Assignment 2: Blind-Source Separation of Mixed-Input Signals through Maximum-Likelihood Independent Component Analysis

John Duncan
CS 391L: Machine Learning
Spring 2020
Instructor: Ballard

I. Introduction

This assignment explores Independent Component Analysis (ICA), a computational method which separates a linear mixture of signals into its separate components. The sources are assumed to be *blind*—that is, the independent components are not known a-priori, and it is assumed that only linear mixtures of the component signals are available.

ICA can be applied in situations where a blended signal is available, but the source signal is desired. This includes the proverbial *cocktail party problem*, in which a listener must try to discern one person's voice in a crowded, noisy room.

The remaining sections of this report are organized as follows: Section II provides formal definitions and derivations of the equations used in this project; Section III provides implementation details as well as the results of the ICA method and code. Lastly, Section IV concludes the report.

II. Experimental Method

ICA decomposes a linear mixture of signals into its independent components. Beginning with a m-by-t matrix of mixed signals, X, (where m is the number of mixed signals, and t is the number of data points in each signal), it can be expressed as a linear combination of a mixing matrix, A, and the source signals, U:

$$X = AU \tag{1}$$

where A is an m-by-n matrix of mixing coefficients, and U is n-by-t, where n is the number of independent source signals.

To extract the independent source signals, U, ICA seeks a functional inverse of A. This is hereafter referred to as W, the unmixing matrix, where:

$$WA = I \tag{2}$$

Combining 1 and 2 yields Y, the estimate of the independent source signals:

$$Y = WX = W(AU) = U \tag{3}$$

Therefore, with an unmixing matrix W, the mixed signals X can be decomposed into the independent component source signals, U.

A. Maximum-Likelihood Method for Determining Unmixing Matrix

There are several well-documented methods of determining W. This assignment uses a Maximum Likelihood (ML) method, which is relatively simple to implement. As the name suggests, ML ICA seeks to define and maximize the likelihood that the parameters of W match the mixed data X. First, define the probability of the mixture X:

$$p(x = X) = p_x(X) = p_u(U)|W| = \prod_i p_i(u_i)|W| \tag{4}$$

where $p_u(U) = p(u = U)$, the probability density of the independent source signals, U, and $p_i(u_i)$ is the probability density of the i^{th} independent component. Note that multiplication by $|W|$, the determinant of W, is required to ensure the distribution is normalized.

Substituting Equation 3 into Equation 4 and expressing W in terms of its rows $w_1^T, w_2^T, ..., w_n^T$, the probability density becomes:

$$p_x(X) = \prod_i p_i(w_i^T X)|W| \tag{5}$$

Finally, if there are t observations of X such that $x(j)$ is the j^{th} column of X, the likelihood function $\mathcal{L}(W)$ is the product of $p_x(X)$ evaluated at each observation:

$$\mathcal{L}(W) = \prod_{j=1}^{t} \prod_{i=1}^{n} p_i(w_i^T x(j))|W| \tag{6}$$

With the likelihood function defined, the next step is to maximize it.

B. Maximizing the Likelihood Function

Any maxima of Equation 6 must satisfy the first order necessary condition; i.e. $\frac{\partial}{\partial W}\mathcal{L}(W) = 0$. By observation of Equation 6, it does not appear to have easily-computed derivatives and may be computationally complex.

To simplify the derivative of the likelihood equation, its logarithm—the *log-likelihood* function—can be maximized instead. Since a logarithm is monotonic, the same W that

maximizes the log-likelihood function will also maximize the likelihood function. The log-likelihood function has more desirable algebraic properties; namely, it allows the product terms in Equation 6 to be expressed as sum terms:

$$\begin{aligned} log\mathcal{L}(W) &= \sum_{j=1}^{t}\sum_{i=1}^{n} log\left(p_i(w_i^T x(j)|W|\right) \\ &= \sum_{j=1}^{t}\sum_{i=1}^{n} log\left(p_i(w_i^T x(j)\right) + \sum_{j=1}^{t} log|W| \\ &= \sum_{j=1}^{t}\sum_{i=1}^{n} log\left(p_i(w_i^T x(j)\right) + t\left(log|W|\right) \end{aligned} \tag{7}$$

