Independent Component Analysis

1st Venkata Ramkiran Balivada
CDS
Indian Institute of Science
Bangalore, India
ramkiranb@iisc.ac.in

2nd Vaddadi Venkatesh
CDS
Indian Institute of Science
Bangalore, India
venkateshvad@iisc.ac.in

***Abstract*—Independent component analysis (ICA) is a method in which the goal is to find a linear representation of non-gaussian data so that the components are statistically independent, or as independent as possible. It is very useful in capturing the essential structure of the data in many applications, including feature extraction and signal separation.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Imagine that you are in a room where two people are speaking simultaneously. You have two microphones, which you hold in different locations. The microphones give you two recorded time signals, which we could denote by $x_1(t)$ and $x_2(t)$, with x_1 and x_2 the amplitudes, and t is the time index. Each of these recorded signals is a weighted sum of the actual source signals from two speakers, which we denote by $s_1(t)$ and $s_2(t)$.

We could express this as a linear equation:

$$x_1(t) = a_{11}s_1 + a_{12}s_2$$
$$x_2(t) = a_{21}s_1 + a_{22}s_2$$

Here, a_{11}, a_{12}, a_{21} , and a_{22} are some parameters that depend on the distances of the microphones from the speakers.

It would be very useful if you could now estimate the two original speech signals $s_1(t)$ and $s_2(t)$, using only the recorded signals $x_1(t)$ and $x_2(t)$. This is called the cocktail-party problem.

For the time being, we omit any time delays or other extra factors from our simplified mixing model. As an illustration, consider the waveforms in Fig:1 and Fig:2. These are, of course, not realistic speech signals, but suffice for this illustration.

The original speech signals could look something like those in Fig. 1 and the mixed signals could look like those in Fig. 2. The problem is to recover the data in Fig. 1 using only the data in Fig. 2.

Actually, if we knew the parameters a_{ij} , we could solve the linear equation in (1) by classical methods. The point is, however, that if you don't know the a_{ij} , the problem is considerably more difficult.

Identify applicable funding agency here. If none, delete this.

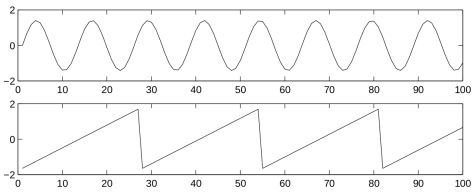

Fig. 1. Original Signals

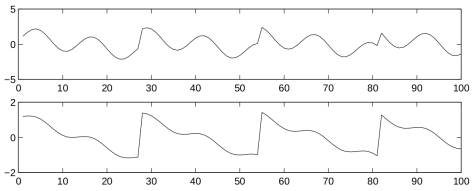

Fig. 2. Observed signals

One approach to solving this problem would be to use some information on the statistical properties of the signals $s_i(t)$ to estimate the a_{ij} . Actually, and perhaps surprisingly, it turns out that it is enough to assume that $s_1(t)$ and $s_2(t)$, at each time instant t, are statistically independent.

This is not an unrealistic assumption in many cases, and it need not be exactly true in practice.

The technique Independent Component Analysis, or ICA, can be used to estimate the a_{ij} based on the information of their independence, which allows us to separate the two original source signals $s_1(t)$ and $s_2(t)$ from their mixtures $x_1(t)$ and $x_2(t)$.

Fig:3 gives the two signals estimated by the ICA method. We can see in the Fig:3 the signals which are recovered that are closer to the original signals.

Independent component analysis was originally developed to deal with problems that are closely related to the cocktail-party problem. Since the recent increase of interest in ICA, it has become clear that this principle has a lot of other interesting applications as well. Consider, for example, electrical recordings of brain activity as given by an electroencephalogram (EEG).

The EEG data consists of recordings of electrical potentials

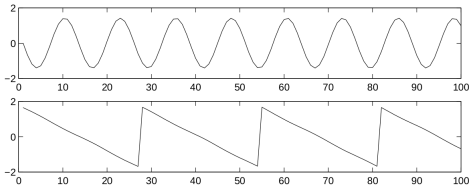

Fig. 3. Estimated signals

in many different locations on the scalp. These potentials are presumably generated by mixing some underlying components of brain activity. This situation is quite similar to the cocktail-party problem: we would like to find the original components of brain activity, but we can only observe mixtures of the components.

