WORKSHEETS

Trine Nyholm Kragh & Laura Nyrup Mogensen
Mathematical Engineering, MATTEK

Master's Thesis

STUDENT REPORT
AALBORG UNIVERSITY

Chapter 1

Sparse Signal Recovery

This chapter gives an introduction to the concept compressive sensing. Associated theory regarding sparse signal recovery is described along side the limitations of the common solution approaches. Finally the state of the art methods regarding non-sparse signal is presented.

1.1 Compressive Sensing

Compressive sensing is the theory of efficient recovery or reconstruction of a signal from a minimal number of observed measurements. Assuming linear acquisition of the original information the relation between the measurements and the signal to be recovered is described by a linear model[4]

$$\mathbf{y} = \mathbf{A}\mathbf{x}. \tag{1.1}$$

Here $\mathbf{y} \in \mathbb{R}^M$ is the measured data consisting of M observations, $\mathbf{x} \in \mathbb{R}^N$ is the original signal consisting of N possible sources. $\mathbf{A} \in \mathbb{R}^{M \times N}$ is a matrix which models the linear measurement process or in other words it projects each of the N possible sources on to the M observations. $\mathbf{A}$ is referred to as a dictionary or a mixing matrix .

In compressive sensing terminology, $\mathbf{x}$ is the signal of interest which is sought recovered by solving the linear system (1.1). In the typical compressive sensing case where $M < N$ the system becomes underdetermined and there are infinitely many solutions, provided that a solution exist. Such system is also referred to as overcomplete *(as the number of basis vectors is greater than the dimension of the input).*
However, by enforcing certian sparsity constraints it is possible to recover the wanted signal[4], hence the term sparse signal recovery.

Yet another argument; If $M << N$, it leads to the matrix $\mathbf{A}$ being rank-deficient *(but not necessarily?)* which imply that $\mathbf{A}$ has a non-empty null space and this leads to infinitely many signals which yield the same solution $\mathbf{y} = \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}'$ [3, p. ix]. Thus it is necessary to limit the solution space to a specific class of signals $\mathbf{x}$, for this a certain constraint on sparseness is introduced.

1.1.1 Sparseness

A signal is said to be k-sparse if the signal has at most k non-zero coefficients. For the purpose counting the non-zero entries of a vector representing a signal the ℓ_0-norm is defined

$$\|\mathbf{x}\|_0 := \mathrm{card}(\mathrm{supp}(\mathbf{x})) \cancel{\leq k.}$$

The function card$(\cdot)$ gives the cardinality of the input and the support vector of $\mathbf{x}$ is given as

$$\mathrm{supp}(\mathbf{x}) = \{j \in [N] \ : \ x_j \neq 0\},$$

where $[N]$ a set of integers $\{1, 2, \cdots, N\}$ [4, p. 41]. Let the support vector of $\mathbf{x}$ be denoted by $S = \mathrm{supp}(x)$. The set of all k-sparse signals is denoted as

$$\Omega_k = \{\mathbf{x} \ : \ \|\mathbf{x}\|_0 \leq k\}.$$

1.1.2 Optimisation Problem

The signal of interest is assumed to be k-sparse with $k < M$.
From the desire of finding a k-sparse solution $\mathbf{x}$ the following optimisation problem is considered

$$\min_{\mathbf{z}} \|\mathbf{z}\|_0 \quad \text{s.t} \quad \mathbf{y} = \mathbf{A}\mathbf{z},$$

~~where $\mathbf{z}$ is all possible candidates to a k-sparse signal $\mathbf{x}$~~. Unfortunately, this optimisation problem is non-convex due to the definition of ℓ_0-norm and is therefore difficult to solve – it is a NP-hard problem. Instead by replacing the ℓ_0-norm with its convex approximation, the ℓ_1-norm, the optimisation problem become computational feasible [3, p. 27]

$$\min_{\mathbf{z}} \|\mathbf{z}\|_1 \quad \text{s.t} \quad \mathbf{y} = \mathbf{A}\mathbf{z}, \tag{1.2}$$

and instead we find the best k-term approximation of the signal $\mathbf{x}$. This method is referred to as Basis Pursuit and makes the foundation of several algorithms solving alternative versions of (1.2) where noise is incorporated. A different type of solution method includes greedy algorithm such as the Orthogonal Matching Pursuit.

1.1.3 Conditions on the dictionary

The construction of the matrix **A** is of course essential for the solution of the optimisation problem. So far no one has manage to construct a matrix which is proved to be optimal for some compressive sensing set up. However some certain constructions have shown sufficient recovery guarantee.

