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 $S(t) \in \mathbf{R}^n$ for $t > 0, n > 0$ denote the signals generated by n sources. Similarly, we let $X(t) \in \mathbf{R}^m$ for $t > 0, n > 0$ the observed sensor readings resulting from the emitted signals. A *mixing model* $f(S, t)$ defines the relationship between source and observed signal:

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

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

1.1.1 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 SSMSS problem, we have one or more source signals, but the observed signal $X(t)$ 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.

1.1.2 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:

$$X = AS \tag{1.2}$$

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

$$S = WX \tag{1.3}$$

can be recovered. This is to say that the estimate of the original signal j at time t is computed as the jth row of W times the tth column of X.

From Equation 1.2, we can see that the blind source separation problem, even in this simplest case, is ill-poised, that is we are trying to determine both A and S given only X. This implies that we need to impose certain assumptions on the nature of the data. 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 terms of the assumptions we make, hence the term *blind* source separation. The type of assumptions made are primarily related to statistical properties of the sources, such as for instance statistically independence, an assumption that leads to the ICA solution.

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.

Method	**Data Characteristic**	**Description**
PCA	Blabla	Blablabla

Table 1.1: Overview over different approaches to blind source separation.

Chapter 2

Principal Component Analysis

Principal component analysis (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, before we finally look at a non-linear extension of PCA.

2.1 Formal Statement

Let $x_i \in \mathbf{R}^n$ denote the i'th observation of a dataset of m observations. We now want to project our data onto a vector u in $\mathbf{R}^n$ so as to maximize

the variance of the resulting projection $\sum_{i=1}^{m} x_i^T u$ subject to the constraint $|u| = 1$. Under the assumption that X is standardized to zero mean and unit variance, the Lagrangian is then given by Equation 2.1:

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

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

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

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.

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 X, the SVD is a factorization such that:

$$X = USV^T \tag{2.3}$$

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 $\Sigma = X^T X$

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

Figure 2.1: MATLAB code for SVD.

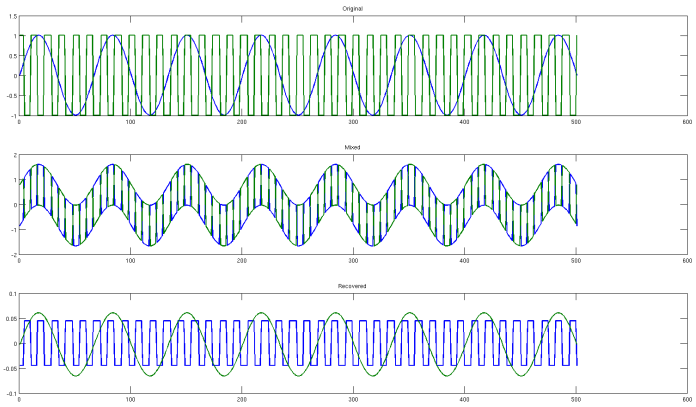

Figure 2.2: PCA Source Separation.

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

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

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

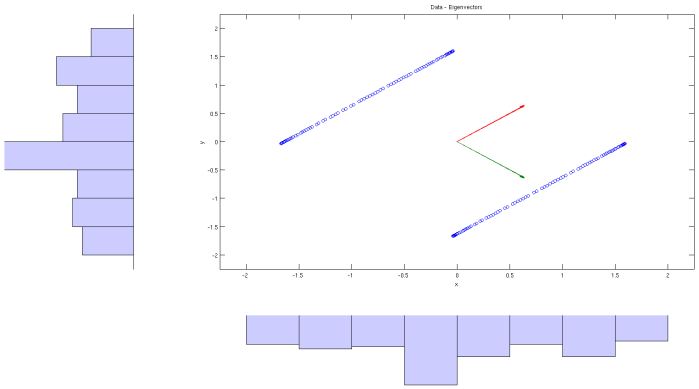

Figure 2.3: Standardized data points vs eigenvectors.

