

BLIND SOURCE SEPARATION VIA ICA: MATH BEHIND METHOD*

Akshay Dayal
John Steinbauer
Angela Qian
Mark Eastaway

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 2.0[†]

Abstract

This module describes the math behind how our project on blind source separation using ICA works.

1 The Math of ICA

The Independent Components Analysis algorithm allows two source signals to be separated from two mixed signals using statistical principles of independence and nongaussianity.

1.1 Defining the Problem

ICA assumes that the value of each source at any given time is a random variable. It also assumes that each source is statistically independent, meaning that the values of one source cannot be correlated to values in any of the other sources.

With these assumptions, ICA allows us to separate source signals from mixtures of these source signals. The algorithm requires that there be as many sensors as input signals. For example, with three independent sources and three mixtures being recorded, the problem could be modeled as:

$$\begin{aligned}x_1(t) &= as_1(t) + bs_2(t) \\x_2(t) &= cs_1(t) + ds_2(t)\end{aligned}\tag{1}$$

Using matrix notation, the problem can be generalized to any number of mixtures. For some number of sources n to be identified, n mixtures would need to be recorded.

$$x = As\tag{2}$$

*Version 1.1: Dec 19, 2007 8:41 pm +0000

[†]<http://creativecommons.org/licenses/by/2.0/>

The goal of blind source separation using ICA is to invert this procedure; that is, given the mixtures \mathbf{x} as inputs, ICA finds \mathbf{s} . Because the mixing matrix \mathbf{A} is square, we can write the reverse procedure as

$$x = A^{-1}s \quad (3)$$

or, if we define \mathbf{W} to be equal to the inverse of \mathbf{A} ,

$$x = Ws \quad (4)$$

is an equivalent expression of the problem at hand.

1.2 Isolating the Independent Sources

The Central Limit Theorem provides the key to unlocking the mystery matrix \mathbf{W} . The central limit theorem says that a sum of independent random variables can be approximated by a normal curve. The greater the number of variables summed, the more normal, or gaussian, the distribution. Since each of the mixtures being received by the sensors represents a linear combination of samples from each source in \mathbf{s} , the distribution of the mixed signals is more gaussian than either of the two independent sources (by the central limit theorem).

In order for the ICA algorithm to use this principle, the algorithm needs a way of determining how gaussian a particular signal is. There are two main quantitative measures of nongaussianity. The first of these measures is kurtosis, which measures the “spikiness” of a signal. Kurtosis is 0 for gaussian random variables, positive for random variables that are more spiky than gaussian variables, and negative for random variables that are flatter than gaussian variables. The second measure is negentropy, which measures the “simplicity” of a signal. Negentropy is also 0 for gaussian random variables.

The FastICA package that we used for our project uses both of these measures of nongaussianity to identify independent source signals. It begins by guessing a row of the matrix \mathbf{W} , which we can call \mathbf{w} . This row represents the weighting coefficients for finding one of the original source signals. It then measures the nongaussianity of the proposed independent source defined by its guess of \mathbf{w} , and finds the gradient of nongaussianity in an n -dimensional space to determine how the coefficients in \mathbf{w} should change. It then uses a projection of the gradient to create a new guess of the coefficients in \mathbf{w} , and continues in a cycle until the coefficients converge on certain values. Once this occurs, the resulting independent source is as nongaussian as it can be. This in turn means that it is the furthest the algorithm could get from the source being a sum, which means that one of the independent sources has been isolated.

The algorithm repeats this process for finding all the rest of the independent sources, taking care not to find the same source twice.

1.3 Ambiguities and Limitations

Just by examining the statement of the problem at the beginning of this module, two significant ambiguities arise in the ICA algorithm.

1.3.1 Indeterminate Energy

Because a scalar multiplier could be pulled out of \mathbf{s} and multiplied to \mathbf{A} with no change in the above equations, the ICA algorithm cannot determine the energy contained in any of the independent sources it finds. The amplitudes it gives the output components are arbitrary, and the true source signal could be one the isolated sources multiplied by any scalar multiple. This includes a negative multiple, which means that often, the output signals are also inversions of the original signals.

The seriousness of this ambiguity depends on the application. For sound signals, inversions are irrelevant because the only important part of the signal is the different between voltages, not the polarity. Gain can also be added on to sound systems to deal with the amplitude ambiguity. In other applications, such as image processing, the inability to distinguish energy is much more significant.

1.3.2 Order Ambiguity

Because the algorithm chooses coefficients of \mathbf{w} at random when it searches for the sources, the isolated sources that the algorithm finds can come out in any order. So, it would take some additional processing to determine which independent source is the one of interest to you.

1.3.3 Under-Determination

There *must* be as many sensors as there are sources in order to properly isolate the sources. If there are not enough sensors, the resulting signals will not match any of the sources, but rather will still be mixtures of multiple sources.

1.3.4 Under-Determination

ICA can only handle linear mixtures that can be represented in the form $\mathbf{x} = \mathbf{A}\mathbf{s}$. The algorithm cannot accurately guess the independent sources if the sources are out of phase in the mixtures or if the mixtures have other nonlinear features.