CS 391L Machine Learning Homework 2: ICA

Yuege Xie, EID: yx4256, Email: yuege@ices.utexas.edu*

1 Introduction

Blind Source Separation is the separation of a set of mixed source signals, with very little information about the source signals or the mixing process (from Wikipedia). Assume the original source signals are in $\mathbf{U}$, where each row $\mathbf{u_i}$ is an individual source signal. The observation (mixed source signals) is given by $\mathbf{X} = \mathbf{AU}$, where $\mathbf{A}$ is the ground truth matrix for mixing signals. The goal is to find a matrix $\mathbf{W} \approx \mathbf{A^{-1}}$ without any information of $\mathbf{A}$ such that $\mathbf{WX} \approx \mathbf{A^{-1}AU} = \mathbf{U}$, so we can recover the source signals by $\mathbf{U} \approx \mathbf{WX}$. One of the most classical examples is to separate mixed people's talking sound source signals from a cocktail party. Other examples include image processing and separation of musical signals. In practice, we often use unsupervised learning methods such as independent component analysis (ICA), principal components analysis (PCA) and singular value decomposition (SVD), to do the blind source separation task due to the lack of information. This project mainly focuses on using ICA to separate targeted sound source signals.

Independent Component Analysis (ICA) aims at decomposing a multivariate signal into independent non-Gaussian signals. The basic idea in this project is using natural gradient descent algorithm based on maximum likelihood . More details are in "Method" section. If the source signals satisfy two important assumptions: independence and non-Gaussian, then ICA separates the mixed source signals well. In this project, the original source signals are in Figure 1, and I verified the correlation in Figure 2 (The matrix is approximate to identity matrix, so they are almost independent) and distributions of the signals in Figure 3 (The third one is similar to Gaussian but the top part is away from Gaussian, and others are obviously non-Gaussian).

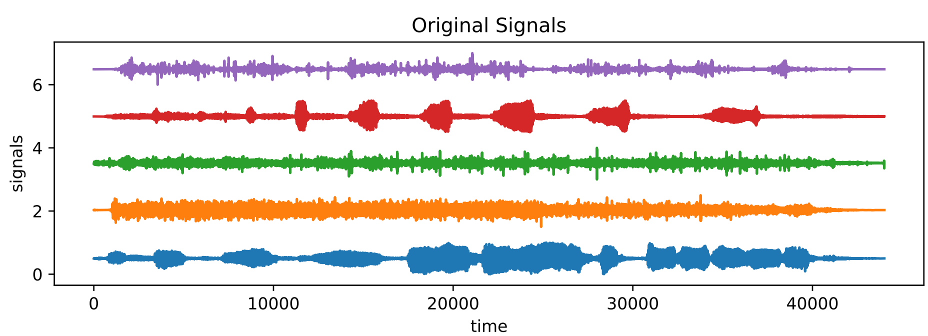

Figure 1: Original Source Signals.

```
S = loadmat('./sounds.mat')['sounds']
np.corrcoef(S)
```

```
array([[ 1.        , -0.0053306 , -0.00315613,  0.00549875, -0.00383568],
       [-0.0053306 ,  1.        ,  0.00123874,  0.00698466, -0.00109667],
       [-0.00315613,  0.00123874,  1.        , -0.01758172,  0.00801588],
       [ 0.00549875,  0.00698466, -0.01758172,  1.        ,  0.00502297],
       [-0.00383568, -0.00109667,  0.00801588,  0.00502297,  1.        ]])
```

Figure 2: Correlation matrix of signals.

*Oden Institute for Computational Engineering and Sciences, UT Austin.