To ensure an exact or an approximate reconstruction of the sparse signal **x** some conditions associated to the matrix **A** must be satisfied.
This includes at first the null space condition, a property of the A matrix which is hard to check in practise. The Restricted isometry condition is a stronger condition concerning the orthogonality of the matrix. Furthermore the coherence of a matrix is a measure of quality and is used to determine whether the matrix A is a good choice for the optimisation problem.
(Is it necessary describe these conditions on A in details?)

Dictionary learning

In cases where the dictionary is unknown it is possible to learn the dictionary from the observed measurement provided that several observations are available $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n]$. In dictionary learning framework the inverse problem is defined as

$$\min_{A,X} = \frac{1}{2} \sum_{s=1}^{N_d} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \gamma \sum_{s=1}^{N_d} g(\mathbf{x}_s),$$

where the function $g(\cdot)$ promotes sparsity of the source vector at time t The true dictionary **A** is recovered if the sources $\mathbf{x}_s$ are sparse $(k_s < M)$.

Include specific dictionary learning algorithm here...

1.1.4 Multiple Measurement Vector Model

The linear model (1.1) is also referred to as a single measurement model. In order to adapt the model to practical use the model is expanded to include multiple measurements and take noise into account.
A multiple measurement vector (MMV) model consist of the observation matrix $\mathbf{Y} \in \mathbb{R}^{M \times L}$, the source matrix $\mathbf{X} \in \mathbb{R}^{N \times L}$ which have $k \leq M$ rows that are non-zero and a dictionary matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$:

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{E},$$

where L denote the number of observed samples each consisting of M measurements and $\mathbf{E} \in \mathbb{R}^{M \times L}$. From the MMV model the non-zero rows of the source matrix **X** are the ones of interest, which one want to recover [2, p. 11].

1.1.5 Limitations of compressive sensing

(we are not yet sure where to put a section explaining the issue of $k > M$.)

- If the source signal is sparse it is enough just to find the non-zero rows of X denoted by the set S, because then the source signal can be obtained by the psudo-inverse solution $\hat{\mathbf{X}} = \mathbf{A}_S^{perp}\mathbf{Y}$ where $\mathbf{A}_S$ is derived from the dictionary matrix $\mathbf{A}$ by deleting the columns associated with the zero rows of X. S is called the support. (We identify the locations of sources)

- in general for $k > M$ it is not possible to recover $\mathbf{x}$ as the system is underdetermined/overcomplete, furthermore we can not find the true dictionary $\mathbf{A}$ by dictionary learning methods because when $k > M$, where M is the dimension of $\mathbf{y}$, any random dictionary can be used create $\mathbf{y}$ from $\geq M$ basis vectors. that is generally the accuracy of recovery a $\mathbf{A}$ increases as $k << M$.

1.2 Covariance-Domain Dictionary Learning

Covariance-domain dictionary learning (Cov-DL) is an algorithm proposed in [1] which claims to be able to identify more sources N than available observations M for the linear model.

Consider the multiple measurement vector model as above

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{E}.$$

Let s be the index of time segments that the observed data $\mathbf{Y} \in \mathbb{R}^{M \times L}$ is divided into and let S_f be the sample frequency. As such the observed data is divided into segments $\mathbf{Y}_s \in \mathbb{R}^{M \times t_s S_f}$, possibly overlapping, where t_s is the length of the segments in seconds. For each segment the linear model still holds and is rewritten into

$$\mathbf{Y}_s = \mathbf{A}\mathbf{X}_s + \mathbf{E}_s, \quad \forall s.$$

Cov-DL takes advantage of the dictionary framework and transformation into another domain – covariance domain – to recover the mixing matrix $\mathbf{A}$ from the observed data $\mathbf{Y}$. An important aspect of this method is the prior assumption that the sources are statistical independent within the defined time segments. This implies the entries in $\mathbf{X}_s$ to be uncorrelated.

Cov-DL works together with another algorithm to find the source matrix $\mathbf{X}$, in this thesis M-SBL is used for the source recovery and is described in section 2.1.
In this section we assume that $\mathbf{X}$ is known but in practice a random sparse matrix will be used to represent the sources.
The section is inspired by chapter 3 in [2] and the article [1].

