HW-2: Independent Component Analysis

Subhashini Venugopalan (EID: SV8754)
vsubhashini@utexas.edu

Abstract

This report presents the experiments and results of implementing independent component analysis for accomplishing blind source separation.

1 Introduction

This experiment focuses on the task of separating a mixture of independent (non-gaussian) sound signals using the technique of independent component analysis (ICA). The idea of blind source separation (BSS) is to determine the original signals given the linear mixtures. In ICA, one assumes that the mixtures and the sources are random variables (with zero mean) instead of proper time signals. This allows the final mixed signals to be considered as samples of these random variables. To denote these mathematically, consider $\mathbf{s}$ to be the matrix of source signals (where each row consists of the samples of the source); let the matrix $\mathbf{A}$ be the mixing matrix, which produces the mixed signals $\mathbf{x}$ by applying a linear transformation on the sources. We have that:

$$\mathbf{x} = \mathbf{As} \tag{1}$$

Now, the objective of BSS is to recover the source signals $\mathbf{s}$. given only the matrix $\mathbf{x}$. More explicitly, using ICA, the objective is to find a matrix $\mathbf{W}$ such that:

$$\mathbf{s} = \mathbf{Wx} \tag{2}$$

Note that although the goal is to recover the source signals as accurately as possible, it is not necessary for the sources (rows) to be ordered in the same manner as in the original matrix $\mathbf{s}$ in Equation(2). In other words, the first row of the recovered matrix $\mathbf{s}'$ may not be the same as that of the signal in the first row of $\mathbf{s}$, but it has to be very close to one of the rows of $\mathbf{s}$.

Non-Gaussian sources The primary assumption (and the only necessity for ICA) is that the source signals are non-gaussian with zero mean. Even if the signals are not zero mean, they can be standardized by subtracting the sample mean. However, non-gaussianity is a necessary condition. To understand why, consider the joint distribution of two gaussian variables. This distribution would be spread like a circle. It can be proved that any orthogonal transformation of the joint gaussian will still have the exact same distribution as the orginal joint distribution (i.e. we still get back a similar circle after the transformation) giving us no useful information on the directions of the mixing matrix.

2 Maximum Likelihood estimation and ICA algorithm

Likelihood. Estimation of maximum likelihood is a popular approach for estimating the ICA model. The problem is formulated as the likelihood in the ICA model and then the model is

estimated by a maximum likelihood method. Continuing with the already mentioned notation, expand $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_n)^T$. The log-likelihood takes the form:

$$L = \sum_{i=1}^{t} \sum_{j=1}^{n} log\left(g_i(\mathbf{W}_j^T x^{(i)})\right) + log(|det\mathbf{W}|) \tag{3}$$

where g_i is the density function of the source s_i. This form for the likelihood comes from the fact that, for any random vector $\mathbf{x}$ with density p_x, and for any matrix $\mathbf{W}$, the density $\mathbf{y} = \mathbf{W}\mathbf{x}$, is given by $p_x(\mathbf{W}\mathbf{x})|det\ \mathbf{W}|$. In this assignment, we define the density function as $g_i(x) = \frac{1}{(1+e^{-\mathbf{x}})}$.

Using gradient descent to compute the gradient $\partial L/\partial \mathbf{W}$, we get the following algorithm for BSS:

Algorithm 1 BSS with ICA

1: Let the training matrix $\mathbf{X} = \mathbf{A}\mathbf{U}$.
2: Initialize matrix $\mathbf{W}$ with small values in range $(0, 0.1)$. $\mathbf{W} \leftarrow rand(n, m)./10$
3: Initialize learning rate η.
4: Initialize annealing rate T. $T \leftarrow 1000$
5: Set number of iterations I $\leftarrow 10000$
6: **for** i $= 1$:I **do**
7:     Estimate: $\mathbf{Y} = \mathbf{W}\mathbf{X}$
8:     Compute $\mathbf{Z}$, where $\mathbf{Z} = g(\mathbf{Y}) = \frac{1}{(1+e^{-\mathbf{Y}})}$
9:     $\Delta\mathbf{W} \leftarrow \eta * (\mathbf{I} + (1 - 2\mathbf{Z})\mathbf{Y}^T) * \mathbf{W}$
10:     Update: $\mathbf{W} \leftarrow \mathbf{W} + \Delta\mathbf{W}$
11:     Anneal: $\eta = \eta/(1 + i/T)$
12: **end for**

2.1 Recovered and source signal comparison

The output signals and source signals were compared by considering the **pairwise correlation co-efficients** for each pair of signals. The output signal with the highest correlation with a specific source signal was matched as the recovered signal (corresponding to that source).