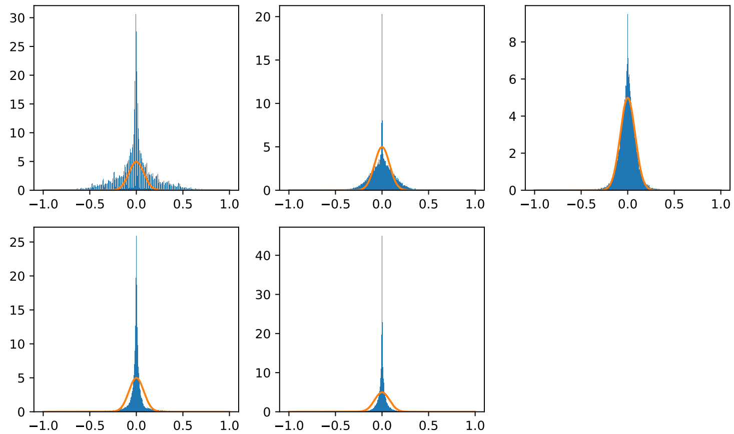

Figure 3: Distributions of original signals.

2 Method

In this report, we focus on using ICA to do blind source separation. The detailed steps are as follows:

1. **Prepare data (get and mix):** I load the source signals from ".mat" file as source matrix $\mathbf{S}$, generate a random mixing matrix $\mathbf{A}$ s.t. $\mathbf{A_{ij}} \sim \mathcal{N}(0,1)$, and then use $\mathbf{X} = \mathbf{AU}$ to get the mixed source signals. Hence, our ground truth signals (the goal of recovery is $\mathbf{S}$).
2. **Implement ICA to recover signals:** Our ICA algorithm use the maximum likelihood estimation. The log of likelihood function is given by $L(W) = \sum_{i=1}^{m}(\sum_{j=1}^{n} \log g'(w_j^T x^{(i)}) + \log |W|)$. We use (natural) gradient descent algorithm to optimize the problem. The implementation details are in Algorithm 1. I set $\epsilon = 1e-12$ as convergence threshold for $\|\mathbf{W^{(t+1)}} - \mathbf{W^{(t)}}\|_\mathbf{F}$ or $\|\mathbf{\Delta W^{(t)}}\|_F < \epsilon$ and $T = 1000000$ as the max iterations, where $\|.\|_F$ is the Frobenius norm. Note that in the homework description, there is not d in front of the identity matrix, but the results without d, which is the length of signals, are not good. By adding the scalar from TA's instruction, the recovery results are surprisingly good.

Algorithm 1 Independent Component Analysis (ICA) Algorithm

Input: Initialize $\mathbf{W^{(0)}} \in \mathbb{R}^{n \times n} \sim Unif(0, 0.1)$, learning rate $\eta > 0$, max iteration $T > 0$, stop threshold $\epsilon > 0$
for $t = 0, 1, \ldots, T$ **do**
    $\mathbf{Y^{(t)}} = \mathbf{W^{(t)}X}$ # current estimate source signal
    $z_{i,j}^{(t)} = 1/(1 + e^{-y_{i,j}^{(t)}})$ # get matrix $\mathbf{Z^{(t)}}$
    $\Delta\mathbf{W^{(t)}} = (\mathbf{dI} + (\mathbf{1} - \mathbf{2Z^{(t)}})\mathbf{Y^{(t)T}})\mathbf{W^{(t)}}$ # with d as the dimension of signal
    $\mathbf{W^{(t+1)}} = \mathbf{W^{(t)}} + \eta\mathbf{\Delta W^{(t)}}$
    if $\|\mathbf{W^{(t+1)}} - \mathbf{W^{(t)}}\|_\mathbf{F} < \epsilon$ or $\|\mathbf{\Delta W^{(t)}}\|_F < \epsilon$ **then**
        break

3. **Test for blind source separation:** Test on "icaTest.mat" and "sounds.mat" dataset, and the results are Section 3.

3 Results

The code, plots and more example of eigendigits and corresponding reconstruction are in the appendix "hw2-ica.ipynb". I show and analyze the plots in the following subsections.

