CS391L Machine Learning HW2: Independent Component Analysis

Zeyuan Hu, iamzeyuanhu@utexas.edu

EID:zh4378 Fall 2017

1 Introduction

In this task, we are asked to perform blind source separation by applying the Independent Component Analysis (ICA) to sounds data. In details, suppose there exists an n by t matrix U of n source signals of length t and we have an m by t matrix X of m mixed signals ($m \geq n$) of length t that consist of different linear mixtures of U, then, we can recover the original signals U under certain condition.

This writeup is organized in the following way: In the first section, I will describe the steps to finish the task. Next, I will talk about the experimentation result. Lastly, I will give a breif summary of the whole work.

2 Method

In this section, I will talk about how we work with the raw data, how we perform the ICA on the data, and how we carry out the experiment.

2.1 Work with the data

In this task, we work with two data files: `sounds.mat` and `icaTest.mat`. The first dataset contains a varaible called `sounds`, which is 5 by 44000 matrix. Each row represents a roughly four second sound clip and these signals are not mixed. The second dataset contains two matrices: U, which is a 3 by 40 matrix and A, which is 3 by 3 matrix. Again, the signals in this dataset is also unmixed. The first thing to do is to create the mixed signal matrix X. I apply two approaches: for the `sound` matrix, I randomly create the mixed signal matrix by randomly creating the original signal matrix, and matrix A, which is an m by n matrix such that $A_{i,j}$ is the weight of the jth source signal in the ith mixed signal. The details of the implementation can be found in function `mixSignal` in `hw2.py`. The second approach is to use the A directly coming from `icaTest.mat` and generate the mixed signal matrix X with $X = AU$ using the U from `icaTest.mat` as well. One important step is to convert the datatype of all the matrices into `float32` in Python, which allows us to keep precision when we do the calculation.

2.2 Algorithm Implementation

In order to uncover the original signals from the mixed signals, we assume that there is no correlation between source signals and so any correlation between different mixed signals is due to a common signal showing through the mixture. Then, our task is to find a matrix W that recovers the original n source signals (possibly in a different order and with different scale factors). In this project, I use gradient descent method to decrease the mutual information between signals and thus cover the original signals:

1. Assume $X = AU$.
2. Initialize the (n by m) matrix W with small random values.
3. Calculate $Y = WX$.
4. Y is our current estimate of the source signals.
5. Calculate Z where $z_{i,j} = g(y_{i,j}) = 1/(1 + e^{-y_{i,j}})$ for $i \in [1 \dots n]$ and $j \in [1 \dots t]$ (where t is the length of the signals). This helps us traverse the gradient of maximum information separation.
6. Find $\Delta W = \eta(I + (1 - 2Z)Y^T)W$ where η is a small learning rate.
7. Update $W = W + \delta W$ and repeat from step 3 until convergence or some max iterations reached.

The implementation of the algorithm can be found in function `ica`. There are a couple of implementation details need to notice:

- We use uniform distribution between 0 and 1 to generate the initial weights W.
- I setup a threshold value H, which equals to 1×10^{-9}. Let our unmixing matrix in i-th iteration be W_i. Let the norm of difference of unmixing matrices in two neighbor

