CS 391L Machine Learning **Nidhi Kadkol**
Assignment 2 **UT EID: nk9368**
February 2019 **Email id: nidhik@cs.utexas.edu**

1 Introduction

This project uses Independent Component Analysis (ICA) for accomplishing blind source separation. We have a finite number of source signals which get mixed by an unknown process resulting in a number of mixed signals. Recovering the original signals from these mixed signals without having any information about the mixing process or the source signals is blind source separation. The number of mixed signals must be greater than or equal to the number of source signals to be able to use ICA.

We deal with audio signals in our project. The problem is analogous to a cocktail party where the sources are people speaking, and there are different microphones that pick up a mixture of sounds from different people. From the microphone signal, we want to separate out the mixture into the individual sound signals from each person.

2 Method

Given n sources and m microphones such that $m \geq n$. Let s be an n-dimensional vector which represents the value of each of the n sources at a given time. Then $x = As$, where A is an m x n mixing matrix. x is the mixed signal from n microphones at the same given time. Our goal is to find the unmixing matrix $W = A^{-1}$, so that we can recover s by calculating Wx.

Using the notation in Andrew Ng's notes, let w_i^T denote the i^{th} row of W. Suppose the distribution of each source s_i is given by p_s, then we assume that the sources are independent and the joint distribution of the source signals is

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

This gives the distribution of x as

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

We assume that the cumulative distribution function (cdf) of s is the sigmoid function, i.e.

$$g(s) = \frac{1}{1 + e^{-s}}$$

Then the pdf is

$$g'(s) = g(s)(1 - g(s))$$

Substituting $p_s(w_i^T x) = g'(w_i^T x)$ in (1) and taking the log, we get

$$\sum_{i=1}^{n} \log(g'(w_i^T x)) + \log|W|$$

We want to find the value of W which maximizes this function. So taking the derivative, we can get a stochastic gradient update rule. For a training example x at a particular time, the update rule is

$$W = W + \alpha \left(\begin{bmatrix} 1 - 2g(w_1^T x) \\ 1 - 2g(w_2^T x) \\ \vdots \\ 1 - 2g(w_n^T x) \end{bmatrix} x^T + (W^T)^{-1} \right) \tag{2}$$

This is for a single training example at a single timestamp. Now if we want to update the weight matrix using batch gradient descent, that is look at all the training examples before doing a gradient descent step, then our training example becomes a matrix X of all our training examples. Let U be the matrix of all our sources. In addition, each source is no longer a value for a single timestep, but is a t-length vector of values from times 1 to t. Similarly, row i of X tells gives us the signal captured by microphone i from time 1 to t. X is an m x t matrix and U is an n x t matrix. Thus, $X = AU$, and $Y = WX$ is our estimate of the source signals. If we calculate a matrix Z where $Z_{i,j} = g(y_{i,j})$ then equation (2) gets rewritten as

$$W = W + \alpha((W^T)^{-1} + (1/t) \cdot (1 - 2Z)X^T) \tag{3}$$

We divide the second term in the coefficient of α by the number of timesteps t as we have to take the average of all the t timesteps of the mixed signals in X, since we are taking all the timesteps together at once.

Since W is a non-square matrix, its inverse is not defined. To get around this, we multiply the equation with $W^T W$, and using the fact that $Y^T = X^T W^T$, we end up with the following update equation:

$$\mathbf{W} = \mathbf{W} + \alpha(\mathbf{I} + (\mathbf{1/t}) \cdot (\mathbf{1 - 2Z})\mathbf{Y^T})\mathbf{W} \tag{4}$$

We stop updating our matrix W when the change in Y after a gradient update is less than 1e-4. We say that we have achieved convergence at this point. If W does not converge within 8000 iterations, we break out of the gradient descent loop.

The recovered signals may be scaled by a factor, and the order of the recovered signals may be in a different order from that of the source signals. We scale each of the recovered signals to be between -1 and 1 and use the absolute value of Pearson's correlation coefficient to find how closely related a particular recovered signal is to a source signal. We calculate this value for every pair of source signal and recovered signal and find which source signal a recovered signal corresponds to by finding which source signal it has the highest absolute value of the correlation coefficient with.