Equation 7 can be further simplified by expressing the sample sum as an expected value (i.e. the average of the observed samples) and dividing by t:

$$\frac{1}{t} log\mathcal{L}(W) = E\left[\sum_{i=1}^{n} log\left(p_i(w_i^T x(j)\right)\right] + log|W| \tag{8}$$

The derivative of Equation 8 is of the form:

$$\tfrac{1}{t}\tfrac{\partial}{\partial W} log\mathcal{L}(W) = E\left[\tfrac{\partial}{\partial W} log(g'(WX))X^T\right] + [W^T]^{[-1]}$$

where $g'(WX) = \sum_{i=1}^{n}(p_i)$. Setting $\frac{1}{t}\frac{\partial}{\partial W} log\mathcal{L}(W) = 0$ and adding W to each side yields W_{ML}, the value of the unmixing matrix which maximizes likelihood:

$$\begin{aligned} W_{ML} &= E\left[\frac{\partial}{\partial W}\left(log(g'(WX))\right)X^T\right] + [W^T]^{[-1]} + W \\ &= \Delta W + W \end{aligned} \tag{9}$$

Where W is the current a-priori estimation of the unmixing matrix. Equation 9 is commonly known as the *Bell-Sejnowski* algorithm. Unfortunately, computing a matrix inversion $[W^T]^{[-1]}$ for each iteration can become computationally expensive. Multiplying ΔW by $W^T W$ preserves the first-order necessary condition and creates a better-conditioned form of Equation 9:

$$\begin{aligned} W_{ML} &= (\Delta W)W^T W + W \\ &= \left(E\left[\frac{\partial}{\partial W}\left(log(g'(Y))\right)X^T\right] + W^{-T}\right)W^T W + W \\ &= \left(E\left[\frac{\partial}{\partial W}\left(log(g'(Y))\right)(XW)^T W\right] + W^{-T}W^T W\right) \\ &\quad + W \\ &= \left(E\left[\frac{\partial}{\partial W}\left(log(g'(Y))\right)(Y)^T W\right] + IW\right) + W \\ &= \left(E\left[\frac{\partial}{\partial W}\left(log(g'(Y))\right)(Y)^T\right] + I\right)W + W \end{aligned} \tag{10}$$

Equation 10 is the *Natural Gradient* algorithm. Recall that $Y = WX$ is the current estimate of the unmixed signals.

Up to this point, the probability density function $p(U = Y) = p_u(Y) = p_u(WX)$ and its counterpart g have not been discussed in detail. The following section selects a probability function and provides the final form of the maximum likelihood unmixing matrix W_{ML}.

C. Selecting a Probability Function

An ideal probability function $p_u(WX)$ has two qualities: easily-computed derivatives [1], and a high degree of *Kurtosis* (i.e. "tailedness" of the probability distribution). The latter is desirable to ensure that small changes in probability can be more easily related to changes in WX. Ultimately, the $\frac{\partial}{\partial W} log(g'(WX))$ term in Equation 10 must be simplified for computation.

For this assignment, we use the following function:

$$g(Y) = g(WX) = \frac{1}{1 + e^{-WX}} \tag{11}$$

which has easily-computed derivatives, and varies in value from $g(-\infty) = 0$ to $g(\infty) = 1$. $g(Y)$ serves as a cumulative distribution function, and its derivative $g'(Y) = p_u(Y)$ serves as the probability density function.

The first derivative of $g(Y)$ is computed using the quotient and chain rules as follows:

$$g'(Y) = \frac{d}{dY}g(Y) = \frac{-\frac{d}{dY}(1 + e^{-Y})}{(1 + e^{-Y})^2} = \frac{e^{-WX}}{(1 + e^{-WX})^2} \tag{12}$$

Equations 11 and 12 can be combined and rearranged to show that:

$$g'(Y) = g(Y)(1 - g(Y)) \tag{13}$$

Lastly, the $\frac{\partial}{\partial W} log(g'(WX))$ term can be evaluated using Equation 13 (dropping the (WX) notation for brevity):

$$\begin{aligned} \frac{\partial}{\partial W}\left(log(g')\right) &= \frac{\partial}{\partial W}\left(log(g(1-g))\right) \\ &= \frac{\partial}{\partial W} log(g) + \frac{\partial}{\partial W} log(1-g) \\ &= \frac{g'}{g} + \frac{(g')(-1)}{(1-g)} = \frac{g'(1-g)}{g(1-g)} + \frac{g(g')(-1)}{g(1-g)} \\ &= \frac{g'}{g(1-g)}\left((1-g) - (g)\right) \\ &= 1 - 2g \end{aligned} \tag{14}$$