2.2 Sigmoid function, Computing gradient

The sigmoid function is used to describe the density function of the sources. This is obtained from the following code:

Listing 1: Sigmoid function

```
function g= sigmoid(y)
% The sigmoid/logistic function
% g(y) = 1/(1 + e^{-y})
% Returns the sigmoid of y, where y can be a single value, vector or a matrix

g = zeros(size(y));
g = 1 ./ (1 + exp(-y));

end
```

The gradient is computed using the following matlab code:

Listing 2: Gradient

```matlab
function [grad] = gradient(eta, Y, W)
%Gradient function, it gives the value by which W must be updated to get new W
% \del W = eta (I + (1-2Z)Y')W where
% eta - learning rate
% Z = sigmoid(Y)
% W is the predicted matrix

delW = zeros(size(W));

Z = sigmoid(Y);
Id = eye(size(Y,1));
grad = eta * (Id + (1-2*Z)*Y') * W;

end
```

3 Experiments

3.1 Methodology

The basic developmental algorithm was first tested on a simple data set consisting of 3 sound sources with 40 samples for each source. The primary data for the main experiments consisted of 5 sound source samples, each approximately 4 seconds in length and sampled at the rate 11025 samples per second. The mixing matrix $\mathbf{A}$ was created by choosing random elements in the range $(0 - 0.1)$. The experiment was first conducted by considering all 3 source combinations (i.e. $\binom{5}{3}$ source combinations), to compare which source combinations were easy (equivalently hard) to separate. Correlation co-efficient was used to compare the goodness of recovery. The recovered signals were also plotted for visual comparison of the results.
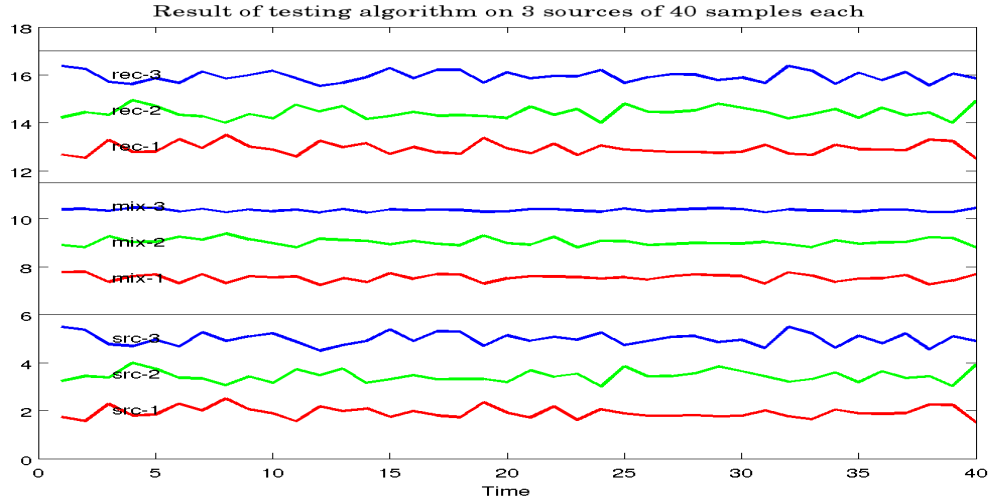

Figure 1: Result of testing the algorithm on small 40 sample signal sources.

3

Test Separation For the test case, we set $\eta = 0.01$ and number of iterations to I= 1000000. The mixing matrix contained random values (chosen uniformly at random) in the range 0 to 1. The recovered signals showed high correlation to original signals and can also be observed in Figure(1).

4 Results

The original sound sources in the signal plots are marked from 1 through 5 based on the ordering in the sounds data matrix. The mapping is as follows: 1 - correponds to speaking (laws of thermodynamics). 2 corresponds to vaccum cleaner. 3 to clapping, 4 to laghter, and 5 to static noise.

4.1 Recovering mixture of 3 sources with annealing

The figures below show the effect of recovering a combination of 3 sources (mixed using a random mixing matrix). This experiment was performed with annealing.
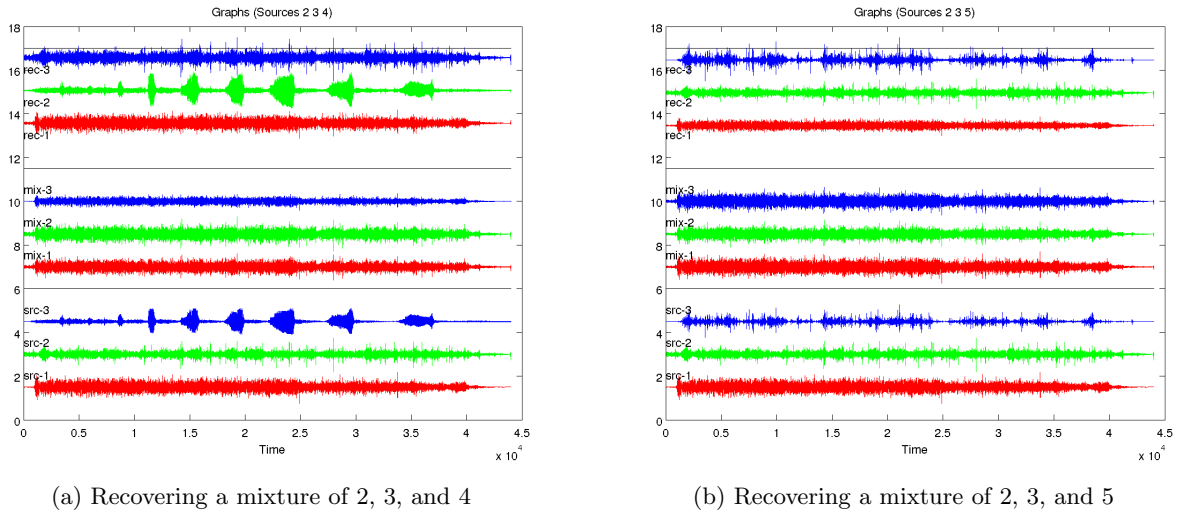