ICA can reveal interesting information on brain activity by giving access to its independent components.

Another, very different application of ICA is on feature extraction. A fundamental problem in digital signal processing is to find suitable representations for image, audio or other kind of data for tasks like compression and denoising.

ICA can be used as a very general-purpose method of signal processing and data analysis. All of the applications that are described in the above can actually solve with ICA.

II. Independent Component Analysis

A. Definition

Let us assume we have n independent components and n- linear mixtures of independent components. $x_1, ..., x_n$ are those linear mixtures.

$$x_j = a_{j1}s_1 + a_{j2}s_2 + ... + a_{jn}s_n \text{ , for all j.}$$

We assume that each mixture x_j as well as each independent component s_k is a random variable.

The observed values $x_j(t)$ are then a sample of this random variable. Without loss of generality, we can assume that both the mixture variables and the independent components have zero mean: If this is not true, then the observable variables x_i can always be centered by subtracting the sample mean, which makes the model zero-mean. For convenience, we can represent above summing notations as a vector matrix notation.

Let us denote by x the random vector whose elements are the mixtures $x_1, ..., x_n$, and likewise by s the random vector with elements $s_1, ..., s_n$. Let us denote by A the matrix with elements a_{ij}.

All vectors are understood as column vectors; thus x^T , or the transpose of x, is a row vector. Using this vector-matrix notation, the above mixing model is written as $x = As$.

The starting point for ICA is the very simple assumption that the components s_i are statistically independent. Independent components must have non-gaussian distributions. However, in the basic model we do not assume these distributions are known. For simplicity, we are also assuming that the unknown mixing matrix is square.

Then, after estimating the matrix A, we can compute its inverse, say W, and obtain the independent component simply by: $s = Wx$.

ICA is very closely related to the method called blind source separation (BSS). A "source" means here an original signal, i.e. independent component, like the speaker in a cocktail party problem. "Blind" means that we know very little, if anything, on the mixing matrix, and make little assumptions on the source signals. ICA is the most widely used method for Blind Source Separation.

B. Principles of ICA estimation

The key to estimating the ICA model is nongaussianity. Actually, without non-gaussianity the estimation is not possible at all.

In most of classical statistical theory, random variables are assumed to have gaussian distributions, those things prevent any methods related to ICA.

The Central Limit Theorem, a classical result in probability theory, tells that the distribution of a sum of independent random variables tends toward a gaussian distribution, under certain conditions. Thus, a sum of two independent random variables usually has a distribution that is closer to gaussian than any of the two original random variables.

Let us now assume that the data vector x is distributed according to the ICA data model. Therefore, it is a mixture of independent components. For simplicity, let us assume in this section that all the independent components have identical distributions. To estimate one of the independent components, we consider a linear combination of the x_i.

Let us denote this y as following to determine vector w.

$$y = w^T x = \sum_i w_i x_i$$

If w were one of the rows of the inverse of A, this linear combination would actually equal one of the independent components.

But the hidden thing is " How could we use the Central Limit Theorem to determine w?"

In practice, we cannot determine such a w exactly, because we have no knowledge of matrix A, but we can find an estimator that gives a good approximation. To see how this leads to the basic principle of ICA estimation. Let us make a change of variables, by defining z as following:

$$z = A^T w.$$

The above equation makes the changes to y as following:

$$y = w^T x = w^T As = z^T s$$

Since, y is a linear combination of s_i (with weights given by z_i). Since a sum of even two independent random variables is more gaussian than the original variables, $z^T s$ is more gaussian than any of the s_i and becomes least gaussian when it in fact equals one of the s_i.

In this case, obviously only one of the elements z_i of z is nonzero. (Note that the s_i were here assumed to have identical distributions.) Therefore, we could take as w as a vector that

maximizes the nongaussianity of $w^T x$. Such a vector would necessarily correspond to a z which has only one nonzero component.

This means that $w^T x = z^T s$ equals one of the independent components.

Maximizing the nongaussianity of $w^T x$ thus gives us one of the independent components.

III. Measures of Non-Gaussianess

A. Kurtosis

Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution. That is, data sets with high kurtosis tend to have heavy tails, or outliers. Data sets with low kurtosis tend to have light tails, or lack of outliers. A uniform distribution would be the extreme case.

