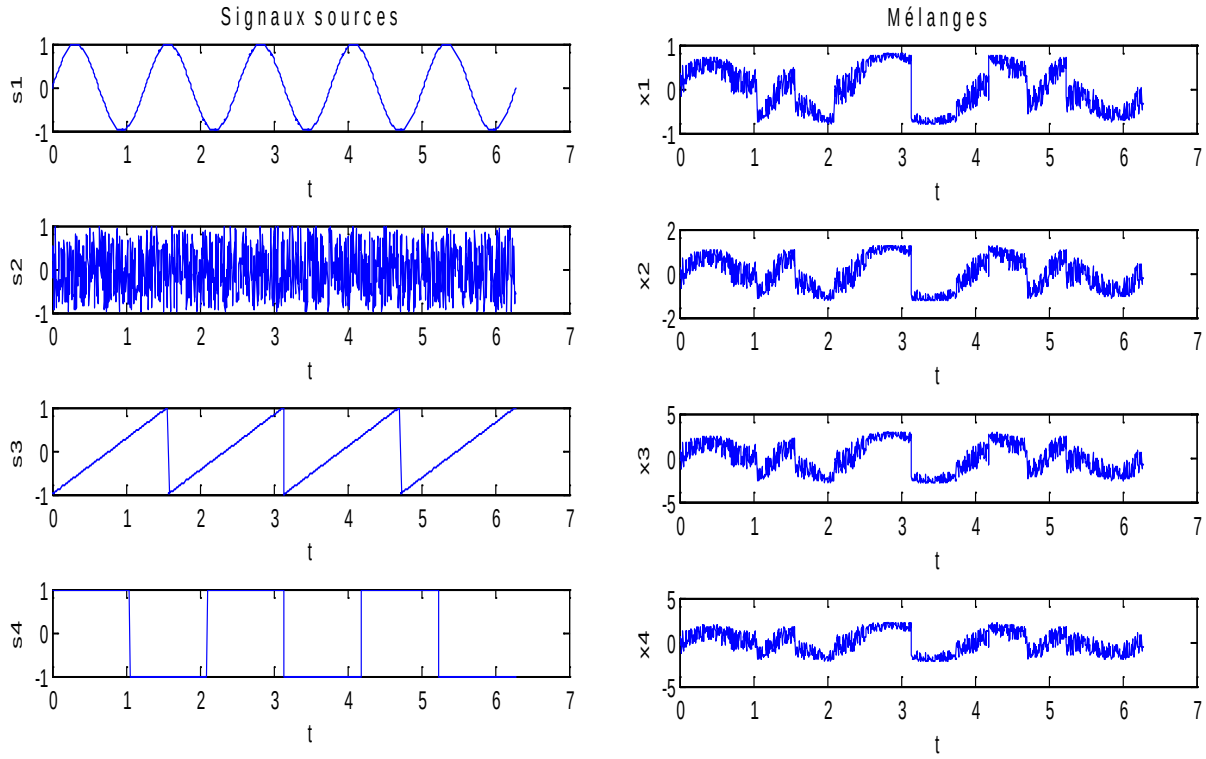


Fig. 5.28. De gauche à droite : Signaux sources, et mélanges PNL

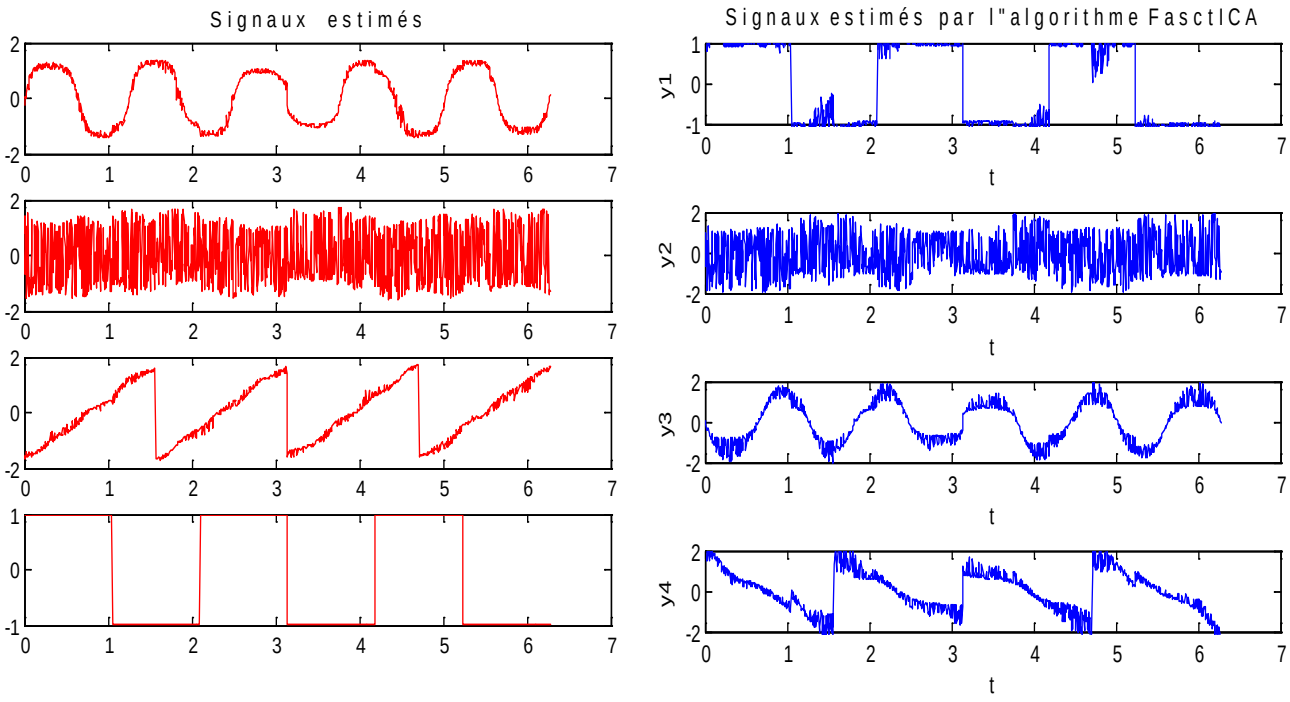


Fig. 5.29. Signaux estimés par MLP et Signaux estimés par l'algorithme FastICA

A partir de la figure (5.29) on remarque une légère déformation des signaux estimés par rapport aux signaux d'origine. Cependant, malgré cette déformation l'allure de ces signaux est très claire et ils sont très proches des sources. Mias, la déformation présente dans les signaux restitués par l'algorithme FastICA est très importante et plus significative que l'algorithme présenté.

5.4. CONCLUSION

Ce chapitre regroupe toutes les contributions de ce travail de recherche. Nous avons commencé par la modélisation des lois de probabilité par réseau de neurones allant d'un neurone seule jusqu'au perceptron multicouches selon la complexité de la loi de densité de probabilité. Cette modélisation est basée sur la loi exponentielle dont le choix est justifié par le fait que les densités exponentielles s'adaptent pour la modélisation de la plupart des densités usuelles (normale, uniforme, exponentielles, …etc.). Cette modélisation de loi de probabilité marginale a été implantée dans un algorithme de séparation de source en minimisant un critère d'information mutuelle sous contraintes utilisant les fonctions scores marginales estimées à partir des pdf marginales. Les résultats de simulation dans cette première partie montrent non seulement l'efficacité de la modélisation neuronale des pdf, mais aussi la bonne qualité de séparation des signaux sources à partir des mélanges linéaires.

La deuxième partie de ce chapitre a été consacrée au développement d'un algorithme d'apprentissage non supervisé basé sur descente de gradient en utilisant un réseau de neurone de type perceptron multicouches à une seule couche cachée. En se basant sur la fonction d'entropie des sortie du réseau, nous avons établi les équations de mise à jour des poids de connexion et des biais du