(a) Recovering a mixture of 2, 3, and 4

(b) Recovering a mixture of 2, 3, and 5

Figure 2: Mixture of sources 2,3, and 4; and that of 2,3, and 5, were recovered well.
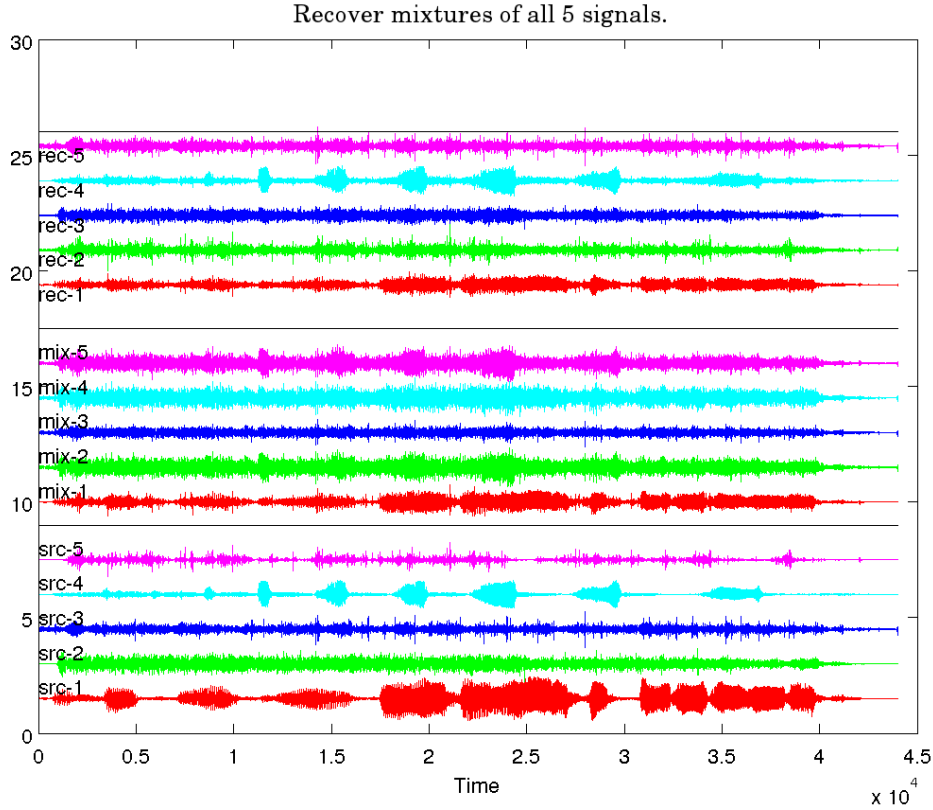
Observations The following are some observations based on the experiment of combining three source signals and using annealing.

1. Annealing was extremely useful in leading the signals to converge on a good optima (atleast prevented converging on a poor local optima). Without annealing, the gradient descent algorithm usually converged to a local optimum which was not useful in separating the sources. Annealing also reduced the number of iterations required to converge (to as few as 10000).

2. Some signal combinations were better recoverable than others. It was observed that the noise of the vaccum cleaner (sound source 2), hands clapping (source 3) and laughter (source 4),

4

were recovered fairly well. The sources corresponding to 2,3, and 5 (static noise) were also recovered reasonably well. Figure (4), shows the recovered signals.

4.2   Recovering all 5 signals

The figures below show the effect of recovering a linear mixtures of all 5 sources (mixed using a random mixing matrix). This experiment was performed with annealing.

(a) Recovering mixture of all 5 signals

Figure 3: Mixture of sources 2,3, and 4; and that of 2,3, and 5, were recovered well.

Observations   The following are some observations based during the experiment when recovering all 5 sources from the mixtures.