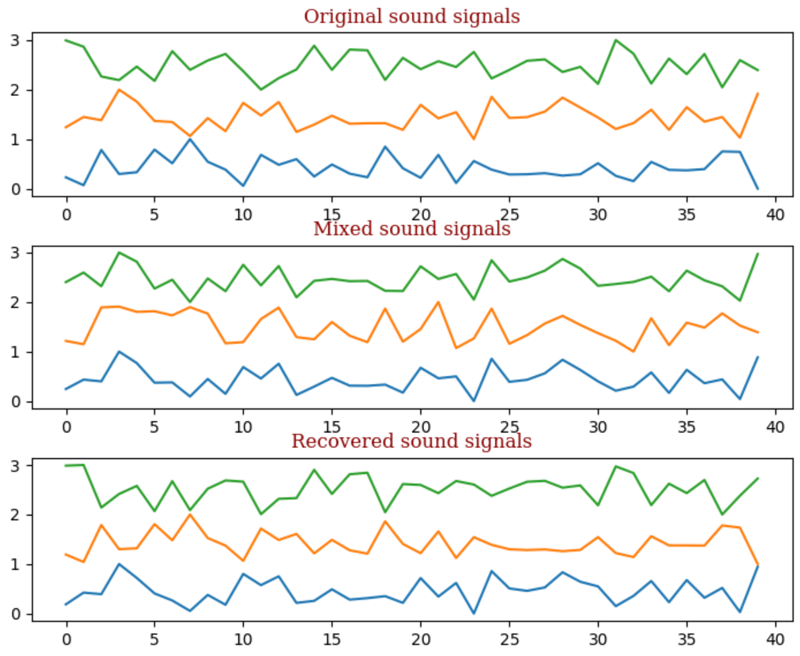


Figure 1: ICA on icaTest.mat

iterations be Δ. Then, in addition to the max iterations termination condition, the algorithm can also stop when $\Delta < H$. In the implementation, I set the max iterations to 1000000 and the algorithm usually stops at around 6000 iterations.

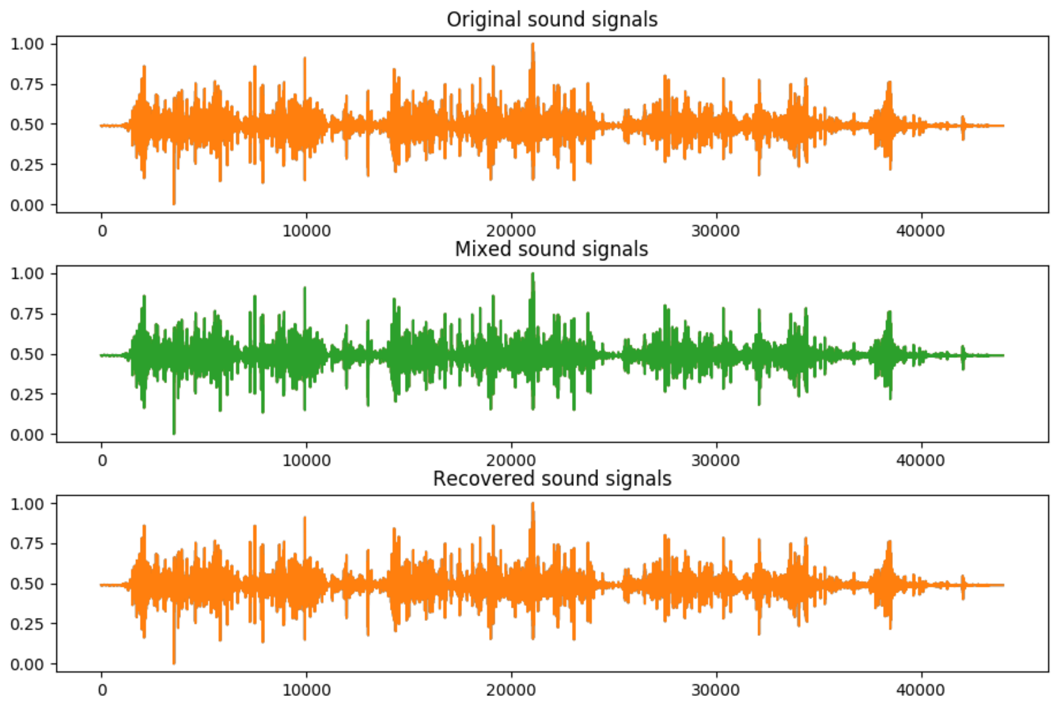
- The learning rate η I set in the implementation is 0.01. After several experimentations, I find this learning rate gives the best result in the recovered signals.

3 Results

I run the ICA algorithms on both `icaTest.mat` and `sounds.mat` datasets and plot the original signals, the mixed signals, and the recovered signals.

3.1 IcaTest.mat

Once we implement our algorithm, we want to verify if it works on a small set of data. As described in the section 2.1, `icaTest.mat` is the ideal candidate for this purpose. The figure 1 is what I get after running ICA algorithm on this dataset:



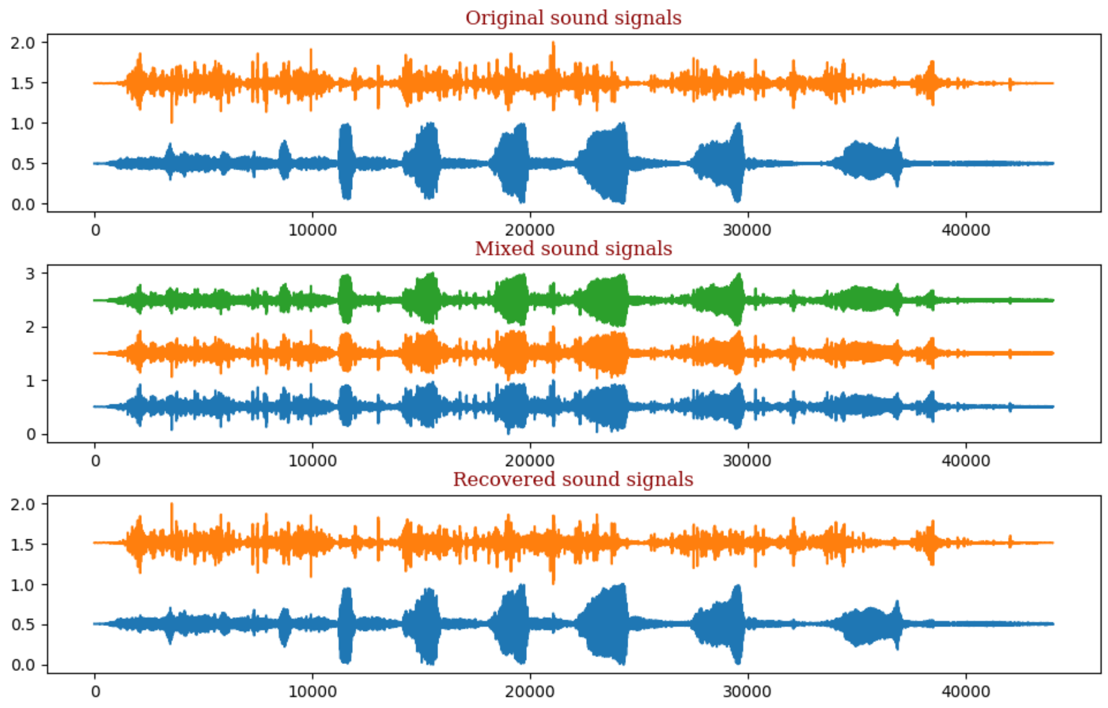
The result heavily relies on the learning rate η and the number of iterations we perform the gradient descent. In this experiment I find out that once the Δ belows the threshold value H, there is not much gain from performing extra iterations of gradient descent.

