Universität Karlsruhe (TH)
Institut für
Musikwissenschaft/Musikinformatik

Blind Source Separation

Tobias Gehrig

Projektarbeit

Verantwortlicher Betreuer (Prof.): Prof. Dr. Thomas A. Troge

19. September 2007

Hiermit erkläre ich, die vorliegende Arbeit selbständig verfaßt und keine anderen als die angegebenen Literaturhilfsmittel verwendet zu haben.

I hereby declare that this thesis is a work of my own, and that only cited sources have been used.

Karlsruhe, den 19. September 2007

_____

Tobias Gehrig

Abstract

Diese Ausarbeitung soll einen Überblick über existierende instantane und konvolutive Blind Source Separation (BSS) Algorithmen verschaffen. Hierfür werden einerseits die zur Verfügung stehenden Optimierungskriterien, wie auch passende Optimierungsalgorithmen vorgestellt. Desweitern werden einige konkrete Algorithmen im Detail betrachtet.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Angenommen wir befinden uns in einem Raum, in dem sich mehrere Personen befinden, die alle durcheinander bzw. miteinander reden, und eventuell noch im Hintergrund Musik läuft. Wenn wir nun automatisch, zum Beispiel mittels eines Spracherkenners, erkennen möchten, was die einzelnen Personen sagen, würden wir dafür im einfachsten Fall das Szenario mit Mikrophonen aufnehmen und durch den Spracherkenner schicken. Das Ergebnis wäre allerdings miserabel, denn die Aufnahmen der einzelnen Mikrophone sind ein Gemisch des gesamten akustischen Szenarios. Der Spracherkenner benötigt aber eine klare Aufnahme von nur einem einzelnen Sprecher. Um so eine Signal zu erhalten, müssen aus den vielen Aufnahmen die einzelnen Quellen herausgefiltert werden. Dieses Problem ist auch bekannt als *"Cocktail-Party Problem"*. Menschen fällt es im Vergleich zu einem Computer wesentlich einfacher, sich auf eine bestimme Quelle zu fokusieren und die Nebengeräusche und Störquellen auszublenden.

Ein Ansatz, dieses Problem technisch zu lösen, ist die *blind source separation* (BSS). Hierbei geht man davon aus, keine Information über die Position der Quellen bzw. der Mikrophone zu haben. Dies steht im Gegensatz zu *geometric source separation* (GSS) mittels *Beamforming* oder ähnlichen Verfahren, die gerade die Information über die Positionen der Mikrophone und Quellen ausnutzen. Desweiteren gibt es auch Ansätze, die beides versuchen, zu kombinieren [PA].

Hier wollen wir uns allerdings mit ersterem beschäftigen. Selbst von BSS gibt es etliche Ausprägungen, die einerseits von der Problemstellung abhängen, andererseits von dem Lösungsverfahren.

Die Problemstellung ist im einfachsten Fall eine simple gewichtete Summe der Quellen, wie es zum Beispiel bei einer einfachen Abmischung einer CD der Fall ist. Hier werden die einzelnen Spuren mit verschiedenen Pegeln auf die Ausgabespuren gemischt. Dies nennt man *instantaneous blind source separation*. Wenn wir allerdings eine Aufnahme eines Sinphonieorchesters mittels mehrerer Raummikrophone betrachten, so kommt zu dem auch noch die Raumimpulsantwort zwischen Quellen und Mikrophonen hinzu, die *konvolutiv* ist. Es kommen also zusätzlich zu den Lautstärkeunterschieden auch Zeitverzögerungen hinzu. Dies wird versucht mittels *convolutive blind source separation* zu lösen.

Die Lösungsverfahren kann man grob in solche unterteilen, die im Zeitbereich arbeiten und welche, die im Frequenzbereich arbeiten. Es geht jedoch im allgemeinen immer darum, eine *Kostenfunktion* zu minimieren. Die Kostenfunktion kann unterschiedlich aussehen, je nach dem auf welche Charakteristik der Quellensignale man

Wert legt. Da es sich bei den Verfahren um statistische Verfahren handelt, kann man die Kostenfunktionen in 2 Gruppen unterteilen, solche die auf *second-order statistics* und andere die auf *higher-order statistics* basieren. Bei second-order statistics wird darauf optimiert, dass die Ausgabesignale danach nicht mehr bzw. möglichst wenig korreliert sind, wohingegen bei higher-order statistics noch höhere statistische Momente mit einbezogen werden, um echte statistisch unabhängige Ausgabesignale zu erhalten. Die *independent component analysis* versucht genau dies, indem sie nach Signalen sucht, die möglichst nicht gaußisch sind. Kriterien dafür, wie nichtgaußisch ein Signal ist, ist z.B. die *Kurtosis* oder *Negentropy*. Eine weitere Möglichkeit ist es, die *mutual information* der Signale untereinander zu minimieren.

Im folgenden soll erst einmal in §2 auf die Problemstellung für den instantanen Fall eingegangen werden. Danach werden in §3 ein paar essentielle Vorverarbeitungschritte vorgestellt. In §4 gibt es dann einen Überblick über existierende Optimierungskriterien für BSS, gefolgt von Optimierungsalgorithmen in §5. Zum Abschluss werden dann einige konkrete Algorithmen für den instantanen Fall in §6 und den konvolutiven Fall in §7 näher betrachtet.

Kapitel 2

Problemstellung

Angenommen wir haben zwei Schallquellen s_1 und s_2, die mit zwei Mikrophonen x_1 und x_2 aufgenommen werden. Dann sind die beiden resultierenden Aufnahmen im vereinfachten Fall eine Mischung der Ursprungssignale mit jeweils anderen Lautstärkeverhältnissen, also eine gewichtete Summe, die durch folgende lineare Gleichung ausgedrückt werden kann:

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) \tag{2.1}$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t) \tag{2.2}$$

wobei a_{ij} die, von der Entfernung zwischen Quelle und Mikrophon abhängigen, Lautstärkeparameter sind und t der Zeitpunkt der Abtastung des Signals ist. Nun wollen wir anhand der zwei Aufnahmen die Ursprungssignale rekonstruieren, ohne die Lautstärkeparameter zu kennen. Dieses Problem ist auch bekannt unter dem Namen "Cocktail-Party Problem".

Auf eine beliebige Anzahl n von Signalen erweitert ergibt sich

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{2.3}$$

wobei $\mathbf{x}(t) = [x_1(t), \ldots, x_n(t)]^T$ der Vektor, bestehend aus den aufgenommenen Signalen $x_i(t)$, $\mathbf{s}(t) = [s_1(t), \ldots, s_n(t)]^T$ der Vektor, bestehend aus den Quellensignalen $s_i(t)$, und $\mathbf{A}$, mit den Elementen a_{ij}, die *Mischungsmatrix* ist. Gleichung (2.3) wird auch das *generative Modell* oder *Forward Modell* genannt.

Hätten wir die Parameter a_{ij} gegeben, könnten wir mittels klassischer Methoden die Lösung errechnen. Also geht es nun darum diese Parameter zu schätzen. Dies geschieht meistens direkt mit Hilfe des *Backward Modells*:

$$\tilde{\mathbf{s}}(t) = \mathbf{W}\mathbf{x}(t) \tag{2.4}$$

Dies erspart eine nachträgliche Invertierung von $\mathbf{A}$, um die Quellen zu trennen, und ermöglicht so die direkte Schätzung der Inversen $\mathbf{W}$.

Eine mögliche Lösung für dieses Problem bedient sich der statistischen Eigenschaften der Ursprungssignale $s_i(t)$. Im allgemeinen kann man davon ausgehen, dass die beiden Signale *statistisch unabhängig* sind. Dies ist eine realistische Annahme, wenn die Schallquellen unterschiedlichen Ursprung haben. Die *Independent Component Analysis* [HO00] bildet eine Klasse von Algorithmen, mittels denen $\mathbf{W}$ unter genau dieser Annahme, nämlich dass die Quellensignale statistisch unabhängig sind, geschätzt werden kann.

2.1 Probleme und Mehrdeutigkeiten

Allerdings gibt es bei dieser Schätzung zwei Mehrdeutigkeiten [HO00, §2.2], die nicht aufgelöst werden können.

2.1.1 Skalierungsproblem

Zum einen können die Varianzen (Energien) der Quellensignale nicht ermittelt werden. Dies liegt daran, dass wir weder $\mathbf{s}$ noch $\mathbf{A}$ kennen und jede Verstärkung einer der Quellen s_i kann durch eine entsprechende Abschwächung in der zugehörigen Spalte $\mathbf{a}_i$ in $\mathbf{A}$ wieder egalisiert werden. Diesem Problem begegnet man, indem man annimmt, dass die Komponenten Einheitsvarianzen haben:

$$E\{s_i^2\} = 1 \tag{2.5}$$

Allerdings bleibt dann das Vorzeichen der Komponenten immernoch ungewiss, was aber in der Regel kein großes Problem darstellt.

2.1.2 Permutationsproblem

Die zweite Mehrdeutigkeit liegt darin, dass die Reihenfolge der Komponenten nicht festgestellt werden kann. Auch dies liegt an der Unbekanntheit von $\mathbf{s}$ und $\mathbf{A}$, da eine Vertauschung der Komponenten in $\mathbf{s}$ durch eine Vertauschung der entsprechenden Spalten in $\mathbf{A}$ ausgeglichen werden kann.

Das Problem der Permutation führt vor allem dann zu Problemen, wenn man die Signale, statt im Zeitbereich, im Frequenzbereich verarbeitet. Dort fallen die Frequenzbänder unterschiedlichen Permutationen zum Opfer, wenn die Frequenzbänder getrennt von einander verarbeitet werden. Deshalb wird eine spezielle Nachverarbeitung notwendig, welche diese Permutation wieder korrigiert [PLKP07, §7].