1.2.1 Covariances domain representation

The observed data $\mathbf{Y}_s$ can be described in the covariance domain by the sample covariance matrix. Considering the observations $\mathbf{Y}_s \in \mathbb{R}^{M \times L}$ the sample covariance is defined to find the covariance among the M variables across the L observations, that is essentially the covariance matrix averaged over all observations, resulting in a $M \times M$ matrix $\Sigma_{\mathbf{Y}_s} = [\sigma_{jk}]$. Each entry is defined by[wiki?]

$$\sigma_{jk} = \frac{1}{L} \sum_{i=1}^{L} (y_{ji} - \mathbb{E}_s[y_j])(y_{ki} - \mathbb{E}_s[y_k])$$

Let the observations(argument for at vi antager zero mean på vores observationer, pre-normalisering?) be normalised resulting in zeros mean $\mathbb{E}_s[\mathbf{Y}_s] = 0$.
Using matrix notation the sample covariance of $\mathbf{Y}_s$ can be written as

$$\Sigma_{\mathbf{Y}_s} = \frac{1}{L} \mathbf{Y}_s \mathbf{Y}_s^T$$

Similar the sources $\mathbf{X}_s$ can be described in the covariance domain by the sample covariance matrix

$$\mathbf{\Sigma}_{\mathbf{X}_s} = \frac{1}{L_s} \mathbf{X}_s \mathbf{X}_s^T$$

From the assumption of uncorrelated sources the sample covariance matrix is expected to be nearly diagonal, thus it can be expressed as

$$\mathbf{\Sigma}_{\mathbf{X}_s} = \Lambda + \mathbf{E}_s.$$

where Λ is a diagonal matrix consisting of the diagonal entries of $\Sigma_{\mathbf{X}_s}$ and $\mathbf{E}_s$ contains the remaining entries which is expected to be zeros or nearly zeros[1].

Each segment of observations can be modelled as

skal vi have $\mathbb{E}_s[]$ omkring her eller ej?, wiki covariance. giver det mening at udelade $\frac{1}{L}$ på begge sider her?

$$\begin{aligned} \mathbf{Y}_s \mathbf{Y}_s^T &= (\mathbf{A}\mathbf{X}_s)(\mathbf{A}\mathbf{X}_s)^T \\ &= \mathbf{A}\mathbf{X}_s \mathbf{X}_s^T \mathbf{A}^T \\ &= \mathbf{A}\Sigma_{X_s}\mathbf{A}^T \\ &= \mathbf{A}\mathbf{\Lambda}\mathbf{A}^T + \mathbf{E}_s = \sum_{i=1}^{N} \Lambda_{ii} \mathbf{a}_i \mathbf{a}_i^T + \mathbf{E}_s \qquad (1.3) \end{aligned}$$

Remember that N is the dimension of X hence the number of possible sources and k is the number of active sources. It has been shown that by this model it is possible to identify $O(M^2)$ sources given the true dictionary[6](dog er vektoriseringen også includeret i resultatet - så måske flyttes udtalelsen?). The purpose of the Cov-DL algorithm is instead to find the dictionary $\mathbf{A}$ from this expression and then still allow for $O(M^2)$ sources to be identified.

1.2.2 Determination of dictionary

er det muligt at vektoriceringen kun er til for at gøre dictionary learning problemet simplere? og ikke har noget med $O(M^2)$ at gøre, læs Pal2015

In order enable the possibilities of identifying $O(M^2)$ sources and learning the corresponding dictionary A the model in (1.3) is rewritten. At first the both sides of the expression is vectorized. Because the covariance matrix is symmetric it is sufficient to vectorize only the lower triangular parts, including the diagonal. For this the function vec$(\cdot)$ is defined to map a symmetric $M \times M$ matrix into a vector of size $\frac{M(M+1)}{2}$ making a vectorisation of its lower triangular part. Further let $\text{vec}^{-1}(\cdot)$ be the inverse function. Note that $\Lambda_{s_{ii}}$ is a scalar hence not vectorised

$$\text{vec}(\mathbf{\Sigma}_{\mathbf{Y}_s}) = \sum_{i=1}^{N} \Lambda_{s_{ii}} \text{vec}(\mathbf{a}_i \mathbf{a}_i^T) + \text{vec}(\mathbf{E}_s) \tag{1.4}$$

$$= \sum_{i=1}^{N} \mathbf{d}_i \Lambda_{s_{ii}} + \text{vec}(\mathbf{E}_s) \tag{1.5}$$

$$= \mathbf{D}\boldsymbol{\delta}_s + \text{vec}(\mathbf{E}_s), \quad \forall s. \tag{1.6}$$

