Newton update based independent vector analysis with various source density models

Master's thesis of statistics

May 27, 2022
Mika Sipilä
Department of Mathematics and Statistics
University of Jyväskylä

UNIVERSITY OF JYVÄSKYLÄ
Department of Mathematics and Statistics
Mika Sipilä: *Newton update based independent vector analysis with various source density models*
Master's thesis of statistics, 48 pages + 21 pages (Appendix)
May 27, 2022

Abstract
Blind source separation methods (BSS) are used to estimate latent source signals from their mixed observations when the mixing environment is unknown. Independent component analysis (ICA) is a BSS method, which aims to recover the sources by maximizing the independence between the estimated sources. A more recently developed method, independent vector analysis (IVA), is an extension of ICA to analyse multivariate source signals or multiple datasets jointly. IVA assumes that the source components are dependent on each other between the datasets, which is used to achieve better results than by applying ICA to each dataset separately. IVA uses the Kullback-Leibler divergence as an objective function, which is minimized to achieve as independent source estimates as possible.

To minimize the objective function, the source density models and the optimization method need to be selected. In this thesis, four different algorithms are investigated, each of which is using a Newton update based optimization method. The source density models of the algorithms are the multivariate Gaussian (IVA-G), the multivariate Laplace with any covariance structure (IVA-L), the multivariate Laplace with diagonal covariance structure (IVA-L-diag) and the multivariate Cauchy (IVA-C) distributions.

The algorithms are compared under different situations using simulation studies. IVA-L, IVA-L-diag and IVA-C tend to converge often to local optima, which is avoided by initializing IVA-L, IVA-L-diag and IVA-C with the estimated unmixing matrices of IVA-G and fastIVA. FastIVA is the original IVA algorithm, which restricts the unmixing matrices to be orthogonal. After the initialization, IVA-L becomes the most flexible and consistent algorithm in all setups. IVA-G performs well when the sources are mostly second-order dependent, and is superior in terms of computation time. IVA-L-diag and IVA-C improve the results of fastIVA only marginally, and perform well when the sources are purely higher-order dependent and the number of datasets is significantly higher than the number of sources.

The algorithms are applied to mixed image separation task, where five random mixtures of five colored images are separated. In this application IVA-L and IVA-G algorithms provide sufficient results, but the separated images of IVA-L-diag and IVA-C are not recognizable. The IVA algorithms and their performance indices are implemented in R package *ivaBSS* as a part of the thesis.

Keywords: Independent vector analysis, independent component analysis, blind source separation, Newton update

JYVÄSKYLÄN YLIOPISTO
Matematiikan ja tilastotieteen laitos
Mika Sipilä: *Newton update based independent vector analysis with various source density models*
Tilastotieteen pro gradu -tutkielma, 48 sivua + 21 sivua (Liite)
Toukokuu 27, 2022

Tiivistelmä

Sokea signaalin käsittely tarkoittaa latenttien lähdesignaalien estimointia havaittujen sekoitesignaalien avulla, kun sekoitusympäristö on tuntematon. Riippumattomien komponenttien analyysi (ICA) on sokean signaalin käsittelyn menetelmä, jolla pyritään estimoimaan todellisia lähdesignaaleja maksimoimalla niiden välinen riippumattomuus. Riippumattomien vektoreiden analyysi (IVA) on ICA:n laajennos, jolla estimoidaan moniulotteisia lähdesignaalivektoreita olettaen, että jokaisen lähdesignaalivektorin komponentit ovat riippuvia toisistaan.

IVA:n tavoitefunktiona käytetään Kullback-Leibler divergenssiä, jota minimoimalla lähdesignaaliestimaattien välinen riippumattomuus maksimoidaan. Minimointia varten täytyy valita optimointimenetelmä sekä lähdesignaaleille sopiva lähdejakaumamalli, jotka määrittävät yhdessä IVA algoritmin suorituskyvyn. Tässä tutkielmassa tarkastellaan neljää algoritmia, joista jokainen perustuu Newtonin menetelmään. Algoritmien lähdejakaumamallit ovat moniulotteinen normaalijakauma (IVA-G), moniulotteinen Laplace-jakauma (IVA-L), moniulotteinen Laplace-jakauma diagonaalisella kovarianssirakenteella (IVA-L-diag) ja moniulotteinen Cauchy-jakauma (IVA-C).

Algoritmeja vertaillaan simulointien avulla useissa eri simulaatioasetelmissa. IVA-L, IVA-L-diag ja IVA-C konvergoivat usein lokaaliin minimiin, mikä ratkaistaan alustamalla IVA-L, IVA-L-diag ja IVA-C algoritmit IVA-G:n ja fastIVA:n tuloksilla. FastIVA on alkuperäinen, ortogonaalisiin palautusmatriiseihin rajoittunut IVA-algoritmi. Alustuksen jälkeen IVA-L on tulosten perusteella paras ja monikäyttöisin algoritmi kaikissa tilanteissa. IVA-G on ylivoimaisesti nopein algoritmi, ja suoriutuu hyvin, kun lähdesignaalit ovat riippuvia enimmäkseen toisen asteen momentista. IVA-L-diag ja IVA-C algoritmit parantavat fastIVA:n tuloksia vain marginaalisesti, mutta ovat varteenotettavia vaihtoehtoja, kun lähdesignaalit ovat riippuvia ainoastaan korkeamman asteen momentista.

IVA algoritmeja sovelletaan sekoitettujen kuvien erotteluun, jossa viisi alkuperäistä värillistä kuvaa pyritään erottelemaan niiden viidestä satunnaista sekoitteesta. Tässä sovelluksessa IVA-L ja IVA-G algoritmit tuottivat kelvollisia tuloksia, mutta IVA-L-diag ja IVA-C algoritmien tulokset eivät olleet tunnistettavissa. Tutkielmassa käytetyt IVA algoritmit sekä niiden suorituskykyyn liittyvät indeksit ovat julkaistu R-paketissa *ivaBSS* osana tutkielmaa.

Avainsanat: Riippumattomien vektorien analyysi, riippumattomien komponenttien analyysi, sokea signaalin käsittely, Newtonin päivitys

Contents

1 Introduction

Blind source separation (BSS) has interested many researchers in multiple fields in past few decades (Comon and Jutten, 2010). BSS is used to estimate latent source signals from their mixed observations, where the sources and the mixing environment are unknown (Hyvärinen et al., 2001). The objective is to recover the original signals using only the mixtures of the signals and some reasonable assumptions. A large amount of different algorithms have been proposed to solve the BSS problem. There are many fields such as biomedical signal processing, image processing and audio processing, where BSS is an useful tool (Kim et al., 2006c).

One well-known example in the field of blind source separation is a cocktail party problem (Haykin and Chen, 2005), which is a problem of separating individual acoustic signals from conversations recorded with multiple microphones (Comon and Jutten, 2010). Consider a meeting room where two people are talking simultaneously and two microphones are recording the speeches on opposite sides of the room as illustrated in Figure 1. Each microphone captures the combination of the speeches differently. Speech coming from near the microphone is recorded more strongly than the speech coming from the other side of the table. This results in two different mixtures of two original independent speech signals. The recorded signals are denoted as $x_1(t)$ and $x_2(t)$ and the original speech signals as $s_1(t)$ and $s_2(t)$, where t is time index. The recorded signals can be expressed as linear mixtures

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t),$$
$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t),$$

where parameters a_{11}, a_{12}, a_{21} and a_{22} are unknown weights that depend on the distances between the speakers and the microphones. The problem is, how to recover the original speech signals $s_1(t)$ and $s_2(t)$ using only the recorded mixtures $x_1(t)$ and $x_2(t)$. One commonly used BSS method is independent component analysis (ICA, Comon, 1994). ICA aims to estimate latent independent components from the observed mixed signals by maximizing independence between the estimated components. ICA is a commonly used approach for solving the cocktail party problem. In previously introduced case, ICA uses the recorded speech mixtures to estimate the original speech signals, i.e. independent components. Figure 2 shows the examples of the source signals, their observed mixtures and the signals estimated by a Newton update based fast fixed-point IVA (fastIVA) algorithm with Laplace source

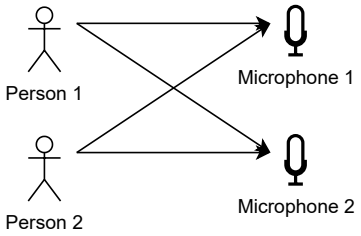

Figure 1: An illustration of the cocktail party problem. Two people are talking simultaniously and two microphones record mixtures of the speeches.

density model (Lee et al., 2007). Even though fastIVA algorithm is built for the IVA, it solves the ICA problem, if there is only one dataset. Source density models and IVA algorithm with Newton update are discussed in Section 3.

Now, let us assume the same situation except that both recording spots have two different microphones stacked on top of each other. One records only low frequency sounds and the other records only high frequency sounds. This results in two sets of recorded mixtures, one for low frequency sounds and one for high frequency sounds. Each speech's low frequency signal and high frequency signal can be assumed to be dependent as they are from the same source. Now the recorded mixtures x_1 and x_2, are

$$
\begin{aligned}
x_1^{[1]}(t) &= a_{11}^{[1]} s_1(t)^{[1]} + a_{12}^{[1]} s_2(t)^{[1]}, \\
x_2^{[1]}(t) &= a_{21}^{[1]} s_1(t)^{[1]} + a_{22}^{[1]} s_2(t)^{[1]}, \\
x_1^{[2]}(t) &= a_{11}^{[2]} s_1(t)^{[2]} + a_{12}^{[2]} s_2(t)^{[2]}, \\
x_2^{[2]}(t) &= a_{21}^{[2]} s_1(t)^{[2]} + a_{22}^{[2]} s_2(t)^{[2]},
\end{aligned}
$$

where $\cdot^{[1]}$ refers to first dataset, i.e. low frequency signals and $\cdot^{[2]}$ refers to second data sets, i.e. high frequency signals. The mixing weights $a_{ij}^{[d]}$ can be different between the datasets. The mixtures $\mathbf{x}_i(t) = (x_i^{[1]}(t), x_i^{[2]}(t))^\top$ as well as original independent components $\mathbf{s}_i(t) = (s_i^{[1]}(t), s_i^{[2]}(t))^\top$ are now vectors. How would the original signals be estimated in this scenario?

One can still apply ICA to these datasets separately and obtain estimates for the original signals. However, only a part of the available information will be used as the dependence of the datasets will be neglected, and thus the ICA solution might not be optimal (Kim et al., 2006c). The other problem is that the estimated signals can be in different order for each dataset (Hyvärinen and Oja, 2000), which means that the ICA approach does not solve which low frequency estimate and high frequency estimate belong together. For situations where a large

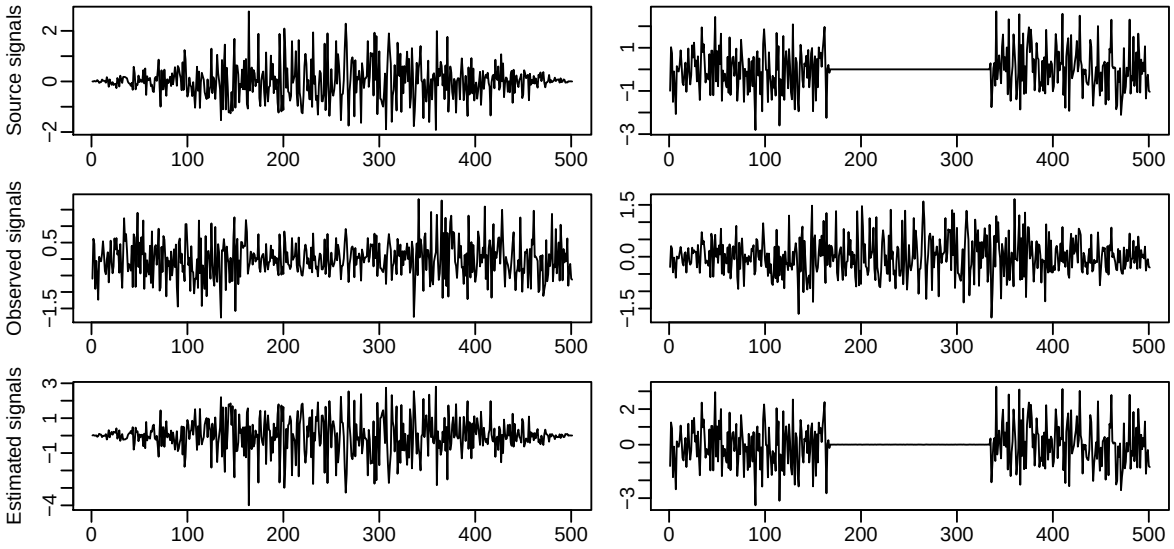

Figure 2: Example of original source signals, observed mixtures and signals estimated by fast fixed-point IVA algorithm with the multivariate diagonal Laplace source density model for the original sources. The algorithm solves ICA problem, if there is only one dataset. The source signals are artificially generated examples, not real acoustic signals.

number of datasets are used, the fact that the estimated signals may be ordered differently for each dataset, causes a challenging clustering problem of solving which estimated components belong together (Kim et al., 2006c).

Another approach to solve this problem is independent vector analysis (IVA, Kim et al., 2006c). IVA is an extension of ICA, where the mixtures and the estimates are vectors. It solves both problems of the ICA approach, the lack of using dependence information and the clustering problem. IVA uses the dependence information to estimate the original signal vectors, which results the components of the estimated signal vectors in same order for each dataset. Figure 3 shows an example of two dependent datasets of independent source signals, their mixtures and the source signals estimated by fastIVA algorithm with the multivariate diagonal Laplace source density model.

IVA has numerous applications, where analyzing multiple dependent datasets jointly is necessary (Adalı et al., 2014). For example, in the field of biomedical signal processing there are already multiple use cases such as electroencephalography (EEG) and functional magnetic resonance imaging (fMRI) collected from multiple subjects, among many others (Adalı et al., 2014). EEG collects multisensor data of patient's brain activity. When IVA is applied to extract the original sources from the EEG data, the data collected from one patient is

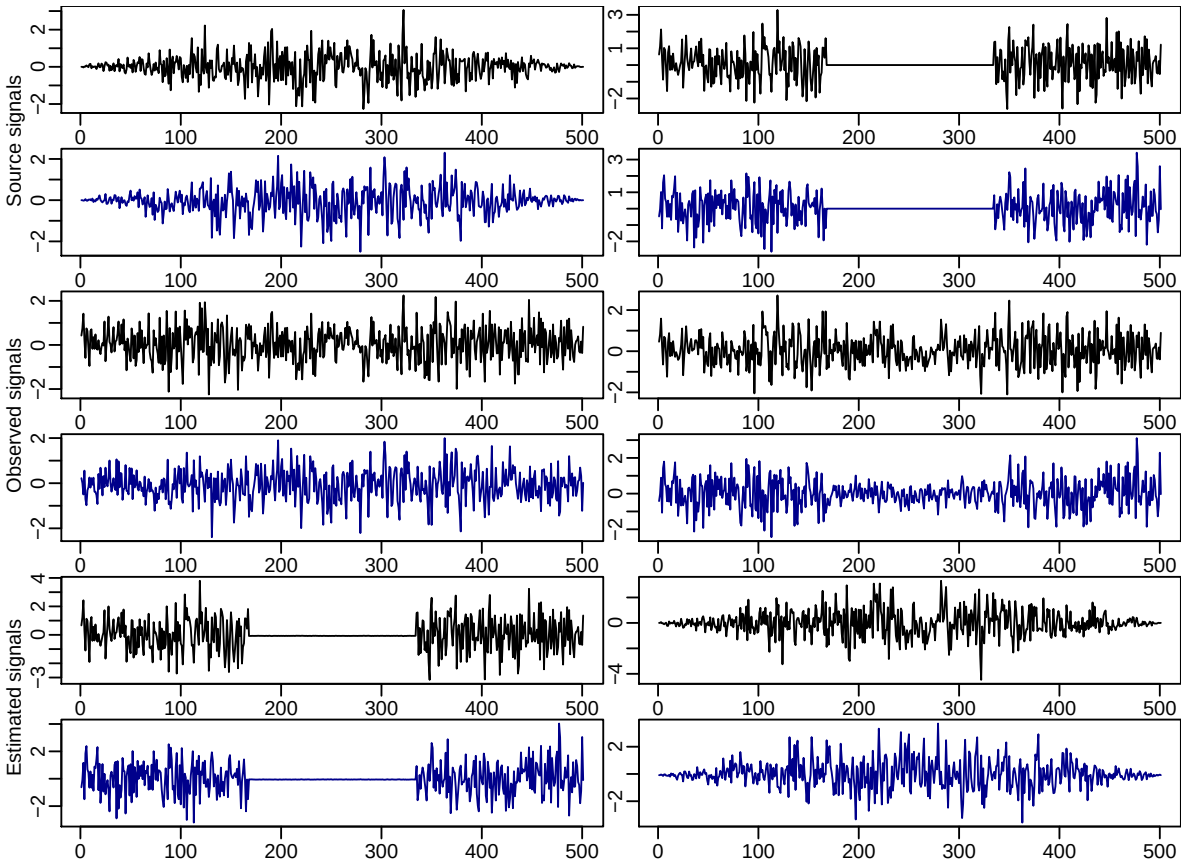

Figure 3: Example of original source signals, observed mixtures and estimated source signals when there are two dependent datasets. The signals of the first dataset are colored black and the signals of the second one are colored blue. Signals are separated with fast fixed-point IVA (fastIVA) algorithm using the multivariate diagonal Laplace source density model. Source signals are artificially generated examples, not real acoustic signals.

considered as one dataset, and the signals from the different sensors are the observed mixtures. IVA have been used for example to remove muscle artifacts from EEG data to perceive the brain activity better (Chen et al., 2017).

In fMRI group studies, the same fMRI imaging is performed to multiple patients. Different images of the brains of one patient compose one dataset, and the voxels of one image compose one observed signal (Lee et al., 2008b; Ma et al., 2014). The images are composed of three dimensional voxels rather than pixels, as the fMRI images are three dimensional slices of the brain. The group fMRI studies using IVA have been used for example to find the differences of the brain activity between patients with schizophrenia and the healthy controls (Ma et al., 2014). Recently IVA has been adopted together with independent subspace analysis to more complex data, where the number of datasets can be hundreds, allowing for example even larger group studies of fMRI or EEG (Silva et al., 2021; Long et al., 2020).

IVA is a well studied method in the field of acoustic signal processing. In acoustic signal applications, multiple acoustic signals have been recorded and transformed into time-frequency domain, where acoustic signals are composed of different frequency bands for each time frame. One frequency domain is considered as one dataset and the recorded acoustic signals as the observed mixtures. IVA have been used a lot for acoustic signal separation tasks when there are multiple interesting sources to recover (Lee et al., 2007; Scheibler and Ono, 2019; Ono, 2011), but also for acoustic signal enhancement, where the acoustic signal by is enhanced by extracting any noise of it (Zhao et al., 2017).

This thesis focuses on Newton update based algorithms to solve the IVA problem. In Section 2 the theory of ICA and IVA is introduced, and an objective function of IVA, i.e. a function to minimize to solve the IVA problem, is defined. In Section 3 Newton update based IVA algorithm and four possible source density models are introduced, and other possible source density models are discussed shortly. The aim is to implement the methods in R software (R Core Team, 2020) and verify and compare the algorithms with simulations in Section 4. In Section 5 final comparisons of implemented algorithms are made with real data.

2 Independent Vector Analysis

This section covers the general idea of independent vector analysis. Independent component analysis is first explained in Section 2.1, as IVA is a multidimensional extension of ICA (Kim et al., 2006c). In Section 2.2 the main idea of IVA is formulated and the assumptions for IVA are defined. Section 2.2 also explains the differences, similarities and a relationship between independent component analysis and independent vector analysis, and the benefits of using IVA model over several ICA models are considered. The objective function for IVA is defined in Section 2.3.

2.1 Independent component analysis

Before introducing independent vector analysis it is necessary to understand independent component analysis, because IVA is an extension of ICA to multivariate components that has not one but several dependent datasets of observed signals (Kim et al., 2006c). ICA is nowadays a well-known and commonly used approach to solve blind source separation problems (Lee et al., 2007). The goal of ICA is to find linear transformation for the observed signals so that it minimizes the statistical dependence between its components (Comon, 1994). In this section ICA is defined and restrictions and ambiguities of ICA are stated (Hyvärinen and Oja, 2000; Comon, 1994; Lee et al., 2007; Comon, 1992; Hyvärinen et al., 2001).