Kapitel 3

Vorverarbeitung

Viele Algorithmen benötigen vor der eigentlichen Bearbeitung der Signale diverse Vorverarbeitungsschritte, um den eigentlichen Algorithmus möglichst einfach halten zu können. Diese können aber auch den Algorithmus an sich stabiler machen und helfen, schneller zum gewünschten Ergebnis zu kommen. Dies beruht auf der Tatsache, dass dadurch mit Bedingungen gearbeitet werden kann, die sonst nicht unbedingt gegeben sind.

3.1 Centering

Eine der meist verwendeten Vorverarbeitungen ist das *Centering* [HO00, §5.1] der Signale. Hier wird sichergestellt, dass die Daten für den weiteren Verlauf einen Mittelwert von Null haben, was es erspart, den Mittelwert später schätzen zu müssen. Dafür wird zuerst der Mittelwertsvektor $\mathbf{m} = E\{\mathbf{x}\}$ des Signals $\mathbf{x}$ geschätzt und dann von den Signalvektoren subtrahiert, um *mittelwertfreie* Signalvektoren zu erhalten.

Nach der eigentlichen Quellentrennung mit der Mischungsmatrix $\mathbf{A}$, kann dann der eigentliche Mittelwert der getrennten Signale $\mathbf{s}$ als $\mathbf{A}^{-1}\mathbf{m}$ wiederhergestellt werden und auf die mittelwertfreien Signalvektoren $\mathbf{s}$ addiert werden. Der Mittelwert der getrennten Signale wird also, so wie die getrennten Signale selbst, als die Transformation des ursprünglichen Mittelwerts mit derselben *Trennungsmatrix* $\mathbf{W} = \mathbf{A}^{-1}$ errechnet.

3.2 Whitening

Eine weitere Vereinfachung bringt das *Whitening* [Hyv99b, HO00, §5.2]. Dafür wird eine lineare Transformation

$$\tilde{\mathbf{x}} = \mathbf{Q}\mathbf{x} \tag{3.1}$$

gesucht, so dass die Kovarianzmatrix der transformierten Signalvektoren $\tilde{\mathbf{x}}$ der Identitätsmatrix entspricht und somit die einzelnen Komponenten unkorreliert und die Varianzen auf 1 normiert sind:

$$E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = \mathbf{I} \tag{3.2}$$

Die lineare Transformation kann mittels einer *Eigenwertzerlegung* der Kovarianzmatrix $E\{\mathbf{x}\mathbf{x}^T\}$ ermittelt werden. Dadurch erhält man eine Zerlegung der Kovarianzmatrix

$$E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^T \tag{3.3}$$

in eine orthogonale Matrix $\mathbf{E}$, deren Spalten den *Eigenvektoren* entsprechen, und eine diagonale Matrix $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$, deren Einträge die zugehörigen *Eigenwerte* sind. Die *Whiteningmatrix* ergibt sich dann als

$$\mathbf{Q} = \mathbf{\Lambda}^{-1/2}\mathbf{E}^T \tag{3.4}$$

wodurch wir das weiße Signal

$$\tilde{\mathbf{x}} = \mathbf{\Lambda}^{-1/2}\mathbf{E}^T\mathbf{x} \tag{3.5}$$

erhalten.

Hierdurch vereinfacht sich das Schätzen der Trennungsmatrix im folgenden, da nicht mehr ein beliebige Matrix mit n^2 freien Parametern geschätzt werden muss, sondern nur noch eine orthogonale Matrix $\mathbf{U}$, die $n(n-1)/2$ Freiheitsgrade besitzt.

3.2.1 Adaptives Whitening

Wenn der Algorithmus in Echtzeit laufen soll, kann nicht davon ausgegangen werden, dass alle Daten auf einmal zur Verfügung stehen. In diesem Fall muss die Whiteningmatrix über die Zeit hinweg adaptiv ermittelt werden. Dafür haben Cardoso und Laheld [CL96] einen *Least Mean Squares (LMS) Algorithmus* vorgestellt, der mit dem *relativen Gradienten* arbeitet:

$$\mathbf{Q}_t = \mathbf{Q}_{t-1} + \eta_t(\mathbf{I} - \tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^T)\mathbf{Q}_{t-1} \tag{3.6}$$

wobei η_t die Lernrate und $\tilde{\mathbf{x}}_t = \mathbf{Q}_{t-1}\mathbf{x}_t$ das, ähnlich wie in (3.1) definierte, transformierte Signal zum Zeitpunkt t ist.

3.3 Dimensionsreduktion

Oft ist es der Fall, dass man mehr Mixturen zur Verfügung hat als Signale, die man trennen möchte. Deshalb ist es sinnvoll, zuvor eine *Dimensionsreduktion* durchzuführen. Außerdem arbeiten die meisten Algorithmen unter der Annahme, dass die Anzahl M der Mixturen der Anzahl N der Signale entspricht.

Dies kann durch eine *Hauptkomponentenanalyse* (PCA) [Hyv99b,Wik07d] erreicht werden. Hierbei geht es darum eine lineare Transformation wie beim Whitening zu finden, so dass die transformierten Komponenten maximale Varianz haben. Da für das Whitening bereits eine Eigenwertzerlegung durchgeführt wurde, kann dies einfach integriert werden. Für eine PCA müssen nur noch die Eigenwerte absteigend sortiert werden, so dass der erste Eigenwert auf der Diagonale der Eigenwertmatrix den grössten Wert hat usw. Die Spalten der Eigenvektormatrix $\mathbf{E}$ werden dann entsprechend umsortiert. Für die Dimensionsreduktion können schließlich einfach alle Dimensionen mit zu kleinen Eigenwerten vernachlässigt werden. Als Resultat bekommen wir dann statt der $N \times N$-Eigenwertmatrix $\mathbf{\Lambda}$ eine $M \times M$-Eigenwertmatrix $\tilde{\mathbf{\Lambda}}$ und statt der $N \times N$-Eigenvektormatrix $\mathbf{E}$ eine $N \times M$-Eigenvektormatrix $\tilde{\mathbf{E}}$.

Die Entscheidung, ab welchem Eigenwert die Dimensionen vernachlässigt werden können, kann auf verschiedene Arten getroffen werden. Eine einfache Möglichkeit ist es, einen festen Schwellwert t für die Eigenwerte λ_i zu nehmen. Es werden dann alle Eigenwerte, für die $\lambda_i \geq t$ gilt, behalten. Für eine weitere Variante [Wik07d] wird

zuerst der *Gesamtenergieinhalt* $g(m)$ für jeden Eigenvektor $\mathbf{v}_m$ berechnet:

$$g(m) = \sum_{i=1}^{m} \lambda_i, \quad m = 1 \ldots N \tag{3.7}$$

Anschließend wird der kleinste Wert für $M = m$ gewählt, für den $g(m)/g(N) \geq t$, wobei t einen Prozentwert darstellt wie z.B. 90%.

Kapitel 4

Optimierungskriterien

Im folgenden werden nun einige Optimierungskriterien vorgestellt, die für BSS verwendet werden können. Diese sind in nach ihren statistischen Eigenschaften in Second- und Higher-Oorder Statistics unterteilt.

4.1 Second-Order Statistics

Kostenfunktionen, die auf *Second-Order Statistics (SOS)* (Momenten zweiten Grades) basieren, haben als Ziel die gegebenen Signale zu dekorrelieren. Sie beruhen auf der klassischen Annahme, dass die Quellensignale gauß'schen Charakter haben. In diesem Fall ist eine Dekorrelation ausreichend für eine Trennung der Signale. Handelt es sich allerdings um nichtnormalverteilte Signale, so reicht eine Dekorrelation allein nicht aus, um eine Trennung zu erreichen. Wenn man allerdings die Charakteristiken der Signale einschränkt, kann unter der Annahme, dass die unabhängigen Signale nichtstationär oder nichtweiß sind, auch eine Kostenfunktion die auf SOS basiert zu einer erfolgreichen Trennung führen [PS03, §2].

Diese Kriterien werden auch oft als Vorverarbeitungsschritt genutzt, um die Konvergenz des eigentlichen Algorithmus' zu verbessern.

Außerdem zeichnen sie sich dadurch aus, dass sie wenig Rechenleistung benötigen und sind somit auch ideal für den Einsatz in Echtzeitsystemen.

4.1.1 PCA-Kriterium / Dekorrelation

Das *PCA-Kriterium* ist dasselbe wie das Whitening §3.2. Dies allein ist allerdings nur für gauß'sche Signale genug, um eine Trennung durchzuführen.

4.1.2 Verallgemeinertes Eigenwert-Kriterium

Das *Verallgemeinerte Eigenwertproblem* ist eine Erweiterung des Eigenwertproblems, was der PCA zu Grunde liegt, und wird folgendermassen formuliert [Wik07a]:

$$\mathbf{R}\mathbf{v} = \lambda\mathbf{Q}\mathbf{v} \tag{4.1}$$

wobei λ der verallgemeinerte Eigenwert und $\mathbf{v}$ der verallgemeinerte Eigenvektor ist. Wenn man nun alle verallgemeinerten Eigenvektoren in der Eigenvektormatrix $\mathbf{W} = [\mathbf{v}_1 \cdots \mathbf{v}_n]$ und die verallgemeinerten Eigenwerte in der diagonalen Eigenwertmatrix

$\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$ zusammenfasst, ergibt sich folgende Matrixdarstellung [PS03, §1]:

$$\mathbf{RW} = \mathbf{QW\Lambda} \tag{4.2}$$

wobei $\mathbf{R}$ die Kovarianzmatrix und $\mathbf{Q}$ eine weitere Kreuzstatistik der Signalmixturen ist. Die Schätzung von $\mathbf{W}$ wird dann in der Regel durch eine *Simultane Diagonalisierung* erreicht.