réseau afin d'aboutir à la valeur optimale de la fonction coût qui est le maximum de l'entropie des sorties. Les résultats de simulation dans cette deuxième partie, montrent aussi l'efficacité de la modélisation neuronale de la partie séparante du mélange post nonlinéaire en compensant les nonlinéarités de la partie linéaire et en estimant la matrice de séparation de la partie linéaire. Ces résultats sont vérifiés pour le cas de deux et de quatre sources.

Conclusion Générale

6.1. CONCLUSIONS

Aujourd'hui, les réseaux de neurones artificiels ont de nombreuses applications dans de variétés de secteurs :

- Traitement d'images : reconnaissance de formes, classification, compression d'images, reconnaissance de caractères, etc.
- Traitement du signal : traitement de la parole, identification de signaux, filtrage, etc.
- Optimisation : Allocation de ressource, planification, etc.
- Contrôle : Commande de processus, asservissement de robots, etc.
- Simulation : Simulation de boites noires, prévision météorologique, etc.

L'architecture d'un réseau de neurones est l'organisation des neurones entre eux au sein d'un même réseau. Autrement dit, il s'agit de la façon dont ils sont ordonnés et connectés.

Une nouvelle génération de réseaux de neurones, capables de traiter avec succès des phénomènes non-linéaires : le perceptron multicouche (MLP : Multi

Layer Perrceptron) qui ne possède pas les défauts mis en évidence par *Minsky* (1969) a été introduite en 1986.

Dans ce travail, nous avons exploité la structure multicouche du MLP dans le but de restituer des signaux mélangés (linéairement et non linéairement) en ne disposant que d'une seule connaissance a priori : l'indépendance statistique de ces signaux sources, d'où l'application des réseaux MLP en SAS.

Pour des mélange linéaires, un algorithme d'optimisation sous contraintes de l'information mutuelle a fait appel à l'estimation des fonctions score marginales déduites des fonctions de densité de probabilité marginales dont l'estimation a fait appel à plusieurs idées de modélisation. Cependant, une modélisation des pdf marginale par une structure MLP traduisant une loi exponentielle dont les paramètres ont été estimés de manière non supervisée a été appliquée avec succès. Les résultats de simulation ont montré une qualité d'estimation très bonne et un rapport signal sur résidus important.