Substituting Equation 14 into Equation 10 yields the final form of the Natural Gradient equation:

[1] Since the likelihood is maximized by computing the first derivative

$$W_{ML} = \left(E\left[\frac{\partial}{\partial W} \left(log(g'(WX)) \right) (Y)^T \right] + I \right) W + W$$
$$= \left(E\left[(1 - 2g(WX)) (Y)^T \right] + I \right) W + W \tag{15}$$

III. Implementation & Results

The method derived in Section II was implemented in a Python script, and is included as an attachment.

First, the provided *sounds.mat* file was imported, which provided five independent signals U. These independent signals were mixed with a randomized, n-by-n matrix A to generate a matrix of mixed signals, X as in Equation 1. Plots of these signals were saved, and .wav sound files were generated for each of the original source signals. Qualitatively, the five source signals are:

- U_0.wav: Homer Simpson
- U_1.wav: A vacuum cleaner
- U_2.wav: Applause
- U_3.wav: Laughter
- U_4.wav: Crackling

These signals and their mixes are shown in Figure 1.

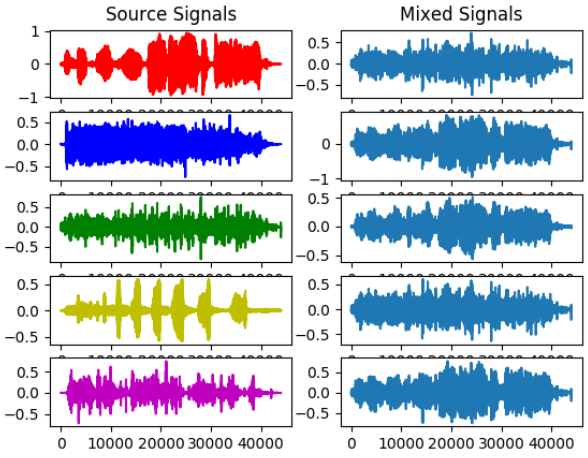

Fig. 1. Original and mixed signals.

Next, a randomized n-by-m initial W matrix was generated. The initial W matrix was propagated forward with a stepsize $\alpha = .01$ for $i = 1000, 3000, 5000, 10000, 30000, 50000, 100000$ iterations. The final unmixing matrices W_i were then used to reconstruct signals: $Y_i = W_i X$. Reconstructed signals are shown in Figures 2 and 3, respectively, for 10,000 and 100,000 computations.

Qualitatively, the recovered signals become more distinct as the number of computations increases. For lower numbers of computations as in Figure 2, the unmixed signals look more like the mixed signals in Figure 1. For higher numbers of computations (Figure 3), the reconstructed signals more closely approximate the source signals in Figure 1.

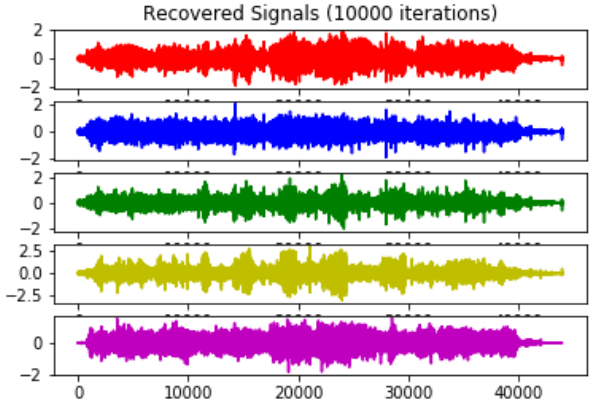

Fig. 2. Reconstructed signals (10,000 computations, $\alpha = .01$)

