Blind Source Separation for Musical Recordings

Dmitry Lekhovitsky, Illia Kachko

Abstract

In this project, we consider the problem of blind source separation – given multiple observations of a linearly mixed signal, we aim to restore the mixing matrix and original sources. There are two main methods to perform this problem that heavily rely on linear algebra, namely, Independent Component Analysis (ICA) and Non-negative Matrix Factorization (NMF). We compare their performance against artificially mixed signals. Particularly, we apply them to the problem of separating a musical recording into individual instruments, which might be useful for learning and analysis of tracks. The main restriction of the classical approaches is the requirement that the number of recordings must be equal to the number of sources, which is rarely the case, so in our future work, we plan to remove this restriction by applying deep learning methods.

1 Introduction

Blind source separation [1] (or blind signal separation) is a classical problem in signal processing that lies in restoring the original sources from multiple mixed observations. The typical example is a "cocktail party problem": a few people talk in the same room simultaneously and we need to extract the speech of every person given multiple recordings from different microphones placed in that room.

There are multiple methods for solving this task. Mostly, they reduce to classical Linear Algebra algorithms: Principal Components Analysis [2], Independent Component Analysis [3] (which was developed particularly for this problem), Non-negative Matrix Factorization [4], etc. We describe in detail and compare the performance of second and third of them. Besides classical solutions, there exist some attempts to apply Machine Learning tools [5] to address the limitations of the former.

An interesting application of the problem is separating a musical recording into individual tracks (vocals, guitar, bass, drums) with a hope that the considered classical methods are able to produce a reasonable solution. The quality might be assessed directly by listening to the produced tracks (each of them must represent a particular instrument without, desirable, notable artifacts), but we also consider some domain-specific metrics that measure timbre (a characteristics that makes each instrument sound unique), which are described in more detail in section 2.

As it will be described later, to separate n instruments with the chosen methods, we need exactly n recordings (say, microphones in different locations of a studio). Since such data is usually unavailable, we found a few tracks that are not yet mixed and artificially created the required number of observations for each of them. Our future work will address this issue by using such data as supervisor to a deep learning model that is able to perform separation from just one recording.

2 Problem statement

Formally, we denote each observed recording as $\mathbf{x}_i$ and each latent original signal as $\mathbf{s}_j$. Then, linear mixing procedure might be written as

$$\mathbf{x}_i = a_{i1}\mathbf{s}_1 + \ldots + a_{in}\mathbf{s}_n, i = \overline{1, m},$$

or, in a matrix form,

$$X = AS,$$

where $A = [a_{ij}]$ is called a mixing matrix and is unknown. Usually, we assume that the number of recordings m equals the number of original sources n.

Our problem is then to find an estimate of the "unmixing matrix" $W = A^{-1}$ and restore original sources $S = WX$.

3 Solution approaches

This section explains the chosen solution approaches.

3.1 ICA

ICA description

Algorithm 1 FastICA algorithm

Input: C Number of desired components

```
1: procedure FastICA
2:     stringlen ← length of string
3:     i ← patlen
4:     top:
5:     if i > stringlen then return false
6:     j ← patlen
7:     loop:
8:     if string(i) = path(j) then
9:         j ← j − 1.
10:        i ← i − 1.
11:        goto loop.
12:        close;
13:    i ← i + max(delta1(string(i)), delta2(j)).
14:    goto top.
```

3.2 NMF

NMF description.

4 Numerical experiment

4.1 Data

We used data from an online source Puremix, that provides a set of individualy recorded instruments from a single track. For our experiment we used *Iron Sheik – Dry Clean Only* track. It contains a 30 unique recordings in a .wav format: spacing

- 16 drums
- 5 guitars
- 9 vocals

4.2 Input

We have picked one audio from each class and mixed them in a different combinations.

Let S be a $N \times K$ matrix, where K is a number of independent signals and N be a length of each signal:

$G, V, D - -$vectors of a guitar, vocal and drums signals

$$S = \begin{bmatrix} G & V & D \end{bmatrix}$$

Then, we used a $K \times K$ coefficient matrix H that will multiply each instrument by a coefficient and sum them up.

For out experiment we used matrix with such coefficients:

$$H = \begin{bmatrix} 0.2 & 0.6 & 0.4 \\ 0.5 & 0.2 & 0.1 \\ 0.3 & 0.2 & 0.5 \end{bmatrix}$$

So our resulting matrix of mixed tracks is:

$$X = HS$$

that can be interpreted like this:

$X_i = V * H_{i,1} + G * H_{i,2} + D * H_{i,3}$ – i-th mixed track

4.3 Tools

To compute a numerical experiments we used a **sklearn.decomposition** module.

For an ICA we used **sklearn.decomposition.FastICA**

For an NMF we used **sklearn.decomposition.NMF**

5 Results

This is results comparison section.