Une deuxième catégorie des mélanges nonlinéaires qui est largement rencontrée dans des situations réelles est celle des mélanges Post-NonLinéaires (PNL). Ce type de mélange qui présente un étage linéaire suivi d'une transformation nonlinéaire a été modélisé par un réseau MLP à une seule couche cachée pour la compensation des nonlinéarités (estimation des fonctions nonlinéaires inverses) d'une part, et pour restituer les signaux sources en optimisant un critère de performance basé sur l'entropie des sortie de ce réseau d'autre part. Les équations de mise à jour des poids et des biais ont été développées à partir du critère de descente du gradient. Cet algorithme d'apprentissage non supervisé maximisant l'entropie des sorties a donné de bons résultats de simulations.

6.2. PERSPECTIVES

Dans l'algorithme de Babie-Zadeh [NL28, NL32], l'auteur fait appel à l'estimation de la fonction score différentielle par les méthodes : Histogramme, méthode du noyau, méthode de PHAM [L60] et enfin la méthode polynomiale. Cependant, il serait très intéressant de développer une méthode neuronale pour l'estimation de la densité de probabilité conjointe multidimensionnelle.

Il serait aussi très intéressant de prendre un nombre plus élevé de neurones dans la couche cachée du réseau MLP (supérieur au nombre de sources), et de développer les équations de mise à jour des paramètres du réseau, ce qui pourrait améliorer énormément la qualité d'estimation et la vitesse d'apprentissage.

Une dernière application qui est très intéressante est celle de l'estimation des directions d'arrivée par réseaux d'antennes. Dans cette discipline, peu de travaux basés sur la modélisation neuronale ont été réalisés. Cependant, on peut envisager quelques architectures de réseaux d'antennes et chercher une modélisation neuronale qui prend en compte ces architectures ainsi que les fonctions cout correspondantes.

ANNEXES

ANNEXE A

• Matrice jacobienne

La matrice jacobienne est la matrice des dérivées partielles du premier ordre d'une fonction vectorielle (multivariables).

Soit F une fonction d'un ensemble de $\mathcal{R}^n$ à valeurs dans $\mathcal{R}^m$. Une telle fonction est définie par ses m fonctions composantes à valeurs réelles :

$$F : \begin{pmatrix} x_1 \\ . \\ . \\ . \\ x_n \end{pmatrix} \longmapsto \begin{pmatrix} f_1(x_1, \dots, x_n) \\ . \\ . \\ . \\ f_m(x_1, \dots, x_n) \end{pmatrix} \qquad \text{(A.1)}$$

Les dérivées partielles de ces fonctions en un point M, si elles existent, peuvent être rangées dans une matrice à m lignes et n colonnes, appelée matrice jacobienne de F :

$$J_F(M) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & . & . & . & \frac{\partial f_1}{\partial x_n} \\ & & . & & \\ & & . & & \\ & & . & & \\ \frac{\partial f_m}{\partial x_1} & . & . & . & \frac{\partial f_m}{\partial n} \end{pmatrix} \qquad \text{(A.2)}$$

ANNEXE B

• Information Mutuelle du modèle PNL

L'information mutuelle du vecteur des sorties $\boldsymbol{y}$ peut être décomposé sous la forme suivante :

$$I(\boldsymbol{y}; \boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{\theta}) = -H(\boldsymbol{v}) - log|\boldsymbol{W}_1| + \sum_{i=1}^{p} H(y_i) \qquad \text{(B.1)}$$

Pour que l'équation (B.1) soit utile, nous avons besoin de trouver un estimateur des entropies marginales. On peut penser dans ce cas au développement de Gram-Charlier [L54] :

$$H(y_{\boldsymbol{k}}) \approx \widehat{H}(y_{\boldsymbol{k}}) = \frac{1}{2} ln(2\pi e) - \frac{(\rho_3^k)^2}{2.3!} - \frac{(\rho_4^k)^2}{2.4!} + \frac{3}{8}(\rho_3^k)^2 \rho_4^k + \frac{1}{16}(\rho_4^k)^3 \qquad \text{(B.2)}$$

Où

$$\rho_3^k = m_3^k \quad , \quad \rho_4^k = m_4^k - 3$$

avec

$$m_i^k = E[(y_k)^i]$$

Dans ce cas, l'équation (B.1) devient alors :

$$I(\boldsymbol{y}; \boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{\theta}) \approx -H(\boldsymbol{v}) - log|\boldsymbol{W}_1| + \sum_{k=1}^{p} \widehat{H}(y_k) \qquad \text{(B.3)}$$

En combinant l'équation (5.41) du cinquième chapitre et l'équation (B.3), nous pouvons calculer les expression des gradients par rapport aux paramètres du réseau : $\frac{\partial I}{\partial \boldsymbol{W}_1}$, $\frac{\partial I}{\partial \boldsymbol{W}_2}$ et $\frac{\partial I}{\partial \boldsymbol{\theta}}$.

Bibliographie

[B1] S. Haykin. “Neural Networks - A Comprehensive Foundation.” Prentice Hall, 1998, Second edition.

[B2] A. Hyvärinen, J. Karhunen, and E. Oja. “Independent Component Analysis.” John Wiely & Sons, 2001.

[B3] A. Papoulis, “Probability, Random Variables, and Stochastic Processes”, McGraw-Hill, 1991.

[B4] C. Jutten et P. Comon, "Séparation de sources - Tome 2 : au-delà de l’aveugle et applications", chapitre 13 par Y. Deville. Collection "Traité IC2, Information - Commande - Communication", Hermès - Lavoisier, Paris, 2007.