The classical measure of non-gaussianity is kurtosis or the fourth-order cumulant. The kurtosis of y is classically defined as following:

$$Kurt(y) = E(y^4) - 3(E(y^2))^2$$

Actually, since we assumed that y is of unit variance. Therefore, $Kurt(y) = E(y^4) - 3$. Kurtosis is basically a normalized version of the fourth moment. kurtosis is zero for a gaussian random variable. For most non-gaussian random variables, kurtosis is nonzero. Kurtosis can be both positive or negative. Random variables that have a negative kurtosis are called sub-gaussian, and those with positive kurtosis are called super-gaussian.

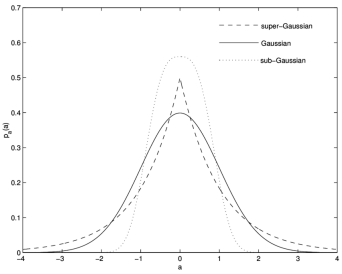

Fig. 4. Guassian vs Sub-Guassian vs Super-Guassian

Typically non-gaussianity is measured by the absolute value of kurtosis. The square of kurtosis can also be used. These are zero for a gaussian variable, and greater than zero for most nong-aussian random variables. There are non-gaussian random variables that have zero kurtosis, but they can be considered as very rare. Kurtosis, or rather its absolute value, has been widely used as a measure of nongaussianity in ICA and related fields. The main reason is its simplicity, both computational and theoretical. Computationally, kurtosis can be estimated simply by using the fourth moment of the sample data. Theoretical analysis is simplified because of the following linearity property: If x_1 and x_2 are two independent random variables, it holds following:

$$kurt(x1 + x2) = kurt(x1) + kurt(x2)$$
$$kurt(\alpha x1) = \alpha^4 kurt(x1)$$

However, kurtosis has some drawbacks in practice, when its value has to be estimated from a measured sample. The main problem is that kurtosis can be very sensitive to outliers. Its value may depend on only a few observations in the tails of the distribution, which may be erroneous or irrelevant observations. In other words, kurtosis is not a robust measure of nongaussianity.

B. Negentropy

A second very important measure of nongaussianity is given by negentropy. Negentropy is based on the information theoretic quantity of entropy. Entropy is the basic concept of information theory. The entropy of a random variable can be interpreted as the degree of information that the observation of the variable gives. The more "random", i.e. unpredictable and unstructured the variable is, the larger its entropy.

Entropy H is defined for a discrete random variable Y as following:

$$H(y) = -\sum_i p(y = a_i) log(p(y = a_i))$$

where, the a_i are the possible values of Y .

A fundamental result of information theory is that a gaussian variable has the largest entropy among all random variables of equal variance. This means that entropy could be used as a measure of nongaussianity. It means gaussian distribution is the least structured of all distributions.

To obtain a measure of nongaussianity that is zero for a gaussian variable and always nonnegative, one often uses a slightly modified version of the definition of differential entropy, called negentropy. Negentropy J is defined as follows:

$$J(y) = H(y_{guass}) - H(y)$$

Negentropy is always non-negative, and it is zero if and only if y has a Gaussian distribution. Negentropy has the additional interesting property that it is invariant for invertible linear transformations.

C. Mutual Information

Mutual information is a measure of the information that members of a set of random variables have on the other random variables in the set. Using entropy, we can define the mutual information I between m (scalar) random variables, $y_1, y_2, ..., y_m$ as follows:

$$I(y_1, y_2, y_3.....y_m) = \sum_i^m H(y_i) - H(y)$$

Mutual information is a natural measure of the dependence between random variables. It is always non-negative, and zero if and only if the variables are statistically independent. Thus, mutual information takes into account the whole dependence structure of the variables, and not only the covariance, like PCA and related methods.