Ist $\mathbf{Q} = \mathbf{I}$, so vereinfacht sich das Kriterium zu einem gewöhnlichen Eigenwertproblem und ist somit äquivalent zur PCA.

Nichtstationäre Quellen

Unter der Annahme, dass die Quellen nichtstationäre Energie haben, kann man $\mathbf{Q}$ wie folgt wählen [PS03, §2.1]:

$$\mathbf{Q}(t) = E\{\mathbf{x}(t)\mathbf{x}^H(t)\} \tag{4.3}$$

Dies ist allerdings nicht ausreichend bei Quellen, die nur im Frequenzbereich nichtstationär sind, aber insgesamt gesehen konstante Energie haben.

Ein auf Maximum-Likelihood und Transinformation basierendes Kriterium für nichtstationäre Quellen ist in [PC00] zu finden. Außerdem wird dort sowie in [Pha99] eine simultane approximierte Diagonalisierung mehrerer Kovarianzmatrizen, die über verschiedene stationäre Zeiträume geschätzt werden, vorgeschlagen, um die Trennungsmatrix zu schätzen.

Nichtweiße Quellen

Wenn es sich um nichtweiße Quellen handelt, deren Autokorrelation also nichtnull ist, so kann man $\mathbf{Q}$ wie folgt wählen [PS03, §2.2]:

$$\mathbf{Q}(\tau) = E\{\mathbf{x}(t)\mathbf{x}^H(t+\tau)\} \tag{4.4}$$

wobei τ eine Zeitverzögerung angibt, für die die Korrelation berechnet werden soll. Dies wurde auch in [MS94] unter dem Namen *Zeitverzögerte Dekorrelation (Time-delayed Decorrelation)* vorgeschlagen.

4.2 Higher-Order Statistics

Kostenfunktionen, die auf *Higher-Order Statistics (HOS)* (Momenten höheren Grades) basieren, versuchen mehr Informationen als nur den Mittelwert und die Kovarianzmatrix zu nutzen. Wenn die Quellensignale allerdings gauß'schen Charakter haben, so bieten HOS keine zusätzliche Information mehr.

4.2.1 Wölbung (Kurtosis)

Ein Kriterium, die Gaußförmigkeit eines Signals numerisch zu erfassen, ist die *Wölbung*, auch bekannt als *Kurtosis* [HO00, §4.2.1] [Wik07e]. Diese ist definiert als:

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2 \tag{4.5}$$

Wenn zuvor als Vorverarbeitung Centering und Whitening durchgeführt werden, das Signal also mittelwertfrei ist und eine Varianz von 1 hat, vereinfacht sich die Gleichung (4.5) zu:

$$kurt(y) = E\{y^4\} - 3 \tag{4.6}$$

Somit ist die Wölbung eine Normierung des vierten Momentes $E\{y^4\}$. Da für ein normalverteiltes Signal das vierte Moment $3(E\{y^2\})^2$ entspricht, ist die Kurtosis in diesem Falle 0. Für die meisten anderen Signale, wenn auch nicht für alle, ist sie ungleich 0. Der Ausnahmefall tritt aber nur äußerst selten ein.

Das Vorzeichen der Kurtosis gibt Auskunft über die Form der Wahrscheinlichkeitsverteilung. Ist der Wert positiv, so handelt es sich um eine *steilgipflige* oder *supergauß'sche* (leptocurtic) Verteilung, d.h. sie ist im Vergleich zur Normalverteilung um den Nullpunkt spitzer. Bei einer flacheren Verteilung ist die Kurtosis negativ und man spricht von *flachgipflig* oder *subgaußförmig* (platycurtic).

Da wir aber allgemein an nichtgauß'schen Signalen interessiert sind, genügt es den Betrag oder auch das Quadrat der Kurtosis zu betrachten.

Allerdings hat die Kurtosis in der Praxis einige Nachteile. So reagiert sie sehr empfindlich auf Ausreißer in der geschätzten Wahrscheinlichkeitsverteilung [Hub85], was sie zu einem schwachen Maß für Nichtgaußverteiltheit macht.

4.2.2 Negentropie

Eine weitere Möglichkeit, die Gaußförmigkeit zu messen, ist die *Negentropie* [HO00, §4.2.2]. Diese basiert auf dem informationstheoretischen Maß der *(differentiellen) Entropie*.

Die Entropie gibt eine Aussage darüber, wieviel Information in der Beobachtung einer Zufallsvariable steckt. Ihr Wert ist also umso größer, je "zufälliger" und unvorhersehbarer eine Variable ist.

Die Entropie H der diskreten Zufallsvariablen Y ist definiert als

$$H(Y) = -\sum_i P(Y = a_i) \log P(Y = a_i) \tag{4.7}$$

wobei die a_i alle möglichen Werte von Y repräsentieren. Im kontinuierlichen Fall ist die, dann sogenannte, differentielle Entropie [CT91,Pap91] H des Zufallsvektors $\mathbf{y}$ mit der Dichte $f(\mathbf{y})$ definiert als

$$H(\mathbf{y}) = -\int f(\mathbf{y}) \log f(\mathbf{y}) d\mathbf{y} \tag{4.8}$$

Für eine gauß'sche Variable nimmt die Entropy, unter allen Zufallsvariablen mit gleicher Varianz, den größten Wert an, da eine Gaußverteilung die geringste Struktur aufweist. Dadurch wird es möglich, die Entropie als Maß für die Gaußförmigkeit zu verwenden.

Die Negentropie ist ein Maß, das nur für eine gauß'sche Variable 0 und immer nichtnegativ ist. Sie leitet sich aus der Definition der differentiellen Entropie wie folgt her:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) \tag{4.9}$$

wobei $\mathbf{y}_{gauss}$ eine gauß'sche Variable mit der gleichen Kovarianzmatrix wie $\mathbf{y}$ ist.

Der Vorteil der Negentropie als Maß liegt in ihrem theoretischen Fundament in der Statistik. Betrachtet man statistische Eigenschaften, so ist die Negentropie ein optimaler Schätzer für Nichtgaußverteiltheit. Allerdings ist ihre Berechnung äußerst aufwendig.

Approximationen der Negentropie

Da in der Praxis die Berechnung der Negentropie sehr aufwendig ist, gibt es diverse Approximationen der Negentropie [HO00, §4.2.3].

Die klassische Methode, die Negentropie anzunähern, verwendet Momente höheren Grades [JS87]:

$$J(y) \approx \frac{1}{12} E\{y^3\}^2 + \frac{1}{48} kurt(y)^2 \tag{4.10}$$

Allerdings leidet diese Approximation unter denselben Schwächen wie schon die Kurtosis.

Hyvärinen [Hyv98] hat, basierend auf dem Prinzip der maximalen Entropie eine Annäherung vorgeschlagen, die diese Probleme umgeht:

$$J(y) \approx \sum_{i=1}^{p} k_i [E\{G_i(y)\} - E\{G_i(\nu)\}]^2 \tag{4.11}$$

Hierbei sind die k_i positive Konstanten und ν eine gauß'sche Variable, die als Referenz dient. y und ν werden jeweils als mittelwertsfrei mit Einheitsvarianz angenommen. Die G_i sind nichtquadratische Funktionen, die, wenn sie nicht zu schnell wachsen, zu einer robusteren Schätzung beitragen. Im einfachsten Fall, $p = 1$, spielt der konkrete Wert von k_1 keine Rolle und (4.11) vereinfacht sich zu:

$$J(y) \propto [E\{G(y)\} - E\{G(\nu)\}]^2 \tag{4.12}$$

Diese Approximation ist identisch mit (4.10), wenn y symmetrisch ist und $G(y) = y^4$ gewählt wird, und ist somit eine Generalisierung der momentbasierten Approximation.

Passende Funktionen für G sind beispielsweise:

$$G_1(u) = \frac{1}{a_1} \log \cosh a_1 u, \quad 1 \leq a_1 \leq 2 \tag{4.13}$$

$$G_2(u) = -\exp(-u^2/2) \tag{4.14}$$

wobei oft $a_1 = 1$ gewählt wird.

Diese Approximation bietet einen guten Kompromiss zwischen den Eigenschaften von Kurtosis und Negentropie.

4.2.3 Transinformation (Mutual Information)

Aus dem Bereich der Informationstheorie kommt auch das Maß der *Transinformation* [Hyv99b, §4.3.2] [HO00, §4.3]. Hierbei handelt es sich um ein Kriterium, was keine Aussage über ein einzelnes Signal macht, sondern über die Abhängigkeit zwischen mehreren Zufallsvariablen. Die Transinformation I läßt sich mittels der differentiellen Entropie für m (skalare) Zufallsvariablen y_i definieren als:

$$I(y_1, y_2, \ldots, y_m) = \sum_{i=1}^{m} H(y_i) - H(\mathbf{y}) \tag{4.15}$$

Auch diese Größe ist immer nichtnegativ und Null genau dann, wenn die Variablen statistisch unabhängig sind.

