# Homework 2: Independent Component Analysis

Name: Shun Zhang
Email address: jensen.zhang@utexas.edu
EID: sz4554

## 1 Independent Component Analysis

In this report, I applied Independent Component Analysis on Blind Source Separation problem. Let $U$ be the original sources, which is a $n \times t$ matrix. $n$ is the number of sources and $t$ is the number of data. Let $A$ be mixing operation, and $X = AU$. The goal is to find a matrix $W$ such that $U = WX$, which recover the sources from the observations.

I used the batch update method described on the webpage (with the mistake fixed),

$$\Delta W = \eta(tI + (1 - 2Z)Y^T)W \tag{1}$$

This is slightly different from the update equation proved in Ng's note.

$$\Delta W = \eta((1 - 2Z)x^T + (W^T)^{-1}) \tag{2}$$

where $x$ stands for an observation at one time. It uses one sample to update I first convert it to the batch version. Data are added as columns, which make a matrix $X$. $W^T$ is simply added $t$ times. So,

$$\Delta W = \eta((1 - 2Z)X^T + t(W^T)^{-1}) \tag{3}$$

Equation 1 is more efficient, as there is no inverse of matrix. I tried to derive Equation 1 from Equation 3,

$$\begin{aligned}
\Delta W &= \eta((1 - 2Z)X^T + t(W^T)^{-1}) && (4)\\
&= \eta((1 - 2Z)X^T + t(W^T)^{-1})W^T(W^T)^{-1} && (5)\\
&= \eta((1 - 2Z)X^T W^T + tI)(W^T)^{-1} && (6)\\
&= \eta((1 - 2Z)(WX)^T + tI)(W^T)^{-1} && (7)\\
&= \eta((1 - 2Z)Y^T + tI)W(W^{-1})(W^T)^{-1} && (8)\\
&= \eta((1 - 2Z)Y^T + tI)W(W^T W)^{-1} && (9)
\end{aligned}$$

So the different is the term, $W^T W$. If $W$ is orthogonal, then it's $W^T W = I$. We know this is not a valid assumption. However, I still used Equation 1 in the following experiments.

## 2 Experiment

### 2.1 Experiments on Small Set

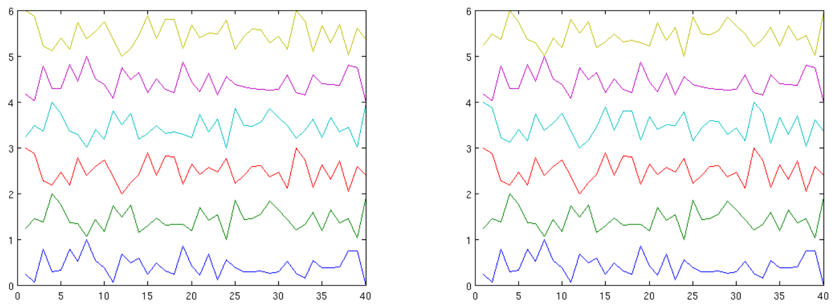


Figure 1: The bottom 3 lines are original signals from `icaTest.mat`. The top 3 lines are reconstructed signals with $\eta = 0.01$ and 1000 iterations. Two independent experiments are shown.

| | Recon. 1 | Recon. 2 | Recon. 3 |
|---|---|---|---|
| Source 1 | -0.4900 | **0.9905** | -0.4248 |
| Source 2 | **0.9918** | -0.3957 | -0.5454 |
| Source 3 | -0.4829 | -0.5073 | **0.9924** |

Table 1: Linear correlation between source and reconstructed signals, for the left figure in Figure 1. The maximum correlation is highlighted.

| | Recon. 1 | Recon. 2 | Recon. 3 |
|---|---|---|---|
| Source 1 | -0.4248 | **0.9905** | -0.4900 |
| Source 2 | -0.5454 | -0.3957 | **0.9918** |
| Source 3 | **0.9924** | -0.5073 | -0.4829 |

Table 2: Linear correlation between source and reconstructed signals, for the right figure in Figure 1. The maximum correlation is highlighted.

First I did experiment on `icaTest.mat`. $\eta = 0.01$ and 1000 iterations. The results are in shown in Figure 1.