The vector $\boldsymbol{\delta}_s \in \mathbb{R}^N$ contains the diagonal entries of the source sample-covariance matrix $\mathbf{\Sigma}_{\mathbf{X}_s}$ and the matrix $\mathbf{D} \in \mathbb{R}^{M(M+1)/2 \times N}$ consists of the columns $\mathbf{d}_i = \text{vech}(\mathbf{a}_i \mathbf{a}_i^T)$. Note that $\mathbf{D}$ and δ_s are unknown while the left hand side is known from the observed data.

include noise in the original problem

From this expression it is now possible to learn $\mathbf{D}$ and then find the associated matrix $\mathbf{A}$. Comparing this expression to the original compressive sensing problem (1.1) it is clear that higher dimensionality is achieved by representing the problem in the covariance domain. Which allows for the number of active sources to exeed the number of observartions(måske for tidligt at konkludere her).

The Cov-DL method consists of two different algorithms for recovery of $\mathbf{D}$ and $\mathbf{A}$ depending on the number of total sources N relative to the number of observations M.

Cov-DL1 – overcomplete D

The case where $N > \frac{M(M+1)}{2}$ results in an overcomplete system similar to the original system being overcomplete when $N > M$. Though, it is again possible to solve the overcomplete system if certain sparsity is withhold. Namely $\boldsymbol{\delta}_s$ being $\frac{M(M+1)}{2}$-sparse. Note that this sparsity constraint is weaker than the original constraint $k < M$, which allows for identification of remarkably more sources than within the original domain as it is not necessarily violated when $k > M$. Assuming a sufficient sparsity on $\boldsymbol{\delta}_s$ it is possible to learn the dictionary matrix of the covariance domain $\mathbf{D}$ by traditional dictionary learning methods, as introduced in section 1.1.3, applied to the observations represented in the covariance domain $\text{vec}(\mathbf{\Sigma}_{\mathbf{Y}_s})\ \forall s$.
When $\mathbf{D}$ is known it is possible to find the original mixing matrix $\mathbf{A}$ through the

relation $\mathbf{d}_i = \text{vec}(\mathbf{a}_i \mathbf{a}_i^T)$.
To do so each column is found by the optimisation problem

$$\min_{\mathbf{a}_i} \|\text{vec}^{-1}(\mathbf{d}_i) - \mathbf{a}_i \mathbf{a}_i^T\|_2^2$$

for which the global minimizer is $\mathbf{a}_i^* = \sqrt{\lambda_i}\mathbf{b}_i$. Here λ_i is the largest eignevalue of $\text{vec}^{-1}(\mathbf{d}_i)$ and $\mathbf{b}_i$ is the corresponding eigenvector.

redegørelse for resultatet her skal laves

Cov-DL2 – undercomplete D

Chapter 2

Independent Component Analysis

(it is not the intention to keep a whole chapter about ICA within the rapport, but rather a smaller section within the chapter of compressive sensing as a common solution method, and the rest as appendix.)

Independent component analysis (ICA) is a method *(within sparse signal recovery?)* which can separate statistical independent sources $\mathbf{X}$ and identify the mixing matrix $\mathbf{A}$ given the observed measurements $\mathbf{Y}$.
The main aspect of ICA is to assume statistical independent between the wanted components and nongaussiantity of the data. [5, p. 3].
Through this section the mathematical concepts of ICA will be explained and defined in the noise-less case.

To understand what ICA is let us describe a situation where ICA could be used. Two people are having a conversation inside a room full of people which talk simultaneous with each other. The conversation between the two first mentioned people will the be affected by the surrounding conversations and noise. Such environment is often referred to as the cocktail party problem and is a difficult environment to be in as a hearing impaired.
Let us describe the situation with some mathematical notations. The observed conversation which is affected by surrounding noise is denoted $\mathbf{y}$, the original individual conversations in the room is denoted by $\mathbf{x}$. All the original conversations are effected by each other and possibly noise, let us denoted this mixture by a matrix $\mathbf{A}$. We omit the time indexes and the time dependence to view the problem with random

vectors. The problem can then be described as a linear model

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \sum_{i=1}^{n} \mathbf{a}_i x_i, \tag{2.1}$$

$$y_i = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n, \quad i = 1, \ldots, n. \tag{2.2}$$

The only known variables in this model are the observed conversation $\mathbf{y}$, the rest are unknown.