3.1 Results on "icaTest.mat"

On the "icaTest.mat" dataset, the recovered signals are almost the same as the original ones. I used $\mathbf{W_0} \sim Unif(0, 0.1)$ with learning rate $\eta = 0.01$. It early breaks at iteration: $T = 379$ with gradient norm = $9.86e - 11$. The plots of signals are in Figure 10 and the correlations between recovered and original signals are all around 0.99 (Figure 4). I calculated the inner correlations of original (up) and recovered (down) signals of test set in Figure 5. Although they are not independent, the inner correlations of original and recovery signals are quite similar.

```
Correlations between orignal and recovered signals of test set:
singal 0: 0.9866785786048746
singal 1: 0.988386527240567
singal 2: 0.991185036478111
```

Figure 4: Correlations between original and recovered signals of test set.

```
np.corrcoef(test['U'])

array([[ 1.        , -0.42146365, -0.42193968],
       [-0.42146365,  1.        , -0.48858069],
       [-0.42193968, -0.48858069,  1.        ]])

np.corrcoef(recover_test)

array([[ 1.        , -0.50842141, -0.51647105],
       [-0.50842141,  1.        , -0.45067516],
       [-0.51647105, -0.45067516,  1.        ]])
```

Figure 5: Inner correlations of original (up) and recovered (down) signals of test set.

3.2 Results on "sounds.mat"

On the "sound.mat" dataset, the recovered signals are surprisingly almost the same as the original ones as well. I used $\mathbf{W_0} \sim Unif(0, 0.1)$ with learning rate $\eta = 0.00001$. It early breaks at iteration: $T = 582$ with gradient norm = $9.67e - 08$. The plots of signals are in Figure 9 and the absolute value of correlations between recovered and original signals are all over 0.99 (Figure 6). The inner correlations of original (Figure 2) and recovery (Figure 7) signals are close to identity matrix, which means that both the original and recovered signals are almost independent.

```
Correlations between orignal and recovered signals of sound set:
singal 0: 0.9999649090932449
singal 1: 0.9999279577812625
singal 2: -0.99986749543762
singal 3: 0.9999379639280448
singal 4: -0.9999383517240533
```

Figure 6: Correlations between original and recovered signals of sound set.

```
[[ 1.00000000e+00 -1.57430481e-03 -2.05467443e-03  8.02515027e-04
   1.83378606e-03]
 [-1.57430481e-03  1.00000000e+00  1.09168410e-03 -6.95436219e-04
   1.22378966e-03]
 [-2.05467443e-03  1.09168410e-03  1.00000000e+00  1.58994994e-03
   2.08975775e-03]
 [ 8.02515027e-04 -6.95436219e-04  1.58994994e-03  1.00000000e+00
  -3.05699948e-04]
 [ 1.83378606e-03  1.22378966e-03  2.08975775e-03 -3.05699948e-04
   1.00000000e+00]]
```

Figure 7: Inner correlations of recovered signals of sound test set.

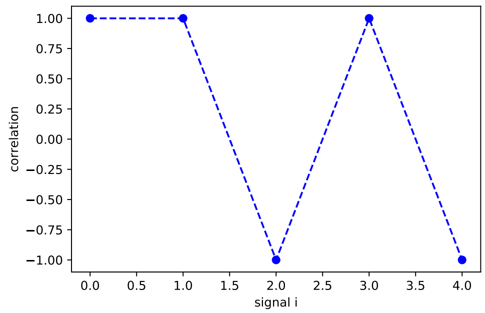

Figure 8: Plot of correlations between original and recovered signals of sound set.