Let P to be the number of independent source components and the number of different observed signals. Each source component and each observed signal have the length of T. The tth observation of the observed mixture i is denoted as $x_i(t)$, $i = 1, ..., P$, $t = 1, ..., T$. One observation means one measurement of a specific signal. Observations are often measurements over time, when t is a time point of a measurement and T is a total number of time points. The observed signals can be denoted also as vectors $\mathbf{x}_i = [x_i(1), x_i(2), ..., x_i(T)]^\top$ containing all observations as components of the vector or as $P \times T$ matrix $\boldsymbol{X}$, which contains all observed signals $\mathbf{x}_i$ as rows. The superscript $\top$ denotes the transpose. The matrices are presented in a form typical to signal processing, when the signals $i = 1, ...P$ are presented row wise and the observations $t = 1, ..., T$ are in columns. The observed signals are assumed to be linear mixtures of the independent source components $s_j(t)$, $j = 1, ..., P$, denoted as

$$x_i(t) = a_{i1}s_1(t) + a_{i2}s_2(t) + ... + a_{iP}s_P(t), \tag{1}$$

where weights a_{ij} indicate how strongly each independent source component $s_j(t)$ is

contributing in observed signal $x_i(t)$. Similarly as the observed signals, the independent source components can be displayed as vectors $\mathbf{s}_j = [s_j(1), s_j(2), ..., s_j(T)]^\top$ containing all observations or as $P \times T$ matrix $\boldsymbol{S}$ containing the independent components as rows. In this thesis the sources are assumed to be independently and identically distributed (iid). The mixing environment is assumed to remain the same for each observation t. In other words, parameters a_{ij} are fixed for each observation t. The observed mixtures (1) are usually displayed in a matrix form

$$\boldsymbol{X} = \begin{bmatrix} x_1(1) & \dots & x_1(T) \\ \vdots & \ddots & \vdots \\ x_P(1) & \dots & x_P(T) \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1P} \\ \vdots & \ddots & \vdots \\ a_{P1} & \dots & a_{PP} \end{bmatrix} \begin{bmatrix} s_1(1) & \dots & s_1(T) \\ \vdots & \ddots & \vdots \\ s_P(1) & \dots & s_P(T) \end{bmatrix} = \boldsymbol{A}\boldsymbol{S},$$

where $\boldsymbol{X}$ contains the observed signal vectors $\mathbf{x}_i$ as rows, $P \times P$ matrix $\boldsymbol{A}$ contains the weights for each linear mixture $\mathbf{x}_i$ as rows and $\boldsymbol{S}$ contains the the independent component vectors as rows. Matrix $\boldsymbol{A}$ is called a mixing matrix, which is fixed for each observation, as stated earlier.

In ICA model the mixing matrix $\boldsymbol{A}$ and the latent independent components $\boldsymbol{S}$ are unknown. The only observed variables are the observed mixtures $\boldsymbol{X}$, which are used to estimate the independent source components $\boldsymbol{S}$. To achieve this, the following assumptions must hold:

A1. $\mathbf{s}_j \perp\!\!\!\perp \mathbf{s}_i$ for all $i, j = 1, ..., P$, $i \neq j$.

A2. $E(\mathbf{s}_j) = 0$, $j = 1, ..., P$ and $E(\mathbf{S}\mathbf{S}^\top) = \boldsymbol{I}_P$, where $\boldsymbol{I}_P$ is a $P \times P$ identity matrix.

The assumption of source components being independent of each other (A1) is natural, as ICA estimates the sources by maximizing the independence. The assumption of whiteness (A2) simplifies the theory and algorithms a lot, which is discussed more in Section 2.3 and in Section 3. If the assumption (A2) does not hold, the observed mixtures are whitened before applying ICA. The whitening includes centering the rows of the observed data $\boldsymbol{X}$ and finding a whitening matrix $\boldsymbol{V}$, which transforms the data to be uncorrelated with unit variances, i.e. that $E(\mathbf{S}\mathbf{S}^\top) = \boldsymbol{I}_P$ (Hyvärinen et al., 2001). The whitening is explained more in Section 3.1. With (A1) and (A2) assumed, ICA has the following identification condition:

Theorem 1 (ICA Nonidentifiability with iid samples)**.** *The sources cannot be identified if and only if* $\exists\, i \neq j$ *such that* $\mathbf{s}_i$ *and* $\mathbf{s}_j$ *follows Gaussian distribution.*

In other words, to make the sources identifiable, at most one of the source components can be Gaussian. For the proof of Theorem 1, see e.g. Hyvärinen et al. (2001).

In this thesis, the number of source components or vectors P is assumed to be the same as the number of observed mixtures in both ICA and IVA. This is because it makes the mixing matrix $\boldsymbol{A}$ square and invertible, and most of the ICA and IVA algorithms rely on calculating inverse matrices during the optimization process. In addition, the number of observed mixtures have to be larger or equal to the number of independent components to make mixing matrix $\boldsymbol{A}$ identifiable. If there are more observed mixtures than sources, one way is to reduce dimension of observations with principal component analysis to achieve square mixing matrices (Hyvärinen et al., 2001). If the number of sources is larger than the number of observed mixtures, the mixing matrix $\boldsymbol{A}$ becomes unidentifiable.

For recovering the original independent source components, one would have to solve the mixing matrix $\boldsymbol{A}$ and calculate its inverse $\boldsymbol{A}^{-1}$ to obtain

$$\boldsymbol{S} = \boldsymbol{A}^{-1}\boldsymbol{X}.$$

It is not possible to solve the mixing matrix $\boldsymbol{A}$ exactly, but it can be estimated using the assumptions of independence and non-Gaussianity of source components. The objective is to solve $P \times P$ unmixing matrix $\boldsymbol{W}$ which estimates the inverse of the mixing matrix $\boldsymbol{A}$. It is estimated by solving an optimization problem so that the resulting components

$$\hat{\boldsymbol{S}} = \boldsymbol{W}\boldsymbol{X},$$

become as statistically independent as possible. Figure 4 illustrates the whole ICA structure from original source signals to observations, and again, from observations to estimated sources.

If the unmixing matrix $\boldsymbol{W}$ and independent components $\hat{\boldsymbol{S}}$ were estimated for the centered observations $\boldsymbol{X}'$, the estimates for non-centered independent components can be calculated as

$$\hat{\boldsymbol{S}} = \boldsymbol{W}\boldsymbol{X}'.$$

Because of the fact that both true mixing matrix $\boldsymbol{A}$ and independent source components $\boldsymbol{S}$ are unknown, the mixing matrix can be recovered only up to signs of the rows $\boldsymbol{a}_i$ and arbitrary permutation. This means that $\boldsymbol{A}$ can be identified up to $\boldsymbol{A}\boldsymbol{J}\boldsymbol{P}$, where $\boldsymbol{J} = diag(\pm 1, ..., \pm 1)$ and $\boldsymbol{P}$ is any permutation matrix. ICA cannot determine the order of the source components, because one can freely change the order the terms of summation (1) without affecting the outcome $x_i(t)$.

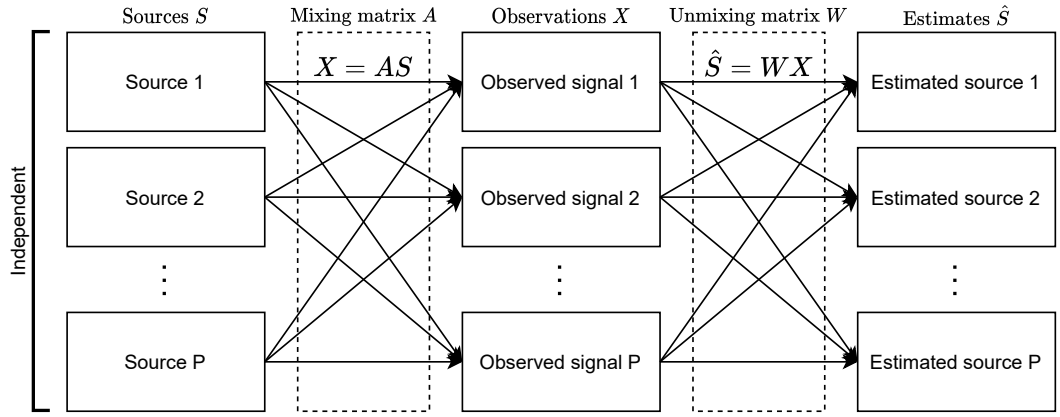

Figure 4: A graph of ICA model with P sources and P observed signals. The graph illustrates the mixing process from independent source components $\boldsymbol{S}$ to observations $\boldsymbol{X}$ and the unmixing process from observations to source estimates. The matrix $\boldsymbol{A}$ is a mixing matrix and the matrix $\boldsymbol{W}$ is an estimated unmixing matrix.

There are multiple different approaches to solve the ICA problem. One approach is to maximize non-Gaussianity of estimated source components, which also maximizes the independence of the components. Non-Gaussianity can be measured by kurtosis or negentropy (Hyvärinen et al., 2001). Kurtosis is a classic measure of non-Gaussianity based on fourth-order statistics. Kurtosis of a random variable x, denoted by $kurt(x)$, is defined as

$$kurt(x) = E(x^4) - 3(E(x^2))^2.$$

Negentropy $\boldsymbol{J}$ is a normalized version of differential entropy, defined as

$$\boldsymbol{J}(\mathbf{x}) = H(\mathbf{x}_{gauss}) - H(\mathbf{x}),$$

where $H(\cdot)$ denotes entropy, $\mathbf{x}$ is a random vector and $\mathbf{x}_{gauss}$ is a Gaussian random vector with same correlation matrix as vector $\mathbf{x}$. The entropy is discussed more in Section 2.3. Another approach is maximum likelihood estimation of the unmixing matrix $\boldsymbol{W}$, where the maximum likelihood is solved for density function of observed mixture $\boldsymbol{X}$ with respect to unmixing matrix $\boldsymbol{W}$. Some of the other approaches are minimization of mutual information, tensorial methods and nonlinear decorrelation and nonlinear PCA (Hyvärinen et al., 2001).

Dozens of different algorithms have been developed to solve the optimization problem of ICA. For example, two types of algorithms that are well known, are gradient algorithms and fast-fixed point algorithms (FastICA). The gradient algorithms updates the elements of

the unmixing matrix $\boldsymbol{W}$ based on gradient descent. The fast-fixed point algorithms can be described as an approximative Newton algorithms. For other approaches and algorithms, based on minimization of mutual information, tensorial methods and nonlinear decorrelation and nonlinear PCA, see Hyvärinen et al. (2001). The approaches and algorithms for ICA are not discussed more deeply in this thesis as the main focus is on algorithms for IVA.

2.2 IVA problem formulation and assumptions

Independent vector analysis (IVA) is a multidimensional extension of ICA. In IVA multiple dependent datasets of the same phenomenon are analysed jointly. In this section IVA problem is formulated, and assumptions and identification condition for IVA model are stated (Kim et al., 2006c; Lee et al., 2007; Na et al., 2013; Anderson, 2013; Anderson et al., 2014).

IVA aims to recover independent source components jointly for D dependent datasets using observations $\boldsymbol{X}^{[d]}$. Let P be the number of independent source vectors and the number of observed mixtures, each containing T observations. Each observation contains D dependent elements, one for each dataset $d = 1, ..., D$. The tth observation of the observed mixture i is denoted as $\mathbf{x}_i(t) = (x_i^{[1]}(t), x_i^{[2]}(t), ..., x_i^{[D]}(t))^\top$, where $x_i^{[d]}(t)$ is the observation of dataset d. Observed signals can be displayed dataset-wise as vectors $\mathbf{x}_i^{[d]} = (x_i^{[d]}(1), x_i^{[d]}(2), ..., x_i^{[d]}(T))^\top$, which contains all observations of the ith mixture signal for dataset d or as matrices $\boldsymbol{X}^{[d]}$, which contains observed signals $\mathbf{x}_i^{[d]}$ as rows. Observed signals are assumed to be linear mixtures of independent source vectors $\mathbf{s}_j(t) = (s_j(t)^{[1]}, s_j(t)^{[2]}, ..., s_j(t)^{[D]})^\top$, $j = 1, ..., P$, denoted as

$$\mathbf{x}_i(t) = \begin{bmatrix} x_i(t)^{[1]} \\ \vdots \\ x_i(t)^{[D]} \end{bmatrix} = \sum_{j=1}^{P} \begin{bmatrix} a_{ij}^{[1]} \\ \vdots \\ a_{ij}^{[D]} \end{bmatrix} \begin{bmatrix} s_j(t)^{[1]} & \dots & s_j(t)^{[D]} \end{bmatrix} = \sum_{j=1}^{P} \mathbf{a}_{ij} \mathbf{s}_j^\top(t)$$

where weights $a_{ij}^{[d]}$ indicate how strongly independent component $s_j^{[d]}$ is contributing in observed signal $x_i^{[d]}$. Independent source vectors $\mathbf{s}_j(t) = (s_j(t)^{[1]}, ..., s_j(t)^{[D]})^\top$ contain D dependent elements, one for each dataset. The mixing vector $\boldsymbol{a}_{ij} = (a_{ij}^{[1]}, ..., a_{ij}^{[D]})^\top$ contains weights for the whole source vector $\mathbf{s}_j(t)$. IVA model is usually displayed in a matrix form for each dataset d separately, similarly as in the ICA model:

$$\boldsymbol{X}^{[d]} = \begin{bmatrix} x_1^{[d]}(1) & \dots & x_1^{[d]}(T) \\ \vdots & \ddots & \vdots \\ x_P^{[d]}(1) & \dots & x_P^{[d]}(T) \end{bmatrix} = \begin{bmatrix} a_{11}^{[d]} & \dots & a_{1P}^{[d]} \\ \vdots & \ddots & \vdots \\ a_{P1}^{[d]} & \dots & a_{PP}^{[d]} \end{bmatrix} \begin{bmatrix} s_1^{[d]}(1) & \dots & s_1^{[d]}(T) \\ \vdots & \ddots & \vdots \\ s_P^{[d]}(1) & \dots & s_P^{[d]}(T) \end{bmatrix} = \boldsymbol{A}^{[d]}\boldsymbol{S}^{[d]},$$

where $\boldsymbol{X}^{[d]}$ contains observed signal vectors as rows, $\boldsymbol{A}^{[d]}$ contains weights for each linear mixture $\mathbf{x}_i^{[d]}$ as rows and $\boldsymbol{S}^{[d]}$ contains the source signal vectors as rows. The mixing matrices $\boldsymbol{A}^{[d]}$ are fixed for each observation $x_i^{[d]}(t)$, but can be different for each dataset d. The sources can be presented also as $D \times T$ source vector matrices $\boldsymbol{S}_j = [\boldsymbol{s}_j^{[1]}, ..., \boldsymbol{s}_j^{[D]}]^\top$, $j = 1, ..., P$. Mixing matrices $\boldsymbol{A}^{[d]}$ and sources $\boldsymbol{S}^{[d]}$ are unknown quantities to be estimated with using only the observed signals $\boldsymbol{X}^{[d]}$ and the following assumptions:

B1. $\mathbf{s}_j^{[d]} \perp\!\!\!\perp \mathbf{s}_i^{[d]}$ for all $i, j = 1, ..., P$, $d = 1, ..., D$, $i \neq j$.

B2. $E(\mathbf{s}_j^{[d]}) = 0$ and $E(\boldsymbol{S}^{[d]}(\boldsymbol{S}^{[d]})^\top) = \boldsymbol{I}_P$, $j = 1, ..., P$, $d = 1, ..., D$. $\boldsymbol{I}_P$ is a $P \times P$ identity matrix.

B3. The rows in $\boldsymbol{S}_j$, $j = 1, ...P$, are dependent on each other.

The assumption (B1) is obvious, as IVA aims to recover the original source components by maximizing independence between source components. The source components are assumed to be white (B2) to simplify the algorithms and to solve the scaling ambiguity of IVA. To make the assumption of whiteness hold, the observed data are whitened, which is explained in Section 3.1. When the sources possess dependence across the datasets (B3), the source estimates are expected to be in same order for each dataset. However, if (B3) does not hold, the sources can still be separated, but not necessarily aligned. If the components within the dataset are independent of each other, the IVA is not different from using ICA to each dataset separately.

IVA aims to find D $P \times P$ unmixing matrices $\boldsymbol{W}^{[d]}$ as well as the corresponding source estimates $\hat{\boldsymbol{S}}^{[d]}$ for each dataset d, which is denoted as

$$\hat{\boldsymbol{S}}^{[d]} = \begin{bmatrix} \hat{s}_1^{[d]}(1) & \dots & \hat{s}_1^{[d]}(T) \\ \vdots & \ddots & \vdots \\ \hat{s}_P^{[d]}(1) & \dots & \hat{s}_P^{[d]}(T) \end{bmatrix} = \begin{bmatrix} w_{11}^{[d]} & \dots & w_{1P}^{[d]} \\ \vdots & \ddots & \vdots \\ w_{P1}^{[d]} & \dots & w_{PP}^{[d]} \end{bmatrix} \begin{bmatrix} x_1^{[d]}(1) & \dots & x_1^{[d]}(T) \\ \vdots & \ddots & \vdots \\ x_P^{[d]}(1) & \dots & x_P^{[d]}(T) \end{bmatrix} = \boldsymbol{W}^{[d]}\boldsymbol{X}^{[d]},$$

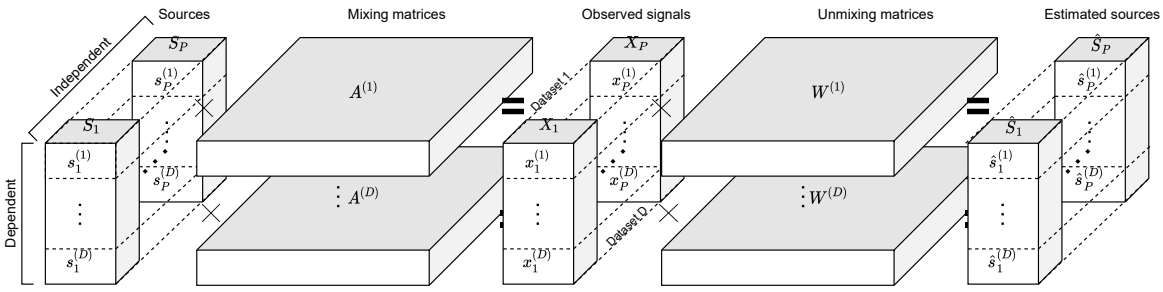

Figure 5: A graph of IVA model with P sources, P observed signals and D datasets. The graph illustrates the mixing process from independent source vectors $\boldsymbol{S}_j$ to observed signals $\boldsymbol{X}_i$ and the unmixing process from observed signals to source estimates. Matrices $\boldsymbol{A}^{[d]}$ are mixing matrices and matrices $\boldsymbol{W}^{[d]}$ are estimated unmixing matrices. Source signals $\boldsymbol{S}_j$, observed signals $\boldsymbol{X}_i$ and estimated sources $\hat{\boldsymbol{S}}_j$ are denoted as matrices containing all observations for each dataset.

where matrix $\hat{\boldsymbol{S}}^{[d]}$ contains estimated source vectors as rows and matrix $\boldsymbol{W}^{[d]}$ contains the unmixing weights. The column vector $\boldsymbol{w}_j^{[d]}$ contains the unmixing weights for jth component of $\hat{\boldsymbol{S}}^{[d]}$. Estimated source vectors can be displayed also as vectors for each observation $t = 1, ..., T$ as

$$\hat{\mathbf{s}}_j(t) = \sum_{j=1}^{P} \mathbf{w}_{ij} \mathbf{x}_i^{\top}(t),$$

where $\hat{\mathbf{s}}_j(t) = (\hat{s}_j(t)^{[1]}, ...\hat{s}_j(t)^{[D]})^{\top}$ is jth estimated source vector and $\mathbf{w}_{ij} = (w_{ij}^{[1]}, ..., w_{ij}^{[D]})^{\top}$ contains the corresponding unmixing weights. The unmixing matrices $\boldsymbol{W}^{[d]}$ are estimates of the inverse of true mixing matrices $\boldsymbol{A}^{[d]}$. These are estimated by solving an optimization problem so that the resulting components $\hat{\boldsymbol{S}}^{[d]}$ become as independent as possible. Figure 5 illustrates the whole structure of IVA model from real sources to observed mixtures, and from observed mixtures to source estimates.

The IVA problem can be solved also as several ICA problems, one for each dataset d. However, if one uses ICA separately for each dataset, the dependence information is neglected and thus it is not an optimal solution. Also, the permutation of the source estimates might be different for each dataset, as stated in ambiguities of ICA, causing a clustering problem of solving which estimates belong together between the datasets.

As IVA is an multidimensional extension of ICA, IVA has the same ambiguities, but dataset wise. The mixing matrices $\boldsymbol{A}^{[d]}$, $d = 1, ...D$, can be estimated only up to signs of rows $\boldsymbol{a}_i^{[d]}$ and arbitrary permutation. This means that the ordering and the signs of the estimated

sources might not be correct. Hence the sources $\boldsymbol{S}^{[d]}$ are considered identified if the mixing matrix $\boldsymbol{A}^{[d]}$ is identified up to $\boldsymbol{A}^{[d]}\boldsymbol{J}^{[d]}\boldsymbol{P}^{[d]}$, where $\boldsymbol{J}^{[d]} = diag(\pm 1, ..., \pm 1)$ and $\boldsymbol{P}^{[d]}$ is any permutation matrix.