ICA is a method which can be use to recover the unknown mixture $\mathbf{A}$ and sources $\mathbf{x}$, which in the ICA aspect are the components, from the known $\mathbf{y}$. By use of the statistical properties of the components $\mathbf{x}$ it is then possible to estimate/recover the original conversations and then the mixture $\mathbf{A}$.
One of the properties is the assumption that the components $\mathbf{x}$ are statistical independent. By independence, one means that the joint probability density function (pdf) can be factorised into the marginal pdfs of the components $\mathbf{x}$, that is

$$p(x_1, x_2, \ldots, x_n) = p_1(x_1)p_2(x_2)\cdots p_n(x_n),$$

for the random variables x_i.
Furthermore it is assumed that the independent components $\mathbf{x}$ do not have Gaussian distributions as this will effect the recovery of $\mathbf{x}$ due to ICA using higher-order cumulant method. For Gaussian distribution the cumulant will be zero which is not wanted in the recovery process. Thus the distribution of the independent components are unknown. This will further be described in section 2.0.1.
The last assumption is that the mixing matrix $\mathbf{A}$ must be a square matrix – there must be the same number of independent components as observation [5, p. 152-153].

From the right-hand side of (2.1) being unknown some statistics can not be computed e.g. the variance of $\mathbf{x}$. Instead ICA assume that $\mathbf{x}$ has unit variance (= 1). This of course must also apply to the mixing matrix $\mathbf{A}$ which will be restricted in the recovery method, described in section 2.0.1. By this addition the algorithm for ICA is simplified. To simplify even more, without loss of generality assume that $\mathbb{E}[\mathbf{y}] = 0$ and $\mathbb{E}[\mathbf{x}] = 0$.
To achieve zero mean, the observed data $\mathbf{y}$ is normalised as a preprocessing. This addition do not affect the recover/estimation of $\mathbf{A}$ [5, p. 154].

Another preprocessing step is to whiten the observed data $\mathbf{y}$. This ensures that the independent components $\mathbf{x}$ are uncorrelated and have variance equal 1. Furthermore, this also reduce the complexity of ICA and therefore simplifies the recovering process.
Whiting is a process which uses the eigenvalue decomposition (EVD) to find the

eigenvalues and associated eigenvectors from the covariance of observed data $\mathbf{y}$:

$$\mathbb{E}[\mathbf{y}\mathbf{y}^T] = \mathbf{E}\mathbf{D}\mathbf{E}^T,$$

where $\mathbf{D}$ is a diagonal matrix of eigenvalues and $\mathbf{E}$ is the associated eigenvectors. With $\mathbf{E}$ and $\mathbf{D}$ a whiting matrix is constructed as

$$\mathbf{V} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T.$$

By multiplying the observed data $\mathbf{y}$ with a whiting matrix $\mathbf{V}$ the data becomes white:

$$\begin{aligned} \mathbf{y}_{\text{white}} &= \mathbf{V}\mathbf{y}, \qquad (2.3)\\ \mathbf{y}_{\text{white}} &= \mathbf{V}\mathbf{A}\mathbf{x} = \mathbf{A}_{\text{white}}\mathbf{x} \end{aligned}$$

By the whiting process the mixing matrix $\mathbf{A}_{\text{white}}$ becomes orthogonal because

$$\mathbb{E}[\mathbf{y}_{\text{white}}\mathbf{y}_{\text{white}}^T] = \mathbf{A}_{\text{white}}\mathbb{E}[\mathbf{x}\mathbf{x}^T]\mathbf{A}_{\text{white}}^T = \mathbf{A}_{\text{white}}\mathbf{A}_{\text{white}}^T = \mathbf{I}.$$

This mean that ICA can restrict its search for $\mathbf{A}_{\text{white}}$ to the orthogonal matrix space- That is instead of estimateing n^2 parameters ICA now only has to estimate an orthogonal matrix which have $n(n-1)/2$ parameters/degrees of freedom [5, p. 159].
Furthermore, to support the assumption of nongaussiantity for an orthogonal mixing matrix $\mathbf{A}_{\text{white}}$ it is not possible to distinct the pdfs of the $\mathbf{y}_{\text{white}}$ and $\mathbf{x}$ as $\mathbf{A}_{\text{white}}$ is no longer included in the pdf of $\mathbf{y}_{\text{white}}$ and therefore are the two pdfs equal [5, p. 161-163].

can we elaborate this last part

2.0.1 Recovery of the Independent Components

A way to recover the independent components could be to take advantage of the assumption of nongaussiantity. By finding the estimate which maximise the nongaussiantity the real independent component can be found based on the fact that it must have a nongaussian distribution as mention in section 2. But before the estimation a measure of nongaussiantity must be introduce – this could be the kurtosis.