[L1] A. Bell and T. Sejnowski. ‘’An information-maximization approach to blind separation and blind deconvolution’’. Neural computation, Vol. 7, pp. 1129-1159, 1995.

[L2] A. Belouchrani and M. G. Amin. “Blind source separation based on time-frequency signal representations”. IEEE Trans. on Signal Processing, Vol. 46, pp. 2888-2897, 1998.

[L3] A. Belouchrani and M. G. Amin. “On the use of spatial time frequency distributions for signal extraction”. Multidimensional Systems and Signal Processing, Special issue of the journal, Vol. 9, No. 4, pp. 349-354, October 1998.

[L4] A. Belouchrani. “Séparation autodidacte de sources : Algorithmes, Performances et Applications à des signaux expérimentaux“. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 1995.

[L5] A. Hyvärinen. ‘’ A unifying model for blind separation of independent sources’’. Signal processing, Vol. 85, pp. 1419-1425, 2005.

[L6] A. Mansour, M. Kawamoto, and N. Ohnishi. “Blind separation for instantaneous mixture of speech signals : Algorithms and performances”. IEEE Conf., pages 26_32, 2000.

[L7] A. Mansour. “Blind separation of sources : Methods, assumptions and applications”. IEICE TRANS. Fundamentals, Vol. E83-A, No. 8, pp. 1798-1512, August 2000.

[L8] B. Ans, J. Hérault, and C. Jutten, “Adaptive neural architectures: Detection of primitives”. in *Proceedings of COGNITIVA’85*, Paris, pp. 593-597, June 1985.

[L9] C. Jutten and A. Taleb. “Source separation: From dusk till dawn”. In ICA2000, Helsinki, Finland, pp. 15-26, June 2000.

[L10] C.S. Ray, and Y. Singh. ‘’Source distribution models for blind source separation’’. Neurocomputing, Vol. 57, pp. 501-505, 2004.

[L11] D. Nuzillard, and J.M. Nuzillard. ‘’ Second-order blind separation in the Fourier space of data’’. Signal processing, Vol. 83, pp. 627-631, 2003.

[L12] D.-T. Pham and J.-F. Cardoso. “Blind separation of instantaneous mixtures of non stationary sources”. IEEE Trans. Signal Processing, Vol. 49, N°.9, pp.1837-1848, 2001.

[L13] E. Weinstein, M. Feder and A. Oppenheim. ‘’ Multi-channel signal separation by decorrelation”. IEEE Trans. On speech and Audio Processing, Vol. 1; no. 4, pp. 405-413, Oct. 1993.

[L14] F. Vrins, and M. Verleysen. ‘’ On the entropy minimization of a linear mixture of variables for source separation’’. Signal processing, Vol. 85, pp. 1029-1044, 2005.

[L15] H. Broman, U. Lindgren, H. Sahlin and P. Stoica. ‘’ Source separation: A TITO System Identification Approach’’. signal Processing, Vol. 73, pp. 169-183, 1999.

[L16] J. Cardoso. '' Blind signal separation: statistical principles''. proceedings of IEEE Special issue on blind system identification and estimation, Vol.9, pp. 2009-2025, Oct. 1998.

[L17] J. Even and E. Moisan. ''Blind source separation using order statistics''. Signal processing, Vol. 85, pp. 1744-1758, 2005.

[L18] J. Hérault and C. Jutten. "Space or time adaptive signal processing by neural networks models". In Intern. Conf. on Neural Networks for Computing, Snowbird (Utah, USA), pp. 206-211, 1986.

[L19] J. Hérault, C. Jutten, and B. Ans. "Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé". in *Actes du X^{ème} colloque GRETSI*, Nice, France, pp. 1017-1022, Mai 1985.

[L20] J.-F. Cardoso and A. Souloumiac. "Blind beamforming for non gaussian signals". IEE proceeding, Vol. 140, pp.362-370, 1993.

[L21] K. I. Diamantaras. "Oriented PCA and blind signal separation". In ICA proceedings, pp. 609-613, April 2003.

[L22] L. Parra and C. Spence. '' Convolutive blind source separation of non-stationary sources''. IEEE Trans. On speech and Audio processing, pp. 320-327, May 2000.

[L23] L. Tong, V. Soon, Y. F. Huang, and R. Liu. "Amuse : a new blind identification algorithm". In Proceeding of IEEE ISCAS, pp. 1784-1787, 1990.

[L24] M. Taoufiki, A. adib, and D. Aboutadjdine. ''Blind separation of any source distributions via high-order statistics''. Signal processing, Vol. 87, pp. 1882-1889, 2007.

[L25] M. V. Hulle. '' Clustering approach to square and non-square blind source separation''. in IEEE Workshop on Neural Networks for signal Processing (NNSP), Madison, Wisconsin, Aug. 23-25 1999, pp. 315-323.

[L26] P. Comon. "Blind identification and source separation in 2x3 under-determined mixtures". IEEE Trans. on Signal Processing, Vol. 52, No. 1, pp. 11-22, 2004.

[L27] P. Comon. '' Blind Channel Identification and Extraction of more sources than sensors''. in SPIE Conference, San Diego, july 19-24 1998, pp. 2-13.

[L28] P. Jallon, and A. Chevreuil. "Separation of instantaneous mixtures of cyclo-stationary sources". Signal processing, Vol. 87, pp. 2718-2732, 2007.