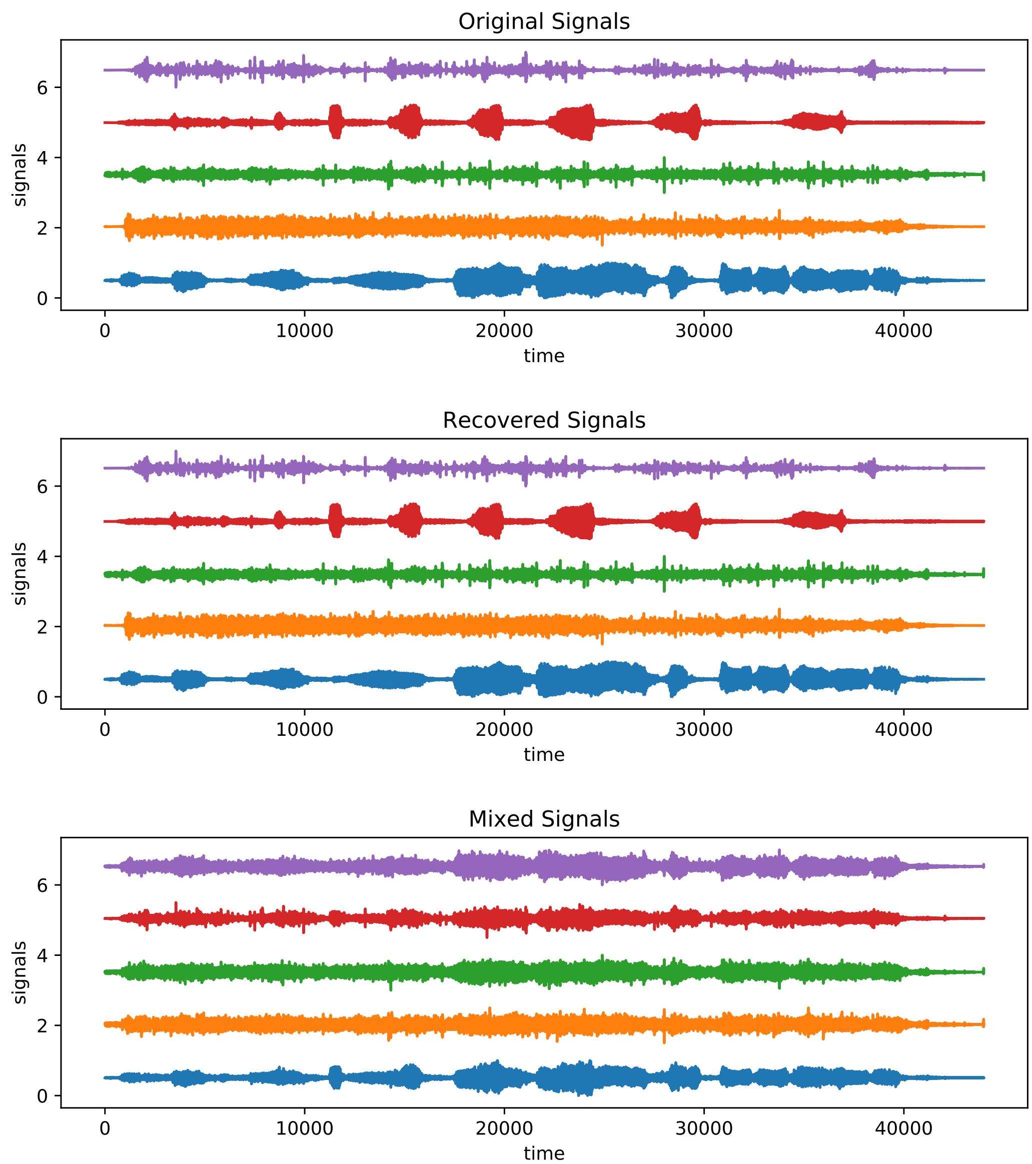

Figure 9: Scaled Signals. Up: Original; Middle: Recovered. Bottom: Mixed.