3 Results

3.1 Original and Recovered Signals

3.1.1 Experiment 1

In the first experiment, we took the first 3 source signals and used 3 microphones to get 3 mixed signals. We initialized W with non-negative random numbers less than 0.001 and used a learning rate of 0.01. Each source signal is a sound clip approximately 4 seconds long. 5058 iterations were required for convergence. After convergence, we scaled the recovered signals to between -1 and 1. We also scale the mixed signals to between -1 and 1 before listening to and visualizing them.

We calculate the Pearson correlation coefficient matrix for the recovered signal matrix Y and the source matrix U as follows:

$$\begin{bmatrix} -0.0076 & 1.0000 & 0.0071 \\ -0.0034 & -0.0074 & 1.0000 \\ 1.0000 & 0.0015 & 0.0006 \end{bmatrix}$$

Entry (i, j) corresponds to the correlation of the i^{th} recovered signal with the j^{th} source signal. We look at the indices of the highest absolute value in each column, which tells us how to permute our recovered signals so they correspond to the source signals in the right order. We see that the highest values in each column are all very close to 1 while the rest of the values are close to 0, telling us that we have achieved a good reconstruction. We now plot the graphs of the mixed signals, the source signals, and the correctly ordered recovered signals to visualize our results.

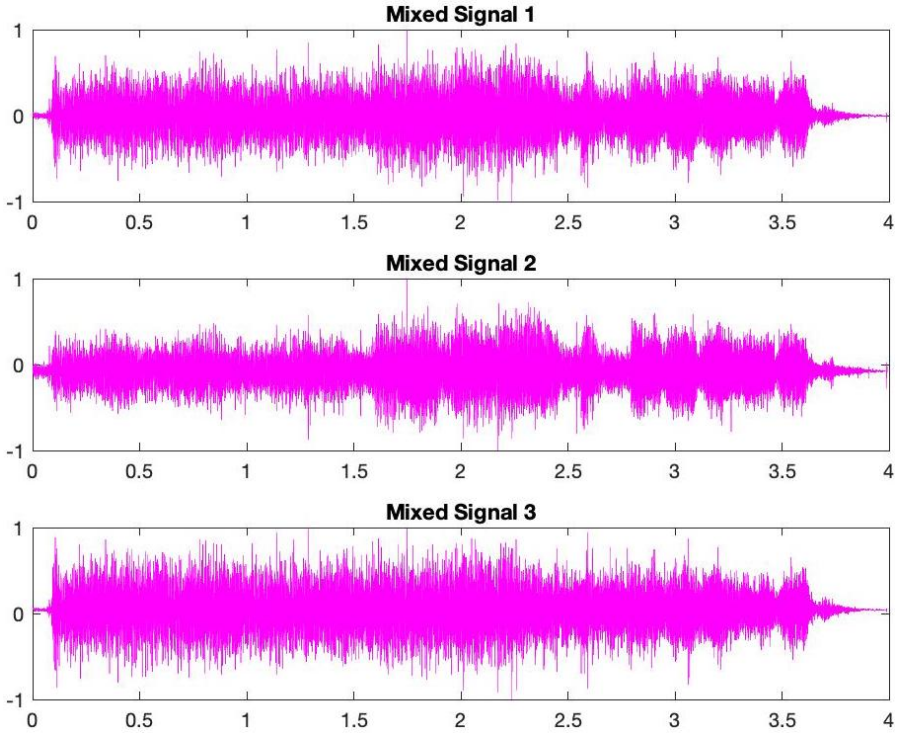

Figure 1: Plot of the 3 mixed Signals when we mix 3 source signals. The y-axis is amplitude and the x-axis is time in seconds.

