Blind Source Separation

**Abstract**

# Contents

# Chapter 1

# Introduction

Blabla... gpp

## 1.1 Formal Problem Statement

We now provide a notation leading to a mathematical statement of the blind source separation (BSS) problem. We let $\boldsymbol{S}(t) \in \mathbf{R}^n$ for $t > 0, n > 0$ denote the signals generated by $n$ sources. Similarly, let $\boldsymbol{X}(t) \in \mathbf{R}^m$ for $t > 0, n > 0$ the observed sensor readings resulting from the emitted signals. A *mixing model* $f(\boldsymbol{S}, t)$ defines the relationship between source and observed signal:

$$\boldsymbol{X} = f(\boldsymbol{S}, t) \tag{1.1}$$

As only the observed value $\boldsymbol{X}$ is known, we need to determine the inverse $f^{-1}(\boldsymbol{S}, t)$, that is, the *unmixing model*.

### Single Sensor Blind Source Separation

A particular instance of the BSS problem, is the single sensor blind source separation (SSBSS) problem, to which we will devote particular attention. In the SSBSS problem, we have one or more source signals, but the observed signal $\boldsymbol{X}(t)$ is a scalar. This introduces problems as this instance does not lend itself to solutions by means of the "standard" methods we consider in the standard BSS problem. Chapter 4 is devoted to the SSBSS problem.

### 1.1.1 A Linear Mixing Model

The simplest mixing model is a noiseless, stationary linear mixing model. The stationarity assumption means that the mixing model does not change as a function of time, so the $t$ argument in Equation 1.1 can be omitted. With $T$ measurements, $N$ sources, and $M$ sensors, this model can be defined as:

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} \tag{1.2}$$

With $X \in \mathbf{R}^{N \times T}$, $\boldsymbol{A} \in \mathbf{R}^{N \times M}$ and $\boldsymbol{S} \in \mathbf{R}^{M \times T}$. The problem of determining the unmixing model now consists of computing the inverse $\boldsymbol{W} = \boldsymbol{A}^{-1}$, so that the original signal:

$$\boldsymbol{S} = \boldsymbol{W}\boldsymbol{X} \tag{1.3}$$

can be recovered. This is to say that the estimate of the original signal $j$ at time $t$ is computed as the $j$th row of $\boldsymbol{W}$ times the $t$th column of $X$.

From Equation 1.2, we can see that the blind source separation problem, even in the simplest case, is ill-poised, as we are trying to determine $M \times T + N \times M$ parameters (both $\boldsymbol{A}$ and $\boldsymbol{S}$) given only $N \times T$ ($\boldsymbol{X}$). This implies that we need to impose some kind of assumptions on the nature of the data. These assumptions, often called the *generative model*, state something about the nature of the signals and how they are mixed. As will be made apparent later, which assumptions are made, gives rise to different solution approaches. For the purpose of this study, we will be quite restrictive in what assumptions we are willing make, hence the term *blind* source separation. The type of assumptions made are primarily related to statistical properties of the sources. The textbook assumptions are uncorrelated and independendent sources, leading to the PCA and ICA solutions, respectively[1].

## 1.2 Overview

In the next chapters we will be looking at a few different algorithms for solving various instances of the BSS problem. Each algorithm has its own merits depending to a large extent on the assumptions we make about the data. An overview of these follow in the Table 1.1.

| **Data characteristic** | **Method** | **Description** |
|---|---|---|
| Uncorrelated sources. | PCA | Blabla |
| Independent, non-gaussian sources. | ICA | blabla |
| Fewer sources than observations. | HMM | blabla |

Table 1.1: Overview over the different approaches to blind source separation covered in this report.

[1]Under the assumptions that the number of observations are greater than or equal to the number of sources.

# Chapter 2

# Principal Component Analysis

Principal component analysis [1] (PCA) is a eigenvector-based, non-probabilistic technique that uses orthogonal projection to represent data in a lower dimensional subspace spanned by the $k$ first eigenvectors of the covariance matrix. The eigenvectors form an orthogonal basis for the data such that a projection onto the eigenvectors will decorrelate the data. In the next section we will derive this result by maximizing the variance of an axis of projection.