Zur Veranschaulichung der Transinformation kann man die Interpretation der Entropie als Codelänge heranziehen. In diesem Fall geben die $H(y_i)$ die Längen der Codes

für y_i an, wenn diese separat kodiert werden, und $H(\mathbf{y})$ gibt die Codelänge an, wenn alle Komponenten von $\mathbf{y}$ zusammen kodiert werden. Somit gibt die Transinformation die Codelängenreduktion an, wenn man den ganzen Vektor, statt seine Komponenten separat, kodiert. Wenn allerdings die Komponenten y_i statistisch unabhängig sind, geben sie keine Information über die jeweils anderen Komponenten und die Codes sind in beiden Fällen gleich lang, ob nun separat oder als Ganzes kodiert.

4.2.4 Likelihood

Eine sehr bekannte Methode der Parameterschätzung ist die *Maximum Likelihood Schätzung* [Hyv99b, §4.3.1] [HO00, §4.4.1]. Hierfür wird die *Likelihood* eines Parametersatzes für die beobachteten Daten maximiert. Die Likelihood $L(\mathbf{W}|\mathbf{X})$ des Parametersatzes $\mathbf{W}$, wenn die Daten $\mathbf{X}$ gegeben sind, ist proportional zur Wahrscheinlichkeit $P(\mathbf{X}|\mathbf{W})$ der Daten, wenn der Parametersatz gegeben ist. Um die spätere Maximierung einfacher zu machen, wird oft statt der Likelihood die *logarithmierte Likelihood* verwendet, da diese an derselben Stelle ein Maximum besitzt, aber einfacher abzuleiten ist. Diese Log-Likelihood ergibt sich im rauschfreien Fall zu [PGJ92]:

$$L = \sum_{t=1}^{T} \sum_{i=1}^{n} \log f_i(\mathbf{w}_i^T \mathbf{x}(t)) + T \log |\det \mathbf{W}| \tag{4.16}$$

wobei f_i die Dichtefunktionen der Quellensignale s_i sind. Diese Dichtefunktion wurde in [Mac96, §2.4] für Zeitbereichsignale als $f_i(s_i) = cosh^{-1}(s_i)$ und in [Ane01, §2.3] und [AG99, §1] für Frequenzbereichsignale als $f_i(S_i) \propto cosh^{-1}(||S_i||)$ bzw. in [SMAM02, §4] und [SMAM03, §2.3] als $f_i(S_i) = \alpha/cosh^2(S_i)$ gewählt.

4.2.5 Netzwerkentropie

Ein verwandtes Kriterium ist die *Netzwerkentropie* [Hyv99b, §4.3.1] [HO00, §4.4.2], deren Herleitung vom Gesichtspunkt eines Neuronalen Netzes erfolgte [NP94, BS95]. Hierfür wird ein Neuronales Netz angenommen, das aus dem Eingangssignal $\mathbf{x}$ Signale der Form $\phi_i(\mathbf{w}_i^T \mathbf{x})$ ausgibt, wobei ϕ_i nichtlineare Skalarfunktionen und $\mathbf{w}_i$ die Gewichtsvektoren der Neuronen sind. Ziel ist es dann die Entropie bzw. den *Informationsfluss* dieser nichtlinearen Ausgaben zu maximieren:

$$L = H(\phi_1(\mathbf{w}_1^T \mathbf{x}), \ldots, \phi_n(\mathbf{w}_n^T \mathbf{x})) \tag{4.17}$$

Diese Maximierung wird auch *Infomax* genannt und ist äquivalent [PP97, Car97] mit der Maximum Likelihood Schätzung (§4.2.4), wenn die Nichtlinearitäten ϕ_i als die kumulative Verteilungsfunktion gewählt werden, die den Dichten f_i aus (4.16) entsprechen, also $\phi_i'(\cdot) = f_i(\cdot)$.

4.2.6 Nichtlineares PCA Kriterium

Die *nichtlineare PCA* [Hyv99b, §4.3.4] ist eine Erweiterung der PCA. Allgemein wird die nichtlineare PCA durch folgendes zu minimierendes Kriterium [LZJ05,PK98,Oja97] definiert:

$$J(\mathbf{W}) = E\{||\mathbf{x} - \mathbf{W}^T \mathbf{g}(\mathbf{W}\mathbf{x})||^2\} \tag{4.18}$$

Hier wird eine nichtlineare Funktion g genutzt, um aus dem linearen Kriterium ein nichtlineares zu machen. Dieses g wird komponentenweise auf den Vektor $\mathbf{Wx}$ angewandt und ist normalerweise eine ungerade Funktion wie z.B. $g(t) = tanh(t)$ oder $g(t) = t^3$.

4.2.7 Verallgemeinertes Eigenwert-Kriterium

Das Verallgemeinerte Eigenwert-Kriterium in §4.1.2 ist nicht begrenzt auf Second-Order Statistics. Es kann ebenso auf Higher-Order Statistics angewandt werden.

Nichtgauß'sche Quellen

Für stationäre und weiße Quellen genügen die Informationen in den Kreuzstatistiken aus §4.1.2 nicht. Verwendet man *Kumulanten* vierten Grades, eine Verallgemeinerung der Kurtosis §4.2.1 für mehrere Signale, so ergibt sich die Kreuzstatistik als die *Kumulantenmatrix* [PS03, §2.3] [Car99, §3.2.1] für eine beliebige $n \times n$-Matrix $\mathbf{M}$ zu:

$$\mathbf{Q}(\mathbf{M}) = E\{\mathbf{x}^H\mathbf{M}\mathbf{x}\mathbf{x}\mathbf{x}^H\} - \mathbf{R}\text{tr}(\mathbf{MR}) - E\{\mathbf{x}\mathbf{x}^T\}\mathbf{M}^T E\{\mathbf{x}^*\mathbf{x}^H\} - \mathbf{RMR} \quad (4.19)$$

wobei $\mathbf{R} = E\{\mathbf{x}\mathbf{x}^H\}$ die Kovarianzmatrix von $\mathbf{x}$ und tr$(\mathbf{MR})$ die Spur von $\mathbf{MR}$ ist.

Im einfachsten Fall kann $\mathbf{M} = \mathbf{I}$ gewählt werden, allerdings ist dies sehr fehleranfällig [PS03, §2.3]. Robuster ist es n^2 verschiedene Kumulantenmatrizen simultan zu diagonalisieren [Car99, §3.2.1] [CS93], da, um die ganze Information vierten Grades zu erhalten, $O(n^4)$ Kumulanten vierten Grades berechnet werden müssen.

Um solch ein maximales Set an Kumulantenmatrizen $\{\mathbf{Q}(\mathbf{M}_i)|i = 1, \ldots, n^2\}$ zu erhalten, müssen die $\{\mathbf{M}_i|i = 1, \ldots, n^2\}$ eine beliebige Basis für den n^2-dimensionalen Linearraum von $n \times n$-Matrizen bilden [Car99, §4.2]. Die einfachste Wahl hierfür ist $\{\mathbf{e}_p\mathbf{e}_q^T|1 \leq p, q \leq n\}$, wobei $\mathbf{e}_p$ ein Spaltenvektor mit einer 1 an der p-ten Stelle und sonst nur Nullen ist. Besser noch ist eine symmetrische bzw. schiefsymmetrische Basis. So ergibt sich dann:

$$\mathbf{M}_{pq} = \begin{cases} \mathbf{e}_p\mathbf{e}_p^T, & \text{wenn } p = q \\ 2^{-1/2}(\mathbf{e}_p\mathbf{e}_q^T + \mathbf{e}_q\mathbf{e}_p^T), & \text{wenn } p < q \\ 2^{-1/2}(\mathbf{e}_p\mathbf{e}_q^T - \mathbf{e}_q\mathbf{e}_p^T), & \text{wenn } p > q \end{cases} \quad (4.20)$$

Dies ist eine Orthonormalbasis des $\mathbb{R}^{n\times n}$. Durch die Symmetrie ergibt sich außerdem eine Vereinfachung, wodurch nur $n + n(n-1)/2$ Kumulantenmatrizen berechnet werden müssen:

$$\mathbf{Q}(\mathbf{M}_{pq}) = \begin{cases} 2^{-1/2}\mathbf{Q}(\mathbf{e}_p\mathbf{e}_q^T), & \text{wenn } p < q \\ 0, & \text{wenn } p > q \end{cases} \quad (4.21)$$

4.2.8 Amplitudenmodulationskorrelation

Ein weiteres Bewertungskriterium ist die *Amplitudenmodulationskorrelation* [Ane99, AK00] [Ane01, §3.3]. Diese ist ein Kriterium, das die Eigenschaft natürlicher Schallquellen ausnutzt. Für diese gilt im allgemeinen, dass Amplitudenmodulationen in unterschiedlichen Frequenzbändern korreliert sind. Dies rührt beispielsweise bei Sprache daher, dass die Energiequelle für die Sprachproduktion die Stimmlippen sind. Diese emittieren ein breitbandiges Signal, was spektrale Spitzen in den Obertönen der Grundfrequenz des Sprechers aufweist. Jede Modulation der Stimmlippenbewegung

beeinflusst somit alle emittierten Frequenzen gleichzeitig. Auch die nachfolgende Filterung durch den Vokaltrakt hat Auswirkungen auf mehrere Frequenzen zur selben Zeit.