In addition to previous ambiguities, the identification condition for IVA is presented in Theorem 2. For Theorem 2, Definition 1 of $\boldsymbol{\alpha}$-Gaussian component is introduced. For Theorem 2 and Definition 1, $\boldsymbol{s}_j = [s_j^{[1]}, ..., s_j^{[d]}]^\top$ is considered as a population level random variable, $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_D]$ is $1 \times D$ vector, where α_d equals either 1 or 0, $\boldsymbol{\alpha^c} = [1-\alpha_1, ..., 1-\alpha_D]$ is $1 \times D$ complement vector of $\boldsymbol{\alpha}$. Then $1 \times D_\alpha$ vector $\boldsymbol{\alpha}^\top \boldsymbol{s}_j$ contains the subset of $\boldsymbol{s}_j$ and $1 \times D - D_\alpha$ complement vector $\boldsymbol{\alpha}^{c\top} \boldsymbol{s}_j$ contains the complement subset of $\boldsymbol{s}_j$, where $D_\alpha = \sum_{d=1}^{D} \alpha_d$.

Definition 1 ($\boldsymbol{\alpha}$-Gaussian)**.** *The multivariate random variable $\boldsymbol{s}_j$ has an $\boldsymbol{\alpha}$ − Gaussian component when $\boldsymbol{\alpha}^\top \boldsymbol{s}_j \perp\!\!\!\perp \boldsymbol{\alpha}^{c\top} \boldsymbol{s}_j$ and $\boldsymbol{\alpha}^\top \boldsymbol{s}_j \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_{j,\alpha})$, where the $D_\alpha \times D_\alpha$ covariance matrix $\boldsymbol{R}_{j,\alpha} = E(\boldsymbol{\alpha}^\top \boldsymbol{s}_j (\boldsymbol{\alpha}^\top \boldsymbol{s}_j)^\top)$ is nonsingular.*

The $\boldsymbol{\alpha}$-Gaussian definition is used to identify that there exists a subset of components of source vector that is independent of the other components in the same source vector and that the source vector is multivariate Gaussian. The identification condition for IVA in case of iid samples is following:

Theorem 2 (IVA Nonidentifiability with iid samples)**.** *The sources cannot be identified if and only if $\exists\ i \neq j$ such that $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ have $\boldsymbol{\alpha}$-Gaussian components and $\exists\ \boldsymbol{\alpha}$ such that $0 < \sum_{d=1}^{D} \alpha_d < D$, and $\boldsymbol{R}_{i,\alpha} = \boldsymbol{D}\boldsymbol{R}_{j,\alpha}\boldsymbol{D}$, where $\boldsymbol{D}$ is any full-rank diagonal matrix.*

See Anderson et al. (2014) or Anderson (2013) for proof and more about the IVA identification condition.

2.3 Objective function

To solve the IVA optimization problem, an objective function has to be defined for measuring how good the solution is. The mutual information $I(\cdot)$ is a measure of information obtained about members in a set of random variables by observing the other random variables in the set. This is often used as an objective function for ICA and IVA. Mutual information can be defined with Kullback-Leibler divergence (Kullback and Leibler, 1951) $D_{KL}(\cdot|\cdot)$ as

$$I(x_1, ..., x_P) = D_{KL}\left(p_{\mathbf{x}}(x_1, ..., x_P) | \prod_{j=1}^{P} p_{x_i}(x_i)\right),$$

where $p_{\mathbf{x}}(x_1, ..., x_P)$ is a joint probability distribution of variables $x_1, ..., x_P$ and $\prod_{j=1}^{P} p_{x_i}(x_i)$ is a product of marginal probability distributions of $x_1, ..., x_P$ (Lee et al., 2007). The variables $x_1, ..., x_P$ can be either univariate or multivariate. In context of IVA, the variables are multivariate. The first one is exact joint probability distribution of the source estimates $p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P)$ and the other one is a product of the marginal probability distributions $\prod_{j=1}^{P} p_{s_j}(\hat{\mathbf{s}}_j)$, where true source distributions $p_{\mathbf{s}}$ and p_{s_j} are unknown. The estimation of the unknown source distributions is discussed later. The mutual information can be considered as a metric which measures the distance between a distribution and the product of its marginal distributions. When the distance is zero, the joint distribution and the product of the marginal distributions are the same, implying independence between the source estimates. In this thesis the mutual information, measured with Kullback-Leibler divergence, is used as an objective function (Kim et al., 2006c; Lee et al., 2007; Anderson, 2013; Na et al., 2013). In this section source estimates $\hat{\mathbf{s}}_j = (\hat{s}_j^{[1]}, ..., \hat{s}_j^{[D]})^\top$ are considered as population level random variables, and thus the index t is dropped out of notations.

To understand the Kullback-Leibler divergence, it is essential to recall the formulas of differential entropy (Kolmogorov, 1956), which is an extension of the original Shannon entropy (Shannon, 1948) for continuous variables, defined as

$$H(p) = -\int_{x \in \mathbf{x}} p(x) \log(p(x)) dx$$

and differential cross entropy, defined as

$$H(p, q) = -\int_{x \in \mathbf{x}} p(x) \log(q(x)) dx,$$

where p and q are probability distributions. Entropy measures the average level of information in the variable's possible outcomes and cross entropy measures the average level of information needed to identify variable under distribution p, when variable is estimated with distribution q. The better the distribution q estimates the true distribution p, the closer the cross entropy gets to entropy. In case of IVA, the entropy is the average level of information of source estimates $\hat{\mathbf{s}}_j$ under joint distribution $p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P)$. Cross entropy, on the other hand, is the average level of information needed to identify the source estimates $\hat{\mathbf{s}}_j$ when they are estimated with the product of the marginal probability distributions $\prod_{j=1}^{P} p_{s_j}(\hat{\mathbf{s}}_j)$. The

Kullback-Leibler divergence is defined as

$$D_{KL}(p|q) = \int_{x\in\mathbf{x}} p(x)\log\frac{p(x)}{q(x)}dx,$$

which can be written with the entropy functions as

$$\begin{aligned} D_{KL}(p|q) &= \int_{x\in\mathbf{x}} p(x)\log\frac{p(x)}{q(x)}dx \\ &= \int_{x\in\mathbf{x}} p(x)\log(p(x)) - \int_{x\in\mathbf{x}} p(x)\log(q(x)) \\ &= H(p,q) - H(p). \end{aligned}$$

When cross entropy $H(p,q)$ equals entropy $H(p)$, the distribution q estimates the outcomes of variable $\mathbf{x}$ exactly correct, meaning that the distributions are the same. In case of IVA, this means that the joint probability distribution $p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P)$ is equal to the product of the marginal distributions $\prod_{j=1}^{P} p_{s_j}(\hat{\mathbf{s}}_j)$, implying independence of the source estimates $\hat{\mathbf{s}}_j$ by definition. Thus, by minimizing Kullback-Leibler divergence, the independence of the source estimates is maximized.

To simplify the task of minimizing Kullback-Leibler divergence, the source estimates are assumed to be white. This is done by whitening the observed signals $\boldsymbol{X}^{[d]}$, meaning that

$$E[\boldsymbol{X}^{[d]}] = \mathbf{0}, \tag{2}$$

$$E[\boldsymbol{X}^{[d]}(\boldsymbol{X}^{[d]})^{\top}] = \boldsymbol{I}_P,\ i = 1, ..., P,\ d = 1, ..., D \tag{3}$$

where $\boldsymbol{I}_P$ is $P \times P$ identity matrix. The whitening process is explained in Section 3.

The Kullback-Leibler divergence for the joint probability distribution $p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P)$ and the

product of the marginal probability distributions $\prod_{j=1}^{P} p_{s_j}(\hat{\mathbf{s}}_j)$ is

$$
\begin{aligned}
& D_{KL}(p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P) | \prod_{j=1}^{P} p_{s_j}(\hat{\mathbf{s}}_j)) \\
&= H[p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P), \prod_{j=1}^{P} p_{s_j}(\hat{\mathbf{s}}_j)] - H[p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P)] \\
&= \sum_{j=1}^{P} H[p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P), p_{s_j}(\hat{\mathbf{s}}_j)] - H[p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P)] \qquad (4) \\
&= \sum_{j=1}^{P} H[p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P), p_{s_j}(\hat{\mathbf{s}}_j)] - H[p_{\mathbf{s}}(\boldsymbol{W}^{[1]}\mathbf{x}^{[1]}, ..., \boldsymbol{W}^{[D]}\mathbf{x}^{[D]})] \qquad (5) \\
&= \sum_{j=1}^{P} H[p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P), p_{s_j}(\hat{\mathbf{s}}_j)] - \sum_{d=1}^{D} \log|\det \boldsymbol{W}^{[d]}| - H[p_{\mathbf{s}}(\mathbf{x}^{[1]}, ..., \mathbf{x}^{[D]})]. \qquad (6)
\end{aligned}
$$

In equation (4) the cross entropy $H[p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P), \prod_{j=1}^{P} p_{s_j}(\hat{\mathbf{s}}_j)] = p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P)\log[\prod_{j=1}^{P} p_{s_j}(\hat{\mathbf{s}}_j)]$ is presented as summation $\sum_{j=1}^{P} p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P)\log[p_{s_j}(\hat{\mathbf{s}}_j)] = \sum_{j=1}^{P} H[p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P), p_{s_j}(\hat{\mathbf{s}}_j)]$. In equation (5) the estimated sources are presented dataset-wise as a product of the unmixing matrices and the observed signals, $\hat{\mathbf{s}}^{[d]} = \boldsymbol{W}^{[d]}\mathbf{x}^{[d]}$. Finally in equation (6) the entropy identity for linear transformations is used to obtain $H[p_{\mathbf{s}}(\boldsymbol{W}^{[1]}\mathbf{x}^{[D]}, ..., \boldsymbol{W}^{[1]}\mathbf{x}^{[D]})] = \sum_{d=1}^{D} \log|\det \boldsymbol{W}^{[d]}| - H[p_{\mathbf{s}}(\mathbf{x}^{[1]}, ..., \mathbf{x}^{[D]})]$.

The entropy $H[p_{\mathbf{s}}(\mathbf{x}^{[d]}, ..., \mathbf{x}^{[d]})]$ is constant and does not change over the optimization process, as the values $\mathbf{x}^{[d]}$ are observed and fixed. The constant can be dropped as it does not change the location of the minimum. Hence, the Kullback-Leibler divergence simplifies to

$$
D_{KL}(p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P) | \prod_{j=1}^{P} p_{s_j}(\hat{\mathbf{s}}_j)) = \sum_{j=1}^{P} H[p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P), p_{s_j}(\hat{\mathbf{s}}_j)] - \sum_{d=1}^{D} \log|\det \boldsymbol{W}^{[d]}|.
$$

The entropy $H[p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P), p_{s_j}(\hat{\mathbf{s}}_j)]$ can be expressed as an expected value of $\log(p_{s_j}(\hat{\mathbf{s}}_j))$:

$$\begin{aligned} & H[p_{\mathbf{s}}(\hat{\mathbf{s}}_1, ..., \hat{\mathbf{s}}_P), p_{s_j}(\hat{\mathbf{s}}_j)] \\ &= -\int_Z p_{\mathbf{s}}(z_1, ..., z_P)\log[p_{s_j}(z_j)]dZ \\ &= -\int_{\boldsymbol{z}_j} p_{s_j}(z_j)\log[p_{s_j}(z_j)]d\boldsymbol{z}_j \\ &= -E[\log(p_{s_j}(\hat{\mathbf{s}}_j))], \end{aligned}$$

where Z is a set of possible values of the sources and $\boldsymbol{z}_j$ is a set of possible values of the jth source.

The marginal distributions p_{s_j} are multivariate, $p_{s_j}(\hat{\mathbf{s}}_j) = p_{s_j}(\hat{s}_j^{[1]}, ..., \hat{s}_j^{[D]})$, where source components $\hat{s}_j^{[d]}$ are dependent. The expected value is estimated using a sample mean $E[\log(p_{s_j}(\hat{\mathbf{s}}_j))] \approx \frac{1}{T}\sum_{t=1}^{T}\log(p_{s_j}(\hat{\mathbf{s}}_j(t)))$. Because the true distributions p_{s_j} are unknown, a proper source density models $\hat{p}_j$ must be chosen. With source density models $\hat{p}_j$ selected, the objective function O_{IVA} is finally defined as

$$O_{IVA} = -\sum_{j=1}^{P} E[\log(\hat{p}_j(\hat{\mathbf{s}}_j))] - \sum_{d=1}^{D}\log|\det\boldsymbol{W}^{[d]}|. \tag{7}$$

The objective function is then minimized with respect to unmixing matrices $\boldsymbol{W}^{[d]}$ to maximize the independence of the source estimates $\hat{\mathbf{s}}_j$. Possible options for the source density models $\hat{p}_j$ and the Newton update based IVA algorithm for minimizing the objective function are considered in Section 3.

3 IVA Algorithms

Most IVA algorithms consist of two main parts, choosing proper source density models $\hat{p}_j$ and choosing an optimization method (Lee et al., 2007). In this thesis the optimization method is chosen to be Newton update based because of its popularity and efficiency in recent studies, see e.g. Anderson (2013). Other existing optimization methods for IVA are for example gradient descent based algorithms (Kim et al., 2006b; Anderson, 2013), expectation maximization (EM) based algorithm (Lee et al., 2008a; Hao et al., 2010) and auxiliary function based algorithms (Ono, 2011; Ikeshita and Nakatani, 2022). To simplify the optimization process, the data are preprocessed to be whitened, which is explained in Section 3.1. In Section 3.2 the Newton update based IVA algorithm is introduced and in Section 3.3 different source models are introduced and their use cases and differences are discussed. In this thesis the multivariate values within each source $\boldsymbol{s}_j$ are assumed to be real valued and distributed independently and identically (iid), and hence the algorithms presented here will not be optimal for data, which possess serial dependence.

3.1 The data preprocessing

As discussed earlier, the data must be whitened to satisfy constraints of zero mean (2) and identity covariance matrix (3). There are multiple ways to whiten the data, but in this thesis the data preprocessing is accomplished according to Hyvärinen et al. (2001). To make the data zero mean, it is simply centered by subtracting the rows $\boldsymbol{x}_i'^{[d]}$ of $\boldsymbol{X}'^{[d]}$ with their sample mean. The rows $\boldsymbol{x}_i^{[d]}$ of the centered data $\boldsymbol{X}^{[d]}$ are obtained by

$$\boldsymbol{x}_i^{[d]} = (\boldsymbol{x}_i'^{[d]}) - E(\boldsymbol{x}_i'^{[d]}),\ i = 1, ..., P,\ d = 1, ..., D,$$

where $\boldsymbol{x}_i^{[d]}$ is the ith row of the matrix $\boldsymbol{X}^{[d]}$. By centering the observed signals, the source estimates become zero mean:

$$E[\boldsymbol{S}^{[d]}] = E[(\boldsymbol{A}^{[d]})^{-1}\boldsymbol{X}^{[d]}] = (\boldsymbol{A}^{[d]})^{-1}E[\boldsymbol{X}^{[d]}].$$

A zero-mean random vector $\boldsymbol{y} = (y_1, ..., y_P)^\top$ is white, if its elements y_i are uncorrelated and have unit variances, i.e. $E[\boldsymbol{y}\boldsymbol{y}^\top] = \boldsymbol{I}_P$, where $\boldsymbol{I}_P$ is $P \times P$ identity matrix. Whiteness can

be obtained by finding a linear transformation $\boldsymbol{V}$ into vector $\boldsymbol{z}$ so that

$$\boldsymbol{z} = \boldsymbol{V}\boldsymbol{y}$$

becomes white. The whitening matrix $\boldsymbol{V}$ can be found in multiple ways (Ilmonen et al., 2012), one of which is by calculating the eigenvectors and the eigenvalues of the covariance matrix $\boldsymbol{C} = E[\boldsymbol{y}\boldsymbol{y}^\top]$. Let $\boldsymbol{E} = (\boldsymbol{e}_1, ..., \boldsymbol{e}_P)$ be the matrix, whose columns are eigenvectors of covariance matrix $\boldsymbol{C}$ and let $\boldsymbol{\Lambda} = diag(\lambda_1, ..., \lambda_P)$ be the matrix, whose diagonal elements λ_i are the eigenvalues of the matrix $\boldsymbol{C}$. The whitening matrix $\boldsymbol{V}$ is then given by

$$\boldsymbol{V} = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{E}^\top,$$

where $\boldsymbol{\Lambda}^{-1/2} = diag(\lambda_1^{-1/2}, ..., \lambda_P^{-1/2})$. The whitening matrix $\boldsymbol{V}$ exists always, when the eigenvalues λ_i are positive. In practice, the covariance matrix $\boldsymbol{C}$ is positive definite for almost any natural data, so its eigenvalues will be positive.

By recalling that the covariance matrix $\boldsymbol{C}$ can be written in terms of its eigenvector and eigenvalue matrices $\boldsymbol{E}$ and $\boldsymbol{\Lambda}$ as $\boldsymbol{C} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^\top$, where $\boldsymbol{E}$ is an orthogonal matrix satisfying $\boldsymbol{E}\boldsymbol{E}^\top = \boldsymbol{E}^\top\boldsymbol{E} = \boldsymbol{I}_P$, it is easy to show that $\boldsymbol{V}$ is indeed a whitening matrix. When the random vector $\boldsymbol{z}$ is written as $\boldsymbol{z} = \boldsymbol{V}\boldsymbol{y}$, it holds that

$$E[\boldsymbol{z}\boldsymbol{z}^\top] = \boldsymbol{V}E[\boldsymbol{y}\boldsymbol{y}^\top]\boldsymbol{V}^\top = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{E}^\top\boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^\top\boldsymbol{E}\boldsymbol{\Lambda}^{-1/2} = \boldsymbol{I}_P.$$

The random vector $\boldsymbol{z}$ is now white, as its covariance matrix is the unit matrix.

The centered data $\boldsymbol{X}^{[d]}$ is whitened for each dataset separately with algorithm described above. By calculating the covariance matrix $\boldsymbol{C}^{[d]}$, its eigenvector matrix $\boldsymbol{E}^{[d]}$ and eigenvalue matrix $\boldsymbol{\Lambda}^{[d]}$ for each $d = 1, ..., D$, and set $\boldsymbol{V}^{[d]} = (\boldsymbol{\Lambda}^{[d]})^{-1/2}(\boldsymbol{E}^{[d]})^\top$, the whitened data $\boldsymbol{Z}^{[d]}$ is obtained as

$$\boldsymbol{Z}^{[d]} = \boldsymbol{V}^{[d]}\boldsymbol{X}^{[d]}.$$

3.2 IVA with Newton update

The non-orthogonal Newton's method based IVA algorithm (Anderson et al., 2010), uses Newton update together with decoupling trick to update the unmixing matrices. The update is performed concurrently for each row $\boldsymbol{w}_j^{[d]}$ of the unmixing matrix $\boldsymbol{W}^{[d]}$, after which the

rows are normalized to achieve unit variance of source estimates. The unmixing vector $\boldsymbol{w}_j^{[d]}$ corresponds to jth source estimate of dataset d, $\hat{s}_j^{[d]} = \boldsymbol{w}_j^{[d]}\boldsymbol{x}^{[d]}$, where $\hat{s}_j^{[d]}$ and $\boldsymbol{x}^{[d]} = (x_1^{[d]}, ..., x_P^{[d]})^\top$ are considered as population level random variables.

Let $\hat{p}$ be the selected source density model for all sources and function G to be defined as

$$G(\hat{\boldsymbol{s}}_j) = -\log\hat{p}(\hat{\boldsymbol{s}}_j).$$

With the function G the objective function O_{IVA} in (7) becomes

$$O_{IVA} = \sum_{j=1}^{P} E[G(\hat{\mathbf{s}}_j))] - \sum_{d=1}^{D} \log|\det\boldsymbol{W}^{[d]}|.$$

When solving the unmixing matrices, it is beneficial to update each row of the unmixing matrix separately, i.e. decoupled, because using a single step size for each direction might be undesirable depending of the shape of source density's surface. Easiest way of decoupling is to restrict the unmixing matrices to be orthogonal. However, an orthogonal solution might not be optimal and hence, so called decoupling trick (Anderson, 2013) is used to achieve more flexible algorithm.

The decoupling trick is to find any unit length $P \times 1$ vector $\boldsymbol{h}_i^{[d]}$ such that $\tilde{\boldsymbol{W}}_i^{[d]}\boldsymbol{h}_i^{[d]} = \boldsymbol{0}$, where $\tilde{\boldsymbol{W}}_i^{[d]}$ is $(P-1)\times P$ matrix which contains all $P-1$ rows of the matrix $\boldsymbol{W}^{[d]}$ besides the ith row, meaning that $\tilde{\boldsymbol{W}}_i^{[d]} = [\boldsymbol{w}_1^{[d]}...\boldsymbol{w}_{i-1}^{[d]}\boldsymbol{w}_{i+1}^{[d]}...\boldsymbol{w}_P^{[d]}]$ (Anderson et al., 2012). By calculating the vectors $\boldsymbol{h}_1^{[d]}, ..., \boldsymbol{h}_P^{[d]}$ the derivative and the second derivative of $\log|\det\boldsymbol{W}^{[d]}|$ with respect of $\boldsymbol{w}_j^{[d]}$ are obtained as

$$\frac{\partial\log|\det\boldsymbol{W}^{[d]}|}{\partial\boldsymbol{w}_j^{[d]}} = \frac{\boldsymbol{h}_j^{[d]}}{(\boldsymbol{h}_j^{[d]})^\top\boldsymbol{w}_j^{[d]}} \text{ and } \frac{\partial^2\log|\det\boldsymbol{W}^{[d]}|}{\partial\boldsymbol{w}_j^{[d]}(\boldsymbol{w}_j^{[d]})^\top} = \frac{-\boldsymbol{h}_j^{[d]}(\boldsymbol{h}_j^{[d]})^\top}{((\boldsymbol{h}_j^{[d]})^\top\boldsymbol{w}_j^{[d]})^2}.$$

The vectors $\boldsymbol{h}_i^{[d]}$ are calculated as according to Anderson (2013).