PCA is useful in several applications, hereunder visualization and detection of so-called *latent variables*. The principal components (PCs) are the basis of the subspace onto which the data is projected, and are such that the variance explained by each component is maximized; that is, the first PC explains a higher proportion of variance than the second PC and so forth. We can therefore, by retaining only the first few components acheive a representation of the data containing the most of the variance exhibited by the assumption that the PCs accounting for the smallest portion of variance are noise.

The next section presents PCA from two different but equivalent perspectives; first solving for the direction of maximal variation using the method of Lagrange multipliers, and subsequently by singular value decomposition which. The latter is the more computationally efficient, and the rationale for this approach is easy to see once the first perspective is known. We then proceed to looking at how PCA can be applied to the blind source problem and how the assumptions made about the data affect the results of a real-world mixing case.

## 2.1 Formal Statement

Let $\boldsymbol{x}_i \in \mathbf{R}^n$ denote the $i$'th observation of a dataset of $m$ observations. We now want to project our data onto a vector $\boldsymbol{u}$ in $\mathbf{R}^n$ so as to maximize the variance of the resulting projection $\sum_{i=1}^m \boldsymbol{x}_i^T \boldsymbol{u}$ subject to the constraint $|\boldsymbol{u}| = 1$. Under the assumption that $\boldsymbol{X}$ is standardized to zero mean and unit variance, the Lagrangian is then given by Equation 2.1:

$$\begin{aligned}
\mathcal{L}(u, \lambda) &= \tfrac{1}{m} \textstyle\sum_{i=1}^m (\boldsymbol{x}_i^T \boldsymbol{u})^2 - \lambda(\boldsymbol{u}^T \boldsymbol{u} - 1) \\
&= \tfrac{1}{m} \textstyle\sum_{i=1}^m (\boldsymbol{u}^T \boldsymbol{x}_i)^T (\boldsymbol{x}_i^T \boldsymbol{u}) - \lambda(\boldsymbol{u}^T \boldsymbol{u} - 1) \\
&= \tfrac{1}{m} \textstyle\sum_{i=1}^m \boldsymbol{u}^T (\boldsymbol{x}_i \boldsymbol{x}_i^T) u - \lambda(\boldsymbol{u}^T \boldsymbol{u} - 1) \qquad (2.1) \\
&= \tfrac{1}{m} u^T \textstyle\sum_{i=1}^m (\boldsymbol{x}_i \boldsymbol{x}_i^T) \boldsymbol{u} - \lambda(\boldsymbol{u}^T \boldsymbol{u} - 1) \\
&= \tfrac{1}{m} \boldsymbol{u}^T \boldsymbol{\Sigma} \boldsymbol{u} - \lambda(\boldsymbol{u}^T \boldsymbol{u} - 1)
\end{aligned}$$

Here, $\boldsymbol{\Sigma} = \sum_{i=1}^m \boldsymbol{x}_i \boldsymbol{x}_i^T$ is the covariance matrix. Setting the gradient of 2.1 equal to zero yields Equation 2.2:

$$\nabla_u \mathcal{L}(\boldsymbol{u}, \lambda) = \boldsymbol{\Sigma} \boldsymbol{u} - \lambda \boldsymbol{u} = 0 \qquad (2.2)$$

Equation 2.2 shows that the direction of maximum variance $u$, which we will refer to as the first principal component, is the first eigenvector of the covariance matrix of the dataset. By similar means it can be shown that the second eigenvector points in the direction of largest variance *orthogonal* to the first eigenvector and so forth. Finally it is worth noting that the portion of the total variance explained by a principal component is proportional to its associated eigenvalue.

### 2.1.1 Singular Value Decomposition

For a high dimensional dataset (e.g. $n = 10,000$), which is frequently the case working with for instance image or video data, the covariance matrix will have $10,000 \times 10,000 = 100,000,000$ entries, which is computationally untractable. Hence, PCA is usually implemented in terms of *singular value decomposition* (SVD). For an $m \times n$ matrix $\boldsymbol{X}$, the SVD is a factorization such that:

$$X = USV^T \qquad (2.3)$$

Here, $U \in \mathbf{R}^{m \times m}$, $S \in \mathbf{R}^{m \times n}$, and $U \in \mathbf{R}^{n \times n}$. The SVD relates to the eigenvalue problem (Equation 2.2) as follows:

- The columns of $U$ form the projections of $X$ onto the eigenvectors $V$.