An important property of mutual information is that we have for an invertible linear transformation $y = Wx$:

$$I(y_1, y_2, y_3.....y_m) = \sum_i^m H(y_i) - H(y) - log(det(W))$$

Now, let us consider what happens if we constrain the y_i to be uncorrelated and of unit variance. It implies the following things:

$$E(yY^T) = wE(xx^T)w^T = I$$

$$det(WE(xx^T)W^T) = det(W)det(E(xx^T))det(W^T) = det(I) = 1$$

From the above thing, it concludes detW must be constant. Moreover, for y_i of unit variance, entropy and negentropy differ only by a constant, and the sign. The following shows the fundamental relationship between mutual information and negentropy.

$$I(y_1, y_2, y_3.....y_m) = c - \sum_i^m J(y_i)$$

Since mutual information is the natural information-theoretic measure of the independence of random variables, we could use it as a measure for ICA. We define the ICA of a random vector x as an invertible transformation as $s = Wx$, where the matrix W is determined so that the mutual information of the transformed components s_i is minimized.

From the above equation, finding an invertible transformation W that minimizes the mutual information is roughly equivalent to finding directions in which the negentropy is maximized.

More precisely, it is roughly equivalent to finding 1-D subspaces such that the projections in those subspaces have maximum negentropy.

Minimization of mutual information is equivalent to maximizing the sum of non-gaussianities of the estimates, when the estimates are constrained to be uncorrelated. The constraint of uncorrelatedness is in fact not necessary, but simplifies the computations considerably. Therefore, the formulation of ICA as minimization of mutual information leads to finding maximally non-gaussian directions.

IV. Implementation

Before applying an ICA algorithm on the data, it is usually very useful to do some preprocessing. In this section, we discuss some preprocessing techniques that make the problem of ICA estimation simpler and better .

A. Centering

The most basic and necessary preprocessing is to center x, i.e. subtract its mean vector $m = E\{x\}$. Therefore, x will become a zero-mean vector. This preprocessing is made solely to simplify the ICA algorithms: It does not mean that the mean could not be estimated. After estimating the mixing matrix A with centered data, we can complete the estimation by adding the mean vector of s back to the centered estimates of s. The mean vector of s is given by $A^{-1}m$, where m is the mean that was subtracted in the preprocessing.

B. Whitening

Another useful preprocessing strategy in ICA is to first whiten the observed variables. This means that before the application of the ICA algorithm (and after centering), we transform the observed vector x linearly so that we obtain a new vector x which is white, i.e. its components are uncorrelated and their variances equal unity. In other words, the covariance matrix of x equals the identity matrix:

$$E(\tilde{x}\tilde{x}^T) = I$$

The whitening transformation is always possible. One popular method for whitening is to use the eigen-value decomposition (EVD) of the covariance matrix.

where, E is the orthogonal matrix of eigen vectors of $E\left\{xx^T\right\}$ and D is the diagonal matrix of its eigenvalues.

$$D = diag(d_1, ..., d_n).$$

Note that $E\left\{xx^T\right\}$ can be estimated in a standard way from the available sample $x_1, ..., x_m$. Whitening can now be done by

$$\tilde{x} = ED^{-\frac{1}{2}}E^T x$$

where the matrix $D^{-\frac{1}{2}}$ is computed by a simple component-wise operation as $D^{-\frac{1}{2}}$ = diag($d_1{}^{-\frac{1}{2}}$,.....$d_m{}^{-\frac{1}{2}}$) is easy to check that now $E(\tilde{x}\tilde{x}^T) = I$

Whitening transforms the mixing matrix into a new one, $\tilde{A}$.
$\tilde{x} = ED^{-\frac{1}{2}}E^T As = \tilde{A}s$

V. The FastICA Algorithm

In the preceding sections, we introduced different measures of non-gaussianity, i.e. objective functions for ICA estimation. In practice, one also needs an algorithm for maximizing the contrast function like the following.

$$J(y) = \sum_{i=1}^p K_i \left[E\{G_i(y)\} - E\{G_i(v)\}\right]^2$$

In this section, we introduce a very efficient method of maximization suited for this task. It is here assumed that the data is preprocessed by centering and whitening.

A. Fast-ICA for one unit

This section illustrates the one-unit version of FastICA. The FastICA learning rule finds a direction, i.e. a unit vector w such that the projection $w^T x$ maximizes nongaussianity. Non-gaussianity is here measured by the approximation of negentropy $J(w^T x)$. The variance of $w^T x$ must here be constrained to unity; for whitened data this is equivalent to constraining the norm of w to be unity. The FastICA is based on a fixed-point iteration scheme for finding a maximum of the nongaussianity of $w^T x$. The following functions G1 or G2 are used (g) to approximate the input signal distribution in the algorithm.

$$G_1(u) = \frac{1}{a_1} log(cosh(a_1 u)))$$

$$G_2(u) = -exp(\frac{u^2}{2})$$

where, $1 \leq a_1 \leq 2$ is some suitable constant, often taken as $a_1 = 1$.