Now, if Newton's method is applied to update each row $\boldsymbol{w}_j^{[d]}$, the update rule becomes

$$\boldsymbol{w}_j^{[d]} \leftarrow \boldsymbol{w}_{j,old}^{[d]} - H_{d,j}^{-1}\frac{\partial O_{IVA}}{\partial\boldsymbol{w}_j^{[d]}},$$

where the first derivative of O_{IVA} is

$$\frac{\partial O_{IVA}}{\partial \boldsymbol{w}_j^{[d]}} = E\left(\frac{\partial G(\hat{\mathbf{s}}_j)}{\partial \boldsymbol{w}_j^{[d]}}\right) - \frac{\boldsymbol{h}_j^{[d]}}{(\boldsymbol{h}_j^{[d]})^\top \boldsymbol{w}_j^{[d]}} = E\left(\frac{\partial G(\hat{\mathbf{s}}_j)}{\partial \hat{s}_j^{[d]}} \frac{\partial \hat{s}_j^{[d]}}{\partial \boldsymbol{w}_j^{[d]}}\right) - \frac{\boldsymbol{h}_j^{[d]}}{(\boldsymbol{h}_j^{[d]})^\top \boldsymbol{w}_j^{[d]}}$$
$$= E\left(\frac{\partial G(\hat{\mathbf{s}}_j)}{\partial \hat{s}_j^{[d]}} \boldsymbol{x}^{[d]}\right) - \frac{\boldsymbol{h}_j^{[d]}}{(\boldsymbol{h}_j^{[d]})^\top \boldsymbol{w}_j^{[d]}}$$

and the inverse of $P \times P$ Hessian matrix

$$H_{d,j}^{-1} = \frac{\partial^2 O_{IVA}}{\partial \boldsymbol{w}_j^{[d]} \partial (\boldsymbol{w}_j^{[d]})^\top} = E\left(\frac{\partial^2 G(\hat{\mathbf{s}}_j)}{\partial \boldsymbol{w}_j^{[d]} \partial (\boldsymbol{w}_j^{[d]})^\top}\right) + \frac{\boldsymbol{h}_j^{[d]} (\boldsymbol{h}_j^{[d]})^\top}{((\boldsymbol{h}_j^{[d]})^\top \boldsymbol{w}_j^{[d]})^2}$$
$$= E\left(\frac{\partial^2 G(\hat{\mathbf{s}}_j)}{\partial \hat{s}_j^{[d]} \partial (\hat{s}_j^{[d]})^\top} \frac{\partial^2 \hat{s}_j^{[d]}]}{\partial \boldsymbol{w}_j^{[d]} \partial (\boldsymbol{w}_j^{[d]})^\top}\right) + \frac{\boldsymbol{h}_j^{[d]} (\boldsymbol{h}_j^{[d]})^\top}{((\boldsymbol{h}_j^{[d]})^\top \boldsymbol{w}_j^{[d]})^2}$$
$$= E\left(\frac{\partial^2 G(\hat{\mathbf{s}}_j)}{\partial \hat{s}_j^{[d]} \partial (\hat{s}_j^{[d]})^\top} \boldsymbol{x}^{[d]} (\boldsymbol{x}^{[d]})^\top\right) + \frac{\boldsymbol{h}_j^{[d]} (\boldsymbol{h}_j^{[d]})^\top}{((\boldsymbol{h}_j^{[d]})^\top \boldsymbol{w}_j^{[d]})^2}.$$

In practice, the expected values $E(\cdot)$ are usually estimated by sample means

$$E\left(\frac{\partial G(\hat{\mathbf{s}}_j)}{\partial \hat{s}_j^{[d]}} \boldsymbol{x}^{[d]}\right) \approx \frac{1}{T} \sum_{t=1}^{T} \frac{\partial G(\hat{\mathbf{s}}_j(t))}{\partial \hat{s}_j^{[d]}} \boldsymbol{x}^{[d]}(t) \text{ and} \tag{8}$$

$$E\left(\frac{\partial^2 G(\hat{\mathbf{s}}_j)}{\partial \hat{s}_j^{[d]} \partial (\hat{s}_j^{[d]})^\top} \boldsymbol{x}^{[d]} (\boldsymbol{x}^{[d]})^\top\right) \approx \frac{1}{T} \sum_{t=1}^{T} \frac{\partial^2 G(\hat{\mathbf{s}}_j(t))}{\partial \hat{s}_j^{[d]} \partial (\hat{s}_j^{[d]})^\top} (\boldsymbol{x}^{[d]}(t))^\top \boldsymbol{x}^{[d]}(t), \tag{9}$$

unless the expected values can be estimated using the estimated unmixing matrices and covariance matrices of observed data, which is discussed more in Section 3.3. After updating the row $\boldsymbol{w}_j^{[d]}$ of unmixing matrix, the row is normalized by

$$\boldsymbol{w}_j^{[d]} \leftarrow \frac{\boldsymbol{w}_j^{[d]}}{\|\boldsymbol{w}_j^{[d]}\|}.$$

If the source density model $\hat{p}$ is a parametric distribution with parameter vector $\boldsymbol{\theta}$, the parameters $\boldsymbol{\theta}$ are estimated before each iteration. Different possibilities for the source density models are discussed in Section 3.4. It should be noted that Newton update based algorithm does not guarantee the estimated sources to be uncorrelated, because the decoupling trick is used instead of restricting the unmixing matrices to be orthogonal. However, when the

Input: Observed signals $\boldsymbol{X}^{[d]}$, $d = 1, ..., D$;
Center and whiten observed signals $\boldsymbol{X}^{[d]}$;
Initialize: $\boldsymbol{W}^{[d]}$ and $\hat{\boldsymbol{S}}^{[d]} = \boldsymbol{W}^{[d]}\boldsymbol{X}^{[d]}$, $d = 1, ..., D$;
while *not converged* **do**
  Estimate distribution parameters $\boldsymbol{\theta}$ if needed;
  for $d = 1, ...D$ **do**
    for $j = 1, ...N$ **do**
      Update the row j of the unmixing matrix d: $\boldsymbol{w}_j^{[d]} \leftarrow \boldsymbol{w}_{j,old}^{[d]} - H_{d,j}^{-1} \frac{\partial O_{IVA}}{\partial \boldsymbol{w}_j^{[d]}}$;
      Normalization: $\boldsymbol{w}_j^{[d]} \leftarrow \frac{\boldsymbol{w}_j^{[d]}}{\|\boldsymbol{w}_j^{[d]}\|}$;
    end
  end
  Calculate new estimates: $\hat{\boldsymbol{S}}^{[d]} = \boldsymbol{W}^{[d]}\boldsymbol{X}^{[d]}$;
end
Output: Estimated source signals $\hat{\boldsymbol{S}}^{[d]}$, $d = 1, ..., D$;

Algorithm 1: IVA algorithm with Newton update

algorithm separates the sources successfully, the source estimates will be nearly uncorrelated.

The convergence of the algorithm is measured by

$$\epsilon = 1 - min_d(diag(\boldsymbol{W}_{old}^{[d]}(\boldsymbol{W}^{[d]})^\top)),$$

which measures the maximum change in the rows of the unmixing matrices. It holds that for any unit length vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, $-1 \leq \boldsymbol{a}^\top\boldsymbol{b} \leq 1$, where -1 is obtained when $\boldsymbol{a} = -\boldsymbol{b}$ and 1 is obtained when $\boldsymbol{a} = \boldsymbol{b}$. Because the rows of the unmixing matrices are unit vectors, it holds that $0 \leq \epsilon \leq 2$, where $\epsilon = 0$ is obtained when $\boldsymbol{w}_{old,j}^{[d]} = \boldsymbol{w}_j^{[d]}$ for all $i = 1, ...P$ and $d = 1, ..., D$, and $\epsilon = 2$ is obtained when there exists such i and d that $\boldsymbol{w}_{old,j}^{[d]} = -\boldsymbol{w}_j^{[d]}$. The unmixing matrices can be initialized for example as identity matrices, matrices with random values or estimated unmixing matrices of another IVA algorithm. The complete algorithm is summarized in Algorithm 1.

3.3 Source density models

As mentioned in Section 2.3, proper source density models $\hat{p_j}$ must be chosen to model the true sources. The effectiveness of IVA is determined by how well the selected density models match the true source distributions and by the optimization algorithm chosen (Mowakeaa

et al., 2020). Many source density models with varying properties and use cases have been proposed since the original IVA (Kim et al., 2006b,c) was first introduced.

In this thesis all source density models are symmetric and most likely will not be optimal for asymmetrically distributed sources. For IVA, the elements within each source are assumed to be dependent on each other. The dependence can be divided into two types, second-order (linear) dependence and higher-order (non-linear) dependence. The elements are second-order dependent, if they are correlated, i.e. $Cov(s_j^{[d_1]}, s_j^{[d_2]}) \neq 0,\ d_1 \neq d_2,\ d_1, d_2 = 1, ..., D$. If the elements are uncorrelated, but still not independent, they are dependent through some higher-order moment.

The original IVA was proposed using multivariate Laplace distribution with diagonal covariance structure as source density model. Although the original papers never mention Laplace distribution, it has been later stated to be multivariate Laplace distribution indeed (Lee et al., 2008b). The original IVA algorithm, called fastIVA in this thesis, is also a Newton update based algorithm, but restricted only for orthogonal unmixing matrices. The multivariate extension of univariate Laplace distribution can be defined in many ways, one of which is

$$p_L(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\det(\boldsymbol{\Sigma})^{-1/2}}{2^D \boldsymbol{\pi}^{(D-1)/2} \Gamma\left(\frac{D+1}{2}\right)} \exp\left(-\sqrt{(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}\right), \tag{10}$$

where D is the dimension of multivariate variable $\boldsymbol{x}$, $\boldsymbol{\mu}$ is a $1 \times D$ location parameter and $\boldsymbol{\Sigma}$ is $D \times D$ dispersion parameter (Arslan, 2010). By setting $D = 1$, p_L simplifies to density function of univariate Laplace distribution. The original IVA algorithm assumes the sources $\boldsymbol{s}_j$ to be higher-order dependent and second-order uncorrelated with unit variance, i.e. the covariance matrix $\boldsymbol{\Sigma}_j = E[\boldsymbol{s}_j \boldsymbol{s}_j^\top] = \boldsymbol{I}_D$, where $\boldsymbol{I}_D$ is a $D \times D$ identity matrix (Kim et al., 2006b). When the zero mean constraint of source vectors is taken into account, the location parameter $\boldsymbol{\mu} = \mathbf{0}$. The assumed form of distribution of source estimates $\hat{\boldsymbol{s}}_j$ is thus

$$p_L(\hat{\mathbf{s}}_j) = \frac{1}{2^D \boldsymbol{\pi}^{(D-1)/2} \Gamma\left(\frac{D+1}{2}\right)} \exp\left(-\sqrt{\hat{\mathbf{s}}_i^\top \hat{\mathbf{s}}_i}\right), \tag{11}$$

where the distribution has no unknown parameters. By dropping the constant part in (11), it

is clear that

$$p_L(\hat{\mathbf{s}}_i) \propto \exp\left(-\sqrt{\hat{\mathbf{s}}_i^\top \hat{\mathbf{s}}_i}\right).$$

When using the multivariate Laplace distribution with diagonal covariance structure as source density model, the G function and its first and second derivatives are

$$G(\hat{\boldsymbol{s}}_j) = \sqrt{\hat{\boldsymbol{s}}_j^\top \hat{\boldsymbol{s}}_j}, \quad \frac{\partial G(\hat{\boldsymbol{s}}_j)}{\partial \hat{s}_j^{[d]}} = \frac{\hat{s}_j^{[d]}}{\sqrt{\hat{\boldsymbol{s}}_j^\top \hat{\boldsymbol{s}}_j}} \text{ and}$$

$$\frac{\partial^2 G(\hat{\boldsymbol{s}}_j)}{\partial \hat{s}_j^{[d]} \partial (\hat{s}_j^{[d]})^\top} = (\hat{\boldsymbol{s}}_j^\top \hat{\boldsymbol{s}}_j)^{-1/2} - (\hat{\boldsymbol{s}}_j^\top \hat{\boldsymbol{s}}_j)^{-3/2} (\hat{s}_j^{[d]})^2.$$

The Newton update based IVA algorithm with multivariate diagonal Laplace is called IVA-L-diag in this thesis. The multivariate Laplace distribution is more heavy-tailed than the multivariate Gaussian distribution, which makes it more suitable for sparsely distributed data (Lee et al., 2008b). The second-order uncorrelated multivariate Laplace distribution, as introduced in (11), assumes that elements in a source vector have only higher-order dependence, and no second-order dependence. This makes the multivariate diagonal Laplace distribution suitable for applications such as frequency domain blind source separation, where second-order correlation is minimal (Kim et al., 2006a).

Another widely used source density model is multivariate Gaussian distribution, which was introduced in context of IVA by Anderson et al. (2011). Newton update based IVA algorithm with multivariate Gaussian source density model, called IVA-G, is suitable for sources with second-order correlation within the source vectors. IVA-G assumes the source estimates $\hat{\boldsymbol{s}}_j$ to be distributed in multivariate Gaussian distribution

$$p_G(\hat{\boldsymbol{s}}_j | \boldsymbol{\Sigma}_j) = \frac{1}{(2\pi)^{K/2} \det(\boldsymbol{\Sigma}_j)^{1/2}} \exp\left(-\frac{1}{2} \hat{\boldsymbol{s}}_j^\top \boldsymbol{\Sigma}_j^{-1} \hat{\boldsymbol{s}}_j\right),$$

which models only second-order correlation and no higher-order dependence. The location parameter $\boldsymbol{\mu}$ is dropped, as the sources are zero mean and thus location is fixed to zero. To satisfy the assumption of components being dependent in a source vector, $\boldsymbol{\Sigma}_j$ cannot be assumed identity matrix as in IVA-L-diag. Rather, it must be estimated during the optimization process. In case of multivariate Gaussian distribution, the G function and its

first and second derivatives are

$$G(\hat{\boldsymbol{s}}_j) = \frac{1}{2}\log(\det(\boldsymbol{\Sigma}_j)) + \frac{1}{2}\hat{\boldsymbol{s}}_j^\top \boldsymbol{\Sigma}_j^{-1}\hat{\boldsymbol{s}}_j, \quad \frac{\partial G(\hat{\boldsymbol{s}}_j)}{\partial \hat{s}_j^{[d]}} = \hat{\boldsymbol{s}}_j^\intercal \boldsymbol{\Sigma}_j^{-1}\boldsymbol{e}_d, \quad \text{and} \quad \frac{\partial^2 G(\hat{\boldsymbol{s}}_j)}{\partial \hat{s}_j^{[d]}\partial(\hat{s}_j^{[d]})^\top} = \boldsymbol{\Sigma}_j^{-1}\boldsymbol{e}_d\boldsymbol{e}_d^\top,$$

where $\boldsymbol{e}_d$ stands for dth standard basis vector of D-dimensional space. One of the major advantages the multivariate Gaussian source model has, is that the expected values (8) and (9) can be calculated using only the estimated covariance matrices $\hat{\boldsymbol{R}}_x^{[d_1,d_2]}, d_1, d_2 = 1, ..., D$ of the observed signals and the estimated unmixing matrices. The calculation of the expected values becomes invariant of sample size T, which makes it much more viable than other source density models in terms of computation time. The covariance matrix

$$\boldsymbol{R}_x^{[d_1,d_2]} = E[\boldsymbol{x}^{[d_1]}(\boldsymbol{x}^{[d_2]})^\top]$$

is estimated in practice using

$$\hat{\boldsymbol{R}}_x^{[d_1,d_2]} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}^{[d_1]}(t)(\boldsymbol{x}^{[d_1]}(t))^\top.$$

Then, by observing that

$$\boldsymbol{\Sigma}_j^{[d_1,d_2]} = (\boldsymbol{w}_j^{[d_1]})^\top \boldsymbol{R}_x^{[d_1,d_2]}\boldsymbol{w}_j^{[d_2]} \text{ and } E[\boldsymbol{x}^{[d]}\hat{\boldsymbol{s}}_j^\top] = [\boldsymbol{R}_x^{[d,1]}\boldsymbol{w}_j^{[1]}, ..., \boldsymbol{R}_x^{[d,D]}\boldsymbol{w}_j^{[D]}],$$

the expected values (8) and (9) can be presented in forms of

$$E\left(\frac{\partial G(\hat{\mathbf{s}}_j)}{\partial \hat{\boldsymbol{s}}_j^{[d]}}\hat{\boldsymbol{x}}^{[d]}\right) = E\left(\boldsymbol{x}^{[d]}\frac{\partial G(\hat{\mathbf{s}}_j)}{\partial \hat{\boldsymbol{s}}_j^{[d]}}\right) = E\left(\boldsymbol{x}^{[d]}\hat{\boldsymbol{s}}_j^\top \boldsymbol{\Sigma}_j^{-1}\boldsymbol{e}_d\right)$$
$$= [\boldsymbol{R}_x^{[d,1]}\boldsymbol{w}_j^{[1]}, ..., \boldsymbol{R}_x^{[d,D]}\boldsymbol{w}_j^{[D]}]\boldsymbol{\Sigma}_j^{-1}\boldsymbol{e}_d \approx [\hat{\boldsymbol{R}}_x^{[d,1]}\boldsymbol{w}_j^{[1]}, ..., \hat{\boldsymbol{R}}_x^{[d,D]}\boldsymbol{w}_j^{[D]}]\hat{\boldsymbol{\Sigma}}_j^{-1}\boldsymbol{e}_d \text{ and}$$
$$E\left(\frac{\partial^2 G(\hat{\mathbf{s}}_j)}{\partial \hat{\boldsymbol{s}}_j^{[d]}\partial(\hat{\boldsymbol{s}}_j^{[d]})^\top}\boldsymbol{x}^{[d]}(\boldsymbol{x}^{[d]})^\top\right) = E\left(\boldsymbol{\Sigma}_j^{-1}\boldsymbol{e}_d\boldsymbol{e}_d^\top\boldsymbol{x}^{[d]}(\boldsymbol{x}^{[d]})^\top\right) = (\boldsymbol{\Sigma}_j^{[d,d]})^{-1}\boldsymbol{R}_x^{[d,d]} \approx (\hat{\boldsymbol{\Sigma}}_j^{[d,d]})^{-1}\hat{\boldsymbol{R}}_x^{[d,d]},$$

where the elements of the matrix $\hat{\boldsymbol{\Sigma}}_j$ are calculated as

$$\hat{\boldsymbol{\Sigma}}_j^{[d_1,d_2]} = (\boldsymbol{w}_j^{[d_1]})^\top \hat{\boldsymbol{R}}_x^{[d_1,d_2]}\boldsymbol{w}_j^{[d_2]}, \; d_1, \; d_2 = 1, ..., D. \tag{12}$$

Here the estimates of the expected values are invariant of the sample size T. The parameter

estimates $\hat{\mathbf{\Sigma}}_j$, $j = 1, ...P$, are calculated before each iteration.

The multivariate Gaussian distribution is suitable source density model for sources with higher second-order correlation such as group fMRI studies (Anderson et al., 2011). If second-order correlation is not present or is minimal, the multivariate Gaussian distribution is not suitable option for source density model.

Anderson et al. (2011) also proposed a method called IVA-GL, which is to initialize fastIVA with the solution of IVA-G. Their results show that IVA-GL gives better and more consistent solutions regardless of the amount of second-order correlation in source vectors.