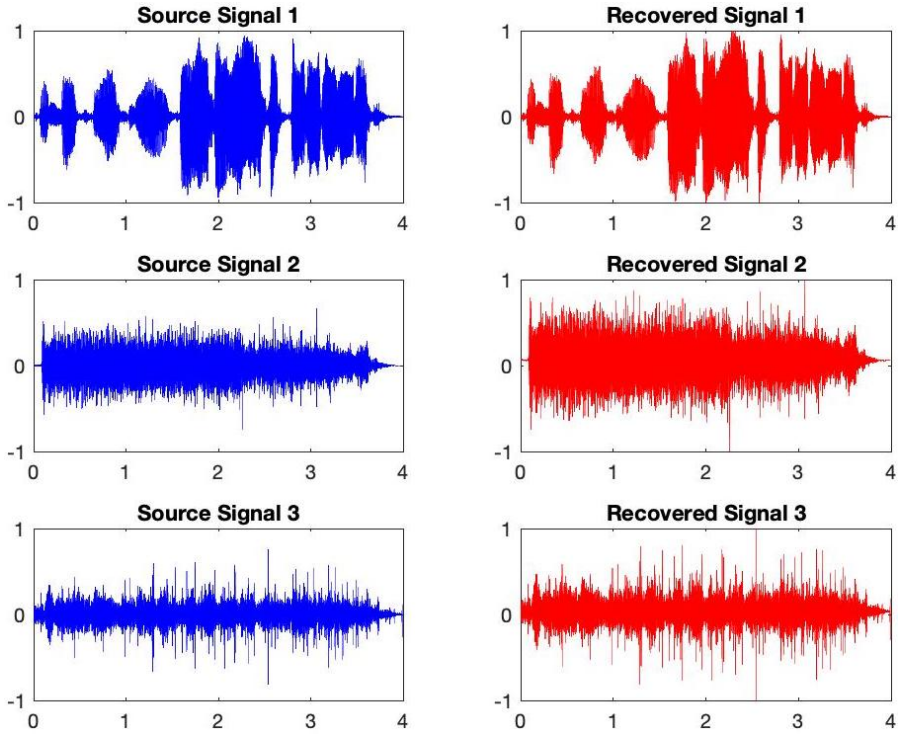

Figure 2: Plot of the source signals and reconstructed signals when we mix 3 source signals with 3 mics. The y-axis is amplitude and the x-axis is time in seconds.

From figures 1 and 2, we see that the signals are recovered almost perfectly, and may be scaled from the original ones, for example in recovered signal 2.

3.1.2 Experiment 2

In the second experiment, we took 4 source signals and used 8 microphones to get 8 mixed signals.We initialized W with non-negative random numbers less than 0.001 and used a learning rate of 0.01. Each source signal is a sound clip approximately 4 seconds long. 4890 iterations were required for convergence. After convergence, we scaled the recovered signals to between -1 and 1. We also scale the mixed signals to between -1 and 1 before listening to and visualizing them.

We calculate the Pearson correlation coefficient matrix for the recovered signal matrix Y and the source matrix U as follows:

$$\begin{bmatrix} -0.0064 & 1.0000 & 0.0070 & 0.0083 \\ -0.0027 & -0.0073 & 0.9999 & -0.0102 \\ 1.0000 & 0.0005 & -0.0000 & 0.0041 \\ 0.0030 & -0.0003 & -0.0095 & 0.9999 \end{bmatrix}$$

Entry (i, j) corresponds to the correlation of the i^{th} recovered signal with the j^{th} source signal. We look at the indices of the highest absolute value in each column, which tells us how to permute our recovered signals so they correspond to the source signals in the right order. We then plot the graphs of the mixed signals, the source signals, and the correctly ordered recovered signals to visualize our results. We see that the highest values in each column are all very close to 1 while the rest of the values are close to 0, telling us

that we have achieved a good reconstruction.