The basic form of the FastICA algorithm is as follows:

1) Choose an initial (e.g. random) weight vector w.
2) Find $w^+ = E\left\{xg(w^T x)\right\} - E\left\{g^{'}(w^T x)\right\} w$
3) Update $w = \frac{w^+}{\|w^+\|}$
4) If not converged, go back to 2.

Here, the convergence means the old and new values of w point in the same direction, i.e. their dot-product is (almost) equal to 1. Convergence may happen to either w or $-w$. This is again because the independent components can be defined only up to a multiplicative sign. It is also assumed that the data is pre-whitened.

The one-unit vector independent component algorithm estimates just one of the independent components. To estimate several independent components, we need to run the one-unit FastICA algorithm using several units with weight vectors $w_1, ..., w_n$ parallely. This is done by Gram-Smith orthogonalization of the estimated vector at each iteration.

VI. Results

Using ICA we tried to experiment with 2 different problem statements
1. Audio source separation from mixed signals
2. Finding basis images on MNIST hand written digit data using ICA

A. Audio Source Separation

In this problem 2 independent sources (S1 ans S2) are mixed into 2 different mixtures and these 2 mixed signals are available as measurements (X1 and X2). Our job here is to extract the source signals from the measured data. ICA is applied on the measured data to find the source signals. Fig. 5 illustrates the source distributions of the signal. One can clearly see that these distributions are non-Gaussian with flat tails (sub-Gaussian). Fig. 6 illustrates the convergence of the ICA algorithm. Left plot shows the increase in entropy while the right plot shows the decrease in gradient magnitude proving we reached the maximum entropy. In Fig. 7 first row shows the source signals, second rwo shows the mixed signals and third row shows the unmixed signals after applying ICA.

B. Basis images on MNIST

In this problem we tried to find the independent components that generated the MNIST images. This is equivalent to unsupervised learning of Neural Networks where independent classes are found using ICA. We generated 25 independent components on the MNIST dataset as shown in the Fig. 8. These independent components are placed in a row major order from 0 to 24. Fig. 9 compares the performance of the ICA independent component images against the actual classes. Each row describes the weight a particular class label (0-9) gives to all the independent components. We can clearly see that there is a specific component that generates specific label represented by the yellow square. Note in Fig. 8 lighter color represents higher weight and vice versa.