Dieses Kriterium arbeitet im Gegensatz zu den vorgenannten Kriterien im Frequenzbereich. Um den Grad der Verbundenheit der Amplitudenmodulation in zwei Frequenzbändern zweier Signale zu klassifizieren, wird die Korrelation der entsprechenden frequenzspezifischen Hüllkurven berechnet. Diese Korrelation kann wiederum als Korrelation der Zeitverläufe in den jeweiligen Frequenzbändern des Amplitudenspektrums berechnet werden. Es ergibt sich also für die Amplitudenmodulationskorrelation zwischen dem Frequenzband f_k des Spektrogramms $x(T, f)$ und dem Frequenzband f_l des Spektrogramms $y(T, f)$ folgende Formel:

$$c(x(T, f_k), y(T, f_l)) = E\{|x(T, f_k)||y(T, f_l)|\} - E\{|x(T, f_k)|\}E\{|y(T, f_l)|\} \tag{4.22}$$

Wenn man diese Funktion für alle möglichen Frequenzpaare (f_k, f_l) zweier Quellen s_i und s_j berechnet, erhält man die *AM Kreuzkorrelationsmatrix*:

$$[\mathbf{C}(s_i, s_j)]_{kl} = c(s_i(T, f_k), s_j(T, f_l)) \tag{4.23}$$

Die Einträge dieser Matrix sollten für unterschiedliche bzw. unabhängige Quellen alle gleich Null sein. Handelt es sich bei s_i und s_j um dieselbe Quelle, so spricht man von einer *AM Autokovarianzmatrix*. Diese Matrix hat in der Regel für alle Frequenzpaare von Null verschiedene Einträge.

Somit ergibt sich als zu minimierendes Optimierungskriterium für die Trennung der Quellen folgende Funktion der getrennten Signale u_i:

$$H = \sum_{i \neq j} \sum_{k,l} [\mathbf{C}(u_i, u_j)]_{kl}^2 \tag{4.24}$$

Da dieses Kriterium alle Frequenzbänder mit einbezieht, wird hier das Permutationsproblem der Frequenzbänder, was in §2.1.2 angesprochen wurde, implizit gelöst.

Kapitel 5

Optimierungsalgorithmen

Um nun mittels eines Optimierungskriteriums die Signale zu trennen, wird ein Optimierungsalgorithmus benötigt. Die Optimierungsalgorithmen lassen sich grob in zwei Klassen unterteilen. Auf der einen Seite gibt es die *Stapelverarbeitungsalgorithmen*, bei denen alle Daten zur selben Zeit dem Algorithmus zur Verfügung stehen und der Algorithmus einen konstanten Parametersatz für die gesamte Zeit schätzt. Auf der anderen Seite gibt es die *Adaptiven Algorithmen*, die in einem Zeitschritt nur mit den aktuellen Daten arbeiten und sich über die Zeit hinweg auch anpassen können. Letzteres ist vor allem für Echtzeitanwendungen nötig und bietet auch die Möglichkeit, sich über die Zeit hinweg verändernde, Parameter zu schätzen.

5.1 Stapelverarbeitungsalgorithmen

Bei den Stapelverarbeitungsalgorithmen handelt es sich in Regel um offline Algorithmen, also solche, die nicht echtzeitfähig sind, da sie die kompletten Daten auf einmal benötigen. In der Regel wird bei BSS hier auch nur eine globale, also für die gesamte Dauer der Aufnahme gültige, Trennungsmatrix geschätzt.

5.1.1 Gradient Descent

Beim *Gradient Descent* wird versucht, die Parameter $\mathbf{W}$ einer Kostenfunktion f zu finden, für die die Kostenfunktion ihr Minimum erreicht, indem man sich iterativ entlang des Gradienten an der aktuellen Position $\mathbf{W}_k$ dem globalen Minimum annähert. Dies lässt sich wie folgt ausdrücken:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \alpha(-\nabla f(\mathbf{W}_k)) \tag{5.1}$$

wobei die Schrittweite α angibt, wie groß die Schritte entlang des Gradienten sein sollen. Diese Schrittweite wird in der Regel über die Zeit dynamisch angepasst, um schneller zum Minimum zu gelangen und die Gefahr zu reduzieren, in einem lokalen Minimum hängen zu bleiben. Praktisch gesehen passiert es aber oft, dass der Algorithmus in einem lokalen Minimum hängen bleibt.

Natural Gradient

Eine kleine Modifikation des Gradienten, namens *Natural Gradient* [ACY96, §4], führt zu einer schnelleren Konvergenz. Hierfür wird der Gradient rechtsseitig mit $\mathbf{W}^T\mathbf{W}$

multipliziert. Da es sich bei dieser Matrix um eine positiv definite Matrix handelt, wird das Konvergenzkriterium dadurch nicht beeinflusst und es handelt sich somit um einen zulässigen *Pseudogradienten* [PP96, §B]. Außerdem vereinfacht sich dadurch oft auch die Gradientberechnung. In [CL96, §II-C] wurde dasselbe Prinzip unter dem Namen *Relative Gradient* eingeführt.

5.1.2 Newton Iteration

Die *Newton Iteration* [Wik07c, Wik07b] ist ein Näherungsverfahren zur numerischen Lösung nichtlinearer Gleichungen. Um dieses zum Suchen lokaler Extrema zu verwenden, kann man es auf den Gradienten der Bewertungsfunktion anwenden und dessen Nullstelle bestimmen. Die Iteration sieht dann wie folgt aus:

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \alpha[Hf(\mathbf{W}_k)]^{-1}\nabla f(\mathbf{W}_k) \tag{5.2}$$

wobei $Hf(\mathbf{W}_k)$ die Hessematrix der Bewertungsfunktion und α die Schrittweite ist. Damit dieses Verfahren funktioniert muss die Bewertungsfunktion f zweifach differenzierbar sein.

5.1.3 Simultane Diagonalisierung

Um einen etwas anderen Optimierungsalgorithmus handelt es sich bei der *simultanen Diagonalisierung* [CS93, §3.3] [BGBM93, CS96, BMCM97, Pha99]. Hierbei ist das Ziel eine Menge von K komplexwertigen $n \times n$ Matrizen $\mathcal{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_K\}$ (approximiert) zu diagonalisieren. Dies geschieht, indem man die Kostenfunktion [BMCM97]

$$C(\mathcal{M}, \mathbf{U}) = \sum_{k=1}^{K} \text{off}(\mathbf{U}^H \mathbf{M}_k \mathbf{U}) \tag{5.3}$$

für jedes mögliche Paar (i, j) mit $i \neq j$ minimiert, wobei $\mathbf{U}$ eine unitäre Matrix ist und $\text{off}(\mathbf{M})$ die Summe der quadrierten Matrixelemente M_{ij}, die nicht auf der Diagonalen liegen:

$$\text{off}(\mathbf{M}) = \sum_{1 \leq i \neq j \leq n} |M_{ij}|^2 \tag{5.4}$$

In [CS96] wurde die simultane Diagonalisierung als eine Erweiterung des *Jacobi-Verfahrens* vorgestellt, bei der die unitäre Matrix $\mathbf{U}$ als das Produkt mehrerer Givens-Drehungen gewählt wird, was den Algorithmus numerisch effizient macht. Bei den Givens-Drehungen handelt es sich um komplexe *Rotationsmatrizen* $\mathbf{R}(i, j, c, s)$, die bis auf die folgenden Einträge der Identitätsmatrix entsprechen:

$$\begin{pmatrix} r_{ii} & r_{ij} \\ r_{ji} & r_{jj} \end{pmatrix} = \begin{pmatrix} c & -\bar{s} \\ s & \bar{c} \end{pmatrix}, \quad c, s \in \mathbb{C}, \quad |c|^2 + |s|^2 = 1 \tag{5.5}$$

Die Werte für c und s, die (5.3) minimieren, lauten wie folgt:

$$c = \sqrt{\frac{x+r}{2r}}, \quad s = \frac{y - \mathrm{i}z}{\sqrt{2r(x+r)}}, \quad r = \sqrt{x^2 + y^2 + z^2} \tag{5.6}$$

wobei $[x, y, z]^T$ der Eigenvektor mit dem grössten Eigenwert folgender 3×3 reellwertigen symmetrischen Matrix $\mathbf{G}$ ist:

$$\mathbf{G} = \text{Re}\left(\sum_{k=1}^{K} h^H(\mathbf{M}_k) h(\mathbf{M}_k)\right) \tag{5.7}$$

mit

$$h(\mathbf{M}) = [M_{ii} - M_{jj}, M_{ij} + M_{ji}, \mathrm{i}(M_{ji} - M_{ij})] \tag{5.8}$$

Wenn es sich bei $\mathcal{M}$ um reelle symmetrische Matrizen handelt, dann sind auch die Rotationsparameter c und s reellwertig. Damit vereinfacht sich das Problem in sofern, dass die letzte Komponente des Vektors $h(\mathbf{M})$ Null ist und somit die Matrix $\mathbf{G}$ zu einer 2×2 Matrix reduziert werden kann, da die jeweils letzte Spalte und Zeile wegfällt.

In [CS93, §3.3] wird, statt die Summe der quadrierten Nichtdiagonalelemente zu minimieren, die Summe der quadrierten Diagonalelemente maximiert:

$$C(\mathcal{M}, \mathbf{U}) = \sum_{k=1}^{K} |\mathrm{diag}(\mathbf{U}^H \mathbf{M}_k \mathbf{U})|^2 \tag{5.9}$$

maximiert.

5.2 Adaptive Algorithmen

Bei den Adaptiven Algorithmen werden nur kurze Segmente für die Schätzung herangezogen. Dies macht es auch möglich, sich bewegende Quellen zu trennen.

5.2.1 Stochastic Gradient Descent

Der *Stochastic Gradient Descent* Algorithmus ist die adaptive Variante des Gradient Descent, wobei der Gradient $\nabla f(\mathbf{W}_k)$ durch seine instantane Approximation ersetzt wird.

