HW2: Independent Component Analysis

Manu Agarwal
UTEID: ma53767

Abstract. In this paper, we use Independent Component Analysis for blind source separation. We use Gradient Descent approach for minimizing the mutual information between signals. We finally test our approach on several mixtures of three sound signals.

1 Introduction

Blind source separation is the process of separating a set of mixed signals into its original source components without any information about the nature of signals or the mixing process. As the number of source signals is greater than the number of mixed signals available, this problem is usually underdetermined in nature. The techniques for recovering the source signals generally try to narrow down on the number of possible solutions without excluding the desired solution from this space.
Independent Component Analysis (ICA) is one technique that is applied quite a lot in practice for solving the blind source separation problem. ICA assumes that the source signals are non-Gaussian signals and that they are statistically independent. Other possible approaches to solving the blind source separation problem include imposing structural constraints on the source signals.

2 Our approach - Independent Component Analysis

In this assignment, we use Independent Component Analysis to separate mixtures of signals into their source components. Suppose we have n linear mixtures $x_1, x_2, \ldots, x_n$ of n independent components.

$$x_j = a_{j1}s_1 + a_{j2}s_2 + \ldots + a_{jn}s_n$$

The above equation can be written in the matrix notation as

$$x = As$$

Our goal is to estimate A and then compute its inverse W so that

$$s = Wx$$

Since, we are treating source samples as independent, we can write

$$p(s) = \prod_{i=1}^{n} p_s(s_i)$$

Then, we can write the mixed signal probability density function as follows:

$$p(x) = \prod_{i=1}^{n} p_s w_i^T(x).|W|$$

As the probability density function is unknown to us, we use the cumulative distribution function $g(s) = \frac{1}{1+e^{-s}}$ so that pdf would be given by $g'(s)$. This helps us traverse the gradient of maximum information separation.
Finally, ΔW is given by $\eta(I + (1 - 2Z)Y^T)W$ where η is the learning rate and z_{ij} is given by $g(y_{ij})$.
Our algorithm is as follows:

Algorithm 1 Independent Component Analysis

Input: Signal matrix U and mixing matrix A
Output: Matrix Y of estimated source signals
1: $X = AU$
2: Initialize the matrix W with small random values
3: Calculate $Y = WX$
4: Calculate Z where $z_{i,j} = g(y_{i,j}) = \frac{1}{1+e^{-y_{i,j}}}$
5: Compute $\Delta W = \eta(I + (1 - 2Z)Y^T)W$
6: Update $W = W + \Delta W$ and repeat from step 3 until the algorithm converges or we have run enough iterations of the algorithm

3 Experiments

We first test ran our algorithm on three easy mixtures of three source signals given in icaTest.mat. We use the largest singular value of ΔW as a measure of convergence. The algorithm performed considerably well being able to recover the original signals with a good enough accuracy. One thing to note is that ICA does not preserve the order or scale of signals. Hence, the output signals may be a permutation of the original source signals and may be of different scales.
Figure 3 shows the recovered signals along with the source signals. Here we have scaled both the source and recovered signals to be in the range zero to one. It is quite easy to see that source signal 1 corresponds to the recovered signal 2, source signal 2 to recovered signal 3 and source signal 3 to recovered signal 1. Correlation between the souce and recovered signals can be a good measure to compute how well a signal was recoverd. We compute the correlation between each of the source and recovered signals(Table 1). For instance, the correlation between source signal 1 and recovered signal 2 is 0.9976 which indicates the two signals are quite similar.
Here we used 1000000 iterations of our algorithm and $\eta = 0.01$. Figure 1 shows the decline in the 2-norm (the largest singular value) of ΔW for the first 10,000 iterations of the algorithm. We see that after about 1000 iterations, ΔW almost

tends to zero. The norm of ΔW gives us a good measure of when to terminate the algorithm. Figure 2 shows the decline in the 2-norm of ΔW if we take the learning rate η to be 0.1. We observe that for higher values of η, ΔW tends to zero faster.