[L29] R. Ballan, A. Jourjine, and J. Rosca. '' A particular case of singular multivariate AR identification and BSS problems''. in 1 st International Conference on Independent Component Analysis, Assuis, France, 1999.

[L30] S. Bermejo. "Finite sample effects of the fast ICA algorithm". Neurocomputing, Vol. 71, pp. 392-399, 2007.

L31] S. Ou, X. Zhao, and Y. Gao. ''Linear system identification employing independent component analysis''. International conference on Automation and Logistics, Jinan (China), August 2007.

[L32] S. Samadi, M.B. Zadeh, C. Jutten, and K. Nayebi. ''Blind source separation by adaptive estimation of score function difference''. On idependent component analysis and blind signal separation, LINCS 3195, pp. 9-17, 2004.

[L33] S.H. Mellado, R. Martín-Clemente, C.G. Puntonet, J.I. Acha, and J.M.G. Sáez.. ''Maximization of statistical moments for blind separation of sources revisited''. Neurocomputing, Vol. 69, pp. 1425-1434, 2006.

[L34] S.-I. Amari, and A. Cichocki "Adaptive blind signal processing-neural network approaches". Proceedings of the IEEE, Vol. 86, No. 10, pp. 2026-2048, October 1998.

[L35] T.Y. Sun, C.C. Liu, S.J. Tsai, and S.T. Hsieh. ''Blind source separation with dynamic source number using adaptive neural algorithm''. Expert systems with applications, Vol. 36, pp. 8855-8861, 2009.

[L36] T.Y. Sun, C.C. Liu, S.T. Hsieh, and S.J. Tsai. ''Blind separation with unknown number of sources based on auto-trimmed neural network''. Neurocomputing, Vol. 71, pp. 2271-2280, 2008.

[L37] X. Zhu, X. Zhang, and Y. Su. ''A fast NPCA algorithm for online blind source separation''. Neurocomputing, Vol. 69, pp. 964-968, 2006.

[L38] Y. Deville, and M. Puigt. "Temporal and time-frequency correlation-based blind source separation methods. Part I: Determined and underdetermined linear instantaneous mixtures". Signal processing, Vol. 87, pp. 374-407, 2007.

[L39] Z. shi and C. Zhang. ''Fast nonlinear autocorrelation algorithm for source separation''. Pattern recognition, Vol. 42, pp. 1732-1741, 2009.

[L40] Z. Shi, and C. Zhang. ''Nonlinear innovation to blind source separation''. Neurocomputing, Vol. 71, pp. 406-410, 2007.

[L41] Z. Shi, Z. Jiang, and F. Zhou. ''A fixed-point algorithm for blind source separation with non linear autocorrelation''. Journal of computational and applied mathematics, Vol. 223, pp. 908-915, 2009.

[L42] Z. Shi, Z. Jiang, F. Zhou, and J. Yi. ''Blind source separation with nonlinear autocorrelation and non-Gaussianity''. Journal of computational and applied mathematics, Vol. 229, pp. 240-247, 2009.

[L43] Z.L. Zhang, and Z. Yi. "Robust extraction of specific signals with temporal structure". Neurocomputing letters, Vol. 69, pp. 888-893, 2006.

[L44] R. L. L. Tong and Y. H. V. C. Soon. "Indeterminacy and identifiability of blind identification". IEEE Trans. CS, 38 :499-509, 1991.

[L45] P. Comon. "Independent component analysis, a new concept ? ", IEEE Signal Processing, Vol.36, N°.3, pp. 287-314, 1994

[L46] G. Darmois. "Analyse générale des liaisons stochastiques'', Rev. Inst. Internat. Stat., 21:2-8, 1953.

[L47] A. Belouchrani, K. Abed Meraim, J. F. Cardoso, and E. Moulines. ''A blind source separation technique based on second order statistics'', IEEE Transactions on Signal Processing, 45(2):434-444, 1997.

[L48] J.F. Cardoso. ''High-order contrasts for independent component analysis'', Neural Computation, N°. 11, pp. 157-192, 1999.

[L49] G. Darmois. "Analyse des liaisons de probabilité'', in proc. Of int. Statistics conferences, Vol.III A, Washington (D.C), pp.231, 1951.

[L50] N.Delfosse, and P. Loubaton, "Adaptive blind separation of independent sources: a deflation approach", Signal Processing, Vol. 45, pp. 59–83, 1995.

[L51] P. Huber, "Projection pursuit", The Annals of Statistics, Vol. 13, N°. 2, pp. 435-475, 1985.

[L52] A. Hyvärinen, "Survey on independent component analysis", Neural Computing Surveys, Vol.2, pp. 94–128, 1999.

[L53] M. Jones and R. Sibson, "What is projection pursuit?", Journal of the Royal Statistical Society, pp. 1–36, 1987.

[L54] A. Hyvärinen, "New approximations of differential entropy for independent component analysis and projection pursuit", In Advances in Neural Information Processing Systems, MIT Press, Vol. 10, pp. 273–279, 1998.

[L55] E. Moreau and J. C. Pesquet, "Generalized contrasts for multichannel blind deconvolution of linear systems", IEEE Signal Processing Letters, Vol. 4, N°. 6, pp. 182-183, 1997.