Kurtosis

Kurtosis is a quantitative measure used for nongaussianity of random variables. (Excess) Kurtosis of a random variable y is defined as

Tjek lige op denne definition

$$\begin{aligned} \text{kurt}(y) &= \mathbb{E}\left[\left(\frac{y-\mu}{\sigma}\right)^4\right] - 3\\ &= \mathbb{E}[y^4] - 3(\mathbb{E}[y^2])^2, \end{aligned}$$

which is the fourth-order cumulant of the random variable y. By assuming that the random variable y has variance $\mathbb{E}[y^2] = 1$, the kurtosis is rewritten as

$$\mathrm{kurt}(y) = \mathbb{E}[y^4] - 3.$$

The kurtosis of nongaussian random variables will then almost always be different from zero. For gaussian random variables the fourth moment equals $3(\mathbb{E}[y^2])^2$ thus the kurtosis will then be zero [5, p. 171].
By using the absolute value of the kurtosis gaussian random variables are still zero but the nongaussian random variables will be greater than zero. In this case the random variables are called supergaussian.

One complication with kurtosis as a measure is that kurtosis is sensitive to outliers [5, p. 182].

The Gradient Algorithm To recover the independent components $\mathbf{x}$ the nongaussianity is wish maximised. One way to do this is to use a gradient algorithm to maximise the kurtosis of $\mathbf{y}$.

The idea behind a gradient algorithm is to move in certain directions computed from the gradient of an objective function – in this case it is the kurtosis – until convergence is achieved.
Let $\mathbf{w}$ be an initial vector used to compute the direction and let $y = \mathbf{w}^T\mathbf{y}_{\text{white}}$ given some samples of the observed and preprocessed vector. The direction $\mathbf{w}$ which provide the highest kurtosis is the new direction for the algorithm. This continues until convergence is reach defined by some tolerance.
Regn lige efter
The gradient of $|\mathrm{kurt}(\mathbf{w}^T\mathbf{y}_{\text{white}})|$ is computed as

$$\begin{aligned}\frac{\partial|\mathrm{kurt}(\mathbf{w}^T\mathbf{y}_{\text{white}})|}{\partial\mathbf{w}} &= 4\mathrm{sign}(\mathrm{kurt}(\mathbf{w}^T\mathbf{y}_{\text{white}}))(\mathbb{E}[\mathbf{y}_{\text{white}}(\mathbf{w}^T\mathbf{y}_{\text{white}})^3] - 3\mathbf{w}\mathbb{E}[(\mathbf{w}^T\mathbf{y}_{\text{white}})^2] \\ &= 4\mathrm{sign}(\mathrm{kurt}(\mathbf{w}^T\mathbf{y}_{\text{white}}))(\mathbb{E}[\mathbf{y}_{\text{white}}(\mathbf{w}^T\mathbf{y}_{\text{white}})^3] - 3\mathbf{w}\|\mathbf{w}\|^2).\end{aligned} \tag{2.4}$$

The absolute value of kurtosis is optimised onto the unit sphere, $\|\mathbf{w}\|^2 = 1$, the algorithm must project onto the unit sphere in every step. This can easily be done by dividing $\mathbf{w}$ with its norm.
Furthermore, the last part of (2.4) can be omitted as it do not effect the direction. The expectation operator is omitted to achieve an adaptive algorithm:

$$\frac{\partial|\mathrm{kurt}(\mathbf{w}^T\mathbf{y}_{\text{white}})|}{\partial\mathbf{w}} = 4\mathrm{sign}(\mathrm{kurt}(\mathbf{w}^T\mathbf{y}_{\text{white}}))\mathbf{y}_{\text{white}}(\mathbf{w}^T\mathbf{y}_{\text{white}})^3$$

The expectation operator from the definition of kurtosis can not be omitted and must therefore be estimated. This can be done by a time-average estimate, denoted as γ:

$$\gamma = ((\mathbf{w}^T \mathbf{y}_{\text{white}})^4 - 3) - \gamma$$

This results in the following gradient algorithm.