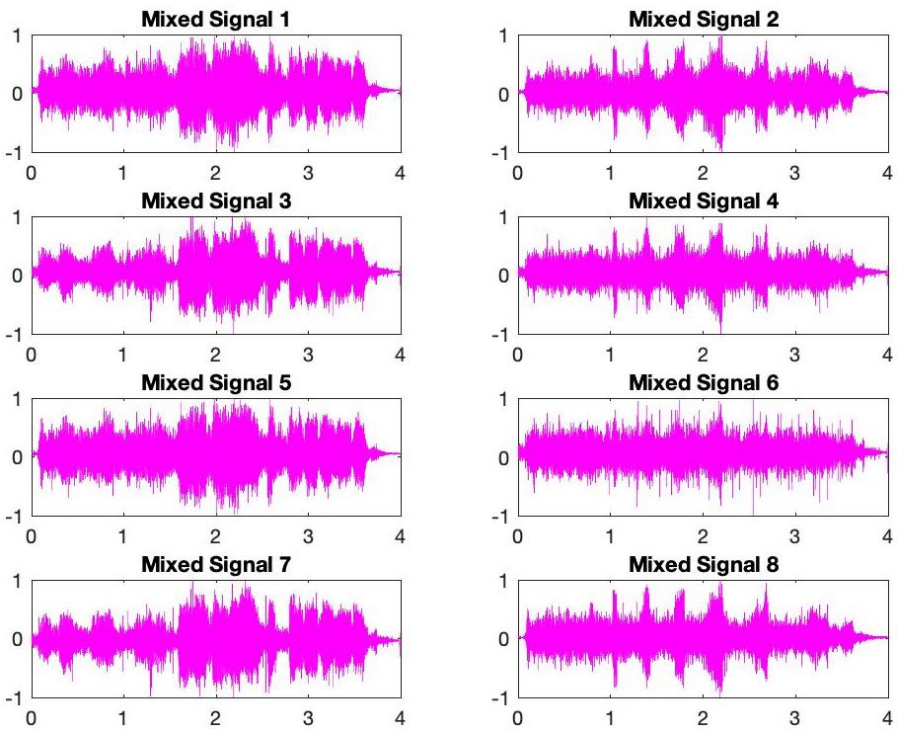

Figure 3: Plot of the 8 mixed signals when we mix 4 source signals. The y-axis is amplitude and the x-axis is time in seconds.

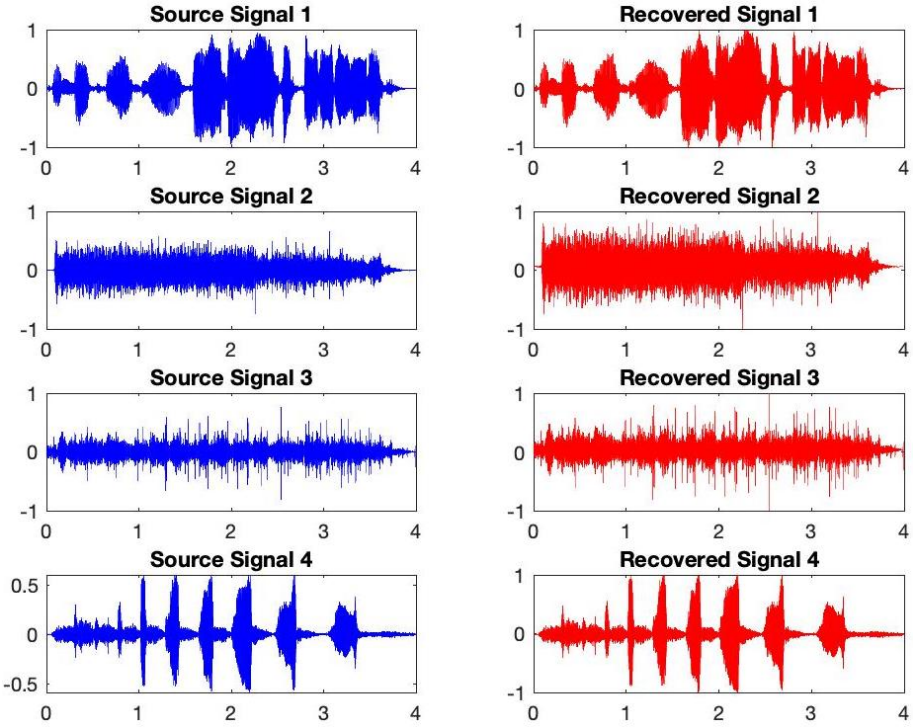

Figure 4: Plot of the source signals and reconstructed signals when we mix 4 source signals with 8 mics. The y-axis is amplitude and the x-axis is time in seconds.

From figure 3 and 4 we see that the original signals are mixed in different proportions to give us the 8 mixed signals. The second and third recovered signals are scaled slightly higher than the original one, but otherwise the signals are recovered with negligible difference.

