Independent Component Analysis
Tyler McDonnell
tyler@cs.utexas.edu

Introduction

Independent Component Analysis (ICA) is a method for blind source separation, the separation of a set of source signals from a set of mixed signals. It is an unsupervised learning approach where we want to recover the original signals with no knowledge of them. To accomplish this, ICA assumes that each of the components is non-Gaussian and independent. Similar to Principal Component Analysis (PCA), ICA computes a new basis for the data, but unlike PCA, the goal of ICA is to use this computed basis to recover the source signals, subject to some scale factor. Here, I introduce the ICA algorithm and apply it to mixed audio signals. I explore some of the variables at play with applying ICA and their effects on signal separation.

Independent Component Analysis

Independent Component Analysis seeks to find a new basis for a set of mixed signals which can separate these signals into their sources, subject to some scale factor. Formally, we assume that there exists an n by t matrix $\mathbf{U}$ of n source signals, $\{s_1, \ldots, s_n\}$, where t is the dimensionality of each signal; an m by t matrix of m mixed signals, $\mathbf{X}$; and a mixing matrix $\mathbf{A}$, such that:

$$\mathbf{X} = \mathbf{A}\mathbf{U}$$

ICA thus seeks to find $\mathbf{W} = \mathbf{A}^{-1}$, an *unmixing matrix*, which can recover the original source signals from $\mathbf{X}$. In other words, it constructs $\mathbf{W}$ such that:

$$\mathbf{W}\mathbf{X} = \mathbf{U}$$

To accomplish this, ICA assumes that each of the source signals is non-Gaussian and independent. Thus, the probability distribution of $\mathbf{X}$ can be described as a product of the probability distributions of each of the input signals, $\{s_1, \ldots, s_n\}$. ICA models the density of each source signal using a sigmoid cumulative distribution function (CDF) of the form:

$$p_s(s_i) = \frac{1}{1 + e^{-t}}$$

ICA iteratively computes $\mathbf{W}$ to minimize mutual information between the signals using gradient descent. In general, it is not possible for the ICA algorithm to perfectly discern the number of original source signals, and such a distinction may not always make sense. Additionally, methods like PCA may be applied before ICA to reduce the dimensionality of the input and focus on recovering high-variance signals. For the toy examples that follow, we will assume that we know n, the number of source signals. The high-level ICA algorithm is shown below.

1. Assume $\mathbf{X} = \mathbf{A}\mathbf{U}$ where $\mathbf{U}$ is a matrix of n source signals and $\mathbf{A}$ is a mixing matrix.
2. Compute $\mathbf{Y} = \mathbf{W}\mathbf{X}$
3. Compute $\mathbf{Z} = \frac{1}{1+e^{\mathbf{Y}}}$.
4. Compute $\mathbf{W} := \mathbf{W} + \eta \left(\left[I + (1 - 2\mathbf{Z})\mathbf{Y}^T \right] \right)$

5. Repeat Steps 2 - 4 for i iterations or until convergence.
6. Recover the source signals by computing $\mathbf{U}_{recovered} = \mathbf{WX}$.

Experiments

Here I apply ICA to different sets of mixed signals and explore the key variables at play. First, I'll look at how ICA performs for mixtures of very different source signals. Then, I'll look at how the number of iterations used for gradient descent impacts performance. Finally, I'll consider the effects of the selected learning rate on the ICA algorithm.

For these experiments, I'll be mixing from a set of 3 source signals, each of which is a 5-second audio clip. A brief qualitative description of each is provided below.

- s_1: A famous Homer Simpson sound bite.
- s_2: A clip of a man laughing.
- s_3. Crackling noise.

Tuning The Variables: Input Signals

Though in practice, the different source signals constituting the mixed signal will not be a variable under our control, it's useful to understand how effectively ICA can separate different types of signals. Here, I construct two different sets of mixed signals with a common mixing matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} 2 & 5 \\ 3 & 7 \end{bmatrix} \qquad \mathbf{X}_1 = \mathbf{A} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \qquad \mathbf{X}_2 = \mathbf{A} \begin{bmatrix} s_1 \\ s_3 \end{bmatrix}$$

Figure 1 shows the results of ICA for these two sets of mixed signals.

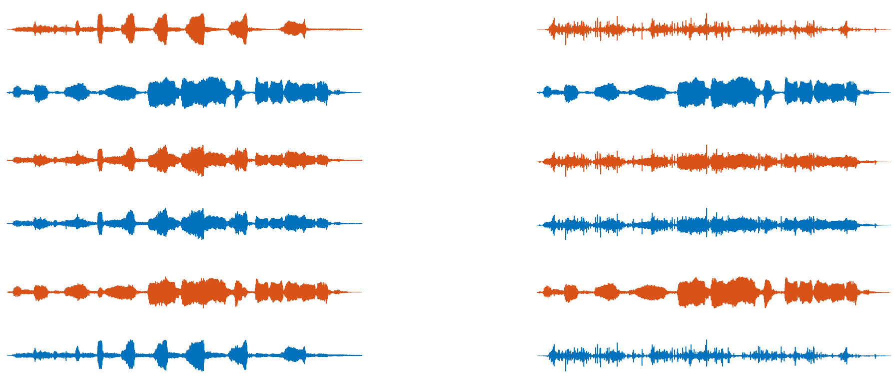

Figure 1: ICA Comparison.