Least Mean Square Estimation

Bei der *Least Mean Square Estimation* handelt es sich um einen Spezialfall des Stochastic Gradient Descent. Hier wird als Kostenfunktion der mittlere quadratische Fehler zwischen dem gewünschten und rekonstruierten Signal verwendet:

$$f(\mathbf{W}) = E\{|\mathbf{s}(t) - \mathbf{W}_k \mathbf{x}(t)|^2\} \tag{5.10}$$

5.2.2 Kalman Filter

Ein dynamisches System wird beim Standard *Kalman Filter* normalerweise durch die *Prozessgleichung*

$$\mathbf{x}_{t+1} = \mathbf{F}_{t+1,t} \mathbf{x}_t + \boldsymbol{\nu}_{1,t} \tag{5.11}$$

und die *Beobachtungsgleichung*

$$\mathbf{y}_t = \mathbf{C}_t \mathbf{x}_t + \boldsymbol{\nu}_{2,t} \tag{5.12}$$

definiert, wobei $\mathbf{x}_t$ und $\mathbf{y}_t$ der Zustands- und Beobachtungsvektor zur Zeit t darstellen. $\mathbf{F}_{t+1,t}$ ist die Zustandsübergangsmatrix, die eine Verbindung zwischen den Zeiten $t+1$ und t herstellt. $\mathbf{C}_t$ ist die Messmatrix, welche eine Verbindung zwischen der Beobachtung und dem Zustand herstellt. Die beiden Vektoren $\boldsymbol{\nu}_{1,t}$ und $\boldsymbol{\nu}_{2,t}$ repräsentieren das Prozess- und Messrauschen.

Um diese Modelle bzw. den Algorithmus für BSS nutzen zu können, wurde in [LZJ05] der Zustandsvektor $\mathbf{x}$ durch die Zustandsmatrix $\mathbf{W}^T$ ersetzt und der Algorithmus entsprechend angepasst. Als Optimierungskriterium wurde hier die nichtlineare PCA verwendet. Dies resultiert in der neuen Prozessgleichung

$$\mathbf{W}_{opt,t+1}^T = \mathbf{W}_{opt,t}^T \tag{5.13}$$

und der Beobachtungsgleichung

$$\mathbf{v}_t^T = \mathbf{g}^T(\mathbf{y}_t)\mathbf{W}_{opt,t}^T + \mathbf{e}_t^T \tag{5.14}$$

wobei $\mathbf{v}_t$ der bereits vorverarbeitete, weiße Beobachtungsvektor, $\mathbf{y}_t = \mathbf{W}_{opt,t}^T\mathbf{v}_t$ der Ausgabevektor und $\mathbf{e}_t$ der Fehlervektor ist. Hierbei wurde vereinfacht angenommen, dass die Zustandsübergangsmatrix $\mathbf{F}_{t+1,t}$ der Identitätsmatrix entspricht und das Prozessrauschen $\boldsymbol{\nu}_{1,t}$ gleich Null ist. Desweiteren wurde in der Beobachtungsgleichung aus den Spaltenvektoren $\mathbf{y}_t$ und $\boldsymbol{\nu}_{2,t}$ jeweils die Zeilenvektoren $\mathbf{v}_t^T$ und $\mathbf{e}_t^T$. Außerdem wurde die Meßmatrix $\mathbf{C}_t$ zu dem Zeilenvektor $\mathbf{g}^T(\mathbf{y}_t)$ vereinfacht.

Somit ergibt sich der Kalman Filter Algorithmus für die nichtlineare PCA zu:

$$\mathbf{z}_t = \mathbf{g}(\mathbf{W}_{t-1}^T\mathbf{v}_t) = \mathbf{g}(\mathbf{y}_t) \tag{5.15}$$

$$\mathbf{h}_t = \mathbf{K}_{t,t-1}\mathbf{z}_t \tag{5.16}$$

$$\mathbf{m}_t = \mathbf{h}_t/(\mathbf{z}_t^T\mathbf{h}_t + Q_t) \tag{5.17}$$

$$\mathbf{K}_{t+1,t} = \mathbf{K}_{t,t-1} - \mathbf{m}_t\mathbf{h}_t^T \tag{5.18}$$

$$\mathbf{W}_t^T = \mathbf{W}_{t-1}^T + \mathbf{m}_t(\mathbf{v}_t^T - \mathbf{z}_t^T\mathbf{W}_{t-1}^T) \tag{5.19}$$

Kapitel 6

Instantaneous Blind Source Separation

Die *Instantaneous Blind Source Separation* ist eine Klasse von Algorithmen, die sich damit befassen, instantane Mixturen zu trennen. Die Signale sind also einfach nur eine gewichtete Summe der Quellensignale.

6.1 FastICA

Die *FastICA* [Hyv99b, §5.9] [HO00, §6] [Hyv99a, HO97] ist eine Variante der Independent Component Analysis, die auf einer Fixpunkt-Iteration basiert. Sie kann auch als eine Approximation einer Newton-Iteration (§5.1.2) hergeleitet werden.

Zuerst betrachten wir den Algorithmus für eine einzige unabhängige Komponente. Dafür wird der Gewichtevektor $\mathbf{w}_0$ für die unabhängige Komponente zuerst (zufällig) initialisiert. Die Iteration läuft dann wie folgt ab:

$$\tilde{\mathbf{w}}_{k+1} = \mathbf{C}^{-1} E\{\mathbf{x} g(\mathbf{w}_k^T \mathbf{x})\} - E\{g'(\mathbf{w}_k^T \mathbf{x})\} \mathbf{w}_k \tag{6.1}$$

$$\mathbf{w}_{k+1} = \tilde{\mathbf{w}}_{k+1} / \sqrt{\tilde{\mathbf{w}}_{k+1}^T \mathbf{C} \tilde{\mathbf{w}}_{k+1}} \tag{6.2}$$

Diese Iteration wird so lange durchgeführt bis das Ergebnis konvergiert. Die Konvergenz ist dann erreicht, wenn der alte und der neue Wert von $\mathbf{w}$ in dieselbe Richtung zeigen. Dabei spielt das Vorzeichen keine Rolle, weil sowohl $\mathbf{w}$ als auch $-\mathbf{w}$ dieselbe Richtung definieren, da das Vorzeichen der Komponenten sowieso nicht ermittelt werden kann (§2.1.1).

Bei $\mathbf{C} = E\{\mathbf{x}\mathbf{x}^T\}$ handelt es sich um die Kovarianzmatrix der Daten. Dies erlaubt, es den Algorithmus auch auf Daten anzuwenden, ohne zuvor ein Whitening durchführen zu müssen. Ist $\mathbf{C}$ allerdings singulär oder annähernd singulär, so muss eine Dimensionsreduktion (§3.3) durchgeführt werden. Bei bereits weißen Daten vereinfacht sich der Algorithmus, da $\mathbf{C} = \mathbf{I}$ ist. In diesem Fall kann man die Vorkommnisse von $\mathbf{C}$ einfach weglassen.

Bei den Funktionen g und g' handelt es sich um die erste und zweite Ableitung einer nichtquadratischen Kontrastfunktion G (siehe §4). Im Falle der Approximation der Negentropy (§4.2.2), die in der Regel für die FastICA verwendet wird, ist die erste

Ableitung der in (4.13) und (4.14) definierten Funktionen:

$$g_1(u) = tanh(a_1 u), \quad 1 \leq a_1 \leq 2 \tag{6.3}$$
$$g_2(u) = u \exp(-u^2/2) \tag{6.4}$$

Die Normalisierung im letzten Schritt (6.2) dient dazu, den Algorithmus stabiler zu machen.

Wenn mehrere unabhängige Komponenten ermittelt werden sollen, wird der vorgenannte Algorithmus für jede der Komponenten durchgeführt, wodurch man die Gewichtevektoren $\mathbf{w}_1, \ldots, \mathbf{w}_n$ erhält. Um zu verhindern, dass mehrere dieser Vektoren zum selben Extremum hin konvergieren, müssen die Ausgänge $\mathbf{w}_1^T\mathbf{x}, \ldots, \mathbf{w}_n^T\mathbf{x}$ nach jeder Iteration dekorrelatiert werden.

Dies kann auf mehrere Arten erfolgen. Eine einfache Methode ist es, ein Deflationsschema zu verwenden. Dies basiert auf einer Gram-Schmidt-ähnlichen Dekorrelation. Hierfür wird eine unabhängige Komponente nach der anderen ermittelt. Angenommen es wurden bereits p Komponenten mit den Gewichtevektoren $\mathbf{w}_1, \ldots, \mathbf{w}_p$ ermittelt, dann wird der Algorithmus für eine Komponente für $\mathbf{w}_{p+1}$ durchgeführt. Nach jedem Iterationsschritt werden nun von $\mathbf{w}_{p+1}$ die "Projektionen" $\mathbf{w}_{p+1}^T\mathbf{w}_j\mathbf{w}_j, j = 1, \ldots, p$ der zuvor geschätzten p Vektoren subtrahiert und $\mathbf{w}_{p+1}$ erneut normalisiert:

$$\mathbf{w}_{p+1} = \mathbf{w}_{p+1} - \sum_{j=1}^{p} \mathbf{w}_{p+1}^T \mathbf{C} \mathbf{w}_j \mathbf{w}_j \tag{6.5}$$
$$\mathbf{w}_{p+1} = \mathbf{w}_{p+1} / \sqrt{\mathbf{w}_{p+1}^T \mathbf{C} \mathbf{w}_{p+1}} \tag{6.6}$$

Auch hier gilt, dass bei weißen Daten die Kovarianzmatrix $\mathbf{C}$ ignoriert werden kann.

Diese Methode priviligiert allerdings Vektoren gegenüber anderen. Wenn das unerwünscht ist, muss eine symmetrische Dekorrelation verwendet werden.