- The entries $s_{ii}$ on the leading diagonal of $S$ are the eigenvalues of $\Sigma = X^T X$.
- The top $k$ columns of $V$ are the top $k$ eigenvectors of $\Sigma = X^T X$

In MATLAB, we can perform SVD by a single line of code (subsequent to standardizing the data to zero mean and unit variance):

```
1   [U,S,V] = svd(X' * X);
```

Figure 2.1: MATLAB code for SVD.

We will not go into the derivation of this result as SCD is covered in most textbooks on linear algebra or basic numerical mathematics. Rather, we will proceed to show how PCA can be applied to BSS, and what assumptions it requires us to make about the data.

## 2.2 PCA Application to Blind Source Separation

The top graph of Figure 2.2 shows two periodic signals $s_1$ and $s_2$ contaminated by an additive Gaussian white noise with standard deviation $\sigma = .2$.

$$\begin{array}{rll} s_1 = & \sin(\pi x) & 0 < x < 5 \\ s_2 = & \cos(7\pi x) & 0 < 5 < x \end{array} \tag{2.4}$$

The signals are subsequently mixed, as shown in the middle part of Figure 2.2 by the matrix:

$$A = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \tag{2.5}$$

where $\alpha = \pi/4$. Here the mixing matrix $A$ corresponds to a rotation operator that will rotate the data by $\alpha$ radians in counterclockwise direction.

The lower part of Figure 2.2 shows the recovered signal[1].

- Example where it works - why
- Example where it fails - why

[1]The estimated unmixing matrix here is $\hat{W} = \hat{A}^{-1} = \begin{bmatrix} .7071 & .7071 \\ -.7071 & .7071 \end{bmatrix}$ which is, as expected, the inverse of $A$ for the value of $\alpha$ above.

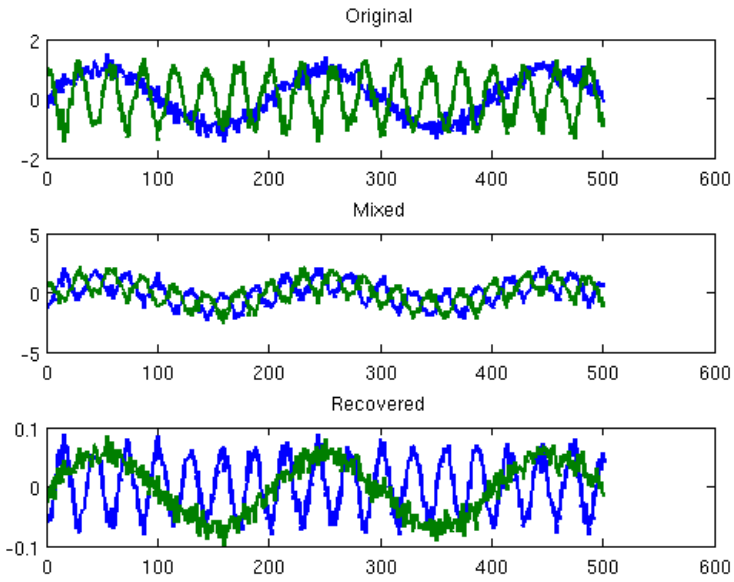


Figure 2.2: PCA Source Separation.