Another possible source density model is multivariate Laplace distribution (11) without the diagonal covariance matrix restriction. The advantage of this is that it captures both second-order and higher-order dependence. Source density model is then

$$p_L(\hat{\mathbf{s}}_j|\mathbf{\Sigma}_j) \propto \exp\left(-\sqrt{\hat{\mathbf{s}}_j^\top \mathbf{\Sigma}_j^{-1}\hat{\mathbf{s}}_j}\right),$$

where $\mathbf{\Sigma}_j$ has to be estimated before each iteration as done in (12). The Newton update based IVA algorithm with non-diagonal Laplace distribution as source density model is called IVA-L in this thesis. For IVA-L, the G function and its first and second derivatives are

$$G(\hat{\boldsymbol{s}}_j) = \sqrt{\hat{\boldsymbol{s}}_j^\top \mathbf{\Sigma}_j^{-1}\hat{\boldsymbol{s}}_j},\ \frac{\partial G(\hat{\boldsymbol{s}}_j)}{\partial \hat{s}_j^{[d]}} = \frac{\hat{\boldsymbol{s}}_j \mathbf{\Sigma}_j^{-1}\boldsymbol{e}_d}{\sqrt{\hat{\boldsymbol{s}}_j^\top \mathbf{\Sigma}_j^{-1}\hat{\boldsymbol{s}}_j}} \text{ and}$$

$$\frac{\partial^2 G(\hat{\boldsymbol{s}}_j)}{\partial \hat{s}_j^{[d]} \partial (\hat{s}_j^{[d]})^\top} = (\hat{\boldsymbol{s}}_j^\top \mathbf{\Sigma}_j^{-1}\hat{\boldsymbol{s}}_j)^{-1/2}\mathbf{\Sigma}_j^{-1}\boldsymbol{e}_d\boldsymbol{e}_d^\top - (\hat{\boldsymbol{s}}_j^\top \mathbf{\Sigma}_j^{-1}\hat{\boldsymbol{s}}_j)^{-3/2}(\hat{\boldsymbol{s}}_j \mathbf{\Sigma}_j^{-1}\boldsymbol{e}_d)^2.$$

The Student's t source density model with identity covariance structure, proposed by Liang et al. (2013), takes the form of

$$p_S(\hat{\boldsymbol{s}}_j) \propto \left(1 + \frac{\hat{\mathbf{s}}_j^\top \hat{\mathbf{s}}_j}{\upsilon}\right)^{-(\upsilon + D/2)},$$

where υ is the degree of freedom and D is the total number of datasets. The logarithm of $p_S(\hat{\boldsymbol{s}}_j)$ is then divided by $(\upsilon + D/2)$, so that G function becomes $G(\hat{\boldsymbol{s}}_j) = \log(1 + \frac{\hat{\mathbf{s}}_j^\top \hat{\mathbf{s}}_j}{\upsilon})$. The

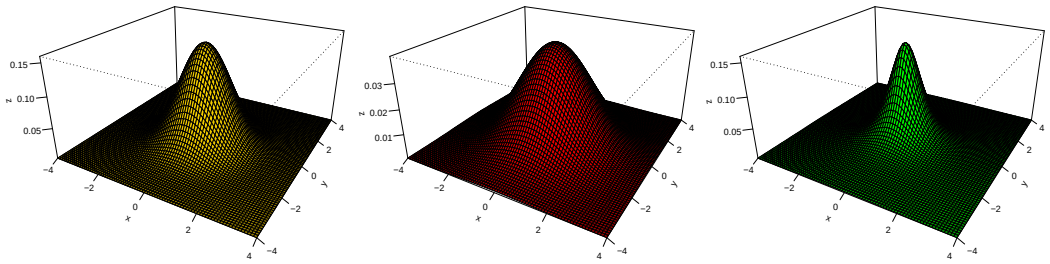

Figure 6: Density plots of bivariate Gaussian distribution (left), bivariate Laplace distribution (center) and bivariate Cauchy distribution (right), when the covariance matrix is identity matrix.

first and the second derivatives of G are then

$$\frac{\partial G(\hat{\boldsymbol{s}}_j)}{\partial \hat{s}_j^{[d]}} \propto \frac{\hat{s}_j^{[d]}}{1 + \frac{\hat{\boldsymbol{s}}_j^\top \hat{\boldsymbol{s}}_j}{v}} \quad \text{and} \quad \frac{\partial^2 G(\hat{\boldsymbol{s}}_j)}{\partial \hat{s}_j^{[d]} \partial (\hat{s}_j^{[d]})^\top} \propto \frac{1}{1 + \frac{\hat{\boldsymbol{s}}_j^\top \hat{\boldsymbol{s}}_j}{v}} + \frac{2(\hat{s}_j^{[d]})^2}{\left(1 + \frac{\hat{\boldsymbol{s}}_j^\top \hat{\boldsymbol{s}}_j}{v}\right)^2 v}.$$

The Student's t source density model presented here takes only higher-order dependence into account as the covariance structure is fixed to be an identity matrix. With decreasing parameter v, the distribution becomes more heavy-tailed. In this thesis only a single value $v = 1$ is considered rather than estimating it during the optimization process. With $v = 1$ the distribution is actually a multivariate Cauchy distribution and hence in this thesis the Newton update based IVA algorithm with the multivariate Cauchy distribution as source density model is called IVA-C.

Figure 6 represents the difference between the shapes of the multivariate Gaussian distribution, the multivariate Laplace distribution and the multivariate Cauchy distribution in bivariate case. The multivariate Laplace distribution is more heavy tailed than the multivariate Gaussian distribution and the multivariate Cauchy distribution has less probability mass near the center, but the tails lower towards zero much slower.

Many other source density models have been also proposed since the original IVA, such as mixture of Student's t distribution and Laplace distribution (Rafique et al., 2016), Gaussian mixture models (Hao et al., 2010) and multivariate generalized Gaussian distribution (Liang

et al., 2014). Xi-Lin (2020) proposed deep neural network (DNN) priors together with natural gradient descent optimizer for speech separation task. They estimated the derivative of density of speeches by recurrent neural networks and feed forward neural networks by training the networks using artificially mixed speeches of books read in English. This thesis focuses on studying the diagonal multivariate Laplace distribution, the non-diagonal multivariate Laplace distribution, the multivariate Gaussian distribution and the multivariate Cauchy distribution.

4 Simulation studies

Simulation studies are used to verify the performance of Newton based IVA algorithm with different source density models as well as to compare the source density models under different scenarios.

The multivariate Gaussian source density model captures only second-order dependence, the multivariate diagonal Laplace and Cauchy source density models capture only higher-order dependence, and the multivariate Laplace distribution with non-diagonal covariance captures both second-order and higher-order dependence. The aim of the simulations is to compare how well the source density models perform when the sources are generated from different distributions and the number of datasets and the number of observations change. The simulations are performed in R 4.1.1 (R Core Team, 2020) using the packages *doMPI* (Weston, 2017) and *LaplacesDemon* (Statisticat and LLC., 2021). All IVA algorithms and the performance indices used in the simulations are available in R package *ivaBSS* (Sipilä et al., 2022) which is made as a part of this thesis. The manual of *ivaBSS* is attached in the Appendix.

4.1 Performance metrics

Two different performance metrics are used for evaluating the performance of the algorithms. Intersymbol inference (ISI), often also called Amari index (Amari et al., 1996), is a performance measure commonly used in context of ICA. ISI is first introduced for blind source separation by Moreau and Macchi (1994). It is a function of the gain matrix $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{A}$, defined

as

$$ISI(\boldsymbol{G}) = \frac{\sum_{j=1}^{P}(\sum_{i=1}^{P}\frac{|g_{j,i}|}{\max_k|g_{j,k}|} - 1) + \sum_{i=1}^{P}(\sum_{j=1}^{P}\frac{|g_{j,i}|}{\max_k|g_{k,i}|} - 1)}{2P(P-1)},$$

where $|g_{j,i}|$ is the absolute value of element in the jth row and the ith column of the gain $\boldsymbol{G}$, $\max_k|g_{j,k}|$ is the maximum of the absolute values in the jth row of $\boldsymbol{G}$ and $\max_k|g_{k,i}|$ is the maximum of the absolute values of the ith column of $\boldsymbol{G}$. The metric is normalized based on the number of sources, P, so that $0 \leq ISI(\boldsymbol{G}) \leq 1$, where 0 is the optimal result. The maximum ISI occurs when $\boldsymbol{G}$ is a matrix of ones and the minimum ISI occurs when each row and each column of the matrix $\boldsymbol{G}$ has only one non-zero element. For any scalar $k \neq 0$ it holds that $ISI(k\boldsymbol{G}) = ISI(\boldsymbol{G})$.

For IVA, the extension of the intersymbol inference, joint intersymbol inference (joint ISI), (Anderson, 2013), has to be used as there are multiple datasets of source mixtures to be separated. Joint ISI calculates the average ISI over all gain matrices $\boldsymbol{G}^{[d]} = \boldsymbol{W}^{[d]}\boldsymbol{A}^{[d]}$ as well as penalizes if the source estimates are not aligned in same order for each dataset. The joint ISI is defined as

$$\text{jISI}(\boldsymbol{G}^{[1]}, ..., \boldsymbol{G}^{[D]}) = \text{ISI}\left(\frac{1}{D}\sum_{d=1}^{D}|\boldsymbol{G}^{[d]}|\right) = \text{ISI}\left(\sum_{d=1}^{D}|\boldsymbol{G}^{[d]}|\right),$$

where $|\boldsymbol{G}^{[d]}|$ denotes the matrix whose elements are the absolute values of the corresponding elements of $\boldsymbol{G}^{[d]}$. A small value of joint ISI indicates that the sources are separated well and the permutation of the source estimates is identical in each dataset. To measure the average separation performance across all datasets without taking the permutation in account, the average ISI can be used. The average ISI is calculated as

$$\text{avgISI}(\boldsymbol{G}^{[1]}, ..., \boldsymbol{G}^{[D]}) = \sum_{d=1}^{D}\frac{1}{D}\text{ISI}\left(\boldsymbol{G}^{[d]}\right).$$

If the joint ISI is close to zero, it indicates that the average ISI is also close to zero (Anderson, 2013). However, when the joint ISI is not close to zero, it is useful to calculate the average ISI to identify if the separation is achieved for each dataset separately.

To ease the interpretation of the mean joint ISI, the mean joint ISI is calculated for 10000 gain matrices $\boldsymbol{G}$, whose elements are generated from standard normal distribution, when the number of sources $P = 4$. The results are presented in Table 1. The values correspond

Table 1: The mean joint ISI of 10000 gain matrices, whose elements are generated from standard normal distribution for different numbers of datasets. All trials are computed using the number of sources $P = 4$.

The number of datasets	Mean jISI
2	0.530
5	0.660
10	0.740
20	0.805

to mean joint ISI of the case where the unmixing matrices $\boldsymbol{W}^{[d]}$ are identity matrices, i.e. $\hat{\boldsymbol{S}}^{[d]} = \boldsymbol{X}^{[d]}$ for $d = 1, ..., D$. The mean joint ISI should be compared to this limit rather than to the theoretical maximum value of 1.

Another metric that is used is the success ratio, which a proportion of the successful trials. A trial is considered successful, if the location of maximum absolute values of each row of $\boldsymbol{G}^{[d]}$ is unique within the dataset, but shared between the dataset $1, ..., D$ (Anderson, 2013). The first indicates that the sources are separated within dataset and the second indicates that the estimated sources are aligned in same order for each dataset. The success ratios are investigated to identify how consistent the algorithms are in different setups.

4.2 Generation of the data

The performance of each source density model is measured under four different source density distributions and multiple different simulation setups. The sources are generated separately from multivariate Laplace distribution, multivariate diagonal Laplace distribution, multivariate Gaussian distribution and multivariate power exponential distribution. In addition to these, the fifth setup is generated where each source is from different distribution. Each source density is investigated with number of datasets $D \in (2, 5, 10)$, the number of sources $P = 4$ and the number of observations $T \in (1000, 5000, 10000)$. Each setup is repeated 1000 times.

The sources following the multivariate Gaussian distribution and the multivariate Laplace distribution are generated with random covariance structures

$$\boldsymbol{\Sigma}_j = \boldsymbol{U}_j \boldsymbol{U}_j^\top, \; j = 1, ..., P, \tag{13}$$

where $\boldsymbol{U}_j$ is $D \times D$ matrix, whose elements are from standard normal distribution. For the

multivariate diagonal Laplace distribution, the identity covariance matrix is used, i.e. $\boldsymbol{\Sigma}_j = \boldsymbol{I}_D$.

The multivariate Gaussian distributed sources are then generated from the multivariate Gaussian distribution so that $\boldsymbol{s}_j \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_j)$. The multivariate Laplace distributed sources are generated using $\boldsymbol{s}_j = \sqrt{\boldsymbol{h}_j}\boldsymbol{z}_j$ (Anderson, 1992), where the elements of $\boldsymbol{h}_j$ are from exponential distribution with unity rate parameter and $\boldsymbol{z}_j \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_j)$. Here $\boldsymbol{\Sigma}_j$ is a random covariance matrix (13) for non-diagonal Laplace and identity matrix for diagonal Laplace.

The multivariate power exponential distribution (Gómez et al., 1998), also called the multivariate generalized Gaussian distribution, takes the form of

$$p(\boldsymbol{s}_j|\boldsymbol{\Sigma}_j) = \frac{D/2\Gamma(D/2)}{\sqrt{\pi^D \det\boldsymbol{\Sigma}_j}\Gamma(1 + D/(2\kappa))2^{D/(2\kappa)}} \exp\left(-\frac{1}{2}(\boldsymbol{s}_j^\top \boldsymbol{\Sigma}_j^{-1}\boldsymbol{s}_j)^\kappa\right),$$

when the expected value equals zero. Above $\boldsymbol{\Sigma}_j$ is the jth covariance matrix and $\kappa > 0$ effects on how heavy tailed the distribution is. The sources are generated with random $\kappa \in [0.25, 3]$, where smaller κ makes the distribution more heavy tailed. The multivariate Gaussian and the multivariate Laplace distributions are special cases of the multivariate power exponential distribution when $\kappa = 1$ and $\kappa = \frac{1}{2}$, respectively.

The fifth source density setup is with each source generated from different distribution. The sources are generated so that $\boldsymbol{s}_1$ follows the multivariate Gaussian distribution, $\boldsymbol{s}_2$ follows the multivariate Laplace distribution with random covariance structure $\boldsymbol{\Sigma}_2$, $\boldsymbol{s}_3$ follows the multivariate Laplace distribution with diagonal covariance structure $\boldsymbol{\Sigma}_3 = \boldsymbol{I}_D$ and $\boldsymbol{s}_4$ follows the multivariate Student's t distribution with degree of freedom $\upsilon = 4$ and diagonal covariance structure $\boldsymbol{\Sigma}_4 = \boldsymbol{I}_D$. In this setup the components of $\boldsymbol{s}_1$ are only second-order dependent, the components of $\boldsymbol{s}_2$ are both second-order and higher-order dependent and the components of $\boldsymbol{s}_3$ and $\boldsymbol{s}_4$ are only higher-order dependent.

The elements of the mixing matrices $\boldsymbol{A}^{[1]}, ..., \boldsymbol{A}^{[D]}$ are generated from the standard normal distribution, and the mixtures $\boldsymbol{X}^{[1]}, ..., \boldsymbol{X}^{[D]}$ are then obtained by calculating $\boldsymbol{X}^{[d]} = \boldsymbol{A}^{[d]}\boldsymbol{S}^{[d]}$.

4.3 Comparisions

The performance of different algorithms are compared by joint intersymbol-inference, success ratio and computation time in different simulation setups. Every simulation setup is repeated 1000 times and each generated data mixture is separated using each of the source density models discussed in Section 3.3. The joint ISI is calculated for each separation. The mean

joint ISI and the success ratios are then compared for the different simulation setups and source densities.

The initial simulations were performed using identity matrices as initial unmixing matrices. It was discovered that IVA-L, IVA-L-diag and IVA-C algorithms provided insufficient results by converging to local optima almost always, although they are the algorithms which should be theoretically able to separate the higher-order dependent sources. The main problem about these algorithms is that when the sources are higher-order dependent, they do not solve the common permutation problem, meaning that the source estimates might be aligned in different order in the datasets. Instead, they solve the BSS problem only separately for each dataset. When the sources are second-order and higher-order dependent, IVA-L solves the common permutation problem sufficiently exploiting the second-order dependence, but for IVA-L-diag and IVA-C the problem persists. This issue was identified by inspecting the joint ISI, which was usually large, and the average ISI, which was usually close to zero implying that the sources were separated successfully, but the common permutation was not discovered. To overcome this issue the goal is to initialize the unmixing matrices so that the algorithms converge to global optima as often as possible. According to Anderson (2013), whenever the sources possess any second-order dependence, it is beneficial to initialize the algorithms with the result of IVA-G, as IVA-G solves the common permutation problem reliably in most cases where the sources are second-order dependent. It also decreases the computation time, as IVA-G is hundreds of times faster than the other algorithms, which is discussed later this section. After initializing the algorithms with the estimated unmixing matrices of IVA-G, much fewer iterations are needed till convergence. According to previous simulation studies, when the sources are only higher-order dependent, the original IVA algorithm, fastIVA, with unmixing matrices restricted to be orthogonal (Lee et al., 2007) tend to converge to global optima more often. Finding the orthogonal unmixing matrices is usually sufficient if the sources are purely higher-order dependent, because the inverse mixing matrices are nearly orthogonal after whitening. FastIVA is most reliable when the number of sources is significantly less than the number of datasets (Anderson, 2013). When the number of sources increase or the number of datasets decrease, fastIVA converges more often to local optima. It still solves the common permutation problem more often than IVA-L, IVA-L-diag or IVA-C, which is why it is reasonable to initialize the algorithms with the result of fastIVA when the sources are only higher-order dependent.

In following simulation studies, fastIVA is initialized with the estimated unmixing matrices of IVA-G, and then IVA-L, IVA-L-diag and IVA-C are initialized with the estimated

unmixing matrices of fastIVA. With this procedure, IVA-G attempts to solve the permutation in case the sources possess second-order dependence and then fastIVA is used as a backup to obtain the best possible results when the sources are only higher-order dependent. The algorithms are run for maximum of 400 iterations or until the algorithm converged with the tolerance of 10^{-6}. From the trials where the source density models were theoretically able to capture the dependence between the datasets, less than 1% of the trials were not converged in 400 iterations. If the algorithm did not converge, the result of the last iteration was used.

The mean joint ISI of different setups are collected in Figure 7 and the success ratios in Figure 8. When the sources are generated from the multivariate Laplace distribution with random covariance structure, the sources are second-order dependent as well as higher-order dependent. In this setup the best algorithm is IVA-L, which is not surprising as the Laplace source density model matches the distribution of the true sources. When the number of datasets is two, the success ratio of IVA-G is only approximately 0.5 and joint ISI is over 0.1, but when the number of datasets grows, it performs nearly as well as the other algorithms. FastIVA, IVA-L-diag and IVA-C perform almost as well as IVA-L, especially when the number of datasets is small. IVA-L-diag achieves the smallest joint ISI of these three, and the joint ISI of fastIVA and IVA-C are almost identical. The common permutation is solved consistently for IVA-L, IVA-L-diag, IVA-C and fastIVA when using the previously introduced initialization of the unmixing matrices.

When the sources are generated from diagonal Laplace distribution, IVA-G is not able to separate the sources as the sources are purely higher-order dependent. The mean joint ISI is larger than the mean joint ISI of random gain matrices presented in Table 1 in all setups. IVA-L, IVA-L-diag, IVA-C and fastIVA provide similar results with decreasing joint ISI and increasing success ratio when the number of datasets grows. The mean joint ISI of IVA-L and IVA-L-diag are slightly smaller than the mean joint ISI of fastIVA and IVA-C, but the differences are almost unnoticeable.

When the sources are generated from the multivariate Gaussian distribution or from the multivariate power exponential distribution, the algorithms exploiting the second-order dependence, IVA-G and IVA-L, perform the best. IVA-G is the best algorithm when the number of datasets is small, although the success ratio is still only approximately 0.5. When the number of datasets grow, IVA-L is the best option producing lower joint ISI and higher success ratio. IVA-L-diag, IVA-C and fastIVA also provide reasonably good results, especially when the number of datasets and the number of observations are large. However, it should be noted that the performances of IVA-L-diag, IVA-C and fastIVA are mostly defined

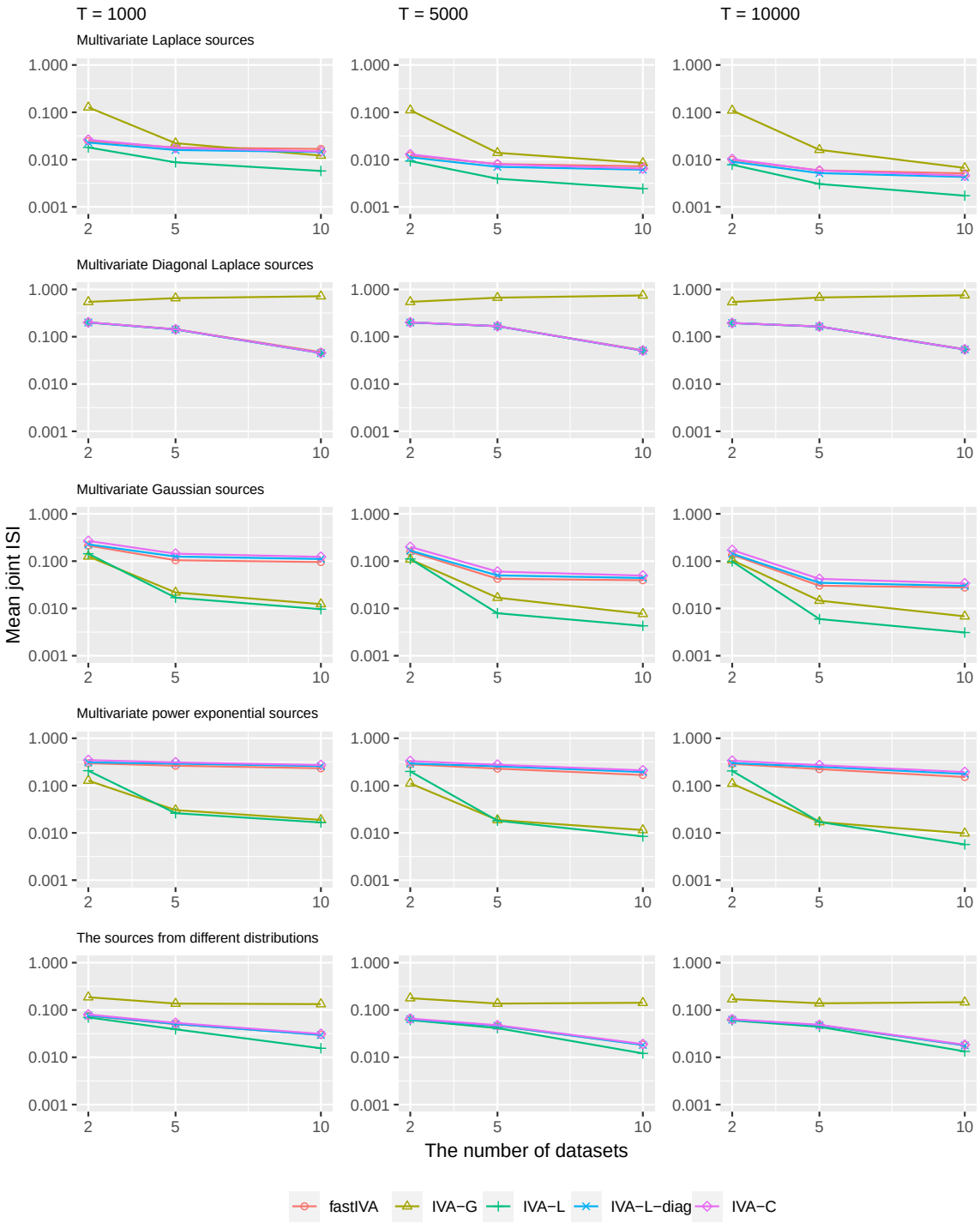

Figure 7: Mean joint intersymbol-inferences (log scale) of 1000 trials of each simulation setup. Different numbers of observations T are presented in columns and the number of datasets in horizontal axis. The true source density varies between the rows. Each generated data mixture is separated using fastIVA, IVA-G, IVA-L, IVA-L-diag and IVA-C algorithms.

by the initialization with the estimates of IVA-G in this setup.