Dafür kann z.B. die klassische Methode [Hyv99a], die die Matrixquadratwurzel nutzt, angewandt werden:

$$\tilde{\mathbf{W}} = (\mathbf{W}\mathbf{C}\mathbf{W}^T)^{-1/2}\mathbf{W} \tag{6.7}$$

Hierbei setzt sich die Matrix $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_n)^T$ aus den einzelnen Vektoren in den jeweiligen Zeilen zusammen. Die inverse Quadratwurzel $(\mathbf{W}\mathbf{C}\mathbf{W}^T)^{-1/2}$ kann mittels einer Eigenwertzerlegung von $\mathbf{W}\mathbf{C}\mathbf{W}^T = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^T$ als $(\mathbf{W}\mathbf{C}\mathbf{W}^T)^{-1/2} = \mathbf{E}\mathbf{\Lambda}^{-1/2}\mathbf{E}^T$ erhalten werden.

Einfacher geht es mit folgendem iterativem Algorithmus [Hyv99a]:

$$\mathbf{W}_0 = \mathbf{W} / \sqrt{||\mathbf{W}\mathbf{C}\mathbf{W}^T||} \tag{6.8}$$
$$\mathbf{W}_{k+1} = \frac{3}{2}\mathbf{W}_k - \frac{1}{2}\mathbf{W}_k \mathbf{C} \mathbf{W}_k^T \mathbf{W}_k \tag{6.9}$$

Die letzte Gleichung (6.9) wird solange durchgeführt, bis das Ergebnis konvergiert. Für die Norm in (6.8) kann außer der Frobenius Norm fast jede Matrixnorm verwendet werden, z.B. die 2-Norm oder die grösste absolute Zeilen- (oder Spalten-) Summe.

Die Inverse $\mathbf{C}^{-1}$ in (6.1) muss nicht explizit berechnet werden, sondern kann, für jedes dekorrelierende $\mathbf{W}$, durch die Identität $\mathbf{C}^{-1} = \mathbf{W}^T\mathbf{W}$ ersetzt werden [Hyv99a]. Dies macht diesen Algorithmus zu einem Fixpunktalgorithmus für Maximum-Likelihood Schätzung.

6.1.1 Stabilisierte FastICA

Da die Konvergenz einer Newton-Iteration relativ unsicher ist, gibt es eine Erweiterung der FastICA [Hyv99a]. Dazu wird der Updateschritt (6.1) wie folgt ersetzt:

$$\tilde{\mathbf{w}}_{k+1} = \mathbf{w}_k - \mu_{k+1}[\mathbf{C}^{-1}E\{\mathbf{x}g(\mathbf{w}_k^T\mathbf{x})\} - \beta\mathbf{w}_k]/[E\{g'(\mathbf{w}_k^T\mathbf{x})\} - \beta] \tag{6.10}$$

wobei $\beta = E\{\mathbf{w}_k^T\mathbf{x}g(\mathbf{w}_k^T\mathbf{x})\}$ und μ_k ein Schrittweitenparameter ist. Die Schrittweite kann in jedem Iterationsschritt einen anderen Wert annehmen. So kann man mit $\mu_0 = 1$ anfangen, was der ursprünglichen FastICA (6.1) entspricht. Wenn die Konvergenz problematisch erscheint, kann die Schrittweite allmählich reduziert werden, bis der Algorithmus zufriedenstellend konvergiert. Damit wird der Algorithmus zu einer Mischung zwischen Newton-Iteration, wenn $\mu = 1$ ist, und Gradient Descent, wenn μ einen sehr kleinen Wert hat.

6.2 Equivariant adaptive source separation (EASI)

Bei der *Equivariant adaptive source separation (EASI)* [CL96, §IV.A] handelt es sich um eine Klasse von adaptiven Algorithmen, deren Struktur recht simpel gestaltet ist und deren Performanz nicht von der Mischmatrix abhängt, sondern allein von der (normalisierten) Verteilung der Quellensignale.

Diese Algorithmen basieren auf einer aus der Kurtosis und dem relativen Gradienten hergeleiteten und verallgemeinerten Funktion, die durch beliebige nichtlineare Funktionen $g_i(y_i)$ parametrisiert werden kann:

$$\mathbf{g}(\mathbf{y}) = [g_1(y_1), \ldots, g_n(y_n)]^T \tag{6.11}$$

$$H\mathbf{g}(\mathbf{y}) = \mathbf{y}\mathbf{y}^T - \mathbf{I} + \mathbf{g}(\mathbf{y})\mathbf{y}^T - \mathbf{y}\mathbf{g}(\mathbf{y})^T \tag{6.12}$$

Ist der Erwartungswert dieser Funktion gleich Null, so ist der stationäre Punkt des seriellen Updating Algorithmus' erreicht. Dieser stationäre Punkt ist dann unsere Trennungsmatrix, da dann gilt:

$$E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{I} \tag{6.13}$$

$$E\{\mathbf{g}(\mathbf{y})\mathbf{y}^T - \mathbf{y}\mathbf{g}(\mathbf{y})^T\} = 0 \tag{6.14}$$

Die Ausgabesignale sind also in diesem Fall dekorreliert (6.13) und paarweise unabhängig (6.14).

Der Algorithmus an sich lässt sich dann wie folgt schreiben:

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \lambda_t \left[\mathbf{y}_t\mathbf{y}_t^T - \mathbf{I} + \mathbf{g}(\mathbf{y}_t)\mathbf{y}_t^T - \mathbf{y}_t\mathbf{g}(\mathbf{y}_t)^T\right]\mathbf{W}_t \tag{6.15}$$

6.2.1 Normalisierte Form der EASI Algorithmen

Die *normalisierte Form der EASI Algorithmen* [CL96, §IV.B] bietet mehr Stabilität, wenn es z.B. darum geht, durch eine große Schrittweite eine schnelle Konvergenz zu erreichen, was normalerweise zu einer erhöhten Sensitivität gegenüber Ausreißern führt. Da aber bei den EASI Algorithmen die Trennungsmatrizen beliebige Werte annehmen dürfen, sollten keine Einschränkungen dieser Matrix gemacht werden. Stattdessen wird die Stabilisierung anhand einer Modifikation der Funktion $H\mathbf{g}$ erreicht:

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \lambda_t \left[\frac{\mathbf{y}_t\mathbf{y}_t^T - \mathbf{I}}{1 + \lambda_t\mathbf{y}_t^T\mathbf{y}_t} + \frac{\mathbf{g}(\mathbf{y}_t)\mathbf{y}_t^T - \mathbf{y}_t\mathbf{g}(\mathbf{y}_t)^T}{1 + \lambda_t|\mathbf{y}_t^T\mathbf{g}(\mathbf{y}_t)|}\right]\mathbf{W}_t \tag{6.16}$$

6.3 Joint Approximate Diagonalization of Eigen-matrices (JADE)

Ein etwas anderer Algorithmus für die Quellentrennung ist der *JADE Algorithmus* [CS93, §4.1]. Dieser baut auf dem Konzept der simultanen Diagonalisierung §5.1.3 [Car99, §4.2] auf. Hier wird die Trennungsmatrix als die Rotationsmatrix geschätzt, welche die Kumulantenmatrizen der weißen Daten simultan diagonalisiert.

Als Vorverarbeitung wird für JADE zuerst ein Centering §3.1 und Whitening §3.2 der Daten durchgeführt. Danach werden die Kumulantenmatrizen berechnet §4.1.2. Für diese Matrizen wird dann mittels Simultaner Diagonalisierung §5.1.3 eine Rotationsmatrix geschätzt, durch die die vorverarbeiteten Daten dann getrennt werden können.

Kapitel 7

Convolutive Blind Source Separation

Die *Convolutive Blind Source Separation* befasst sich, im Gegensatz zur instantanen Variante, mit konvolutiven Signalmixturen. Da dabei die Faltung eine große Rolle spielt, existieren neben Algorithmen, die im Zeitbereich arbeiten, hauptsächlich solche, die im Frequenzbereich arbeiten, da dort die Faltung effizienter berechnet werden kann.

7.1 Zeitbereich

Im Zeitbereich setzt sich das i-te Signal zum Zeitpunkt t wie folgt zusammen:

$$x_i(t) = \sum_{j=1}^{N} \sum_{t'} a_{ij}(t') s_j(t - t') \tag{7.1}$$

Algorithmen im Zeitbereich wurden z.B. von Pan et al. [PA07] oder Joho [Joh] vorgeschlagen.

7.2 Frequenzbereich

Im Frequenzbereich vereinfacht sich das Problem in sofern, dass man es in mehrere instantane Probleme aufspalten kann:

$$\mathbf{x}(T, f) = \mathbf{A}(f)\mathbf{s}(T, f) \tag{7.2}$$

wobei $\mathbf{x}(T, f)$, $\mathbf{A}(f)$ und $\mathbf{s}(T, f)$ jeweils die Frequenzbereichsdarstellungen von $\mathbf{x}$, $\mathbf{A}$ und $\mathbf{s}$ für die Frequenz f zum Zeitpunkt T sind. Allerdings muss in diesem Fall dem Permutationsproblem §2.1.2 Sorge getragen werden.

7.2.1 Amplitudenmodulationsdekorrelations Algorithmus

Ein BSS Algorithmus der in Experimenten sehr gute Ergebnisse auf realen Daten liefert, ist der *Amplitudenmodulationsdekorrelations Algorithmus* [Ane99,AK00] [Ane01,

§3.4.4] von Jörn Anemüller. Dieser verwendet das gleichnamige Amplitudenmodulationskorrelations Kriterium §4.2.8. Für das Minimieren des Kriteriums wird ein gradientenbasierter Optimierungsalgorithmus, wie der Gradient Descent Algorithmus aus §5.1.1, verwendet.