1. The learning rate and annealing parameters were left unchanged (compared to recovering from 3 sources). This shows that even the simeple annealing method is effective in controlling learning.

2. It was difficult to recover all the signals most of the times. Only in some of the runs all sources were recovered well. But in most of the trials, atleast a couple of sources were recovered with reasonable accuracy.

3. During the experiment it was noted that large number of iterations did not make that much of a difference.

4.3 Additional experiments: varying slope of sigmoid

In addition to the regular algorithm, one can change the slope of the sigmoid to see if that affects the performance of the algorithm.We consider the new density function to be $g_i(x) = \frac{1}{(1+e^{-\beta \mathbf{x}})}$ where β is unique for each source signal. We now get the following gradient for β:

$$\frac{\partial L}{\partial \beta_j} = \sum_{i=1}^{t} x_{ji}(1 - 2g(x_{ji}))(x_{ji})^T$$

The β values were then changed as follows:

$$\beta_{new} = \beta_{old} + \kappa \left(\frac{\partial L}{\partial \beta_j} \right)$$
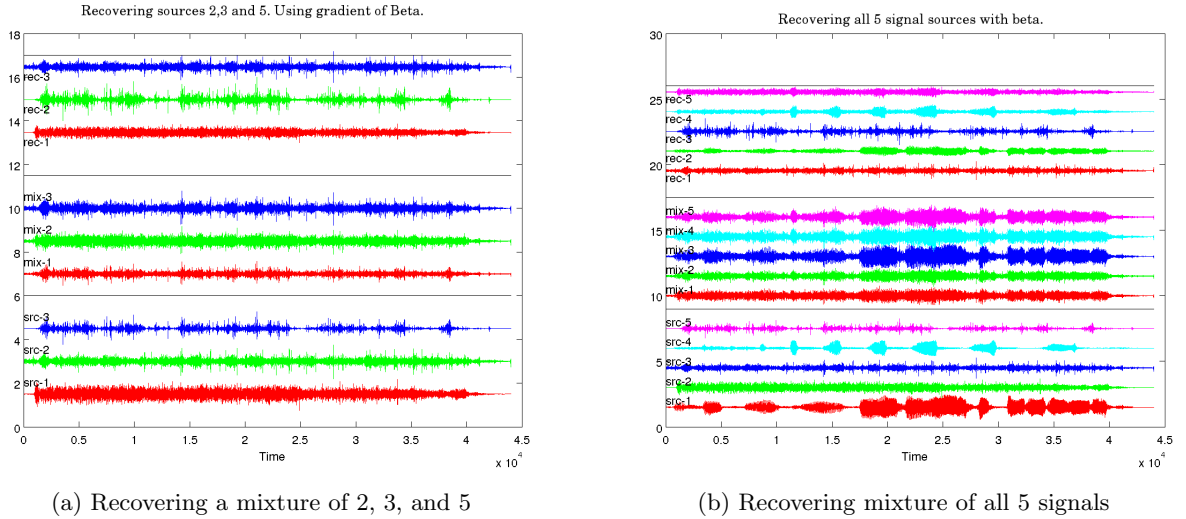

(a) Recovering a mixture of 2, 3, and 5

(b) Recovering mixture of all 5 signals

Figure 4: Recovering signals by additionally varying β.

Observations The following are some observations based during the experiment with additional variation of β.

1. Setting κ to very small values (0.0001, 0.001) did not make any change to β values (i.e.$\beta_{new} = \beta_{old}$). The sources were recovered just as before.

2. Setting $\kappa = 0.01$ appeared reasonable. However, the actual values of β_j (for each source were more or less very close). Here again, sources were recovered as before, but appeared slightly better (i.e running them on the same sources several times produced slightly better recovery than with out changing).

6

3. Setting κ to higher values did not improve the signal recovery, infact it made it only slightly worse.

4. On increasing the number of source signals (taking all 5) considering β did stabilize the recovery, if only marginally. i.e. On average (several trials) it appeared as though, the algorithm varying β may have performed better.[1]

5 Conclusion

This experiment proves that the ICA algorithm is surprisingly powerful and can recover a cocktail of mixed signals. However, there are several tiny caveats to note, such as, the signals need to be non-gaussian and independent, and the mixture has to be a linear mix. The additional experiment of varying β only seemed to improve the recovery marginally. Comparing recovered signals across different trial runs should conclusively determine if variation in β stabilizes the recovery.

6 Appendix

Comments.

1. The correlation matrices were obtained, but not presented in the report.

2. The recovered sound signals have been saved as .mat files and can be made available upon request.

Code To see a sample run of the code, do the following on a terminal:

1. `cd src`

2. `matlab icafull.m`

The README file explains the contents of other folders.

---

[1]I have not consolidated the correlation values of recovered signals across experiment trials to prove this conclusively.