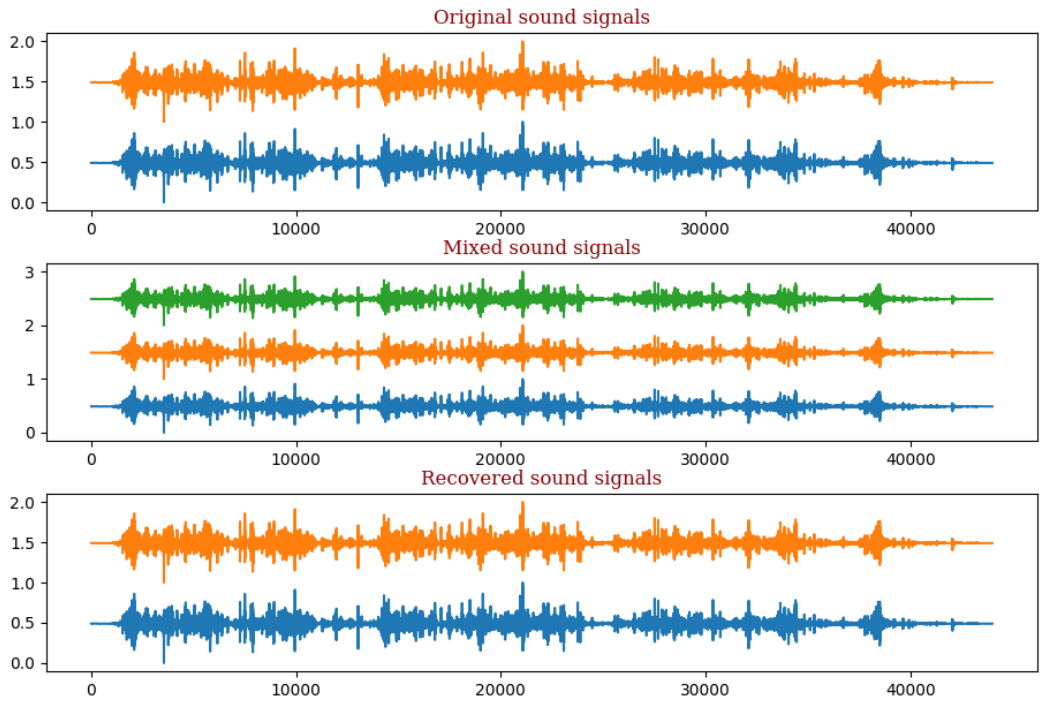
3.2 Sounds.mat

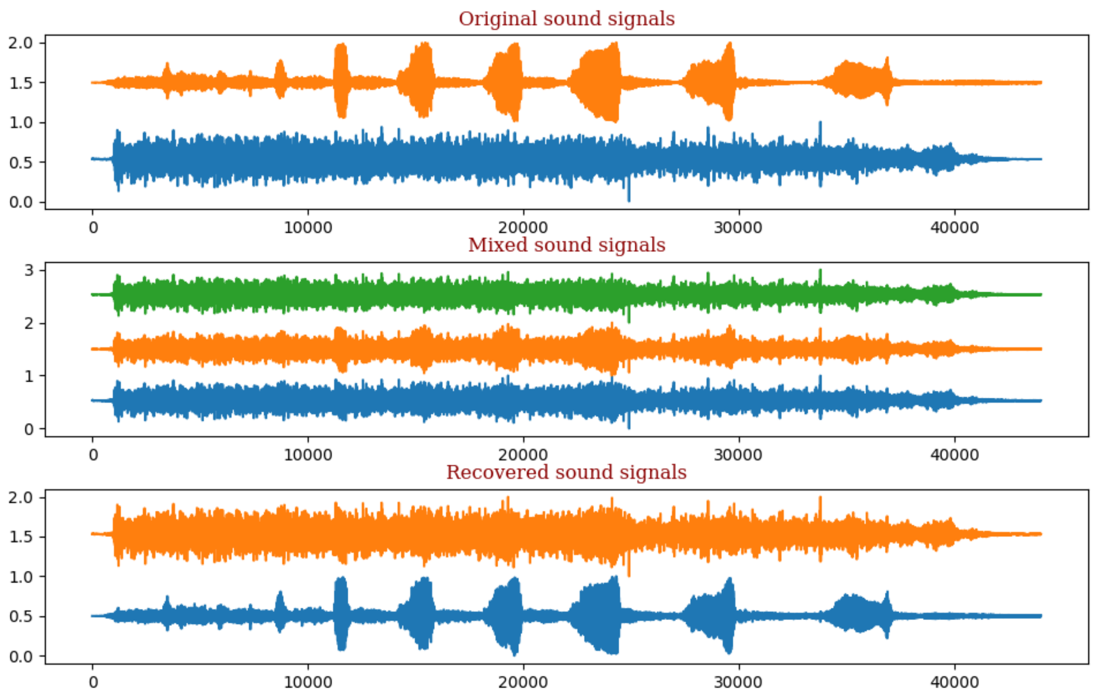
`sounds.mat` offers an opportunity for us to test out our ICA algorithm in a "big data" setting and the result is quite good for my implementations. Figures 2-6 show the plot of the original signals, the mixed signals, and the recovered signals under the different kind of mixture.

4 Summary

In this task, I apply ICA method to sound data to perform blind source separation. I perform the algorithm using different size of data and find out the performance of the algorithm is quite good under the both situations. In addition, I tune the learning rate, the number of iterations, and the threshold value in the gradient descent in order to get the


Original sound signals
Mixed sound signals
Recovered sound signals


Original sound signals
Mixed sound signals
Recovered sound signals



best performance of the algorithm.

Appendices

A How to run the code

To run my code, unzip the `hw2.zip` and put the data file `sounds.mat` and `icaTest.mat` under the same directory with `hw2.py`. Then, you can run the code with the command `python3 hw2.py ICATEST` or `python3 hw2.py SOUNDS`. The former command applies ICA towards `icaTest.mat` dataset and the latter command applies ICA towards `sounds.mat` dataset. The output of both commands will be the plot of the original signals, the mixed signals, and the recovered signals. My code is heavily commented and please take a look if there are any type of questions.