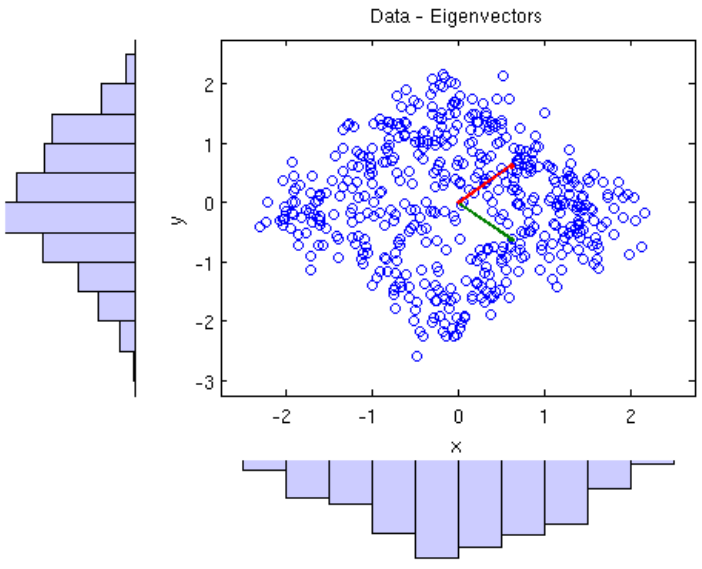


Figure 2.3: Standardized data points vs eigenvectors.

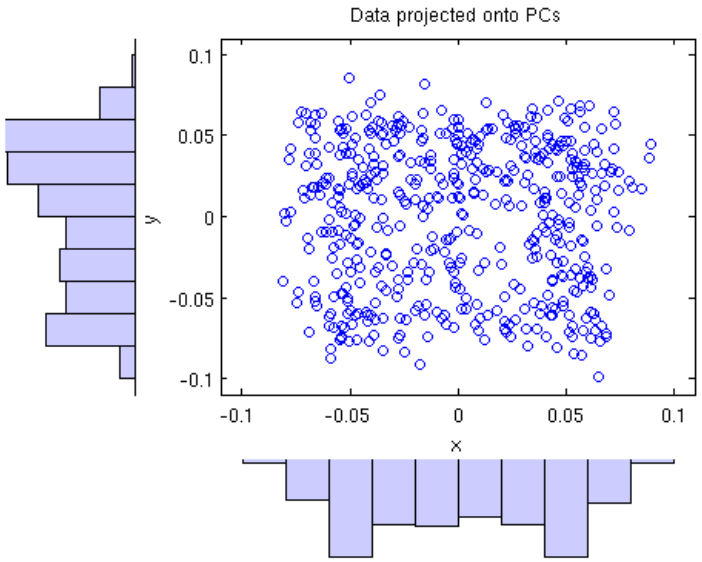


Figure 2.4: Standardized data projected onto eigenvectors.

# Chapter 3

# Independent Component Analysis

PCA finds the basis of a subspace in which the variance is maximized in the direction of the basis vectors and the covariance between the data is zero. ICA seeks to find basis vectors that are statistically independent, which is a stronger property than simply being uncorrelated as independence implies uncorrelatedness, while the opposite is not true. ICA in contrast to PCA does not have analytic solutions in the general case, so a numerical optimization method is usually applied in computing the ICA transform.

## 3.1 Limitations of the ICA Model

ICA imposes a few critical assumptions about the nature of the sources and the extentent to which they can be recovered. As in PCA, we cannot recover the original ordering of the signals; i.e. the rows of the source matrix $\boldsymbol{S}$ may be swapped in the resulting $\hat{\boldsymbol{S}}$. Furthermore, the correct scaling of the source compents, including their sign cannot be recovered. This can be seen in that $\boldsymbol{X} = \boldsymbol{AS} = (.5\boldsymbol{A})(2\boldsymbol{S})$.

The final limitation of ICA is that the source signals must be non-Gaussian. To see why this must hold, we rely on the fact that the multivariate gaussian distribution is rotationally symmetric, and that to fully recover the sources, we must be able to "undo" any rotation caused by applying the mixing operator. Consider a single observation $\boldsymbol{x} = \boldsymbol{x}(t) = \boldsymbol{As}(t) = \boldsymbol{As}$. The covariance matrix of $\boldsymbol{x}$ is

$$\mathbb{E}(\boldsymbol{x}\boldsymbol{x}^T) = \boldsymbol{As}(\boldsymbol{As})^T = \boldsymbol{A}\boldsymbol{s}\boldsymbol{s}^T\boldsymbol{A}^T = \boldsymbol{A}\boldsymbol{A}^T \tag{3.1}$$

Now, let $\boldsymbol{R}$ be a rotation operator and $\boldsymbol{A}' = \boldsymbol{AR}$.