[L56] E. Moreau and O. Macchi, “New self-adaptatif algorithms for source separation based on contrast functions”, In Proc. HOS'93, SP Workshop on Higher-order Statistics, Lake Tahoe, USA, N°.2, pp. 215-219, June 1993.

[L57] N. Thirion and E. Moreau, “New criteria for blind signal separation”, in IEEE Workshop on Statistical Signal and Array Processing, Pennsylvania, US, pp. 344-348, 2000.

[L58] P. Comon, ”From source separation to blind equalization, contrast-based approaches”, in ICISP 01, Int. Conf. on Image and Signal Processing, Agadir, Morocco, pp. 20-32, 2001

[L59] A. Taleb and C. Jutten, “Entropy optimization, application to blind source separation”, in ICANN, Lausanne, Switzerland, pp. 529–534, October 1997.

[L60] D. T. Pham, “Mutual information approach to blind separation of stationary sources”, in Proceedings of ICA’99, Aussois, France, pp. 215–220, January 1999.

[L61] A. Belouchrani and J.-F. Cardoso, “Maximum likelihood source separation for discrete sources”, in Proc. EUSIPCO, pp. 768-771, 1994.

[L62] A. Belouchrani and J.-F. Cardoso, “Maximum likelihood source separation by the expectation-maximization technique : deterministic and stochastic implementation”, in Proc. International Symposium on Non-Linear Theory and Applications NOLTA, Las Vegas, NV, USA, pp. 49-53, 1995.

[L63] J.-F. Cardoso, “Infomax and maximum likelihood for blind source separation”, IEEE Letters on Signal processing, Vol.4, N°.4, pp.112-114, April 1997.

[L64] S. I. Amari, “Neural learning in structured parameter spaces-natural riemannian gradient. Neural Information Processing System Natural and Synthetic”, Colorado, USA, pp. 127-133, December 1996.

[L65] E. Moulines, J.-F. Cardoso, and E. Gassiat, “Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models”, In Proceedings of ICASSP-97, Munich, Germany, April 21-24, 1997.

[L66] M. Gaeta and J.-L. Lacoume, “Estimateurs du maximum de vraisemblance étendus à la séparation de sources non gaussiennes”, Traitement du Signal, Vol.7N°.5, pp. 419-434, 1990.

[L67] M. Gaeta and J.-L. Lacoume, “Source separation without a priori knowledge: the maximum likelihood solution”, In EUSIPCO, pp. 621-624, 1990.

[L68] D.-T. Pham, P. Garat, and C. Jutten, “Separation of a mixture of independent sources through a maximum likelihood approach”, In Proc. EUSIPCO, pp. 771-774, 1992.

[L69] M.M. Touba and S. Touba “Probability density function estimation using neural networks : Application to Blind source separation of linear instantaneous mixtures”, Conférence Internationale Sciences, Electroniques, Technologie de l’Information et des Télécommunications, 23-26 Mars 2011, Sousse, Tunisie.

[L70] A. Kachenoura, L. Albera, L. Senhadji, "Séparation aveugle de sources en ingénierie biomédicale", IRBM, Vol. 28, N°. 2, pp. 20-34, Mars 2007.

[L71] O. Berné et al., "Analysis of the emission of very small dust particles from Spitzer spectro-imagery data using Blind Signal Separation methods ", Astronomy & Astrophysics, vol. 469, N° 2, pp. 575-586, juillet 2007.

[LC1] A.Westner and V. M. Bove. “Applying blind source separation and deconvolution to real-world acoustic environments”. Proceeding 106th of Audio Engineering Society, 1999.

[LC2] C. Jutten, L. Nguyen Thi, E. Dijkstra, E. Vittoz, and J. Caelen. “Blind separation of sources : an algorithm for separation of convolutive mixtures”. Int. Signal Processing Workshop on Higher Order Statistics, pp. 273-276, July 1991.

[LC3] E. Weinstein, M. Feder, and A. V. Oppenheim. "Multi-channel signal separation by decorrelation". IEEE Trans. on Speech and Audio Processing, Vol. 1, No 4, pp. 405-413, 1993.

[LC4] H. Sawada, R. Mukay, S. Araki, and S. Makino. ''A robust and precise method for solving the permutation problem of frequency-domain blind source separation''. IEEE transactions on speech and audio processing, Vol. 12, No. 5, pp. 530-538, September 2004.

[LC5] H.-C. Wu and J. C. Principe. "Simultaneous diagonalization in the frequency domain (SDIF) for source separation". In Proceeding of ICA, pp. 245-250, 1999.

[LC6] H.-L. Nguyen Thi and C. Jutten. "Blind source separation for convolutive mixtures". Signal Processing,Vol. 45, pp. 209-229, 1995.

[LC7] I. Lee, T. Kim, and T.W. Lee. ''Fast fixed-point independent vector analysis algorithms for convolutive blind source separation''. Signal processing, Vol. 87, pp. 1859-1871, 2007.

[LC8] J. Anemüler and B. Kollmeier. "Amplitude modulation decorrelation for convolutive blind source separation". In Proceeding of ICA, pp. 215-220, June 2000.

[LC9] L. Parra and C. Spence. "Convolutive blind source separation of non-stationary sources". In Proceeding of IEEE Trans. on Speech and Audio Processing, Vol. 8, No. 3, pp. 320-327, May 2000.