In the fifth setup where the sources are from different distributions, IVA-L performs the best capturing both second-order and higher-order dependence. IVA-L-diag, IVA-C and fastIVA provide almost identical results losing only very little to IVA-L. IVA-G provides insufficient results, because there are two sources which are purely higher-order dependent in this setup. The success ratios of IVA-L, IVA-L-diag, IVA-C and fastIVA are almost identical, and growing when the number of datasets grows. The success ratio of IVA-G decreases close to zero when the number of datasets grows.

In general, the results get better when the number of datasets increases and when the number of observations increase. However, the effect of the number of datasets growing is much larger. The initialization of the unmixing matrices with the results of IVA-G and fastIVA solves the common permutation problem reasonably well, and increases the performances of IVA-L, IVA-L-diag and IVA-C. Especially IVA-L becomes very flexible and well performing algorithm in almost all setups. When the sources are purely higher-order dependent, the performances of IVA-L, IVA-L-diag and IVA-C are defined mostly by the performance of fastIVA, as it is used to initialize the algorithms. Even though the estimated unmixing matrices of fastIVA are restricted to be orthogonal, IVA-L, IVA-L-diag and IVA-C improve the results of fastIVA only marginally. This is most likely because of the fact that the mixing matrices of purely higher-order dependent sources become nearly orthogonal after the whitening.

The mean computation times per iteration are presented in Figure 9. The most important result in terms of computation time is that IVA-G algorithm is by far fastest of the algorithms. It is over 100 times faster than the IVA-L, IVA-L-diag and IVA-C when $T = 1000$, 400-550 times faster when $T = 5000$ and 600-800 times faster when $T = 10000$. FastIVA algorithm is approximately twice as fast as IVA-L, IVA-L-diag and IVA-C. The computation times of IVA-L, IVA-L-diag and IVA-C are close to each other, IVA-L being the slowest one. The number of iterations are higher when the source density model is a mismatch to true sources, otherwise there are no meaningful differences between the number of iterations. By initializing IVA-L, IVA-L-diag and IVA-C algorithms using the initialization procedure introduced previously usually lowers significantly the number of iterations required till convergence, which makes the overall computation time usually much lower.

In conclusion, IVA-L with the unmixing matrix initialization performs best in most cases and should be used especially when there are no prior knowledge about the true source densities. If the computation resources are limited and the sources possess at least some second-order dependence, IVA-G is the best choice as it is the fastest by far and performs

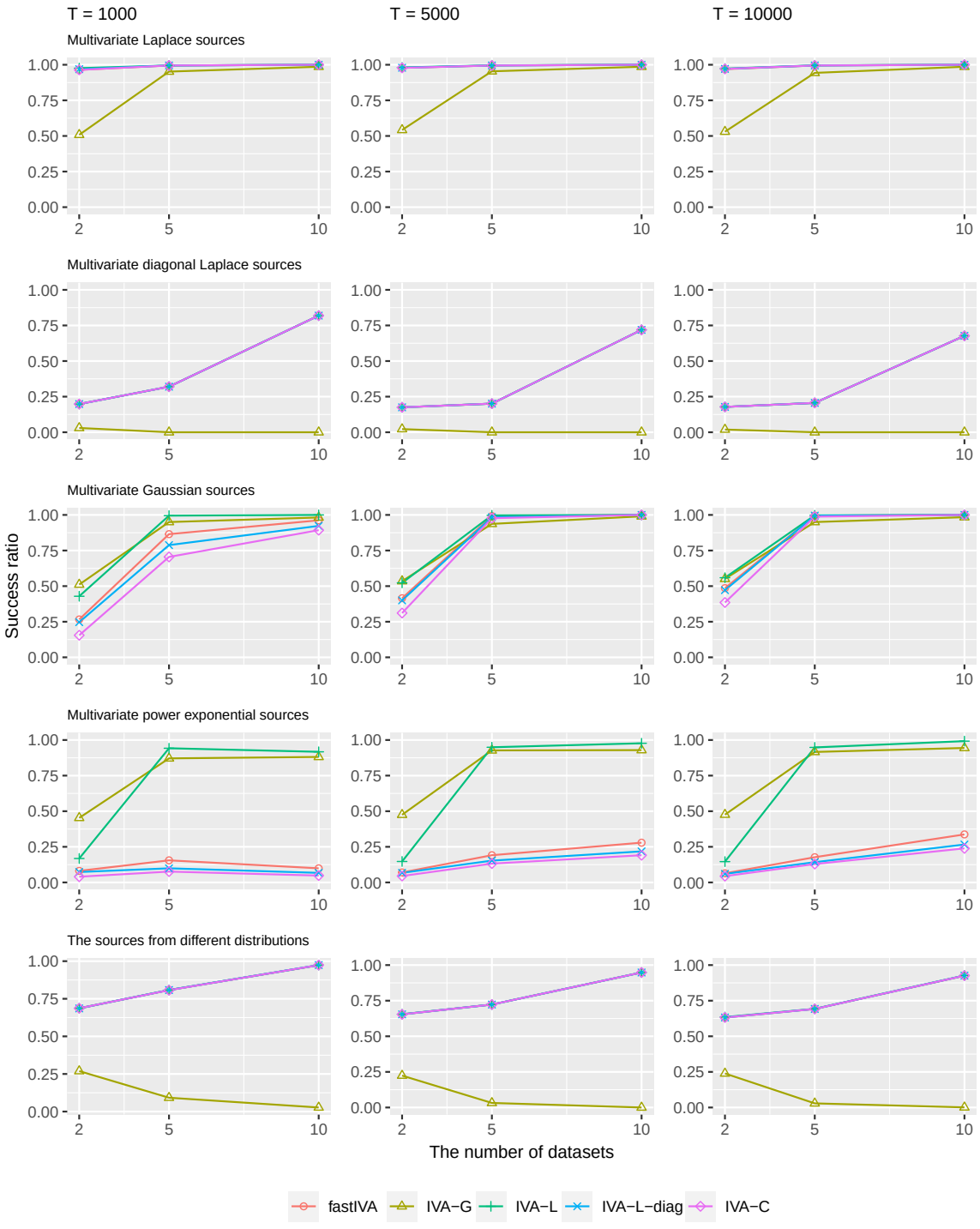

Figure 8: The success ratios of each simulation setup. Different numbers of observations T are presented in columns and the density of the true sources vary between the rows. Each simulation setup is repeated 1000 times and each generated data mixture is separated using fastIVA, IVA-G, IVA-L, IVA-L-diag and IVA-C algorithms.

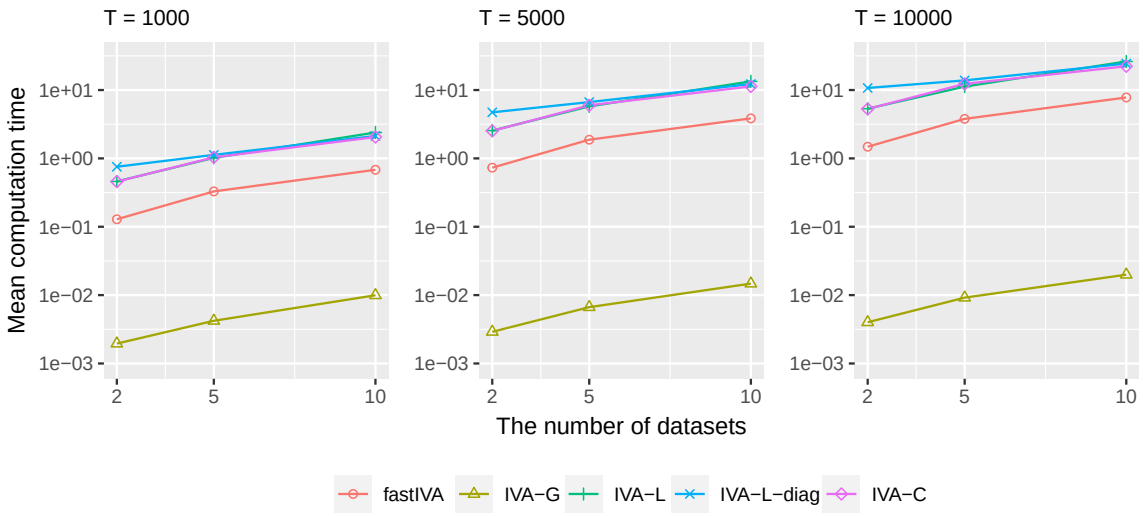

Figure 9: The mean computation times per iteration, when the sources are generated from multivariate power exponential distribution. Different number of observations T are presented in columns. Each simulation setup is repeated 1000 times and each generated data mixture is separated with each one of the algorithms.

nearly as well as IVA-L with the initialization. IVA-L-diag, IVA-C and fastIVA can be considered when the sources are mostly higher-order dependent. They perform best when the number of sources P is significantly less than the number of datasets D. For IVA-L, IVA-L-diag and IVA-C, the unmixing matrices should always be initialized with IVA-G when the sources possess second-order dependence, with fastIVA when the sources are only higher-order dependent, or with both IVA-G and fastIVA as done in these simulations when there are no prior knowledge about the true source densities.

5 Application to mixed images

Color images are often represented as RGB images, which consist of pixels with three channels, red (R), green (G) and blue (B). Each channel of a pixel has a brightness value between 0 and 255, where larger value means brighter color. In natural images the red, green and blue channels tend to be highly correlated, meaning that values of one channel are good predictors of the values of the other channels (Dufaux et al., 2016). The illustration of different channels presented in gray scale and the correlation between the channels is presented in Figure 10. When the gray scale representation of a single channel has a bright color, it means

that the image has the corresponding color stronger. For example the yellow car has red and green channels stronger than the blue channel, and the ceiling of the car has blue channel stronger than the other channels. It can be also seen that for example the white, black and brown areas of the original image tend to have similar values of red, green and blue channels.

In this section IVA-G, IVA-L-diag, IVA-L and IVA-C algorithms are used to separate five original images from five randomly mixed images. In this application, the source signals are composed of the $P = 5$ independent 100×100 pixel images. The rows of the images are stacked in a vector making the sample size $T = 10000$. As each pixel consists of three dependent elements, red, green and blue, the number of datasets equals $D = 3$. The mixed images are then obtained by generating the mixing matrices $\boldsymbol{A}^{[1]}$, $\boldsymbol{A}^{[2]}$ and $\boldsymbol{A}^{[3]}$ whose elements follow the standard normal distribution. The mixtures are then preprocessed by normalizing each channel of each mixture to range $[0, 255]$ to represent the values of the RGB channels. The normalization is done by calculating

$$\boldsymbol{x}_i^{[d]}(t) = 255 \cdot \frac{\boldsymbol{x}_i^{[d]}(t) - \min_t(\boldsymbol{x}_i^{[d]}(t))}{\max_t(\boldsymbol{x}_i^{[d]}(t)) - \min_t(\boldsymbol{x}_i^{[d]}(t))}, \; i = 1, ..., P, \; d = 1, ..., D, \; t = 1, ..., T. \quad (14)$$

The original images and the observed mixtures are presented in Figure 11.

Figure 10: RGB image of a yellow car and its red, green and blue channels represented in grayscale. The original picture is owned by Mika Sipilä.

Table 2: Joint intersymbol inferences of estimated unmixing matrices and true mixing matrices, when different algorithms are applied to mixed image separation task.

Algorithm	**Joint ISI**
IVA-L-diag	0.376
IVA-G	0.103
IVA-L	0.118
IVA-C	0.340

Figure 11: The original source images are presented in the first row and the mixtures of the source images in the second row. The original pictures are owned by Mika Sipilä.

The observed mixtures are separated using each of the algorithms, IVA-G, IVA-L-diag, IVA-L and IVA-C. IVA-L, IVA-L-diag and IVA-C algorithms are initialized with the estimated unmixing matrices of IVA-G to speed up the separation process. The initial values are obtained using the convergence tolerance of 10^{-6}. Then the convergence tolerance is set to 10^{-8} to obtain sufficient results. The results are postprocessed again by normalizing each channel of each recovered image to range $[0, 255]$, similarly as in (14). The received images are presented in Figure 12. As the algorithms are not capable of identifying the sign of the source signals, the images are having some of the channels in negative colors. The colors can be corrected manually by searching which channels should be multiplied by -1. By inspecting the results, it seems that the images are mostly second-order dependent as the IVA-G and IVA-L algorithms produced sufficient results, but the results of IVA-L-diag and IVA-C are not satisfying even though they are initialized with the results of IVA-G. This implies that there are no higher-order dependence structures to capture in these images. The joint ISI for the separations are presented in Table 2. In terms of joint ISI, IVA-G provided the best results and IVA-L the second best results. It should be noted that the joint ISI is higher than 0.1 for each result, which means that the algorithms are not optimal.

Finally, the manually corrected images separated by IVA-L and IVA-G are presented in Figure 13. The results look reasonably good taking into consideration that the algorithms are suboptimal for the image separation task. The algorithms introduced in this thesis do not take serial dependence into account, which is why the spatial correlation structures of the images are ignored. In addition, the probability distributions of the natural images are very complex and highly non-random (Ruderman, 1994), so the simple symmetric source density models

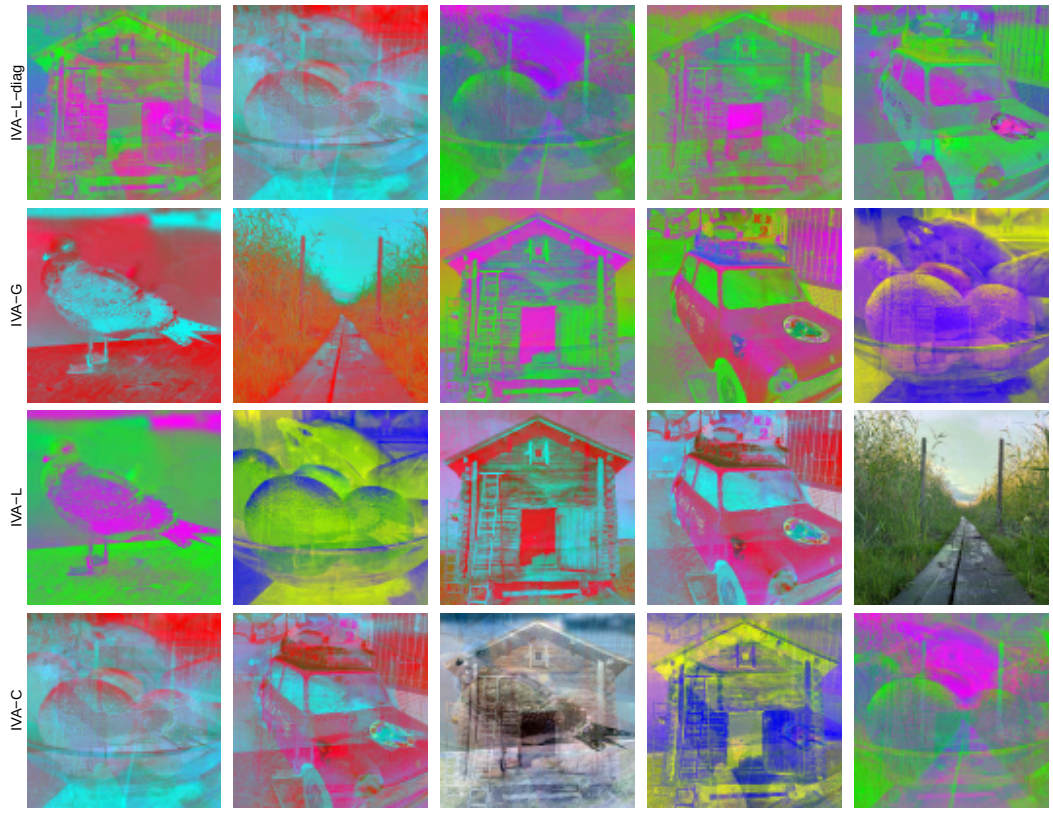

Figure 12: The separated images received by IVA-L-diag, IVA-G, IVA-L and IVA-C. Because the algorithms are not capable of identifing the sign of the true source signals, some color channels of the images are negative.

Figure 13: The separated images by IVA-G (top) and IVA-L (bottom), when the colors are manually corrected by multiplying certain channels of the source estimates by -1.

used in these algorithms do not have the best possible performance. In simulation studies it was also discovered that for mostly second-order dependent sources, the performances of the IVA-G and IVA-L are worse when the number of datasets is small, which is why the mixed image separation, with the number of datasets $D = 3$, is not the easiest task for the algorithms.

6 Conclusions and discussion

In this thesis the theory of ICA and IVA are presented, several IVA algorithms are compared using simulation studies and the algorithms are applied to mixed image separation task. The algorithms are implemented to R software as a part of the thesis and the documentation of the R package *ivaBSS* is attached in Appendix.

IVA has several advantages over ICA when multiple dependent datasets are studied jointly. If the observed mixtures are dependent across the datasets, IVA takes the dependence information into account yielding more accurate results as well as solves the permutation problem, meaning that the components of the estimated source vectors are aligned in same order for each vector. IVA has also more flexible identification condition allowing also Gaussian sources as long as there do not exist two or more $\boldsymbol{\alpha}$-Gaussian source vectors with proportional covariance matrices (Theorem 2).

The objective function was defined for iid data, and four Newton update based IVA algorithms with varying source density models were introduced. The algorithms, IVA-L, IVA-L-diag, IVA-G and IVA-C, were compared in multiple different simulation setups. Based on the initial simulations, IVA-L-diag, IVA-L and IVA-C algorithms converge usually to local optima and do not solve the common permutation problem between the datasets, meaning that the source estimates might be aligned in different order in each dataset. To overcome this issue and to decrease the computation time, the algorithms were initialized with the estimated unmixing matrices of IVA-G and the original IVA algorithm called fastIVA. The IVA-G was used to initialize fastIVA, which was then used to initialize the rest of the algorithms.

When the initialization is used, IVA-L algorithm is the most flexible and the most consistent algorithm in all setups. IVA-L takes both second-order and higher-order dependence into account, making it superior when the sources possess second-order and higher-order dependence, but still well performing when the sources are either second-order or higher-order dependent. The downside of IVA-L is that it is the slowest one of these algorithms, although the differences between IVA-L, IVA-L-diag and IVA-C are small. IVA-L and IVA-G algorithms are the best options when the higher-order dependence is minimal.

Because of the fact that using multivariate Gaussian source density model allows the Newton update to be done by calculating the gradient and the inverse of Hessian matrix using only the estimated mixing matrices and the observed mixtures rather than calculating sample means over the whole data, IVA-G stands out by its computation time. IVA-G was 400-800 times faster than the other algorithms in these simulation setups. This makes IVA-G much more viable when the sample size or the number of datasets is large. However, IVA-G is not able to recover sources when the datasets are only higher-order dependent, and thus it should not be considered when the second-order dependence is minimal. IVA-L-diag and IVA-C algorithms, capturing only the higher-order dependence, are viable options only when the source components are higher-order dependent. In general IVA-L is safer option and should work better whenever there are any second-order dependence between the sources. Even though fastIVA restricts the unmixing matrices to be orthogonal, IVA-L-diag and IVA-C improved the results of fastIVA only marginally in these simulations, which implies that searching only orthogonal unmixing matrices is sufficient for purely higher-order dependent sources.