We then ran our algorithm on random mixtures of three sound signals. Each

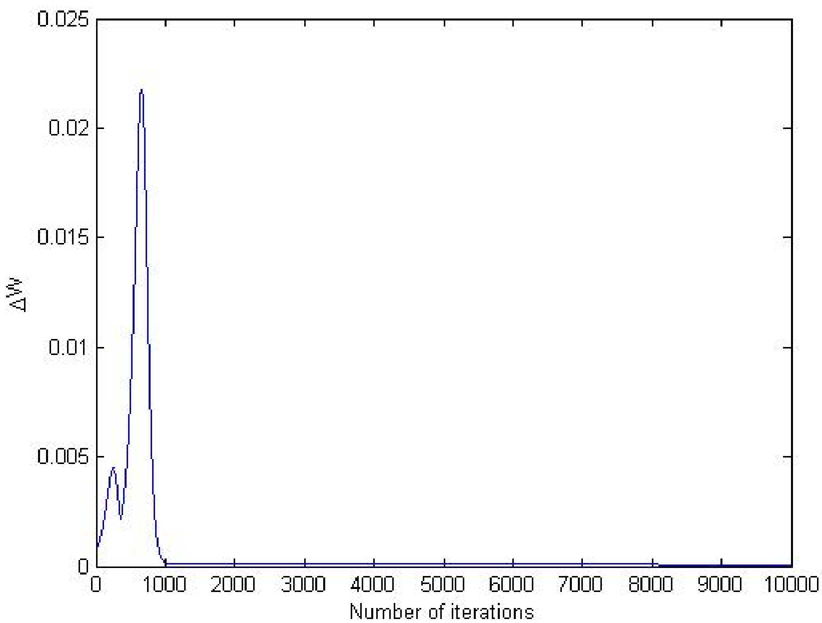

Fig. 1. Decline in ΔW as a function of the number of iterations of our algorithm ($\eta = 0.01$)

signal had 44000 time points. Since these signals were more complex than the previous test case, the recovered signals are not as good this time. However, the results are still reasonable and we are easily able to match source signals to the recovered ones. Figure 4 shows the recovered signals along with the source signals. As is evident from Figure 4, Source signal 1 corresponds to recovered signal 2, source signal 2 corresponds to recovered signal 3 and source signal 3 corresponds to recovered signal 1. The correlation matrix between the source and recovered signals also suggests this matching between the source signals and the recovered signals(Table 2). For instance, the correlation between source signal 1 and recovered signal 2 is the highest in the first row. This clearly indicates source signal 1 corresponds to the recovered signal 2. This time we used $\eta = 0.01$ and 100000 iterations of our algorithm.

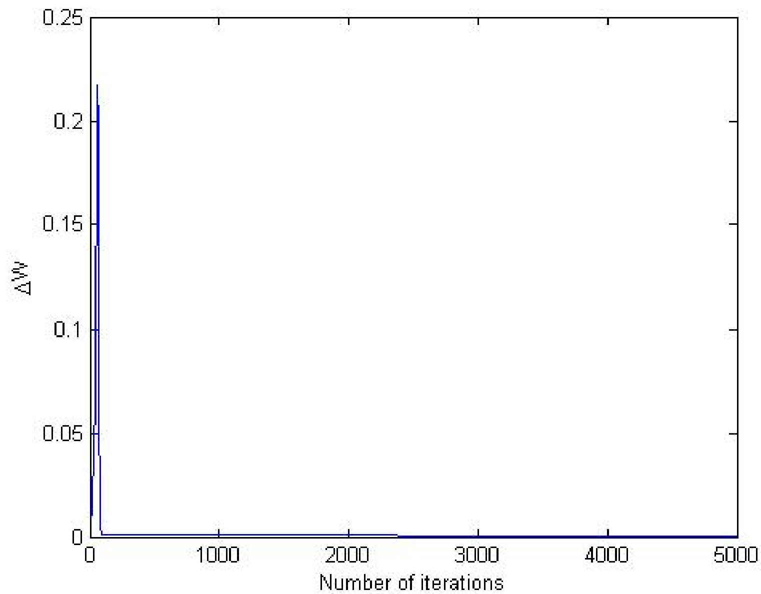

Fig. 2. Decline in ΔW as a function of the number of iterations of our algorithm ($\eta = 0.1$)