3.1.3 Experiment 3

In this experiment, we took 5 source signals and used 8 microphones to get 8 mixed signals. We initialized W with non-negative random numbers less than 0.001 and used a learning rate of 0.01. Each source signal is a sound clip approximately 4 seconds long. 3189 iterations were required for convergence. After convergence, we scaled the recovered signals to between -1 and 1. We also scale the mixed signals to between -1 and 1 before listening to and visualizing them.

We calculate the Pearson correlation coefficient matrix for the recovered signal matrix Y and the source matrix U as follows:

$$\begin{bmatrix} -0.0070 & 0.6846 & 0.7297 & 0.0010 & 0.0079 \\ 0.0030 & -0.0001 & -0.0113 & 1.0000 & 0.0051 \\ -0.0027 & -0.7049 & 0.7084 & -0.0100 & 0.0111 \\ 1.0000 & -0.0017 & 0.0032 & 0.0041 & -0.0013 \\ -0.0033 & 0.0016 & -0.0030 & 0.0021 & 0.9999 \end{bmatrix}$$

Entry (i, j) corresponds to the correlation of the i^{th} recovered signal with the j^{th} source signal. We look at the indices of the highest absolute value in each column, which tells us how to permute our recovered

signals so they correspond to the source signals in the right order. We then plot the graphs of the mixed signals, the source signals, and the correctly ordered recovered signals to visualize our results. We see that the highest value in most of the columns is very close to 1 while the rest of the values are close to 0, telling us that we have achieved close to a prefect reconstruction.

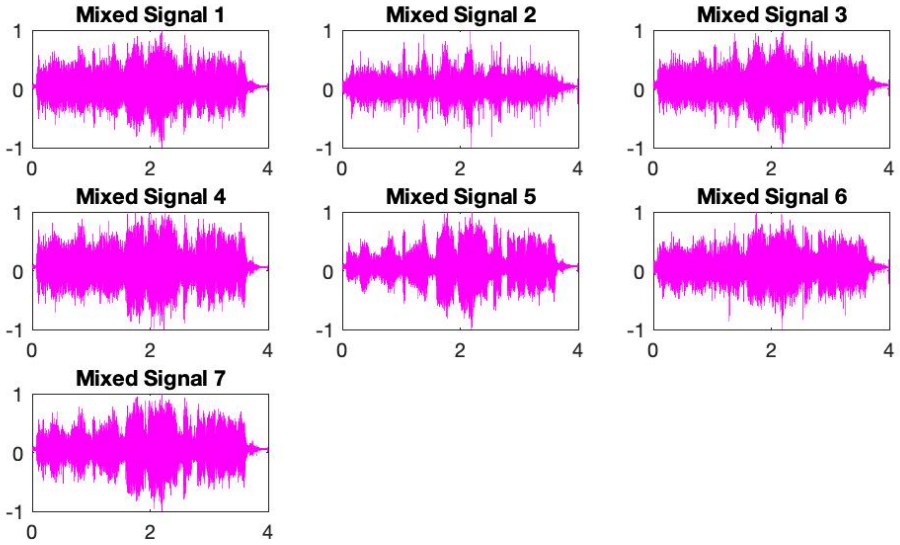

Figure 5: Plot of the 5 mixed signals when we mix 8 source signals. The y-axis is amplitude and the x-axis is time in seconds.