Initializing IVA-L, IVA-L-diag and IVA-C algorithms with the estimated unmixing matrices of IVA-G and fastIVA solves the problem of converging to local optima in most cases, especially when the number of datasets is large. It also decreases the computation time and increases the performance, which is why the proper initialization should be done always. In general, whenever the algorithm is theoretically able to recover the sources, meaning that the algorithm is able to capture some dependence of the sources, the results get better when the number of datasets grows. The number of observations has smaller impact on the performances of the algorithms. The algorithms are more inconsistent when the number of datasets is small. Especially when the sources are purely higher-order dependent, the number of datasets should be significantly higher than the number of datasets to obtain consistent results.

In the thesis, the introduced IVA algorithms were also compared with mixed image separation task. Five real life colored natural images were mixed using random mixing matrices and then separated using each of the algorithms. In this application the algorithms accounting the second-order dependence, IVA-L and IVA-G, performed reasonably well, but the algorithms based on higher-order statistics, IVA-L-diag and IVA-C, did not provide recognizable images as results. The algorithms presented in this thesis are optimal for iid data only which is why the spatial correlation structures of the images are neglected making the results suboptimal.

Although the simulation studies of this thesis showed that the results are usually getting better with increasing number of datasets, this not the case in all situations. When the number of datasets or the number of sources grows large enough, the standard IVA algorithms, as presented here, suffer from a performance drop (Bhinge et al., 2019). To reduce the effect of high dimensionality, different methods have been developed such as various constrained IVA methods (Bhinge et al., 2020, 2019) and IVA methods together with independent subspace analysis (Silva et al., 2021).

Previously, there have been very few simulation studies comparing the performance of the IVA algorithms with different source density models under various setups. Anderson (2013) have compared fastIVA and IVA-G in a few different setups where the sources were either multivariate Laplace distributed or multivariate power exponentially distributed with κ randomly between [0.5,75] or [1.25,1.5]. In their study, IVA-G was outperforming fastIVA in every setup except for the ones where the number of datasets was two. They also compared IVA-G and fastIVA with IVA-G initialization when the sources were multivariate Laplace distributed, which is done also in this thesis. Their results showed that fastIVA with IVA-G initialization obtains more consistent results and outperforms IVA-G or the regular fastIVA almost always. These results are in line with the results of the simulation studies presented here.

The most of the other simulation studies that have been done previously concern some specific application, such as speech separation (Liang et al., 2014; Rafique et al., 2016; Liang et al., 2013) or EEG separation (Acharjee et al., 2015; Chen et al., 2017). In future more simulation studies are needed to investigate the properties and performance of IVA algorithms in various setups, such as when the sources are from asymmetric distribution or when the sources possess serial dependence. One interesting topic to investigate would be, how the amount of serial dependence affects the performance of the algorithms in different setups. It would be also important to derive the statistical properties, such as consistency and limiting distributions of the IVA algorithms presented here.

In general, currently existing algorithms are mostly developed for symmetric iid data. Algorithms for asymmetric data and data with serial dependence are limited. Currently only one BSS algorithm, that accounts serial dependence for multiple datasets, exists (Li et al., 2011). The algorithm is however limited to only orthogonal unmixing matrices and is thus suboptimal for many applications. For ICA, multiple algorithms accounting serial dependence exist, such as variants of second-order blind identification methods (Pan et al., 2021) or weight-adjusted second-order blind identification (Yeredor, 2000). These

are potential candidates to be extended to IVA. Extending IVA to account serial dependence would be highly beneficial, because in most of the applications the data is not iid.

The main contributions of this thesis was to use the simulation studies to provide new information of the performance of the IVA algorithms under different setups, and to implement IVA algorithms to R software for public use.

References

Acharjee, P. P., Phlypo, R., Wu, L., Calhoun, V. D., and Adalı, T. (2015). Independent vector analysis for gradient artifact removal in concurrent EEG-fMRI data. *IEEE Transactions on Biomedical Engineering*, 62:1750–1758.

Adalı, T., Anderson, M., and Fu, G. (2014). Diversity in independent component and vector analyses: Identifiability, algorithms, and applications in medical imaging. *IEEE Signal Processing Magazine*, 31:18–33.

Amari, S.-i., Cichocki, A., and Yang, H. (1996). A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems*, 8.

Anderson, D. N. (1992). A multivariate Linnik distribution. *Statistics & Probability Letters*, 14:333–336.

Anderson, M. (2013). *Independent vector analysis: Theory, algorithms, and applications*. PhD thesis, University of Maryland, Baltimore County.

Anderson, M., Adalı, T., and Li, X.-L. (2011). Joint blind source separation with multivariate Gaussian model: Algorithms and performance analysis. *IEEE Transactions on Signal Processing*, 60:1672–1683.

Anderson, M., Fu, G.-S., Phlypo, R., and Adalı, T. (2014). Independent vector analysis: Identification conditions and performance bounds. *IEEE Transactions on Signal Processing*, 62:4399–4410.

Anderson, M., Li, X.-L., and Adalı, T. (2010). Nonorthogonal independent vector analysis using multivariate Gaussian model. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 354–361. Springer.

Anderson, M., Li, X.-L., Rodriguez, P., and Adalı, T. (2012). An effective decoupling method for matrix optimization and its application to the ICA problem. In *2012 IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1885–1888. IEEE.

Arslan, O. (2010). An alternative multivariate skew Laplace distribution: Properties and estimation. *Statistical Papers*, 51:865–887.

Bhinge, S., Long, Q., Calhoun, V. D., and Adalı, T. (2020). Adaptive constrained independent vector analysis: An effective solution for analysis of large-scale medical imaging data. *IEEE Journal of Selected Topics in Signal Processing*, 14:1255–1264.

Bhinge, S., Mowakeaa, R., Calhoun, V. D., and Adalı, T. (2019). Extraction of time-varying spatiotemporal networks using parameter-tuned constrained IVA. *IEEE Transactions on Medical Imaging*, 38:1715–1725.

Chen, X., Peng, H., Yu, F., and Wang, K. (2017). Independent vector analysis applied to remove muscle artifacts in EEG data. *IEEE Transactions on Instrumentation and Measurement*, 66.

Comon, P. (1992). Independent component analysis. *Higher-Order Statistics*, pages 29–38.

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

Comon, P. and Jutten, C. (2010). *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press.

Dufaux, F., Le Callet, P., Mantiuk, R. K., and Mrak, M. (2016). Color management in HDR imaging. In *High Dynamic Range Video*, pages 237–272. Academic Press.

Gómez, E., Gomez-Viilegas, M., and Marín, J. M. (1998). A multivariate generalization of the power exponential family of distributions. *Communications in Statistics-Theory and Methods*, 27:589–600.

Hao, J., Lee, I., Lee, T.-W., and Sejnowski, T. J. (2010). Independent vector analysis for source separation using a mixture of Gaussians prior. *Neural Computation*, 22:1646–73.

Haykin, S. and Chen, Z. (2005). The cocktail party problem. *Neural Computation*, 17:1875–1902.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control. Wiley.

Hyvärinen, A. and Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks*, 13:411–430.

Ikeshita, R. and Nakatani, T. (2022). ISS2: An extension of iterative source steering algorithm for majorization-minimization-based independent vector analysis. *arXiv preprint arXiv:2202.00875*.

Ilmonen, P., Oja, H., and Serfling, R. (2012). On invariant coordinate system (ICS) functionals. *International Statistical Review*, 80:93–110.

Kim, T., Attias, H. T., Lee, S.-Y., and Lee, T.-W. (2006a). Blind source separation exploiting higher-order frequency dependencies. *IEEE Transactions on Audio, Speech, and Language Processing*, 15:70–79.

Kim, T., Eltoft, T., and Lee, T.-W. (2006b). Independent Vector Analysis: An Extension of ICA to Multivariate Components. In Rosca, J., Erdogmus, D., Príncipe, J. C., and Haykin, S., editors, *Independent Component Analysis and Blind Signal Separation*, pages 165–172. Springer.

Kim, T., Lee, I., and Lee, T.-W. (2006c). Independent vector analysis: Definition and algorithms. In *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, pages 1393–1396.

Kolmogorov, A. (1956). On the Shannon theory of information transmission in the case of continuous signals. *IRE Transactions on Information Theory*, 2:102–108.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22:79–86.

Lee, I., Hao, J., and Lee, T.-W. (2008a). Adaptive independent vector analysis for the separation of convoluted mixtures using EM algorithm. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 145–148. IEEE.

Lee, I., Kim, T., and Lee, T.-W. (2007). Fast fixed-point independent vector analysis algorithms for convolutive blind source separation. *Signal Processing*, 87:1859–1871.

Lee, J.-H., Lee, T.-W., Jolesz, F. A., and Yoo, S.-S. (2008b). Independent vector analysis (IVA): Multivariate approach for fMRI group study. *NeuroImage*, 40:86–109.

Li, X.-L., Adalı, T., and Anderson, M. (2011). Joint blind source separation by generalized joint diagonalization of cumulant matrices. *Signal Processing*, 91:2314–2322.

Liang, Y., Chen, G., Naqvi, S., and Chambers, J. A. (2013). Independent vector analysis with multivariate Student's t-distribution source prior for speech separation. *Electronics Letters*, 49:1035–1036.

Liang, Y., Harris, J., Naqvi, S. M., Chen, G., and Chambers, J. A. (2014). Independent vector analysis with a generalized multivariate Gaussian source prior for frequency domain blind source separation. *Signal Processing*, 105:175–184.

Long, Q., Bhinge, S., Calhoun, V. D., and Adalı, T. (2020). Independent vector analysis for common subspace analysis: Application to multi-subject fMRI data yields meaningful subgroups of schizophrenia. *NeuroImage*, 216:116872.

Ma, S., Calhoun, V. D., Phlypo, R., and Adalı, T. (2014). Dynamic changes of spatial functional network connectivity in healthy individuals and schizophrenia patients using independent vector analysis. *NeuroImage*, 90:196–206.

Moreau, E. and Macchi, O. (1994). A one stage self-adaptive algorithm for source separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3–49. IEEE.

Mowakeaa, R., Boukouvalas, Z., Long, Q., and Adalı, T. (2020). IVA using complex multivariate GGD: application to fMRI analysis. *Multidimensional Systems and Signal Processing*, 31:725–744.

Na, Y., Yu, J., and Chai, B. (2013). Independent vector analysis using subband and subspace nonlinearity. *EURASIP Journal on Advances in Signal Processing*, 2013:1–16.

Ono, N. (2011). Stable and fast update rules for independent vector analysis based on auxiliary function technique. In *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 189–192. IEEE.

Pan, Y., Matilainen, M., Taskinen, S., and Nordhausen, K. (2021). A review of second-order blind identification methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, page e1550.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rafique, W., Erateb, S., Naqvi, S. M., Dlay, S. S., and Chambers, J. A. (2016). Independent vector analysis for source separation using an energy driven mixed Student's t and super

Gaussian source prior. In *2016 24th European Signal Processing Conference*, pages 858–862. IEEE.

Ruderman, D. L. (1994). The statistics of natural images. *Network: Computation in Neural Systems*, 5:517–548.

Scheibler, R. and Ono, N. (2019). Independent vector analysis with more microphones than sources. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 185–189. IEEE.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

Silva, R., Plis, S., Adalı, T., Pattichis, M., and Calhoun, V. (2021). Multidataset independent subspace analysis with application to multimodal fusion. *IEEE Transactions on Image Processing*, 30:588–602.

Sipilä, M., Nordhausen, K., and Taskinen, S. (2022). *ivaBSS: Tools for Independent Vector Analysis*. R package version 1.0.0.

Statisticat and LLC. (2021). *LaplacesDemon: Complete Environment for Bayesian Inference*. R package version 16.1.6.

Weston, S. (2017). *doMPI: Foreach Parallel Adaptor for the Rmpi Package*. R package version 0.2.2.

Xi-Lin, L. (2020). Independent vector analysis with deep neural network source priors. *arXiv: Audio and Speech Processing*.

Yeredor, A. (2000). Blind separation of Gaussian sources via second-order statistics with asymptotically optimal weighting. *IEEE Signal Processing Letters*, 7:197–200.

Zhao, Q., Guo, F., Zu, X., Chang, Y., Li, B., and Yuan, X. (2017). An acoustic signal enhancement method based on independent vector analysis for moving target classification in the wild. *Sensors*, 17.

Appendix

Package 'ivaBSS'

May 20, 2022

Type Package

Title Tools for Independent Vector Analysis

Version 1.0.0

Date 2022-05-03

Imports stats, graphics, BSSprep

Suggests LaplacesDemon

Encoding UTF-8

Maintainer Mika Sipilä <mika.e.sipila@student.jyu.fi>

Description Independent vector analysis (IVA) is a blind source separation (BSS) model where several datasets are jointly unmixed. This package provides several methods for the unmixing together with some performance measures. For details, see Anderson et al. (2011) <doi:10.1109/TSP.2011.2181836> and Lee et al. (2007) <doi:10.1016/j.sigpro.2007.01.010>.

License GPL (>= 3)

LazyData true

RoxygenNote 7.1.2

R topics documented:

ivaBSS-package *Tools for Independent Vector Analysis*

Description

Independent vector analysis (IVA) is a blind source separation (BSS) model where several datasets are jointly unmixed. This package provides several methods for the unmixing together with some performance measures. For details, see Anderson et al. (2011) <doi:10.1109/TSP.2011.2181836> and Lee et al. (2007) <doi:10.1016/j.sigpro.2007.01.010>.

Details

The package contains tools for independent vector analysis. The main functions to perform IVA are `"IVANewton"` and `"fastIVA"`. `"NewtonIVA"` performs Newton update based IVA and `"fastIVA"` performs fixed-point iteration based IVA. Both of the algorithms have multiple options for source density models.

Author(s)

Authors: Mika Sipilä, Klaus Nordhausen, Sara Taskinen

Maintainer: Mika Sipilä

References

Anderson, M., Adalı, T., & Li, X.-L. (2011). Joint blind source separation with multivariate Gaussian model: Algorithms and performance analysis. IEEE Transactions on Signal Processing, 60, 1672–1683. <doi:10.1109/TSP.2011.2181836>

Anderson, M. (2013). Independent vector analysis: Theory, algorithms, and applications. PhD dissertation, University of Maryland, Baltimore County.

avg_ISI *Average Intersymbol Inference*

Description

Calculates the average intersymbol inference for two sets of matrices.

Usage

```
avg_ISI(W, A)
```

Arguments

`W`	Array of unmixing matrices with dimension `[P, P, D]`.
`A`	Array of true mixing matrices with dimension `[P, P, D]`.

Details

The function returns the average intersymbol inference for the set of estimated unmixing matrices and the set of true mixing matrices. The average ISI gets the value between 0 and 1, where 0 is the optimal result. The average ISI is calculated as the mean ISI over each dataset separately. The average ISI does not take the permutation of the estimated sources into account.

Value

Numeric value between 0 and 1, where 0 is the optimal result indicating that the sources are separated perfectly in each dataset.

Author(s)

Mika Sipilä

References

Anderson, M. (2013). Independent vector analysis: Theory, algorithms, and applications. PhD dissertation, University of Maryland, Baltimore County.

See Also

joint_ISI, jbss_achieved

Examples

```
# Mixing matrices and unmixing matrices generated
# from standard normal distribution
P <- 4; D <- 4;
W <- array(rnorm(P * P * D), c(P, P, D))
A <- array(rnorm(P * P * D), c(P, P, D))

avg_ISI(W, A)

if (require("LaplacesDemon")) {
  # Generate sources from multivariate Laplace distribution
  P <- 4; N <- 1000; D <- 4;
  S <- array(NA, c(P, N, D))

  for (i in 1:P) {
    U <- array(rnorm(D * D), c(D, D))
    Sigma <- crossprod(U)
    S[i, , ] <- rmvl(N, rep(0, D), Sigma)
  }

  # Generate mixing matrices from standard normal distribution
  A <- array(rnorm(P * P * D), c(P, P, D))

  # Generate mixtures
  X <- array(NaN, c(P, N, D))
  for (d in 1:D) {
    X[, , d] <- A[, , d] %*% S[, , d]
  }

  # Estimate sources and unmixing matrices
```

```
  res_G <- NewtonIVA(X, source_density = "gaussian")
  avg_ISI(coef(res_G), A)
}
```

`coef.iva` *Coefficient of the Object of Class iva*

Description

`coef` method for class `"iva"`.

Usage

```
## S3 method for class 'iva'
coef(object, which.dataset = NA, ...)
```

Arguments

`object`	an object of class `"iva"`, usually the result of a call to `NewtonIVA` or `fastIVA`.
`which.dataset`	positive integer. Provides the index in case the unmixing matrix only for a specific data set is desired. Default is to return all unmixing matrices.
`...`	further arguments are not used.

Details

Returns the unmixing matrices for all datasets or only for the requested dataset.

Value

Unmixing matrix or all unmixing matrices of the object of class `"iva"`. If a single unmixing matrix is requested, it is an array with dimension `[P, P]` and if all unmixing matrices are requested, it is an array with dimension `[P, P, D]`.

Author(s)

Mika Sipilä

See Also

`NewtonIVA`, `fastIVA`

Examples

```
if (require("LaplacesDemon")) {
  # Generate sources from multivariate Laplace distribution
  P <- 4; N <- 1000; D <- 4;
  S <- array(NA, c(P, N, D))

  for (i in 1:P) {
    U <- array(rnorm(D * D), c(D, D))
    Sigma <- crossprod(U)
    S[i, , ] <- rmvl(N, rep(0, D), Sigma)
  }
```

```
  # Generate mixing matrices from standard normal distribution
  A <- array(rnorm(P * P * D), c(P, P, D))

  # Generate mixtures
  X <- array(NaN, c(P, N, D))
  for (d in 1:D) {
    X[, , d] <- A[, , d] %*% S[, , d]
  }

  # Estimate sources and unmixing matrices
  res_G <- NewtonIVA(X, source_density = "gaussian")


  # All D unmixing matrices
  coef(res_G)

  # The unmixing matrix for the second dataset
  coef(res_G, 2)
}
```

`components.iva` *Components of the Object of Class iva*

Description

Returns the estimated source components of object of class `"iva"`.

Usage

```
components.iva(object, which.dataset = NA, ...)
```

Arguments

`object`	an object of class `"iva"`, usually the result of a call to `NewtonIVA` or `fastIVA`.
`which.dataset`	positive integer. Provides the index in case the unmixing matrix only for a specific data set is desired. Default is to return all unmixing matrices.
`...`	further arguments are not used.

Details

Returns the estimated source components for all datasets or only for the requested dataset.

Value

Estimated source components for requested dataset or for all datasets of the object of class `"iva"`. If a single dataset is requested, it is an array with dimension `[P, N]` and if all datasets are requested, it is an array with dimension `[P, N, D]`.

Author(s)

Mika Sipilä

See Also

NewtonIVA, fastIVA

Examples

```
if (require("LaplacesDemon")) {
  # Generate sources from multivariate Laplace distribution
  P <- 4; N <- 1000; D <- 4;
  S <- array(NA, c(P, N, D))

  for (i in 1:P) {
    U <- array(rnorm(D * D), c(D, D))
    Sigma <- crossprod(U)
    S[i, , ] <- rmvl(N, rep(0, D), Sigma)
  }

  # Generate mixing matrices from standard normal distribution
  A <- array(rnorm(P * P * D), c(P, P, D))

  # Generate mixtures
  X <- array(NaN, c(P, N, D))
  for (d in 1:D) {
    X[, , d] <- A[, , d] %*% S[, , d]
  }

  # Estimate sources and unmixing matrices
  res_G <- NewtonIVA(X, source_density = "gaussian")


  # Source estimates for all D datasets
  components.iva(res_G)

  # Source estimates for the second dataset
  components.iva(res_G, 2)
}
```

`fastIVA` *Fast Fixed-point IVA Algorithm*

Description

The algorithm estimates the sources from multiple dependent datasets jointly using their observed mixtures. The estimation is done by maximizing the independence between the sources, when the estimated unmixing matrices are restricted to be orthogonal. The options for different source densities are provided.

Usage

```
fastIVA(X, source_density="laplace_diag", student_df=1,
max_iter = 1024, eps = 1e-6, W_init = NA, verbose = FALSE)
```

Arguments

X	numeric data array containing the observed mixtures with dimension [P, N, D], where P is the dimension of the observed dataset, N is the number of the observations and D is the number of the datasets. The number of datasets D should be at least 2. Missing values are not allowed.
source_density	string to determine which source density model should be used. The options are "laplace_diag", "student" or "entropic". For more information see the details section.
student_df	integer. The degree of freedom for multivariate Student's distribution. Used only if source_denisty = "student".
max_iter	positive integer, used to define the maximum number of iterations for algorithm to run. If max_iter is reached, the unmixing matrices of the last iteration are used.
eps	convergence tolerance, when the convergence measure is smaller than eps, the algorithm stops.
W_init	numeric array of dimension [P, P, D] containing initial unmixing matrices. If not set, initialized with identity matrices.
verbose	logical. If TRUE the convergence measure is printed during the learning process.

Details

The algorithm uses fixed-point iteration to estimate to estimate the multivariate source signals from their observed mixtures. The elements of the source signals, or the datasets, should be dependent of each other to achieve the estimates where the sources are aligned in same order for each dataset. If the datasets are not dependent, the sources can still be separated but not necessarily aligned. This algorithm restricts the estimates unmixing matrices to be orthogonal. For more of the fast fixed-point IVA algorithm, see Lee, I. et al (2007).