Algorithm 1 Gradient Algorithm with Kurtosis

```
1. Center the observed data to achieve zero mean
2. Whiten the centered data
3. Create the initial random vector w and the initial value for $\gamma$
4. Compute $\mathbf{w} = \gamma \mathbf{y}_{\text{white}}(\mathbf{w}^T \mathbf{y}_{\text{white}})^3$
5. Normalise $\mathbf{w} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
6. Update w
7. Update $\gamma = ((\mathbf{w}^T \mathbf{z})^4 - 3) - \gamma$
8. Repeat until convergence
9. Independent components are found as $\mathbf{x} = \mathbf{w}\mathbf{y}_{\text{white}}$
```

Fixed-Point Algorithm - FastICA A fixed-point algorithm to maximise the nongaussianity is more efficient than the gradient algorithm as the gradient algorithm converge slow depending on the choice of γ. The fixed-point algorithm is an alternative that could be used. By using the gradient given in (2.4) and then set the equation equal with $\mathbf{w}$:

$$\mathbf{w} \propto (\mathbb{E}[\mathbf{y}_{\text{white}}(\mathbf{w}^T \mathbf{y}_{\text{white}})^3] - 3\|\mathbf{w}\|^2 \mathbf{w})$$

By this new equation, the algorithm find $\mathbf{w}$ by simply calculating the right-hand side:

$$\mathbf{w} = \mathbb{E}[\mathbf{y}_{\text{white}}(\mathbf{w}^T \mathbf{y}_{\text{white}})^3] - 3\mathbf{w}$$

As with the gradient algorithm the fixed-point algorithm do also divide the found $\mathbf{w}$ by its norm. Therefore is $\|\mathbf{w}\|$ omitted from the equation.
Instead of γ the fixed-point algorithm compute $\mathbf{w}$ directly from previous $\mathbf{w}$.

The fixed-point algorithm have been summed in the following algorithm.

Algorithm 2 Fixed-Point Algorithm with Kurtosis

```
1. Center the observed data to achieve zero mean
2. Whiten the centered data
3. Create the initial random vector $\mathbf{w}$
4. Compute $\mathbf{w} = \mathbb{E}[\mathbf{y}_{\text{white}}(\mathbf{w}^T\mathbf{y}_{\text{white}})^3] - 3\mathbf{w}$
5. Normalise $\mathbf{w} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
6. Update $\mathbf{w}$
7. Repeat until convergence
8. Independent components are found as $\mathbf{x} = \mathbf{w}\mathbf{y}_{\text{white}}$
```

The fixed-point algorithm is also called for FastICA as the algorithm has shown to converge fast and reliably, then the current and previous $\mathbf{w}$ laid in the same direction [5, p. 179].

Negentropy

Another measure of nongaussianity is the negentropy which is based on the differential entropy. The differential entropy H of a random variable/vector $\mathbf{y}$ with density $p_y(\boldsymbol{\theta})$ is defined as

$$H(\mathbf{y}) = -\int p_y(\boldsymbol{\theta}) \log(p_y(\boldsymbol{\theta}))\ d\boldsymbol{\theta}.$$

The entropy describes the information of a random variable. For variables becoming more random the entropy becomes larger, e.g. gaussian random variables have a high entropy, in fact gaussian random variables have the highest entropy among the random variables of the same variance [5, p. 182].

To use the negentropy to define the nongaussianity within random variables, the differential entropy is normalised to obtain a entropy value equal to zero when the random variable is gaussian and non-negative otherwise. The negentropy J is defined as

$$J(\mathbf{y}) = H(\mathbf{y}_{\text{gaus}}) - H(\mathbf{y}),$$

with $\mathbf{y}_{\text{gaus}}$ being a gaussian random variable of the same covariance and correlation as $\mathbf{y}$ [5, p. 182].

As the kurtosis is sensitive for outliers the negentropy is instead difficult to compute computationally as the negentropy require a estimate of the pdf. Instead it could be an idea to use an approximation of the negentropy.

Approximation of Negentropy

The way to approximate the negentropy is to look at the high-order cumulants using polynomial density expansions such that the approximation could be given as

$$J(y) \approx \frac{1}{12}\mathbb{E}[y^3]^2 + \frac{1}{48}\mathrm{kurt}(y)^2. \tag{2.5}$$

The random variable y has zero mean and unit variance and the kurtosis is introduced in the approximation. The approximation suffers from nonrobustness with the kurtosis and therefore a more generalised approximation is presented to avoid the nonrobustness.