Comparison of ICA for $\mathbf{X}_1$ (left) and $\mathbf{X}_2$ (right). The three pairs of signals in each column are: source, mixed, and recovered. All signals are normalized to [0,1]. I used 10,000 iterations for gradient descent with $\eta = 0.0005$.

As you can see, with these parameters we were able to recover the original signals spectacularly well. However, the results are not perfect. In particular, you can observe some noise from s_3 in the recovered S1 signal in the right-hand column. This noise is partially quantified by the the Mean Square Error (MSE) between the recovered and source signals, a common measure of signal fidelity. The MSE for the s_2 recovered from $\mathbf{X}_1$ was 0.0145, while the MSE for the same signal recovered from $\mathbf{X}_1$ was 0.0199.

Unfortunately, this noise is easier to observe qualitatively. The crackling noise of s_3 can be clearly heard in the s_2 signal recovered from $\mathbf{X}_2$. There are two hypotheses for this behavior. The first notes that the s_3 signal vaguely resembles Gaussian white noise, which violates one of our ICA assumptions. The second would be that the probability distributions of s_2 and s_3 are not independent.

Tuning The Variables: Iterations for Gradient Descent

The most obvious tune-able variable at play in this variation of ICA is the number of iterations used for gradient descent. In general, we want to perform enough iterations for the matrix $\mathbf{W}$ to converge to some near-optimal value. Figure 2 shows s_1 and the recovered s_1 signals for various values of i. Table shows the MSE for each of these recovered signals. Intuitively, the MSE decreases and the quality of the reproduction increases with more iterations of gradient descent. However, we also reach a point of convergence where further iterations do not improve the recovered signals. In this particular case, that convergence was around 100,000 iterations.

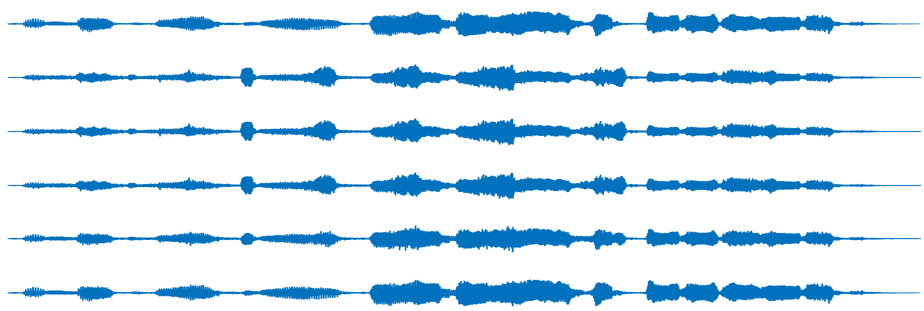

Figure 2: s_1 (top) and recovered s_1 for i = 10, 100, 1000, 10000, 100000.

Iterations	MSE
10	0.0067
100	0.0064
1000	0.0057
10000	0.0014
100000	0.0001

Table 1: MSE of recovered s_1 for various i.

Tuning The Variables: η

Finally, I take a brief look at the learning rate parameter in ICA, η. With Gradient Descent, the learning rate specifies how slowly or quickly we will move towards the optimal weights. In general, η can be either too large or too small: in some cases, it may skip the optimal solution entirely, and in others it may take too many iterations to converge to a near-optimal value.

Figure 3 shows s_1 and the recovered s_1 for two different values of η. We can see how the learning rate can affect the speed of convergence of ICA, as both of these learning rates eventually produced very similar recovered versions of s_1 after 100,000 iterations.

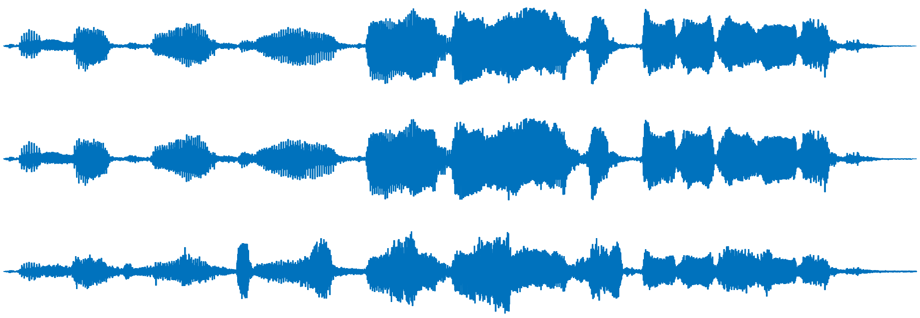

Figure 3: s_1 (top) and recovered s_1 for $\eta = 0.001, 0.01$ and $i = 5000$.

Conclusion

Independent Component Analysis (ICA) is a powerful unsupervised learning method that can be used to separate source signals from a set of mixed signals. ICA assumes that each of the source signals is non-Gaussian and independent and estimates a probability distribution for each of the source signals, using Gradient Descent to effectively minimalize the the mutual information between signals. It ultimately constructs a matrix capable of separating the mixed signals into their sources. Here, I applied ICA to linear combinations of three input source signals and observed the effects of several variables on ICA efficacy. In general, some source signals can be harder to separate than others, and it is import to reach convergence during gradient descent. The learning rate can affect the speed of convergence and optimality of the final solution. Despite all of the variables at play in ICA, it works remarkably well, provided our signals loosely adhere to the non-Gaussian and independent assumptions of ICA.

References

1. Schaul, Zhang, LeCun. No More Pesky Learning Rates.
2. Ng. Independent Component Analysis