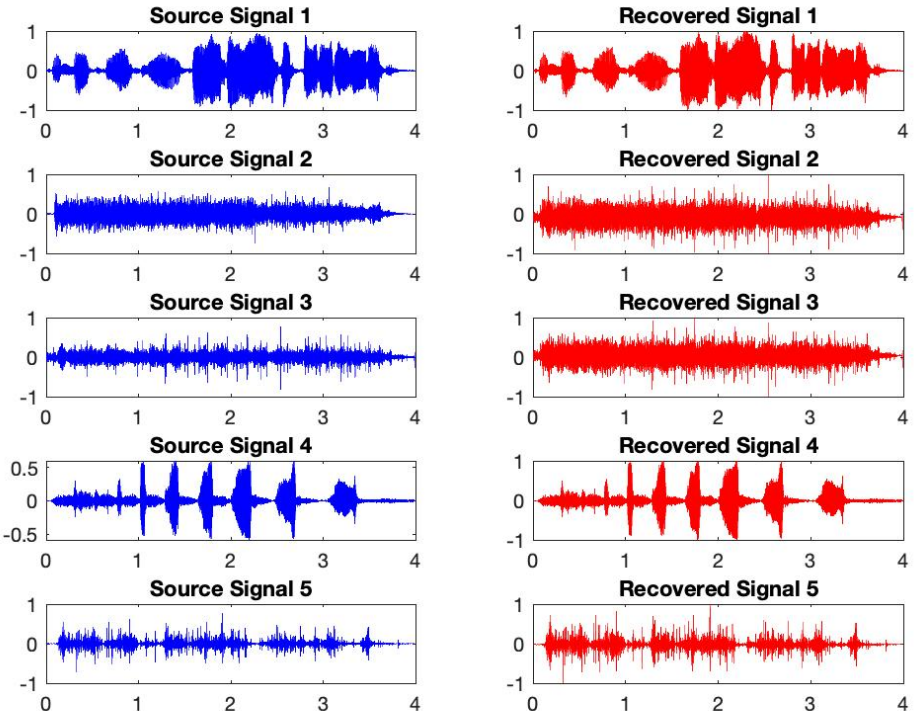

Figure 6: Plot of the source signals and reconstructed signals when we mix 5 source signals with 8 mics. The y-axis is amplitude and the x-axis is time in seconds.

From figure 5 and 6 we see that the original signals are mixed in different proportions to give us the 8 mixed signals. The second, third, and fifth recovered signals are scaled slightly higher than the original one. All the signals are recovered with negligible difference compared to the corresponding original signals.

3.2 Effect of Learning Rate on Convergence

We investigate the effect of learning rate on convergence in this experiment. We vary the learning rate in logarithmic step sizes from 0.001 to 1.0 and keep track of the number of iterations required for the model to converge. We repeat this in different settings of number of sources and number of microphones.

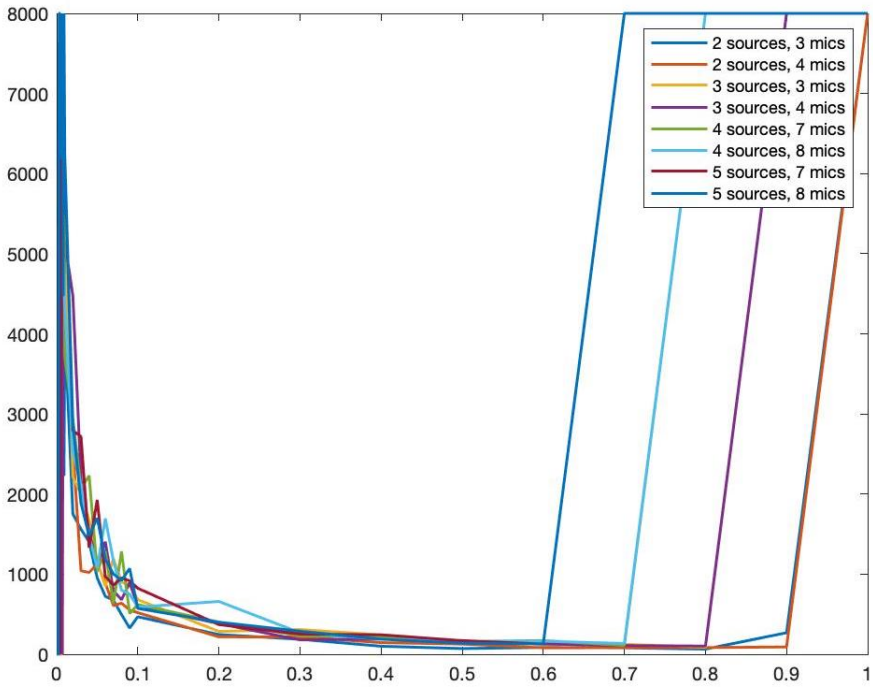

Figure 7: Graph of number of iterations needed to achieve convergence vs learning rate for different combinations of number of sources and number of mics.