## 3.2 ICA in the Linear Mixing Model

### 3.2.1 Equivalent Specifications of ICA

ICA can be derived by several different approaches:

- Maximum likelihood
- Kurtosis maximization
- Maximum differential entropy
- Blabla..

### 3.2.2 Derivation of ICA Log Likelihood Function

Let $p_s(s_i)$ be the probability density function for source $i$, then, assuming the sources are independent the joint distribution of all the $n$ sources is given by the product of the marginals:

$$p(s) = \Pi_{i=1}^{n} p_s(s_i) \tag{3.2}$$

We now substitute in the unmixing model (Equation 1.3) and obtain:

$$p(s) = \Pi_{i=1}^{n} p_s(WX) \cdot |W| \tag{3.3}$$

The unmixing matrix is the target parameter of our maximum likelihood approach. That is, we seek set the coefficients of the unmixing matrix so as to maximize the likelihood of observing the actual data. If our dataset consists of $T$ observations $X = \{x_1, x_2, ..., x_T\}$, the log-likehliood function is:

$$l(W) = \log Prob(X|W) = \Sigma_{t=1}^{T} \log p_s(WX) + \log |W| \tag{3.4}$$

As the ICA is incompatible with a Gaussian source distribution, common choices for specifying $P_s$ include the sigmoid $p_s(s) = \frac{1}{1+e^{-s}}$ and hyperbolic tangent ($\tanh(s)$).

### 3.2.3 Derivation of Stochastic Gradient Descent for ICA

Given the log likelihood function of Equation 3.4, we will now show how this can be maxmimized by stochastic gradient descent. This derivation leads directly to a working MATLAB implementation shown in Figure **??**.

### 3.2.4 Preprocessing

Whitening transform...

STFT?

```
for i = 1:Niter
  w = update(x,w);
end

function w = update(x,w)
  x=x(:,perm);
  t=1;
  noblocks=fix(P/Blocks);
  BlocksI=Blocks*Id;
  for t=t:Blocks:t-1+noblocks*Blocks,
   u=w*x(:,t:t+Blocks-1);
   w=w+alpha * ( BlocksI + ...
     (1-2*(1./(1+exp(-u))))*u') * w;
 end
end
```

Figure 3.1: MATLAB code for ML ICA by stochastic block gradient descent.

## 3.3 BSS by ICA

## 3.4 Limitations and Comparison with PCA

Refer to section 2.2 in discussion.

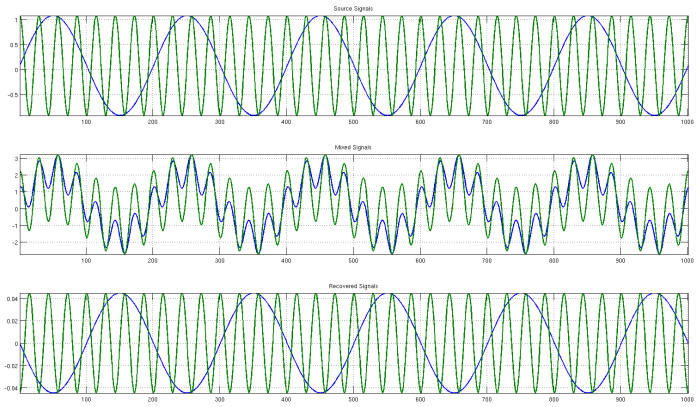


Figure 3.2: ICA on a $2 \times 2$ BSS problem. Note the "sign reversal" for the blue sine wave (cf. Section 3.1).

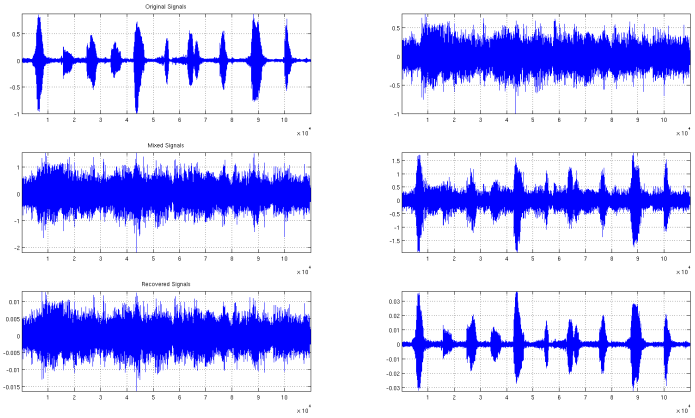


Figure 3.3: Separating a speech signal (top left) from background music (top right) by ICA. Here we also observe that the sign of the original speech signal is reversed in the bottom right recovered signal.