The source density model should be selected to match the density of the true source signals. When source_density = "laplace_diag", the multivariate Laplace source density model with diagonal covariance structure is used. When source_density = "entropic", the approximated entropy based source density model is used. For more about multivariate Laplace and entropic source density models, see Lee, I. et al (2007). When source_density = "student" the multivariate Student's source density model is used, for more see Liang, Y. et al (2013).

The algorithm assumes that observed signals are multivariate, i.e. the number of datasets D >= 2. The estimated signals are zero mean and scaled to unit variance.

Value

An object of class "iva".

S	The estimated source signals with dimension [P, N, D]. The estimated source signals are zero mean with unit variance.
W	The estimated unmixing matrices with dimension [P, P, D].
W_whitened	The estimated unmixing matrices with dimension [P, P, D] for whitened data.
V	The whitening matrices with dimension [P, P, D].
X_means	The means for each observed mixture with dimension [P, D].
niter	The number of iterations that the algorithm did run.
converged	Logical value which tells if the algorithm converged.

source_density	The source density model used.
N	The number of observations.
D	The number of datasets.
P	The number of sources.
student_df	The degree of freedom for Student's source density model.
call	The function call.
DNAME	The name of the variable containing the observed mixtures.

Author(s)

Mika Sipilä

References

Lee, I., Kim, T., & Lee, T.-W. (2007). Fast fixed-point independent vector analysis algorithms for convolutive blind source separation. Signal Processing, 87, 1859–1871. <doi:10.1016/j.sigpro.2007.01.010>

Liang, Y., Chen, G., Naqvi, S., & Chambers, J. A. (2013). Independent vector analysis with multivariate Student's t-distribution source prior for speech separation. Electronics Letters, 49, 1035–1036. <doi:10.1049/el.2013.1999>

See Also

NewtonIVA

Examples

```
if (require("LaplacesDemon")) {
  # Generate sources from multivariate Laplace distribution
  P <- 2; N <- 1000; D <- 5;
  S <- array(NA, c(P, N, D))

  for (i in 1:P) {
    S[i, , ] <- rmvl(N, rep(0, D), diag(D))
  }

  # Generate mixing matrices from standard normal distribution
  A <- array(rnorm(P * P * D), c(P, P, D))

  # Generate mixtures
  X <- array(NaN, c(P, N, D))
  for (d in 1:D) {
    X[, , d] <- A[, , d] %*% S[, , d]
  }

  # Estimate sources and unmixing matrices
  res <- fastIVA(X)
}
```

jbss_achieved *JBSS Achieved*

Description

The function calculates if the joint blind source separation (JBSS) is achieved.

Usage

```
jbss_achieved(W, A)
```

Arguments

W	Array of unmixing matrices with dimension [P, P, D].
A	Array of true mixing matrices with dimension [P, P, D].

Details

The function calculates if the joint blind source separation is achieved. JBSS is considered achieved when the the location of maximum absolute values of each row of gain matrix G[,,d] = W[,,d] %*% A[,,d] is unique within the dataset, but shared between the datasets 1, ...,D. The first indicates that the sources are separated within dataset and the second indicates that the estimated sources are aligned in same order for each dataset.

Value

Logical. If TRUE the JBSS is considered achieved.

Author(s)

Mika Sipilä

References

Anderson, M. (2013). Independent vector analysis: Theory, algorithms, and applications. PhD dissertation, University of Maryland, Baltimore County.

See Also

joint_ISI, avg_ISI

Examples

```
# Mixing matrices and unmixing matrices generated
# from standard normal distribution
P <- 4; D <- 4;
W <- array(rnorm(P * P * D), c(P, P, D))
A <- array(rnorm(P * P * D), c(P, P, D))

jbss_achieved(W, A)

if (require("LaplacesDemon")) {
  # Generate sources from multivariate Laplace distribution
```

```
  P <- 4; N <- 1000; D <- 4;
  S <- array(NA, c(P, N, D))

  for (i in 1:P) {
    U <- array(rnorm(D * D), c(D, D))
    Sigma <- crossprod(U)
    S[i, , ] <- rmvl(N, rep(0, D), Sigma)
  }

  # Generate mixing matrices from standard normal distribution
  A <- array(rnorm(P * P * D), c(P, P, D))

  # Generate mixtures
  X <- array(NaN, c(P, N, D))
  for (d in 1:D) {
    X[, , d] <- A[, , d] %*% S[, , d]
  }

  # Estimate sources and unmixing matrices
  res_G <- NewtonIVA(X, source_density = "gaussian")
  jbss_achieved(coef(res_G), A)
}
```

`joint_ISI` *Joint Intersymbol Inference*

Description

Calculates the joint intersymbol inference for two sets of matrices.

Usage

```
joint_ISI(W, A)
```

Arguments

`W`	Array of unmixing matrices with dimension `[P, P, D]`.
`A`	Array of true mixing matrices with dimension `[P, P, D]`.

Details

The function returns the joint intersymbol inference for the set of estimated unmixing matrices and the set of true mixing matrices. The joint ISI gets the value between 0 and 1, where 0 is the optimal result. The joint ISI calculates the average intersymbol inference over each dataset as well as penalizes if the sources are not aligned in same order for each dataset.

Value

Numeric value between 0 and 1, where 0 is the optimal result indicating that the sources are separated perfectly and aligned in same order in each dataset.

Author(s)

Mika Sipilä

References

Anderson, M. (2013). Independent vector analysis: Theory, algorithms, and applications. PhD dissertation, University of Maryland, Baltimore County.

See Also

avg_ISI, jbss_achieved

Examples

```
# Mixing matrices and unmixing matrices generated
# from standard normal distribution
P <- 4; D <- 4;
W <- array(rnorm(P * P * D), c(P, P, D))
A <- array(rnorm(P * P * D), c(P, P, D))

joint_ISI(W, A)

if (require("LaplacesDemon")) {
  # Generate sources from multivariate Laplace distribution
  P <- 4; N <- 1000; D <- 4;
  S <- array(NA, c(P, N, D))

  for (i in 1:P) {
    U <- array(rnorm(D * D), c(D, D))
    Sigma <- crossprod(U)
    S[i, , ] <- rmvl(N, rep(0, D), Sigma)
  }

  # Generate mixing matrices from standard normal distribution
  A <- array(rnorm(P * P * D), c(P, P, D))

  # Generate mixtures
  X <- array(NaN, c(P, N, D))
  for (d in 1:D) {
    X[, , d] <- A[, , d] %*% S[, , d]
  }

  # Estimate sources and unmixing matrices
  res_G <- NewtonIVA(X, source_density = "gaussian")
  joint_ISI(coef(res_G), A)
}
```

NewtonIVA *Newton Update Based IVA Algorithm*

Description

The algorithm estimates the sources from multiple dependent datasets jointly using their observed mixtures. The estimation is done by maximizing the independence between the sources. The options for different source densities are provided.

Usage

```
NewtonIVA(X, source_density="laplace", student_df=1,
init = "default", max_iter = 1024, eps = 1e-6, W_init = NA,
step_size=1, step_size_min = 0.1, alpha = 0.9, verbose = FALSE)
```

Arguments

X	numeric data array containing the observed mixtures with dimension [P, N, D], where P is the dimension of the observed dataset, N is the number of the observations and D is the number of the datasets. The number of datasets D should be at least 2. Missing values are not allowed.
source_density	string to determine which source density model should be used. The options are "laplace", "laplace_diag", "gaussian" or "student". For more information see the details section.
student_df	integer. The degree of freedom for multivariate Student's distribution. Used only if source_denisty = "student".
init	string, to determine how to initialize the algorithm. The options are "default", "IVA-G+fastIVA", "IVA-G", "fastIVA" or "none". For more information see the details section.
max_iter	positive integer, used to define the maximum number of iterations for algorithm to run. If max_iter is reached, the unmixing matrices of the last iteration are used.
eps	convergence tolerance, when the convergence measure is smaller than eps, the algorithm stops.
W_init	numeric array of dimension [P, P, D] containing initial unmixing matrices. If not set, initialized with identity matrices.
step_size	initial step size for Newton step, should be between 0 and 1, default is 1.
step_size_min	the minimum step size.
alpha	multiplier for how much to decrease step size when convergence is not getting smaller.
verbose	logical. If TRUE the convergence measure is printed during the learning process.

Details

The algorithm uses Newton update together with decoupling trick to estimate the multivariate source signals from their observed mixtures. The elements of the source signals, or the datasets, should be dependent of each other to achieve the estimates where the sources are aligned in same order for each dataset. If the datasets are not dependent, the sources can still be separated but not necessarily aligned. The algorithm does not assume the unmixing matrices to be orthogonal. For more of the nonorthogonal Newton update based IVA algorithm, see Anderson, M. et al (2011) and Anderson, M. (2013).

The source density model should be selected to match the density of the true source signals. When source_density = "laplace", the multivariate Laplace source density model is used. This is the most flexible choice as it takes both second-order and higher-order dependence into account.

When source_density = "laplace_diag", the multivariate Laplace source density model with diagonal covariance structure is used. Multivariate diagonal Laplace source density model should be considered only when the sources are mainly higher-order dependent. It works best when the number of sources is significantly less than the number of datasets.

When source_density = "gaussian" the multivariate Gaussian source density model is used. This is the superior choice in terms of computation power and should be used when the sources are mostly second-order dependent.

When source_density = "student" the multivariate Student's source density model is used. Multivariate Student's source density model should be considered only when the sources are mainly higher-order dependent. It works best when the number of sources is significantly less than the number of datasets.

The init parameter defines how the algorithm is initialized. When init = "default", the default initialization is used. As default the algorithm is initialized using init = "IVA-G+fastIVA" when source_density is "laplace", "laplace_diag" or "student", and using init = "none" when source_density = "gaussian".

When init = "IVA-G+fastIVA", the algorithm is initialized using first the estimated unmixing matrices of IVA-G, which is NewtonIVA with source_density = "gaussian", to initialize fastIVA algorithm. Then the estimated unmixing matrices W of fastIVA are used as initial unmixing matrices for NewtonIVA. IVA-G is used to solve the permutation problem of aligning the source estimates when ever the true sources are second-order dependent. If the true sources are not second-order dependent, fastIVA is used as backup as it solves the permutation problem more regularly than NewtonIVA when the sources are purely higher-order dependent. When the sources possess any second-order dependence, IVA-G also speeds the computation time up a lot. This option should be used whenever there is no prior information about the sources and source_density is either "laplace", "laplace_diag" or "student".

When init = "IVA-G", the estimated unmixing matrices of IVA-G are used to initialize this algorithm. This option should be used if the true sources are expected to possess any second-order dependence and source_density is not "gaussian".

When init = "fastIVA", the estimated unmixing matrices of fastIVA algorithm is used to initialize this algorithm. This option should be used if the true sources are expected to possess only higher-order dependence. For more details, see fastIVA.

When init = "none", the unmixing matrices are initialized randomly from standard normal distribution.

The algorithm assumes that observed signals are multivariate, i.e. the number of datasets D >= 2. The estimated signals are zero mean and scaled to unit variance.

Value

An object of class "iva".

S	The estimated source signals with dimension [P, N, D]. The estimated source signals are zero mean with unit variance.
W	The estimated unmixing matrices with dimension [P, P, D].
W_whitened	The estimated unmixing matrices with dimension [P, P, D] for whitened data.
V	The whitening matrices with dimension [P, P, D].
X_means	The means for each observed mixture with dimension [P, D].
niter	The number of iterations that the algorithm did run.
converged	Logical value which tells if the algorithm converged.
source_density	The source density model used.
N	The number of observations.
D	The number of datasets.
P	The number of sources.

student_df	The degree of freedom for Student's source density model.
call	The function call.
DNAME	The name of the variable containing the observed mixtures.

Author(s)

Mika Sipilä

References

Anderson, M., Adalı, T., & Li, X.-L. (2011). Joint blind source separation with multivariate Gaussian model: Algorithms and performance analysis. IEEE Transactions on Signal Processing, 60, 1672–1683. <doi:10.1109/TSP.2011.2181836>

Anderson, M. (2013). Independent vector analysis: Theory, algorithms, and applications. PhD dissertation, University of Maryland, Baltimore County.

Liang, Y., Chen, G., Naqvi, S., & Chambers, J. A. (2013). Independent vector analysis with multivariate Student's t-distribution source prior for speech separation. Electronics Letters, 49, 1035–1036. <doi:10.1049/el.2013.1999>

See Also

fastIVA

Examples

```
if (require("LaplacesDemon")) {
  # Generate sources from multivariate Laplace distribution
  P <- 4; N <- 1000; D <- 4;
  S <- array(NA, c(P, N, D))

  for (i in 1:P) {
    U <- array(rnorm(D * D), c(D, D))
    Sigma <- crossprod(U)
    S[i, , ] <- rmvl(N, rep(0, D), Sigma)
  }

  # Generate mixing matrices from standard normal distribution
  A <- array(rnorm(P * P * D), c(P, P, D))

  # Generate mixtures
  X <- array(NaN, c(P, N, D))
  for (d in 1:D) {
    X[, , d] <- A[, , d] %*% S[, , d]
  }

  # Estimate sources and unmixing matrices
  res_G <- NewtonIVA(X, source_density = "gaussian")
}
```

plot.iva *Plotting an Object of Class iva*

Description

plot method for the class "iva".

Usage

```
## S3 method for class 'iva'
plot(x, which.dataset = NA, which.source = NA,
type = "l", xlabs = c(), ylabs = c(), colors = c(),
oma = c(1, 1, 0, 0), mar = c(2, 2, 1, 1), ...)
```

Arguments

x	An object of class "iva", usually the result of a call to NewtonIVA or fastIVA.
which.dataset	Positive integer to determine which dataset is returned. If not set, returns all datasets.
which.source	Positive integer to determine which dataset is returned. If not set, returns all datasets.
type	1-character string giving the type of plot desired. For details, see plot.
xlabs	Vector containing the labels for x-axis.
ylabs	Vector containing the labels for y-axis.
colors	Vector containing the colors for each plot.
oma	A vector of the form c(bottom, left, top, right) giving the size of the outer margins in lines of text. For more details, see par.
mar	A numerical vector of the form *c(bottom, left, top, right)* which gives the number of lines of margin to be specified on the four sides of the plot. For more details, see par.
...	Further arguments passed to plot function.

Details

Plots either all estimated sources of the object of class "iva" or the estimates for specific dataset and/or source.

Value

No return value, called for plotting the estimated sources of the object of class "iva".

Author(s)

Mika Sipilä

See Also

NewtonIVA, fastIVA

Examples

```
if (require("LaplacesDemon")) {
  # Generate sources from multivariate Laplace distribution
  P <- 4; N <- 1000; D <- 4;
  S <- array(NA, c(P, N, D))

  for (i in 1:P) {
    U <- array(rnorm(D * D), c(D, D))
    Sigma <- crossprod(U)
    S[i, , ] <- rmvl(N, rep(0, D), Sigma)
  }

  # Generate mixing matrices from standard normal distribution
  A <- array(rnorm(P * P * D), c(P, P, D))

  # Generate mixtures
  X <- array(NaN, c(P, N, D))
  for (d in 1:D) {
    X[, , d] <- A[, , d] %*% S[, , d]
  }

  # Estimate sources and unmixing matrices
  res_G <- NewtonIVA(X, source_density = "gaussian")

  # Plot all estimated sources
  plot(res_G)

  # Plot the source estimates for the first dataset only
  plot(res_G, which.dataset = 1)

  # Plot the source estimates for the second source only
  plot(res_G, which.source = 2)

  # Plot the source estimate of the second dataset and third source
  plot(res_G, which.dataset = 2, which.source = 3, type = "p")

  # Plot all source estimates with custom colors and labels
  plot(res_G, col=c(rep(1, 4), rep(2, 4), rep(3, 4), rep(4, 4)),
      xlabs = c("Subject 1", "Subject 2", "Subject 3", "Subject 4"),
      ylabs = c("Channel 1", "Channel 2", "Channel 3", "Channel 4"))
}
```

`predict.iva` *Predict Method for Object of Class iva*

Description

Predict the new source estimates best on fitted object of "`iva`" class.

Usage

```
## S3 method for class 'iva'
predict(object, newdata, which.dataset = NA, ...)
```

Arguments

object	An object of class "iva", usually the result of a call to NewtonIVA or fastIVA.
newdata	A numeric data array containing new observed mixtures. Either with dimension [P, N, D] (if which.dataset = NA) or [P, N], where P is the number of sources, N is the number of observations and D is the number of datasets.
which.dataset	Positive integer to determine which dataset is returned. If not set, returns all datasets.
...	further arguments are not used.

Details

The function calculates the source estimates for new observed mixtures based on the model fitted originally. The estimates are zero mean and scaled to unit variance.

Value

Numeric array containing the estimated sources with dimension [P, N] if which.dataset is provided and with dimension [P, N, D] if which.dataset is not provided.

Author(s)

Mika Sipilä

See Also

NewtonIVA, fastIVA

Examples

```
if (require("LaplacesDemon")) {
  # Generate sources from multivariate Laplace distribution
  P <- 4; N <- 1000; D <- 4;
  S <- array(NA, c(P, N, D))
  sigmas <- list()

  for (i in 1:P) {
    U <- array(rnorm(D * D), c(D, D))
    sigmas[[i]] <- crossprod(U)
    S[i, , ] <- rmvl(N, rep(0, D), sigmas[[i]])
  }

  # Generate mixing matrices from standard normal distribution
  A <- array(rnorm(P * P * D), c(P, P, D))

  # Generate mixtures
  X <- array(NaN, c(P, N, D))
  for (d in 1:D) {
    X[, , d] <- A[, , d] %*% S[, , d]
  }

  # Estimate sources and unmixing matrices
  res_G <- NewtonIVA(X, source_density = "gaussian")

  # Generate new observarions
```

```
  N_new <- 10
  S_new <- array(NA, c(P, N_new, D))
  for (i in 1:P) {
    S_new[i, , ] <- rmvl(N_new, rep(0, D), sigmas[[i]])
  }
  X_new <- array(NaN, c(P, N_new, D))
  for (d in 1:D) {
    X_new[, , d] <- A[, , d] %*% S_new[, , d]
  }

  # Get source estimates for the new observations
  pred <- predict(res_G, X_new)

  # Get source estimates for only the second dataset
  pred2 <- predict(res_G, X_new[, , 2], which.dataset = 2)
}
```

`print.iva` *Print an Object of Class iva*

Description

`print` method for the class `"iva"`.

Usage

```
## S3 method for class 'iva'
print(x, ...)
```

Arguments

`x`	An object of class `"iva"`, usually the result of a call to `NewtonIVA` or `fastIVA`.
`...`	Further arguments are not used.

Details

The function prints all information of `"iva"` object, except the estimated source signals.

Value

No return value, called for printing information of the object of class `"iva"`.

Author(s)

Mika Sipilä

See Also

`NewtonIVA`, `fastIVA`

Examples

```
if (require("LaplacesDemon")) {
  # Generate sources from multivariate Laplace distribution
  P <- 4; N <- 1000; D <- 4;
  S <- array(NA, c(P, N, D))

  for (i in 1:P) {
    U <- array(rnorm(D * D), c(D, D))
    Sigma <- crossprod(U)
    S[i, , ] <- rmvl(N, rep(0, D), Sigma)
  }

  # Generate mixing matrices from standard normal distribution
  A <- array(rnorm(P * P * D), c(P, P, D))

  # Generate mixtures
  X <- array(NaN, c(P, N, D))
  for (d in 1:D) {
    X[, , d] <- A[, , d] %*% S[, , d]
  }

  # Estimate sources and unmixing matrices
  res_G <- NewtonIVA(X, source_density = "gaussian")
  print(res_G)
}
```

`summary.iva` *Summarize an Object of Class iva*

Description

`summary` method for the class `"iva"`.

Usage

```
## S3 method for class 'iva'
summary(object, ...)
```

Arguments

`object`	An object of class `"iva"`, usually the result of a call to `NewtonIVA` or `fastIVA`.
`...`	Further arguments are not used.

Details

The function print all the information of the `"iva"` object except the estimated sources and the estimated unmixing matrices.

Value

No return value, called for summarizing the object of class `"iva"`.

Author(s)

Mika Sipilä

See Also

NewtonIVA, fastIVA

Examples

```
if (require("LaplacesDemon")) {
  # Generate sources from multivariate Laplace distribution
  P <- 4; N <- 1000; D <- 4;
  S <- array(NA, c(P, N, D))

  for (i in 1:P) {
    U <- array(rnorm(D * D), c(D, D))
    Sigma <- crossprod(U)
    S[i, , ] <- rmvl(N, rep(0, D), Sigma)
  }

  # Generate mixing matrices from standard normal distribution
  A <- array(rnorm(P * P * D), c(P, P, D))

  # Generate mixtures
  X <- array(NaN, c(P, N, D))
  for (d in 1:D) {
    X[, , d] <- A[, , d] %*% S[, , d]
  }

  # Estimate sources and unmixing matrices
  res_G <- NewtonIVA(X, source_density = "gaussian")
  summary(res_G)
}
```

Index