	Recovered Signal 1	Recovered Signal 2	Recovered Signal 3
Source Signal 1	0.6508	0.9976	0.6811
Source Signal 2	0.6573	0.6674	0.9981
Source Signal 3	0.9984	0.6637	0.6492

Table 1. Correlation between the source signals and the recovered signals(test data)

	Recovered Signal 1	Recovered Signal 2	Recovered Signal 3
Source Signal 1	0.9805	0.9918	0.9619
Source Signal 2	0.9862	0.9930	0.9983
Source Signal 3	0.9984	0.9844	0.9920

Table 2. Correlation between the source sound signals and the recovered signals(sound signals)

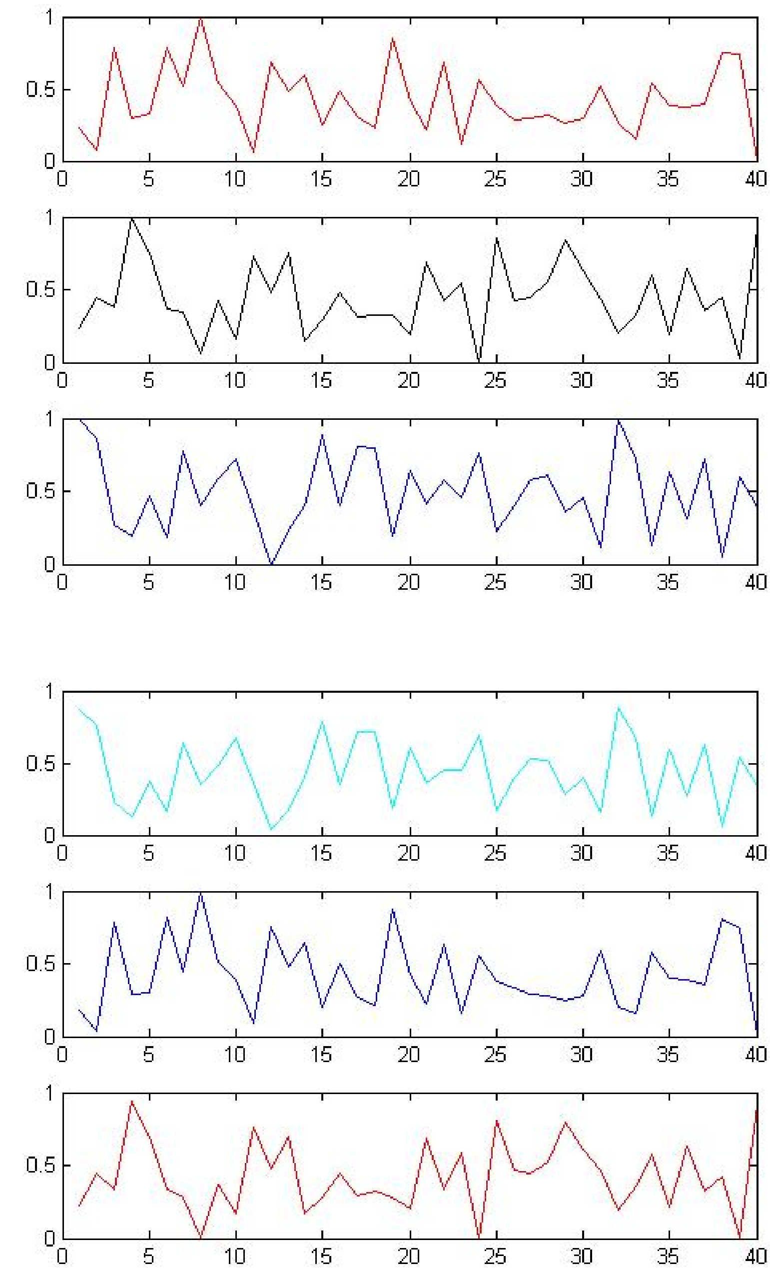

Fig. 3. The upper half shows the original source signals and the bottom half shows the recovered signals