[LC10] L. Parra and C. Spence. "On line blind source separation of non stationary signals". J. VLSI Signal Proceeding Systems for Signal, Images and Video Tech., Vol. 26, No. 8 , pp. 15-24, 2000.

[LC11] N. Mitianoudis and M. Davies. "Audio source separation of convolutive mixtures". IEEE Trans. on Speech and Audio Processing, Vol. 11, No. 5, pp. 489-497, Septembre 2003.

[LC12] N. Murata and S. Ikeda. "An on-line algorithm for blind source separation on speech signals". In Proceeding of NOLTA, 1998.

[LC13] P. Smaragdis. "Blind source separation of convolved mixtures in the frequency domain". In International Workshop on Independence & Artificial Neural Networks, Feb. 1998.

[LC14] R. Mukai, S. Araki, and S. Makino. "Separation and dereverberation performance of frequency domain blind source separation for speech in a reverberant environment". In Proceeding of Eurospeech 2001, pp. 2599-2603, 2001.

[LC15] R. Mukai, S. Araki, and S. Makino. "Separation and dereverberation performance of frequency domain blind source separation". In Proceeding of ICA, pp. 230-235, 2001.

[LC16] R. Mukai, S. Araki, H. Sawada, and S. Makino. "Evaluation of separation and dereverberation performance in frequency domain blind source separation". Acoustical Science and Technology, Vol. 25, No. 2, pp. 119-126, Mars 2004.

[LC17] S.C. Douglas, M. Gupta, H. Sawada, and S. Makino. ''Spatio-temporal FastICA algorithms for the blind separation of convolutive mixtures''. IEEE transactions on audio, speech, and language processing, Vol. 15, No. 5, pp. 1511-1520, July 2007.

[LC 18] T. Kim, H.T. Attias, S.-Y. Lee, and T.-W. Lee. ''Blind source separation exploiting higher-order frequency dependencies''. IEEE transactions on audio, speech, and language processing, Vol. 15, No. 1, pp. 70-79, January 2007.

[LC19] T. Nishikawa, H. Saruwatari, K. Shikano, and S. Makino. "Multistage ica for blind source separation of real acoustic convolutive mixture". In Proceeding of ICA, pp. 523-528, 2003.

[LC20] T. Takatani, T. Nishikawa, H. Saruwatari, and K. Shikano. "High-fidelity blind separation for convolutive mixture of acoustic signals using SIMO model based independent component analysis". In Proceeding of ISSPA, 2003.

[LC21] W. Wang, J. A. Chambers, and S. Sanei. "A joint diagonalization method for convolutive blind separation of nonstationary sources in the frequency domain". In Proceeding of ICA'03, pp. 939-944, 2003.

[LC22] Y. Zhou, and B. Xu. ''Blind source separation in frequency domain''. Signal processing, Vol. 83, pp. 2037-2046, 2003.

[NL1] A. Honkela, H. Valopa, A.Ilin, and J. Karhunen . ''Blind separation of nonlinear mixtures by variational bayesian learning.'' dDigital signal processing, Vol. 17, pp. 914-934, 2007.

[NL2] A. Taleb and C. Jutten. "Batch algorithm for source separation in post-nonlinear mixtures." in ICA'99, Aussois, France, January 1999, pp. 155–160

[NL3] A. Taleb and C. Jutten. "Source separation in post nonlinear mixtures." IEEE Transactions on Signal Processing, vol. 47, no. 10, pp. 2807–2820, 1999.

[NL4] D. Malathi, and N. Gunasekaran. ''A novel learning algorithm for separation of blind signals.'' International journal of soft computing 4(1), pp. 16-24, 2009.

[NL5] C. Jutten and J. Karhunen. ''Advances in nonlinear blind source separation.'' In proceedings of the 4th international symposium on independent component analysis and blind signal separation (ICA2003), pp. 245-256, April 2003.

[NL6] C. jutten, M.B. Zadeh, and S. Hosseini. ''Three easy ways for separating nonlinear mixtures.'' Signal processing, Vol. 84, pp. 217-229, 2004.

[NL7] C. Wei, W.L. Woo, and S.S. Dlay. ''Nonlinear underdetermined blind signal separation using bayesian neural network approach.'' Digital signal processing, Vol. 17, pp. 50-68, 2007.

[NL8] G. Burel. "Blind separation of sources: a nonlinear neural algorithm", Neural Networks, vol.5, N°. 6, pp. 937–947, 1992.

[NL9] G. Deco and W. Brauer. "Nonlinear higher-order statistical decorrelation by volume conserving architectures". Neural Networks, vol. 8, pp. 525–535, 1995.

[NL10] H. Valpola, X. Giannakopoulos, A. Honkela, and J. Karhunen. "Nonlinear independent component analysis using ensemble learning: Experiments and discussion." in ICA2000, Helsinki, Finland, 2000, pp. 351–356.

[NL11] H. Valpola. "Nonlinear independent component analysis using ensemble learning: Theory." in ICA2000, Helsinki, Finland, 2000, pp. 251–256.

[NL12] H.-H. Yang, S.I. Amari, and A. Cichocki. "Information-theoretic approach to blind separation of sources in non-linear mixtures." Signal Processing, pp. 291–300, February 1998.