We see in figure 7 that around learning rates of 0.5 we require the fewest number of iterations (around 300) for all the experimental settings. This decreases until learning rates from 0.6 to 0.9, after which the number of iterations required suddenly explodes. This is because if the learning rate is too high, the algorithm will never be able to take small enough step sizes to be able to reach the minima of the function we are trying to optimize. So it constantly moves back and forth around but not close to the minima, never being able to converge. We see that the general trend is the same for varying number of mics and sources for the lower learning rates. For higher rates, the number of iterations required for convergence explodes for all of them in the same manner but at different thresholds for learning rates.

3.3 Effect of Learning Rate on Correlation

To see the effect of varying learning rate on our correlation metric, we set up an experiment with 5 sources and 7 microphones. We varied the learning rates in logarithmic step sizes from 0.001 to 1.0 and performed

gradient descent steps for 350 iterations for each learning rate. After 350 iterations we kept track of the absolute correlation value for the recovered signal corresponding to source 1, the recovered signal corresponding to source 2, etc. until source 5 (to find the recovered signal corresponding to source i, we check which of the recovered signals has the highest absolute value of correlation coefficient with source i). We plot these absolute correlation values after 350 iterations for different values of the learning rate corresponding to each source as in figure 8.

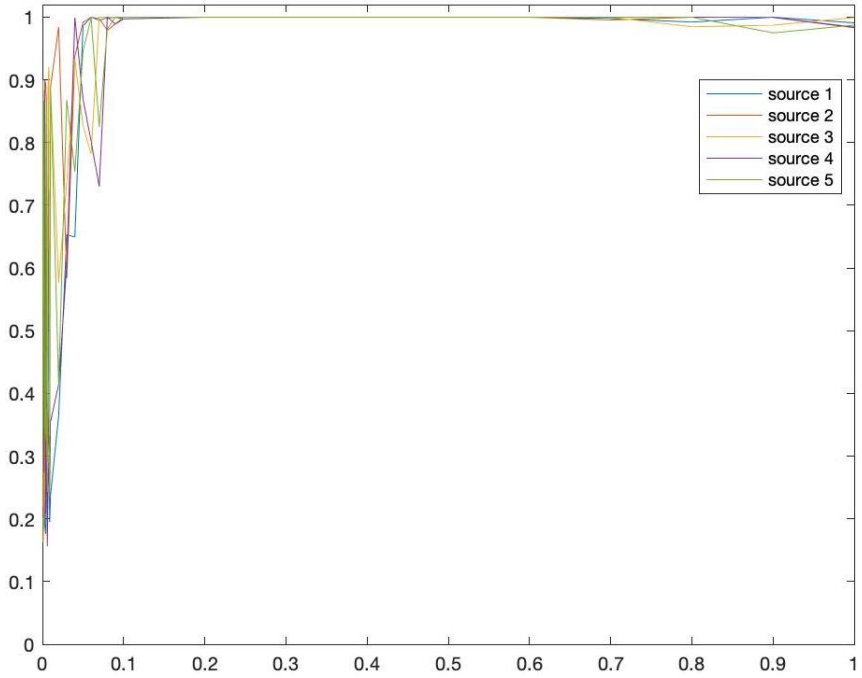

Figure 8: Graph of absolute value of Pearson's correlation coefficient vs learning rate after 350 iterations of gradient descent with 5 sources and 7 microphones.

We see that by learning rates from 0.2 onwards, all the correlation coefficients stabilize to 1.0 at the end of 350 iterations. Learning rates lower than that do not reach convergence before 350 iterations, and so we see a lot of fluctuation for smaller learning rates. After a learning rate of 0.9, we see the correlations start to decrease, as at that point the learning rate is too high to achieve convergence.

4 Summary

In this project we used the technique of Independent Component Analysis to separate out original signals from a mixture of signals with no information about the nature of the source signals or the technique of mixing. Additionally, we performed experiments varying the learning rate and seeing its effect on convergence and the correlation coefficient, and saw that if the learning rate is too high, we will not be able to achieve convergence or get a high correlation value. Our recovered signals had a correlation coefficient of close to 1 for more than 80% of our signals, thus we were able to recover the original signals almost perfectly. We also visually compared the original and recovered signals as well as plotted the mixed signals and saw how the recovery process can scale the original signals. We also listened to the signals as audio files and noted how the reconstruction was almost perfect.