For the generalised approximation the use of expectations of nonquadratic functions is introduced. The polynomial functions y^3 and y^4 from (2.5) are replaced by G^i with i being an index and G being some function. The approximation in (2.5) then becomes

$$J(y) \approx (\mathbb{E}[G(y)] - \mathbb{E}[G(\nu)])^2.$$

The choice of G can lead to a better approximation than (2.5) and by choosing one which do not grow to fast more robust estimators can be obtained. The choice of G could be the two following functions

$$G_1(y) = \frac{1}{a_1}\log(\cosh(a_1 y)), \quad 1 \leq a_1 \leq 2$$
$$G_2(y) = -\exp\left(\frac{-y^2}{2}\right)$$

Gradient Algorithm with Negentropy As described in section 2.0.1 the gradient algorithm is used to maximising negentropy. The gradient of the approximated negentropy is given as

$$\mathbf{w} = \gamma \mathbf{y}_{\text{white}} g(\mathbf{w}^T \mathbf{y}_{\text{white}})$$

with respect to $\mathbf{w}$ and where $\gamma = \mathbb{E}[G(\mathbf{w}^T \mathbf{y}_{\text{white}})] - \mathbb{E}[G(\nu)]$ with ν being the standardised gaussian random variable. g is the derivative of the nonquadratic function G. To omitted the expectation γ as we did with the sign of kurtosis, γ is estimated as

$$\gamma = (G(\mathbf{w}^T \mathbf{y}_{\text{white}}) - \mathbb{E}[G(\nu)]) - \gamma.$$

For the choice of g the derivative of the functions presented in (2.6) could be use to achieve a robust result. Alternative a derivative which correspond to the fourth-power as seen in the kurtosis could be used. The functions g could be

$$\begin{aligned} g_1(y) &= \tanh(a_1 y), \quad 1 \leq a_1 \leq 2 \\ g_2(y) &= y \exp\left(\frac{-y^2}{2}\right) \\ g_3(y) &= y^3 \end{aligned} \tag{2.6}$$

Algorithm 3 Gradient Algorithm

1. Center the observed data to achieve zero mean
2. Whiten the centered data
3. Create the initial random vector $\mathbf{w}$ and the initial value for γ
4. Update
$$\mathbf{w} = \gamma \mathbf{y}_{\text{white}} g(\mathbf{w}^T \mathbf{y}_{\text{white}})$$
5. Normalise $\mathbf{w}$
$$\mathbf{w} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$$
6. Check sign of γ, if not a known prior, update
$$\gamma = (G(\mathbf{w}^T \mathbf{y}_{\text{white}}) - \mathbb{E}[G(\nu)]) - \gamma$$
7. Repeat until convergence
8. Independent components are found as $\mathbf{x} = \mathbf{w}\mathbf{y}_{\text{white}}$

Fixed-Point Algorithm with Negentropy As described in the section with kurtosis, the fixed-point algorithm removed the learning parameter and compute $\mathbf{w}$ directly:

$$\mathbf{w} = \mathbb{E}[\mathbf{z} g(\mathbf{w}^T \mathbf{z})]$$

.

Write the expression from this equation to the on in the algorithm

Algorithm 4 Fixed-Point Algorithm with Negentropy (FastICA)

```
1. Center the observed data to achieve zero mean
2. Whiten the centered data
3. Create the initial random vector $\mathbf{w}$
4. Update
       $\mathbf{w} = \mathbb{E}[\mathbf{y}_{\text{white}} g(\mathbf{w}^T \mathbf{y}_{\text{white}})] - \mathbb{E}[g'(\mathbf{w}^T \mathbf{y}_{\text{white}})]\mathbf{w}$
5. Normalise $\mathbf{w}$
       $\mathbf{w} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
6. Repeat from 4. until convergence
7. Independent components are found as $\mathbf{x} = \mathbf{w}\mathbf{y}_{\text{white}}$
```

2.1 MSB

...

Bibliography

[1] Balkan, Ozgur, Kreutz-Delgado, Kenneth, and Makeig, Scott. “Covariance-Domain Dictionary Learning for Overcomplete EEG Source Identification”. In: *ArXiv* (2015).

[2] Balkan, Ozgur Yigit. “Support Recovery and Dictionary Learning for Uncorrelated EEG Sources”. Master thesis. University of California, San Diego, 2015.

[3] C. Eldar, Yonina and Kutyniok, Gitta. *Compressed Sensing: Theory and Application.* Cambridge University Presse, New York, 2012.

[4] Foucart, Simon and Rauhut, Hoger. *A Mathematical Introduction to Compressive Sensing.* Springer Science+Business Media New York, 2013.

[5] Hyvarinen, A., Karhunen, J., and Oja, E. *Independent Component Analysis.* Ed. by Haykin, Simon. John Wiley and Sons, Inc., 2001.

[6] Pal, Piya and Vaidyanathan, P. P. “Pushing the Limits of Sparse Support Recovery Using Correlation Information”. In: *IEEE TRANSACTIONS ON SIGNAL PROCESSING* VOL. 63, NO. 3, Feb. (2015).