[NL13] L. B. Almeida. "ICA of linear and nonlinear mixtures based on mutual information." in International Joint Conference on Neural Networks, Washington, DC, USA, July 2001.

[NL14] P. Pajunen and J. Karhunen, "A maximum likelihood approach to nonlinear blind source separation." in ICANN97, Lausanne (Switzerland), October 1997, pp. 541–546.

[NL15] P. Pajunen, A. Hyvärinen, and J. Karhunen. "Non linear source separation by self organizing maps." in ICONIP 96, Hong-Kong, September 1996.

[NL16] R. Martín-Clemente, S. Hornillo-Mellado, J.I. Acha, and C.G. Puntonet,.. ''Neural Net with two hidden layers for non-linear blind source separation.'' Proceedings of the international joint conference on neural networks, Vol.1, pp. 726- 731, July 2003.

[NL17] R. Martín-Clemente, S.H. Mellado, J.I. Acha, F. Rojas, and C.G. Puntonet. '' MLP-based sourcce separation for MLP-like non-linear mixtures.'' 4th international symposium on independent component analysis and blind signal separation (ICA2003), April 2003, Nara, Japan.

[NL18] S. Achard and D.-T. Pham. "Blind source separation in post nonlinear mixtures." in *ICA2001*, San Diego, California, December 2001, pp. 295–300.

[NL19] S. Achard, and C. Jutten. ''Identifiability of post nonlinear mixtures.'' IEEE Signal processing letters, Vol. 12, NO. 5, May 2005

[NL20] S. Achard, D.T. Pham, and C. Jutten. ''Criteria based on information minimization for blind source separation in post non linear mixtures.'' Signal processing, Vol. 85, pp. 965-974, 2005

[NL21] S. Achard. "Initiation a la Séparation aveugle de sources dans des mélanges post non linéaires." DEA de l'INP de Grenoble, June 2000, (in French).

[NL23] W. Y. Leong, W. Liu, and D.P. Mandic. ''Blind source extraction : Standard approaches and extensions to noisy and post-nonlinear mixing.'' Neurocomputing, Vol. 71, pp. 2344-2355, 2008.

[NL24] Y. Deville, and S. Hosseini. ''Recurrent networks for separating extractable-target nonlinear mixtures. Part I: Non-Blind configurations.'' Signal processing, Vol. 89, pp. 378-393, 2009.

[NL25] A. Ziehe, M. Kawanabe, S. Harmeling, and K.-R. Müler. "Separation of post-nonlinear mixtures using ACE and temporal decorrelation". in ICA2001, pp. 433-438, San Diego (California), December 2001,

[NL26] M.B.-Zadeh, C. Jutten, and K. Nayebi. "Blind separating Convolutive Post-Nonlinear mixtures," in *ICA2001*, San Diego, California, pp. 138-143,December 2001.

[NL27] M.B.-Zadeh, C. Jutten, and K. Nayebi. "A geometric approach for separating Post Non-Linear mixtures," in *EUSIPCO*, vol. 2, Toulouse, France, pp. 11-14, September 2002.

[NL29] V. Danielle, P. Rafaelle, and U. Aurelio. "A Novel recurrent network for independent component analysis of post-nonlinear convolutive mixtures". IEEE international conference on acoustics, speech, and signal processing, No. 29, Montreal (Canada), May 2004.

[NL28] M.B. Zadeh, and C. Jutten. "A general approach for mutual information minimization and its application to blind source separation". Signal processing, Vol. 85, pp. 975-995, 2005.

[NL30] J. Zhang, L.C. Khor, W.L. Woo, and S.D. Satnam. "A maximum likelihood approach to nonlinear convolutive blind source separation". 6th international conference ICA(2006), Charleston (USA), March 2006.

[NL31] J. Eriksson and V. Koivunen, "Blind identifiability of a class of nonlinear instantaneous ICA models". In Proceedings of EUSIPCO 2002, Toulouse (France), September 2002.

[NL32] A. M. Kagan, Y. V. Linnik, and C. R. Rao, "Extension of Darmois-Skitovich theorem to functions of random variables satisfying an addition theorem". Communications in Statistics, 1(5), pp.471–474, 1973.

[NL33] S. Achard, D.T. Pham, and C. Jutten, "Quadratic dependence measure for non linear blind sources separation", In Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation, ICA2003, Nara, Japan, pp. 263-268, Avril 2003.

[NL34] M.M. Touba and R. Ksouri, " Blind Source Separation of Post-Non-Linear Mixtures using Multilayer Perceptron ", International Conference on digital ecosystems and technologies. Dubai Knowledge Village, United Arab Emirates, 12-15 April 2010.

[pdf1] H. White, “Mathematical perspectives on Neural Networks”, M. Moser, D. Rumelhart (Eds), 1992.

[pdf2] A. Likas, “Probability density estimation using artificial neural networks”, Computer physics communications, Vol. 135, pp. 167-175, 2000.

[Th1] C. Jutten. “Calcul neuromimétique et traitement du signal : analyse en composantes indépendantes“, Thèse d’état ès sciences physiques“. UJF-INP Grenoble, 1987.

[Th2] A. Taleb. “Séparation de sources dans des mélanges post non-linéaires“, Thèse de l’INP de Grenoble, 1999.