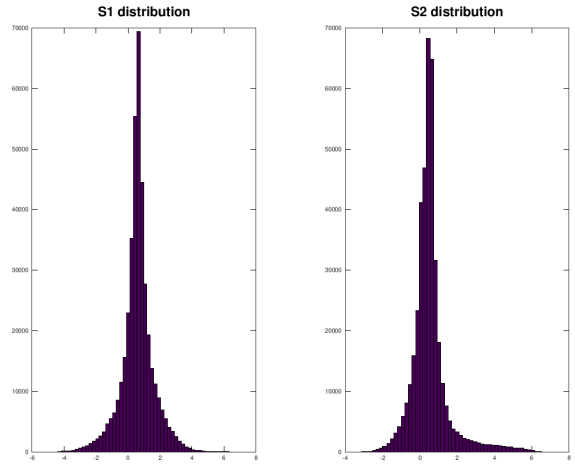

Fig. 5. Source Signal Histograms

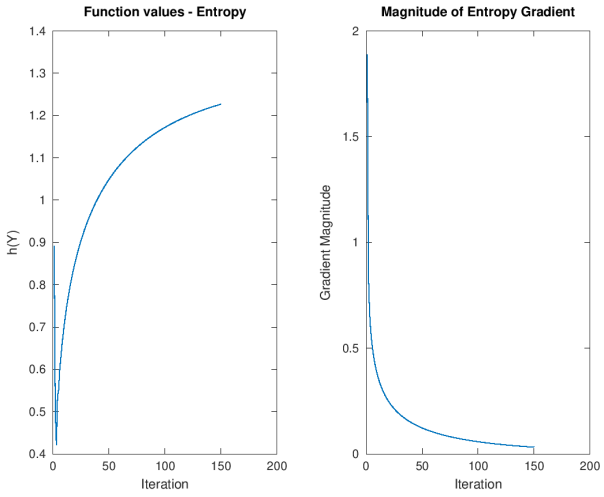

Fig. 6. ICA Convergence

References

[1] Hyvärinen, Aapo, and Erkki Oja. *"Independent component analysis: algorithms and applications."*. Neural networks 13.4-5 (2000): 411-430.
[2] Marc Beltrán Segarra. *"STUDY OF RECONSTRUCTION ICA FOR FEATURE EXTRACTION IN IMAGES AND SIGNALS"*. 2017.
[3] Erkki Oja Aapo Hyvrinen, Juha Karhunen. *"Independent Component-Analysis. "* J. Wiley, 1 edition, 2001.

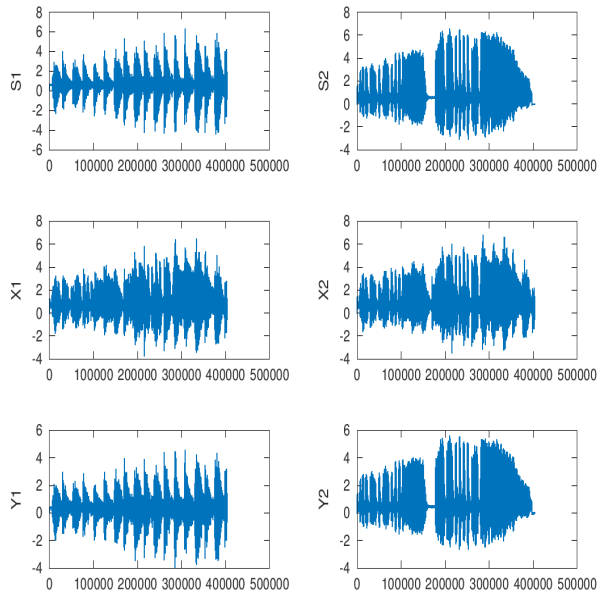

Fig. 7. Signals

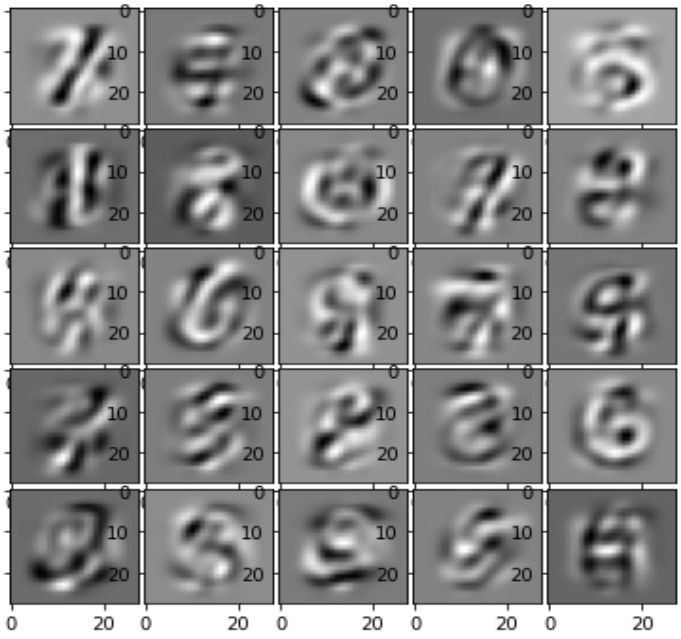

Fig. 8. Independent Component Images

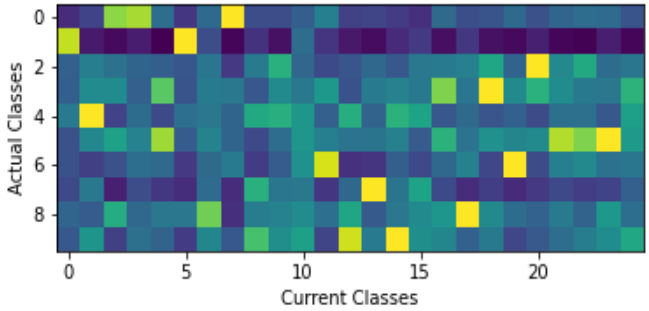

Fig. 9. Actual Classes vs ICA Classes