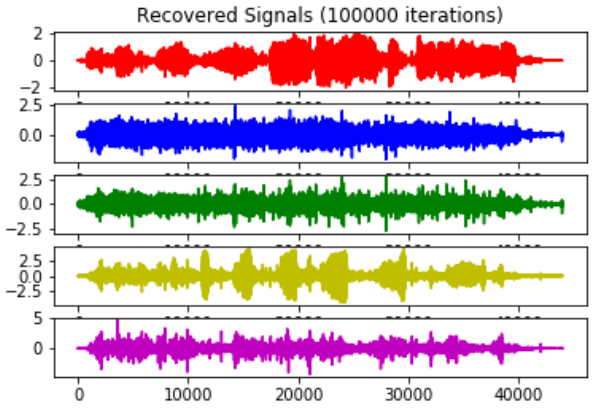

Fig. 3. Reconstructed signals (100,000 computations, $\alpha = .01$)

A more quantitative analysis was performed by computing the correlation between reconstructed and original signals, as shown in Figure 4. The *Homer*, *Laughter*, and *Vacuum* signals had the highest correlation with unmixed signals, and correlation increased as number of calculations increased. *Applause* and *Crackling* had the lowest correlation with any of the unmixed signals, and *Crackling*'s maximum correlation decreased as computations increased.

This suggests that while the number of computations can affect the correlation with the reconstructed signal, the nature of the signal matters as well. *Homer* and *Laughter* are distinct and percussive, while *Vacuum*, *Applause*, and *Crackling* are more noisy and steady, making it difficult to extract the independent data from their mixtures.

The correlations are shown in a matrix in Table I for 10,000 computations. An ideal correlation—one where the mutual information was removed from all reconstructed signals—

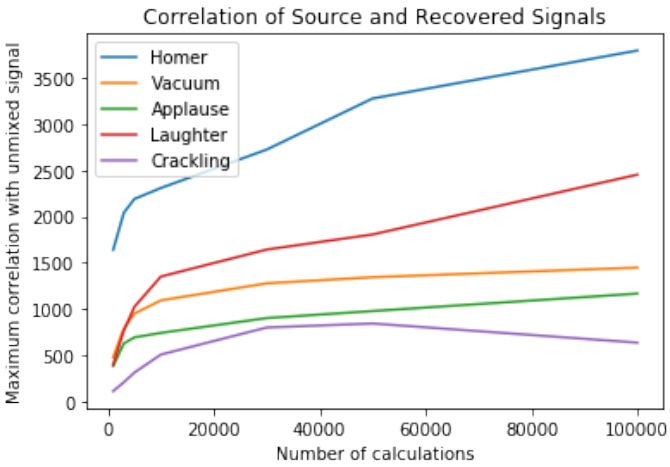

Fig. 4. Correlation between reconstructed and original signals.

would have positive correlation in the diagonal terms, and zero or negative values in the off-diagonal terms. Instead, several of the reconstructed signals have high correlation with *multiple* of the source signals. For example, the reconstructed signal Y_4 correlates positively with *Homer* and *Vacuum*, meaning that it still contains a mixture of both.

TABLE I
CORRELATION BETWEEN SOURCE AND RECOVERED SIGNALS, 10,000 COMPUTATIONS

Source Signal	**Source & Reconstructed Signal Correlation**				
	Y_0	Y_1	Y_2	Y_3	Y_4
Homer	2308	-924	9	1375	2081
Vacuum	-506	1093	-574	114	1093
Applause	627	718	742	-689	-10
Laughter	-654	246	695	1351	214
Crackling	205	148	152	508	-240

IV. CONCLUSION

This assignment explored the derivation, implementation, and results of a Maximum-Likelihood Independent Component Analysis method. Five independent source signals were mixed and unmixed. Correlation of source signals with reconstructed signals was computed. The properties of a signal as well as the number of computations performed affect the correlation of the reconstructed signal.

Depending on the application, another type of ICA, such as singular value decomposition or InfoMax, may yield better results. Likewise, pre-processing or ”whitening” of data (e.g. by performing a principle component analysis before ICA) may improve results.

V. ACKNOWLEDGEMENTS

The information in this report was obtained from the course notes and textbook, as well as the following websites:
https://www.mv.helsinki.fi/home/amoaning/movies/uml/ica_handout3.pdf
https://en.wikipedia.org/wiki/Independent_component_analysis