The results are scaled into $[0, 1]$ interval. This experiment is repeated twice. Actually, the result can be permutation of the original sources. Scaling is also possible (as the results are scaled into $[0, 1]$, the signal could be flipped in this case, though I didn't observe this in the experiment). The correlation between source and reconstructed signals are shown in Table 1 and 2. By definition, they are highly correlated if the correlations is close to 1 or $-1$. We can tell that, for the left figure, the mapping is 0-4, 1-5, 2-3. While in the right one, the mapping is 0-5, 1-3, 2-4. This result can also be verified visually. We can also observe that the algorithm overall generate highly close signal, with has more than 0.99 correlations with its source signal.

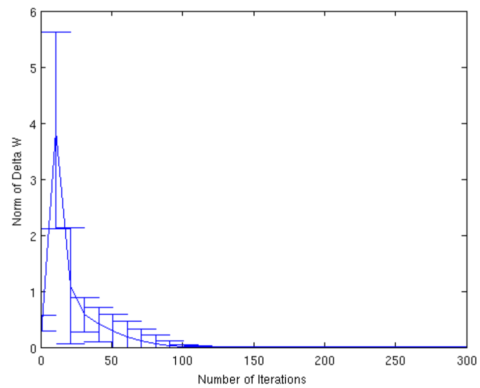


Figure 2: $||\Delta W||$ over number of iterations. Average of 100 runs. The length of vertical bars is $\sigma$ assuming Gaussian distribution of data points at each iteration.

I also examine how progress is made in each iteration in the learning process. As the algorithm uses gradient descent, the update on $W$ each step, which is $\Delta W$, is useful. For the convenience of visualization, I need the magnitude of this matrix. I use the largest singular value here, which can be got using `norm` function when applied to a matrix in Matlab.

The result is shown in Figure 2. Not surprisingly, steps become smaller after more iterations. The algorithm converges after $2 \times 10^5$ iterations.

## 2.2 Experiments on Sound

| | Recon. 1 | Recon. 2 | Recon. 3 | Recon. 4 | Recon. 5 |
|---|---|---|---|---|---|
| Source 1 | -0.0032 | 0.0030 | -0.0076 | -0.0035 | **1.0000** |
| Source 2 | -0.0014 | -0.0003 | **0.9999** | -0.0126 | 0.0016 |
| Source 3 | -0.0026 | -0.0095 | 0.0126 | **0.9999** | 0.0007 |
| Source 4 | 0.0020 | **0.9999** | 0.0083 | -0.0102 | 0.0041 |
| Source 5 | **0.9999** | 0.0051 | 0.0020 | 0.0125 | -0.0014 |

Table 3: Linear correlation between source and reconstructed signals.

The data are from `sounds.mat`. Matrix $A$ is generated randomly, in the range of $[0, 1]$ for each element. I set $\eta = 10^{-6}$ as $t$ is large in one step. It runs 200 iterations. Result on sound are shown in Figure 3. The correlation result is shown in Table 3. The scaling is same as the previous experiment.

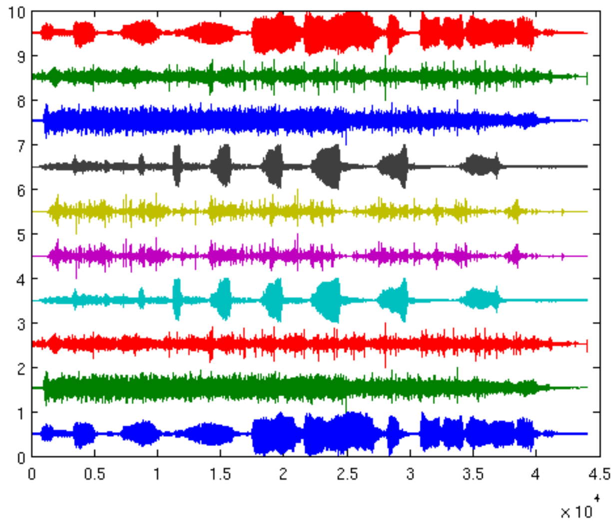


Figure 3: The bottom 3 lines are original signals from `sounds.mat`. The top 3 lines are reconstructed signals with $\eta = 10^{-6}$ and 200 iterations.

# 3 Discussion and Conclusion

I used a batch version of ICA for processing sound information. The number of samples is 44000. It's feasible process these samples as a whole. Actually, when the number of samples is even larger, or when the learning is online, it would be better to use sample-wise version.

Equation 1 is not an accurate one. Experiments have shown that it's useful. It mainly avoids inverse of an matrix.

I also set a static number of iterations for both experiments. I can also set a threshold on $||\Delta W||$ to let it stop.