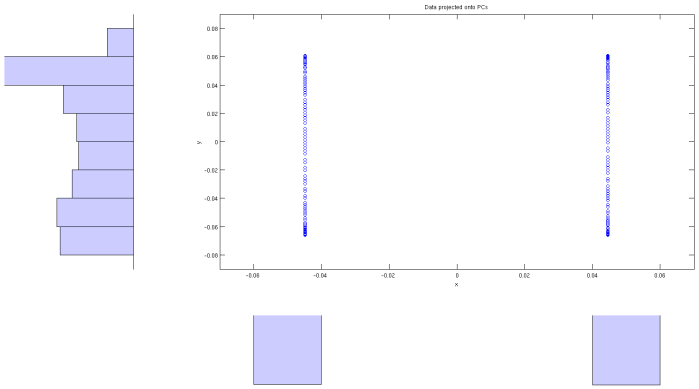

Figure 2.4: Standardized data projected onto eigenvectors.

Chapter 3

Independent Component Analysis

We have seen that 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 on the other hand, seeks to find basis vectors that are statistically independent. This 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

Blabla

- Non-Gaussian..
- Ordering of signals..
- "Sign reversal" (rotational invariance??)..
- Blabla..

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 Maximum Likelihood Derivation

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:

$$p(s) = \Pi_{i=1}^{n} p_s(s_i) \tag{3.1}$$

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

$$p(s) = \Pi_{i=1}^{n} p_s(WX) \cdot |W| \tag{3.2}$$

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:

$$l(W) = \log Prob(X|W) = \Sigma_{t=1}^{T} \log p_s(WX) + \log |W| \tag{3.3}$$

As the ICA is incompatible with a Gaussian source distribution, a common choice for specifying P_s is either the sigmoid $p_s(s) = \frac{1}{1+e^{-s}}$, or the laplacian $p_s(s) = \frac{1}{2}e^{-|s|}$.

```
1    code here...?
```

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

3.2.3 FastICA

```
1    code here...?
```

Figure 3.2: MATLAB code for FastICA algorithm.

3.2.4 Preprocessing

Whitening transform...
STFT?

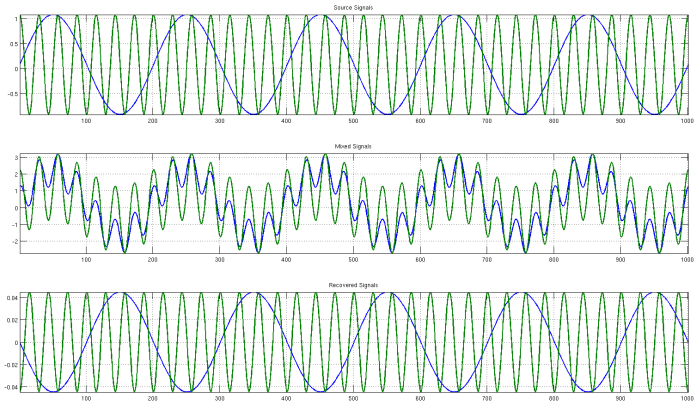

Figure 3.3: ICA on a 2×2 BSS problem. Note the "sign reversal" for the blue sine wave (cf. Section 3.1).

3.3 BSS by ICA

3.4 Limitations and Comparison with PCA

Refer to section 2.2 in discussion.

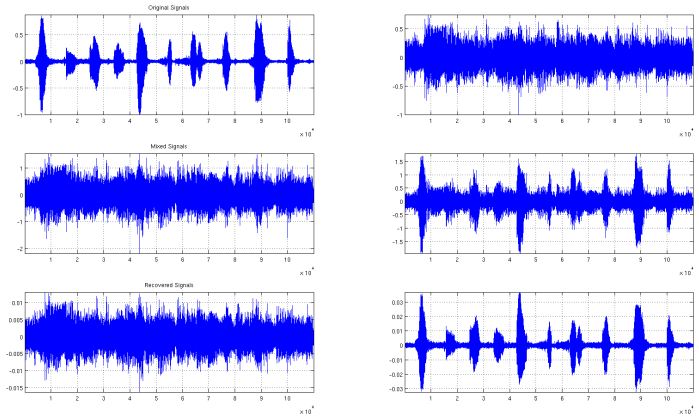

Figure 3.4: Separating a speech signal (top left) from background music (top right) by ICA. Here we also observe that the sign of the original speech signal is reversed in the bottom right recovered signal.

Chapter 4

Single Sensor Blind Source Separation

todo.

Chapter 5

Conclusion

todo.

Bibliography

todo.