# Chapter 4

# Single Sensor Blind Source Separation

Single sensor BSS[1] is a particularly importnat case of the BSS problem where the observed signal consists of only a scalar value at any point in time as if the source signals were recorded a sole microphone. This presents us with particular challenges, and we often need to make further assumptions about the data generating process – i.e. we need a more complex generative model.

In this chapter we present a solution to the single sensor BSS problem proposed by Roweis (XXXX)[2] that relies on a factoral hidden markov model system. The key idea put forward by Roweis is to learn the transition and sensor models for every source separately using a time-frequency representation of the original signal.

This chapter is structured as follows. [TODO]

## 4.1 Time Frequency Signal Representation

A time frequency representation (TFR) is a redundant signal representation in comparison to a simple time domain representation that contains only the amplitude values at given point in time. A TFR is often preferred to a perfect frequency or Fourier domain representation as the latter contains no information about the temporal location of events.

```
1 window = 256; noverlap=250; nfft=256; Freq = 16000;
2 [S,F,T,P] = spectrogram(s1,window,noverlap,nfft,Freq);
```

Figure 4.1: MATLAB code for computing the short term Fourier transform of a signal sampled at 16kHz with given parameter. P is here the power spectral density (PSD) of the signal.

[1] Also called single channel BSS.

[....]

## 4.2 Factoral Hidden Markov Model for BSS

[...] tekst og figurer...

### 4.2.1 Initialization

The factoral hidden markov model consists of one HMM per speaker which is trained on separate non-mixed training data for each source.

The initialization of the FHMM training consists in estimating the emission probabilities $P(X|Z)$ (see figure, lag figur). While Roweis operates with a finite state model for the latent variables, the observable variables are real valued intensities. This indicates that a mixture model may be appropriate in the inital estimate of the emission model (vis til andre artikler med samme greier ).

We follow Roweis in estimating a Gaussian mixture model with a single shared covariance matrix $\mathbf{\Sigma}$. For a spectrogram with $N$ frequency bands, our approach is to estimate a GMM with $k$ $N$-dimensional components or latent variables. The mean vector $\mu_i \in \mathbb{R}^N$ for each component $i$ represents the expected intensity (power spectral density value) in each frequency band given that the system is in state $i$. The pair $(\{\mu_i\}, \Sigma)$ then forms the initial parametrization of the emissions model.

# Chapter 5

# Conclusion

todo.

# Appendix A

# Mathematical Concepts

In this appendix we will provide a brief background on some of the mathematical notions that are central to understanding the methods used in this report.

## A.1 Linear Algebra

In this section we will define a few important concepts that are necessary. These concepts are particularly important for understanding PCA, but are also relevant in the analysis of markov models. With stationary transition probabilities, the steady state distribution of the system is the solution to the eigenvector-eigenvalue problem.

**The Eigenvector - Eigenvalue Problem**

**Singular Value Decomposition**

## A.2 Statistics and Optimization

A large portion of machine learning relies on statistical methods; the methods considered in this report being no exception. As the learning problems having relevance in practical life are often far too complex to describe by analytic formulae, we therefore often need mathematical methods to find models that have the best fit to the observed data.

In this section, we therefore consider one very important method for finding "optimal" parameters in a given model; the maximum likelihood method which is presented in Section A.2.1. We then proceed to show how to actually solve the resulting maximzation in Section A.2.2.