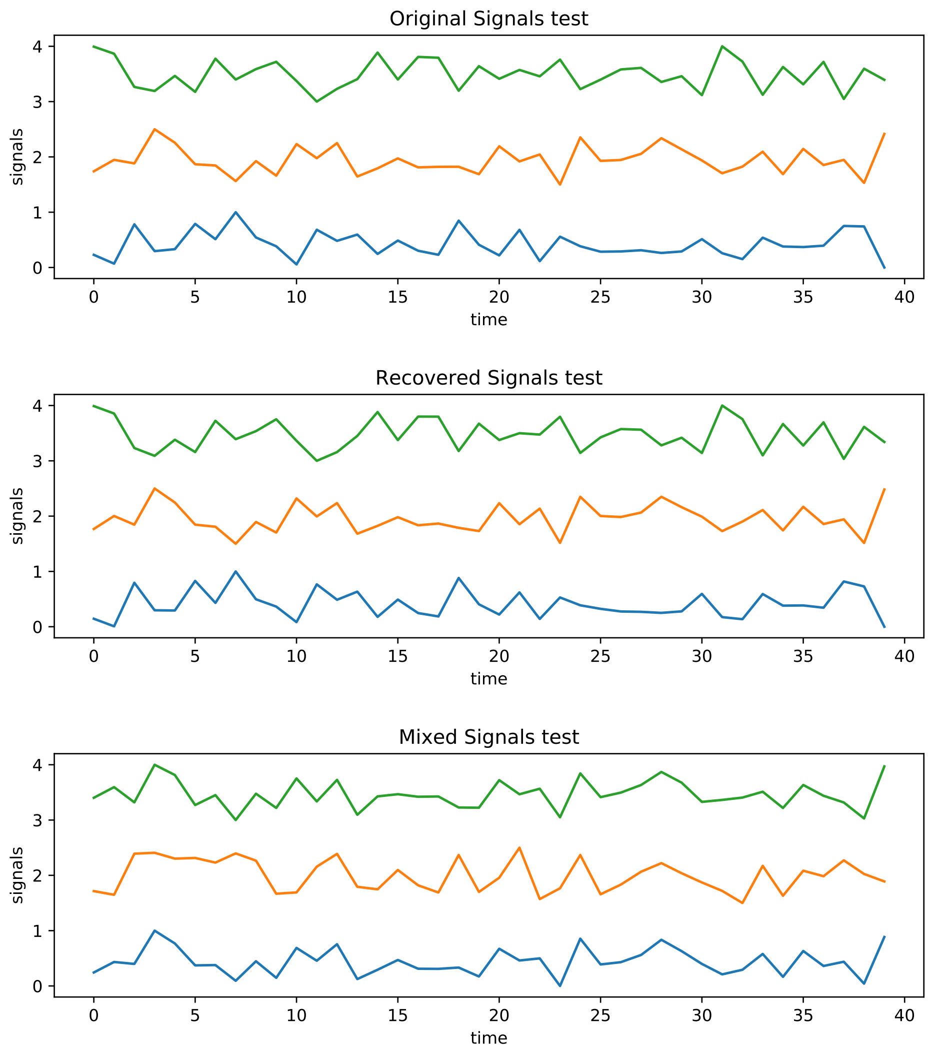

Figure 10: Scaled Test Signals. Up: Original; Middle: Recovered. Bottom: Mixed.

4 Summary

We use the individual correlation to evaluate the algorithm for blind source separation. The tables of correlations (after reordering) are in Figure 4 and 6, respectively. For the "sounds" dataset, the absolute value of correlations between recovered and original signals are all over 0.99, which means the algorithm recovers the source signals very well compared to the ground truth. For "icaTest" dataset, the correlations are around 0.99, but they are a little bit lower than the "sound" dataset. The reason may be that the signals in "icaTest" dataset are not strictly independent (See correlation matrix in Figure 5). As we know, ICA works well when the dataset are independent and non-Gaussian. When I used the algorithm without d, which does not work well, the recovery of signal 1 (orange) and 3 (red) are still good, the two may be easier to recover, while the signal 0 (blue) is difficult to recover. In summary, the experiment verifies the ICA algorithm works well in blind source separation problem, and we can investigate on the results to see the Independence and non-Gaussian assumptions are important,