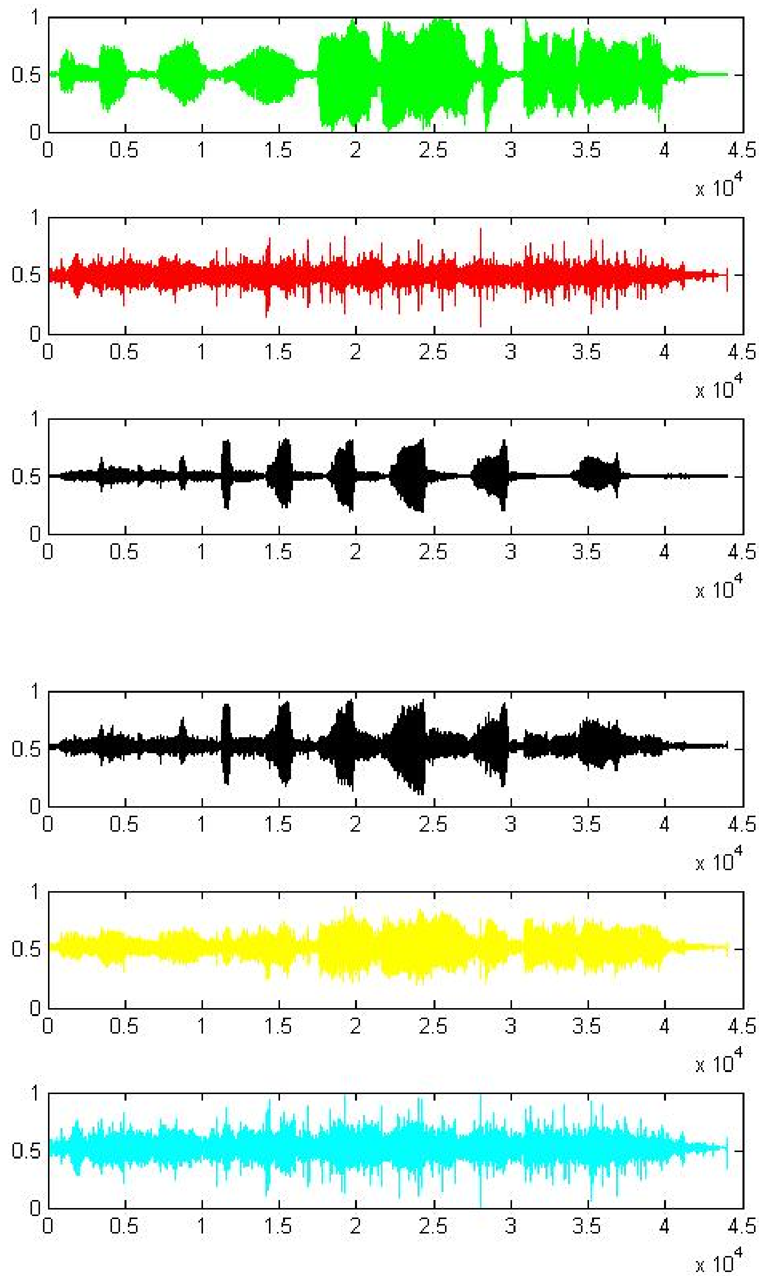

Fig. 4. The upper half shows the original sound signals and the bottom half shows the recovered signals

Conclusion

In this experiment, we observed how ICA can prove to be a useful algorithm for blind source separation. It serves as a useful method to skip finding the inverse of the mixing matrix. It makes a strong assumption of statistical independence between the source signals. Nonetheless, the algorithm performs quite well in practice even under this strong assumption which may not be true in general. However, we see that as the matrices become larger, the algorithm becomes really slow. In fact, that may be attributable to the use of gradient descent method in our algorithm.

References

1. Pattern Recognition and Machine Learning by Christopher Bishop
2. Resources on the Machine Learning course homepage
   http://www.cs.utexas.edu/ dana/MLClass/446outline.html