Es wurde anhand von Experimenten festgestellt, dass die Minimierung der Optimierungsfunktion H für alle Frequenzen gleichzeitig oft in einem lokalen Minimum endet, was eine schlechte Quellentrennung als Ergebnis hat. Dies scheint teilweise daran zu liegen, dass diese lokalen Minima Lösungen mit lokalen Permutationen in den Frequenzbändern sind, die zwar lokal optimal sind, aber eben nicht global.

Um die Gefahr, in einem lokalen Minimum hängen zu bleiben, zu verringern, wird stattdessen die Optimierungsfunktion sequenziell für eine Frequenz nach der anderen minimiert. Dabei müssen Beschränkungen auf die Trennungsmatrizen $\mathbf{W}(f)$ angewendet werden, damit diese nicht gegen Null konvergieren. Eine Möglichkeit ist es die Diagonalelemente der Matrizen auf Eins und die des Gradienten entsprechend auf Null festzusetzen. Eine andere Lösung ist es, die Zeilen der Trennungsmatrizen auf Einheitsnorm zu normalisieren und die Imaginärteile der Diagonalelemente auf Null.

Angefangen wird mit dem Frequenzband f_{start}, das die höchste Signalenergie aufweist. Nur für diese Frequenz wird die Trennungsmatrix $\mathbf{W}(f_{curr})$ optimiert, während die Trennungsmatrizen der anderen Frequenzen festgehalten werden. Danach wird dasselbe iterativ für die nächst höheren Frequenzen gemacht. Wenn die höchste Frequenz erreicht ist, wird von der Anfangsfrequenz f_{start} iterativ für die nächst niedrigeren Frequenzen die Optimierung durchgeführt. Diese Prozedur wird insgesamt dreimal abgearbeitet.

Um die Trennung robuster zu machen, hat Anemüller außerdem eine Variante des Whitenings vorgeschlagen [Ane01, §A.1], bei der das Whitening für jedes Frequenzband separat, ähnlich wie in §3.2, durchgeführt wird. Allerdings wird nicht die Identitätsmatrix für die Kovarianzmatrizen der Frequenzbänder angestrebt, sondern eine Diagonalmatrix, deren Elemente sich aus der Signalenergie des entsprechenden Frequenzbandes ergeben. Dies ist notwendig, da bei dem Standard Whitening alle Frequenzen gleich gewichtet werden und somit höhere Frequenzen, zumindest bei Sprachsignalen, eine Verstärkung erfahren würden.

Literaturverzeichnis

[ACY96] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 757–763. The MIT Press, 1996.

[AG99] Jörn Anemüller and Tino Gramss. On-line blind separation of moving sound sources. In *ICA 99: First international workshop on independent component analysis and signal separation*, pages 331–334, Aussois, France, January 11–15 1999.

[AK00] J. Anemüller and B. Kollmeier. Amplitude modulation decorrelation for convolutive blind source separation. In *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, pages 215–220, Helsinki, Finland, 2000.

[Ane99] Jörn Anemüller. Correlated modulation: A criterion for blind source separation. In *Joint meeting of the Acoustical Society of America and the European Acoustics Association*, Berlin, Germany, 1999.

[Ane01] Jörn Anemüller. *Across-Frequency Processing in Convolutive Blind Source Separation*. PhD thesis, University of Oldenburg, Oldenburg, Germany, 2001.

[BGBM93] Angelika Bunse-Gerstner, Ralph Byers, and Volker Mehrmann. Numerical methods for simultaneous diagonalization. *SIAM Journal of Matrix Analysis and Applications*, 14(4):927–949, 1993.

[BMCM97] A. Belouchrani, K. Abed Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique based on second order statistics. *IEEE Trans. on Signal Processing*, 45(2):434–444, 1997.

[BS95] A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[Car97] Jean-François Cardoso. Infomax and maximum likelihood for source separation. *IEEE Letters on Signal Processing*, 4(4):112–114, April 1997.

[Car99] Jean-François Cardoso. High-order contrasts for independent component analysis. *Neural Computation*, 11(1):157–192, jan 1999.

[CL96] J. Cardoso and B. Laheld. Equivariant adaptive source separation, 1996.

[CS93] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. *IEE Proceedings-F*, 140(6):362–370, 1993.

[CS96] Jean-Francois Cardoso and Antoine Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM J. Matrix Anal. Appl.*, 17(1):161–164, 1996.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

[HO97] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[HO00] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Netw.*, 13(4-5):411–430, 2000.

[Hub85] Peter J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[Hyv98] Aapo Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 273–279, Cambridge, MA, USA, 1998. MIT Press.

[Hyv99a] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Networks*, 10(3):626–634, 1999.

[Hyv99b] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[Joh] Marcel Joho. Blind signal separation of convolutive mixtures: A time-domain joint-diagonalization approach.

[JS87] M.C. Jones and R. Sibson. What is projection pursuit? *A Journal of the Royal Statistical Society*, 150:1–36, 1987.

[LZJ05] Qi Lv, Xian-Da Zhang, and Yng Jia. Kalman filtering algorithm for blind source separation. In *Proc. ICASSP*, volume 5, pages 257–260, 2005.

[Mac96] D. J. C. MacKay. Maximum likelihood and covariant algorithms for independent component analysis. Technical report, Dept. of Physics, Cambridge University, England, 1996. http://www.inference.phy.cam.ac.uk/mackay/abstracts/ica.html.

[MS94] L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72(23):3634–3637, June 1994.

[NP94] J.-P. Nadal and N. Parga. Non-linear neurons in the low noise limit: a factorial code maximizes information transfer. *Network*, 5:565–581, 1994.

[Oja97] E. Oja. The nonlinear PCA learning rule in independent component analysis. *Neurocomputing*, 17(1):25–46, 1997.

[PA] L. Parra and C. Alvino. Geometric source separation: merging convolutive source separation with geometric beamforming.

[PA07] Qiongfeng Pan and Tyseer Aboulnasr. Time-domain convolutive blind source separation employing selective-tap adaptive algorithms. *EURASIP Journal on Audio, Speech, and Music Processing*, 2007:Article ID 92528, 11 pages, 2007. doi:10.1155/2007/92528.

[Pap91] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.

[PC00] D.-T. Pham and J.-F. Cardoso. Blind separation of instantaneous mixtures of non-stationary sources. In *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, pages 187–193, Helsinki, Finland, 2000.

[PGJ92] D.-T. Pham, P. Garrat, and C. Jutten. Separation of a mixture of independent sources through a maximum likelihood approach. In *Proc. EUSIPCO*, pages 771–774, 1992.

[Pha99] Dinh Tuan Pham. Joint approximate diagonalization of positive definite hermitian matrices. Technical report, Laboratory LMC/IMAG, University of Grenoble, France, April 1999. http://www-lmc.imag.fr/lmc-sms/Dinh-Tuan.Pham/jadiag/jadiag.ps.gz.

[PK98] P. Pajunen and J. Karhunen. Least-squares methods for blind source separation based on nonlinear PCA. *Int. J. of Neural Systems*, 8(5-6):601–612, 1998.

[PLKP07] M. S. Pedersen, J. Larsen, U. Kjems, and L. C. Parra. A survey of convolutive blind source separation methods. In *Springer Handbook of Speech (to appear)*. Springer Press, sep 2007.

[PP96] Barak A. Pearlmutter and Lucas C. Parra. A context-sensitive generalization of ica. In *Proc. ICONIP*, Hong Kong, September 1996.

[PP97] Barak A. Pearlmutter and Lucas C. Parra. Maximum likelihood blind source separation: A context-sensitive generalization of ICA. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 613. The MIT Press, 1997.

[PS03] Lucas Parra and Paul Sajda. Blind source separation via generalized eigenvalue decomposition. *J. Mach. Learn. Res.*, 4:1261–1269, 2003.

[SMAM02] Hiroshi Sawada, Ryo Mukai, Shoko Araki, and Shoji Makino. Polar coordinate based nonlinear function for frequency-domain blind source separation. In *Proc. ICASSP*, volume 1, pages 1001–1004, Orlando, FL, USA, 2002.

[SMAM03] Hiroshi Sawada, Ryo Mukai, Shoko Araki, and Shoji Makino. Polar coordinate based nonlinear function for frequency-domain blind source separation. *IEICE Trans. Fundamentals*, E86-A(3):590–596, March 2003.

[Wik07a] Wikipedia. Eigenvalue, eigenvector and eigenspace — wikipedia, the free encyclopedia, 2007. http://en.wikipedia.org/w/index.php?title=Eigenvalue%2C_eigenvector_and_eigenspace&oldid=132721482.

[Wik07b] Wikipedia. Newton-verfahren — wikipedia, die freie enzyklopädie, 2007. http://de.wikipedia.org/w/index.php?title=Newton-Verfahren&oldid=31150085.

[Wik07c] Wikipedia. Newton's method in optimization — wikipedia, the free encyclopedia, 2007. http://en.wikipedia.org/w/index.php?title=Newton%27s_method_in_optimization&oldid=113595857.

[Wik07d] Wikipedia. Principal components analysis — wikipedia, the free encyclopedia, 2007. http://en.wikipedia.org/w/index.php?title=Principal_components_analysis&oldid=106444434.

[Wik07e] Wikipedia. Wölbung (statistik) — wikipedia, die freie enzyklopädie, 2007. http://de.wikipedia.org/w/index.php?title=W%C3%B6lbung_%28Statistik%29&oldid=26043847.