### A.2.1 Maximum Likelihood Estimation

ML estimation is a method for determining the parameters of a statistical model by setting the parameters so as to maximize the *likelihood* of observing the actual data under the given model. Denoting $f(\boldsymbol{X}|\boldsymbol{\Theta})$ the probability distribution of $\boldsymbol{X} = \{x_1, x_2, ..., x_n\}$ with parameters $\boldsymbol{\Theta} = \{\theta_1, \theta_2, ..., \theta_m\}$, the maximum likelihood estimate of $\Theta$ solves EquationA.1

$$\arg\max_{\Theta} f(\boldsymbol{X}|\boldsymbol{\Theta}) \tag{A.1}$$

An important case is if the $x_i$ are i.i.d., where the joint density is the product of the marginal densities. This means we can write the likelihood function $f$ as Equation A.2.

$$f(\boldsymbol{X}|\boldsymbol{\Theta}) = \prod_{i=1}^{n} f(x_i|\Theta) \tag{A.2}$$

The solution to an optimzation problem is the same under any monotone transformation. Therefore it is often times useful to deal with the log-likelihood function instead of $f$ directly.

### A.2.2 Mathematical Optimization

**The Lagrange Multiplier Method**

**Gradient Descent**

## A.3 Spectral Analysis

In working with hidden markov models for blind source separation, we rely on a redundant signal representation in the time-frequency domain, rather than the standard representations in the time domain. Such a representation is advantageous in comparison with a pure spectral representation as the latter contains no information on when different components of a signal occur in time. While many different time-frequency representations exist, our presentation relies on the *short-term Fourier transform* (STFT).

The time-frequency representation, often called a *spectrogram* produced by the STFT maps the energies in various parts of the spectrum over the timespan of the signal. For a low amplitude portion of the signal will have its energy concentrated in the upper part of the spectrogram and vice versa. This is illustrated in Figure XX.

[[[[[[[[[ FIGURE HERE!!! ]]]]]]]]]

### A.3.1 Formal statement

Equation A.3 defines the discrete time STFT for the $n$th segment (which is centered around $m$):

$$\text{STFT}\{x[n]\}(m,\omega) = X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]\omega[n-m]e^{-i\omega n} \tag{A.3}$$

Here, $\omega$ is a zero centered window function, typically uniform or gaussian. The window determines which part of the signal $x$ is to be included in the spectrogram near $m$. In practice, the STFT is computed using the fast fourier transform (FFT). To better allow for visualization of the STFT, which is a complex number, the spectrogram is defined as the squared magnitude ofthe STFD (Equation A.4).

$$\text{spectrogram}\{x[n]\}(m,\omega) = |X(m,\omega)|^2 \tag{A.4}$$

## A.4 Hidden Markov Models

A hidden markov model (HMM) is a probabilistic model relating two sequences of discrete random variables $\mathcal{S} = \{S_1, S_2, ..., S_T\}$ and $\mathcal{X} = \{X_1, X_2, ..., X_T\}$. Here, we will refer to $\mathcal{S}$ as the *source* or *hidden* variable, and $\mathcal{X}$ as the *observed* variable. Often, we assume there is some causal relationship whereby the hidden variable affects the observable, but this does not need be the case.

A HMM consists of two probabilistic statement; the *transition* model:

$$\mathbb{P}(S_t|S_{t-1}, S_{t-2}, ..., S_1) \tag{A.5}$$

and the *sensor* or *observation* model:

$$\mathbb{P}(X_t|S_t, S_{t-1}, X_{t-1}, ..., S_1, X_1) \tag{A.6}$$

The order of a markov model is the number of realizations of $\mathcal{S}$ conditioned on in the transition model. In order to make HMMs computationally tractable, we often operate with 1st order markov models:

$$\mathbb{P}(S_t|S_{t-1}, S_{t-2}, ..., S_1) = \mathbb{P}(S_t|S_{t-1}) \tag{A.7}$$

This can be stated as as "the future being conditionally independent of the past given the present". Another common simplification is known as the *markov sensor model assumption*:

$$\mathbb{P}(X_t|S_t, S_{t-1}, X_{t-1}, ..., S_1, X_1) = \mathbb{P}(X_t|S_t) \tag{A.8}$$

The sensor markov assumption states that the sensor is independent of everything else given the current value of the hidden variable.

# Bibliography

[1] Pearson, K. (1901). "On Lines and Planes of Closest Fit to Systems of Points in Space". Philosophical Magazine 2 (6): 559–572.

[2] Roweis, S. T. (XXXX). "One Microphone Source Separation". ????

[3] Russell, S. and Norvig, P. (XXX). "Artificial Intelligence: A Modern Approach". XXXX